



Vysoká škola báňská – Technická univerzita Ostrava



POČÍTAČOVÉ SYSTÉMY

učební text

Marek Babiuch

Ostrava 2007

Recenze: Ing. Pavel Buřil, Prof. RNDr. Alena Lukasová, CSc.

Název: Počítačové systémy
Autor: Marek Babiuch
Vydání: první, 2007
Počet stran: 243

Vydavatel a tisk: Ediční středisko VŠB – TUO

Studijní materiály pro studijní obor Automatické řízení a inženýrská informatika na fakultě strojní

Jazyková korektura: nebyla provedena.

Určeno pro projekt:

Operační program Rozvoj lidských zdrojů

Název: E-learningové prvky pro podporu výuky odborných a technických předmětů

Číslo: CZ.O4.01.3/3.2.15.2/0326

Realizace: VŠB – Technická univerzita Ostrava

Projekt je spolufinancován z prostředků ESF a státního rozpočtu ČR

© Marek Babiuch

© VŠB – Technická univerzita Ostrava

ISBN 978-80-248-1503-9

Obsah

POKYNY KE STUDIU	5
1. HARDWAROVÉ KOMPONENTY PC	8
1.1 ZÁKLADNÍ TERMINOLOGIE.....	8
1.2 PRINCIP ČINNOSTI POČÍTAČE	10
1.3 ZÁKLADNÍ DESKY	10
1.4 SLOTSY A SOCKETY.....	12
2. ÚVODNÍ SEZNÁMENÍ S PROSTŘEDÍM MS VISUAL C++ 6.0	18
2.1 ZALOŽENÍ PROJEKTU VE VISUAL C++ 6.0.....	18
2.2 ORIENTACE VE VÝVOJOVÉM PROSTŘEDÍ.....	19
2.3 PSÁNÍ ZDROJOVÉHO KÓDU	21
2.4 KOMPILACE A SPUŠTĚNÍ PROGRAMU	22
2.5 VYHLEDÁVÁNÍ CHYB A JEJICH OPRAVA.....	24
2.6 NÁPOVĚDA MSDN	25
3. LADĚNÍ VLASTNÍHO PROGRAMU VE VISUAL STUDIU 6.0	29
3.1 NASTAVENÍ LADĚNÍ PROGRAMU.....	29
3.2 KROKOVÁNÍ PROGRAMU	30
3.3 KROKOVÁNÍ DOVNITŘ FUNKCÍ A VEN Z FUNKCÍ.....	31
3.4 ZARÁŽKY (BREAKPOINTY)	32
3.5 ZARÁŽKA S PODMÍNKOU.....	34
3.6 SLEDOVÁNÍ HODNOT PROMĚNNÝCH.....	36
4. ZÁKLADY JAZYKA C.....	41
4.1 ÚVOD DO JAZYKA C.....	41
4.2 ZÁKLADY JAZYKA	43
4.3 ZÁKLADNÍ TYPY DAT.....	44
4.4 KONSTANTY A PROMĚNNÉ	45
4.5 OPERAND, OPERÁTOR, VÝRAZ	47
4.6 FUNKCE STANDARDNÍHO VSTUPU A VÝSTUPU	50
4.7 ŘÍDICÍ STRUKTURY	53
4.8 PREPROCESOR	60
4.9 FUNKCE.....	62
4.10 UKAZATELE	64
4.11 POLE	65
4.12 ŘETĚZCE	67
4.13 PRÁCE SE SOUBORY	68
4.14 ARGUMENTY PŘÍKAZOVÉHO ŘÁDKU.....	72
4.15 DEKLARACE ODVOZENÝCH A STRUKTUROVANÝCH DATOVÝCH TYPŮ.....	72
5. ARCHITEKTURA ČIPOVÉ SADY.....	78
5.1 ZÁKLADNÍ POJMY V OBLASTECH ČIPOVÝCH SAD.....	78
5.2 VÝVOJ ČIPOVÉ SADY	79

5.3	ČIPOVÉ SADY ZALOŽENÉ NA ARCHITEKTUŘE ROZBOČOVAČŮ	82
5.4	SOUČASNÝ VÝVOJ ČIPOVÝCH SAD	85
6.	PROCESORY I	93
6.1	STRUČNÁ HISTORIE	93
6.2	SBĚRNICE PROCESORŮ	93
6.3	RYCHLOST PROCESORŮ A ZÁKLADNÍ DESKY	95
6.4	INSTRUKČNÍ SADA	97
6.5	VÝROBA PROCESORU	99
6.6	POUZDRA PROCESORU	101
6.7	CHLAZENÍ PROCESORU	104
7.	PROCESORY II.....	109
7.1	PROCESORY PÁTÉ GENERACE	109
7.2	PROCESORY ŠESTÉ GENERACE.....	112
7.3	PROCESORY SEDMÉ GENERACE.....	114
7.4	PROCESORY OSMÉ GENERACE	118
7.5	DUALCORE A SOUČASNOST (ROK 2007)	122
8.	FUNKCE A VLASTNOSTI SBĚRNIC.....	127
8.1	ROZDĚLENÍ SBĚRNIC PC.....	127
8.2	LOKÁLNÍ SBĚRNICE	128
8.3	AGP PORT	132
8.4	PCI EXPRESS.....	134
9.	VSTUPNĚ/VÝSTUPNÍ PORTY	140
9.1	SÉRIOVÝ VERSUS PARALELNÍ PŘENOS	140
9.2	USB A FIREWIRE PORT	140
9.3	SÉRIOVÝ PORT	141
9.4	TYPY SÉRIOVÝCH ZAŘÍZENÍ.....	143
9.5	SYNCHRONIZACE V ASYNCHRONNÍ KOMUNIKACI.....	145
9.6	PŘÍSTUP K SÉRIOVÉMU PORTU.....	145
9.7	PARALELNÍ PORT	146
9.8	STANDARDNÍ POUŽITÍ PARALELNÍHO PORTU – SPOJENÍ S TISKÁRNOU	147
9.9	PROGRAMOVÁNÍ S PARALELNÍM PORTEM	148
9.10	PŘÍSTUP K PARALELNÍMU PORTU	149
10.	PAMĚTI	155
10.1	ROZDĚLENÍ PAMĚTÍ.....	155
10.2	POLOVODIČOVÉ PAMĚTI	155
10.3	PAMĚŤOVÉ ČIPY RAM	158
10.4	FYZICKÉ USPOŘÁDÁNÍ PAMĚTI (PAMĚŤOVÉ MODULY)	164
11.	PEVNÉ DISKY, ROZHRAŇÍ A STANDARDY	170
11.1	VÝVOJ ROZHRAŇÍ PEVNÝCH DISKŮ.....	170
11.2	VERZE ROZHRAŇÍ ATA	172

11.3	ROZHRANÍ SERIAL ATA	177
11.4	ROZHRANÍ SCSI.....	178
11.5	ZÁKLADNÍ SOUČÁSTI PEVNÝCH DISKŮ	179
11.6	STRUKTURA SOUBOROVÉHO SYSTÉMU.....	184
12.	GRAFICKÝ SUBSYSTÉM POČÍTAČE – GRAFICKÁ KARTA A MONITOR	191
12.1	STANDARDY GRAFICKÝCH ADAPTÉRŮ.....	191
12.2	GRAFICKÝ AKCELERÁTOR.....	192
12.3	PAMĚŤ GRAFICKÉ KARTY	193
12.4	CHARAKTERISTIKA CRT MONITORU.....	195
12.5	PRINCIP ZOBRAZENÍ NA CRT MONITORU	195
12.6	TYPY OBRAZOVEK A PARAMETRY MONITORŮ	196
12.7	CHARAKTERISTIKA LCD MONITORU.....	197
12.8	PRINCIP ZOBRAZENÍ NA LCD MONITORU	198
13.	OPTICKÁ MÉDIA, MECHANIKY CD A DVD	202
13.1	ČTENÍ A ZÁZNAM DAT Z MAGNETO-OPTICKÉHO DISKU	202
13.2	DISK CD.....	203
13.3	STRUČNÁ HISTORIE	204
13.4	VÝROBA CD DISKU	205
13.5	CDR A CDR-W DISKY	205
13.6	DVD DISK.....	207
13.7	PRINCIP ČINNOSTI CD MECHANIKY	209
13.8	BLU-RAY - BLÍZKÁ BUDOUCNOST?	210
14.	ZVUK, AUDIO HARDWARE.....	213
14.1	ZVUK A JEHO PODSTATA.....	213
14.2	KOMPRES ZVUKU	214
14.3	ZVUKOVÁ KARTA.....	215
14.4	HODNOCENÍ ZVUKOVÝCH ADAPTÉRŮ	216
	KLÍČ K ŘEŠENÍ A REJSTŘÍK.....	239

POKYNY KE STUDIU

Počítačové systémy

Pro předmět Počítačové systémy 1. semestru oboru Automatické řízení a inženýrská informatika jste obdrželi studijní balík obsahující

- integrované skriptum pro distanční studium obsahující i pokyny ke studiu,
- CD-ROM s doplňkovými animacemi vybraných částí kapitol,
- harmonogram průběhu semestru a rozvrh prezenční části,
- rozdělení studentů do skupin k jednotlivým tutorům a kontakty na tutorý,
- kontakt na studijní oddělení.

Prerekvizity

Pro studium tohoto předmětu se nepředpokládá absolvování předchozích předmětů avšak znalosti z oblasti základu algoritmizace a výpočetní techniky jsou výhodou.

Cílem předmětu

je seznámení se základními pojmy z oblasti hardwarových komponent v oblasti teoretické a se základy jazyka C a práce v prostředí Microsoft Visual Studio v oblasti praktické. Předmět je prerekvizitou předmětu Operační systémy a programování. Po prostudování modulu by měl student být schopen ovládat základy práce v prostředí Visual Studio C++, znát základní syntaxi programovacího jazyka C, umět vytvářet různé typy programů a úloh a především v oblasti hardwaru výborně porozumět komponentám osobního počítače.

Pro koho je předmět určen

Modul je zařazen do magisterského studia oboru Automatické řízení a inženýrská informatika studijního programu

Skriptum se dělí na části, kapitoly, které odpovídají logickému dělení studované látky, ale nejsou stejně obsáhlé. Předpokládaná doba ke studiu kapitoly se může výrazně lišit, proto jsou velké kapitoly děleny dále na číslované podkapitoly a těm odpovídá níže popsaná struktura. Předmět se skládá, jak už bylo uvedeno výše, z oblasti teoretické a praktické. V prezenční formě studia probíhá teoretická část formou přednášek na téma základů jazyka C a hardwarových komponent osobního počítače a praktická část probíhá na cvičeních formou řešení úloh v jazyce C.

U distanční formy studia probíhá výuka obdobně, avšak v tomto výukovém modulu jsou cvičení z jazyka C přidružena ke kapitolám 2, 3, 4 a 9. Součástí klasifikace předmětu je samostatný test z jazyka C formou řešení nějakého úkolu, odevzdání programu v jazyce C dle určitého zadání, prezentace na vybrané téma v oblasti hardwaru a také závěrečná zkouška z oblasti hardwarových komponent.

Uvedené úkoly mají celkový bodový zisk 100 b za předmět rozdělen dle následujícího bodování:

Test ze základu jazyka C formou řešení zadané úlohy na PC	20 bodů
Odevzdání programu na zadané téma	20 bodů
Vystoupení a prezentace na téma v oblasti hardwaru	10 bodů
Závěrečná zkouška z oblasti hardwarových komponent a základů jazyka C	50 bodů

Při studiu každé kapitoly doporučujeme následující postup:**Čas ke studiu:** x hodin

Na úvod kapitoly je uveden **čas** potřebný k prostudování látky. Čas je orientační a může vám sloužit jako hrubé vodítko pro rozvržení studia celého předmětu či kapitoly. Někomu se čas může zdát příliš dlouhý, někomu naopak. Jsou studenti, kteří se s touto problematikou ještě nikdy nesetkali a naopak takoví, kteří již v tomto oboru mají bohaté zkušenosti.

**Cíl:** Po prostudování tohoto odstavce budete umět

- popsat ...
- definovat ...
- vyřešit ...

Ihned potom jsou uvedeny cíle, kterých máte dosáhnout po prostudování této kapitoly – konkrétní dovednosti, znalosti.

**Výklad**

Následuje vlastní výklad studované látky, zavedení nových pojmů, jejich vysvětlení, vše doprovázeno obrázky, tabulkami, řešenými příklady, odkazy na animace.

**Řešený příklad**

Zadání a řešení praktického příkladu jako součást výukového textu.

**Korespondenční úkol**

Zadání domácí úlohy nebo souvisejícího úkolu v rámci výkladu.

**Pojmy k zapamatování**

Touto ikonkou je upozorněno na pojem, který bychom si měli pamatovat. Neznamená to ale, že ostatní pojmy výkladu si pomatovat nemusíme.

**Další zdroje**

Seznam další literatury, www odkazů apod. pro zájemce o dobrovolné rozšíření znalostí popisované problematiky. Budou uvedeny na konci každé kapitoly.



CD-ROM

Informace o doplňujících animacích, které si může student vyvolat z CD-ROMu připojeného k tomuto materiálu.



Shrnutí kapitoly

Na závěr kapitoly jsou zopakovány hlavní pojmy, které si v ní máte osvojit. Pokud některému z nich ještě nerozumíte, vraťte se k nim ještě jednou.



Kontrolní otázka

Na konci každé kapitoly bude zveřejněn seznam otázek, které se mohou v plném nebo modifikovaném znění objevit u zkoušky z předmětu Počítačové systémy. Tyto otázky jsou také podkladem ke státnicovým otázkám magisterského oboru Automatické řízení a inženýrská informatika, konkrétně státnicového předmětu Informační technologie.



Úkol k řešení

Na konci kapitol, které kromě teoretických otázek umožňují i praktické procvičení probraného učiva, najdete úkol k řešení. Rovněž s tímto úkolem se můžete v obdobném znění setkat u zkoušky či zápočtového testu.



Klíč k řešení

Odpovědi kontrolních otázek výše jsou uvedeny v závěru učebnice v Klíči k řešení. Používejte je až po vlastním vyřešení úloh, jen tak si samokontrolou ověříte, že jste obsah kapitoly skutečně úplně zvládli.

1. HARDWAROVÉ KOMPONENTY PC



Čas ke studiu: 2 hodiny



Cíl Po prostudování tohoto odstavce budete umět

- definovat základní terminologii v oblasti hardwaru
- popsat podrobněji a technicky správně komponenty PC
- popsat základní princip činnosti počítače dle základní architektury
- definovat formáty základních desek
- popsat vývoj slotů a socketů základních desek a procesorů



Výklad

V úvodní kapitole uvedeme výčet základních pojmů, které je třeba znát a se kterými se neustále setkáváme při práci s hardwarovými komponenty. Takovým pojmem je například *sběrnice*, protože se s ní setkáme téměř u všech součástí počítače. První kapitola tedy uvádí terminologii, se kterou budeme pracovat v oblasti komponent počítačů, jsou to sice základní pojmy, jejich znalost je však nezbytná pro další studium. V následujících kapitolách se seznámíme s principem činnosti počítače dle Von Neumannovy a Harvardské architektury, probereme typy počítačových skříní neboli *case* či *šasi* počítače a poté již nakoukneme dovnitř. V této úvodní kapitole se budeme věnovat základním deskám počítače, dalším neméně důležitým komponentám jsou věnovány samostatné kapitoly, tak vzhůru do práce – respektive do studia!

1.1 Základní terminologie

Ačkoliv některé termíny mezi sebou souvisí a jsou příbuzné jiným, abecední pořadí se stále jeví jako nepřehlednější.

Adaptér pevného disku: (UATA/IDE, EIDE, SCSI). Pevné disky používají svá vlastní rozhraní, podrobněji bude probráno v kapitole 11.

BIOS: (*Basic Input Output System*). Programové vybavení uložené v paměti ROM (EPROM, EEPROM, Flash) zajišťující nejzákladnější funkce (např. zavedení OS).

Cache: Jedná se o vyrovnávací paměti, jež mají podstatný vliv na rychlost a výkon celého systému. V tomto skriptu budeme klást důraz na procesorovou cache L1 a L2.

CISC: Architektura procesoru s komplexní instrukční sadou (*Complex Instruction Set Computer*). Procesor disponuje s instrukčním souborem s takovými instrukcemi, které pod jedním operačním kódem vykonají složité operace, které vyžadují více strojových cyklů.

CPU: Hlavní procesorová jednotka (*Central processor Unit*) je „motorem“ počítače, tedy výkonnou jednotkou, která čte z paměti instrukce a zpracovává data podle programu. Procesoru jsou věnované kapitoly 6 a 7 tohoto učebního textu.

Data: Uložení informací do takového formátu, aby byly zpracovatelné počítačem.

DMA: Přímý přístup do paměti (*Direct memory access*). Je to schopnost sběrnice posílat data ze zařízení (pevný disk, mechanika CD, atd.) pomocí řadiče přímo do paměti bez účasti aritmeticko-logické jednotky (procesoru). Architekturám sběrnic bude věnována kapitola 8.

Firmware: Je programové vybavení zařízení vázané na konkrétní typ hardwaru. Je většinou uložen do paměti typů ROM výrobcem, avšak často do přepisovatelných pamětí (EPROM), aby jej bylo možné aktualizovat.

Grafický akcelerátor: Grafické karty jsou stále výkonnější a v současné době obsahují svůj vlastní procesor označený jako GPU (*Graphics Processing Unit*). Proto tyto výkonné videoadaptéry označujeme pojmem akcelerátor.

Hardware: Z anglického významu „nářadí“, počítačový hardware označuje veškeré fyzicky existující technické vybavení počítače.

IEEE1394: Standard FireWire pro propojování zařízení s velkým objemem dat – typicky videokamery. Vznikl v roce 1995 a jeho výhodou oproti dostupné USB 1.0 (12 Mb/s) byla vysoká přenosová rychlost 400 Mb/s. Dnes je USB 2.0 rychlejší (480 Mb/s) avšak standard FireWire se bude vyvíjet a očekává se až rychlost až 1,6 Gb/s.

Informace: Je výsledek zpracování nebo organizace dat. Jednotkou informace je bit. Jeden bit je přitom vyčíslen jako pravděpodobnost jevu, že nastane nějaká skutečnost. Jeden bit představuje jev s pravděpodobností 50 %.

IRQ (*Interrupt Request*): Označuje signál, kterým požádá nějaké zařízení procesor o přerušení právě probíhajícího procesu za účelem obsluhy důležitější činnosti.

Paměť: Zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat). Existuje celá řada typů pamětí a tomuto tématu je věnována samostatná kapitola 10.

Předpona	Značka	Zápis	Mocnina	Převod
kilo	k	1kB	$2^{10}B$	1024B
mega	M	1MB	$2^{20}B$	1048576 B
giga	G	1GB	$2^{30}B$	1073741824 B
tera	T	1TB	$2^{40}B$	1099511627776 B

Tab. 1.1 Značení velikosti paměti

Porty: Sériové a paralelní rozhraní slouží k připojování periférií. Podrobněji v kapitole 9. Této kapitole bude také věnován projekt v jazyce C.

Software: Je programové vybavení počítače. Dělí se na systémový a aplikační.

Registr: Velmi rychlé paměťové místo malé kapacity (jednotky Bytů) umístěné většinou uvnitř procesoru počítače.

RISC: Architektura procesoru s redukovanou instrukční sadou. Procesor disponuje instrukční sadou složenou pouze s omezeného počtu jednoduchých instrukcí, které svou sekvencí nahrazují složité instrukce. Tyto jednoduché instrukce pak lze vykonat za jeden instrukční cyklus.

Řadič: Zařízení převádějící příkazy v symbolické formě (instrukce) na posloupnost signálů ovládajících připojené zařízení. Jedná se tedy o zařízení, které řídí činnost jiného zařízení.

Sběrnice: Soustava vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače. Pomocí těchto vodičů mezi sebou jednotlivé části počítače komunikují a přenášejí data.

USB rozhraní: (*Universal serial Bus*) Sériová sběrnice má již dvě vývojové verze, **USB 1.1** (max. přenosová rychlost 12 Mbit/s) a **USB 2.0** (480 Mbit/s, pokud je zařízení high-speed). USB 2.0 je zpětně kompatibilní s USB 1.1.

1.2 Princip činnosti počítače

Von Neumannovo schéma bylo navrženo roku 1945 americkým matematikem Johnem Von Neumannem jako první model samočinného počítače. Tento model s jistými výjimkami zůstal zachován dodnes. Princip činnosti počítače podle Von Neumannova schématu je následující: Do operační paměti se pomocí vstupního zařízení přes aritmeticko-logickou jednotku umístí program, který bude provádět výpočty. Následně se obdobným způsobem do operační paměti umístí data, se kterými bude program pracovat. Proběhne program, který provádí operace s daty, jednotlivé kroky programu provádí aritmeticko-logická jednotka. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Jednotlivé dílčí výsledky výpočtu jsou ukládány do operační paměti. Po skončení výpočtu jsou výsledky poslány přes aritmeticko-logickou jednotku na výstupní zařízení.

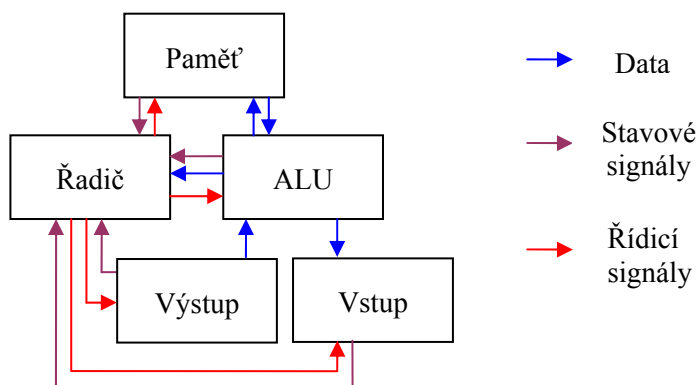


Pojem k zapamatování: Harvardská a Von Neumannova koncepce

Harvardská architektura má na rozdíl od Von Neumannovy architektury oddělený paměťový prostor pro data a pro program. U Von Neumannovy koncepce je paměť pro data a pro program společná. Tímto způsobem se můžeme na instrukce dívat jako na data a během programu je můžeme měnit a ovlivňovat chování programu.

Von Neumannův model počítače je určen pro klasické počítače, tak jak je známe. Naproti tomu s Harvardskou koncepcí pracují např. jednočipové mikropočítače a mikrokontroléry, které jsou určeny již ke konkrétním speciálním úlohám v oblasti techniky a průmyslu.

Dnešní počítače stále pracují podle tohoto schématu, samozřejmě vývoj šel kupředu a tak dnes máme určité odlišnosti od původní činnosti počítače podle Von Neumannova schématu.



Obr. 1.1 Von Neumannovo schéma činnosti počítače

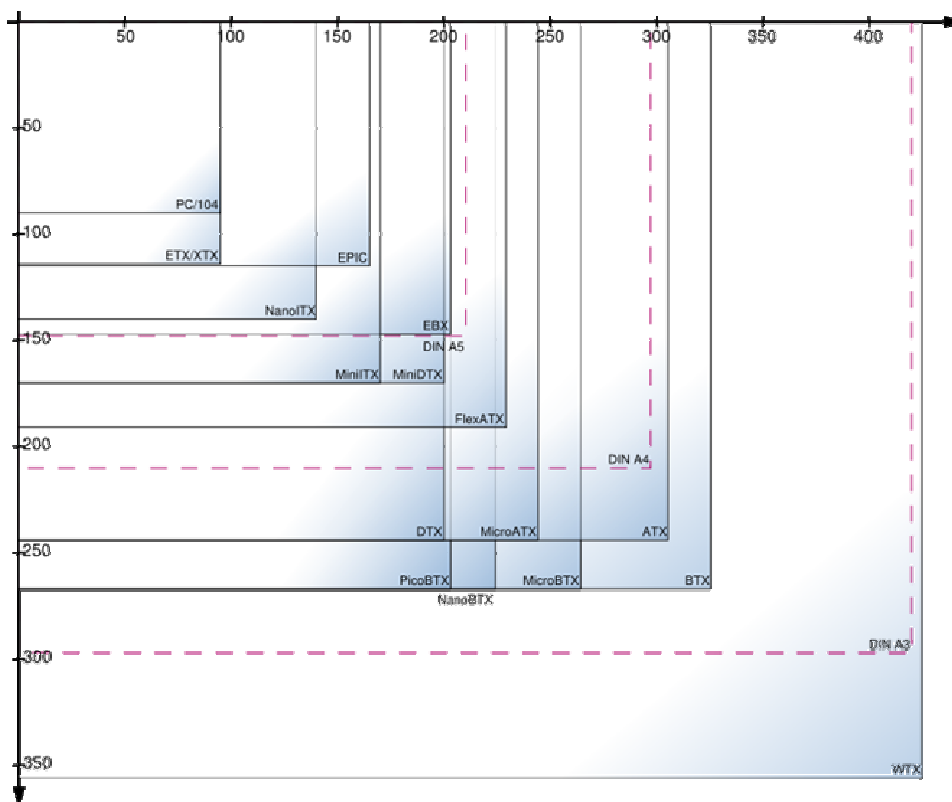
Například dnes již počítač nepracuje pouze nad jedním programem. Je obvyklé, že počítač zpracovává paralelně více programů zároveň – tzv. multitasking. V dnešní době také počítač může disponovat více procesory, popřípadě procesor může být vícejádrový. Dnešní periferní vstupně/výstupní zařízení (*I/O devices*) umožňují jak vstup, tak výstup dat (programu) a v neposlední řadě se do paměti nemusí program zavést celý, ale je možné zavést pouze jeho část a ostatní části zavádět až později.

1.3 Základní desky

Základní deska je nejdůležitější součástí počítače, nazývá se také *motherboard* nebo *mainboard*. Na základní desce jsou umístěné sběrnice, které slouží ke komunikaci a napájení pro ostatní komponenty. V základní desce je umístěn v příslušné patici procesor, systémová paměť, energeticky nezávislá

paměť ROM, ve které je uložen systém BIOS, který slouží k nastavení počítače ihned po jeho spuštění. Dalšími zásuvnými komponentami mohou být grafická či zvuková karta, pokud se nespokojíme s integrovanými součástmi čipové sady. Čipová sada je srdcem celé základní desky a bude jí věnována samostatná kapitola, stejně jako ostatním vyjmenovaným komponentám.

Důležitým parametrem základní desky je její formát. Výrobci se musí dohodnout na variantách velikosti základních desek tak, aby bylo možné tyto desky montovat do počítačových skříní různých výrobců. Desky stejného formátu tedy musí odpovídat předepsaným rozměrům, předepsaným způsobem upevnění a montování, musí mít stejné požadavky na napájení a podobně. Formátům základních desek se říká form faktory. Desky prošly svým vývojem jako každá komponenta PC. Pro první počítače IBM byl znám jediný formát PC/XT. Původní desky pro současná PC byly označovány zkratkou AT (*Advanced technology*). Konektory těchto desek bylo nutno vyvést kabelem na port a připevnit na zadní stranu skříně. Dalším z mnoha rozdílů daným vývojem byl například konektor klávesnice DIN, jež byl nahrazen u novějších desek konektorem PS/2. Mezi zastaralé formáty patří i formát Baby –AT odvozené z formátu AT. Tyto desky se udržely na trhu přibližně do roku 1996 a mezi uživateli pochopitelně déle. Na následujícím obrázku vidíme rozměry form faktorů základních desek.



Obr. 1.2 Form faktory základních desek počítače

V současnosti je základním standardem faktor ATX o základním rozměru (305 x 244 mm), což vidíme i na obrázku. Z tohoto form faktoru jsou odvozeny další varianty mini ATX (284 x 208 mm) a micro ATX (244 x 244 mm). Tyto desky jsou určeny pro menší varianty skříní, jejich počet rozšiřujících karet je omezený, avšak musela být zachována kompatibilita konektorů pro napájení a pro standardní panel vstupně/výstupních konektorů skříně. Dalšími form faktory, které můžeme vidět na obrázku, jsou Flex-ATX (*flexible ATX*) pro specifikaci levnějších menších systémů, WTX (*Workstation Technology Extended*) pro výkonné grafické stanice a servery a další. Za zmínku nepochybně stojí i formát BTX navržený firmou Intel. Tento formát není kompatibilní s deskou ATX, přesto ho montovali a ještě stále montují přední výrobci jako Dell či HP. Důvodem vzniku a použití tohoto formátu je ale jiný a to především možnost zkonstruovat počítačové skříně pro použití tepelně náročných procesorů Intel. Ve světě skládaných počítačů se však tento formát neuchytil a Intel vývoj a varianty těchto desek (microBTX, nanoBTX, picoBTX) ukončil a brzy se také ukončí i jejich výroba.

Nyní v roce 2007 přichází s novým formátem DTX pro změnu společnost AMD. Tento formát však je již kompatibilní s ATX, protože se firma AMD poučila z nezdaru Intelu s BTX, avšak na standardizaci specifikace si musíme počkat po uzavěře psaní tohoto učebního textu.

Při výběru základní desky máme v zásadě tři rozhodovací argumenty. Nejdůležitějším prvkem je volba čipové sady, nejdůležitější součástí desky, určující výkon. Čipová sada je natolik důležitým sloučením mnoha obvodů, že jí bude věnována samostatná kapitola. Dalším argumentem pro výběr desky je bezesporu počet a typ rozšiřujících slotů a také další možnosti desky jako sledování teploty, otáček větráku apod.

1.4 Sloty a sockety

Socket je místo na základní desce, do kterého se zasouvá procesor. Piny procesoru zapadnou do děr socketu. Oproti tomu slot obsahuje drážky do kterých se CPU zasouvá podobně jako rozšiřující karty. Socketové CPU se daly vyrábět levněji, avšak provedení slotové byly pokusem o dosažení vyššího výkonu díky faktu, že čím blíže je cache k procesoru, tím větší je výkon. Máme na mysli nedávnou historii procesorů, protože komfort mít cache přímo jako svou součást si nemohl vývojově dovolit kdejaký procesor.

Vývojově se tedy podívejme do roku 1997, kdy Intel pro své Pentium použil patici Socket 7. Řady patic procesorů samozřejmě začínají od čísla 1, avšak na starších Socketech 1-6 běžely procesory řady 486 a původní Pentia pod hranici 75 MHz.

Socket 7 má 321 vývodů uspořádaných do mřížky 21x21 SPGA (pouzdro *Staggered Pin Grid Array*). Procesory Intel Pentium taktované v rozmezí 75 -233 MHz tedy měli 321 pinů. V této době se ještě procesory různých výrobců daly zasunout do stejné desky a nebyly odděleny platformy Intel a AMD. Na trhu působila tehdy ještě i firma Cyrix a tak patici Socket 7 používalo hned několik výrobců. Firmu AMD reprezentovaly procesory AMD K5, K6, K6-II, K6-III, firmu Cyrix procesory MI a MII.

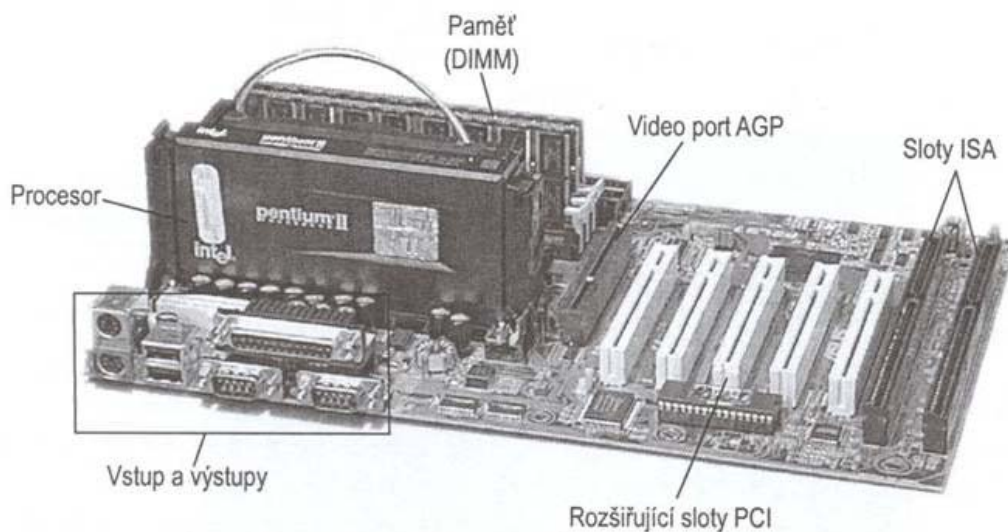


Obr. 1.3 Procesor Pentium pro patici Socket 7 má 321 vývodů

Následujícím krokem firmy Intel byl vývoj patice Socket 8 pro jejich nový procesor šesté generace Pentium Pro. Tento procesor měl 387 pinů a potřeboval novou patici. Firma AMD v této době za Intellem pokulhávala, uvedení prvního procesoru šesté generace K6 a jejich následníky zanechala na předchozí patici Socket 7.

Pro další procesory šesté generace se vyměnila technologie socketu za slot již z výše uvedeného důvodu zvýšení výkonu. Patici nahradil Slot 1 (242 pinů) a procesor je společně s pamětí cache umístěn na desce a zasunut podobně jako paměťový modul v pouzdře typu SECC (*Single Edge Contact Cartridge*) do slotu základní desky. Podobný slot označený Slot 2 (330 pinů) používal serverový procesor XEON. Modernizovalo se také pouzdro na SECC 2, které mělo kryt pouze z jedné strany a umožnilo tak chladiči dotýkat se přímo procesoru. Tohoto faktu bylo využito u výkonnějších

Pentii III, naopak z důvodu ušetřit náklady byla vyvinuta varianta SEP (*Single Edge Processor*) bez pouzdra pro levnější varianty Celeron.



Obr. 1.4 Procesor Pentium II ve slotovém provedení

Firma AMD pokračovala na starších patcích a později vyvinula variantu Slot A pro své první procesory Athlon. Slot A je mechanicky kompatibilní, ale elektricky nekompatibilní s Intelovým Slotem 1. Pro následující verze Athlonů již firma AMD nahrazuje Slot A Socketem A, z důvodu integrované cache procesoru. Píše se rok 2000 a definitivně se na trhu oddělují platformy Intel a AMD s kompatibilitou patic. Od této doby již při výběru základních desek zohledňujeme výběr procesoru.

Z důvodu další nekompatibility a neexistence konkurence pro dva výrobce procesorů si představíme další sockety zvlášť. Nejprve sockety pro procesory Intel.

V roce 1999 představil Intel patici Socket 370 pro procesory Pentium III a Celeron. Integraci paměti cache do procesoru a návrat k socketovému provedení tedy zvládl o rok dříve než AMD, což mu stále dávalo výsadní postavení na trhu. Od této chvíle číslovka v názvu socketu představuje počet pinů procesoru. V roce 2000 bylo uvedeno Pentium 4 a s ním nová patice Socket 423. Tato patice podporovala procesorovou sběrnici o rychlosti 400 MHz propojující procesor s MCH čipem čipové sady (viz. Kapitola 5). O rok později byla uvedena na trh patice Socket 478 podporující FSB až 533 MHz a tudíž vydržela poměrně dlouhou dobu a ještě v roce 2006 jimi byly osazovány desky pro procesory Celeron D310-D335 o taktu 2,2 – 2,8 GHz a Pentium 4 s jádrem Prescott na frekvencích až do 3,0 GHz. Trhu s mobilními procesory v této době vládl Socket 479 pro procesory Pentium M (1,5 - 2,2 GHz) s jádrem Dothan.

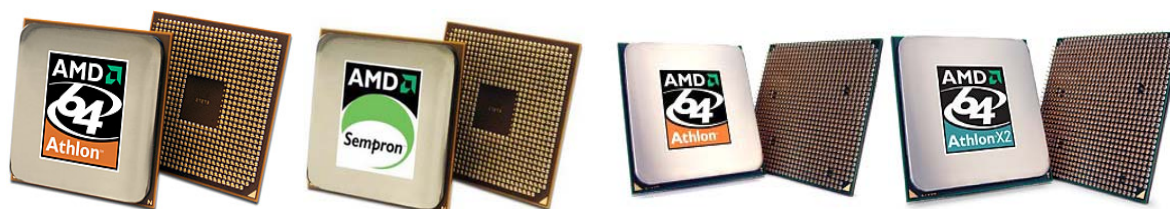
Od roku 2004 ovšem začal Intel uvádět nově vyvíjenou patici **Socket LG775** pro procesory Celeron D a Pentium IV (jádro Prescott) s taktům od 2,5 až po 3,8 GHz. Tato patice jednoznačně vytlačila předchozí typ a nyní v podstatě nemáme jinou možnost, pokud chceme využívat procesory firmy Intel, přestože v současné době nastala změna technologie a honba za vysokými frekvencemi a samotná technologie Net Burst pro Pentium 4 se stala historií.

Vývojový výčet patic procesorů Intel máme za sebou, pojďme se podívat na platformu AMD. Oddělení kompatibility nastartovala firma AMD v roce 2000 paticí Socket A, nástupcem patice Slot1. Příslušné procesory socketu A jsou vybaveny 462 piny. Je určen pro procesory Athlon XP (jádro Burton) 2600+, 2800+, AMD Sempron 2200+, 3000+ (jádra Thoroughbred, Burton).



Obr. 1.5 Socket LGA775 současných (rok 2007) procesorů Intel

Ve své podstatě tento socket vydržel mnoha uživatelům až do současnosti. Přesto dnes je již výroba ukončena a patiči vystřídal další, samozřejmě dle vývoje procesorů. Jak bylo uvedeno výše, u procesoru Intel dnes máme v podstatě jen jednu aktuální patiči socket 775, u konkurenčního AMD máme na výběr stále tři možnosti. První možností je Socket 754 pro procesory Athlon 64 s označením 2800+ až 3400+ (jádro NewCastle a Clawhammer), Sempron 64 od 2500+ po 3400+ (s jádrem Palermo). Druhým současným (rok 2007) socketem na trhu je Socket 939 pro procesory Athlon 64 s označením 3000+ až 4000+ (jádro Venice a SanDiego) a Athlon 64 X2 s označením 4000+ až 4400+ (jádro Manchester a Toledo).



Obr. 1.6 Procesory Athlon 64 a Sempron na socketu 754 a Athlon 64 a X2 na socketu 939

Třetím socketem je 940ti pinový socket AM2 (pozor: neplést se Socketem 940, který využívají serverové procesory Opteron), socket AM2 byl uveden v druhé polovině roku 2006, převážně z důvodu podpory DDR2 pamětí, které předchozí sockety neumožňovaly. Nový socket AM2 je určen pro procesory Sempron s jádrem Manila s označením 2800+ až 3200+, procesory Athlon 64 s jádrem Orleans s označením 3000+ až 3800+ a dvoujádrové procesory X2 s jádrem Windsdor s označením 3600+ až 5200+.



Korespondenční úkol

Zjistěte, co znamená znaménko plus u číselného označení procesorů AMD.



Další zdroje

1. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
2. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
3. Průvodce základními deskami [online] dostupný z <http://www.motherboards.org>



Shrnutí kapitoly

Základním pojmem v oblasti počítačového hardwaru je mimo jiné: CPU, což je hlavní procesorová jednotka (*Central procesor Unit*). Procesor je „motorem“ počítače, tedy výkonnou jednotkou, která čte z paměti instrukce a zpracovává data podle programu. Procesory dělíme dle mnoha kritérií, jedním z nich je architektura CISC a RISC. CISCová architektura procesoru obsahuje komplexní instrukční sadu (*Complex Instruction Set Computer*). Procesor tedy disponuje instrukčním souborem s takovými instrukcemi, které pod jedním operačním kódem vykonají složité operace, které vyžadují více strojových cyklů. Naproti tomu RISCová architektura procesoru má instrukční sadu redukovanou. Procesor tak disponuje instrukční sadou složenou pouze s omezeného počtu jednoduchých instrukcí, které svou sekvencí nahrazují složité instrukce. Tyto jednoduché instrukce pak lze vykonat za jeden instrukční cyklus.

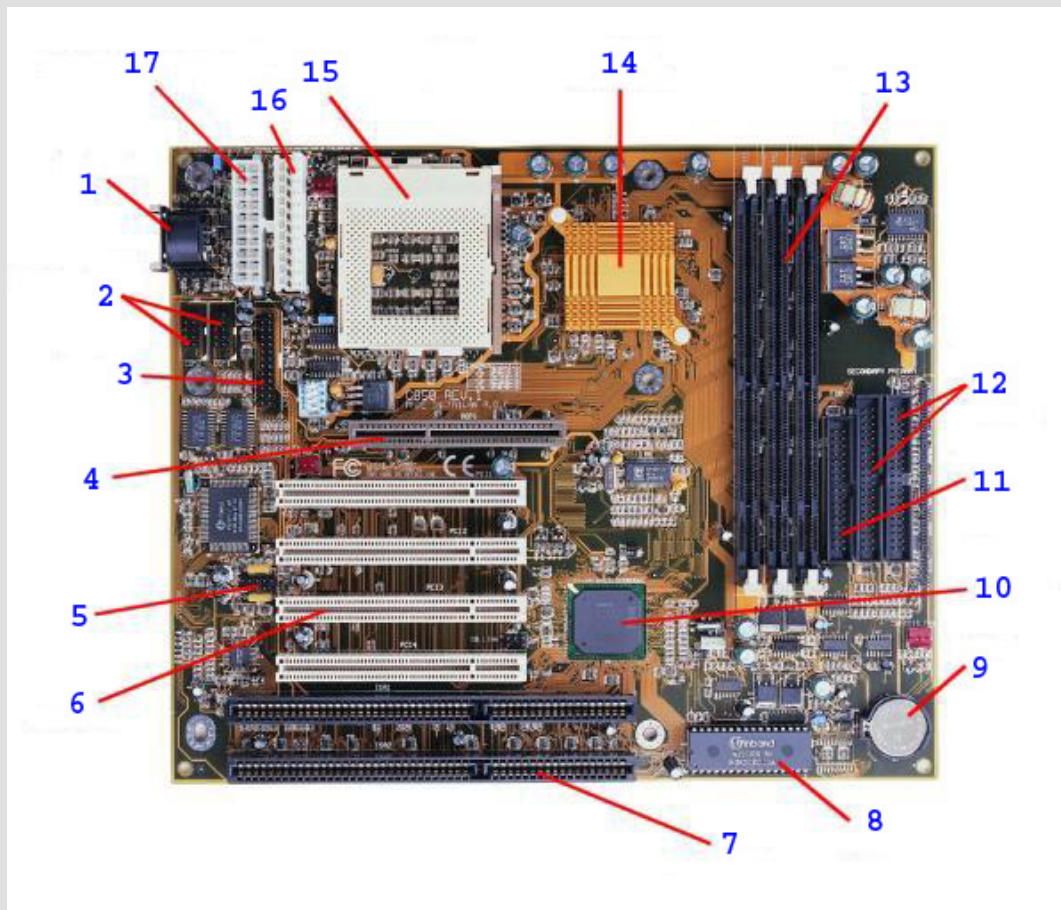
Základní deska je nejdůležitější součást počítače, nazývá se také motherboard nebo mainboard. Na základní desce jsou umístěné sběrnice, které slouží ke komunikaci a napájení pro ostatní komponenty. V základní desce je umístěn v příslušné patici procesor, systémová paměť a také energeticky nezávislá paměť ROM, ve které je uložen systém BIOS, který slouží k nastavení počítače ihned po jeho spuštění. Dalšími zásuvnými komponenty mohou být grafická či zvuková karta, pokud se nespokojíme s integrovanými součástmi čipové sady. Sběrnice je soustava vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače. Pomocí těchto vodičů mezi sebou jednotlivé části počítače komunikují a přenášejí data. Grafické karty jsou stále výkonnější a v současné době obsahují svůj vlastní procesor označený jako GPU (*Graphics Processing Unit*). Proto tyto výkonné videoadaptéry označujeme pojmem akcelerátor. Paměť je zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat).

Von Neumannovo schéma bylo navrženo roku 1945 americkým matematikem Johnem Von Neumannem jako první model samočinného počítače. Tento model s jistými výjimkami zůstal zachován dodnes. Princip činnosti počítače podle Von Neumannova schématu je následující. Do operační paměti se pomocí vstupního zařízení přes aritmeticko-logickou jednotku umístí program, který bude provádět výpočty. Následně se obdobným způsobem do operační paměti umístí data, se kterými bude program pracovat. Proběhne program, který provádí operace s daty, jednotlivé kroky programu provádí aritmeticko-logická jednotka. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Jednotlivé dílčí výsledky výpočtu jsou ukládány do operační paměti. Po skončení výpočtu jsou výsledky poslány přes aritmeticko-logickou jednotku na výstupní zařízení.



Úkol k řešení 1.1

Přiřaďte komponenty základní desky ze seznamu A - P k jednotlivým číslům 1 - 17.



- A. AGP slot
- B. PCI slots
- C. BIOS
- D. ISA Sloty
- E. LPT port
- F. Konektor klávesnice
- G. Jižní most
- H. Baterie pro CMOS

- I. Paměťové sloty DIMM
- J. Severní most
- K. USB
- L. Konektory IDE
- M. Napájecí konektor ATX
- N. Napájecí konektor AT
- O. Patice pro CPU
- P. Sériové porty COM1 a COM2



Úkol k řešení 1.2

Zkuste cvičně navrhnout tři konfigurace PC dle dostupných on-line ceníků a specifikací, např. Alfa Computer, AutoCont, Brain Computer atd. Vytvořte tři konfigurace rozdělené do kategorií: kancelářská, domácí, profesionální.



Kontrolní otázka 1.1

Popište definici základních pojmů: CPU, paměť, grafický akcelerátor, sběrnice, porty, řadič, rozhraní.



Kontrolní otázka 1.2

Definujte princip činnosti počítače podle Von Neumannova schématu. V čem je odlišný od Harvardského?



Kontrolní otázka 1.3

Co je to form faktor základní desky?



Kontrolní otázka 1.4

Vyjmenujte vývojovou posloupnost slotů a socketů procesorů. Pro které procesory byly určeny?



Kontrolní otázka 1.5

Vysvětlete zkratky: SECC, SEP, SPGA, DMA, IRQ, RISC, CISC, BIOS, GPU, USB.

2. Úvodní seznámení s prostředím MS Visual C++ 6.0



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- založit projekt v prostředí Visual Studio 6.0
- orientovat se v programovém prostředí
- kompilovat a spouštět programy
- princip vyhledávání a opravy základních chyb
- používat MSDN nápovědu k jazyku C



Výklad

Kapitoly 2-3 jsou věnovány základnímu kurzu seznámení s Visual Studií 6.0, práci v tomto prostředí včetně pojmů, se kterými se musíme seznámit, abychom mohli začít psát jednoduché programy v jazyce C. Prostředí Visual Studio je mocným programátorským a vývojářským nástrojem a jeho popis by zabral stovky stran. Rovněž čtvrtá kapitola věnovaná jazyku C z vás neudělá programátora, ale vysvětlí vám syntaxi jazyka a umožní programovat v jazyce C různé úlohy. Kapitoly budou doprovázeny příklady ke konkrétnímu výkladu a také animacemi, zobrazující postup práce v prostředí MS Visual Studio. V denním studiu je jazyku C věnována podstatná část cvičení a proto přiložené animace vhodně ilustrují cvičební náplň pro studenty kombinované formy studia. Protože náplní předmětu jsou hardwarové komponenty PC, vyústí naše znalosti jazyka C v projekt komunikace dvou PC prostřednictvím sériového a paralelního portu.

2.1 Založení projektu ve Visual C++ 6.0

Visual C++ 6.0 je součástí balíku Visual Studio 6.0 od společnosti Microsoft. Je to silný nástroj pro vývoj profesionálních aplikací v jazycích C a C++. Prostředí si spustíte přes nabídku Start > Programy > Microsoft Visual Studio 6.0 > Microsoft Visual C++ 6.0. Objeví se samotné vývojové prostředí, které nic neobsahuje. Proto si musíme založit projekt, se kterým budeme dále pracovat.

Co je to projekt?

Projekt je adresář, do kterého si budeme ukládat soubory, se kterými budeme pracovat (např. zdrojové soubory jazyka C nebo hlavičkové soubory). I pro jeden jediný soubor musíme mít vytvořený projekt. Bez otevřeného aktivního projektu není možno kompilovat a spouštět programy.

Založení a volba typu projektu

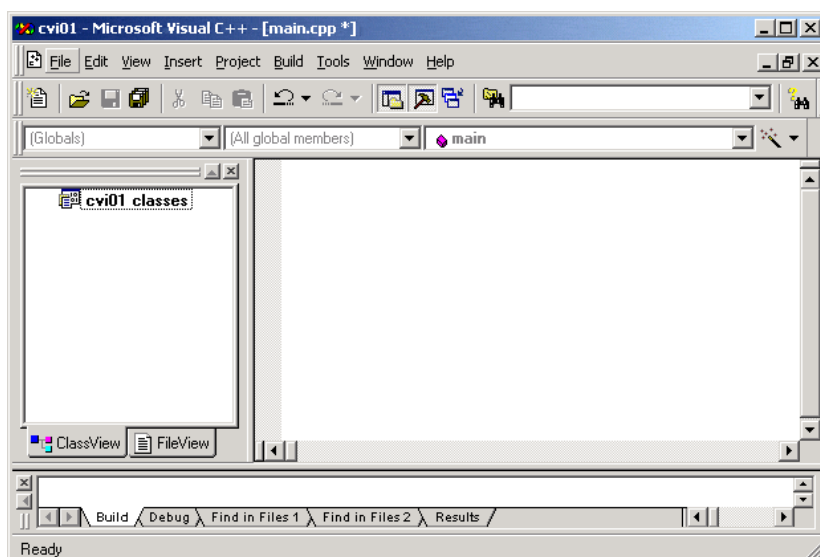
Nový projekt si založte z menu File > New (u založení nového projektu a zdrojového souboru projektu nepoužívejte ikonu , která slouží pouze pro založení nového textového souboru, nikoliv projektu ani souboru projektu!). V dialogovém okně si zvolte záložku "Projects" (pokud nemáte ještě žádný projekt otevřen, zobrazí se záložka "Projects" automaticky). Jako typ projektu zvolte *Win32 Console Application*. Visual C++ obsahuje hodně typů projektů pro celou řadu programů, ale my se budeme zabývat pouze typem *Win32 Console Application*, který je určen pro práci s konzolovými aplikacemi jazyka C. Do kolonky "Project name" napište jméno projektu např. cvic01 (nedoporučuje se používat

čeština a jméno projektu nesmí obsahovat mezery). Dále zvolte umístění projektu "Location" a zvolte cestu umístění vašeho projektu. V prostředí fakultních učeben ukládejte projekty na disk D. Díky bezpečnostní politice a provozním řádu učeben nemáte oprávnění zápisu na disk C. Projekty také nevytvářejte na síťovém disku např: F:\, protože při kompilaci projektu dochází u síťových disků k chybám. Po skončení práce si můžete svůj projekt zkopírovat na svoje konto na tomto disku (F:\). Je-li jméno projektu cvi01 a zvolen disk D:\, tak si program automaticky vytvoří adresář D:\cvi01. Ostatní volby nechejte jak jsou a stiskněte tlačítko OK.

Dále se objeví dialogové okno Wizard (průvodce), které se dotazuje jaký druh konzolové aplikace chcete vytvořit. Jsou zde čtyři možnosti: *An empty project*; *A simple application*; *A "Hello!, World" application* a také *An application that supports MFC*.

Ale nás bude pouze zajímat průvodce *An empty project*. To znamená, že náš projekt neobsahuje žádný zdrojový soubor nebo jiný soubor. Ostatní průvodci jsou pro naše programy nevhodné a nebudeme je vůbec používat.

Stiskněte tlačítko Finish a následným potvrzením informačního okna tlačítkem OK si vytvoříte svůj projekt.



Obr. 2.1 Prostředí MS Visual Studio C++ 6.0

2.2 Orientace ve vývojovém prostředí

Projekt máme založený, takže obrazovka by měla vypadat podle obr.2.1. Nyní si ukážeme vkládání souborů do projektu, vytváření nových souborů a přepínání mezi těmito soubory.

Vložení souboru do projektu

Všimněte si dvou záložek *ClassView* a *FileView*. *ClassView* zobrazuje jednotlivé třídy projektu, včetně jejich funkcí a proměnných. *FileView* zobrazuje všechny soubory, které projekt obsahuje a jsou rozříděny do jednotlivých skupin *Source Files* (zdrojové soubory), *Header Files* (hlavičkové soubory) a *Resource Files*. A nyní k vložení zdrojového souboru do projektu.

Otevřete menu **File > New**. Klikněte na záložku **Files**, ale měla by se zobrazit automaticky. Opět nepoužívat ikonu . Ze seznamu typů souborů vyberte typ "C++ Source File". Do kolonky **File name** napište jméno souboru např. **main**. Příponu souboru **cpp** psát už nemusíte, je zvolena automaticky.

Zkontrolujte zda-li je zaškrtnuta volba "Add to project", protože jinak by se vám soubor vytvořil, ale ne ve vašem projektu. V kolonce **Location** (umístění) projektu nic neměňte, automaticky si nastavuje

adresář zvoleného projektu (v našem případě D:\cvi01). Potvrďte tlačítkem OK a zkontrolujte na záložce FileView jestli se tam soubor *main.cpp* nachází.

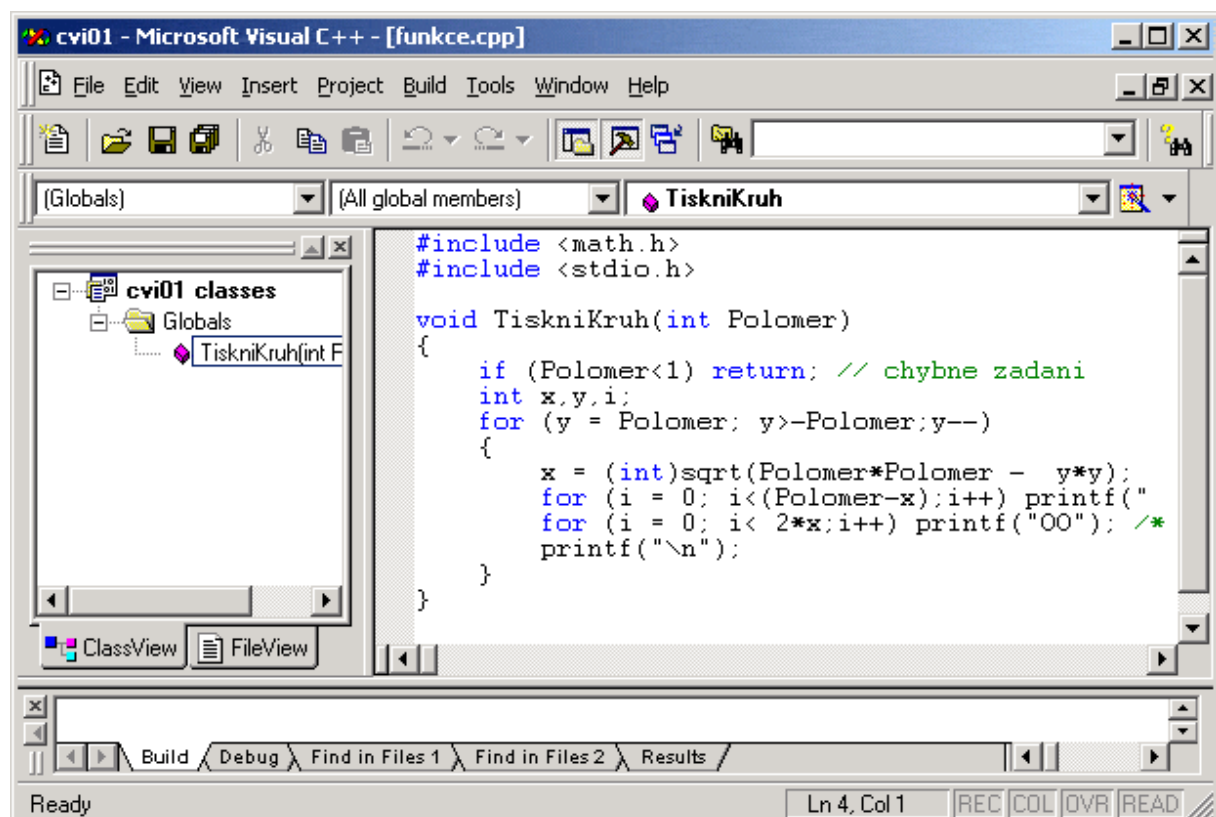
V okně FileView můžete také přidávat soubory do projektu stiskem pravého tlačítka myši přes volbu *Add Files to Folder* a mazat soubory z projektu klávesou *Delete*. Ale jakmile budete přidávat zdrojové soubory do projektu, tak si nejdříve nakopírujte do vašeho adresáře projektu a pak teprve přidejte. Pro pořádek je lepší mít soubory umístěné v adresáři projektu a ne někde v jiném adresáři. Pro vložení již vytvořených souborů do projektu musíme udělat následující. V adresáři se zadáním pro 1. cvičení (nalezneme na CD disku) se nachází dva soubory *funkce.cpp* a *funkce.h*.

Zkopírujte si tyto dva soubory do vašeho adresáře projektu. Přes pravé tlačítko myši "*Add Files to Folder*" přidejte soubor *funkce.cpp* do záložky "*Source Files*" a soubor *funkce.h* (přípona *.h určuje, že se jedná o speciální typ zdrojového souboru tzv. hlavičkový soubor) do záložky "*Header Files*". Faktem je že uvedené syntaxi jazyka C zatím nemusíte rozumět, ale to nevadí, v této chvíli se učíme pracovat s prostředím, se samotným jazykem C se seznámíme v kapitole 4. Snad jen pro pořádek prozradíme, že ve zdrojových souborech budeme mít zdrojový kód našeho programu a v hlavičkových souborech budou uloženy hlavičky (tzv. názvy funkcí se svými parametry), které budou zdrojové soubory využívat.

Přepínání mezi více soubory v projektu

Nyní máte v záložce "FileView" tři zdrojové soubory. Dva s příponou *.cpp v záložce "Source Files" a jeden s příponou *.h v záložce "Header Files". Dvojklikem myši soubory otevřete a pak je zavřete. Při spuštění a kompilaci programu nemusíte mít otevřen na obrazovce ani jeden zdrojový soubor. Kompilují a spouští se pouze ty soubory, které jsou uloženy v projektu (ne ty otevřené na obrazovce).

Přepínání mezi soubory lze vyřešit i programově. V programu v jazyce C si můžeme rozdělit program na více funkcí, z nichž různé funkce mohou být uloženy v jiných souborech. Nyní se na to podíváme ve Visual C++, který umožňuje otevřít danou funkci, aniž bychom věděli, kde se nachází.



Obr. 2.2 Přepínání záložek ClassView a FileView

Otevřete záložku ClassView v nichž se nachází naše funkce. Klikněte na tlačítko + a rozbalte záložky "cvi01classes" a "Globals". Dvojklikem otevřete funkci "TiskniKruh". Všimněte si, že po otevření máte kurzor přímo u dané funkce obr. 2.2.

Zavření a obnovení oken projektu

Může se někdy stát, že nechtěně zavřete nějaké okno ve Visual C++. Buďto bude okno Workspace (obsahuje záložky ClassView a FileView), nebo okno Output (sloužící např. pro kompilaci programu a hledání chyb) nebo okna zdrojových souborů. Ale i s těmito všemi zavřenými okny půjde program spustit, pokud máte aktivní projekt stále otevřen. Zavřete si všechna okna vývojového prostředí Visual C++. Pro zavření použijte malé tlačítko *x* nacházející se v rozích oken. Pozadí prostředí by mělo být šedé. Aniž by se mohlo zdát, že zůstalo něco otevřeno, tak stále máte otevřený svůj projekt, ale jen nejde vidět.

Okno Workspace obnovte z menu View > Workspace (**klávesová zkratka Alt + 0**) a okno Output z menu View > Output (**klávesová zkratka Alt + 2**).

Otevřete a zavřete si okna svých souborů v projektu. Otevřete je dvojklikem a zavřete *x*. Změňte si velikosti oken tažením za okraj okna. Otevřete si soubory funkce.cpp a funkce.h a použijte příkazy *Cascade*, *Tile Horizontally* a *Tile Vertically* z menu Window. Pokud nechcete zajíždět do menu nebo používat klávesové zkratky, tak existuje ještě cesta přes dvě ikony nacházející se na liště pod nabídkou menu. Ikona  pro okno Workspace a ikona  pro okno Output. Obě slouží k zavření a otevření popsaných oken. Vedle těchto ikon se nachází ještě ikona Windows List  sloužící k aktivování, zavírání, ukládání, srovnání horizontálně a vertikálně jednoho nebo více oken souborů umístěných v záložce FileView.



Ke kapitole 2.1 a 2.2 je přiřazena demonstrační animace č. 1

Animace č. 1 je věnována založení projektu, vkládání souborů do projektu a základní popis prostředí.

2.3 Psaní zdrojového kódu

Barevné značení textu

Při psaní zdrojového kódu v editoru Visual C++ si všimněte jeho barevné odlišnosti, která se vztahuje na typ souboru a typ psaného textu. Barevné značení psaného textu rozlišuje tři typy barev:

- černá - obecný a neznámý text, který nespádá do žádné kategorie
- modrá - klíčové slova jazyka C
- zelená - komentáře jazyka C

Jako příklad barevného značení si napíšeme text do vytvořeného souboru, kde nic nemáme, např. *main.cpp* v záložce Source Files. Programování v jazyce C je pro nás stále neznámou, takže se významem zdrojového textu nemusíme zatím zabývat. Následující text jenom opíšeme tak, jak je zobrazeno na následujícím obrázku.

Zarovnání zdrojového textu

Při psaní textu do souboru *main.cpp* jste si mohli všimnout, že editor Visual C++ po napsání závorky { a zmáčknutí tlačítka Enter odsunul kurzor na řádku více doprava. Snažte se tedy dodržovat to, že každý příkaz budete psát na samostatný řádek. Je to přehlednější a lépe se pak odhalují chyby.

Pokud tedy máte napsaný program a pro jeho lepší orientaci chcete provést formátovací změny, používejte tyto klávesy (kurzor je umístěn před textem):

```
# include "Funkce.h"

void main()
{
    TiskniKruh(10); /*vsimnete si ze po zapsani TiskniKruh( se zobrazi
                    napoveda*/
}
```

Obr. 2.3 Barevné odlišení textu

- klávesa **Tab** pro posun textu doprava,
- klávesa **Backspace** pro posun textu doleva,
- kombinace kláves **Shit + Tab** pro posun kurzoru doleva,
- kombinace kláves **Alt + F8** zarovná označený text do optimálního zarovnání (doporučeno používat toto formátování).

Jako praktický příklad na formátování zdrojového textu použijeme soubor *TiskniKruh*.

1. Otevřete soubor *TiskniKruh* ze záložky *ClassView*.
2. Pomocí kláves **Alt + F8** zformátujte zdrojový text podle obr. 2.4, pro označení textu použijte kombinaci kláves **Ctrl + a**.

```
#include <math.h>
#include <stdio.h>

void TiskniKruh(int Polomer)
{
    if (Polomer<1) return; // chybne zadani
    int x,y,i;
    for (y = Polomer; y>-Polomer;y--)
    {
        x = (int)sqrt(Polomer*Polomer - y*y);
        for (i = 0; i<(Polomer-x);i++) printf(" ");
        for (i = 0; i< 2*x;i++) printf("00"); /* Tistení vnitru kruhu */
        printf("\n");
    }
}
```

Obr. 2.4 Zarovnání zdrojového textu

2.4 Kompilace a spuštění programu

Co vše provádí Visual C++ za nás:

- pokud jsme ještě před samotným spuštěním programu neprovedli kompilaci projektu, tak nám ji sám nabídne,
- při každé kompilaci projektu si ukládá soubory na disk (tzn. že kdyby došlo k nějaké havárii a ukončení programu, tak Visual C++ automaticky zálohují soubory, co jsou v projektu).

Při každé kompilaci projektu vytváří spustitelný EXE soubor (ten se standardně ukládá do adresáře vašeho projektu a nachází se v podadresářích Debug nebo Release).

Kompilace programu se provádí z menu *Build > Build cvi01.exe*. Můžeme také použít klávesovou zkratku F7 nebo rovnou ikonu  z nástrojové lišty. Příkaz zkompiluje a sestaví všechny potřebné soubory, ale nespustí je.

Při kompilaci větších programů používáme příkaz z menu *Build > Rebuild All*, který nám provede rekompilaci celého programu (tzn. zkompiluje a sestaví program úplně od začátku). Začne vlastně tím, že smaže všechny předchozí kompilace a vytvoří zcela nové.

Zkompilujte program pomocí příkazu *Build*.

Jestliže kompilace a sestavování projektu proběhla bez problémů, měli byste v okně Output na záložce Build dostat tuto zprávu: „cvi01.exe - 0 error(s), 0 - warning(s“).

Kdyby se vyskytly nějaké chyby u slova error(s), tak musíme provést hledání a opravu chyby.

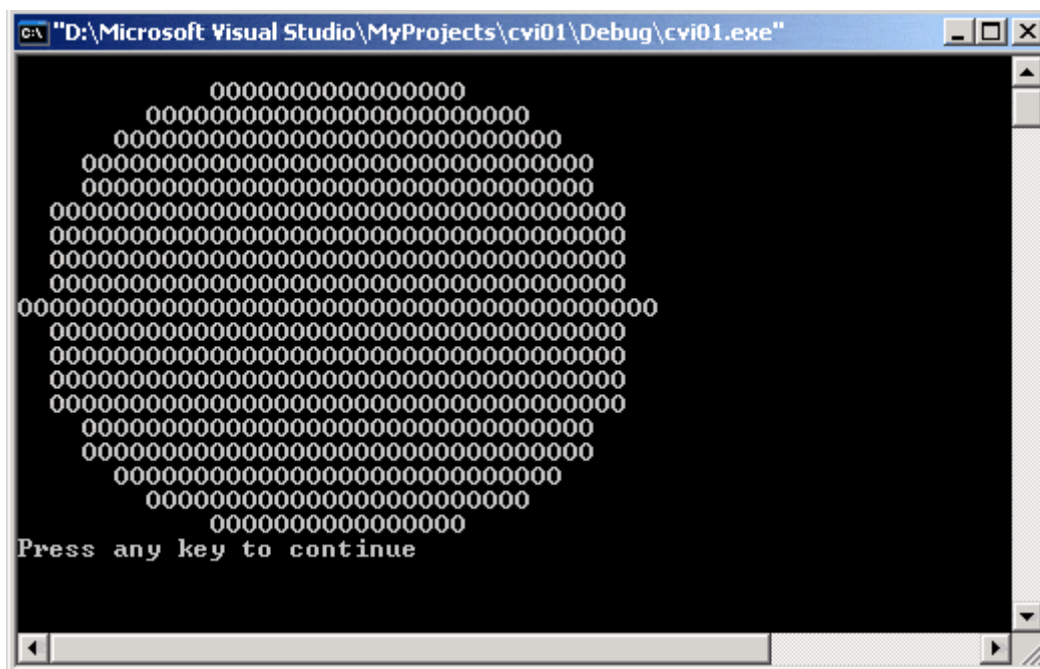
Spuštění programu v běžném módu

Spuštění programu ve Visual C++ lze provádět dvěma způsoby:

- v běžném módu (nejjednodušší způsob, ale nemůžeme program ladit a krokovat)
- v ladícím módu (nejčastější způsob hledání chyby v programu tzv. Debug mód, zastavování a ladění programu), ale tímto módem se budeme zabývat později.

Spustíte program z menu *Build > Execute cvi01.exe*, nebo kombinací kláves **Ctrl + F5** nebo ikonou  na nástrojové liště.

Všimněte si, že po skončení se vám okno nezavře. V běžném módu nemůžete program zastavit, takže po skončení se vám okno automaticky nezavře. Zavře ho až stisk nějaké klávesy (*Press any key to continue*) viz. obr. 2.5.



Obr. 2.5 Výstupní okno programu



Ke kapitole 2.3 a 2.4 je přiřazena demonstrační animace č. 2

Animace č. 2 je věnována psaní zdrojového textu, kompilaci a spuštění programu.

2.5 Vyhledávání chyb a jejich oprava

Kompilátor programu nám nedovolí úspěšně kompilaci dokončit, pokud se v programu vyskytnou chyby. Chyby mohou být různého charakteru. Nejčastější chybou začínajících programátorů je vynechání středníku na konci řádku. Často se také setkáváme s drobnými překlepy či vynecháním některého důležitého znaku. Tyto chyby kompilátor odhalí a upozorní nás na ně v okně s chybovým hlášením. Mnohem horší situací je chyba, kterou kompilátor neodhalí. Tyto chyby totiž nejsou syntaktickou chybou v programu, vše je napsáno správně, avšak program se nechová jak má, v horším případě se chová podivně jen v některých ojedinělých případech. V těchto programech je tedy z hlediska jazyka C vše v pořádku, chyba je logického charakteru a je pouze na programátorovi, aby příslušným laděním programu zajistil její vyhledání. Nyní si pro názornou ukázkou vytvoříme několik chyb, které si následně opravíme. Bude se tedy jednat o syntaktické chyby.

- Vytvořte chybu v souboru *funkce.cpp*, na řádku `printf("\n");` smažte písmeno `f`.
- Vytvořte chybu v souboru *main.cpp*, na řádku `TiskniKruh(10);` změňte písmeno velké `K` na malé `k`.

Oprava vytvořených chyb

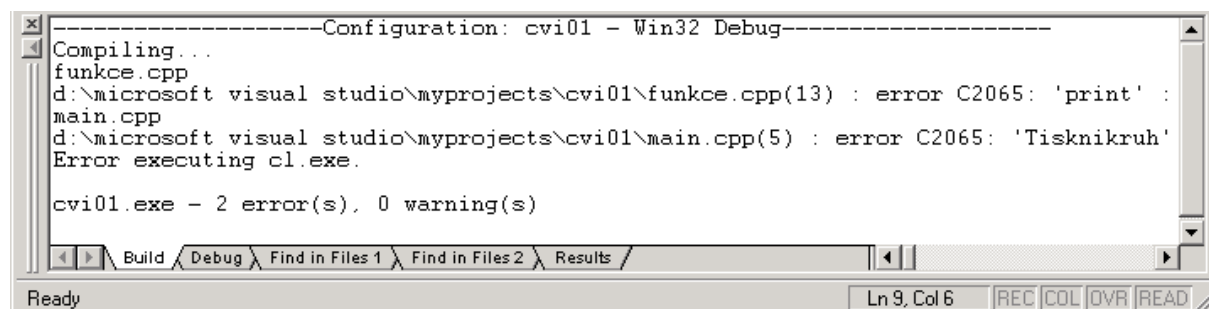
Vytvořené chyby se projeví po kompilaci programu. Okno Output nám napíše celkový počet chyb. Ale jejich skutečný počet bývá menší, protože většina chyb se řetězí. Takže na počet chyb moc nehleďte, protože další chyby mohou nastat vlivem první a nemusí být vůbec chybami.

Vždy začínáme s opravou první chyby. Po každé opravě chyby provedeme kompilaci a pak se pustíme do opravy další chyby. Zkompilujte program klávesou F7. V okně Output obr. 2.6 se vypíše počet nalezených chyb: 2 error(s).

Posuňte posuvník okna úplně nahoru nebo si okno zvětšete, abyste viděli na celý výpis hlášení. Všimněte si, že hlášení píše jméno souboru kde chyba nastala, cestu k souboru a v závorce na jakém řádku chyba vznikla.

Nalezněte první chybu v souboru *funkce.cpp* a dvojklikem se přesunete na řádek, kde chyba nastala. U řádku s chybou se objeví . Opravte chybu a spusťte kompilaci F7.

Počet chyb se snížil na jednu. Kompilace souboru *funkce.cpp* proběhla bez chyb. Poslední chyba nastala v souboru *main.cpp*. Přesuňte se na chybu a opravte ji. Proveďte kompilaci. Počet chyb je nula. Spusťte program klávesou zkratkou Ctrl + F5.



Obr. 2.6 Výstupní okno Output pro hlášení chyb v programu

Popisy jednodušších chyb:

- *undeclared identifier* - neznámé slovo. Jazyk C neví, co znamená. Nejčastěji se jedná o překlep či záměnu velkých a malých písmen,
- *missing function header* - nejčastěji znamená napsaný středník tam, kde nepatří,
- *unexpected* (neočekávaný výskyt)- často znamená, že jsme na předchozím řádku zapomněli středník.

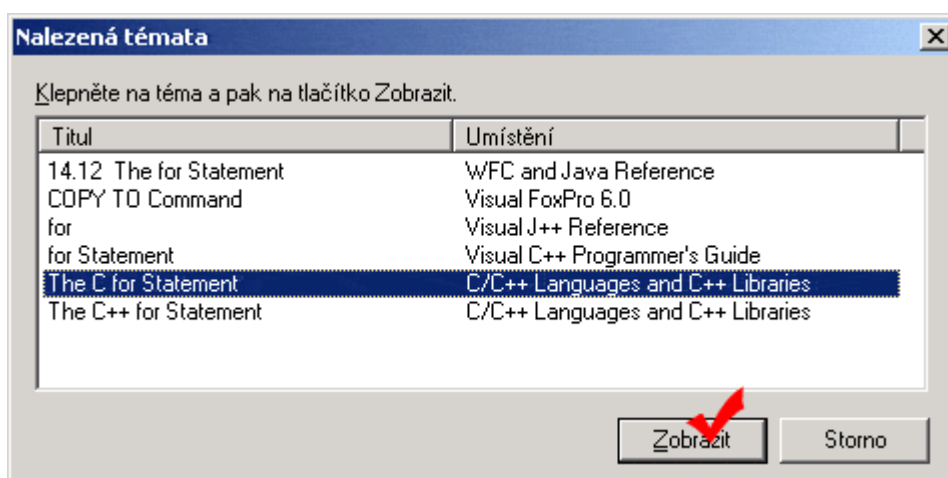
Nyní jsme si ukázali opravu jednoduchých chyb vznikajících nejčastěji překlepy v psaní programu, chybějícím středníkem atd. I programátor je pouze člověk a neexistuje nikdo, kdo by napsal dlouhý program bez jakékoli – byť jen drobné chyby. Z tohoto důvodu si nelamte hlavu, když po spuštění vašich prvních programů kompilátor zahlásí chybu. Postupně se budete v jednotlivých příkladech zdokonalovat, až některé programy budete psát zcela bezchybně.

2.6 Náповěda MSDN

K Visual Studiu není dodávána standardní nápověda jako u jiných programů. Nápovědu si je nutno nainstalovat zvlášť. Pracuje jako samostatná aplikace, ale je také propojena s prostředím Visual Studia.

Použití nápovědy přes kurzor

Nejjednodušší způsob používání nápovědy je umístit kurzor kdekoli v zdrojovém textu a stisknout klávesu F1 (připomeňme si, že klíčová slova jazyku C jsou značena modrou barvou např. *for*, viz. barevné značení textu). Umístěte kurzor na slovo *for* a stiskněte F1. Zobrazí se dialogové okno nápovědy MSDN, které nám vypíše nalezená témata pro náš výraz.



Obr.2.7 Nalezená témata nápovědy pro slovo *for*

Okno s nalezenými tématy nám nabídne nápovědu, pro celou řadu programovacích jazyků. Ale v našem případě, jelikož pracujeme s jazykem C, nás na záložce **Location** (Umístění) bude zajímat položka **C/C++ Languages and C++ Libraries**. Ale pořád nám na záložce **Title** (Titul), zbývá volba mezi významy The C for Statement a The C++ for Statement. Zvolte správný titul **The C for Statement** a stiskněte Display (Zobrazit). Objeví se hlavní okno nápovědy, které je rozděleno na dvě části. V levé části okna je stromový seznam všech knih v knihovně MSDN. V pravé části je zobrazena stránka pro náš hledaný výraz **for**. Stiskněte tlačítko **Locate** (Najít) z nástrojového panelu sloužící pro synchronizaci pravé a levé části okna. V levém okně se objeví kapitoly s podobným významem jako námi hledaný výraz. Ale nemusí tomu tak být vždy, např. u funkce `printf` se nám neobjeví kapitoly s podobným významem, ale abecedně řazený seznam funkcí a výrazů začínající písmenem p.

Záložka Context (Obsah) obsahuje témata, která jsou řazena podle druhu kapitoly do jednotlivých podkapitol (jako v knihách).

Index (Rejstřík)

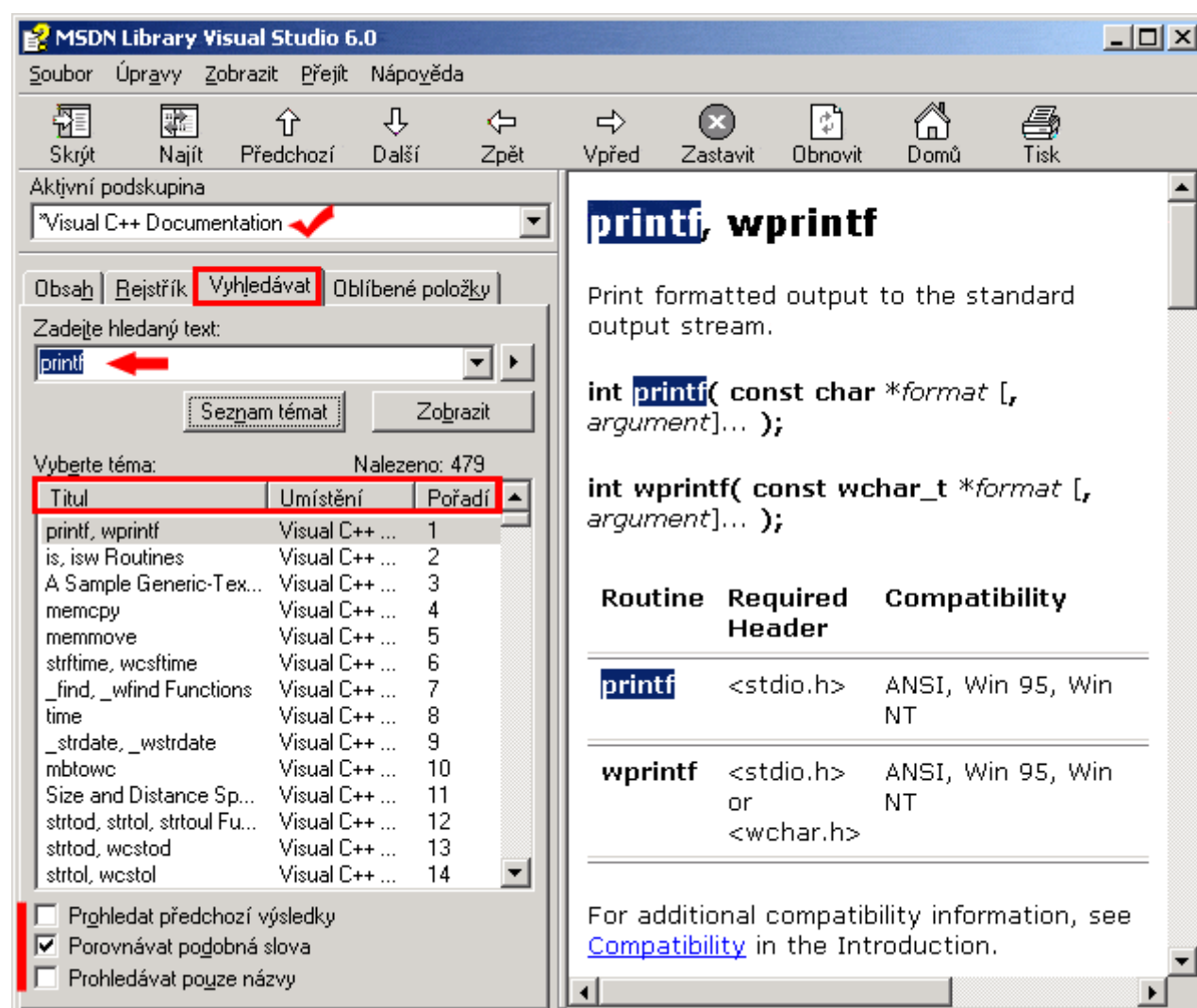
Patří mezi nejčastější vyhledávací metody. Do rámečku zadáte hledané klíčové slovo a MSDN knihovna vám ho okamžitě vyhledá. Vyhledává pouze položky v rejstříku (abecedně řazeném seznamu).

Všimněte si také, že výrazy a funkce mohou začínat malými nebo velkými písmeny. Pro naše potřeby se budeme zaměřovat na nalezené výrazy a funkce psané hlavně malými písmeny (velká písmena na začátku většinou souvisí s jinou problematikou než je jazyk C).

Jako klíčové slovo použijte funkci **printf** a stiskněte Enter nebo dvojklik myši.

Search (Vyhledávat)

Tato metoda používá vyhledávání podobných slov na stránkách, které pak řadí podle úspěšnosti výskytu hledaného výrazu na stránce. Do karty „vyhledávat“ zadáte hledaný text printf viz. obr. 2.8.



Obr. 2.8 Nalezení funkce pomocí záložky Search (Vyhledávat)

Nalezené témata si můžete řadit podle **titulu** (abecední řazení hledaného textu na stránkách), **umístění** (abecední seznam oblastí, kde se náš text nachází) a **pořadí** (ve sloupci pořadí se vypíše nalezené téma podle výskytu hledaného textu). V pravé části okna si všimněte modrého zbarvení pozadí hledaného textu **printf**.

Všimněte si také možností voleb v levém spodním rohu okna, které umožňují prohledávat předchozí výsledky, podobná slova nebo pouze názvy. Zatrhněte si volby „*Porovnávat*“ podobná slova a „*Prohledávat*“ pouze názvy (prohledává jen nadpisy stránek s hledaným textem). Všimněte si také položky o počtu nalezených témat.



Ke kapitole 2.5 a 2.6 je přiřazena demonstrační animace č. 3

Animace č. 3 je věnována vyhledáváním chyb, jejich opravou a prací s nápovědou MSDN.



Další zdroje

1. The documentation for Visual C++. MSDN Library Visual Studio 6.0.
2. Visual C++ Developer Center [online] dostupný z [http:// msdn.microsoft.com/visualc](http://msdn.microsoft.com/visualc)
3. Herout, P. *Učebnice jazyka C*, České Budějovice, Kopp 2005, počet stran: 280. ISBN 80-7232-220-6.



Shrnutí kapitoly

Projekt je adresář, do kterého si budeme ukládat soubory, se kterými budeme pracovat (např. zdrojové soubory jazyka C nebo hlavičkové soubory). Nový projekt si založte z menu File > New. Jako typ projektu budeme v našich cvičeních volit *Win32 Console Application*. Visual C++ obsahuje hodně typů projektů pro celou řadu programů, ale my se budeme zabývat pouze typem *Win32 Console Application*, který je určen pro práci s konzolovými aplikacemi jazyka C. Záložka ClassView zobrazuje jednotlivé třídy projektu, včetně jejich funkcí a proměnných. *FileView* zobrazuje všechny soubory, které projekt obsahuje a jsou rozříděny do jednotlivých skupin *Source Files* (zdrojové soubory), *Header Files* (hlavičkové soubory) a *Resource Files*.

Okno Workspace obnovíme z menu View > Workspace (klávesová zkratka Alt + 0) a okno Output z menu View > Output (klávesová zkratka Alt + 2). Pomocí kláves Alt + F8 zformátujete zdrojový text, pro označení celého textu použijete kombinaci kláves Ctrl + a.

Při psaní zdrojového kódu v editoru Visual C++ si všimneme jeho barevné odlišnosti, která se vztahuje na typ souboru a typ psaného textu. Barevné značení psaného textu rozlišuje tři typy barev: černá - obecný a neznámý text, který nespadá do žádné kategorie, modrá - klíčové slova jazyka C a zelená - komentáře jazyka C.

Při každé kompilaci projektu se vytváří spustitelný EXE soubor. Kompilace programu se provádí z menu *Build*. Můžeme také použít klávesovou zkratku F7 nebo rovnou ikonu z nástrojové lišty. Příkaz zkompile a sestaví všechny potřebné soubory, ale nespustí je. Spuštění programu ve Visual C++ lze provádět dvěma způsoby: v běžném módu (nejjednodušší způsob, ale nemůžeme program ladit a krokovat) a v ladícím módu (nejčastější způsob hledání chyby v programu - tzv. Debug mód, zastavování a ladění

programu).

Kompilátor programu nám nedovolí úspěšně kompilaci dokončit, pokud se v programu vyskytnou chyby. Chyby mohou být různého charakteru. Překlepy a syntaktické chyby kompilátor odhalí a upozorní nás na ně v okně s chybovým hlášením. Mnohem horší situací je chyba, kterou kompilátor neodhalí. Tyto chyby totiž nejsou syntaktickou chybou v programu, vše je napsáno správně, avšak program se nechová jak má, v horším případě se chová podivně jen v některých ojedinělých případech.



Úkol k řešení 2.1

Projděte ještě jednou animace k učebnímu textu a zopakujte si nezbytné úkony k založení projektu, psaní a kompilaci zdrojového kódu.



Úkol k řešení 2.2

Vytvořte vzorový projekt přesně podle animace a vyzkoušejte si vyhledávání chyb a jejich opravu.



Úkol k řešení 2.3

Vyzkoušejte si použití nápovědy MSDN. Vyzkoušejte Index a vyhledávání.



Kontrolní otázka 2.1

Jaké barevné značení textu používá editor Visual Studio?



Kontrolní otázka 2.2

Které chyby neodhalí kompilátor?



Kontrolní otázka 2.3

Chci napsat jednoduchý program na pár řádků, musím vytvářet projekt?



Kontrolní otázka 2.4

Co obsahuje záložka Header Files?

3. Ladění vlastního programu ve Visual Studiu 6.0



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- nastavit prostředí pro krokování programu
- krokovat program po jednotlivých řádcích zdrojového kódu
- krokovat program dovnitř funkcí ve zdrojovém kódu
- používat záložky při ladění programu
- nastavit u záložek sledované podmínky
- sledovat hodnoty proměnných za běhu ladění programu



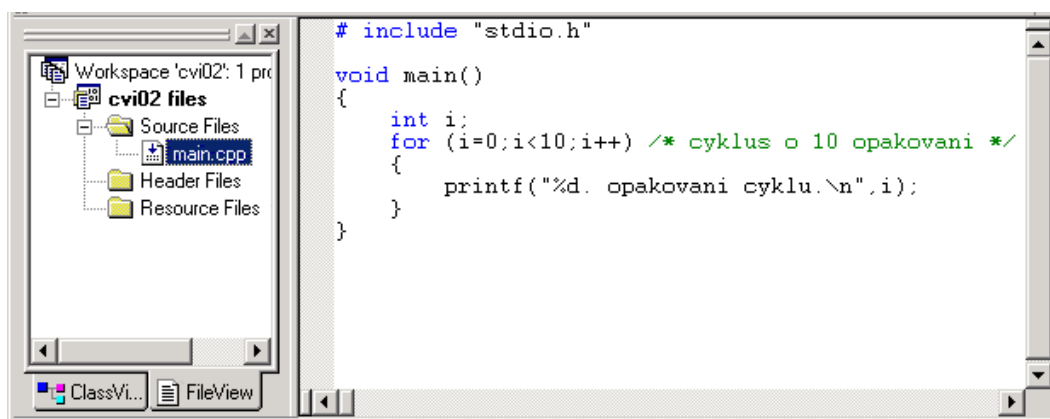
Výklad

Obsahem předchozí kapitoly bylo úvodní seznámení s vývojovým prostředím Visual Studia. Dnešní kapitola, jež je náplní druhého praktického cvičení, je práce s laděním a krokováním programu. Uvedený postup je předveden na velice jednoduchých příkladech, neboť se syntaxí jazyka C se seznámíme na dalších cvičeních. Přesto jsou příklady pochopitelné a získané zkušenosti využijeme při tvorbě složitějších programů v průběhu semestru.

3.1 Nastavení ladění programu

V prvním cvičení jsme si v kapitole Kompilace a spuštění programu ukázali spuštění programu v běžném módu. Nyní si ukážeme spuštění programu v ladícím módu (tzv. *Debug módu*) a hlouběji si ho rozebereme. Při ladění se pokoušíme hledat a odstraňovat chyby programu, zjišťovat hodnoty proměnných v různých částech programu atd.

Abychom mohli vůbec pracovat, musíme si založit nový projekt viz. 1. cvičení: Založení a volba typu projektu. Založte si nový projekt a pojmenujte si ho např. *cvi02* (nezapomeňte na správné umístění projektu: D:\cvi02). Během tohoto cvičení budeme používat pouze zdrojový soubor *main.cpp*, který budeme modifikovat pro naše potřeby. Pro různé podkapitoly budeme používat různé zdrojové texty, všechny zdrojové texty nalezneme na příloženém CD, popřípadě na online e-learningové podpoře. Tento zdrojový kód naleznete u příkladů k animaci č. 4 pod názvem *krokovani.txt*.



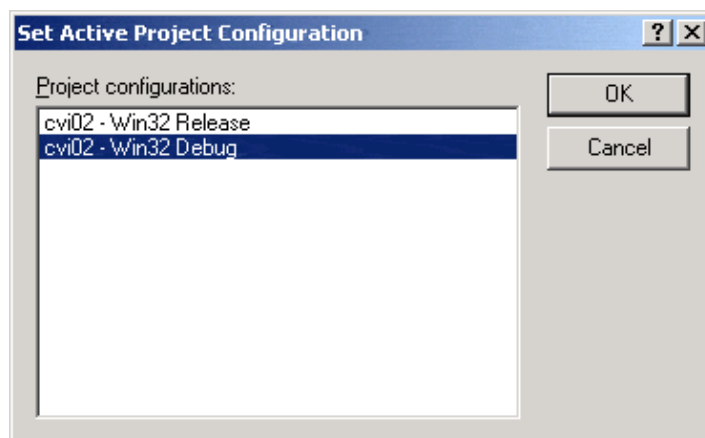
Obr. 3.1 Obsah souboru *main.cpp*, se kterým budeme v této kapitole pracovat

Uvedené příklady a postupy je možno sledovat také v přiložené animaci. Soubor si do projektu můžete vložit dvěma způsoby, viz. předchozí kapitola - odstavec *Vložení souboru do projektu*. Vložte si nový soubor do projektu a pojmenujte ho *main.cpp* (složka Source Files obr. 3.1).

Nastavení konfigurace projektu

K vytvoření spustitelné verze programu v ladícím módu se nejdříve musíte přepnout do konfigurace **Win32 Debug**. Ale v našem případě to slouží pouze jako kontrola, protože Visual C++ si automaticky při vytváření nového projektu nastavuje konfiguraci Win32 Debug (ladění). K přepínání dostupných konfigurací projektu slouží dialogové okno z menu *Build > Set Active Configuration* viz. obr. 3.2.

Svou aktuálně nastavenou konfiguraci projektu si můžete snadno zjistit z menu *Project > Settings*. Kde máte v záložce "Settings For" nastaveno Win32 Debug. Ověřte si konfiguraci nastavenou ve vašem projektu.

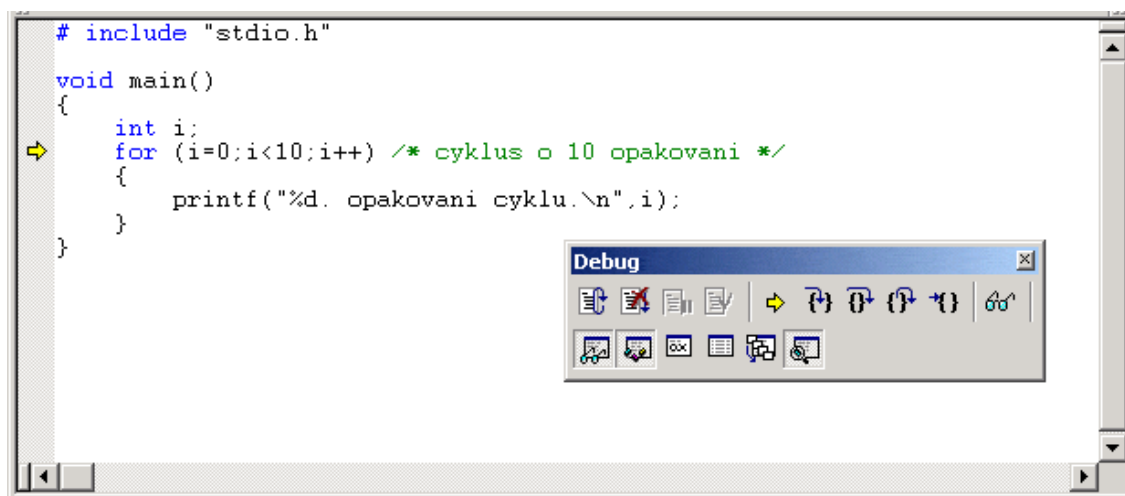


Obr. 3.2 Okno nastavení konfigurace ladění programu

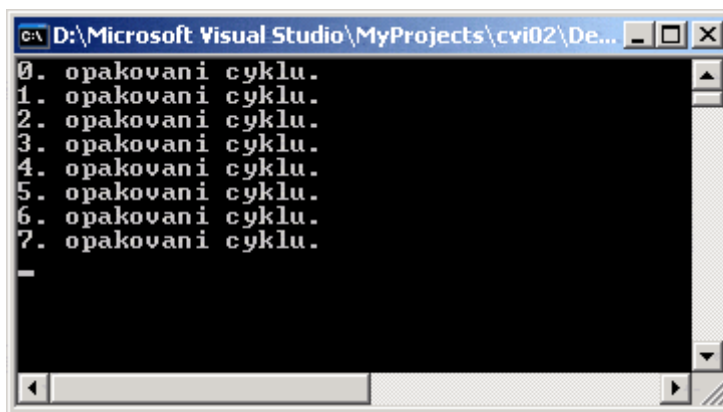
3.2 Krokování programu

Pokud máme program zkompileován, tak ho můžeme krokovat. Postupné krokování se provádí klávesou **F10** nebo ikonou  z panelu Debug. Programový kurzor  se pohybuje po řádcích programu (viz. obr. 3.3) a výsledky se postupně zapisují do výstupního okna programu (obr. 3.4).

1. Spustíte krokování klávesou **F10**.
2. Sledujte změnu výstupního okna.
3. Krokujte až na konec souboru.

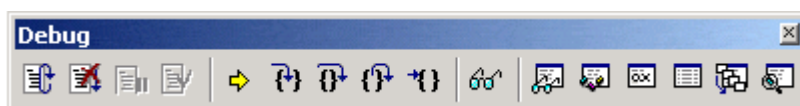


Obr. 3.3 Postupné krokování



Obr.3.4 Výstupní okno programu

Pokud by se vám při ladění automaticky nezobrazil panel **Debug**, tak stačí kliknout pravým tlačítkem myši kdekoli na nástrojovou lištu nebo na šedou plochu za lištou a z nabídky oken a panelů zvolte **Debug**.



Obr. 3.5 Plovoucí panel Debug

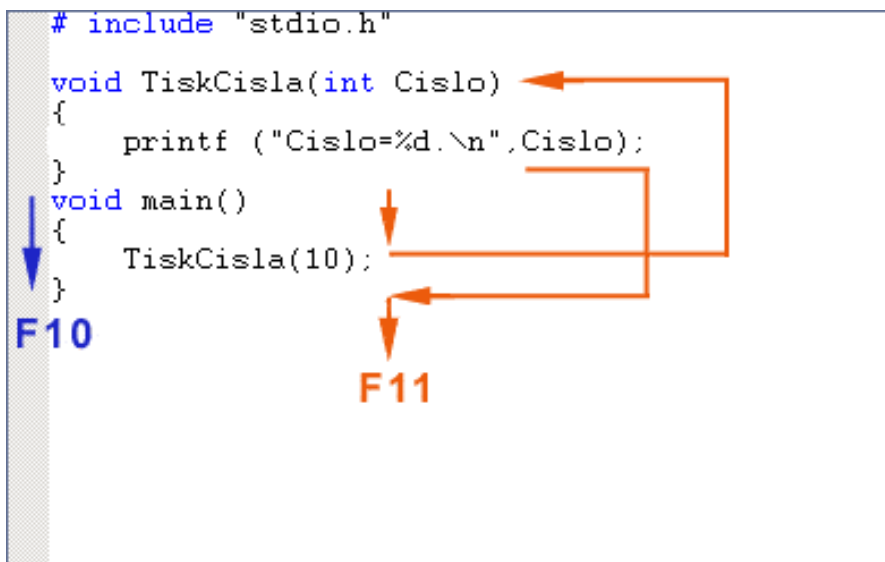
Při krokování jsme se dostali až mimo zdrojový text programu (konec funkce main). Visual C++ nám pak nabídne dialogové okno se zadáním cesty pro další zdrojový text. Uzavřete okno tlačítkem **Cancel**, protože další krokování už provádět nebudeme. Visual C++ pak zobrazí nové okno s neznámou částí programu pomocí Disassembleru. Ukončení ladícího režimu provedeme přes klávesovou zkratku **Shift + F5** nebo z menu *Debug > Stop Debugging* nebo ikonou  z panelu **Debug**.

3.3 Krokování dovnitř funkcí a ven z funkcí

Pro následující výklad použijeme další připravený zdrojový kód, který je uložen v textovém souboru *krokovani_dovnitř_ven.txt*. Rozdíl oproti postupnému krokování klávesou **F10** je ten, že se kurzor přesune i dovnitř funkce (v tomto případě funkce *TiskniCislo*) a po ukončení dále pokračuje v hlavním programu (viz. obr. 3.6).

Krokování dovnitř funkce se provádí klávesou **F11** nebo ikonou  z panelu Debug. Pomocí klávesy **Shift + F11** nebo ikony  z panelu Debug se dostanete ven z krokované funkce.

1. Přepište nebo zkopírujte si zdrojový text (soubor *krokovani_dovnitř_ven.txt*).
2. Vyzkoušejte si krokování klávesou **F11** a také **F10**.
3. Všimněte si rozdílného posunu  po řádcích.



Obr. 3.6 Krokování dovnitř a ven z funkce



Ke kapitolám 3.1 - 3.3 je přiřazena demonstrační animace č. 4

Animace č. 4 je věnována krokování programu a to postupnému i krokování dovnitř a ven z funkcí.

3.4 Zarážky (breakpointy)

Označují místa před nimiž se běh programu při ladění zastaví. Potom si můžeme zkontrolovat nebo změnit hodnoty proměnných a nebo pokračovat dále v krokování. Nejčastěji se používají při krokování rozsáhlejších zdrojových souborů, kde by krokování celého programu zabralo hodně času. Zarážky můžeme vkládat do zdrojového textu jak v režimu editace, tak i v režimu ladění.

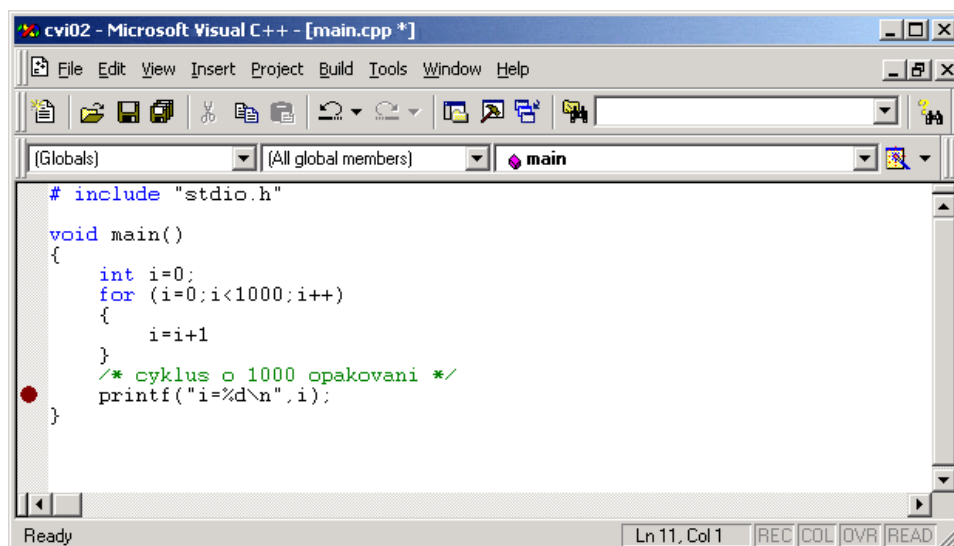
Příklad použití zarážek

1. Přepište nebo zkopírujte si zdrojový text k animaci č. 5 (*zarazky.txt*).

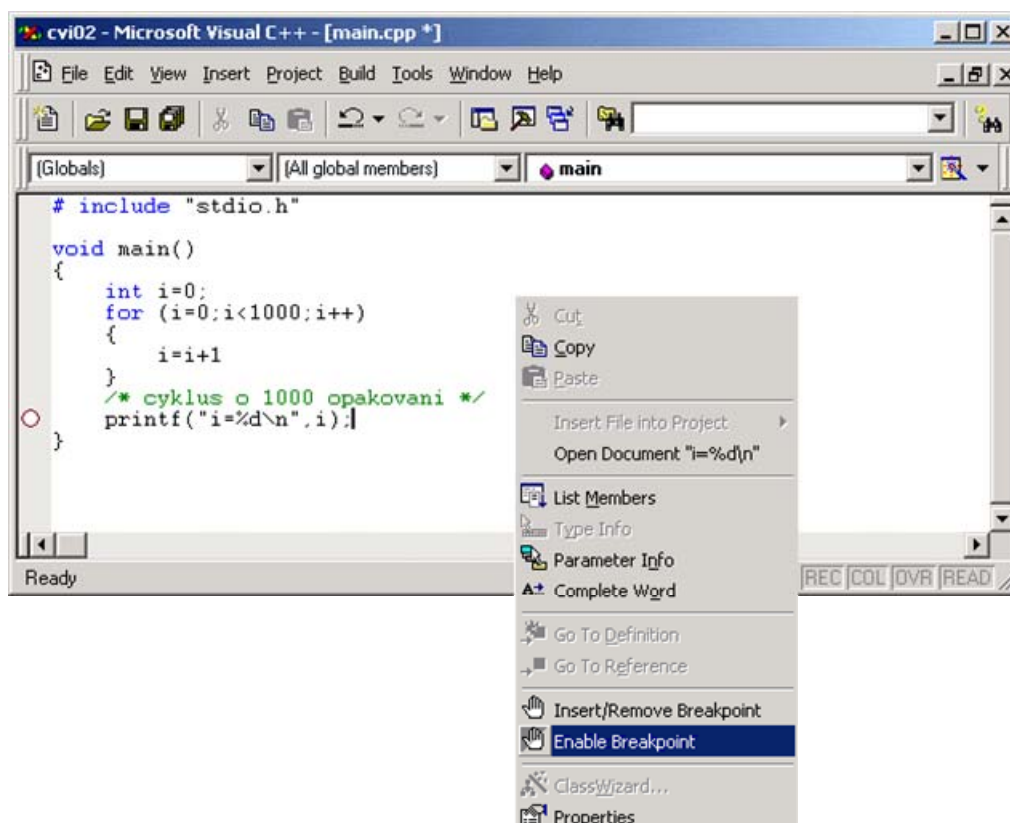
Zarážky (breakpointy)  se umísťují klepnutím kdekoli na řádek a stiskem klávesy **F9**. Klávesa **F9** vkládá a maže zarážky (viz. obr. 3.7). Také stisk pravého tlačítka myši a potvrzení nabídky *Insert/Remove Breakpoint* vloží a odstraní zarážku. Nabídka také obsahuje volbu *Disable Breakpoint*, která zarážku dočasně vypne, ale neodstraní. A volbu *Enable Breakpoint*, která zapne vypnutou zarážku.

Jakmile máte zarážku vloženou, tak spuštění ladění přes zarážky se provádí přes nabídku **Build > Start Debug > Go** nebo klávesou **F5**. Program se rozběhne a zastaví u první zarážky na kterou narazí.

1. Umístěte zarážku klávesou **F9** na řádek 11 zdrojového textu.
2. Zkuste si volbu přes pravé tlačítko myši (Enable/Disable Breakpoint) viz. obr. 3.8.



Obr.3.7 Umístění breakpointu (zarážky)



Obr.3.8 Volba pro práci s breakpointy

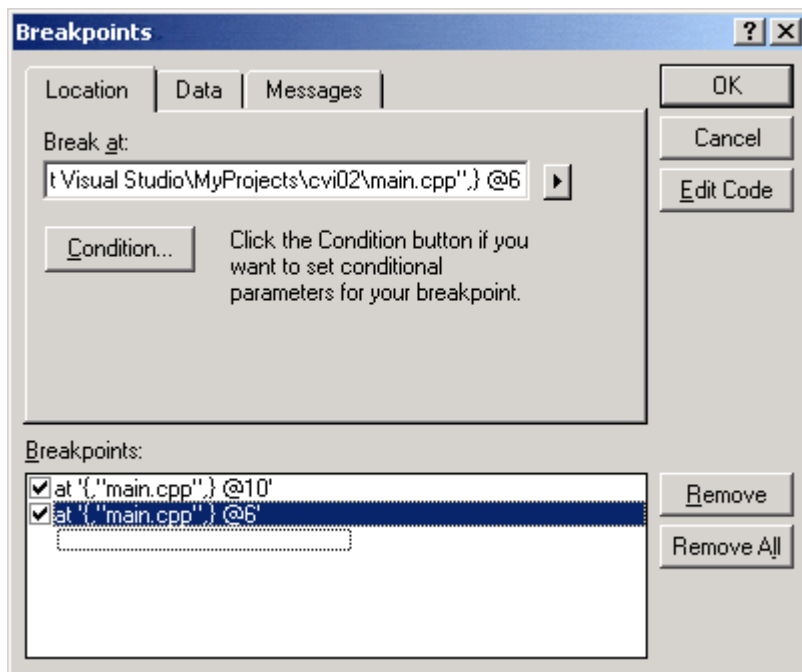
Odstranění všech zarážek

Máte-li nějaký rozsáhlejší program, do kterého umístíte větší množství zarážek, tak může být jejich odstraňování po jedné velmi zdlouhavé. V takovém případě používáme dialogové okno Breakpoints. Okno najdete v menu *Edit > Breakpoints* nebo přes klávesou zkratku **Alt + F9**.

Na záložce **Location** nalezneme v poli *Breakpoints* seznam všech zarážek ve vašem projektu, které jsou popsány jménem souboru a číslem řádku. Zaškrtnuté políčko u zarážky nadpovídá, že je zarážka zapnuta (*Enable Breakpoint*).

Tlačítka *Remove* a *Remove All* odstraní buď jednu nebo všechny zarážky v programu.

1. Umístíte si do souboru více zarážek.
2. Pomocí okna Breakpoints je odstraňte nebo vypněte.

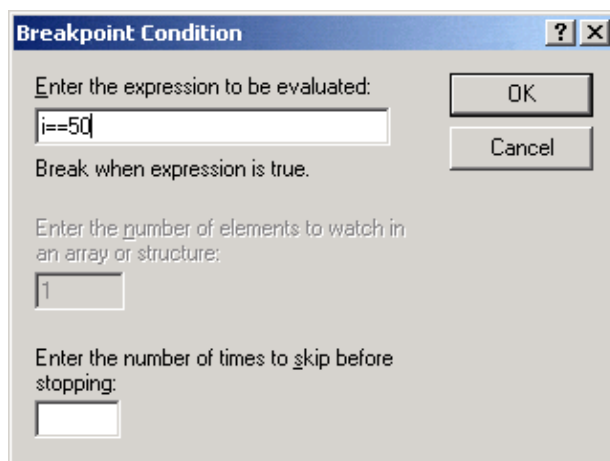


Obr. 3.9 Okno Breakpoints pro práci s breakpointy

3.5 Zarážka s podmínkou

Při ladění často potřebujeme přerušit program, je-li splněna nějaká podmínka. K tomu použijeme okno **Breakpoints**.

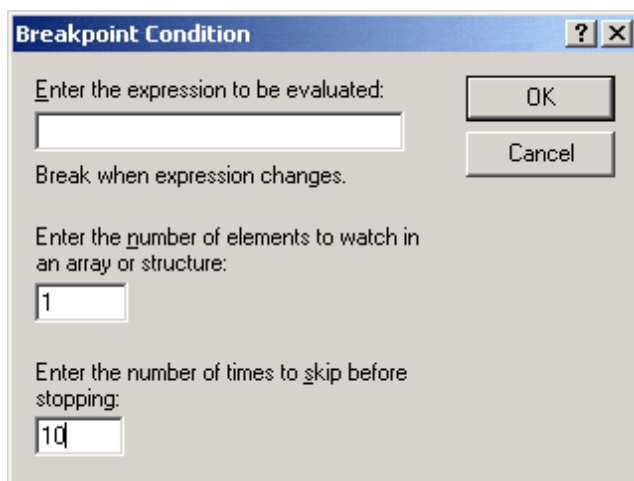
1. Umístíte zarážku na osmý řádek souboru *main.cpp*.
2. Otevřete si okno **Breakpoints** (Alt + F9). V poli Breakpoints si označte zarážku.
3. Stiskněte tlačítko Condition.
4. Objeví se okno Breakpoint Condition obr. 3.10. Do řádku *Enter the expression to be evaluated* (Zadejte výraz, který se má vyhodnotit) napište **i==50**.
5. Potvrďte a všimněte si, že v okně Breakpoints vám vedle zarážky přibyla podmínka **when 'i==50'**.
6. Stiskněte tlačítko OK a spusťte ladění přes zarážky (F5).
7. V informačním okně stiskněte OK.
8. Zastavení hodnoty proměnné v **i = 50** zjistíte pomocí hodnoty v bublině nebo okna **Watch**.



Obr. 3.10 Okno Breakpoint Condition

Následující příklad je zastavení programu přes zarážku při n-tém průchodu programem.

1. Opakujte kroky 1 až 3 z předchozího úkolu.
2. V okně Breakpoint Condition (viz. obr. 3.11) napište do řádku *Enter the number of times to skip before stopping* (Zadejte kolikrát se má zarážka přeskočit před zastavením programu) hodnotu **10**.
3. Spustěte ladění **F5**.
4. Hodnotu proměnné **i** při desátém průchodu přes zarážku zjistíte pomocí hodnoty v bublině nebo v okně **Variables** viz. následující kapitola Sledování hodnot proměnných.



Obr. 3.11 Okno Breakpoint Condition pro použití zarážky při n-tém průchodu



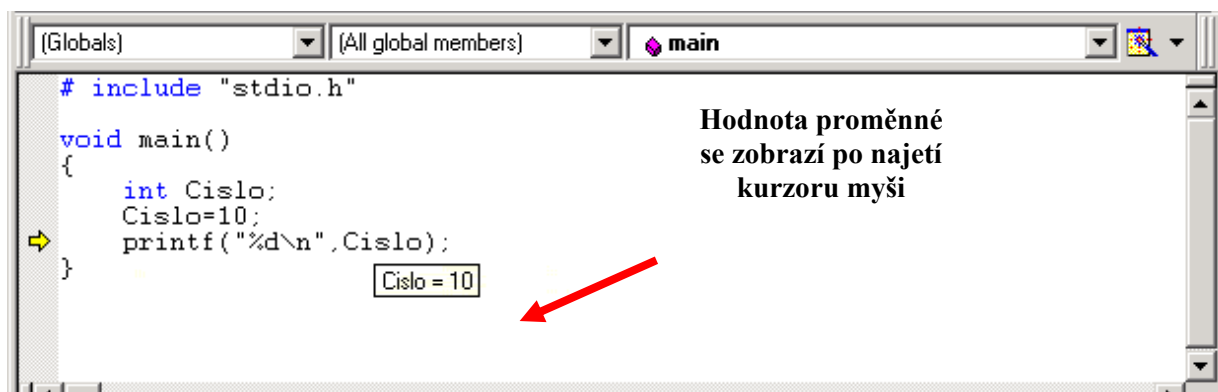
Ke kapitole 3.4. a 3.5 je přiřazena demonstrační animace č. 5

První polovina animace č. 5 je určena k procvičení práce s breakpointy neboli zarážkami.

3.6 Sledování hodnot proměnných

Nejjednodušší způsob, jak zjistit hodnotu proměnné během ladění je ukázat na ni kurzorem. Pod kurzorem se objeví rámeček s hodnotou proměnné (viz. obr. 3.12). Pro objekty, struktury, pole a jiné proměnné tento způsob nelze použít, protože se pod kurzorem zobrazí pouze {...}.

1. Přepište nebo zkopírujte si zdrojový text k animaci č. 5 (*sledovani_prom.txt*).
2. Spust'te krokování **F10**.
3. Sledujte hodnotu proměnné **Cislo** v jednotlivých řádcích pomocí kursoru.



Obr. 3.12 Sledování proměnné pomocí hodnoty v alt bublině

Okno s proměnnými (Variables)

V ladícím režimu by se okno mělo zobrazit automaticky. To samé platí i oknu **Watch**. Okno Variables si můžete vyvolat přes menu *View > Debug Windows > Variables* nebo jednodušeji přes klávesovou zkratku **Alt + 4** nebo přes panel Debug ikonou .

V horní části okna (viz. obr. 3.13) je rozbalovací seznam **Context**, kde si můžete zvolit funkci, ve které chcete prohlížet proměnnou (v případech překrývající se viditelnosti proměnných). Okno obsahující sloupce **Name** (jméno proměnné) a **Value** (hodnota proměnné) má tři záložky:

- **Auto** - zobrazuje proměnné a funkce vracující hodnoty,
- **Locals** - ukazují aktuální proměnné k funkci,
- **this** - používá se u objektových typů.

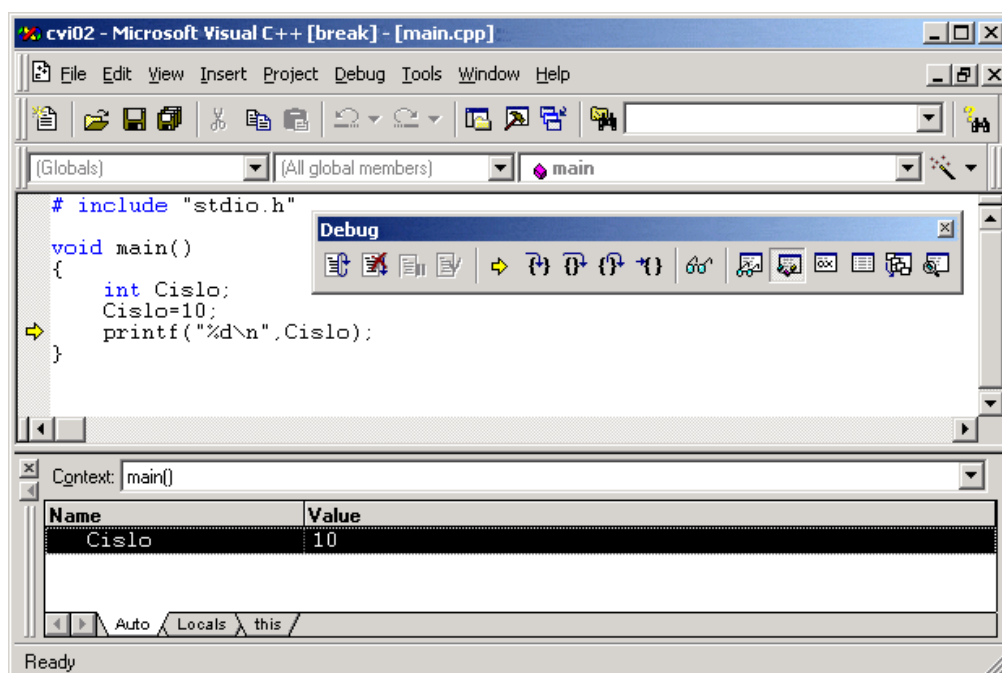
Pro přehlednost používá Visual C++ při zobrazování změněných proměnných (při minulém kroku) **červenou** barvu.

Dvojklikem na hodnotu proměnné ve sloupci *Value* jej můžete změnit. Stiskem klávesy **Enter** potvrdíte novou hodnotu proměnné.

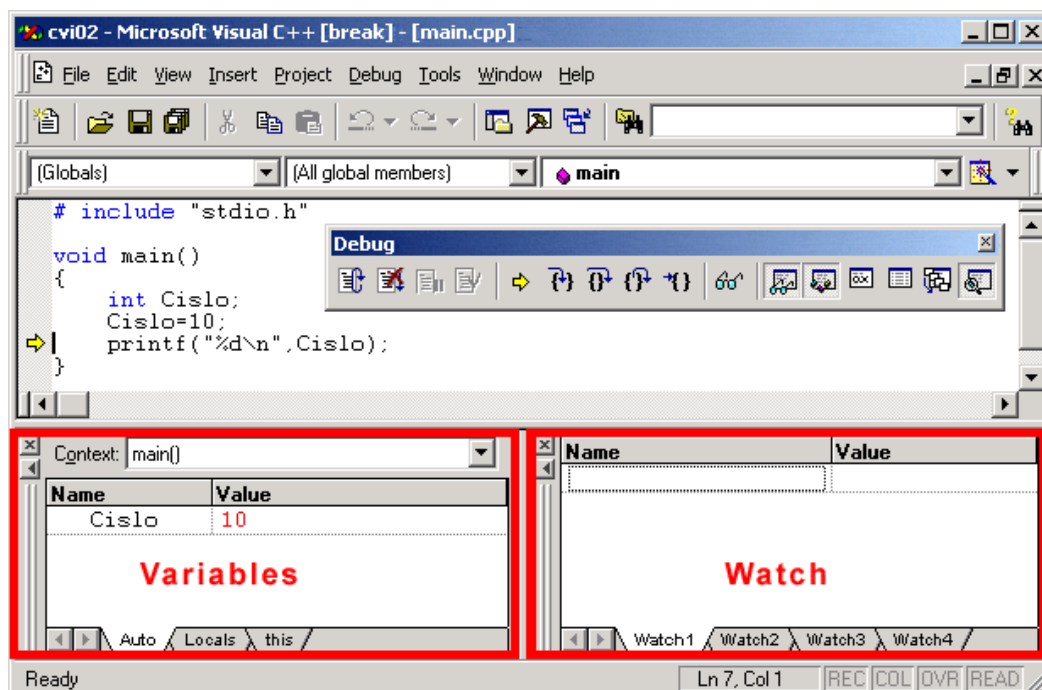
Krokujte klávesou **F10** a sledujte hodnotu proměnné v okně *Variables*.

Okno se sledovanými výrazy (Watch)

Toto okno se objevuje v ladícím režimu automaticky. Pokud ne, tak si ho můžete vyvolat přes menu *View > Debug Windows > Watch* nebo jednodušeji přes klávesovou zkratku **Alt + 3** nebo klikem na ikonu  v panelu Debug. Standardně jsou okna **Variables** a **Watch** umístěna vedle sebe (viz. obr. 3.14). Šířku oken lze měnit tažením myši za dělicí čáru oken.



Obr. 3.13 Okno Variables



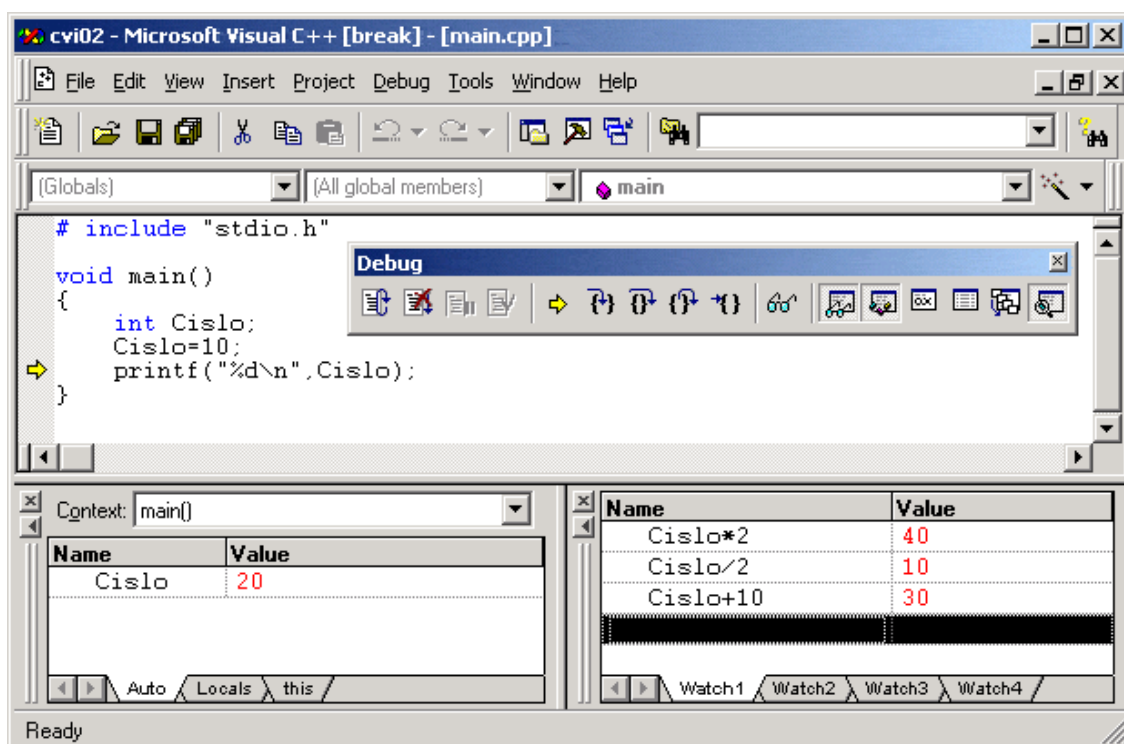
Obr. 3.14 Okna Variables a Watch

Okno **Watch** umožňuje sledovat hodnoty složitějších výrazů (nejen proměnných, ale i výrazů obsahujících operátory a funkce). Editace se provádí stejně jako u okna **Variables**. Odstranění provedete označením výrazu nebo proměnné a stiskem klávesy Delete. Proměnné, které do okna **Watch** vložíte, tam zůstávají uloženy.

Sledovat hodnotu můžete také označením identifikátoru proměnné nebo výrazu (ve zdrojovém textu) a přetažením myši do okna **Watch**.

1. Vynásobte proměnou **Cislo** v okně **Watch** ve sloupci Name dvěma při hodnotě proměnné v okně **Variables** **Cislo** = 10.
2. Ve zdrojovém textu označte proměnou **Cislo** a přetáhněte ji pomocí myši do okna **Watch**.
3. Vyzkoušejte si různé aritmetické operace v okně **Watch**.
4. V okně **Variables** změňte hodnotu proměnné **Cislo** na 20 a sledujte změnu proměnné v okně **Watch** (viz. obr. 3.15).

Okno **Watch** také obsahuje několik záložek (Watch1, Watch2 atd.). Na každé ze záložek můžete zadat jiné sledované výrazy a pak se přepínat podle toho, kterou skupinu (záložku) chcete vidět. Ale pozor, vyhodnocuje se pouze aktuálně vybraná skupina!



Obr. 3.15 Aritmetické operace v okně Watch

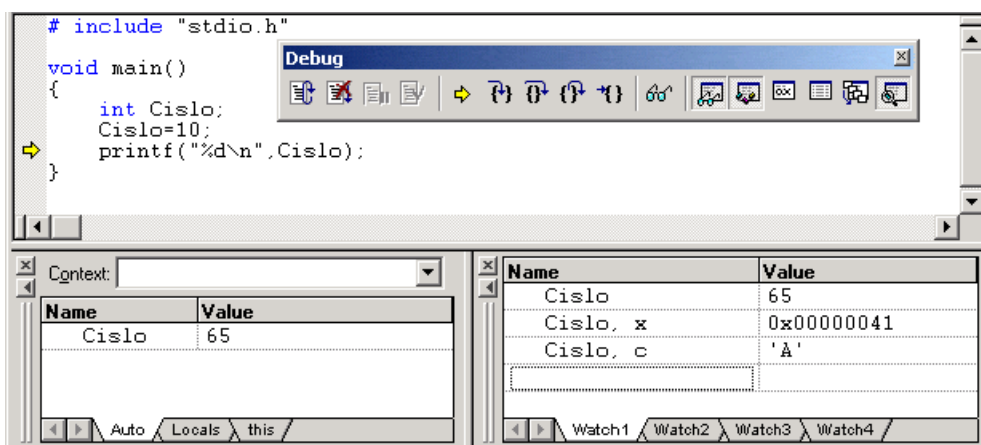
V okně **Watch** si můžete hodnoty proměnných naformátovat do tvaru: - HEX (**proměnná, x**)
- ASCII (**proměnná, c**)

1. Změňte hodnotu proměnné **Cislo** na 65 v okně **Variables**.
2. V okně **Watch** proveďte tyto úpravy (viz. obr. 3.16).



Ke kapitole 3.6 je přiřazena druhá polovina animace č. 5

Druhá polovina animace č. 5 je určena k procvičení sledování hodnot proměnných.



Obr. 3.16 Převod čísla v okně Watch na HEXa a ASCII tvar



Další zdroje

1. The documentation for Visual C++. MSDN Library Visual Studio 6.0.
2. Visual C++ Developer Center [online] dostupný z <http://msdn.microsoft.com/visualc>
3. Herout, P. *Učebnice jazyka C*, České Budějovice, Kopp 2005, počet stran: 280. ISBN 80-7232-220-6.



Shrnutí kapitoly

Pokud máme program zkompileován, tak ho můžeme krokovat. Postupné krokování se provádí klávesou F10 nebo ikonou z panelu Debug. Programový kurzor se pohybuje po řádcích programu a výsledky se postupně zapisují do výstupního okna programu. Krokování dovnitř funkce se provádí klávesou F11. Pomocí klávesy Shift + F11 se dostanete ven z krokované funkce. Rozdíl oproti postupnému krokování klávesou F10 je ten, že se kurzor přesune i dovnitř funkce.

Zarážky označují místa před nimiž se běh programu při ladění zastaví. Potom si můžeme zkontrolovat nebo změnit hodnoty proměnných a nebo pokračovat dále v krokování. Nejčastěji se používají při krokování rozsáhlejších zdrojových souborů, kde by krokování celého programu zabralo hodně času. Zarážky (breakpointy) se umísťují klepnutím kdekoli na řádek a stiskem klávesy F9. Klávesa F9 vkládá a maže zarážky. Také stisk pravého tlačítka myši a potvrzení nabídky Insert/Remove Breakpoint vloží a odstraní zarážku. Při ladění často potřebujeme přerušit program, je-li splněna nějaká podmínka. K tomu použijeme okno Breakpoints.

Nejjednodušší způsob, jak zjistit hodnotu proměnné během ladění, je ukázat na ni kurzorem. Pod kurzorem se objeví rámeček s hodnotou proměnné. Okno se sledovanými výrazy (Watch) se objevuje v ladícím režimu automaticky. Pokud ne, tak si ho můžete vyvolat přes menu View > Debug Windows > Watch nebo jednodušeji přes klávesou zkratku Alt + 3 nebo klikem na ikonu v panelu Debug. Standardně jsou okna Variables a Watch umístěna vedle sebe. V horní části okna je rozbalovací seznam Context, kde si můžete zvolit funkci, ve které chcete prohlížet proměnnou (v případech překrývajících se viditelnosti proměnných). Okno obsahující sloupce Name (jméno proměnné) a Value (hodnota proměnné) má tři záložky: Auto - zobrazuje proměnné a funkce vracející hodnoty, Locals - ukazují aktuální proměnné k funkci, this - používá se u objektových typů. Okno Watch umožňuje sledovat hodnoty složitějších výrazů (nejen proměnných, ale i výrazů obsahujících operátory a funkce). Editace se provádí stejně jako u okna Variables.



Úkol k řešení 3.1

Vyzkoušejte krokování programu podle přiložené animace. Vyzkoušejte krokování z menu i přes ikony nástrojové lišty v celém programu a také dovnitř a ven z funkcí.



Úkol k řešení 3.2

Vyzkoušejte použití zářezů a zářezů s podmínkou dle přiložené animace.



Úkol k řešení 3.3

Vyzkoušejte sledování hodnot obsahu proměnných, v okně Watch vytvořte své vlastní výrazy, krokujte program a sledujte změny hodnot.



Kontrolní otázka 3.1

K čemu je dobré krokování dovnitř funkcí?



Kontrolní otázka 3.2

Jakým nejjednodušším způsobem můžeme během krokování zjistit hodnotu proměnné?

4. Základy jazyka C



Čas ke studiu: 5 x 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- základy jazyka C a jeho elementy
- definovat základní datové typy
- deklarovat a definovat proměnné a konstanty
- operátory jazyka C, jejich prioritu zpracování a asociativitu
- psát správně různé typy výrazů
- používat řídicí struktury a cykly
- charakterizovat vlastnosti a funkci preprocesoru
- definovat a využívat vlastní funkce
- vysvětlit základy práce s ukazateli a poli
- používat řetězce a řetězcové funkce
- pracovat se soubory na úrovni vstupu a výstupu
- využívat argumenty příkazového řádku
- definovat výčtový, uživatelský typ a strukturu
- pracovat s matematickými funkcemi
- řešit celou řadu příkladů v jazyce C



Výklad

Jak už bylo uvedeno, tohle skriptum je z části věnováno hardwarovým prostředkům a z části jazyku C. Nejprve jsme se naučili základům práce v prostředí Visual Studia a nyní již můžeme přistoupit k samotnému jazyku. Je mu věnována tato kapitola, která svým rozsahem a probíraným obsahem patří mezi časově nejnáročnější části kurzu. V prezenční formě studia je jazyku C věnována polovina semestru v praktických cvičeních. Tato kapitola proto plánuje 15 intenzivních hodin studia teorie, procvičení animací, demonstračních příkladů i zadaných úloh. Avšak žádná kniha, nebo výukový materiál z vás za 15 hodin programátora neudělá, může vám však ukázat směr a pozitivně motivovat, to ostatní už záleží na vás.

4.1 Úvod do jazyka C

Jazyk C je fenomenálním jazykem autorů Briana W. Kernighana a Denise M. Ritchieho. Původně byl jazyk navržen v roce 1972 pro operační systém Unix, avšak po dalším vývoji byl implementován na všechny známé vývojové platformy. Jazyk C je tedy hardwarově a softwarově nezávislý. V roce 1983 byla definována norma jazyka ANSI C (zkratka *American National Standards Institute*). Tento standard byl finálně dokončen až v roce 1988. Následovala řada standardů ISO, které ANSI normu přebírají. Tyto normy označovány jako C 90 či C 99 jsou rozšířením původní definice jazyka C. Normu C 90 dnešní překladače plně podporují, avšak pro C 99 to zdaleka neplatí. Tohle výukové skriptum si nedává za úkol seznámit s konkrétními novinkami a různými vylepšeními standardu jazyka, ale naučit studenty základům jazyka tak, aby dokázali programovací jazyk vyžít a pochopili

formu tohoto mocného a univerzálního jazyka, který přes svou silnou využitelnost není mnoho obsáhlý.

V současné době se setkáváme s mnoha moderními jazyky, od C++, přes skriptovací jazyky pro web jako je *php*, univerzální *javu* až po verzi jazyka C# (*čtete si šarp*). Mějme na paměti, jak důležitým stavebním kamenem pro vývoj informačních technologií je jazyk C a vyjděme mu nyní vstříc.

V předchozí kapitole jsme pracovali s hotovým projektem, který tvořily tři soubory: *main.cpp*, *funkce.cpp* a *funkce.h*. Navíc byl využit systémový hlavičkový soubor *stdio.h*. Soubory se zdrojovým kódem s příponou *cpp* chápeme jako překladové jednotky. Ty jsou vstupem pro překladač společně se svými využívanými hlavičkovými soubory. Překladač tyto soubory kompiluje a výsledkem překladu je objektový soubor pro každou překladovou jednotku. Objektový soubor má příponu *obj* a tuto vlastnost si nepleťme s pojmem objektově orientovaného programování. Objektový kód je pak spojovacím programem *linkován* společně s knihovními funkcemi jazyka C (*přípona lib*) a v případě že je vše v pořádku a bez chyb, je vytvořen výsledný spustitelný *exe* soubor. Častou chybou, která je ve fázi linkování odhalena, je právě deklarace proměnné ve více zdrojových modulech.



Obr. 4.1 Postup programu od vytvoření zdrojového kódu až po vytvoření exe souboru

Na úvodních dvou cvičeních jsme pracovali s prostředím visual studia a jednotlivých ukázkách zdrojového programu v jazyce C, ale žádný z nich jsme nevytvářeli. V první kapitole většiny programátorské literatury je nepsaným pravidlem uvádět první prográmek v konkrétním programovacím jazyce, kterému je kniha věnována. První kratičký prográmek se často jmenuje *hello word* (*ahoj světe*). Nyní je nejvyšší čas si představit obdobu takového programu, spustit vývojové prostředí, program napsat, zkompileovat a spustit.



Příklad – první program v jazyce C

```

/*****/
/* tak tohle je první program v jazyce C */
/*****/
#include <stdio.h>
int main(void)
{
    printf("Dobry den, vitame vas na úvodním cviceni k jazyku C.\n");
    return 0;
} /* main */
  
```

První tři řádky představuje komentář. Program začíná vložením hlavičkového souboru *stdio.h* (zkratka **standard input output**) a informuje překladač o skutečnosti, že se budou používat již vytvořené knihovní funkce (jsou součástí jazyka C) vstupu a výstupu. Každý program v jazyce C musí obsahovat hlavní funkci *main*. Protože funkce je bez parametru, uvádíme do závorek prázdný typ *void*, nebo můžeme uvést prázdné kulaté závorky. Následuje tělo programu, jehož součástí je funkce, která nám napíše uvítací hlášení na obrazovku. Funkce *main* je typu *integer*, proto musí vrátet celé číslo. Návrátová hodnota funkce *main* říká operačnímu systému, jakým výsledkem funkce skončila, zdali úspěchem – nula, nebo neúspěchem – jiná hodnota představující číslo chyby.

4.2 Základy jazyka

Program v jazyce C se vždy skládá z elementárních součástí, které rozdělujeme na **klíčová slova, identifikátory, konstanty, operátory a interpunkční znaky**.

Pokud potřebujeme pracovat s proměnnými, konstantami a funkcemi, musíme je pojmenovat. Typům jmen v jazyce C se říká identifikátor. Pro identifikátory platí určitá pravidla, prvním symbolem může být písmeno nebo podtržítka a následuje libovolná kombinace písmen, číslic a podtržitek, nejvýše však do povolené délky identifikátoru. Norma ANSI C tuto hodnotu omezuje na 31.



Pojem k zapamatování: Jazyk C je Case-sensitive

Jazyk C je tzv. Case-sensitive, to znamená, že rozlišuje malá a velká písmena v názvech identifikátorů, proměnných, ale i klíčových slov. Proto proměnná *Jmeno* je odlišná od proměnné *JMENO*. Stejně tak malá písmena v názvech klíčových slov se musí dodržet neboť klíčové slovo *if* je v pořádku, zatímco *IF* nebo *If* není chápáno jako klíčové slovo nýbrž jako identifikátor.

Klíčová slova mají speciální význam pro překladač C. Žádný identifikátor nemůže mít ve fázi překladu stejný název jako klíčové slovo. ANSI norma jazyka C určuje následující klíčová slova:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Tab 4.1 Klíčová slova v jazyce C

Ve zdrojovém kódu často používáme komentáře. Komentář je kompilátorem považován za oddělovací znaky a vypouští je. Komentář začíná zápisem dvojice znaků lomítka-hvězdička. Poté následuje vlastní komentovaný text dlouhý několik slov nebo dokonce několik řádků. Komentář je pak ukončen opačnou sekvencí znaků hvězdička-lomítka.



Příklad použití komentáře

```
/* povolene umístění komentare */
obvod = 2 * PI /* PI je 3.14 */ * r /* r je polomer*/
/* nepovolene umístění komentare */
ob /* obvod kruhu */vod = 2 * PI * r
```

Komentář se může nacházet i uprostřed výrazu, nikoli však uprostřed identifikátoru. Rovněž je třeba zdůraznit, že ANSI norma jazyka nedovoluje vnořené komentáře. Programátoři komentáře často využívají, neboť nějaká vymyšlená programová konstrukce je v danou chvíli srozumitelná, avšak za určitý čas může být hůře pochopitelná. Rovněž pokud na jednom projektu pracuje více lidí, je věci slušnosti komentovat pro dobrou čitelnost své programy.

Pro popis dalších elementů jazyka jako konstanty a operátory zvolíme následující odstavce, neboť nejprve potřebujeme definovat základní datové typy.

4.3 Základní typy dat

Základní typy dat dělíme na aritmetické datové typy, znaky a ukazatele. Aritmetické dále dělíme na celočíselné, které mohou být znaménkové nebo bez znaménka (*signed*, *unsigned*) a reálné. Pro znaky má jazyk C datový typ *char*, který reprezentuje znak z konkrétní znakové sady a je mu vyhrazen jeden byte. Ukazatel je speciální datový typ a představuje adresu paměťového místa. Ukazatelům v této chvíli nebudeme věnovat pozornost, protože jsme na samotném začátku kurzu jazyka C. Přehled základních celočíselných datových typů vidíme v tabulce 4.2.

Typ	Velikost v bitech	Význam	Rozsah
unsigned char	8	znak	0 až 255
char, signed char	8	znak	-128 až 127
unsigned short	16	krátké celé číslo	0 až 65535
signed short, short	16	krátké celé číslo	-32768 až 32767
unsigned int	16 nebo 32	celé číslo	0 až 4294967295
signed int	16 nebo 32	celé číslo	-2147483648 až 2147483647
unsigned long	32	dlouhé celé číslo	0 až 4294967295
signed long, long	32	dlouhé celé číslo	-2147483648 až 2147483647

Tab. 4.2 Celočíselné datové typy

Z tabulky je patrné, že jsme u celočíselných datových typů *short int* a *long int* vypustili klíčové slovo *int*, což si můžeme dovolit, kompilátor pochopí, že se jedná o zkrácenou deklaraci. Rovněž použití klíčového slova *signed* je nadbytečné, protože kompilátor považuje všechna čísla automaticky za znaménková, proto můžeme při deklaraci slovo *signed* vynechat. U datového typu *int* máme v tabulce dvě hodnoty velikosti v bytech. Tento typ je implementačně závislý na verzi překladače v prostředí operačního systému. Na 32 bitových platformách je jeho velikost 32 bitů.



Korespondenční úkol – Rozsahy datových typů

V hlavičkovém souboru *limits.h* máme definovány maxima a minima pro naimplementované datové typy. Vyhledejte tento hlavičkový soubor v adresáři *Include* vývojového prostředí, otevřete jej a zkontrolujte rozsahy jednotlivých datových typů.

Pro reálná čísla máme v ANSI C normě jazyka k dispozici tři datové typy lišící se opět svým rozsahem a velikostí v bytech. Reálná čísla jsou reprezentována ve tvaru, který obsahuje mantisu a desítkový exponent, maximální velikost uloženého čísla je pak dána počtem bitů reprezentující exponent, přesnost je určena počtem bitů mantisy. Reprezentace reálných čísel se řídí definovaným standardem IEEE754. Podobně jako v souboru *limits.h* má jazyk C definovány parametry reálných datových typů v souboru *float.h*.

Je třeba se zmínit ještě o typu *void*. Setkáme se s ním v případě, že funkce nemá návratovou hodnotu, v tom případě bude mít před svým jménem uveden typ *void*. Druhým případem bude *void* typem parametru funkce, v tomto případě to bude znamenat, že funkce nemá žádné formální parametry.

float	double	long double
Velikost v bitech		
32	64	80
Mantisa v bitech		
24	53	64
Exponent v bitech		
7	10	15
Počet desetinných míst		
6	15	18
Maximální exponent		
38	308	4932
Maximální číslo		
3.402823466e+38	1.7976931348623158e+308	1.189731495357231765e+4932
Minimální číslo		
1.175494351e-38	2.2250738585072014e-308	3.3621031431120935063e-4932

Tab. 4.3 Reálné datové typy

4.4 Konstanty a proměnné

Po uvedení základních datových typů již můžeme přistoupit k proměnným a konstantám. S konstantami a proměnnými jsou úzce spojeny dva pojmy: *deklarace* a *definice*. V jazyce C totiž musíme před prvním použitím pojmenovat proměnnou a určit její datový typ. To v některých jazycích není nutné, v C však ano. Této nutné skutečnosti říkáme *deklarace*. Deklarací tedy určujeme typ objektu nezbytný pro překladač jazyka C. S pojmem *deklarace* souvisí pojem *definice*. Definicí určujeme hodnotu proměnné či těla funkce. Definici můžeme spojit s deklarací, můžeme ji však udělat i později.



Příklad deklarace a definice

```
int cislo; /* deklarace prom. cislo */
int krok = 2; /* deklarace a definice prom. krok */
int vypocet( int a); /* deklarace funkce vypocet */
int vypocet( int a) /* definice funkce vypocet */
{
    ... příkazy funkce
}
```

Při deklaraci proměnné je založen identifikátor pro tuto proměnnou deklarovaného typu, není však ještě vyhrazena paměť. Konkrétní vymezení paměťového prostoru pro uložení proměnné je provedeno až při definici této proměnné.

Konstanty jsou čísla, znaky nebo řetězce, které během programu nemění svou hodnotu. Z technických věd a matematiky známe celou řadu konstant a máme tedy prostor si je zadefinovat a používat jejich neměnnou hodnotu v průběhu celého programu. Konstanty mohou nabývat hodnot jak základních datových typů, tak typů uživatelsky definovaných, tak i pole. Konstanty definujeme klíčovým slovem **const**, následuje datový typ, identifikátor, rovnítko a její hodnota. Definici samozřejmě ukončujeme středníkem.

Celočíselné konstanty jsou tvořeny zápisem celého čísla. Mohou být zapsány v desítkové, osmičkové případně v šestnáctkové číselné soustavě. Desítkový zápis používá pouze číslice 0 až 9, přičemž 0 nesmí být použita na začátku, neboť by byla konstanta chápána jako osmičková (*oktalová*). Oktalové konstanty mohou využívat pouze číslic 0-7 a musí začínat nulou. Za číslem 07 tedy následuje 010, 011

atd. Hexadecimální konstanty používají číslice 0 až 9 následovány písmeny A až F. Na počátku se uvádí dvojice znaků 0x.

Následuje zápis čísla 255 v těchto číselných soustavách: **255** **0377** **0xff**.



Příklad definice konstant

```
const int kilo = 1000;
const double Pi = 3.141592654;
const char becko = 'b';
const float policko[2] = {2,2};
```

Reálné konstanty umožňují zapsat číselnou konstantu, která nemusí být celočíselná. Vnitřně je reprezentována ve tvaru, který obsahuje mantisu a exponent, obojí příp. se znaménkem. Není-li u konstanty uvedena žádná přípona, je konstanta chápána jako typ *double* (např. 56.45). Pokud uvedeme příponu *f* nebo *F*, je konstanta typu *float* (56.45F). Obdobně přípona *l* nebo *L* určuje typ *long double* (56.45L).

Znakové konstanty jsou tvořeny příslušným znakem mezi apostrofy: 'a' 'A' 'z' 'Z'. Pozor neplést apostrof ‘ s uvozovkami “.

Některé znaky pokud chceme zapsat, musíme použít jejich náhradního zápisu, tzv. *Escape sekvence*. Pravděpodobně víte, že pokud chcete použít v jazyce HTML znaky <>, které zdrojový kód využívá k označení názvu klíčového tagu, musíte použít náhradní význam < > jímž říkáme escape sekvence. V následující tabulce máme escape sekvence některých znakových konstant a také netisknutelných znaků.

Escape sekvence	Jméno	Význam
\a	Alert	Pípnutí
\\	BackSlash	Obrácené lomítko
\b	BackSpace	Návrat o jeden znak
\?	Qustion mark	Otazník
\f	Formfeed	Nová stránka
\‘	Single quote	Apostrof
\n	Newline	Přechod na nový řádek
\“	Double quote	Uvozovky
\r	Carriage return	Přesun na začátek aktuálního řádku
\ooo	Octal	ASCII znak v oktalové notaci
\t	Horizontal tab	Tabulátor
\xhh	Hexa	ASCII znak v hexadecimální notaci
\v	Vertical tab.	Vertikální tabulátor

Tab. 4.4 Escape sekvence

Řetězcové konstanty jsou pevnou sekvencí znaků umístěných v uvozovkách (“tohle je příklad řetězcové konstanty”). Je chápána jako pole znaků a je vždy ukončen speciálním ukončovacím znakem ‘\0’, který se nesmí vyskytovat uvnitř textu.



Pojem k zapamatování: Řetězcová konstanta

Řetězcová konstanta není ničím jiným v jazyce C než polem znaků, její požadovaná paměť je vždy o jeden bajt větší než počet znaků mezi uvozovkami. Tento bajt je právě vyhrazen pro ukončovací znak ‘\0’.

Proměnné jsou paměťová místa přístupná prostřednictvím identifikátoru. Od konstanty se liší tím, že jejich hodnotu můžeme během programu měnit. Proměnné deklarujeme uvedením datového typu, jež je následován identifikátorem a ukončen středníkem. Jak už bylo uvedeno výše, současně s deklarací proměnné můžeme, ale nemusíme, definovat i její počáteční hodnotu. Pokud deklarujeme či definujeme více proměnných stejného typu, můžeme na jednom řádku uvést seznam identifikátorů navzájem oddělených čárkami.



Příklad deklarace a definice více proměnných stejného typu

```
float a,b,c; /* deklarace třech proměnných stejného typu */
int deka, kilo= 1000; /* deklarace a zároveň definice proměnných stejného typu */
```

4.5 Operand, operátor, výraz

Pro matematické, logické a další výpočetní operace definujeme pojmy operand a operátor. Z matematiky známe pojem výraz, např. $m + n$ je aritmetickým výrazem s operátorem sčítání a dvěma operandy m a n . Operátory můžeme rozdělit podle počtu operandů se kterými mohou pracovat. Této vlastnosti se říká *arita*. Operátory takto rozdělíme na **unární**, které pracují s jedním operandem, **binární** pracující se dvěma operandy a specialitou jazyka C je podmíněný **ternární** operátor pracující se třemi operandy.

Důležitým faktem je umístění operátorů vzhledem k operandům. V jazyce C se setkáváme se všemi třemi způsoby použití, prefixový, infixový a postfixový. Z toho vyplývá umístění operátoru před, mezi nebo za operandy. Způsob zápisu ovlivňuje chování operandů, podrobněji tyto případy probereme v řešených příkladech a přiložené animaci.

Unární operátory jsou prefixové, s malou výjimkou a tou je možné prefixové i postfixové použití operátorů inkrementace a dekrementace. Binární operátory jsou infixové, což je nejběžnější způsob zápisu známý z matematiky. Binární operátory jsou **aritmetické, relační a logické**. Protože jejich použití je nám bližší představíme nejprve binární operátory, až po nich operátory unární.

Aritmetické operátory jsou nejznámější, setkáváme se s nimi při každodenní práci. Operátory sčítání, odčítání, násobení a dělení si představovat nemusíme, v příslušné animaci řešeného příkladu si však ukážeme, jakým způsobem si musíme ohlídat datové typy výsledku a na co si musíme dát pozor. V potaz musíme vzít taky prioritu operátorů, která bude uvedena při dalším výkladu. Dalším aritmetickým operátorem je operátor modulo, neboli zbytek po celočíselném dělení.



Korespondenční úkol – operace modulo

Jaké výsledky dostaneme u konkrétních příkladů použití operátoru modulo? $a = 5 \% 2$;
 $b = 2 \% 3$; $c = 6 \% 3$; $d = 6 \% 9$;

Relační operátory slouží k porovnání operátorů a výsledkem výrazu je buďto pravda nebo nepravda uvedeného výrazu.

Logické operátory máme k dispozici v jazyce C pouze tři, jsou jimi logický součet *OR*, logický součin *AND* a logickou negaci *NOT*. Pro jejich zápis se používají znaky `||`, `&&` a `!`. Pravidla pro určení výsledku bychom měli znát z Booleovy algebry.



Korespondenční úkol – Logické operátory

Vytvořte pravdivostní tabulku logických operátorů **AND** a **OR** pro dvě proměnné x , y .

Unární operátory		Binární operátory		Binární operátory	
+ -	unární plus a mínus	=	přiřazení	.	přístup ke členu
&	získání adresy objektu	+ - * /	plus, mínus, krát děleno	->	přístup ke členu
*	získání objektu dle adresy	%	modulo	,	oddělení výrazů
!	logická negace	<< >>	bitový posun	< >	menší(větší) než
~	bitová negace	&	bitový součin	<= >=	menší(větší) nebo rovno
++	inkrementace		bitový součet	==	rovnost
--	dekrementace	^	bitový xor	!=	nerovnost
(typ)	přetypování	&&	logický součin	Ternární operátor	
sizeof	získání délky objektu		logický součet	? :	podmíněný op.

Tab. 4.4 Rozdělení operátorů

Bitové operátory jsou třetí skupinou binárních operátorů. První skupina bitových operátorů je opět logického charakteru, avšak operandy jsou nyní čísla rozložená na bity. Na sobě odpovídajících bitech se provádějí logické operace. Bitové operátory jsou definované pouze na celých číslech. Kromě bitového součtu, součinu a negace máme ještě k dispozici operátor XOR, kterému se říká operátor neekvivalence. Výsledek operace je tedy pravda, pokud mají operandy odlišné hodnoty.



Korespondenční úkol – Operátor neekvivalence

Vytvořte pravdivostní tabulku operátoru **XOR** pro dvě proměnné x, y .

Druhou skupinu zastoupenou dvěma operátory jsou operátory bitového posuvu. Celé číslo vyjádřeno ve dvojkové soustavě tak můžeme po jednotlivých bitech posouvat doleva nebo doprava. Bity, které se však posunou mimo rozsah čísla, jsou doplněny z opačné strany nulami.



Příklad bitového posunu

10110111 číslo, které budeme bitově posouvat

11011100 číslo po operaci bitového posuvu o 2 bity doleva

01101110 po dalším posuvu, tentokrát o jeden bit doprava

$m \ll 2$; /* bitový posuv proměnné m o dva bity vlevo */

$m \gg n$; /* bitový posuv proměnné m o n bitů vpravo */

Operátor přiřazení přiřazuje proměnné konkrétní hodnotu, lépe řečeno hodnotu výrazu. Operátor přiřazení má v jazyce C široké možnosti. Jeho obecný zápis má tvar:

$v1\ op = v2$,

kde operátor může být složen kombinací rovnítka a jednoho z následujících operátorů: + - * / % ^ | << >>. Nejjednodušším použitím operátoru přiřazení je *proměnná = hodnota*. Použití složeného operátoru $v1\ op = v2$ můžeme využít ve tvarech např. $i = i + j$, který můžeme napsat ve zkráceném tvaru $i += j$.



Příklad složeného operátoru přiřazení

$m * = i + j$; /* tento tvar zápisu je obdobou $m = m * (i + j)$, nikoli však $m * i + j$! */

Vraťme se zpět k unárním operátorům, které jsme zatím jen uvedli v tabulce. **Operátory inkrementace a dekrementace** jsou častou součástí programu a jejich použití oceníme v situacích, kdy je třeba v programu snížit či zvýšit hodnotu o jedničku. To lze samozřejmě i tak, že použijeme aritmetický operátor + nebo - ($a = a + 1$ nebo $a -= 1$). V jazyce C však máme k dispozici operátor inkrementace a dekrementace zvyšující, resp. snižující proměnnou o jedničku ($a++$ resp. $a--$). Výhodou je zde zkrácení zápisu, což je v jazyce C často využíváno. Operátor inkrementace a dekrementace můžeme použít v prefixovém i postfixovém zápisu. Oba zápisy mají stejný význam, avšak postfixová verze provede operaci až po použití výrazu, zatímco prefixová před použitím výrazu. Tento rozdíl nejlépe demonstruje příklad, který je uveden v animaci č. 6.

Operátor **sizeof()** se využívá k určení velikosti paměti, kterou libovolná proměnná zabírá. Operátor můžeme také použít pro zjištění velikosti datového typu. Přestože by se mohlo zdát, že jde o knihovní funkci, není tomu tak, *sizeof()* je operátor.

Čárka je operátorem postupného vyhodnocení. Má nejnižší prioritu ze všech operátorů a vyhodnocuje se zleva doprava. Čárkou můžeme oddělit jednotlivé výrazy v místě, kde je očekáván jediný výraz. Typické použití najdeme v cyklu *for*.

Operátor **přetypování výrazu** často používáme v souvislosti s operacemi, kdy bychom mohli ztratit přesnost výsledku, nebo jej obdržet dokonce zcela nesprávný. To se může např. stát při podílu dvou celočíselných čísel, jejichž výsledek je implicitně celočíselného typu, i když očekáváme typ reálný. Takže výraz $f = i1 / i2$, který obsahuje podíl dvou celočíselných hodnot přiřazený do proměnné reálného typu nejprve provede celočíselné dělení a celočíselný výsledek se ztrátou přesnosti poté uloží do reálné proměnné. Přetypování provádíme tak, že před hodnotu, kterou chceme přetypovat, napíšeme typ, který chceme získat, v kulatých závorkách. Syntakticky tedy zapíšeme přetypování takto:

(type) hodnota



Příklad přetypování

```
f = float(i1) / i2; /* nyní již proběhne dělení v plovoucí čárce */
/* hodnota i1 / i2 tedy bude reálná a nedojde ke ztrátě desetinné části */
```

Adresový operátor umožňuje získat adresu objektu, na nějž je použit. Adresu objektu můžeme použít v nejrůznějších situacích, nejčastěji ale v souvislosti s ukazateli. Bez adresového operátoru nemůžeme pracovat se soubory a v omezeném režimu se standardním vstupem. Použití adresového operátoru si samozřejmě ukážeme v dalších kapitolách.

Podmíněný operátor je ternárním operátorem, to znamená, že pracuje se třemi operandy. Je to specialita jazyka C a jazyků z něj odvozených. Jeho obecný tvar je

(podmínka) ? v1 : v2

Na základě výsledku podmínky která je buď pravda nebo nepravda je vyhodnocen výraz v1 resp. v2.



Příklad podmíněného operátoru

```
max = (a > b) ? a : b; /* do proměnné max se uloží větší hodnota */
/* pokud je podmínka splněna do proměnné max se uloží hodnota a (vyhodnotí se
výraz před dvojtečkou) pokud splněna není uloží se hodnota b (výraz za dvojtečkou)
*/
```

Pro správné fungování aplikací při provádění výpočtů a vyhodnocování výrazu je nutné znát prioritu a asociativitu operátorů. Priorita určuje, které operace jsou ve složených výrazech vyhodnocovány dříve než ostatní. Asociativita určuje zpracování operandů pro jednotlivé operátory. V následující tabulce máme přehledně určenu prioritu a asociativitu operátorů, přičemž operátory na jednom řádku mají prioritu stejnou.

Operátor	Typ	Asociativita
[] () . -> postfixové ++ postfixové --	unární operátory	zleva doprava
prefixové ++ prefixové -- sizeof & * + - ~ !	unární operátory	zprava doleva
(typ)	přetypování	zprava doleva
* / %	násobení dělení modulo	zleva doprava
+ -	sčítání a odčítání	zleva doprava
<< >>	operátor bitového posuvu	zleva doprava
< > <= >=	relační operátor	zleva doprava
== !=	operátor rovnosti a nerovnosti	zleva doprava
&	bitový AND	zleva doprava
^	bitový XOR	zleva doprava
	bitový OR	zleva doprava
&&	logický AND	zleva doprava
	logický OR	zleva doprava
?:	podmíněné vyhodnocení	zprava doleva
= *= /= %= += -= <<= >>= &= = ^=	jednoduché a složené přiřazení	zprava doleva
,	postupné vyhodnocování	zleva doprava

Tab. 4.4 Priorita a asociativita operátorů v jazyce C



Ke kapitolám 4.1 až 4.4 je přiřazena demonstrační animace č. 6

Animace č. 6 je věnována základům jazyka C jako deklarace proměnných a použití různých operátorů s důrazem na zajímavé vlastnosti.

4.6 Funkce standardního vstupu a výstupu

Standardním vstupem a výstupem chápeme klávesnici a monitor počítače. V závěrečné kapitole věnované jazyku C se budeme věnovat práci se soubory. V této kapitole si probereme jednoduché funkce pro vstup a výstup a také funkce pro formátovaný vstup a výstup. Konkrétní ukázka bude opět předvedena v animaci č. 7. Funkce `putchar` zajišťuje výstup jednoho znaku na standardním výstupu. Deklarace funkce je následující:

`int putchar(int znak);`

Funkce je typu *integer* a také jediným parametrem je znak, který bude vypsán.

```
#include <stdio.h>

void main()
{
    putchar ('p');
    putchar ('r');
    putchar ('i');
    putchar ('n');
    putchar ('t');
    putchar (' ');
    putchar ('t');
    putchar ('e');
    putchar ('x');
    putchar ('t');
    putchar ('\n');
}
```

→ \n znamená konec řádku

Obr. 4.2 Funkce `putchar`

Přesto, že se jedná o znak, není parametrem funkce typ *char* nýbrž typ *integer*. Návrátová hodnota je tedy hodnota vypsaného znaku, pokud vše proběhne v pořádku, v opačném případě je návratovou hodnotou *EOF*.

Naopak funkce *getchar* slouží pro načtení jednoho znaku ze standardního vstupu (klávesnice), který vrátí jako svou návratovou hodnotu. Návrátová hodnota funkce tedy reprezentuje jeden znak vracející číslo typu *integer*. Funkce má následující deklaraci:

int getchar(void);

Návratovou hodnotu funkce obvykle přiřazujeme nějaké proměnné, se kterou dále pracujeme. V případě chyby či nalezení konce souboru vrací funkce hodnotu *EOF*.

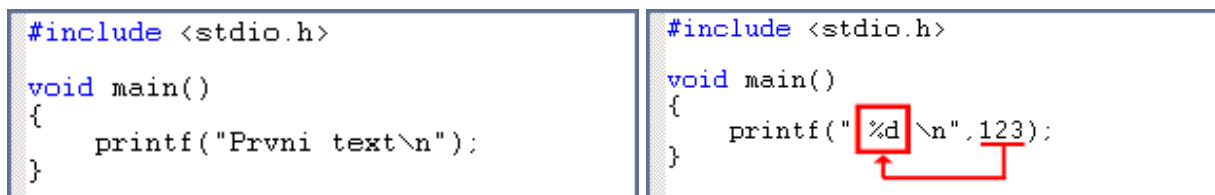
Pro práci se standardním vstupem a výstupem můžeme použít dvě obdobné funkce pro načtení a zobrazení celého řádku. Jedná se o funkce *gets* a *puts* s následující deklarací:

int puts(const char *radek);

char *gets(char *radek);

Funkce *puts* vypíše řádek textu na obrazovku a přidá ukončení tohoto řádku. V případě chyby je vrácena hodnota *EOF*, když operace proběhne v pořádku, je vrácena nezáporná hodnota. Funkce *gets* načte znaky ze standardního vstupu až do znaku */n*, což je přechod na nový řádek. Hvězdička v deklaraci znamená, že návratovou hodnotou funkce je ukazatel na načtený řetězec, jež je polem znaků, jehož délku předem neznáme. Použití funkcí opět procvičíme v příložené animaci.

Pro formátovaný vstup a výstup používáme funkce *printf* a *scanf*. S těmito funkcemi jsme se setkali ještě před samotným výkladem jazyka C, nyní se je tedy detailně popíšeme, nejprve funkci **printf**. Slouží nám pro formátovaný výstup, tzn. tiskne znaky na obrazovku. Formátovací řetězec může obsahovat běžné znaky na vytisknutí nebo speciální formátovací sekvenci začínající znakem *%*.



```
#include <stdio.h>

void main()
{
    printf("Prvni text\\n");
}
```

```
#include <stdio.h>

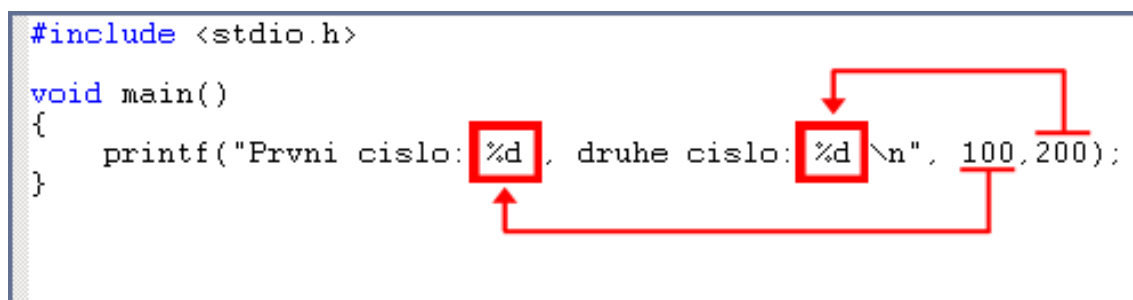
void main()
{
    printf("%d\\n", 123);
}
```

Obr. 4.3 Funkce *printf*

Obecný zápis formátovací sekvence funkce *printf*:

% [příznak] [šířka] [.přesnost] [F | N | h | l | L] typ

Části sekvence uvedené v hranatých závorkách jsou nepovinné. Položka *typ* je povinná a udává nám jaký typ argumentu bude funkce *printf* tisknout. Pokud chcete napsat znak *%*, tak musíte použít zdvojení znaku *%* (např. `printf("Hodnota je v procentech %%\\n");`).



```
#include <stdio.h>

void main()
{
    printf("Prvni cislo: %d, druhe cislo: %d\\n", 100, 200);
}
```

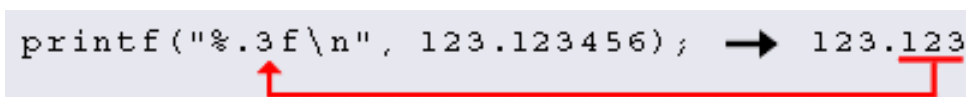
Obr. 4.4 Tisk dvou čísel pomocí funkce *printf*

Příznak určuje zarovnání výstupu, zobrazení znaménka a desetinných míst u čísel, úvodní nuly, prefix pro osmičkový a šestnáctkový výstup. Šířka určuje minimální počet vytištěných znaků na výstupu, čísla pak mohou být doplňovaná nulami či mezerami. Za šířkou může následovat přesnost, vždy ji předchází tečka v zápise. Přesnost určuje maximální počet znaků na výstupu, pro celá čísla minimum zobrazených znaků, pro racionální počet desetinných míst. Dále mohou následovat znaky které indikují dlouhé celé číslo – *l* nebo long double – *L*, znaky *F*, *N*, *h* měly své použití v systému MS DOS. Následuje typ. V tabulce můžete vidět různé formáty pro vstupní hodnoty a jejich zápis:

typ	význam	příklad → výstup
d, i	decimální číslo (normální)	<code>printf("%d", 123);</code> → 123
o	osmičkové číslo	<code>printf("%o", 123);</code> → ???
u	číslo bez znaménka	<code>printf("%u", 123);</code> → 123 <code>printf("%u", -123);</code> → ???
x, X	šestnáctkové (hexadecimální)	<code>printf("%x", 123);</code> → 7b???
f	desetinné číslo	<code>printf("%f", 100.123);</code> → 100.123000
e, E	číslo v exponenciálním tvaru	<code>printf("%e", 100.123);</code> → 1.001230+e002
g, G	automaticky	<code>printf("%g,%g", 100.123, 1230000000000.0);</code> → 100.123, 1.23e+009
c	znak	<code>printf("%c", 'A');</code> → A
s	řetězec	<code>printf("%s", "AAAA");</code> → AAAA

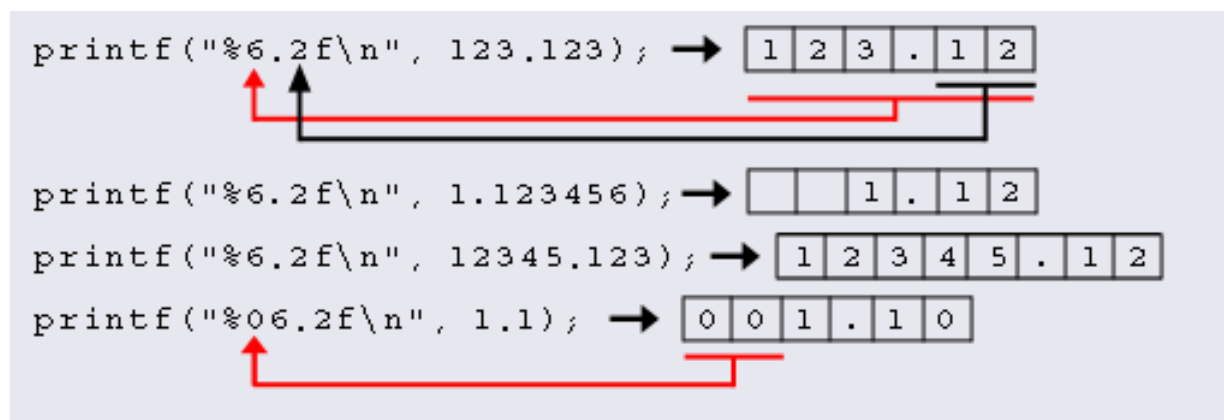
Tab. 4.5 Typy argumentu funkce *printf*

Ještě se krátce vraťme k šířce a přesnosti zápisu. V uvedené animaci si funkce pro formátovací vstup a výstup procvičíme, za krátko si na uvedený formátovací tvar zvykne a brzy ho budeme ovládat. Nyní alespoň krátká ukázka.



```
printf("%.3f\n", 123.123456); → 123.123
```

Obr. 4.5 Tisk čísla s pevným počtem desetinných míst



```
printf("%6.2f\n", 123.123); → 123.12
```

```
printf("%6.2f\n", 1.123456); → 1.12
```

```
printf("%6.2f\n", 12345.123); → 12345.12
```

```
printf("%06.2f\n", 1.1); → 001.10
```

Obr. 4.6 Tisk čísla s desetinou čárkou

Pro formátovaný vstup máme k dispozici funkci **scanf**. Formátovací sekvence má následující tvar:

% [*] [šířka] [F | N] [h | l | L] typ

Formátovací sekvence začíná znakem procento, následuje řada volitelných nepovinných položek a jedné povinné, stejně jako u funkce *printf*. Hvězdička znamená přeskočení dané položky vstupu, Hodnota je ze vstupu načtena, není však zapsána do paměti. Šířka určuje maximální počet znaků načtených ze vstupu, ostatní hodnoty F, N, h, l, L a typ je obdobný jako u funkce *printf* uvedené výše.



Pojem k zapamatování: Použití operátoru získání adresy &

Funkce načítá znaky z klávesnice a vstup hodnoty je uložen do adresy. Proto musíme při použití funkce pamatovat na operátor získání adresy **&**. Obecný zápis formátovacího sekvence funkce *scanf*:

```
#include <stdio.h>

void main()
{
    char znak;
    scanf("%c", &znak);
    printf("Načtený znak %c\n", znak);
}
```

```
#include <stdio.h>

void main()
{
    int cislo;
    scanf("%d", &cislo);
    printf("Načtené číslo %d\n", cislo);
}
```

Obr. 4.7 Příklad načtení znaku a čísla

```
#include <stdio.h>

void main()
{
    char slovo[20]; //slovo max. o 19-ti písmenech
    scanf("%s", &slovo);
    printf("Načtené slovo %s\n", slovo);
}
```

Obr. 4.8 Příklad načtení řetězce jako konstantního pole znaků



Ke kapitole 4.6 je přiřazena demonstrační animace č. 7

Animace č. 7 je věnována funkcím *printf*, *scanf* a jejich formátovacím konstrukcím.

4.7 Řídicí struktury

Pokud chceme psát v jazyce C projekty, které řeší konkrétní úlohy z praxe, nevyhneme se nějakým řídicím strukturám. Z běžného života víme, že na různé situace reagujeme podle různých podmínek, které vyhodnocujeme. Některé úkony provádíme opakovaně, v některých příkladech se rozhodujeme podle více možností. Obdobně je to i při řešení nějaké úlohy, kterou řešíme pomocí jazyka C. V tomto jazyce máme pro řídicí struktury možnosti jako běžné příkazy, bloky složených příkladů, příkazy řízené podle nějaké podmínky, příkazy řízené podle více možností, cyklické provádění příkazu a také skoky. Pojďme si vyjmenované konstrukce popsat pěkně popořádku.

Příkazy a bloky

V předchozích kapitolách jsme podrobně popisovali výrazy, které můžeme tvořit pomocí operátorů a operandů. Pokud za takový, byť jednoduchý výraz, jako například použití inkrementace proměnné $a++$, napíšeme středník, stane se z něj příkaz. Příkazy mohou být libovolně složité od jednoduchých výrazů až po složité kombinace použití výrazů a funkcí. Dokonce i prázdný příkaz bez výrazové části je příkazem!



Pojem k zapamatování: Prázdný příkaz

Prázdný příkaz neobsahuje výrazovou část. Jedná se tedy o samostatný středník, který se stává prázdným příkazem. Nevhodným použitím středníku tam, kde nepatří (většinou u podmínek a cyklů), můžeme často získat nevhodné chování programu, které je však z hlediska syntaxe v pořádku a při kompilaci programu chybu nezjistíme.

Jednotlivé příkazy můžeme slučovat do bloků. Blok obsahuje sekvenci příkazů a je umístěn ve složených závorkách `{}`. Na počátku bloku můžeme také deklarovat proměnné, které mají rozsah platnosti omezen na tento blok, popřípadě vnořené bloky. Tělo každé funkce je ve své podstatě blokem. Proto také můžeme deklarovat lokální proměnné ve funkci *main*, které mají rozsah platnosti právě v jejím bloku.

Podmíněný příkaz

Prvním řídicím příkazem je příkaz *if* nebo jeho konstrukce *if – else*. S podmíněným vyhodnocením jsme se již v jazyce setkali, a to u operátoru podmíněného výrazu `?:`, ten však používáme pro vyhodnocení části výrazu. Pro výběr z příkazů máme v jazyce C k dispozici podmíněný příkaz.

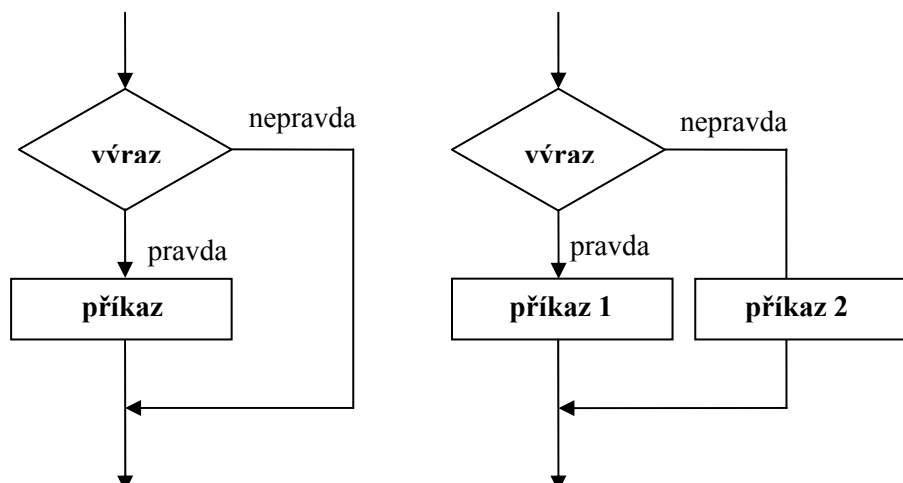
Jeho syntaktický zápis je následující:

if (výraz) příkaz;

nebo:

**if (výraz) příkaz1;
else příkaz 2;**

Výraz pro vyhodnocení musí být v kulatých závorkách. V případě jeho pravdivé hodnoty se provede *příkaz*. Po jeho provedení pokračuje program za tímto příkazem. V případě nepravdivého vyhodnocení výrazu se *příkaz* neprovede a řízení programu předá bezprostředně za podmíněný příkaz. Nesmíme zapomenout na ukončení příkazu středníkem. Nutno připomenout, že nepravdivé vyhodnocení je představováno nulovou návratovou hodnotou výrazu, zatímco pravdivá hodnota je reprezentována celočíselnou nenulovou hodnotou. Tenhle fakt mějme na paměti samozřejmě i u ostatních typů cyklů.



Obr. 4.9 Podmíněný příkaz If a If-else

Podmíněný příkaz *if-else* můžeme použít pro realizaci vícenásobného větvení. Takové konstrukci se říká *if – else – if*. Při vyhodnocování podmínky vnímáme skutečnost, že chceme provést jeden z konkrétních příkazů realizovaný v dané větvi rozhodování. Testovat můžeme také různé výrazy. Syntaxe je následující:

```
if ( výraz1 ) příkaz1;
else if ( výraz2 ) příkaz2;
else if ( výraz3 ) příkaz3;
...
else if ( výraz N ) příkaz N;
else příkaz N+1;
```

Pokud není poslední příkaz **N+1** bez podmínky uveden, nemusí se dokonce provést žádný příkaz neboť testovaná podmínka není splněna v žádné větvi konstrukce.

Přepínač

Přepínačem rozumíme větvení programu do několika větví, obdobně jako konstrukce *if – else – if*. Přepínač je realizován klíčovým slovem *switch* a slouží k rozdělení posloupnosti příkazů na části a následné vybrání a provedení některé varianty, či více variant. Musíme si však uvědomit, že přepínač slouží k větvení výpočtu podle hodnoty celočíselného výrazu.

Syntakticky příkaz přepínače zapíšeme do následujícího tvaru:

```
switch ( celočíselný výraz ) příkaz
kde příkaz je tvořen blokem tvořen návěštími ve tvaru:
    case konstantní výraz : příkaz;
    default : příkaz;
```

Po klíčovém slově *switch* následuje celočíselný výraz, který je uzavřený v závorkách. Následuje příkaz, který se tvoří obvykle blokem. Řídící příkaz je tvořen celočíselným konstantním výrazem uvozeným klíčovým slovem *case* a ukončený dvojtečkou. Poslední varianta návěští bývá (ale nemusí) klíčovým slovem *default* ukončeným dvojtečkou. Program vyhodnotí konstantní výraz a jeho hodnotu porovnává s každým z *case* návěští přepínače.

Nastane-li shoda hodnoty *case* návěští s hodnotou *switch* výrazu, je přeneseno řízení programu na toto návěští. Jinak je řízení přeneseno na návěští *default* v rámci příslušného přepínače. Pro návěští *default* platí stejné zásady jako pro jiná návěští. Pokud nenapišeme *default* návěští a hodnota výrazu *switch* se nebude shodovat s žádným z návěští, bude řízení přeneseno na příkaz následující za přepínačem. Pokud program při provádění přepínače vykoná příkaz *break*, bude řízení přeneseno na příkaz následující za přepínačem. Oproti konstrukci *if-else-if* může být rozhodovací výraz pouze celočíselný. Avšak může být provedeno více příkazů najednou, zatímco u *if-else-if* se provede nejvýše jeden z příkazů. Obdobně u obou konstrukcí se nemusí provést žádný z příkazů.

Cyklus while

Postupně jsme se dostali při výkladu řídicích struktur až k cyklům. Cyklus je způsob, jak určitou část programu budeme provádět vícekrát v závislosti na splnění řídicí podmínky. Řídící podmínka cyklu rozhoduje, zdali bude provedeno tělo cyklu, nebo bude-li cyklus ukončen a provádění programu tak bude pokračovat za příkazem cyklu. Tělem cyklu může být jeden příkaz, či více příkazů v bloku. V jazyce C využíváme třech typů cyklů. Můžeme je rozdělit podle faktu, zdali je příkaz v těle cyklu proveden alespoň jedenkrát, či zdali tělo nemusí být provedeno vůbec.

```

#include <stdio.h>
#include <math.h>

void main()
{
    int volba;
    int x, vysledek;
    do
    {
        printf("1.Vypocti x^2\n");
        printf("2.Vypocti sin(x)\n");
        printf("3.Vypocti x!\n");
        printf("0.Konec \n");

        scanf("%d", & volba);

        switch (volba)
        {
            case 1:
                printf("Zadej x=");
                scanf("%d", & x);
                printf("%d^2=%g\n", x, pow(x,2));
                break;
            case 2:
                printf("Zadej x=");
                scanf("%d", & x);
                printf("sin(%d)=%g\n", x, sin(x));
                break;
            case 3:
                printf("Zadej x=");
                scanf("%d", & x);
                printf("%d!= ", x);
                vysledek=1;
                while (x>0) vysledek=vysledek * x--;
                printf("%d\n", vysledek);
                break;
            case 0:
                break;
            default:
                printf("Spatna volba");
        }
    } while (volba != 0);
}

```

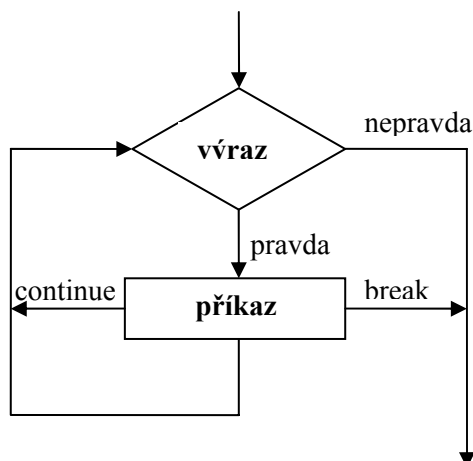
Obr. 4.10 Příklad chodu přepínače

Cyklus **while** vykoná tělo cyklu vícekrát, nebo vůbec, dokud je hodnota vyhodnocení výrazu pravdivá. Obecná syntaxe cyklu *while* je následující:

while (výraz) příkaz;

Příkaz nebo blok příkazů se tedy provede, je-li podmínka splněna.

Pokud použijeme v konstrukci cyklu příkaz *break*, pak tento příkaz ukončuje provádění příkazů těla cyklu a předává řízení prvnímu příkazu za příkazem *while*.



Obr. 4.11 cyklus while

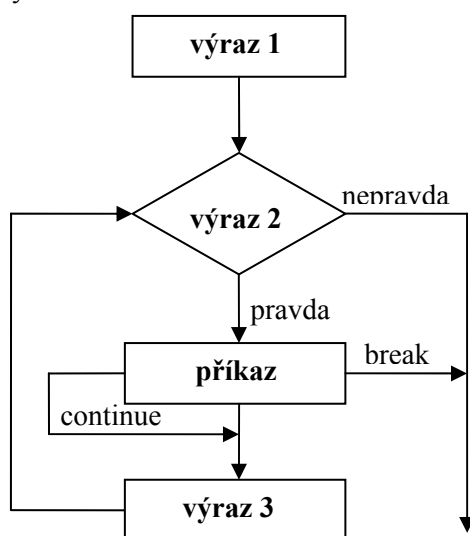
Tímto způsobem můžeme podle potřeby ukončit provádění cyklu bez ohledu na vyhodnocení řídicí podmínky cyklu. Pokud použijeme příkaz *continue*, rovněž ukončíme provádění cyklu, ale řízení programu se vrátí před příkaz cyklu. To znamená, že opět dojde k vyhodnocení testovacího výrazu a cyklus pokračuje dále ve své činnosti. Příkaz *break* a *continue* má tuto úlohu ve všech typech cyklů.

Cyklus for

Cyklus *for* se poněkud liší svou podobou od ostatních cyklů, především svou konstrukcí. Obsahuje ve svém těle tři výrazy: inicializační, podmíněný a opakovací. Inicializační výraz je vyhodnocen pouze jednou a to před vyhodnocením podmíněného výrazu. Tento fakt nám ilustruje následující obrázek. Naproti tomu opakovací výraz je vyhodnocován po každém průchodu těla cyklu, to znamená je vyhodnocován do té doby, dokud je hodnota prostředního podmíněného výrazu splněna. Tento výraz obvykle slouží k úpravě proměnných během provádění cyklu. Syntaxe cyklu *for* je následující:

for (výraz1; výraz2; výraz3) příkaz;

Výrazy jsou v kulatých závorkách a jsou odděleny středníkem. Inicializačním výrazem je výraz1, podmíněným výrazem je výraz2 a opakovacím výrazem je výraz3. Umístění jednotlivých výrazů není povinné, to znamená, že mohou být vynechány. Možný je i případ vynechání všech tří výrazů, tímto způsobem získáme nekonečný cyklus.



Obr. 4.12 cyklus for

Cyklus *for* lze přepsat na variantu příkazu *while*, jehož syntaxe bude následující:

```

výraz1;
while (výraz2)
{
    příkaz;
    výraz3;
}

```



Příklad – použití cyklu *for*

```

/* příklad cyklu for, příklad počítá počty mezer a tabulátorů */
for ( i = space = tab = 0; i < MAX; i++ )
{
    if ( line[i] == ' ')
        space++;
    if ( line[i] == '\t' )
    {
        tab++;
        line[i] = ' ';
    }
}

```

Cyklus *do*

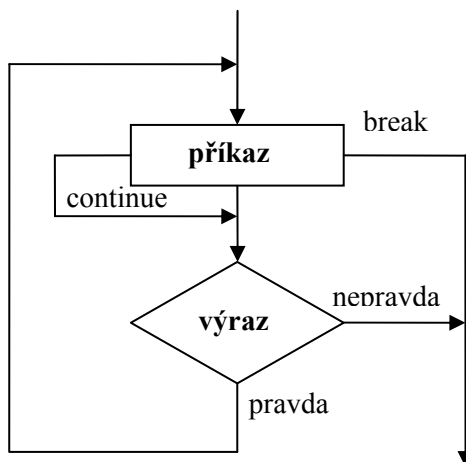
Příkaz cyklu *while* již známe, jeho základní vlastností je fakt, že pokud podmínka testovaného výrazu není splněna, program nemusí projít tělem cyklu ani jednou. Naopak cyklus **do** je jediným z cyklů, který zajišťuje alespoň jedno provedení těla cyklu. To je způsobeno tím, že výraz je testován až po průchodu tělem cyklu. Takže při nesplnění podmínky je tělo cyklu provedeno právě jednou. Syntaktický zápis cyklu *do* je následující:

```

do příkaz;
while (výraz );

```

Příkazy *break* a *continue* se opět mohou vyskytovat v těle cyklu a chovají se stejně, jako v předchozích případech.



Obr. 4.13 cyklus *do*

Časté použití příkazu *do* nalezneme například v matematických výpočtech, kdy je alespoň jedna iterace nezbytná.

Příkaz skoku

Příkaz skoku byl doménou jazyků v dřívějších dobách. Jeho nevhodné použití často vedlo k nežádoucím chybám a hlavně k nepřehlednosti ve zdrojových programech. V éře strukturovaného programování se mu snažíme vyhnout, přesto existují důvody proč ho zcela neztratit. V podstatě příkazy *break* a *continue* jsou označovány jako strukturované skoky. Příkaz skoku **goto** má za úkol přenést běh programu na jiné místo označené návěstím. Jeho syntaxe je následující:

návěstí: příkaz;
goto návěstí;

```
#include <stdio.h>

void main() //čte znaky a tiskne je dokud nanařazí na znak $
{
    int znak;
pokracuj:
    znak = getchar(); putchar(znak);
    if(znak!='$') goto pokracuj;
}
```

Obr 4.14 příkaz goto

Na obrázku 4.14 vidíme použití příkazu skoku na návěstí označené identifikátorem *pokracuj*. Tento příklad není chybný, avšak z hlediska strukturovaného programování zcela nevhodný a samozřejmě princip funkčního programu je možno docílit zcela jiným způsobem bez použití skoku.

Avšak příkaz skoku má přeci jen svůj význam a jsou situace kdy je vhodné ho použít. Sami autoři jazyka Kernighan a Ritchie doporučují použít příkaz *goto* v situacích přerušení nějakého výpočtu v hluboce zanořené části programu, například více zanořených cyklů, protože příkaz *break* nelze aplikovat, jelikož přeruší běh pouze nejzanořenějšího cyklu. Otázkou však zůstává, zdali jsme nepoužili špatnou myšlenku tvorby programu a nemůžeme uvedený problém řešit efektivněji bez takto vzniklých problémů.



Příklad – možné použití příkazu skoku

```
for (...) {
    while (...) {
        do { ...
            if(problem)
                goto chyba; /* odskok na návěstí chyba */
        } while(...)
    }
    ...
    chyba: ... /* návěstí chyba pro záchranu situace */
}
```



Korespondenční úkol – strukturovaný program

Program ilustrovaný na obrázku čte znaky z klávesnice a tiskne je na obrazovku dokud nenarazí na znak \$. Upravte program tak, aby jste zachovali jeho funkčnost a vyhněte se příkazu skoku.



Ke kapitole 4.7 je přiřazena demonstrační animace č. 8

Animace č. 8 je věnována řídicím strukturám jako je větvení a konstrukce cyklů.

4.8 Preprocesor

Nyní na chvíli odskočíme od syntaxe příkazů jazyka C a věnujme se opět trochu zákulisí zpracování programu. V úvodní kapitole jsme se věnovali postupu, jež nám nastínil, co se děje od samotného zápisu zdrojového kódu programu až po vytvoření spustitelné verze programu ve tvaru *exe* souboru. Definovali jsme si pojmy jako hlavičkový soubor, překladač a spojovací program. K úplnému postupu nám ještě chybí vysvětlit si pojem funkce preprocesoru. Osvětlí nám to tak některé situace, které bereme jako dogma a dosud jsme je nerozebírali. Při praktických cvičeních se tak často pedagogové setkávají s otázkami, proč používáme znaky jako je #, proč v některých situacích neukončujeme řádek středníkem, proč někdy při vkládání hlavičkových souborů používáme uvozovky a jindy ostré závorky a další otázky zvědavých studentů. Na tyto otázky najdeme odpověď v této kapitole.

Ještě před samotným překladem zdrojového kódu totiž hraje svou úlohu preprocesor jazyka. Preprocesor tedy předchází překladač, zpracovává vstupní text, provádí v něm textové změny a jeho výstupem je opět text. Neprovádí tedy žádnou kontrolu syntaxe, nebo typovou kontrolu. Preprocesor zpracovává hlavičkové soubory, odbourává komentáře, umožňuje provádět podmíněný překlad zdrojového textu a rozvíjí tzv. makra. Jeho direktivy zvyšují přenositelnost kódu a zpřehledňují program.

ANSI norma jazyka C definuje následující množinu klíčových slov preprocesoru:

#define	#elif	#else	#endif
#error	#if	#ifdef	#ifndef
#include	#line	#pragma	#undef

Tab. 4.6 Direktivy preprocesoru



Pojem k zapamatování: Direktiva preprocesoru

Direktiva preprocesoru musí být vždy uvozena znakem #.

Direktiva preprocesoru není příkaz jazyka C. Neukončujeme ji proto středníkem.

Aniž bychom věděli, že se jedná o direktivu preprocesoru, již od prvního příkladu jsme ji používali. Jedná se o direktivu *#include*.

Direktivu *#include* používáme pro včlenění zdrojového textu jiného souboru. Tento soubor může být začleněn více způsoby.

1. `#include <header_name>`
2. `#include "header_name"`
3. `#include macro_identifier`

První způsob začlenění souboru je uvozen ostrými závorkami, což v principu znamená, že soubor `header_name` je hledán ve standardním adresáři pro `include`. Tímto způsobem začleňujeme standardní hlavičkové soubory. V případě, že soubor není nalezen, je ohlášena chyba. Druhým způsobem začleňujeme hlavičkové soubory vytvořené uživatelem např. `#include "moje_funkce.h"`. Hlavičkový soubor je nejprve hledán v aktivním (pracovním) adresáři. Není-li tam nalezen, postupuje se podle první možnosti. Třetím nejméně obvyklým způsobem můžeme začlenit hlavičkový soubor pomocí makro identifikátoru, který je dosazen na své místo.

Direktivy **#define** a **#undef** využíváme pro definici symbolických konstant a maker. Definice maker je typickým příkladem použití preprocesoru. Při své činnosti preprocesor zpracovává zdrojový text a při výskytu definovaného makra či symbolické konstanty provede její náhradu. Tomuto postupu se říká expanze makra. Za direktivou *#define* definujeme identifikátor makra či symbolické konstanty a jeho hodnotu, kterou preprocesor využije při náhradě, která bude použita při překladu. Program tak vhodně zpřehledníme a máme možnost případnou změnu hodnoty provést pouze jednou bez pracného prohledávání zdrojového kódu.

Pokud v definici uvedeme seznam argumentů, jedná se již o makro příkaz. Definujeme je takto:

#define identifikátor ([seznam argumentů]) [řetězec]



Příklad – definice symbolických konstant

```
#define POCATEK 10
#define KONEC 100
#define KROK 2
...
int main()
{ int soucet = 0, od = START, do = KONEC, krok = KROK;
...
}
```

Seznam argumentů je oddělen čárkou a následuje řetězec, jímž je makropříkaz rozveden. Typickým příkladem makra je v odborné literatuře prezentován příklad vrácení větší hodnoty ze dvou čísel:

```
#define max(x,y) ((x)>(y) ? (x) : (y))
```

Při definici je nutné dbát na použití závorek. I když se mohou jevit jako nadbytečné, není tomu tak, protože musíme mít na paměti, že se jedná o textový rozvoj, neboli náhradu. Závorky tak určí prioritu vyhodnocování výrazů při rozvoji makra. Právě proto, že se jedná pouze o textový rozvoj a předem nevíme, jak uvedené proměnné mohou být použity (není vyloučený např. tvar inkrementace v postfixovém nebo prefixovém tvaru), není doporučováno používat makra jako náhradu funkcí, neboť se můžeme dočkat zcela nečekaných výsledků a problémů.

Podle ANSI standardu umí preprocesor C identifikovat a vyhodnocovat standardně předdefinovaná makra. Ty jsou obklopeny dvěma podtržítky a mají následující význam:

`__DATE__` datum překladu,
`__FILE__` jméno zdrojového souboru,

__LINE__ právě zpracováváný řádek ve zdrojovém souboru,
 __STDC__ definuje typ (úroveň) překladu (STanDard C),
 __TIME__ čas překladu.

#error je direktivou, kterou můžeme zajistit výstup vlastního chybového hlášení. Nejčastěji se používá v souvislosti s podmíněným překladem. Chybové hlášení pak bude součástí protokolu o překladu.

Direktiva **#line** se často využívá u strojově generovaných zdrojových textů a nebudeme se jí podrobněji zabývat.

#pragma je implementačně závislou direktivou, kterou využívají mimo jiné různé speciální kompilátory. Můžeme se s ní například setkat v hojné míře u kompilátorů jazyka C pro různé procesory a mikrokontroléry, využívající jazyk C jako nadstavbu svého assembleru, či jako nastavení specifických parametrů pro různé operační systémy. Tato problematika je mimo rámec tohoto kurzu a tak si alespoň zapamatujme, že pokud jiný překladač speciální direktivu nezná, prostě ji bez ohlášení chybového stavu ignoruje.

Použitím dalších direktiv **#if**, **#elif**, **#else**, **#endif**, **#ifdef** a **#ifndef** můžeme obměňovat dle potřeby části zdrojového textu. Této vlastnosti se říká podmíněný překlad. Za použití klíčového slova preprocesoru *defined* pak může realizovat rozsáhlejší logické podmínky, jimiž můžeme řídit zpracování zdrojového kódu. Argument *defined* nemusí být v závorkách, ale musí být použit s direktivou **#if** nebo **#elif**. V závislosti na vyhodnocení podmínky se určí, bude-li ohraničený úsek programu dále zpracován, nebo bude-li ignorován a nebude zpracován překladačem.



Příklad – ukázka podmíněného překladu

```
/* ukazka podmineneho prekladu */
#define TESTPROGRAMU

...

#ifdef TESTPROGRAMU
printf("program je stale ve verzi testovani\n");
#endif
```

Uvedený příklad ilustruje použití direktiv při podmíněném překladu. Pokud je dané makro nadefinováno, podmínka je splněna a část kódu mezi direktivami **#if defined** a **#endif** se stane součástí kódu, který bude přeložen. V případě, že makro definováno není, tomu tak samozřejmě nebude.

4.9 Funkce

Funkce jsou nedílnou součástí jazyka C. V podstatě ihned při napsání nejjednoduššího programu typu *hello word* jsme použili minimálně dvě funkce. Tou první je funkce *main*, kterou musí obsahovat každý program. Funkce mají za úkol určitou opakovatelnou část kódu sjednotit a zobecnit, tak abychom ji mohli využít v různých situacích, a pomohou nám tak zpřehlednit a zrychlit psaní kódu. Další výhodou je bezpochyby snadnější změna programu, lepší ladění a menší výskyt chyb. Funkce bychom mohli rozčlenit na tři skupiny: **standardní**, které jsou definovány normou jazyka, **uživatelské**, které jsme sami napsali ke konkrétním příkladům a třetí skupinou funkcí, se kterou se ještě můžeme setkat, jsou funkce **nadstavbové** ke konkrétním komerčním produktům a rozšířením překladače od různých softwarových výrobců.

Definice funkce a její návratová hodnota

Každá funkce má svou definici a základních vlastností:

- je pojmenována a tímto jménem se volá v programu,
- může, ale nemusí mít parametry, jejichž prostřednictvím předáváme data do těla funkce ve které jsou následně zpracovány,
- má návratovou hodnotu obsahující výsledky zpracování v těle funkce.

```
typ jméno(parametry)
{
    tělo funkce
}
```



Příklad – definice funkce

```
/* ukazka definice funkce max */
int max(int a, int b)
{
    if(a > b)
        return a;
    return b;
}
```

Deklaraci funkce provádíme v hlavičkovém souboru. Na rozdíl od definice funkce již neobsahuje tělo a je vždy ukončena středníkem. Hlavičkové soubory tak obsahují identifikátor funkce a její parametry. Důvod vychází z předpokladu, že voláme funkce v programu, které jsou fyzicky umístěny v jiných souborech, a překladač při překladu musí znát alespoň tuto deklaraci. Z vlastní zkušenosti víme, že využíváme celou řadu funkcí, které jsou definovány na jiném místě, proto tyto hlavičkové soubory vkládáme direktivou `#include` na začátek zdrojového kódu.

Pro návratovou hodnotu funkce musíme určit její datový typ. Na typ funkce nejsou kladena žádná speciální omezení. V případě že danou návratovou hodnotu nebudeme dále zpracovávat, tak ji jednoduše nepoužijeme. Pokud ale chceme deklarovat funkci, která nevrací žádnou hodnotu, pak musíme použít v deklaraci klíčové slovo `void`.

Návratová hodnota je uvedena za příkazem `return`. Typ `return` výrazu musí být typově slučitelný s deklarovaným typem návratové hodnoty funkce. Příkaz `return` se v těle funkce nemusí vyskytovat pouze jednou. Pokud to odpovídá větvení ve funkci, může se vyskytovat na konci každé větve. V každém případě však příkaz `return` ukončuje činnost funkce.

return výraz_vhodného_typu;

Měli bychom také vědět, jakým způsobem jsou argumenty předány funkci. Parametry, jež používáme, dělíme do dvou skupin: na **formální** parametry, které zadáváme v definici a deklaraci funkce, a **skutečné** parametry, které jsou funkci předány při jejím volání. Obecně v programovacích jazycích předáváme argumenty dvěma způsoby: **hodnotou** nebo **odkazem** (adresou).

Při předání argumentu hodnotou se do formálního parametru zkopíruje hodnota skutečného parametru. Změny hodnot pak neovlivňují skutečný parametr použitý při jeho volání, to znamená, že výpočty a operace s parametrem uvnitř funkce neovlivní parametr (skutečný) mimo funkci. Naopak při předání parametru odkazem se do formálního parametru zkopíruje adresa skutečného parametru. Proto se tento způsob předávání také označuje jako předávání adresou. Uvnitř funkce se adresa použije pro přístup ke skutečnému parametru a změny provedené přes tento formální parametr mají vliv na skutečný parametr.

V jazyce C můžeme definovat funkce, které nemají pevně dán počet parametrů. Tyto funkce v podstatě používáme stále, neboť mezi takové funkce patří například funkce formátovaného vstupu a

výstupu *printf* a *scanf*. ANSI norma jazyka C definuje pro tyto případy makra, avšak v tomto základním kurzu se definicí vlastních funkcí s proměnným počtem argumentů nebudeme zabývat.

Rekurzivní funkce

Rekurzi nazýváme fakt, kdy daná funkce ve svém těle volá sama sebe, musíme v ní zajistit podmínku, při které funkce končí, jinak bychom ji museli násilně ukončit, což je nepřípustné. Nevýhodou rekurzivních funkcí jsou paměťové nároky na zásobník, protože funkce volá sama sebe a je nutné stavy jednotlivých volání ukládat. Musíme tedy zvážit, zdali je použití rekurze pro nás přínosem, či zadaný problém vyřešíme jiným způsobem, protože každou rekurzi lze převést na iteraci. V odborné literatuře se stala klasickou ukázkou rekurze funkce na výpočet faktoriálu:



Příklad – ukázka rekurzivní funkce

```
/* ukázka rekurzivní funkce vypočtu faktoriálu */
double faktorial(int i)
{
    double vysledek;
    if(i==0)
        return 1;

    vysledek = i * faktorial(i-1);
    return(vysledek);
}
```

Funkce faktoriálu při svém zavolání počítá s parametrem, jež udává požadovanou hodnotu faktoriálu. Pokud tato hodnota není nulová, je zavolána opět funkce faktoriál s dekrementovanou hodnotou parametru. Dekrementací je jednoznačně určen fakt, že funkce jednou skončí a to při zavolání hodnoty 0!, což odpovídá hodnotě 1. Poté mají dříve zavolané funkce již podklady pro dokončení svého výpočtu a jsou postupně uvolňovány z návratové adresy zásobníku.



Ke kapitole 4.9 je přiřazena demonstrační animace č. 9

Animace č. 9 je věnována tvorbě funkcí standardních i rekurzivních.

4.10 Ukazatele

Ukazatel v jazyce C reprezentuje adresu objektu. Obsahuje také informaci o datovém typu, který se na této adrese nachází.

Pomocí operátoru získání adresy **&** můžeme provést přiřazení adresy proměnné.



Pojem k zapamatování: Ukazatel

Ukazatel je proměnná, jejíž hodnota je adresa další proměnné nebo funkce. Adresy proměnných tedy můžeme ukládat do proměnné typu ukazatel.

Pokud ukazatel ukazuje na proměnnou v paměti, můžeme s touto proměnnou dále pracovat pomocí ukazatele. K tomu potřebujeme operátor *****. Oba operátory získání adresy i dereference jsou unární a oba mají přednost před aritmetickými operátory.



Příklad – Inicializace ukazatele

```
/* inicializace ukazatele v jazyce C */
int Promenna = 2; /* Promenna je typu int */
int *pPromenna; /* pPromenna je ukazatel na libovolnou promennou typu int */
pPromenna = &Promenna; /* pPromenna ukazuje na promennou Promenna */
```

Ukazatele se mohou objevovat v různých typech výrazů, jejich vyhodnocování se věnuje adresová aritmetika. Samotné ukazatele jsou širokou problematikou s velkým rozsahem využití, avšak překračují náplň základů jazyka C. Podrobněji se s nimi seznámíte v navazujícím předmětu Operační systémy a programování.



Příklad – ukázka výrazu s ukazateli

```
int x, y = 1; /* Promenne x,y jsou typu int */
int *px, *py; /* px a py jsou ukazatele na promennou typu int */
px = &x; /* px ukazuje na promennou x */
py = &y; /* py ukazuje na promennou y */
*px = y + 5; /* do promenne na kterou ukazuje px je ulozen součet y + 5 */
*px += 5; /* k promenne na kterou ukazuje px je přičtena petka */
(*py)++; /* promenna na kterou ukazuje py je inkrementována – závorky nutné */
```



Pojem k zapamatování: Ukazatele

Pokud máme dano přiřazení `px = py`, kde `px` a `py` jsou ukazatele s předchozího příkladu, pak je ukazateli `px` přiřazena hodnota ukazatele `py`. To znamená, že nyní ukazují na stejné místo. Neznačená to tedy, že by byl zkopírován obsah proměnné, ale jejich adresy.

4.11 Pole

Nejen v matematických úlohách, ale i při řešení běžných příkladů z různých oblastí se nevyhneme práci s poli. Jazyk C samozřejmě na tento fakt pamatuje a zavádí pojem pole jako kolekci proměnných stejného typu, uložených v paměti za sebou. Jednotlivé prvky pole jsou označeny společným jménem a k jejich přístupu používáme indexy. Pole je definováno jako jednorozměrné, avšak prvkem pole může být proměnná libovolného typu a samozřejmě i pole. Tento fakt nám tedy zajistí možnost definovat vícerozměrná pole jako pole polí. Definice pole je následující

typ jméno[rozsah];

kde typ určuje typ prvků pole, jméno je identifikátorem a rozsah je počet prvků pole.



Pojem k zapamatování: Rozsah pole

Rozsah pole neznámá nic jiného než vlastní počet prvků pole. Důležitým faktem ale je skutečnost, že pole začíná prvkem s indexem nula, což znamená, že poslední prvek pole má index `rozsah-1`. Pokud překročíme rozsah indexu pole, nezaznamenáme chybu, neboť překladač se o rozsah nestará, avšak budeme pracovat s hodnotami, které se odkazují na paměťová místa bezprostředně za polem. Tyto hodnoty při odkazu na neexistující prvky pole budou dávat nesmyslné výsledky.

Pole tedy můžeme definovat například takto:



Příklad – definice pole

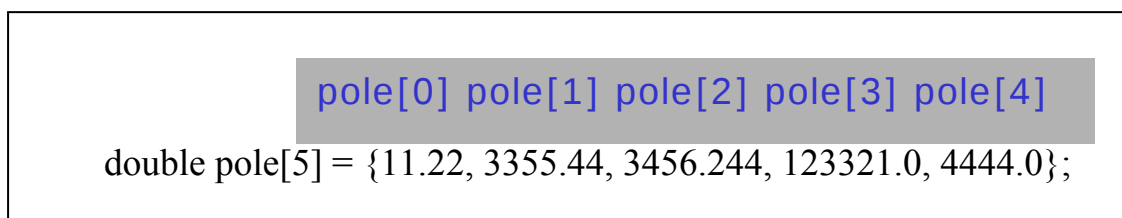
```
/* definice pole v jazyce C */
#define PO CET 100
int pole[PO CET];
```

Pole s názvem *pole* je deklarováno jako pole o 100 prvcích definovaných v proměnné *PO CET*. Všechny prvky jsou jednotného typu *integer*. Na jednotlivé prvky se můžeme odkazovat indexy 0 až 99, neboli *pole[0]*, *pole[1]*, ..., *pole[PO CET-1]*.

Pokud budeme chtít zjistit velikost paměti přidělené pro tohle pole, použijeme vztah:

$$\text{počet obsazených byte} = N * \text{sizeof(int)}$$

Můžeme společně s definicí pole provést i jeho deklaraci obdobně jako u ostatních proměnných.

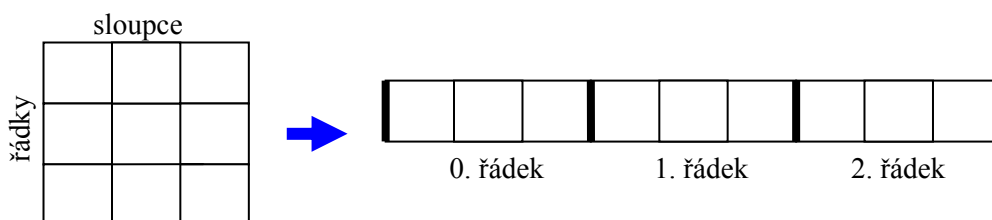


Obr. 4.15 Deklarace pole

Můžeme také využívat vícerozměrná pole. S vícerozměrným polem pracujeme obdobným způsobem jako s jednorozměrným polem. Definice dvourozměrného pole, které využíváme v maticovém počtu, vypadá následovně:

type jmeno_pole[m][n];

kde typ určuje datový typ položek pole a *jmeno_pole* je identifikátor pole *m* x *n*, tedy *m* řádku a *n* sloupců. Počet prvků je tedy jednoznačný a je dán deklarací.



Obr. 4.16 Uspořádání dvourozměrného pole 3x3 v paměti



Příklad – ukázka vícerozměrného pole

```
/* deklarace matice 3x3 */
float matice[3][3]={
    {1, 1, 1},
    {1, 1, 1},
    {1, 1, 1}};
```

..

..

```
/*funkce souctu matic 3x3*/
```

```
void SectiMatice(float m1[3][3], float m2[3][3], float vysledek[3][3])
{
    int i, j;
    for(i=0; i<3; i++)
        for(j=0; j<3; j++)
            vysledek[j][i]=m1[j][i]+m2[j][i];
}
```

4.12 Řetězce

V jazyce C definujeme také pojem řetězec. Řetězec chápeme jako jednorozměrné pole znaků a používá znak `\0` jako ukončovací zarážku. Tento fakt je důležitý při počáteční definici velikosti pole. Pro zarážku musíme rezervovat jeden znak. Definice řetězce jako jednorozměrného pole je následující:

```
char s[SIZE];
```

Identifikátor `s` můžeme chápat jako proměnnou typu `char` délky `SIZE`. K jednotlivým znakům pak přistupujeme pomocí indexů `s[0]` až `s[SIZE-1]`. Druhou možností je pohled na identifikátor `s` jako na konstantní ukazatel na znak, t.j. na první prvek pole `s`.

"ahoj" je konstanta typu řetězec délky 4 znaky plus jeden znak na zarážku, dohromady tedy 5 bajtů paměti.



Pojem k zapamatování: Řetězcová vesus znaková konstanta

Samotný znak "a" v uvozovkách je také chápán jako řetězcová konstanta, a má délku 1+1 na rozdíl od znakové konstanty 'a' v apostrofech.

Pro práci s řetězci jsou definovány řetězcové funkce, jejichž přehled uvádí následující tabulka 4.7.

Přehled nejpoužívanějších řetězcových funkcí	
<code>int strcmp(const char *s1, const char *s2);</code>	Lexikograficky porovnává řetězce, vrací hodnoty menší než 0 je-li <code>s1 < s2</code> , 0 pokud <code>s1 == s2</code> a větší než 0 je-li <code>s1 > s2</code> .
<code>int strncmp(const char *s1, const char *s2, unsigned int n);</code>	Jako předchozí funkce s tím rozdílem, že porovnává nejvýše <code>n</code> znaků.
<code>unsigned int strlen(const char *s);</code>	Vrátí počet významných znaků řetězce (bez zarážky).
<code>char *strcpy(char *dest, const char *src);</code>	Nakopíruje řetězec <code>src(source)</code> do <code>dest(destination)</code> .
<code>char *strncpy(char *dest, const char *src, unsigned int n);</code>	Jako předchozí funkce, ale nejvýše <code>n</code> znaků (je-li jich právě <code>n</code> , nepřidá zarážku).

<i>char *strcat(char *s1, const char *s2);</i>
Řetězec <i>s2</i> přikopíruje za řetězec <i>s1</i> .
<i>char *strncat(char *s1, const char *s2, unsigned int n);</i>
Jako předchozí funkce, ale nejvýše <i>n</i> znaků.
<i>char *strchr(const char *s, int c);</i>
Vyhledá první výskyt (zleva) znaku <i>c</i> v řetězci <i>s</i> .
<i>char *strrchr(const char *s, int c);</i>
Vyhledá první výskyt (zprava) znaku <i>c</i> v řetězci <i>s</i> .
<i>char *strstr(const char *str, const char *substr);</i>
Vyhledá první výskyt (zleva) podřetězce <i>substr</i> v řetězci <i>str</i> .

Tab. 4.7 Řetězcové funkce



Ke kapitole 4.11 a 4.12 je přiřazena demonstrační animace č. 10

Animace č. 10 je věnována některým řetězcovým funkcím a práci s dvourozměrným polem.

4.13 Práce se soubory

Jazyk C umožňuje se soubory pracovat na dvou odlišných úrovních. První z nich je tzv. přímé volání. Tento pojem znamená přímé volání služeb jádra systému. Druhým přístupem je datový proud a způsob práce s ním definuje ANSI norma jazyka C. Základem pro přístup k proudu je datová struktura **FILE**. Při každém spuštění programu máme otevřeny proudy, jež deklarujeme následujícím způsobem:

```
FILE*stdin;
FILE*stdout;
FILE*stderr;
```

Pro práci se soubory máme nadefinovanou řadu funkcí začínající na písmeno *f* (*file* = *soubor*) tedy např. tyto funkce:

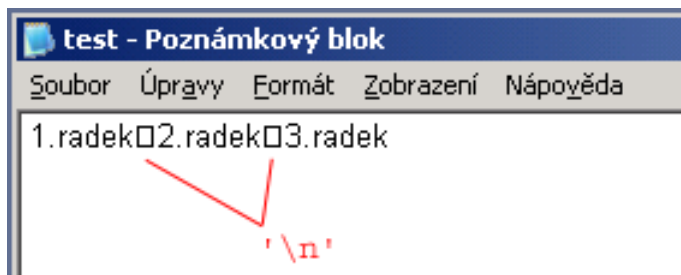
Přehled základních funkcí se soubory	
<i>fopen/fclose</i>	otevře/zavře soubor
<i>fprintf/fscanf</i>	zapisuje/čte ze souboru jako printf/scanf
<i>fputc/fgetc</i>	zapisuje/čte znak ze souboru
<i>feof</i>	funkce pro zjištění konce souboru (file end of file)
<i>fputs/fgets</i>	čte/zapisuje řádek (řetězec)
<i>fread/fwrite</i>	čte/zapisuje blok dat (např. celou strukturu)
<i>fseek</i>	posunuje čtecí ukazatel v souboru (zápisový)

Tab. 4.8 Funkce pro práci s datovými proudy

Všechny výše uvedené funkce se nacházejí ve standardním hlavičkovém souboru `<stdio.h>`. Soubory mohou být v jazyce C dvou druhů:

- textové - t
- binární - b

Rozdíl z hlediska jazyka C je minimální. Soubor v textovém módu je při zápisu/čtení na disk upravován tak, aby platilo: `'\n' <=> '\r' '\n'`. Tedy znak `'\n'` je v souboru nahrazen dvěma znaky a naopak - to vyplývá z prapůvodní definice ASCII zejména ve vztahu k textovým tiskárnám. Každý proud máme možnost otevřít a uzavřít. Při otevření určujeme režim našeho přístupu k datům v proudu.



Obr. 4.17 Otevření souboru pro čtení

FILE *fopen(const char *filename, const char *mode);

Konstantní řetězec *filename* označuje jméno souboru. Řetězec *mode* určuje režim práce se souborem a jeho typ.

```
FILE* soubor;
soubor = fopen("D:\\data.txt", "rt"); //čtení v textovém modu
soubor = fopen("D:\\data.txt", "rb"); //čtení v binárním modu
```

Obr. 4.18 Otevření souboru pro čtení

Řetězec	Význam
<i>r</i>	čtení
<i>w</i>	zápis
<i>a</i>	připojení
<i>r+</i>	aktualizace (update) - jako rw
<i>w+</i>	jako výše (r+), ale existující proud ořízne na nulovou délku, jinak vytvoří nový
<i>a+</i>	pro aktualizaci, pokud neexistuje, vytvoří nový

Tab. 4.9 Význam atributů pro práci se soubory

Pokud při otevření souboru vznikne chyba, tak to může znamenat že:

- soubor neexistuje,
- nemáte k němu přístupová práva,
- je otevřený v jiném programu, tak funkce `fopen` vrací hodnotu `NULL`.

```
FILE* soubor;
soubor = fopen("Q:\\neexist.txt", "rt");
if (soubor==NULL) printf("Soubor nelze otevrit\n"); // a konec programu
```

Obr. 4.19 Test na detekci chyb při otevření souboru:

int fclose(FILE *stream);

Je funkce uzavírající určený proud. V případě úspěchu vrátí hodnotu 0, v opačném případě EOF. Dále pak uvolní paměť vyhrazenou pro strukturu FILE.

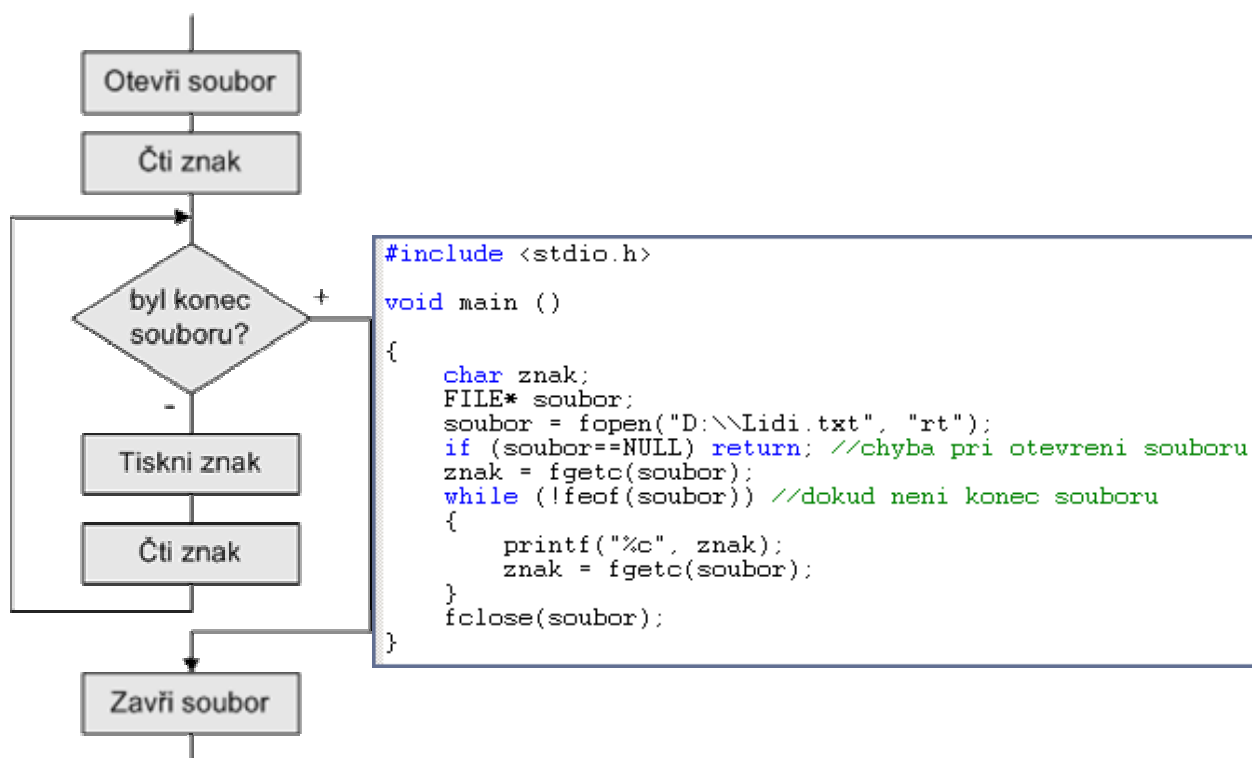
Pro vstup/výstup znaků do/z proudů jsou k dispozici následující funkce. Znak je po načtení z proudu konvertován bez znaménka na typ int. Obdobně je při zápisu do proudu konvertován opačným postupem.

int getc(FILE *stream);

V případě úspěšného načtení znaku z proudu jej bez znaménka převede na typ int. Takto získaná hodnota je hodnotou návratovou. V případě chyby nebo dosažení konce proudu pro getc() vrací EOF.

int putc(int c, FILE *stream);

Zapiše znak c do proudu stream. Vrátí stejnou hodnotu, jakou zapsal. V případě chyby nebo dosažení konce proudu pro putc() vrací EOF.



Obr. 4.19 Příklad procházení souboru

Z proudu nemusíme číst pouze jednotlivé znaky. Můžeme načítat i celé řádky. Ty jsou ukončeny přechodem na nový řádek. Pro vyšší bezpečnost musíme při čtení uvést velikost vyrovnávací paměti. Při zápisu to pochopitelně nutně není. Do proudu se zapíše celý řetězec až po koncovou zarážku (ale bez ní).

char *fgets(char *s, int n, FILE *stream);

Načte řetězec (řádek až po jeho konec) z proudu stream do vyrovnávací paměti, nejvýše dlouhý *n-1* znaků. Vráti ukazatel na tento řetězec (vyrovnávací paměť) nebo při chybě NULL.

int fputs(const char *s, FILE *stream);

Zapiše do proudu řetězec ukončený zarážkou. Ani zarážku, ani případný konec řádku (obsažený na konci řetězce) do proudu nezapiše. V případě úspěchu vrátí počet zapsaných znaků (délku řetězce), jinak EOF.

Jestliže jsme zvládli standardní formátovaný vstup a výstup, nemůže nám činit formátovaný vstup a výstup z a do proudu potíže. Funkce se navzájem liší pouze úvodním *f* a identifikací proudu jako prvního argumentu. Právě srovnání je důvodem, proč uvádíme odpovídající funkce ve dvojici.

int fprintf(FILE *stream, const char *format [, argument, ...]);

int printf (const char *format [, argument, ...]);

int fscanf (FILE *stream, const char *format [, address, ...]);

int scanf (const char *format [, address, ...]);

```
void main ()
{
    FILE* soubor;
    soubor = fopen("D:\\test.dat", "wt"); //otevre textove pro zapis
    fprintf (soubor, "1.radek\n2.radek\n3.radek");
    fclose(soubor);
}
```

textově

```
FILE* soubor;
soubor = fopen("D:\\test.dat", "wb"); //otevre textove
fprintf (soubor, "1.radek\n2.radek\n3.radek");
fclose(soubor);
```

binárně

Obr. 4.20 Zápis do souboru v textovém a binárním tvaru

int feof(FILE *stream);

Je funkce, umožňující zjistit dosažení konce proudu. Její návratová hodnota je *true* (t.j. různá od nuly), nacházíme-li se na konci proudu, nebo *false* (nula).

int fseek(FILE *stream, long offset, int whence);

Přenesení aktuální pozici CP v proudu stream na stanovené místo. To je určeno posunem offset vzhledem k počátku, aktuální pozici nebo konci souboru. Vztažný bod určuje argument whence. Předdefinované konstanty jsou stejné, jako v případě přímého přístupu - tj. lseek() (jsou to SEEK_SET, SEEK_CUR, SEEK_END).



Ke kapitole 4.13 je přiřazena demonstrační animace č.11

Animace č. 11 je věnována funkcím pro práci se soubory.

4.14 Argumenty příkazového řádku

Během spravování či instalace určitých aplikací jste se již určitě setkali se spuštěním programu s příkazového řádku společně s jedním nebo několika argumenty, např. *aplikace.exe /log*.

Jazyk C umožňuje vytváření takového programu a to zajištěním předávání parametrů hlavní funkce *main*.

```
int main (int argc, char *argv[])
```

První argument funkce *main()* udává počet argumentů (anglicky *count*, proto *argc*) a druhý argument představuje hodnoty těchto argumentů, které jsou polem ukazatelů na řetězce.

Počet argumentů příkazové řádky je ≥ 1 , protože počet argumentů roven jedné znamená pouze jméno programu, počet rovný dvěma již znamená jméno programu a jeden argument, atd.). Argument *argc* je proto typu *int*.



Ke kapitole 4.14 je přiřazena demonstrační animace č. 12

Animace č. 12 je věnována argumentům příkazového řádku.

4.15 Deklarace odvozených a strukturovaných datových typů

Odvozené a strukturované datové typy nám umožňují řešit úlohy podle našich potřeb, kdy můžeme lépe modelovat reálnou situaci tak, jak ji známe v běžném životě. Můžeme nadefinovat vlastní datový typ, vytvořit výčtový typ vyjmenováním konečné množiny možností, jež představují výčtové konstanty, definovat strukturu proměnných, která se skládá z dílčích částí s různými datovými typy a tyto pak sloučíme pod jedno jméno, nebo naopak vytvořit unii, která má v různých situacích různý datový typ.

Uživatelský datový typ

V jazyce C máme k dispozici datové typy, které by měly pokrýt veškeré typy řešených úloh. Přesto však může dojít k situaci, kdy cítíme potřebu použít nějaký upravený datový typ podle svých představ. Tento fakt nám umožňuje použít nástroj s názvem **typedef**, jeho syntaxe je následující:

```
typedef typ identifikátor;
```

Po klíčovém slově *typedef* následuje definice typu a jeho identifikátor.



Příklad – ukázka definice vlastního typu

```
/* definice pomocí typedef */
typedef int vyska;
typedef float vaha;
vyska v=178;
vaha w= 78.5;
```

Výčtový typ

Výčtový typ nám umožňuje definovat výčtové konstanty. Po deklaraci nelze hodnotu identifikátoru modifikovat. Syntaxe výčtového typu je následující:

```
enum [typ] {konstanta [= hodnota], ...} [list proměnných];
```

Seznam identifikátorů je seřazen a jazyk C je automaticky čísluje od nuly. Číslování však můžeme ovlivnit a vynutit si vlastní použitím nepovinné konstrukce = *hodnota*.



Příklad – ukázka výčtového typu

```
/* definice výčtového typu */
enum Den          // Deklarace výčtového typu Den
{
    pondeli,  utery,  streda,  ctvrtek,  patek,  sobota,  nedele
};
enum Den dnes, vcera;
/* int utery; chyba nelze predefinovat */
vcera = utery;
dnes = streda;
/* ctvrtek = 0; chyba nelze predefinovat */
```

Typ struktura

Strukturu využíváme při popisu více souvisejících informací různého typu. Často je používáme pro jednodušší manipulaci se záznamy přidruženými k jednomu objektu z reálného života. Např. pro osoby, zaměstnance atd. potřebujeme informace různých datových typů, protože obvykle potřebujeme jméno, příjmení, rodné číslo či jiný číselný údaj. Tyto informace pak přiřadíme jedné logické jednotce, která nese jméno struktury. Uvedeme opět syntaxi:

```
struct [jméno struktury] {
    [typ jméno proměnné[,jméno proměnné, ...]] ;
    [typ jméno proměnné [,jméno proměnné , ...]] ;
    ...
}[proměnné struktury] ;
```

Pro přístup k prvkům struktury používáme selektor struktury, kterým je **tečka**. Tu umístíme mezi identifikátory proměnné typu struktura a identifikátor položky, s níž chceme pracovat.



Příklad – ukázka struktury

```
/* deklarace struktury zamestnanec */
struct Zamestnanec
{
    int vek;
    long id_cislo;
    char *jmeno;
    char *prijmeni;
};
struct Zamestnanec ucitel, skolnik;
ucitel.vek = 50; // pristup k promenne struktury
skolnik.prijmeni = "Buml"; // pristup k promenne struktury
```

Typ Unie

V unii můžeme uchovávat v konkrétním případě jen jeden typ dat, ovšem v různých situacích může mít různý typ a velikost. Syntakticky vypadá konstrukce union:

```
union [jméno unie] {
    typ proměnná;
    typ proměnná;
    ...
} [jméno přiřazené proměnné unie] ;
```

Z položek unie lze tedy používat v jednom okamžiku pouze jednu. Ostatní mají nedefinovanou hodnotu. V paměti je pak pro proměnnou vyhrazeno místo, které odpovídá paměťově nejnáročnější položce.

Důležitým faktem vycházejícím z vlastností unie je nemožnost opětovného použití první položky floatového čísla v uvedeném příkladu (následující příklad – ukázka unie), protože paměťové místo pro unii *ziskaneCislo* bylo přepsáno z hodnoty 51.4 na hodnotu 10.



Příklad – ukázka unie

```
/* deklarace a použití unie */
union Cislo
{
    int i;
    float f;
    double d;
} ziskaneCislo;
ziskaneCislo.f = 51.4;
ziskaneCislo.i = 10;
```



Další zdroje

1. The documentation for Visual C++. MSDN Library Visual Studio 6.0.
2. Visual C++ Developer Center [online] dostupný z [http:// msdn.microsoft.com/ visualc](http://msdn.microsoft.com/visualc)
3. Herout, P. *Učebnice jazyka C*, České Budějovice, Kopp 2005, počet stran: 280. ISBN 80-7232-220-6.
4. Herout, P. *Učebnice jazyka C – II díl*, České Budějovice, Kopp 2005, počet stran: 180. ISBN 80-7232-221-4.
5. Kernighan, B. W., Ritchie D. M. *Programovací jazyk C*, Brno, Computer Press 2006, počet stran: 288. ISBN 80-251-0897-x.
6. Kadlec, V. *Učíme se programovat v jazycce C*, Brno, Computer Press 2002, počet stran: 294. ISBN 80-7226-715-9.
7. Virius, M. *Jazyky C a C++*, Brno, Computer Press, 2005, počet stran: 520. ISBN 80-247-1494-9.



Shrnutí kapitoly

Jazyk C je fenomenálním jazykem autorů Briana W. Kernighana a Denise M. Ritchieho. Původně byl jazyk navržen v roce 1972 pro operační systém Unix, avšak po dalším vývoji byl implementován na všechny známé vývojové platformy. Jazyk C je tedy hardwarově a softwarově nezávislý. V roce 1983 byla definována norma jazyka ANSI C (*zkratka American National Standards Institute*). Tento standard byl finálně dokončen až v roce 1988. Následovala řada standardů ISO, které ANSI normu přebírají. S ANSI normou jazyka budeme pracovat v praktických cvičeních.

Program v jazyce C se vždy skládá z elementárních součástí, které rozdělujeme na klíčová slova, identifikátory, konstanty, operátory a interpunkční znaky.

Jazyk C je tzv. Case-sensitive, to znamená, že rozlišuje malá a velká písmena v názvech identifikátorů, proměnných, ale i klíčových slov. Klíčová slova mají speciální význam pro překladač C. Žádný identifikátor nemůže mít ve fázi překladu stejný název jako klíčové slovo. Ve zdrojovém kódu často používáme komentáře. Komentář je kompilátorem považován za oddělovací znaky a vypouští je. Základní typy dat dělíme na aritmetické datové typy, znaky a ukazatele. Aritmetické dále dělíme na celočíselné, které mohou být znaménkové nebo bez znaménka (signed, unsigned) a reálné. Pro znaky má jazyk C datový typ char, který reprezentuje znak z konkrétní znakové sady a je mu vyhrazen jeden byte. Ukazatel je speciální datový typ a představuje adresu paměťového místa.

V jazyce C musíme před prvním použitím proměnnou pojmenovat a určit její datový typ. Této nutné skutečnosti říkáme deklarace. Deklarací tedy určujeme typ objektu nezbytný pro překladač jazyka C. S pojmem deklarace souvisí pojem definice. Definicí určujeme hodnotu proměnné či tělo funkce. Pro matematické, logické a další výpočetní operace definujeme pojmy operand a operátor. Operátory takto rozdělíme na unární, které pracují s jedním operandem, binární pracující se dvěma operandy a specialitou jazyka C je podmíněný ternární operátor pracující se třemi operandy.

Standardním vstupem a výstupem chápeme klávesnici a monitor počítače. Je důležité ovládat funkce se standardním vstupem a výstupem. Jejich syntaxi si však ve shrnutí opakovat nebudeme.

Pokud chceme psát v jazyce C projekty, které řeší konkrétní úlohy z praxe, nevyhneme se nějakým řídicím strukturám. Těmi jsou příkazy, bloky, podmíněný příkaz, přepínač a cykly. Nedílnou součástí jazyka C jsou funkce. Funkce bychom mohli rozčlenit na tři skupiny: standardní, které jsou definovány normou jazyka, uživatelské, které jsme sami napsali ke konkrétním příkladům, a také nadstavbové. Každá funkce má svou definici a základní vlastnosti, je pojmenována a tímto jménem se volá v programu, může, ale nemusí mít parametry, jejichž prostřednictvím předáváme data do těla funkce, ve které jsou následně zpracovány, má návratovou hodnotu obsahující výsledky zpracování v těle funkce.

Rovněž důležitými podkapitolami jsou pole, práce s řetězci a také se vstupem a výstupem do souboru. Syntaxe jednotlivých příkazů a jejich vysvětlení do závěrečného shrnutí nepatří, zvláště v tomto případě, neboť tato 30 stránková kapitola je v podstatě hlavním shrnutím práce s jazykem C. Je nutné uvedené programové konstrukce ovládat, neboť je budeme potřebovat při tvorbě programů a projektů v jazyce C.

**Úkol k řešení 4.1**

Vytvořte program pro výpočet kořenů kvadratické rovnice.

**Úkol k řešení 4.2**

Vytvořte program který vytvoří na obrazovce čtverec z hvězdiček. Uživatel zadá stranu a . Vytvořte funkce pro výpočet obvodu a obsahu. Nejprve bude vykreslen čtverec, poté zobrazen jeho obsah a obvod.

**Úkol k řešení 4.3**

Vytvořte pole o 10 číslech a to seřadíte od nejmenšího po největší.

**Úkol k řešení 4.4**

Vytvořte program s funkcí malá násobilka. Parametrem bude násobené číslo.

**Úkol k řešení 4.5**

Načtěte znaky ze souboru Data.txt a vytiskněte je do sloupce.

**Úkol k řešení 4.6**

Vytvořte program, který spustíte s parametrem názvu výstupního souboru. Poté uživatel zadá libovolné číslo. Program vypíše na obrazovku a uloží do výstupního souboru všechny prvočísla od 1 do uživatelem zadaného čísla.

**Úkol k řešení 4.7**

Vytvořte textovou hru hádání čísel. Uživatel nejprve zvolí variantu z kolika čísel se hádá (100, 500, 1000). Poté začíná hra, PC vygeneruje náhodné číslo, a uživatel jej hádá, PC odpovídá pouze hlášením, zda hledané číslo je menší nebo větší. Po uhodnutí čísla se zobrazí počet pokusů a hodnocení. Po uhodnutí čísla nebo vyčerpání počtu pokusů má uživatel možnost hrát hru znovu nebo program ukončit, průběh hry je uložen do výstupního souboru.

**Úkol k řešení 4.8**

Vytvořte program pro převod dekadických čísel na binární a naopak. PC dává možnost uživateli zadat dekadické nebo binární číslo a pak jej převede na opačný formát. Např.: 0100110 je $0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6$.

**Kontrolní otázka 4.1**

Co je to ANSI C?

**Kontrolní otázka 4.2**

Uveďte postup vzniku programu od psaní zdroj. kódu až po spustitelnou verzi programu.

**Kontrolní otázka 4.3**

Jak dlouhý může být identifikátor v ANSI C?

**Kontrolní otázka 4.4**

Co znamená označení Case-sensitive?

**Kontrolní otázka 4.5**

Co je to ternární operátor?

**Kontrolní otázka 4.6**

Jaké druhy cyklů znáte?

**Kontrolní otázka 4.7**

Jakou funkci má preprocesor?

**Kontrolní otázka 4.8**

Uveďte základní vlastnosti funkce.

5. Architektura čipové sady



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- definovat základní pojmy v oblasti čipových sad
- popsat architekturu čipové sady
- zakreslit blokové schéma čipové sady
- podrobně popsat dlouholetý vývoj čipových sad
- definovat jednotlivé přenosové rychlosti mezi komponenty



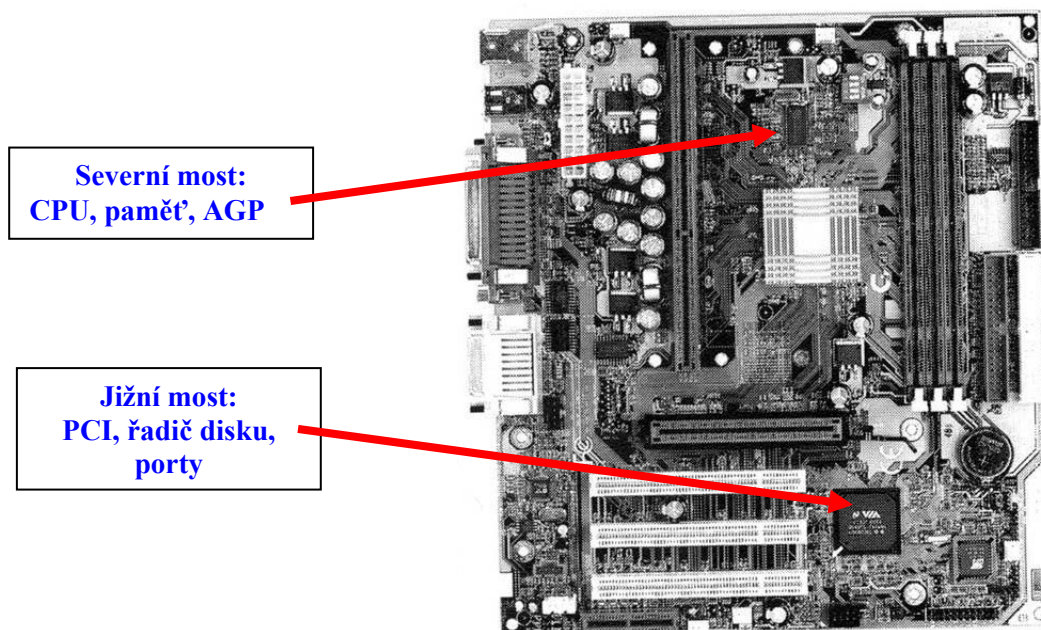
Výklad

Po třech kapitolách věnovaných jazyku C a vývojovému prostředí Visual Studio se opět vrátíme k hardwarovým komponentám osobního počítače. V denní formě studia je této problematice věnována zbývající přednášková část, přičemž na cvičení se dále procvičují příklady z jazyka C.

V úvodní kapitole jsme nastínili některé pojmy, jimž se budeme dále věnovat. Tato kapitola se zabývá čipovými sadami, následovat budou procesory, sběrnice, paměti, porty a další avizované komponenty.

5.1 Základní pojmy v oblastech čipových sad

Na základní desce každého počítače najdeme kromě patice pro procesor a grafickou kartu, sloty pro paměťové moduly a další součásti, hlavně čipovou sadu. Vlastnosti základní desky jsou z 90% právě dané čipovou sadou. Bez dlouhých úvodních slov si popíšeme její architekturu:



Obr. 5.1 Pohled na základní desku obsahující čipovou sadu



Pojem k zapamatování: architektura čipové sady

Čipová sada je soubor 2 čipů, které jsou mezi sebou spojeny sběrnici. Podle rozmístění jsou obvody pojmenovány Severní a Jižní most.

5.2 Vývoj čipové sady

Čipové sady procházejí vývojem srovnatelným s dalšími základními komponenty jako procesor či například grafický akcelerátor. Jsou nejdůležitější součástí základní desky, bez čipové sady by procesor nebyl schopen komunikovat s pamětí, ani s ostatními zařízeními.

U prvních PC vyráběných firmou IBM obsahovaly základní desky kromě procesoru mnoho dalších čipů, které z hlediska funkce představují čipovou sadu. Např.:

- generátor hodin,
- řadič sběrnice,
- systémový časovač,
- řadič přerušení,
- řadič DMA,
- řadič klávesnice, atd., celkem až 100 obvodů.

Všechny tyto obvody byly dodávány firmou Intel. Výroba základní desky tak byla velice nákladná. V roce 1986 firma *Chips and Technologies* vyvinula čip 82C206, který tvořil nejdůležitější část první čipové sady pro PC. Tento jediný obvod zajistil funkce generátoru hodin, řadiče sběrnice, systémového časovače, řadiče přerušení, řadiče DMA a paměti CMOS. Na základní desce byly ještě další obvody pro řadiče paměti a buffery. Počet čipů na základní desce klesl na 5. Výrazně klesla cena základních desek. Do roku 1994 byly jednotlivé čipy dodávány mnoha výrobci a opět si firma *INTEL* uvědomila příležitost upevnit svou pozici na trhu dodáváním čipových sad na míru svým procesorům a zahájila masovou výrobu vlastních čipových sad a ovládla trh. V roce 1996-2001 firma Intel přišla s podporou vývoje modulů RDRAM – výkonné paměti, avšak drahé, a rázem bylo výhradní postavení INTELU ohroženo. Paměti RDRAM a jejich podpora čipovými sadami se ukázalo jako nevhodné řešení. Nejen díky těmto chybám, ale i s upevňující se konkurencí na trhu s procesory, vstoupili na trh čipových sad další výrobci AMD, VIA Technologies, NVIDIA, SiS a další.

Přes tento fakt je firma Intel silným hráčem na tomto poli udávající směr vývoje čipsetů, v této kapitole se tedy budeme věnovat architektuřím čipových sad tohoto výrobce. Čipové sady jiných výrobců však mají obdobnou architekturu, takže to rozhodně nebude problém.



Pojem k zapamatování: FSB – Front Side Bus

Jednotlivé části jsou propojeny sběrnici. Sběrnice mezi CPU a severním mostem je FSB. Od této sběrnice se odvíjí frekvence CPU, AGP, paměti, samotných mostů čipsetu atd. Sběrnice FSB jsou taktovány 100, 133, 200, 266 MHz. U starých Celeronů to bylo ještě 66 MHz. Uvedené taktovací frekvence jsou znásobeny principem DDR – *double data rate*: dvojnásobná propustnost po sběrnici nebo QDR – *quad data rate*: čtyřnásobná propustnost 400, 533, 800, (1066 MHz).

Čipové sady se tedy po roce 1993 ubíraly nastoleným směrem, jež definoval sady založené na architektuře **North/South Bridge**. V domácí terminologii tuto architekturu nazýváme doslovným překladem: severní a jižní most. Tato terminologie je používána dodnes. Přestože současné čipové

sady jsou již navrženy architekturou rozbočovačů, termín severní a jižní most zůstal dále v podvědomí uživatelů. Čip *North Bridge* je nejdůležitější součást základní desky a jediný obvod pracující na plné rychlosti sběrnice procesoru. Čip *South Bridge* byl odvozen od úkolu propojit sběrnici PCI s pomalejší sběrnici ISA. Architektura využívá ještě dalšího čipu s názvem Super/IO čip, ten je oddělený a obvykle ho dodává jiný výrobce (např. *National Semiconductor*). Tento čip slouží k připojení běžných periférií k systému.

Vývojové řady čipových sad po roce 1993 byly určeny pro procesory páté generace, jednalo se tedy o procesory Pentium. Čipové sady pracovaly s FSB 66 MHz. Jižní most byl tvořen samostatným čipem, který nesl své typové pojmenování. To umožnilo použít určitý jižní most ve více typech čipových sad, což vneslo do praxe určitou kompatibilitu mezi sadami a také ušetření nákladů na vývoj a výrobu. Čipové sady se vyvíjely v souladu s vývojem ostatních technologií. Pokud tedy výrobci pamětí zavedou novou technologii, v tabulce například přechod z paměti FPM na EDO, dále pak na SDRAM, čipová sada na to musí být připravena.

V tabulce vidíme přehled čipových sad pro procesory páté generace. Již z prvního pohledu vidíme už zmíněné přechody typů pamětí. Čipová sada 430 FX se stala oblíbenou sadou kupujících, protože umožnila přechod na EDO RAM moduly, které byly zhruba o 20 procent rychlejší než předchozí FPM. Avšak přechod na sadu VX, jež podporovala SDRAM, již tak masový nebyl. Paměti byly v té době nejdražší komponentou PC, a tak se SDRAM začaly masově využívat až s následující generací procesorů. Tento fakt podpořily čipové sady páté generace, neboť umožňovaly paměti *L2 cache* odkazovat se na stále stejně velkou část paměti RAM, jako u předchozích modulů EDO.

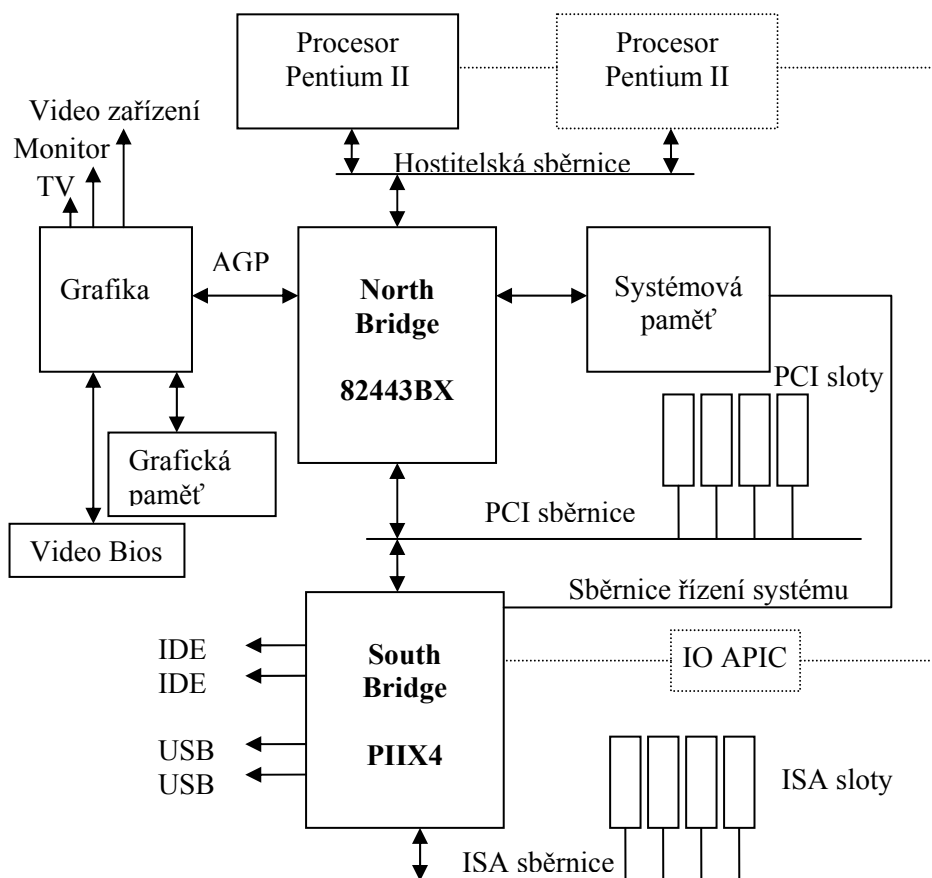
Čipové sady na architektuře mostů pro procesory páté generace							
Čipová sada	430 LX	430 NX	430FX	430MX	430HX	430VX	430TX
Uvedení	3/1993	3/1994	1/1995	10/1995	2/1996	2/1996	2/1997
Rychlost sběrnice	66 MHz	66 MHz	66 MHz	66 MHz	66 MHz	66 MHz	66 MHz
Procesor	P60/66	P75/133	P75/133	P75/133	P75/133	P75/133	P75/133
Podpora SMP	Ne	Ano	Ne	Ne	Ano	Ne	Ne
Typ pamětí	FPM	FPM	FPM/EDO	FPM/EDO	FPM/EDO	FPM/EDO/ SDRAM	FPM/EDO/ SDRAM
Parita/ ECC	Parita	Parita	Ne	Ne	Parita i ECC	Ne	Ne
Max. paměť	192 MB	512MB	128MB	128MB	512MB	128MB	256MB
Podpora PCI	2.0	2.0	2.0	2.0	2.1	2.1	2.1
Podpora AGP	Ne	Ne	Ne	Ne	Ne	Ne	Ne
South Bridge	SIO	SIO	PIIX	MPIIX	PIIX3	PIIX3	PIIX4

Tab. 5.1 Čipové sady na architektuře mostů pro procesory páté generace

Z tabulky dále vidíme možnou podporu více procesorů u dvou čipových sad, SMP znamená *Symmetric multiprocessing*, v tomto případě podporu dvou procesorů. Význam pojmenování jižních mostů je následující: **SIO** znamená *System Input Output* a **PIIX** vyjadřuje *PCI ISA IDE Xcelerator*.

S další generací procesorů přišla na svět celá řada novinek. Mezi nejdůležitější patřil přechod na 100 MHz FSB sběrnici a rozhraní AGP pro grafickou kartu. Intel měl v této době výhradní postavení na trhu s čipovými sadami, díky změnám slotů a socketů se načas stal jediným výrobcem čipových sad pro své procesory. Až po uvolnění a prodeji licencí se dostali na trh i jiní výrobci. Mezi nejpobulárnější čipové sady historie vůbec se všeobecně řadí čipová sada 440BX. Označení nese od názvu čipu severního mostu 82443BX. Tato čipová sada jako první podporovala FSB 100 MHz, procesor Pentium II pro notebooky a paměťové moduly SDRAM rovněž na 100 MHz. Tato čipová sada byla velice úspěšná, vývojáři hardwaru pro ni ladili své komponenty a tak s touto čipovou sadou

měli uživatelé jistotu stability systému a ovladačů pro různé hardwarové komponenty. Sada podporovala 1 GB paměti RAM ve 4 bankách (DIMM), což na dlouho dobu značně převyšovalo možnosti uživatelů a tak se čipová sada těšila v oblibě i procesorů Pentium III. Sada navíc obsahovala vylepšenou správu napájení i podporu pamět'ových modulů s kódem ECC.

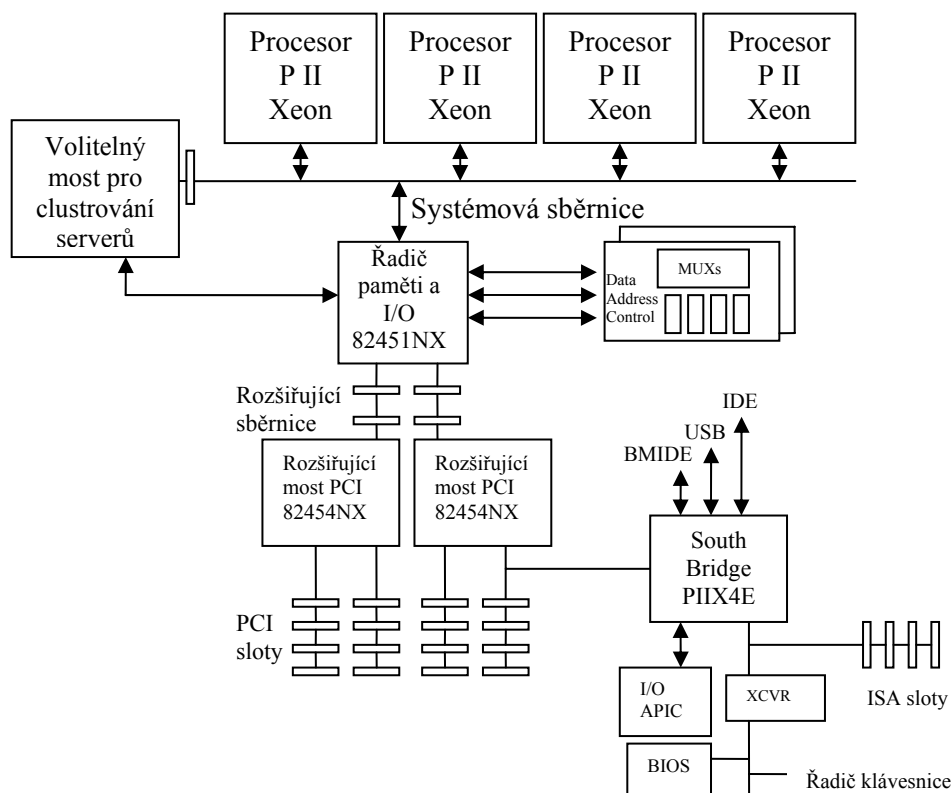


Obr. 5.2 Blokové schéma čipové sady Intel 440BX

Čipové sady na architektuře mostů pro procesory šesté generace							
Čipová sada	440 FX	440 LX	440 EX	440 BX	440 GX	450 NX	440 ZX
Uvedení	5/1996	8/1997	4/1998	4/1998	6/1998	6/1998	11/1998
Rychlost sběrnice	66 MHz	66 MHz	66 MHz	66/100 MHz	100 MHz	100 MHz	66/100 MHz
Procesor	P II	P II	Celeron	PII/III/ Celeron	PII/III/ Xeon	PII/III/ Xeon	PII/III/ Celeron
Podpora SMP	Ano	Ano	Ne	Ano	Ano	Ano (4)	Ne
Typ paměti	FPM/EDO/ BEDO	FPM/EDO/ SDRAM	EDO/SDRAM	SDRAM	SDRAM	FPM/EDO	SDRAM
Parita/ ECC	Parita i ECC	Parita i ECC	Ne	Parita i ECC	Parita i ECC	Parita i ECC	Ne
Max. paměť	1 GB	512 MB	256 MB	1 GB	2 GB	8 GB	256 MB
Podpora PCI	2.1	2.1	2.1	2.1	2.1	2.1	2.1
Podpora AGP	Ne	AGP 2x	AGP 2x	AGP 2x	AGP 2x	Ne	AGP 2x
South Bridge	PIIX3	PIIX4	PIIX4	PIIX4	PIIX4	PIIX4	PIIX4

Tab. 5.2 Čipové sady na architektuře mostů pro procesory šesté generace

V tabulce 5.2 máme přehledně znázorněné čipové sady pro procesory šesté generace. Podpora symetrického multiprocessingu byla zdokonalena na 4 procesory u čipové sady 450NX a tak je vhodné si tuto sadu rovněž představit a tím zakončit výklad o čipových sadách na základě architektury mostů. Čipová sada 450NX byla určena pro nasazení výkonných serverů PENTIUM II,III/XEON. Sada podporuje až 8 GB paměti RAM a čtyři PCI 32 bitové nebo dvě 64 bitové sběrnice. Nepodporuje grafický port AGP, což však u serverů není podstatné.



Obr. 5.3 Blokové schéma čipové sady pro serverové procesory Xeon

5.3 Čipové sady založené na architektuře rozbočovačů

Novější čipové sady jsou založené na rozbočovačích, přičemž North Bridge je nazýván rozbočovačem řadiče paměti *MCH* – *Memory Controller Hub* a čip South Bridge je označován pojmem rozbočovač řadiče vstupů a výstupů *ICH* – *I/O Controller Hub*.

Od roku 1999 si nároky na rychlosti procesorových sběrnic a spojů mezi jednotlivými mosty vyžádaly změnu technologie. Oba nejdůležitější čipy, tedy rozbočovače *MCH* a *ICH*, byly nyní propojeny sběrnici s nižší šířkou, tedy menším počtem vodičů, což mělo pozitivní vliv na zrychlení technologie s menším rušením, signálovým šumem a počtem směřování vodičů na základní desce.



Pojem k zapamatování: rozbočovače čipových sad

MCH - *Memory Controller Hub* (dříve Severní most) má na starost propojení čipové sady s procesorovou sběrnici a grafickou sběrnici. **ICH** - *I/O Controller Hub* (dříve Jižní most) tvoří rozhraní mezi rozbočovačem, IDE řadičem pevných disků a sběrnici PCI a USB.

Rok 1999 přinesl první čipovou sadu na architektuře rozbočovače. **Čipová sada 810**, zaznamenala **zásadní změnu ve vývoji čipových sad, které se od procesorů 486 v podstatě nezměnily**. Hlavním krédem Intelu bylo v této době podstatné zvýšení výkonu, levnější a jednodušší řešení. Charakteristika čipové sady je následující:

- podpora procesorové sběrnice 66/100/133 MHz,
- integrovaná grafika Intel 3D,
- volitelná instalace Cache pro zobrazování,
- výstupní port DVI,
- softwarová podpora přehrávání DVD, MPEG-2,
- podpora Ultra-Ata 66,
- integrovaný řadič AC97 (Audio Codec),
- podpora režimu SLEEP s nízkou spotřebou,
- integrovaný řadič USB,
- ISA není podporována.

Čipová sada se skládá ze tří částí:

- rozbočovač řadiče grafické paměti (obvod 82810) 421 pinů,
- rozbočovač řadiče vstupů a výstupů (82801) 241 pinů,
- rozbočovač firmwaru, 40 vývodů (dodáván výrobcům desek samostatně).

Co je tedy nového? Čipová sada se skládá ze tří komponent, stejně jako architektura North Bridge /South Bridge. Rozdíl je v 2x rychlejší komunikaci a absenci ISA sběrnice. Vývoj následovaly čipové sady 815 a 820. Intel přišel s možností výběru čipsetu podle toho, zdali si koncový zákazník přál využít integrovanou grafiku, či požit AGP port. Tento trend se využívá dodnes a výroba je tak ekonomičtější, uživatel může zvolit variantu šitou na míru svým potřebám a výrobce dodává stále stejné obvody ICH, MCH. Varianty sad tak dostaly do názvu za číselné označení ještě jedno písmeno charakterizující volbu grafického adaptéru.

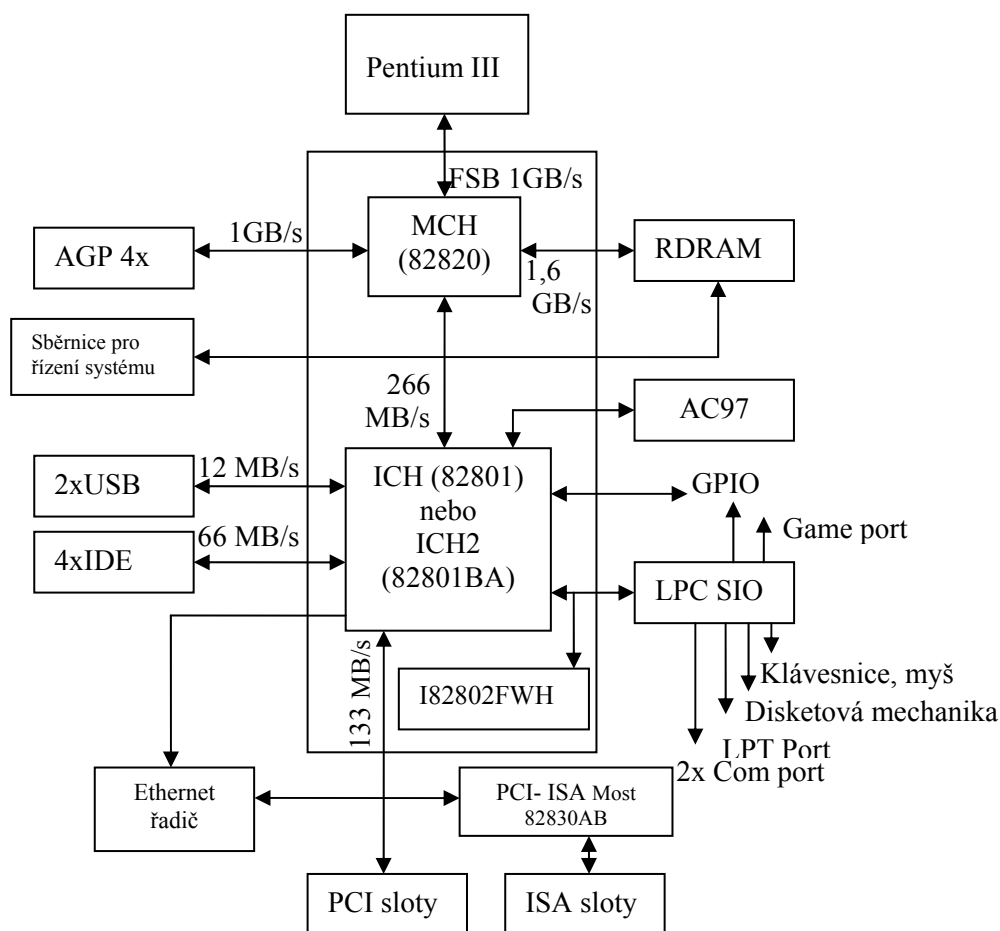
Čipová sada 820 jako první implementovala možnost nasazení výkonných, leč drahých paměťových modulů RDRAM. Situace na trhu se však vyvíjela poněkud jinak a tyto paměti byly drahé a tak se nesetkaly se zájmem uživatelů. Intel tuto situaci řešil přidáním **Memory Translator HUBu**, který umožňoval pro čipovou sadu 820 použití levnějších SDRAM. To se však stalo Intelu osudným problémem neboť čip byl chybný, docházelo k elektrickému rušení signálů, což mělo za následek neočekávaný restart PC. Intel tak musel vyměnit téměř milion základních desek s tímto čipsetem. Další čipová sada 850 tak opět umožňovala použití pouze paměťových modulů RDRAM, a tak majitelé prvních pentiů 4 neměli jinou volbu.

Abychom si ujasnili rozdíly čipové sady s architekturou rozbočovačů pro běžný počítač a pro server, uveďme si následující popis sady i840, určené pro servery, která obsahuje kromě standardních hubů (8240, 82801, 82802) navíc další čipy:

- rozbočovač řadiče 64bitové sběrnice PCI – 82806, který podporuje 64bitové sloty PCI na rychlosti 66MHz, výsledná PCI je 4x rychlejší než 32bitová verze na 33 MHz,
- rozbočovač opakovače paměti RDRAM 82803 pro zvýšení výkonu paměťového systému, čip umožňuje každý paměťový kanál rozdělit na dva,
- rozbočovač opakovače paměti SDRAM 82804 pro překlad protokolu RDRAM na signály pro moduly SDRAM,
- rozhraní pro síť a rozhraní RAID.

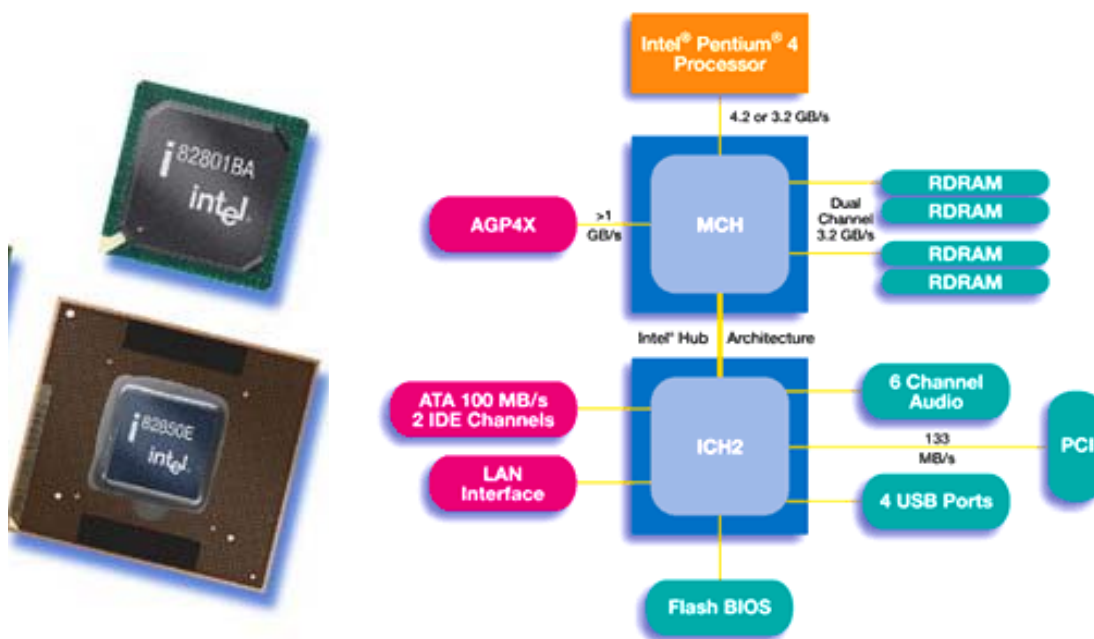
Čipové sady na architektuře rozbočovačů do roku 2000						
Čipová sada	810/810E	815	815E/EP	820/820E	840	850
Uvedení	4 - 9/1999	6/2000	6-11/2000	11/1999-6/2000	10/1999	11/2000
Rychlost sběrnice	66 - 100/133 MHz	66/100/133 MHz	66/100/133 MHz	66/100/133 MHz	66/100/133 MHz	400 MHz
Procesor	P II-III-Celeron	P II-III-Celeron	P II-III-Celeron	P II-III-Celeron	PII/III/Xeon	Pentium 4
Podpora SMP	Ne	Ne	Ne	Ano	Ano	Ano
Typ paměti	EDO/SDRAM PC100	SDRAM PC133	SDRAM PC133	RDRAM PC800	RDRAM PC800	RDRAM dual channel
Parita/ ECC	Ne	Ne	Ne	Parita i ECC	Parita i ECC	Parita i ECC
Max. paměť	512 MB	512 MB	512 MB	1 GB	4 GB	2 GB
Podpora AGP	Ne	AGP 4x	AGP 4x	AGP 4x	AGP 4x	AGP 4x
Integr. grafika	Ano	Ano	Ano/Ne	Ne	Ne	Ne
South Bridge	ICH	ICH	ICH2	ICH/ICH2	ICH	ICH2

Tab. 5.3 Čipové sady na architektuře rozbočovačů do roku 2000



Obr. 5.4 Blokové schéma čipové sady architektury rozbočovačů, konkrétně i820

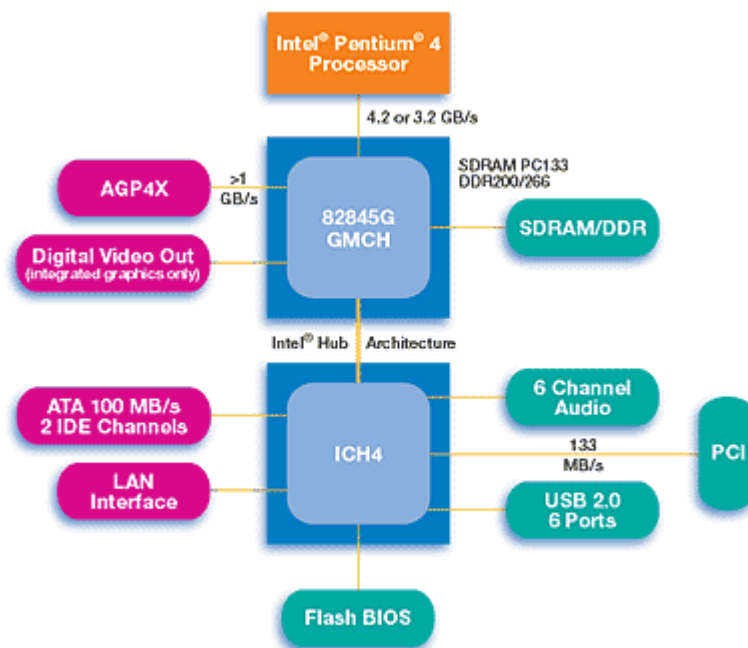
V roce 2000 byla představena **čipová sada 850**, která vydržela na trhu dva roky společně s pozdější revizí **850(E)**, jediná možná volba pro Pentium 4, neboť tehdy ještě neexistovala čipová sada Intel 845, či její pozdější modifikace (podpora DDR SDRAM, integrované grafické jádro, atd.). Čipová sada podporovala systémové sběrnice 400/533 MHz, obsahovala *Intel Hub Architecture*, což je spojení obou čipů sběrnic s průchodností 266 MB/s. Sada obsahuje duální paměťový řadič RDRAM s propustností 3,2 GB/s. Podporovala grafickou sběrnici AGP 4x, což byl tehdejší nadprůměr, ethernetový řadič, dva USB 1.1 řadiče o maximálně čtyřech portech, diskový řadič Ultra ATA/100 a zvukový řadič AC'97. Největším problémem je tedy podpora pouze paměťových modulů RDRAM.



Obr. 5.5 Blokové schéma čipové sady Intel 850

5.4 Současný vývoj čipových sad

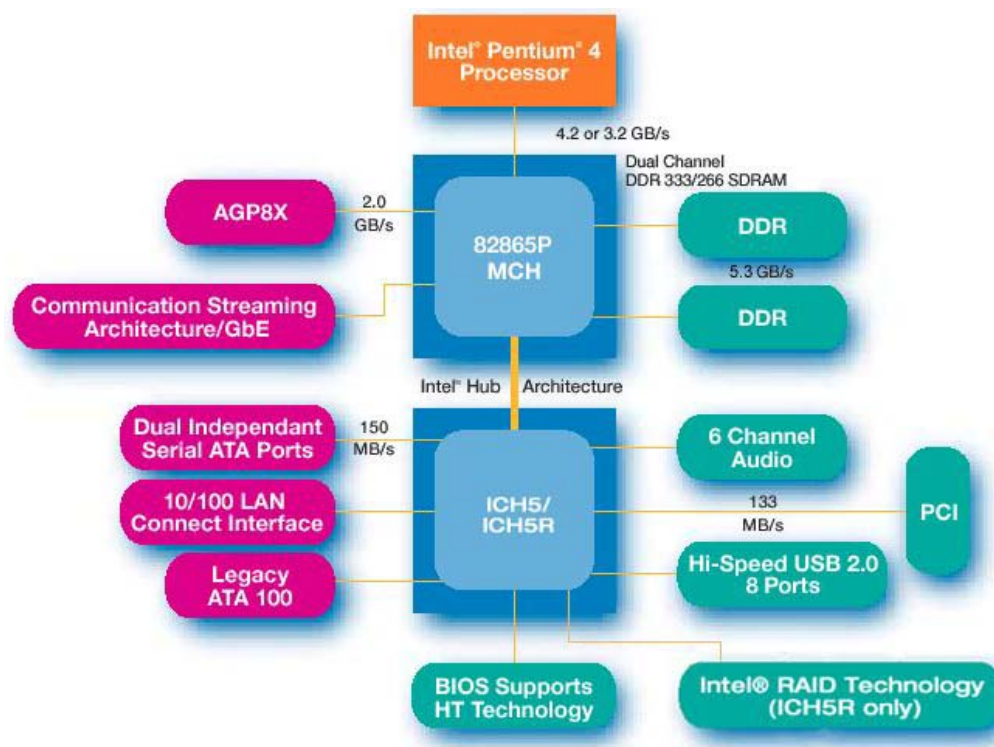
Zhruba dva roky se tedy čekalo na čipovou sadu, podporující paměťové moduly SDRAM. V roce 2002 byla představena **čipová sada i845**, a to hned ve třech verzích E, G, GL. Tyto čipové sady jsou určeny pro procesory Pentium 4 „Celeron“ na 1,7 GHz s 400 MHz FSB a Celeron 1,4 GHz na 100 MHz sběrnici. Čipset s označením GL obsahuje integrovaný grafický systém s podporou 32-bitových barev a vysokého rozlišení. Podporuje oba typy paměti – PC133 i DDR200/266 až po 2GB, nerozumí si ale s ECC. Používá SouthBridge ICH4 včetně šesti USB 2.0 portů a UltraATA100. Sada ve verzi G používá 533 MHz Quad Pumped Bus, AGP 4x a ATA100. Označení „E“ znamená „enhanced“ a značí podporu 533MHz, AGP 4x a pouze DDR200/266. Jak si můžeme všimnout čipové sady disponují ICH čipem ihned ve verzi ICH4. Nepřeskočili jsme jednu vývojovou verzi? Ano, vývoj sady trval poměrně dlouho a během vývoje čip ICH3 nepřinesl oproti ICH2 nic nového, pouze možnost využití šesti USB portů, a tak našel využití převážně v mobilní sféře notebooků, kde jde přeci jen vývoj trošičku pomaleji.



Obr. 5.6 Blokové schéma čipové sady Intel 845

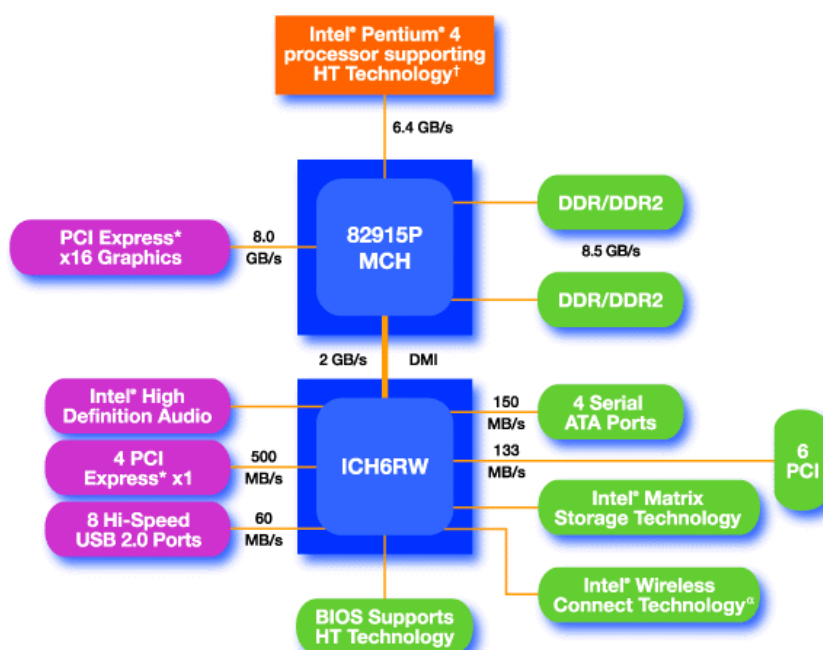
V roce 2003 si vývoj vyžádal novou čipovou sadu pro socket 478 a to **I865**. Taktika vyrábět čipové sady ve třech variantách se Intelu osvědčila, jednou technologií zahrnout co největší škálu požadavků zákazníků, kteří se odlišovali podle požadavků na integrovanou grafiku či port AGP. A tak i tato verze nesla tři označení (**P**, **PE**, **G**). Písmeno G tedy znamená integrovanou grafiku, verze P je pro změnu slabší s hodnotou FSB na 533 MHz. V této kapitole samozřejmě nestačíme popsat všechny čipové sady, není to ani cílem, a tak vynecháme serverové a mobilní čipsety. Čipová sada 875 se v podstatě od této čipové sady neliší, a tak vynecháme i tento čipset. Charakteristika i865 je tedy následující: Podpora platformy Socket 478 pro Pentium 4 / Celeron s jádry *Willamette* a *Northwood*. Frekvence sběrnice podporuje 400MHz, 533MHz a 800MHz (*Quad Pumped Bus*). Podporuje technologii Hyper-Threading, dvoukanálové paměti typu DDR266/333/400 do kapacity max. 4GB, bez kódu ECC, grafickou sběrnici AGP 3.0 (AGP 4x/8x). Ve verzi i865G obsahuje čipset integrované grafické jádro Intel Extreme Graphics. Zajímavostí je architektura CSA (*Communication Streaming Architecture*), čímž lze připojit kartu pro Gigabit Ethernet. Je již vybavena čipem ICH5, který podporuje 8 portů USB 2.0, 2 kanály pro ATA/100, 2 porty pro Serial ATA-150 a obsahuje integrované 6-kanálové audio.

V roce 2004 dochází k zásadním novinkám a to podpoře sběrnice PCI Express. Tuto sběrnici podporuje **čipová sada I915**. Novinkou je podpora paměťové technologie DDR2, nová je i patice Socket T (LGA775). Také čip ICH je v nové verzi **ICH6**. Verze P je bez grafického jádra, verze G obsahuje integrované grafické jádro Intel Graphics Media Accelerator 900. Jeho výkon se prakticky vyrovná Radeonu X300.



Obr. 5.7 Blokové schéma čipové sady Intel 865

Southbridge ICH6 podporuje celkem čtyři SATA porty, sběrnice PCI Express x1 (250 MB/s v každém směru), integruje podporu WLAN, takže již nemusíme instalovat klasické bezdrátové karty do slotů PCI, nová technologie Matrix RAID umožňuje využít výhod RAID1 a RAID0 na dvou discích, navíc přináší konečně NCQ a také hot-plug, což je technologie umožňující připojení a odpojení disku za chodu PC. Integrovaný zvuk prošel také vývojem, takže máme možnost využít Intel High Definition Audio (7.1 kanálů, Dolby Digital a 192 kHz).

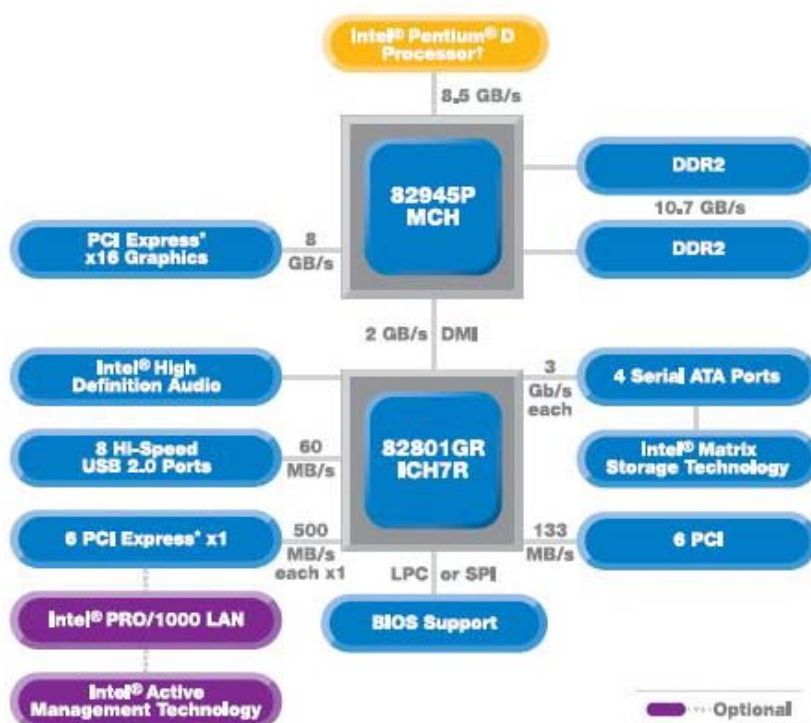


Obr. 5.8 Blokové schéma čipové sady Intel 915

Téhož roku se náročnější uživatelé dočkali **Čipsetu 1925**. Je určen pro ty, kteří již nechtějí využívat DDR paměti a PCI Express je pro ně samozřejmostí. Procesorová sběrnice i nadále pracuje na 800 MHz, pracuje tak s přenosovou rychlostí 6,4 GB/s. Nová je podpora technologie EM64T, což je podpora 64-bitových procesorů. Sběrnice PCI Express x16 zvyšuje přenosovou rychlost až na 4 GB/s a to v obou směrech (na schématu tak často najdete 8 GB/s). Je ohlášeno zvýšení přenosové rychlosti sběrnice PCI Express až na x32 s přenosovou rychlostí 8 GB/s. Blokové schéma se tedy mnoho neliší od předchozího čipsetu.

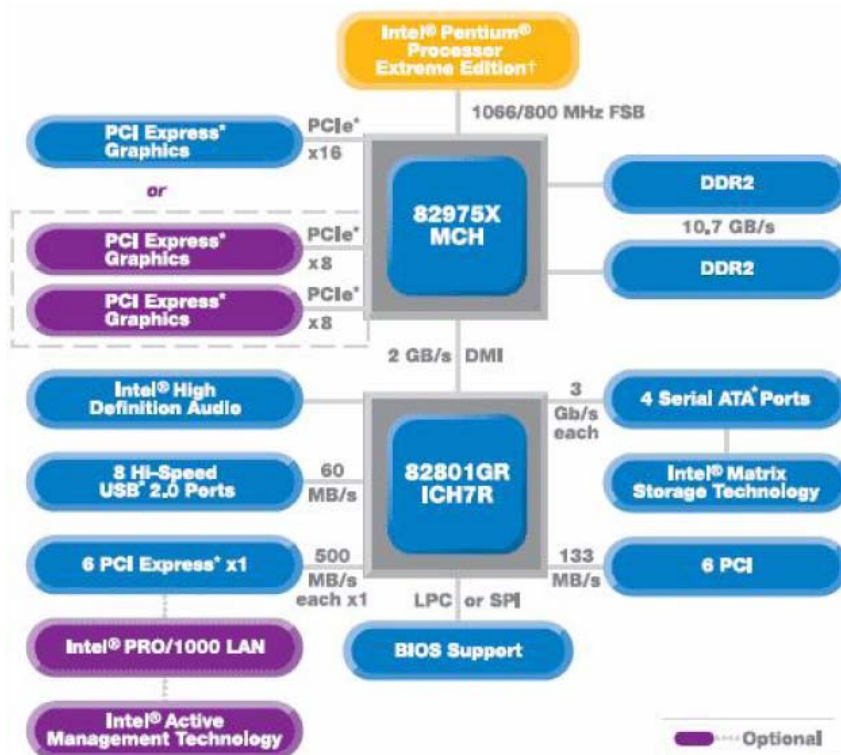
Rok **2005** je datem uvedení **čipové sady 1945**. Ve své době a následujících let až do současnosti (2007) **hovoříme o** sadách Intel 945G a Intel 945P jako o main-streamových čipsetech, neboli čipových sadách hlavního proudu. Nahradily tak úspěšnou řadu Intel 915, i když i tato sada stále osazuje některé v současnosti prodávané základní desky. Oproti nim se vyznačují podporou procesorů Pentium D (dual core), případně novým grafickým jádrem Intel *Graphics Media Accelerator 950* a také novým southbridge čipem **ICH7R**. Mezi podporované systémové sběrnice patří všechny dnes aktuální frekvence, tedy 533, 800 a 1066 MHz. Nejste tedy omezeni v možnostech použití jednotlivých procesorů – což je velká výhoda mainstreamových čipsetů. S nejlevnější čipovou sadou nemůžete použít high-endový procesor protože není podporována jeho sběrnice a s tou nejdražší zase nemáte možnost použít levné Celerony. Spojení mezi northbridge čipem a procesorem může dosahovat datové propustnosti až 8,5 GB/s při nasazení procesoru pracujícího se sběrnicí 1066 MHz. Samozřejmostí je zde podpora pamětí DDR2, které mohou pracovat s teoretickou propustností až 10,7 GB/s (DDR2 667) či 8,5 GB/s (DDR2 533). Paměti DDR400 již podporovány nejsou. Spojení mezi oběma čipy z čipové sady realizuje sběrnice o propustnosti až 2 GB/s. Maximum paměti je zde 4 GB a čipset již podporuje 64-bitové adresování.

Southbridge ICH7R obsahuje čtyři porty sběrnice Serial ATA s max. teoretickou propustností 3 Gb/s. Přestože sběrnice Ultra ATA/100 již v popisu není uvedena, na všech deskách od Intelu je použit jeden konektor IDE, samozřejmostí zůstává podpora technologie Intel Matrix RAID. Sběrnice PCI obsahuje až 6 slotů (133 MB/s), můžete také využít až šest slotů PCI x1 (250 MB/s).



Obr. 5.9 Blokové schéma čipové sady Intel 945

V našem popisu se blížíme k úplnému závěru. V roce **2006** byly uvedeny zatím poslední typy čipových sad. **Čipová sada I975** původně měla nést označení i955D jako DualGraphic, protože obsahuje podporu dvou grafických karet. Její hlavní charakteristikou je podpora nových dvoujádrových procesorů Pentium Extreme Edition vyráběných 65nm technologií, systémová sběrnice 800 nebo 1066 MHz, stále však obsahuje hub ICH 7.



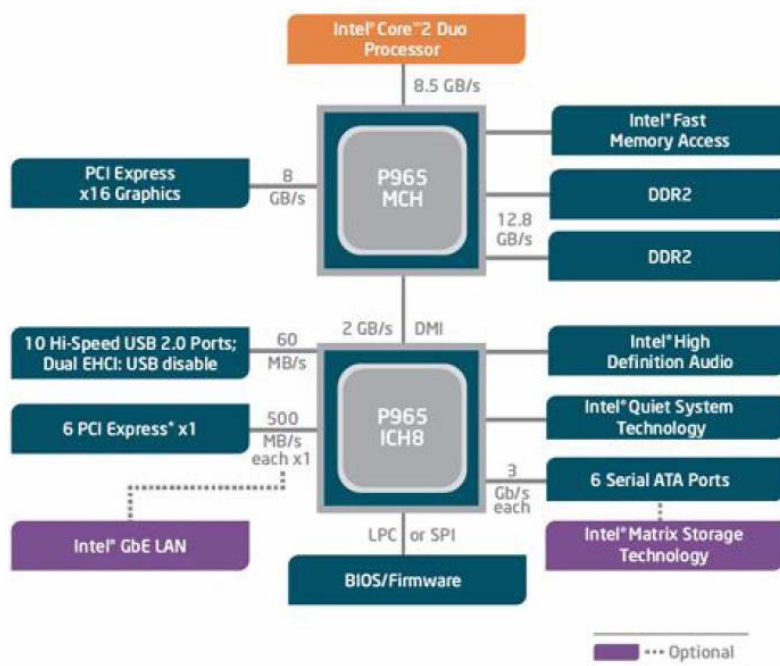
Obr. 5.10 Blokové schéma čipové sady Intel 975

V druhé polovině roku 2006 byla uvedena na trh **čipová sada I965**. I když přišla na trh později než i975, je zařazena vývojově pod tuto sadu, protože systémová sběrnice FSB podporuje mimo 800 a 1066 MHz i slabších 533 MHz. Podpora Intel Core2 Duo je již samozřejmostí, dále je také obsažena technologie *Intel Flex Memory* pro možnost zachování dvoukanálového režimu i při osazení pamětí různých velikostí.

Poslední čipovou sadou v našem vývojovém přehledu (píše se rok 2007) je Čipset i965. Je firmou Intel označen jako *mainstream* pro rok 2007. Podporuje totiž procesory Intel Core™2 Duo, Intel Pentium D řady 900, Pentium 4 s HT a také všechny ostatní procesory v pouzdru LGA775 s FSB 1066, 800 a 533 MHz. Technologie Intel Quiet System Technology disponuje zdokonaleným algoritmem plynulého řízení rychlosti ventilátorů, zajišťuje tišší chlazení a tím i celkově nízkou hlučnost systému. Čipová sada také podporuje veškeré technologie známé z předchozích čipových sad Intelu: *High Definition Audio*, volitelně *Intel Matrix Storage Technology (HW RAID kategorie 0, 1, 5 či 10)* atd. Novinkou je však hub **ICH8**, ten přináší ve srovnání s ICH7 tyto změny:

1. Počet portů USB 2.0 je zvýšen na 10.
2. Můstek ICH8 obsahuje řadič kompatibilní s AHCI – *Advanced Host Controller Interface* umožňuje snazší rozšiřitelnost systému o další disky SATA.
3. Čip obsahuje přímo integrovaný Gigabit LAN.
4. Šest SATA 3.0 Gbps portů obsahuje podporu externích disků.

5. ICH8 již neobsahuje ani jeden paralelní port IDE/ATA, což komplikuje situaci výrobcům základních desek, kteří pokud budou chtít zákazníkům vyjít vstříc, budou muset na základní desku podporu portu přidat.



Obr. 5.11 Blokové schéma čipové sady Intel 965



Další zdroje

1. Minasi, M. *Velký Průvodce Hardwarem*, Praha, Grada 2003, počet stran: 768. ISBN 80-247-0273-8.
2. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
3. Intel Chipsets [online] dostupný z <http://developer.intel.com/products/chipsets/index.htm>.
4. Motherboard [online] dostupný z <http://www.nvidia.com/page/mobo.html>
5. Motherboards [online] dostupný z <http://ati.amd.com/products/integrated.html>
6. Čipové sady, Svět Hardware [online] dostupný z <http://www.svethardware.cz/branch.jsp?name=Čipové+sady>



Shrnutí kapitoly

Na základní desce každého počítače najdeme kromě patice pro procesor a grafickou kartu, sloty pro paměťové moduly a další součásti, hlavně čipovou sadu. Vlastnosti základní desky jsou z 90% právě dané čipovou sadou. Čipová sada je soubor 2 čipů, které jsou mezi sebou spojeny sběrnici. Podle rozmístění jsou obvody pojmenovány severní a jižní most. U prvních PC vyráběných firmou IBM obsahovaly základní desky kromě procesoru mnoho dalších čipů, které z hlediska funkce představují čipovou sadu. Např. generátor hodin, řadič sběrnice, systémový časovač, řadič přerušení, řadič DMA, řadič klávesnice, atd.- celkem až 100 obvodů.

Čipové sady se tedy po roce 1993 ubíraly nastoleným směrem, jenž definoval sady založené na architektuře North/South Bridge. V domácí terminologii tuto architekturu nazýváme doslovným překladem: severní a jižní most. Tato terminologie je používána dodnes. Přestože současné čipové sady jsou již navrženy architekturou rozbočovačů, termín severní a jižní most zůstal dále v podvědomí uživatelů. Čip *North Bridge* je nejdůležitější součástí základní desky a jediný obvod pracující na plné rychlosti sběrnice procesoru. Čip *South Bridge* byl odvozen od úkolu propojit sběrnice PCI s pomalejší sběrnici ISA. Architektura využívá ještě dalšího čipu s názvem Super/IO čip, ten je oddělený a obvykle ho dodává jiný výrobce (např. *National Semiconductor*). Slouží k připojení běžných periférií k systému.

Jednotlivé části jsou propojeny sběrnicemi. Sběrnice mezi CPU a severním mostem je FSB. Od této sběrnice se odvíjí frekvence CPU, AGP, pamětí, samotných mostů čipsetu atd. Sběrnice FSB jsou taktovány 100, 133, 200, 266 MHz. U starých Celeronů to bylo ještě 66 MHz. Uvedené taktovací frekvence jsou znásobeny principem DDR – *double data rate*: dvojnásobná propustnost po sběrnici nebo QDR – *quad data rate*: čtyřnásobná propustnost 400, 533, 800, (1066 MHz).

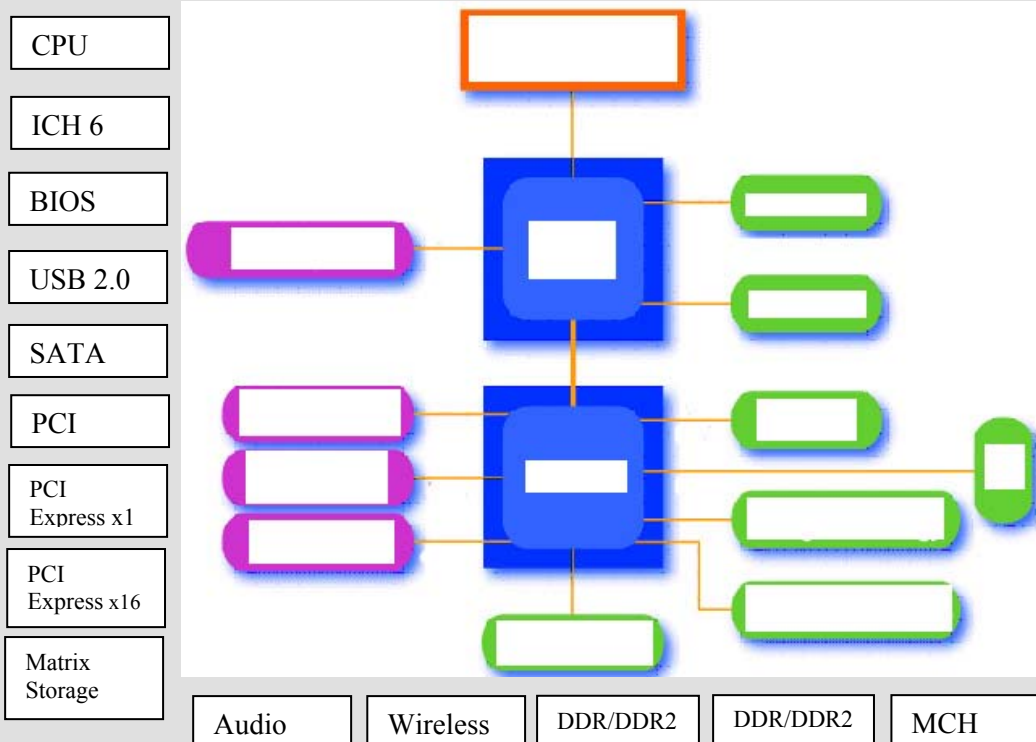
Novější čipové sady jsou založené na rozbočovačích, přičemž North Bridge je nazýván rozbočovačem řadiče paměti *MCH* – *Memory Controller Hub* a čip South Bridge je označován pojmem rozbočovač řadiče vstupů a výstupů *ICH* – *I/O Controller Hub*.

Čipové sady jsou neustále zdokonalovány, neboť musí obstarat bezchybný chod sběrnic a samotných periférií, jejichž vývoj jde neustále dopředu. Proto můžeme sledovat na jednotlivých blokových schématech neustálá vylepšení architektur, což je například vidět v propustnosti paměťové sběrnice či na druhé straně vývojem čipu ICH od počátku až po zatím poslední ICH8.



Úkol k řešení 5.1

Doplňte do rámečků příslušné pojmy dle architektury čipové sady.



**Úkol k řešení 5.2**

Do obrázku z minulého příkladu doplňte následující propustnosti sběrnic.

150 MB/s

8 GB/s

8,5 GB/s

60 MB/s

6,4 GB/s

500 MB/s

2 GB/s

133 MB/s

**Kontrolní otázka 5.1**

Co je to FSB?

**Kontrolní otázka 5.2**

Popište vývoj čipové sady.

**Kontrolní otázka 5.3**

Popište architekturu čipové sady.

**Kontrolní otázka 5.4**

Definujte pojmy DDR, QDR, AGP, FSB, SMP, MCH, ICH, DVI, ECC.

**Kontrolní otázka 5.5**

Jakým způsobem se rozhodnete při výběru základní desky?

**Kontrolní otázka 5.6**

Zakreslete blokové schéma čipové sady.

6. Procesory I



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- popsat historický vývoj procesorů
- definovat sběrnice procesorů
- popsat hierarchicky instrukční sady procesorů
- popsat tovární výrobu procesoru
- vyjmenovat a charakterizovat pouzdra procesorů a zdůvodnit jejich tvar
- charakterizovat základní pojmy v oblasti chlazení procesoru



Výklad

6.1 Stručná historie

Za tvůrce prvního mikroprocesoru je uznána firma INTEL, která v roce 1971 uvedla na trh první čip nazvaný 4004. Vedoucí postavení neměl vždy INTEL, na konci 70 let byly do PC nejčastěji instalovány procesory Z-80 firmy Zilog a 6502 firmy MOS Technologies. Procesor Z-80 byla vylepšená verze Intelu 8080, využíval tehdy operačního systému CP/M. Procesor s číselným označením 6502 byl používán při výrobě prvních počítačů Apple. V roce 1978 uvedl Intel svůj první procesor na architektuře x86. Jednalo se o procesor 8086. V roce 1981 firma IBM uvedla na trh své první PC, používající procesor INTEL 8088, pracující na frekvenci 4,77 MHz s operačním systémem MSDOS v. 1.0. Tento procesor měl 16 bitové registry a 8 bitovou vnější sběrnici a byl v podstatě průkopníkem procesorů v osobních počítačích PC. Nástupci tohoto procesoru jsou procesory zmíněné architektury x86, jedná se o 80188, 80186, 80286, 80386 a 80486. S posledně jmenovanými procesory jsme se mohli setkat ještě v nedávné minulosti. V současné době uživatelé mají v podvědomí různé verze procesoru Pentium a mnoho z nich upřednostňuje konkurenční firmu AMD se svými verzemi procesorů Athlon. Avšak vývojovým řadám procesorů Intel a AMD se budeme věnovat později.



Obr. 6.1 Procesor Intel 8088

6.2 Sběrnice procesorů

Procesory mohou být posuzovány podle dvou hlavních kritérií: podle šířky sběrnic a podle jejich rychlosti. **Pojem šířka sběrnice** se používá k popisu tří různých parametrů CPU:

- šířka vnitřních registrů,
- šířka vstupních a výstupních sběrnic pro data,

- šířka adresové sběrnice.

Datová sběrnice

Pravděpodobně nejdůležitějšími kritérii pro popis procesorů jsou rychlost a šířka datové sběrnice. Je tvořena skupinou vodičů (vývodů) pro přenos dat, čím více signálů může být po sběrnici během časové jednotky přeneseno, tím je sběrnice (širší) rychlejší.



Pojem k zapamatování: Datová a adresová sběrnice

Šířka datové sběrnice určuje počet datových bitů, které mohou být v průběhu jednoho cyklu přesunuty z/do procesoru.

Šířka adresové sběrnice určuje maximální velikost RAM, kterou je schopen procesor využít.

V průběhu časového intervalu vodič sběrnice přenáší buď bit 1 hodnotou napětí (5 V nebo 3,3 V a podobně, v dnešní době již okolo 1,5 V) nebo bit 0 hodnotou napětí 0 V.

Důsledkem plynoucím z šířky datové sběrnice je definice velikosti banky paměti. Procesor se sběrnici o šířce 64 bitů je schopen těchto 64 bitů číst najednou, což například znamená, že 32bitové moduly SIMM musely být instalovány po dvou. U 64bitových pamětí tvoří jednu banku paměti jeden modul.

Adresová sběrnice

Lze ji definovat jako sadu vodičů přenášejících informace o těch adresách v paměti, do kterých budou data ukládána nebo z kterých budou čtena. Stejně jako u datové sběrnice, každý vodič přenáší 1 bit informace. Tento 1 bit je pak číslicí, tvořící údaj o adrese v paměti. Čím více tedy má adresová sběrnice vodičů, tím více adres může být využito. Tzn., že šířka adresové sběrnice určuje maximální velikost RAM, kterou je schopen procesor využít.

2 vodiče $\rightarrow 2^2$ jedinečných adres = 4,

3 vodiče $\rightarrow 2^3$ jedinečných adres = 8,

Tzn.: procesory s 32bitovou adresovou sběrnici adresují paměť o velikosti $2^{32} = 4$ GB.

Vnitřní registry

Velikost vnitřních registrů určuje, kolik informací je procesor schopen zpracovat během jednoho cyklu a jakým způsobem jsou data v procesoru přesouvána. Z tohoto důvodu jsou registry nazývány vnitřní datovou sběrnici.

Registr si můžeme představit jako paměťovou buňku uvnitř procesoru. Velikost vnitřních registrů určuje typ softwaru, příkazů a instrukcí, které je procesor schopen provádět. Procesor s 32bitovými vnitřními registry provádí 32bitové příkazy, zpracovává data po 32 bitech.

Některé procesory mají větší vnitřní registry než šířku vnější datové sběrnice. Takový procesor pak musí naplňovat vnitřní registry ve více cyklech a teprve poté data zpracuje. Příkladem jsou staré procesory 386SX, mající datovou sběrnici o šířce 16 bitů a vnitřní registry o velikosti 32 bitů. Opačným příkladem jsou procesory Pentium, mající 64 bitovou datovou sběrnici a pouze 32bitové registry. Řešením jsou dvě 32bitové větve pro zpracování instrukcí. Architektura CPU obsahující více větví pro zpracování instrukcí se obecně nazývá **superskalární**. Další generace procesorů mají až 6 vnitřních větví pro zpracování instrukcí. I když některé z těchto větví jsou vyhrazeny pro zpracování instrukcí pouze určitého typu, jsou tyto procesory schopny zpracovávat až tři instrukce během jednoho instrukčního cyklu. Všechny 32bitové procesory jsou schopné pracovat v několika různých režimech. Režim procesoru určuje především způsob práce s pamětí.

- Reálný
- Chráněný
- Virtuální reálný

Reálný režim je jednoúlohový – běží pouze jeden program. Neexistuje žádný ochranný mechanismus, který by nějaké aplikaci zabránil v přepsání oblasti paměti využívané jiným programem, či dokonce OS. Chráněný režim lze nazvat 32bitovým režimem v 32bitovém operačním systému. Začalo být plně podporováno s příchodem operačního systému Windows NT, Unix či OS2 a jejich vývojových verzích. Rovněž v případě 64bitového operačního systému - 64bitové verzi Windows XP a Vista hovoříme o chráněném režimu. Virtuální reálný režim je v podstatě spuštění 16bitové aplikace např. příkazové řádky v 32bitovém chráněném režimu.

6.3 Rychlost procesorů a základní desky



Pojem k zapamatování: Základní frekvence - Base frequency, latency

Frekvence generovaná frekvenčním generátorem. Slouží jako základ pro získávání dalších frekvencí.

Latence - doba, která uplyne od vyvolání požadavku do jeho splnění (reakční doba). Udává se běžně v nanosekundách (ns) nebo v hodinových cyklech (např. 4T jsou čtyři cykly). Čím nižší hodnota, tím lépe.

Do roku 1998 pracovaly všechny základní desky a procesory na stejné frekvenci FSB 66MHz. Od roku 1998 se procesory začaly taktovat na určitý násobek rychlosti základní desky, zároveň se FSB zrychlila na 100 MHz a o rok později na 133 MHz. V následujících letech 2000 a dále se rychlost procesorových sběrnic neustále zvyšuje, dochází ke zvyšování frekvencí FSB tzv. *Dual* nebo *Quad Pumped Bus* přenosem.



Pojem k zapamatování: Quad Pumped Bus přenos

Tento termín znamená, že jsou během jednoho cyklu po sběrnici přeneseny čtyři signály, to znamená 4 násobné zvýšení rychlosti. Poté jednoduše stanovíme frekvence FSB:

$$4 \times 100 \text{ MHz} = 400 \text{ MHz}$$

$$4 \times 133 \text{ MHz} = 533 \text{ MHz}$$

$$4 \times 200 \text{ MHz} = 800 \text{ MHz}$$

Z obrázku připomínající spíše tabulku můžeme vypočítat několik skutečností. Za povšimnutí stojí především extrémní hodnoty násobku frekvence procesoru u procesorů Intel. Z tohoto důvodu Intel zavedl *Quad Pumped Bus FSB*, ve kterých dosahoval povětšinou dvojnásobné hodnoty oproti konkurenčnímu AMD. Procesory AMD však dosahovaly lepších výkonů na stejných frekvencích oproti procesorům Intel. Intel v té době zcela jasně vyhrával v honbě za překonávání co nejvyšších frekvencí a společnost AMD v tomto nemohla konkurovat. Zvolila proto strategii P-Ratingu, protože potřebovala dát uživatelům najevo, že jsou na trhu procesory, které jsou stejně výkonné jako Intel, přestože jsou taktovány na nižších frekvencích.

Je zbytečné tabulku protáhnout až do současnosti, neboť honba za dosažením nejvyšší frekvence byla v roce 2006 ukončena. V dnešní době bychom podle původních smělých plánů Intelu pracovali s procesory na frekvenci 10 GHz, avšak technologie narazila na své velké omezení a tím je obrovský tepelný výkon. A proto Pentium 4 s jádrem *Prescott* s označením 670 běžící na frekvenci 3,8 GHz je posledním procesorem generace Pentia 4 na architektuře *NetBurst*, která bude zmíněna dále.

P- Rating procesorů AMD nastartoval Athlon XP

Frekvence CPU (MHz) →

Frekvence základní desky (MHz) →

Násobek rychlosti →

Intel				AMD				
P III/Celeron	1200	100	12x	Athlon/Duron	850	850	200	4,25x
P III/Celeron	1300	100	13x	Athlon/Duron	900	900	200	4,5x
P III/Celeron	1400	100	14x	Athlon/Duron	950	950	200	4,75x
Pentium III	1000	133	7,5x	Athlon/Duron	1000	1000	200	5x
Pentium III	1133	133	8,5x	Athlon/Duron	1100	1100	200	5,5x
Pentium III	1200	133	9x	Athlon/Duron	1200	1200	200	6x
Pentium III	1266	133	9,5x	Athlon/Duron	1300	1300	200	6,5x
Pentium III	1333	133	10x	Athlon/Duron	1400	1400	200	7x
Pentium III	1400	133	10,5x	Athlon	1000	1000	266	3,75x
Pentium 4	1300	400	3,25x	Athlon	1133	1133	266	4,25x
Pentium 4	1400	400	3,75x	Athlon	1200	1200	266	4,5x
Pentium 4	1500	400	3,75x	Athlon	1333	1333	266	5x
Pentium 4	1600	400	4x	Athlon	1400	1400	266	5,25x
Pentium 4	1700	400	4,25x	Athlon XP	1500+	1333	266	5x
266Pentium	1800	400	4,5x	Athlon XP	1600+	1400	266	5,25x
Pentium 4	1900	400	4,75x	Athlon XP	1700+	1466	266	5,5x
Pentium 4	2000	400	5x	Athlon XP	1800+	1533	266	5,75x
Pentium 4	2200	400	5,5x	Athlon XP	1900+	1600	266	6x
Pentium 4	2400	400	6x	Athlon XP	2000+	1666	266	6,25x
Pentium 4	2400	533	4,5x	Athlon XP	2100+	1733	266	6,5x
Pentium 4	2533	533	4,75	Athlon XP	2200+	1800	266	6,75x



Skutečná
frekvence



Extrémní hodnoty
násobku



P Rating označující výkon
CPU u konkurence

Obr. 6.2 P-rating procesorů AMD označoval vyšší výkon procesoru AMD Athlon XP
oproti konkurenčnímu Pentiu 4 na nižší frekvenci

Paměti L1 a L2 Cache

Počínaje řadou 486 všechny CPU obsahují integrovanou L1 Cache a její řadič. Velikost závisí na typu procesoru od původních 8 kB až po 128 kB. Cache je integrována do procesoru, takže pracuje na plné rychlosti jádra. Samotnou L1 si můžeme představit jako poměrně malou oblast v procesoru, obsahující velmi rychlou paměť. Tato paměť se používá pro ukládání právě využívané či potřebné sadě dat a kódu. Pentium III běžící na 1,4 GHz tedy na 10,5 násobku rychlosti zákl. desky, která je 133 MHz. Systémová paměť tedy běží na rychlosti zákl. desky. Cache je jediná paměť schopná pracovat na rychlosti CPU, u PIII má velikost 32 kB, dva bloky po 16 kB. Jsou-li data, která CPU potřebuje uložena v L1, nemusí procesor čekat a rovnou je načte. Pokud nejsou v L1, pokusí se je získat z L2, pak z RAM. Rychlost CPU na 233 MHz odpovídá cyklu 4,3 ns, CPU je zasunut v desce 66 MHz, což odpovídá 15 ns. Paměťové moduly v tomto případě taktované 16 MHz odpovídají cyklu 60 ns. Chybné cachování znamená, že řadič cache nesprávně naplnil cache takovými daty, které procesor v dalších krocích nepotřebuje. Výsledkem je zpomalení práce procesoru. Zmenšení výkonu celého systému je v tomto případě evidentní. Firma Intel ve svých specifikacích uvádí více než 90% správné cachování.

Aby se zmenšilo výrazné zpomalení systému po chybném cachování, může být do systému přidána L2 cache. Podobně jako L1 tak i L2 se treťi v 90% případů, takže čtení ze systémové paměti se snižuje na 1%. Cache se označuje někdy také termínem čtyřcestná (osmi) asociativní paměť. Toto označení znamená rozdělení paměti cache do 4 (8) bloků, z nichž každý má 256 řádek po 16 bytech. Každý z rozdělených bloků obsahuje data z různých částí systémové paměti. Obsah cache musí být vždy synchronizován s RAM. Technika je označovaná jako write-through nebo write-back. Řadič sběrnice zodpovídá za sledování sběrnice a musí zajistit opětovné načtení paměťového místa.



Pojem k zapamatování: HT–HyperTransport a HyperThreading jsou odlišné pojmy

HyperTransport je označení sériové sběrnice použité u procesorů Athlon 64 a z něj odvozených modelů a dále také u některých čipových sad. Původně byla vyvíjena pod názvem *Lightning Data Transport* (LDT). U procesorů Athlon 64 se frekvence *HyperTransportu* získává násobením *base frequency* příslušným HT násobičem - takže například 1 GHz se získá jako 5 krát 200 MHz. Toto násobení provádí procesor podle příslušného nastavení. Termín je tedy zcela odlišný od Intelem zavedeného HyperThreadingu.

HyperThreading je technologie, kdy se jedno jádro procesoru tváří jako dvě jádra. Díky tomu je možné, aby zpracovávalo dvě programová vlákna zároveň. To umožňuje zlepšit využití výpočetních částí jádra, což přináší vyšší výkon. Na druhou stranu však je zapotřebí zpracovávat dobře naprogramované vícevláknové aplikace. Bohužel však může docházet k tomu, že u důležitého programu může výkon se zapnutým HyperThreadingem poklesnout ve prospěch programu nedůležitého z důvodu ne zcela vyřešené prioritizace vláken. Dalo by se říci, že technologie HyperThreading je předvojem dvoujádrových procesorů a tak ji v současné době nemá cenu příliš do detailu rozebírat.

6.4 Instrukční sada

Instrukční sada je soubor instrukcí, kterým procesor rozumí a které byly vytvořeny s cílem provádět určité typy výpočtů. Pokud procesor instrukční sadu nezná, dojde při pokusu o vykonání instrukce k chybě. Základní celočíselnou sadou procesorů pro počítače PC je instrukční sada označená jako **x86**. Všechny procesory umí s touto sadou pracovat, neboť je to instrukční sada pro práci pouze s celými čísly, kterou využívaly první procesory. Rozšíření funkčnosti procesorů o práci s plovoucí desetinnou čárkou (Floating point - FP) měla na starost sada **x87**. Původně byla sada určená pro matematické koprocesory. Starší procesory 486SX, či 386 totiž vyžadovaly pro podporu x87 pořízení dalšího čipu

(externího matematického koprocessoru). Procesory počínaje Pentiem mají koprocessor integrovaný a sadu x87 podporují.

Běžné instrukce **x86** a **x87** jsou schopny pomocí jedné instrukce spočítat pouze jednu informaci (data). Data se načtou do paměti zvané *register* (značené EAX, EBX a pod.) a poté se provede operace, která tyto data změní. Do jednoho registru je možné nahrát pouze jednu informaci (jedno číslo), proto i provedená operace upravuje pouze toto jedno číslo. Dalším krokem vývoje a zrychlování práce procesoru umožnil typ instrukcí **SIMD** (*Single Instruction Multiple Data*). Jedna *SIMD* instrukce provádí výpočet na více datech současně. Použití je vhodné především u multimediálních aplikací, kde je například možné zpracovávat několik částí audia současně. Oproti klasickému způsobu označených jako *SISD* - jedna instrukce jedny data (*Single Instruction Single Data*) je tento způsob významně rychlejší, byť ho nelze využít ve všech případech. První *SIMD* sadou byla instrukční sada **MMX** (*MultiMedia eXtensions*). Tato sada byla opět celočíselnou sadou. Dnes ji podporují prakticky všechny procesory a je často vyžadována různými programy. Tato technologie byla poprvé zavedena u procesoru Pentium jako další vylepšení pro kompresi videa, práci s obrazem, šifrování a IO operace. Technologie **MMX** zahrnovala dvě vylepšení architektury procesoru, první z nich bylo zvětšení L1 Cache a druhé přidání 57 nových *SIMD* instrukcí.

Rozšíření této sady bylo označováno jako **Enhanced MMX**, bylo zavedeno společností Cyrix a podporovalo ho většina výrobců s výjimkou Intelu. Ten přišel s vlastní instrukční sadou rozšiřující původní sady. Instrukční sada uvedená v roce 1999 nesla název **SSE** (*Streaming SIMD Extensions*). Sada se ihned stala součástí procesorů Pentium III a (ve své době) rychlejších Celeronů, obsahovala dalších 70 instrukcí pro zpracování grafiky a zvuku. Umožňuje práci s desetinnými čísly, jež ale vyžadovala v procesoru speciální přídavnou jednotku, ta standardní stále využívala dle potřeby instrukce *MMX*. *SSE* byla považována v době uvedení za nedodělek (nezvládala mnoho operací, které se od ní očekávaly), takže podpora se soustředila spíše na některé vybrané programy z oblasti pracovních stanic. *SSE* byla původně známá pod jménem *KNI* což byla zkratka pro *Katmai New Instructions* (Katmai bylo první jádro Pentia III). Tato sada instrukcí byla jednou z mála věcí, která odlišovala Pentium III od Pentia II. *SSE* později implementovalo i AMD ve svých procesorech Athlon XP. Technologie *SSE* přidala mimo instrukce 8 nových 128-bitových registrů známých jako XMM0 až XMM7. Každý z nich obsahoval čtyři 32-bitové registry pro čísla s plovoucí desetinnou čárkou. Alternativou firmy AMD k instrukční sadě *SSE* byla sada **3DNow!** a později **Enhanced 3DNow!**. Jedná se v podstatě o rozšíření *MMX* a *SSE* o nové instrukce, ty ale nejsou kompatibilní s instrukcemi *SSE*. Tato sada se tak nedočkala například podpory společnosti Adobe ve svých aplikacích Photoshop a tak AMD v dalších sadách musí umět implementovat instrukční sady *SSE*. V roce 2000 se sada *SSE* dočkala velkého rozšíření zavedeného pro procesory Pentium4. Jedná se o sadu **SSE2**, která obsahuje všechny instrukce *MMX* a *SSE* a navíc dalších až 144 instrukcí. Ty jsou určeny pro vyšší rozlišení a kvalitu zpracování grafiky, možnosti kódování MPEG2 a snížení zátěže CPU. Dočkalo se pozitivního přijetí mezi programátory a mnoho programů tak je schopno těžit z vyššího výkonu. Po *MMX* se jedná o druhou nejpoužívanější sadu. Dalším rozšířením, již ne tak rozsáhlým, byla instrukční sada **SSE3**. Jedná se o rozšíření o 13 instrukcí oproti *SSE2*. Sada nesla také označení *PNI* (*Prescott New Instruction*), což napovídá její uvedení pro Pentia 4 s jádrem Prescott po roce 2004. Tyto instrukce využívají i 64bitové procesory AMD. Pohled do blízké budoucnosti, viz. obr. 6.3, nám odkrývá 45 nm technologii s jádrem Nehalem, což s sebou přinese novou instrukční sadu *NNI* neboli *SSE4*.

AMD v převaze svých 64bitových procesů oproti Intelu zavádí sadu AMD64. Jedná se o 64bitové rozšíření x86, původně vyvíjené pod označením x86-64. Vyjma podpory 64bit. výpočtů umožňuje adresaci 64bit. paměťového prostoru (být zatím hardwarem omezenou na 40bit.) a dále také rozšiřuje počet použitelných registrů reálně na trojnásobek - tím eliminuje slabé místo x86. Řada lidí zaměňuje pojmy AMD64 s Athlonem 64. Taková záměna je chybná. Intel reagoval na tuto sadu vypuštěním **EM64T**. Jedná se v podstatě o Intelem vytvořenou kopii sady AMD64, byť s některými menšími rozdíly. Důvod jiného názvu je založen na neochotě Intelu přijmout název AMD64, přestože prakticky všichni vývojáři dávno používají název AMD64 jednoduše proto, že zde byl o mnoho let dříve. Všechny nové programy jsou vyvíjeny tak, aby případné minoritní rozdíly mezi AMD64 a EM64T byly ošetřeny.

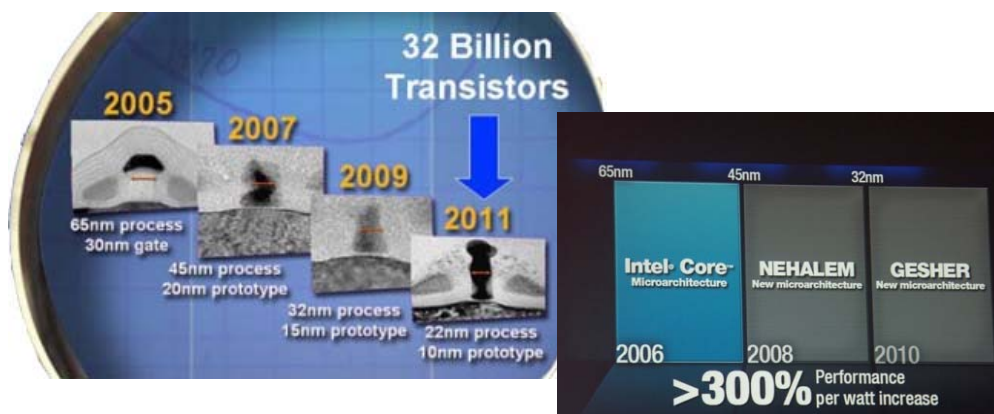
6.5 Výroba procesoru

1. Vytěžený křemík je zbaven příměsí, roztaven, destilován a zformován do tyčí o čistotě 99,999% vhodných k výrobě polovodičů. Tyče jsou rozlámány na kusy a vloženy do elektrických pecí umožňující tažení krystalů.
2. Křemík je zahřát na teplotu 1370 stupňů.
3. Jakmile je křemík roztaven, vloží se do něj malý zárodečný krystal, postupným tvářením se dosáhne toho, že se krystal rozšíří do požadovaného průměru.
4. Vznikne křemíkový ingot o průměru 20 – 30 cm, délka ingotu je 1,5 metru, hmotnost stovky kg.
5. Výsledný ingot je diamantovou pilou rozřezán na více než 1000 plátků o tloušťce 1 mm, každý plátek je pak vyleštěn.
6. Následuje fotolitografie – proces výroby: Vytvoření vrstvy oxidu křemičitého, vytvoření polovodičové vrstvy, nakonec metalizační vrstvy – spoje mezi tranzistory (hliníková metalizační vrstva je nově nahrazena mědí pro menší odpor).

Trendem v oblasti procesorů je snižování velikosti tranzistoru 130 nm, 90 nm, 65 nm tzv. "*Feature size*". Čím menší tranzistor, tím se dokáže rychleji přepínat - jinými slovy, čím má procesor menší tranzistory, tím dokáže běžet na vyšších frekvencích. Menší tranzistory zároveň vystačí s menším napětím, takže mají menší spotřebu energie.

U posledních generací (zejména u 90 nm) se však ukazuje, že menší transistory začínají mít čím dál tím větší potíže s tzv. *Leakage current*, což je stálá (statická) neovlivnitelná spotřeba. Proto výrobci začínají ve větší míře používat tranzistory více odolné proti *leakage*, které však zároveň nejsou tak rychlé. To pak znamená, že v mnoha případech může být procesor vyrobený starší technologií schopnější v dosahování vyšších frekvencí.

Pro první Pentium 4 se používal *wafer* o průměru 200 mm tj. ploše 31416 mm². Z takového plátku bylo možné vyrobit 145 CPU s jádrem *Williamette*, čip měl 217 mm² a obsahoval 42 milionů tranzistorů. P4 s jádrem *Northwood* 0,13 mikronu měl 131 mm² a ze stejného plátku bylo možno vyrobit 240 čipů. V roce 2002 se přešlo na 300 mm plátek o velikosti 70686 mm², takže bylo možno z plátku vyrobit až 540 čipů. Mezníkem se stal rok 2003, technologie se zdokonalila přechodem na 0,09 mikronu. O další tři roky později firma Intel ohlásila přechod na 0,065 mikronu, na což firma AMD nedokázala tak rychle zareagovat. Očekávají se další mezníky technologií, nejbližší očekávanou hranicí je 0,045 mikronu. Rovněž lze očekávat, že v přechodech na neustále hustější plátky bude hrát vedoucí úlohu firma Intel.



Obr 6.3 Plán firmy Intel pro přechod od technologie 65 nm až po 22 nm

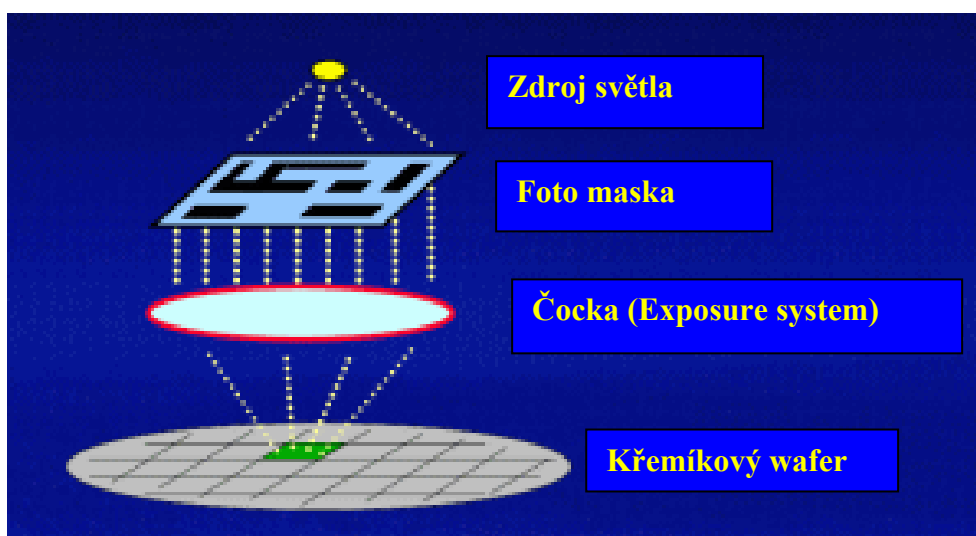
Typy křemíkové směsi pro výrobu procesoru:

Bulk Silicon - Klasický křemík s extrémní čistotou určený pro výrobu polovodičů.

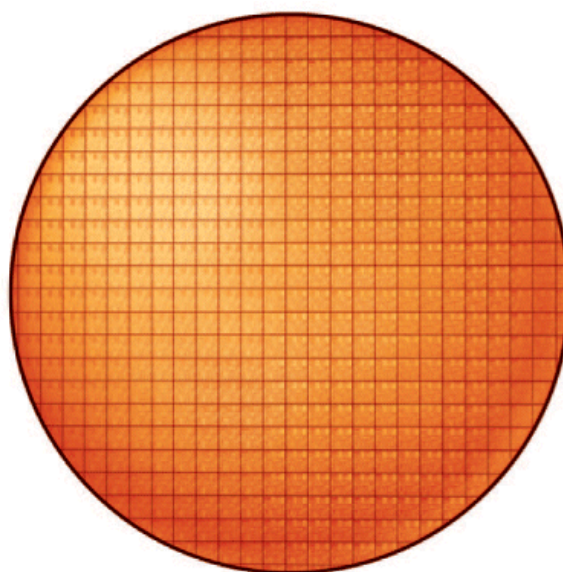
Strained Silicon (SS) - Klasický křemík obohacený o prvky germania, které se postará o vhodnější seskupování atomové struktury křemíku, což snižuje odpor. Strained Silicon umožňuje oproti bulk silicon dosahovat vyšších rychlostí přepínání tranzistorů, tj. i vyšších frekvencí při stejném napětí. Atomy křemíku, základního prvku pro tvorbu polovodičů, jsou napnuty za použití prvku germania. Takto napnuté atomy umožňují o 10 až 20 procent rychlejší proudění elektronů.

Silicon On Insulator (SOI) - Klasický křemík navařený na vrstvu oxidu, který slouží jako izolant. SOI umožňuje snížit spotřebu vznikající z tzv. leakage current a zároveň dosahovat vyšších frekvencí.

Strained Silicon Directly On Insulator (SSDOI) – Kombinace Strained Silicon a Silicon On Insulator. Doposud nejnovější výrobní technologie, umožňuje dosáhnout nízké spotřeby při vyšších frekvencích.



Obr. 6.4 Princip fotolitografie

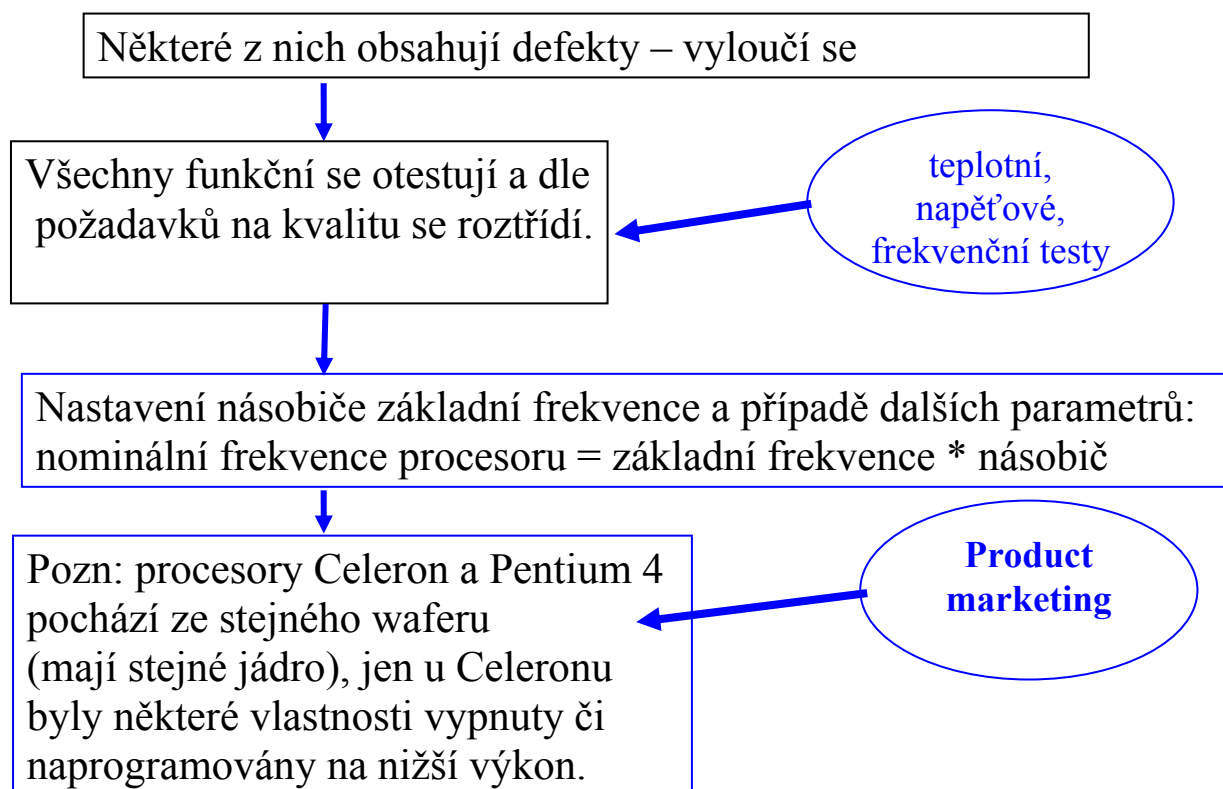


Obr. 6.5 Wafer jádra Northwood (jádro procesorů Pentia 4 a jejich verzí Celeronů)



Pojem k zapamatování: Výroba procesoru

Procesor se vyrábí z kruhové křemíkové desky označované jako *wafer*. Tento je přes fotomasku nasvětlován světlem o extrémně krátké vlnové délce a dále pak chemicky upravován. Díky světlu se vyrobí tranzistory, které pak tvoří základ procesorů - na každém *waferu* se jich nachází stovky.



Obr. 6.6 Další postup při výrobě procesorů

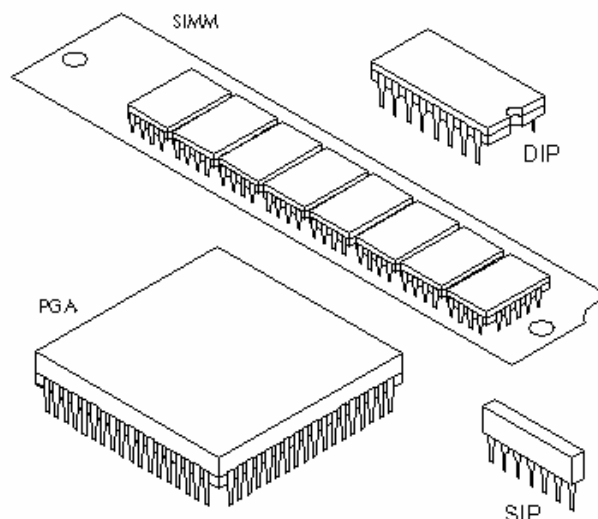
6.6 Pouzdra procesoru

Pouzdro procesoru vychází ze zavedeného vývojového standardu pouzder integrovaných obvodů. Pouzdra integrovaných obvodů se dají rozdělit na skupiny podle způsobu, jakým jsou namontovány na desku plošného spoje PCB (*Printed Circuit Board*). Historicky první se objevila klasická pouzdra s „nožičkami“, jež prochází skrz PCB. Mezi takováto pouzdra patří:

SIP (*Single In-line Package*) – jednoduché pouzdro pro integrované obvody s malým počtem součástek a vývodů (8),

DIP (*Dual In-line Package*) – stále se používá pouze pro malé stupně integrace a má dvojnásobný počet vývodů než SIP,

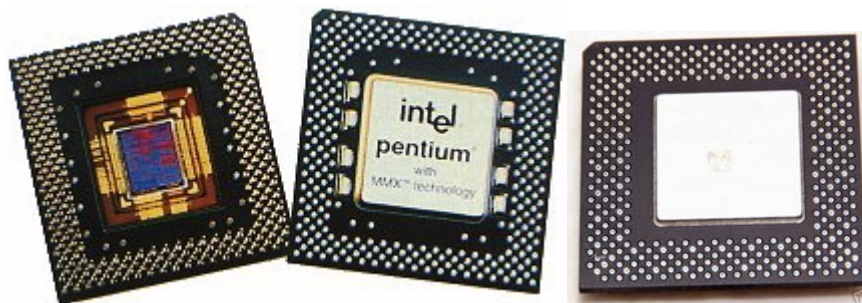
PGA (*Pin Grid Array*) – používá se pro integrované obvody s velmi vysokým počtem součástek i vývodů. PGA je navrženo tak, aby mohlo být osazeno (a případně vyjmuto) z patice. Pouzdra vycházející z tohoto typu s různými obměnami a vylepšeními používá většina procesorů.



Obr. 6.7 Pouzdro SIP, DIP a PGA

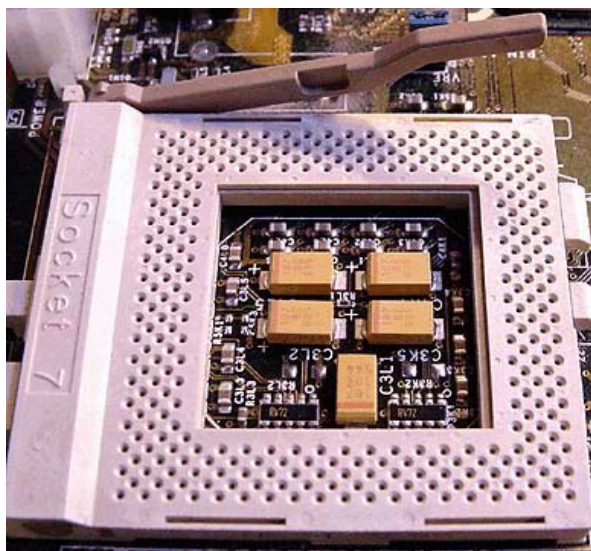
Označení PGA vychází z toho, že vývody procesoru jsou pravidelně uspořádány do mřížky. Takto zapouzdřené procesory se poté vsunují do patic (viz. kapitola Sloty a sockety na začátku učebního textu). Během vývoje procesoru existovalo několik variant pouzder PGA.

Pouzdro SPGA (*Staggered Pin Grid Array*) využilo mezer mezi pravidelnými sloupcemi a řadami pinů pouzdra pro umístění dalších vývodů. Příčinou byl vzrůstající počet vývodů procesoru a snaha o zachování či zmenšení velikosti obvodu. Varianty procesorů Celeron využívaly levnější varianty pouzder PPGA (*Plastic Pin Grid Array*).



Obr. 6.8 Procesor Intel Pentium MMX v pouzdře SPGA a Celeron v pouzdře PPGA

Další variantou bylo pouzdro typu FC-PGA (*Flip Chip PGA*), u kterého bylo uspořádání navrženo tak, aby bylo možné co nejsnáze odvádět teplo. Příkladem může být čip Intel Pentium III s jádrem Coppermine s 370 vývody či procesor Athlon firmy AMD s jádrem Thunderbird s 462 vývody. Pouzdro FC-PGA používají ale i některé čipy grafických karet. Typickým příkladem může být ATI Radeon 9600. Se stále vzrůstajícím zahříváním procesoru se firma Intel rozhodla použít ochranný kovový kryt pro vylepšení odvodu tepla a také z důvodu instalací masivních chladičů na procesory bez rizika jejich poškození. Tento druh pouzdra byl označen jako FS-PGA2. Pro mobilní procesory a také paměťové čipy je vyvinuto pouzdro BGA (*Ball Grid Array*). Místo klasických nožiček má vývody ve tvaru malých kuliček a je tak uzpůsoben k přímé montáži na povrch plošného spoje. Poslední variantou využívanou u současných procesorů je pouzdro typu LGA (*Land Grid Array*). Místo pinů jsou zespodu procesoru malé měděné plošky, zatímco piny jsou na socketu základní desky.



Obr. 6.9 Patice Socket 7 do kterého se zasune procesor v pouzdře typu SPGA



Obr. 6.10 Procesor Intel Pentium s měděnými ploškami místo pinů (LGA)

Ne vždy však procesory byly umístěné v pouzdře PGA nebo jeho variantě. Paměť L2 Cache totiž u těchto procesorů byla umístěna mimo jádro procesoru. Proto ji bylo nutné vložit na jednu desku současně s procesorem. První varianty pouzder bylo pouzdro SECC (*Single Edge Contact Cartridge*). Pouzdro uzavíralo uvedenou desku procesoru s pamětí cache a podobala se paměťovému modulu SIMM. Tento modul s pouzdem se zasouval do slotu na základní desce. Levnější varianta procesoru Celeron dokonce pouzdro neobsahovalo a totožný modul se zasouval do stejného typu slotu (typ SEC – *Single Edge Procesor*). Z důvodu chlazení procesoru bylo u procesoru Pentium III pouzdro upraveno a zakrývalo pouze jednu stranu procesorové desky, tímto dochází k lepšímu odvodu tepla. Jedná se o pouzdro SECC2.



Obr. 6.11 Procesory v zapouzdření SEP, SECC a SECC2

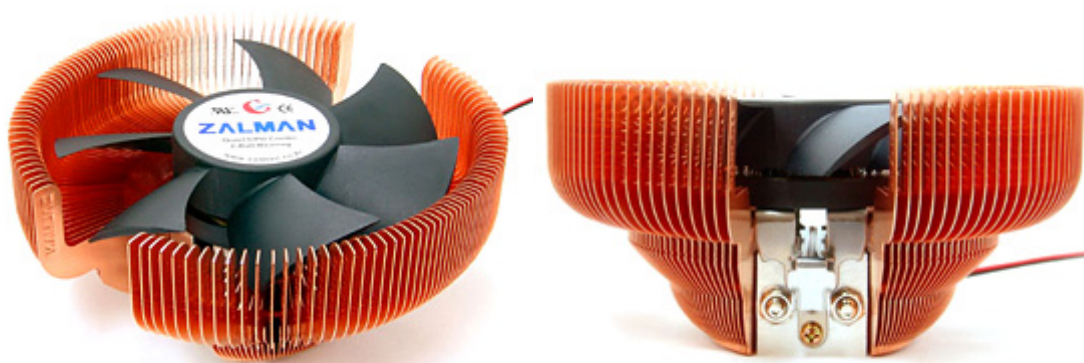
6.7 Chlazení procesoru

Chladiče jsou hodnoceny podle jejich schopnosti chladit, vyjádřené odolnosti vůči přenosu tepla. Jednotka je $^{\circ}\text{C}/\text{W}$ a platí, čím je hodnota nižší, tím je chladič lepší. Aktivní chladiče vyžadují vlastní napájení. Z počátku vývoje aktivních chladičů bylo důležitou volbou ložisko. Chladiče s klouzavými ložisky měly až 10 krát menší životnost než chladiče s kuličkovými ložisky. První procesory chlazení nepotřebovaly, začátkem vývoje a prodejem Pentii se začaly osazovat na CPU **pasivní chladiče**. Tyto chladiče se uchycovaly k procesoru buď pomocí úchytek typu LIF (*Low Insertion Force*) nebo ZIF (*Zero Insertion Force*), nebo se přímo na CPU lepily, což bylo problematické při výměně procesoru. Tvar chladiče kopíroval tvar procesoru - půdorys ve tvaru čtverce a měly žebrování.



Obr. 6.12 Pasivní chladič prvních procesorů Pentium

Materiálem používaným pro konstrukci chladičů je převážně měď kombinovaná s hliníkem. Pro vysokou hustotu vyzářeného tepla má většina moderních chladičů alespoň měděné jádro. V minulosti se používaly pasivní chladiče z hliníku, jejichž výhodou byla menší hmotnost, nižší cena a lehké opracování. S příchodem tepelně náročných procesorů se však do nich začaly montovat měděná jádra. Tato metoda zlepšuje chlazení procesorů s vysokou tepelnou hustotou. S kombinací mědi a hliníku se setkáte dodnes, používá ji Intel i AMD pro většinu svých boxovaných chladičů. Na obrázku 6.13 vidíme měděný chladič i s měděnou základnou.



Obr. 6.13 Chladič procesoru pro sockety 478, 775, 754, 939

Dnešními charakteristickými vlastnostmi chladičů mimo možnost nasazení na konkrétní typ socketu jsou hmotnost, rozměr, hlučnost a počet otáček. Tyto parametry se pohybují dle ceny a kvality konkrétního chladiče takto: hmotnost 300 – 950 gramů, 1000 – 3000 ot./min a hlučnost 15 – 40 dB.

Kromě běžného chlazení vzduchem se v oblasti počítačů můžete setkat i s dalšími metodami. Některé z nich jsou již v oblasti zájmu kutilů a také tzv. „*overclockerů*“ neboli fanoušků přetaktování. K těm běžnějším a také komerčně dostupným patří hlavně vodní chlazení, které je založeno na vodním

okruhu, který odvádí teplo od procesoru protékáním speciálním blokem. Voda je mnohem lepší vodič tepla než vzduch a díky promyšleným výměníkům tepla ji lze i efektivně ochladit. Příkladem vodního chlazení je univerzální chladicí systém pro CPU a grafickou kartu na obr. 6.14.



Pojem k zapamatování: Thermal Design Power (TDP)

Údaj vystihující tepelný výkon procesoru. Čím vyšší tepelný výkon je, tím je vyšší spotřeba a tím účinnější (a často také hlučnější) chladič je zapotřebí. Konkurenti Intel a AMD tento údaj prezentují rozdílně. Společnost AMD udává TDP jakožto maximální dosažitelný tepelný výkon, přičemž Intel jako typický. Faktický rozdíl tedy spočívá v tom, že v případě procesorů AMD lze hodnoty TDP dosáhnout jen velmi obtížně a reálný výsledek může být o několik desítek procent lepší, naopak u procesorů Intel je to hodnota, které bude při zátěži téměř s jistotou dosaženo. Jedná se tedy o marketingový trik Intelu, neboť procesory Intel nemohou v tepelném vyzařování procesorům AMD konkurovat, proto tímto odlišným vyjádřením TDP opticky vylepší svou pozici.



Obr. 6.14 Vodní chlazení pro procesor a grafickou kartu

Tento systém obsahuje kompaktní čerpadlo uzavřené uvnitř věže s průtokem až 300 litrů za hod. Použitým materiálem chladič věže je hliník, chladič procesoru obsahuje měděnou základnu potaženou zlatem (viz. Obr. 6.15). Cena takového systému vodního chlazení dosahuje 6000 Kč, ale na trhu jsou dostupné i dražší chladicí systémy. Extremními variantami chlazení jsou mrazicí okruhy či chlazení olejem.



Obr. 6.15 Vodní chladič procesoru (vlevo) a detail zapojení



Další zdroje

1. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
2. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
3. Křivohlávek, J. *Chlazení počítače*, Brno, Computer Press 2007, počet stran: 184. ISBN 80-251-1509-1.
4. Aspinwall, J. *Osobní počítač na maximum*, Brno, Computer Press 2006, počet stran: 256. ISBN 80-251-1136-9.
5. Mueller, S. *Osobní počítač – Hardware, upgrade, opravy*, Brno, Computer Press 2006, počet stran: 888. ISBN 80-7226-796-5.
6. Procesor – Elektronická učebnice [online] dostupný z http://zsprazska.oknet.cz/navody/arch_PC/TEXTY/PROCESOR.HTML
7. The BalusC Server - Hardware - CPU Types - Intel [online] dostupný z <http://balusc.xs4all.nl/srv/har-cpu-int-p1.php>



Shrnutí kapitoly

Procesory mohou být posuzovány podle dvou hlavních kritérií: podle šířky sběrnic a podle jejich rychlosti. Pojem šířka sběrnice se používá k popisu tří různých parametrů CPU: šířka vnitřních registrů, šířka vstupních a výstupních sběrnic pro data a šířka adresové sběrnice.

Šířka datové sběrnice určuje počet datových bitů, které mohou být v průběhu jednoho cyklu přesunuty z/do procesoru. Šířka adresové sběrnice určuje maximální velikost RAM, kterou je schopen procesor využít. Velikost vnitřních registrů určuje, kolik informací je procesor schopen zpracovat během jednoho cyklu a jakým způsobem jsou data v procesoru přesouvána. Z tohoto důvodu jsou registry nazývány vnitřní datovou sběrníci. Základní frekvence je generovaná frekvenčním generátorem. Slouží jako základ pro získávání dalších frekvencí.

Latence je doba, která uplyne od vyvolání požadavku do jeho splnění (reakční doba). Udává se běžně v nanosekundách (ns) nebo v hodinových cyklech (např. 4T jsou čtyři cykly). Čím nižší hodnota, tím lépe.

Intel zavedl technologii *Quad Pumped Bus FSB*, ve kterých dosahoval povětšinou dvojnásobné hodnoty oproti konkurenčnímu AMD. Quad Pumped Bus znamená, že jsou během jednoho cyklu po sběrnici přeneseny čtyři signály, to znamená čtyřnásobné zvýšení rychlosti. Procesory AMD však dosahovaly lepších výkonů na stejných frekvencích oproti procesorům Intel. Intel v té době zcela jasně vyhrával v honbě za překonávání co nejvyšších frekvencí a společnost AMD v tomto nemohla konkurovat. Zvolila proto strategii P-Ratingu, protože potřebovala dát uživatelům najevo, že jsou na trhu procesory, které jsou stejně výkonné jako Intel, přestože jsou taktovány na nižších frekvencích.

Instrukční sada je soubor instrukcí, kterým procesor rozumí a které byly vytvořeny s cílem provádět určité typy výpočtů. Během vývoje došlo k implementaci různých instrukčních sad: x86, x87, MMX, Enhanced MMX, 3DNow!, Enhanced 3DNow!, SSE, SSE2, SSE3, AMD64 a EM64T.

Trendem v oblasti procesorů je snižování velikosti tranzistoru 130 nm, 90 nm, 65 nm tzv. "Feature size". Čím menší tranzistor, tím se dokáže rychleji přepínat - jinými slovy, čím

má procesor menší tranzistory, tím dokáže běžet na vyšších frekvencích.

Procesor se vyrábí z kruhové křemíkové desky označované jako *wafer*. Tento je přes fotomasku nasvětlován světlem o extrémně krátké vlnové délce a dále pak chemicky upravován. Díky světlu se vyrobí tranzistory, které pak tvoří základ procesorů - na každém *waferu* se jich nachází stovky.

Pouzdro procesoru vychází ze zavedeného vývojového standardu pouzder integrovaných obvodů. Označení PGA vychází z toho, že vývody procesoru jsou pravidelně uspořádány do mřížky. Takto zapouzdržené procesory se poté vsunují do patic. Během vývoje se vystřídal více druhů pouzder založených na PGA.

Chladiče jsou hodnoceny podle jejich schopnosti chladit, vyjádřené odolnosti vůči přenosu tepla. Jednotka je $^{\circ}\text{C}/\text{W}$ a platí, čím je hodnota nižší, tím je chladič lepší. Materiálem používaným pro konstrukci chladičů je převážně měď kombinovaná s hliníkem. Pro vysokou hustotu vyzářeného tepla má většina moderních chladičů alespoň měděné jádro. Dnešními charakteristickými vlastnostmi chladičů mimo možnost nasazení na konkrétní typ socketu jsou hmotnost, rozměr, hlučnost a počet otáček. Tyto parametry se pohybují dle ceny a kvality konkrétního chladiče takto: hmotnost 300 – 950 gramů, 1000 – 3000 ot./min a hlučnost 15 – 40 dB.



Úkol k řešení 6.1

Přiřaďte číslům správné provedení pouzdra či patice.

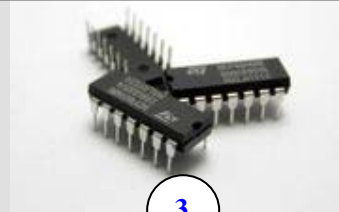
A) LGA B) BGA C) PGA D) SPGA E) SEC F) SEP G) DIP H) PPGA I) SIMM



1



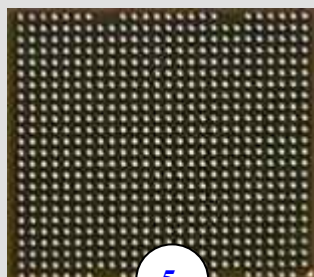
2



3



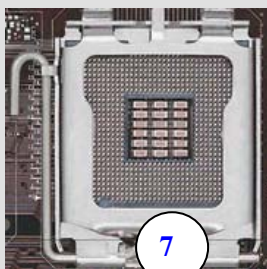
4



5



6



7



8



9

**Kontrolní otázka 6.1**

Vysvětlete co jednotlivé zkratky z minulého úkolu znamenají.

**Kontrolní otázka 6.2**

Rychlost paměťových čipů zdaleka neodpovídá rychlostem CPU, jak tedy zajistíme provozování CPU na takových rychlostech, když načítá data z paměti?

**Kontrolní otázka 6.3**

Jak souvisí šířka datové sběrnice se šířkou adresové sběrnice?

**Kontrolní otázka 6.4**

Popište postup tovární výroby procesoru.

**Kontrolní otázka 6.5**

Popište hierarchicky instrukční sady procesorů.

**Kontrolní otázka 6.6**

Definujte pojem sběrnice procesoru.

**Kontrolní otázka 6.7**

Pohovořte o hlavních pojmech v oblasti chlazení procesoru.

7. Procesory II



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- popsat vývoj procesorů AMD a Intel
- popsat blokové schéma procesoru
- vyjmenovat konkrétní typy jader procesoru a jejich charakteristické rysy
- vysvětlit princip zřetězení instrukcí
- definovat architekturu NetBurst procesorů Pentium 4
- popsat současný vývoj procesorů



Výklad

7.1 Procesory páté generace

V říjnu 92 Intel oznámil, že nebude pokračovat v číselném označení. Důvodem byla snaha o zavedení obchodní značky pro každý procesor. Pentium je kompatibilní směrem dolů, novinkou jsou dvě větve pro zpracování instrukcí tzv. superskalární technologie.

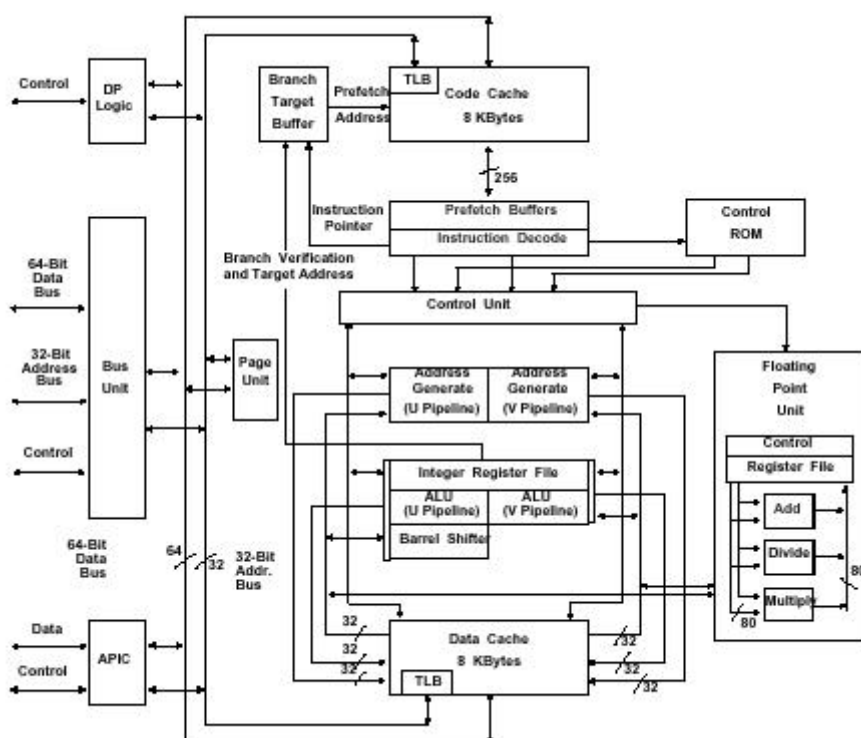
Doposud byla pro zpracování pouze jedna větev, takže CPU mohl v jednom cyklu zpracovávat pouze jednu instrukci. Jedna z větví pro vykonávání instrukcí se nazývá **u** větev a druhá **v** větev. Primární větví je větev **u**, která je schopná vykonávat všechny instrukce, jak s celými čísly, tak s desetinnými. Naopak **v** větev vykonává některé instrukce s celými a jednoduché instrukce s desetinnými čísly.

Proces provádění dvou různých instrukcí současně ve dvou větvích se nazývá párování. Ne všechny instrukce mohou být párovány, pokud k takové situaci dojde, je použita pouze **u** větev.

Procesor také obsahuje zásobník BTB (*Branch Target Buffer*), který umožňuje předpovídat větvení. Jeho účelem je zabránit zpoždění ve vykonávání instrukcí v kterékoliv větvi, jenž by mohlo nastat v okamžiku načítání instrukcí, ukazujících na nesouvisející místa v paměti. BTB se pokouší takové větvení předpovědět a dopředu načíst potřebné instrukce.

Pentium má 32bitovou adresovou a 64bitovou datovou sběrnici. CPU má dvě oddělené paměti cache, každou o velikosti 8 kB. Obvody řadiče cache a samotné cache jsou integrovány přímo do čipu. Obě paměti cache jsou rozděleny do dvou asociativních sad, každá z nich je rozdělena na řádky po 32 bytech. Současně každá cache má svůj zásobník pro překlad adres, nazvaný TLB (*Translation Lookaside Buffer*). L2 cache o velikosti 512 kB je možno přidat na základní desku PC.

Pentium první generace bylo taktováno na 66 MHz, rychlost základní desky je shodná s rychlostí CPU. Procesor měl pouzdro PGA s 273 vývody, použitou technologii 0,8 mikronu a 3,1 milionů tranzistorů. Pentium druhé generace bylo uvedeno na trh v březnu 94. První verze byly určeny pro rychlosti 90 a 100 MHz, postupně 120, 133, 166 a 200 MHz. Základní změnou byla výrobní technologie, pro CPU 100 MHz 0,6 mikronu a rychlejší verze 0,35 mikronu. V důsledku změny technologie došlo ke zmenšení procesoru i ke snížení spotřeby el. energie a napájecí napětí bylo sníženo z 5 na 3,3 V. CPU měl pouzdro SPGA s 296 vývody.



Obr. 7.1 Blokové schéma procesoru Pentium



Pojem k zapamatování: rychlost Procesorů Pentium I

Pentium druhé generace je první procesor, který již běžel na násobcích frekvence základní desky.

Všechny desky umožňovaly tři hodnoty $f = 50, 60, 66$ MHz, CPU běžely na různých násobcích základní desky. Poměr rychlosti jádra čipu k rychlosti základní desky je řízen dvěma vývody čipu nazývanými BF1 a BF2.

Procesor	Násobek rychlosti základní desky	Rychlost základní desky	Vývod BF1	Vývod BF2
Pentium 75	1,5x	50 MHz	1	1
Pentium 90	1,5x	60 MHz	1	1
Pentium 100	1,5x	66 MHz	1	1
Pentium 120	2x	60 MHz	1	0
Pentium 133	2x	66 MHz	1	0
Pentium 150	2,5x	60 MHz	0	0
Pentium 166	2,5x	66 MHz	0	0
Pentium 200	3x	66 MHz	0	1
Pentium 233	3,5x	66 MHz	1	1
Pentium 266	4x	66 MHz	1	0

Tab. 7.1 Procesory Pentium druhé generace

Třetí generace Pentia se objevila na trhu v lednu 1997. Základní změnou bylo přidání technologie MMX. Charakteristickými pojmy pro třetí generaci Pentia jsou:

- CPU běžel na rychlostech 66/166 66/200 66/233,
- zvětšení cache na 16 KB,
- přidání větve pro MMX instrukce,
- počet tranzistorů 4,5 mil.,
- výrobní technologie 0,35 mikronu,
- snížení napájecího napětí na 2,8 V.

Chyby procesorů Pentium

Nejznámější chybou Pentia je legendární chyba **FDIV**, neboť se nejčastěji projevovala u dělení desetinných čísel. Oficiální označení chyby firmou Intel je Errata No. 23 *Slight precision loss for floating point divides on specific operand pairs* (Malá ztráta přesnosti při dělení určitých dvojic čísel mající za výsledek desetinné číslo). Chyba byla poprvé oznámena jedním matematikem v říjnu roku 94. Intel byl donucen stáhnout všechny CPU a bezplatně vyměnit. Chybovost CPU se dá ověřit vydělením dvojic čísel, jejichž seznamy jsou dostupné na internetu.

Např:

$962\,306\,957\,033 / 11\,010\,046 = 87402,6282027341$ (správný výsledek)

$962\,306\,957\,033 / 11\,010\,046 = 87399,5805831329$ (vadné Pentium)

Což je extrémní případ neboť chyba je již na třetí platné číslici výsledku. Celkem bylo testováno asi 5000 dvojic čísel o délce 5 až 15 číslic, u nichž se předpokládaly problémy. Chyba byla nejčastěji na šesté platné číslici výsledku.

AMD K5

- AMD je plně kompatibilní s Pentiem,
- verze PR75, PR90, PR100, PR120, PR133, PR166,
- každá základní deska určená pro Pentium podporovala i AMD K5.

K základním vlastnostem procesoru K5 patří: 16 kB cache pro kód, 8 kB cache pro data (typ write-back), dynamické vykonávání instrukcí - předpovídání větvení se spekulativním vykonáváním, pětistupňová hlavní větev, podobající se svou architekturou procesorům RISC a mající šest paralelních funkčních jednotek, vysoce výkonná jednotka pro práci s desetinnými čísly, rychlost procesoru nastavitelná vývody na 1,5násobek či 2násobek rychlosti základní desky.



Obr. 7.2 Procesory páté generace Intel Pentium a AMD K 5

7.2 Procesory šesté generace

Hlavní novinkou procesorů páté generace byla superskalární architektura a podpora MMX. U procesoru šesté generace to jsou dynamické vykonávání instrukcí, architektura DIB a vylepšení superskalární architektury.

Varianta P6 odstartoval listopad 95 procesorem Pentium Pro. Dynamické vykonávání instrukcí umožňuje procesoru vykonávání více instrukcí současně, tato technologie se skládá z následujících hlavních částí:

1. Vícenásobné předpovídání větvení. Umožňuje předpovídat chod programu několika různými větvemi.
2. Analýza toku dat. Instrukce jsou vykonávány nezávisle na pořadí v původním kódu.
3. Spekulativní vykonávání instrukcí. Rychlost vykonávání instrukcí je zvýšena tím, že procesor se snaží vyhledat a posléze vykonávat v programu ty instrukce, které by mohly být v nejbližších okamžicích potřeba.

Architektura DIP (*Dual Independent Bus*) znamená fakt, že procesor je vybaven dvěma datovými sběrnicemi, z nichž jedna je určena pro přenos dat mezi základní deskou a procesorem a druhá je určena pro přenos dat mezi procesorem a pamětí cache.

Vylepšení superskalární architektury je u procesorů šesté generace dosaženo přidáním více jednotek pro vykonávání instrukcí a rozložení instrukcí na mikrooperace - procesory CISC se začínají podobat procesorům RISC, procesor je pak schopen tyto mikrooperace efektivněji vykonávat v několika paralelních jednotkách.

P6 má oproti dvou větvím pro vykonávání instrukcí u P5 větví šest. Výsledkem je vykonání až 3 instrukcí v jednom cyklu. Další vlastnosti P6 je optimalizace pro 32bitový operační systém. Výkon je tedy využit v systémech NT a 2000 oproti 95, 98.

Pentium Pro byl určen pro patici Socket 8, má 387 vývodů. Počet tranzistorů se zvýšil na 5,5 milionů.

Procesor P6 má v podstatě dvě oddělená jádra. První obsahuje procesor a druhé L2 Cache (256, 512kB nebo 1 MB). L2 cache o velikosti 256 kB obsahuje 15,5 milionu tranzistorů u 1MB až 62 mil.!



Obr. 7.3 Procesory šesté generace Intel Pentium Pro, Pentium II a Celeron ve verzi slot

Pentium II byl uveden na trh v květnu 97. Hlavním rozdílem oproti procesoru Pentium Pro je změna pouzdra PGA na SEC nebo SEC2, což přináší výrobní výhodu pro snazší změnu velikosti L2 cache.

Jádro P II obsahuje 7,5 milionů tranzistorů, zpočátku se vyráběl procesor technologií 0,35 mikronu, později u variant PII od 333 MHz 0,25 mikronu. Přesunutí paměti ven z procesoru získal Intel možnost použít levnější obvody pro samotnou cache. Rychlost Pentia Pro byla omezena na 200 MHz především z důvodu, že Intel nedokázal najít levného dodavatele rychlejší paměti.

V této době dělal Intel všechny možné marketingové a výrobní triky pro dosažení co nejlepší pozice na trhu. Snížením rychlosti L2 cache na polovinu rychlosti jádra umožnilo přejít na CPU o rychlosti

400 MHz. Snížení rychlosti cache se naopak vykompenzovalo zvětšením velikosti této paměti na 512 kB. Takže Intel vymačkává ze svých technologií maximum a zákazníci jsou spokojeni.



Pojem k zapamatování: Procesory Pentium II a Pentium III

U procesoru Pentium II byla velikost paměti zvětšena na 512 kB. Procesor se vyráběl v těchto verzích taktovacích frekvencí: 233 – 266 – 300 – 333 – 350 – 400 a 450 MHz.

Procesory Pentium III jsou taktovány od 400 po 1400 MHz s jádrem Katmai (250 nm, 450-900MHz), Coppermine (180nm, 533-1133 MHz) a Tualatin (130nm, 1000 – 1400 MHz).

Procesory Celeron mají jádro totožné s jádrem procesoru Pentium příslušné číselné řady. Tento procesor je z důvodu konkurence určen pro levnější trh. Rozdílem je velikost L2 cache, typ pouzdra a rychlost. Verze pro rychlost 300 MHz a výše mají L2 cache 128 kB, pod 300 MHz L2 cache vůbec neobsahovaly. Další snížení výrobních nákladů bylo u pouzdra PPGA (*Plastic Pin Grid Array*).

Rychlejší Celerony s jádrem PIII nebo dokonce Pentium IV mají ve skutečnosti 256 kB Cache L2, avšak pouze 128 je použitelných. Důvodem je výrobní strategie, je lepší používat stejné masky pro výrobu dvou typů procesorů a omezit cache.

Pentium III přišel na trh v únoru 1999.

Architektura přinesla zlepšení v podpoře SSE zahrnující 70 nových instrukcí. Procesor byl zpočátku vyráběn technologií 0,25 mikronu a jádro mělo 9,5 milionu tranzistorů. Ze začátku byla cache L2 o velikosti 512 kB na poloviční rychlosti CPU, avšak po přechodu na 0,18 mikronu se dostala cache na stejnou rychlost procesoru, ale její velikost se snížila na 256 kB.



Obr. 7.4 Procesory šesté generace Intel Pentium III různých verzí

Procesory Pentium III jsou taktovány od 400 po 1400 MHz, mohou být použity ve víceprocesorových systémech, speciální verzi PIII je procesor Xeon pro serverová řešení, odlišují se velikostí L2 cache, ta dosahuje až 2 MB. Další novinkou je výrobní číslo procesoru. Je zavedeno díky tlaku informačních technologií na lepší zabezpečení. Procesor již obsahuje tepelné čidlo pro detekci teploty jádra a byla zavedena ochrana proti přetaktování.

AMD K6 výkonnostně patřil mezi Pentium I a II. Mezi základní vlastnosti, které bychom si měli zapamatovat patří:

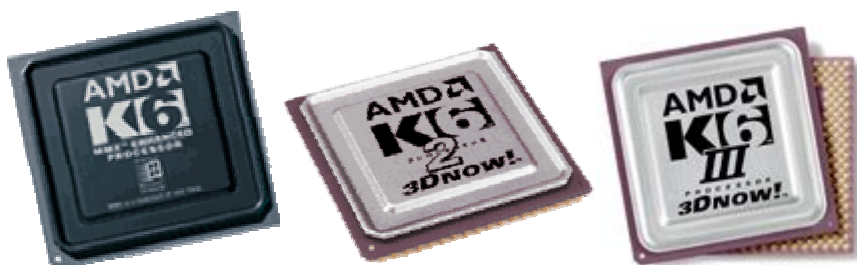
- kompatibilita desky pro Pentium, což je výkonové omezení(Socket7),
- K6 podporuje MMX, verze K6-2 přidala 3DNow!,
- technologie 0,35 mikronu později 0,25 mikronu. 8,8 milionů tranzistorů,
- vnitřní architektura odpovídá šesté generaci, avšak vnější rozhraní je odvozeno od páté generace CPU,
 - jádro typu Risc,
 - dynamické vykonávání instrukcí,
 - předpovídání větvení,

- spekulativní vykonávání instrukcí.

Verze K6-2 přešla na patici Super7, obsahují 9,3 milionů tranzistorů.

- podpora vyšších frekvencí: 300, 333, 350, 380, 400, 450, 500, 533, 550,
- vyšší rychlosti základních desek,
- nemají L2 cache, ta je přítomna na základní desce, takže běží na rychlosti desky,
- oproti Celeronům, jimž konkurují, mají dvojnásobnou L1 cache.

V roce 1998 byl uveden procesor K6-3. Procesor obsahuje 21,3 milionů tranzistorů a je vyráběn 0,25 mikronovou technologií. Osazuje se na základní desku s paticí Super7 se 100 MHz sběrnici. Na procesoru se nachází L1 cache 64 kB, což je dvojnásobek oproti Pentiu III. Na frekvenci procesoru pracuje i L2 cache s velikostí 256 kB. Konkurenční Pentium III obsahuje sice dvojnásobnou L2 cache 512 kB, avšak pracuje na poloviční frekvenci. Jenže tato nevýhoda je vykompenzována ve prospěch Intelu tím, že procesory K6-3 nedokáží konkurovat Pentiu III v taktovacích frekvencích a zůstaly pod hranicí 500 MHz.

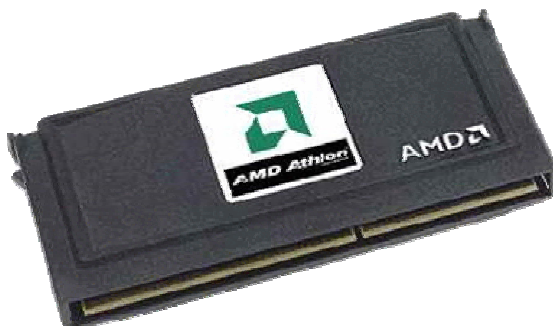


Obr. 7.5 Procesory šesté generace AMD K6

7.3 Procesory sedmé generace

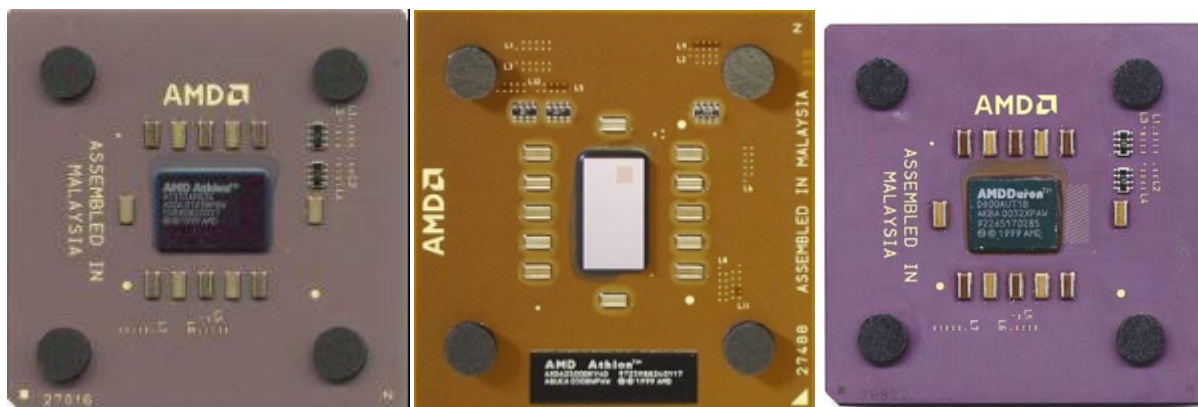
Procesorem sedmé generace Athlon firma AMD vyrovnala vývojový náskok Intelu. Jedná se o zcela nový čip, který je následníkem řady K6. U původních verzí tohoto procesoru používala firma AMD kazetové pouzdro, které se značně podobalo pouzdrům SEC firmy Intel (nutný konektor Slot A). To bylo dáno tím, že firma AMD - stejně jako Intel - vyráběla tyto procesory s vnější pamětí L2 cache o velikosti 512 kB, umístěnou na destičce s procesorem. Samotná cache pak v závislosti na verzi pracovala na jedné polovině, dvou pětinach či jedné třetině rychlosti jádra procesoru. V červnu 2000 však firma AMD uvedla na trh novou verzi procesoru, obsahující 256 kB L2 cache přímo u jádra. Samozřejmě tato cache pracuje na plné rychlosti jádra. Současně s tím začaly být procesory vkládány do pouzder typu PGA (patice socket A). Od této chvíle je rozmístění vývodů naprosto odlišné od Intelu a je nutné použít jiné základní desky.

První procesory Athlon byly taktovány od 550 MHz po 1 GHz. Výrobní technologii zpočátku na 0,25 mikronech vystřídala později technologie 0,18 mikronu a nakonec u procesoru Athlon XP dokonce 0,13 mikronu. Zajímavým faktem je použití mědi při výrobě, čímž firma AMD předstihla Intel.



Obr. 7.6 Procesor sedmé generace AMD Athlon

Procesory sedmé generace lze umístit do doby 1999 – 2005, kdy architekturu K7 vystřídala K8 s 64bitovými procesory Opteron a Athlon 64. Prvního představitele řady K7 jsme již uvedli, avšak firma AMD dosáhla největšího úspěchu a dobyla výborné pozice na trhu s procesory sedmé generace, konkrétně s procesorem AMD Athlon XP. XP znamená *eXtreme Performance*, což není technologie, ale obchodní značka pro souboj s konkurenčním Intel Pentiem 4. Největší změny se dočkala instrukční sada a to podpory SSE. XP disponuje podporou superskalární architektury z 9 částí: 3 jednotky pro výpočty s desetinnými čísly, 3 jednotky pro výpočty s celými čísly a 3 jednotky pro výpočty adres. Načítání adres je zajištěno s předstihem, data jsou ze systémové paměti RAM načítána do cache ještě dříve než je CPU potřebuje, architektura obsahuje také vylepšené zásobníky TLB pro překlad adres. První jádro Palomino uvedeno v roce 2001 obsahovalo 37 milionů tranzistorů. Athlon XP s tímto jádrem byl taktován na frekvenci 1 – 1,733 GHz, což ale v porovnání s procesory Intel dávalo výsledky P-ratingu až 2100+. Avšak sběrnice FSB byla stále 133 MHz *Double pumped*, což je 266 MHz a technologie 0,18 mikronu. Ještě téhož roku došlo k přechodu na technologii 0,13 mikronu u jádra Thoroughbred. Verze jader byly taktovány v rozmezí: T-Bred "A": 1400 - 1800 MHz (P rating 1600+ do 2200+) a T-Bred "B": 1400 - 2250 MHz (P rating 1600+ do 2800+). Poslední verze K7 byly jádra Burton a Thorton. Taktování dosáhlo 1833 - 2333 MHz (P rating 2500+ do 3200+) resp. 1667 - 2200 MHz (P rating 2000+ do 3100+). Tyto jádra obsahovaly 54 milionů tranzistorů a FSB dosáhla 400 MHz. Burton disponoval 512 kB L2 cache, Thorton o polovinu méně. Obdobně jako Intel Celeron, i firma AMD zavedla na trh levnější varianty procesorů Duron odvozených od verzí Athlonu. Tyto procesory disponovaly menší pamětí L2 cache, konkrétně 64 kB.



Obr. 7.7 Procesory AMD Athlon s jádrem Thunderbird, Athlon XP s jádrem Burton a AMD Duron

Procesor Pentium 4, uveden v listopadu roku 2000, byl představitelem zcela nového pojetí procesorů tzv. sedmé generace.



Pojem k zapamatování: Procesor Pentium 4 - rok 2000

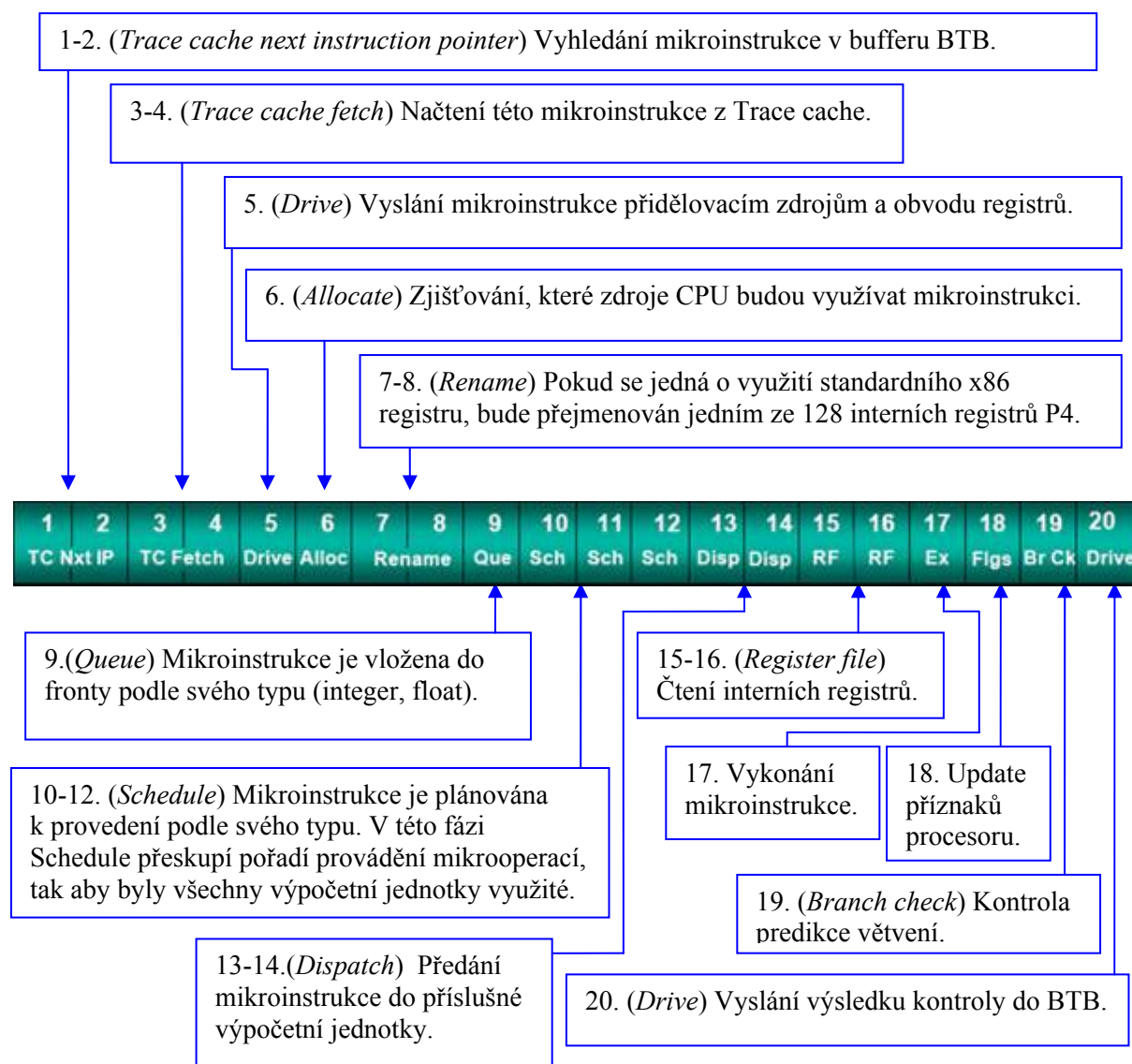
Novinkou této architektury byla podpora procesorové sběrnice 400 a 533 MHz představující přenosové rychlosti 3,2 či 4,26 GB/s, ALU jednotky běží na dvojnásobku rychlosti jádra - instrukce lze provést za polovinu cyklu.

Výrobní technologie byla nejprve 0,18 mikronu, 42 milionů tranzistorů na 217 mm² (jádro Williamette), později 0,13 mikronu 55 milionů tranzistorů 131 mm² (jádro Northwood).

První procesor vyráběný technologií 0,09 mikronu, neboli 90 nm byl Prescott. Byl uveden na trh 1.2.2004, přinesl nejen vyšší frekvence, ale i větší cache a různé úpravy v pipeline. Prescott je bezpochyby největší úprava architektury P4 od jejího uvedení.

Výrobní technologie procesoru Pentium 4 nese název **NetBurst** a je Intelem propagována a vyvíjena až do roku 2006. Její základní charakteristika je následující:

- Hyper řetězová technologie (*Hyper Pipelined Technology*) - procesor P4 zdvojnásobuje hloubku řetězení do dvaceti úrovní, čímž výrazně zvýšil výkonnost a frekvenční možnosti procesoru.
- *Rapid Execution Engine* označuje fakt, že aritmeticko logické jednotky procesoru (ALUs) pracují na dvojnásobné frekvenci jádra, což umožňuje provádět určité instrukce v polovičním taktu oproti jádrovému. Celočíselné instrukce se provádějí dvakrát rychleji, než je rychlost zbytku procesoru. Důsledkem je vyšší výkonnost provádění a snížená čekací doba provádění.
- Systémová sběrnice je taktována na 400 MHz, čemuž nemůže ve stejné době AMD konkurovat. Tato vespělá hloubkově zřetězená systémová sběrnice s rozděleným přenosem přenáší trojnásobnou šířku pásma oproti systémové sběrnici procesoru Intel Pentium III. Má 128-bitové linky s 64-bitovými přístupy (32-bitové linky u předchozí generace). To umožňuje 3,2 gigabytovou přenosovou rychlost mezi procesorem Pentium 4 a řadičem paměti a jedná se o systémovou sběrnici s nejvyšší dostupnou šířkou pásma pro stolní počítače.



Obr. 7.8 Dvacetistupňová pipeline architektury NetBurst

- *Execution Trace Cache* - Jedná se o vespělou instrukční vyrovnávací paměť úrovně Level 1, která vyrovnává dekódované instrukce (~12K micro-ops), čímž odstraňuje čekací dobu dekóderu u hlavního prováděcího cyklu. Tato revoluční technologie poskytuje instrukční vyrovnávací paměť s mnohem vyšším výkonem a zvyšuje účinnost ukládání ve vyrovnávací paměti. Vespělá 256 kB přenosová

vyrovnávací paměť procesoru (*Advanced Transfer Cache*) Level 2 navíc poskytuje 48 GB/sec rozhraní, které se přizpůsobuje růstu jádrové frekvence.

- Vyspělé dynamické provádění (*Advanced Dynamic Execution*). Procesor Pentium 4 má extrémně účinný poruchový spekulativní prováděcí engine, který trvale zatěžuje výkonné jednotky. Nová je také zvýšená schopnost predikce větvení, která udržuje činnost procesoru ve správném programovém toku a snižuje chybu špatné predikce spojenou s hlubším řetězením.
- *Streaming SIMD Extension 2* - SSE2 rozšiřuje technologie MMX a SSE přidáním 144 nových instrukcí, včetně 128-bitových SIMD celočíselných aritmetických a 128-bitových SIMD instrukcí s pohyblivou desetinnou čárkou o dvojnásobné přesnosti, které zvyšují výkon v celé šíři spektra aplikací.

V roce 2004 byl představen procesor na 90 nm technologii s jádrem Prescott. Na ploše 112 mm² se rozprostírá jádro s celkem 125 miliony tranzistorů. Toto jádro je uváděno na frekvencích od 2,8 GHz (s 533 MHz FSB a bez *HyperThreading*), 2,8, 3,0, 3,2 a 3,4 GHz (s 800 MHz FSB a s *HyperThreading*). Čip je plně kompatibilní s čipsety i875 a i865. Napětí čipů pro stolní počítače kolem 1,3 V - Procesory Prescott určené pro stolní počítače pracují s napětím v klidu 1.25V až 1.4V. Starší generace procesorů pracovala s napětími 1,475 V až 1,6 V. Nižší napětí umožňuje snížit spotřebu na jeden tranzistor. Byla implementována nová buňka SRAM pro paměť cache. Ta umožní integrovat na stejnou fyzickou plochu téměř dvojnásobné množství tranzistorů, tj. zvětšit velikost cache paměti (a zvýšit výkon). Nové jádro P4 přineslo řadu změn v architektuře oproti svým předchůdcům *Willamette* a *Northwood*. Prescott integruje zdvojené aritmeticko-logické jednotky (ALU), což mu umožní pracovat nativně s 64bit. čísly. Prescott disponuje dvěma dvojicemi ALU jednotek – ke každé dvojici patří L1 cache o velikosti 16 kB. Vývojáři implementovali lepší predikci větvení kódu. Ta umožňuje vykonávat kód i po podmínkových skocích ještě před vyhodnocením podmínky. Prescott má odhad stavu podmínky na vyšší úrovni než *Northwood*, což mu umožňuje plynulejší práci s podmínkami.

Zdokonalení dosáhl celočíselný multiplikátor. U procesorů *Willamette* a *Northwood* jsou celá čísla násobena v FPU jednotce. Protože ta je od ALU jednotky fyzicky vzdálena, trvá přesun do FPU a zpět dlouho. Prescott provádí násobení celých čísel již v ALU jednotce, tedy mnohem rychleji. Prescott je první procesor Intelu, který po dlouhé době podporuje nové instrukce. Je jich 13, nesly název *Prescott New Instruction* (PNI) a později byly přejmenovány na *Streaming SIMD Extensions 3* (SSE3). Díky vylepšené architektuře mohla být zvětšena pipeline až na 31 stupňů.



Pojem k zapamatování: Obrovská spotřeba energie procesorů Prescott

Typická spotřeba pro modely do 3.0 GHz je 89 W a pro rychlejší CPU dokonce přes 100 W! Prescott tak dosahuje extrémní hodnoty v proudovém odběru, 78 A a 91 A jsou hodnoty, které nedosahuje žádný jiný x86 procesor.

Konkurence Prescottu procesor AMD Athlon 64 má i při plném zatížení výhodu okolo 30W.

Tyto hodnoty zároveň znamenají problémy s výrobou čipu. Čipy pro rok 2005 a 2006 nesoucí značení 630 (3,0 GHz), 640 (3,2 GHz), 650 (3,4 GHz) a 660 (3,6 GHz) jsou posledními CPU a čip 670 na frekvenci 3,8 GHz, završí generaci Pentia 4 a technologii NetBurst.

Poslední revizí procesoru Pentium 4 bylo jádro Cedar Mill na 65 nm technologii. Procesor byl uveden v roce 2006 ale s blížící se technologií Intel Core mu nebyla dávana příliš velká šance na prosazení. Cedar Mill byl o něco lepší ve spotřebě než Prescott, disponoval stejnou pipeline a L2 cache měla velikost rovněž 2 MB.

7.4 Procesory osmé generace

Prakticky již v roce 2000 avizovalo AMD novou technologii, která má být obsažena v procesorech osmé generace. Jde o podporu 64-bitových aplikací při zachování instrukční sady x86, tedy se současnou podporou 32-bitového kódu.

Tato technologie, o které se dosud mluvilo jako o x86-64, získala oficiální název AMD64. Společnost nemusela budovat pro procesory K8 novou architekturu, ale stačilo přidat novou instrukční sadu, která dovolí provozovat i 64-bitový kód. Opteron ale dokáže díky AMD64 využít až 1024 GB fyzické (40-bit) a 256 TB virtuální paměti (48-bit).

Athlon 64 – Opteron

Procesor Athlon 64 – Opteron (viz. obr. 7.9) vznikl v roce 2003. Jedná se 64 bitový procesor běžící na frekvenci 1,4-3 GHz. Procesor obsahuje 105,9 miliónů tranzistorů. Procesor se instaluje do patice Socket 940. Opteron je určen serverům. Vyrábí se ve verzích Single-Core nebo Dual-Core. Jedná se o první procesor osmé generace. Opteron je vyráběn 0,13 mikronovým procesem.

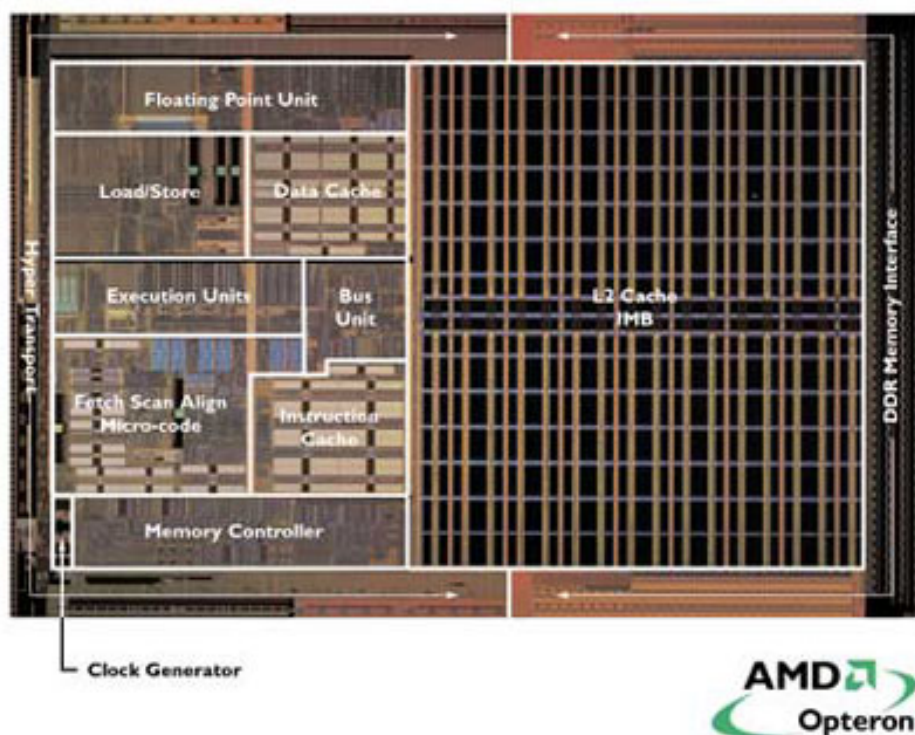


Obr. 7.9 Procesor osmé generace Opteron

Přehled nejdůležitějších vlastností Opteronu:

- 0,13mikronový výrobní proces SOI,
- 105,9 miliónů tranzistorů, velikost čipu 193mm²,
- 64kB instrukční a 64kB datové L1 cache (ECC), 1 MB L2 cache (ECC),
- procesor pro pracovní stanice a 1- až 8-cestné servery,
- nová patice Socket 940, integrovaný Heat Spreader,
- technologie AMD64 pro podporu 64bitových aplikací,
- integrovaný paměťový řadič pro DDR333 (dva 72bitové kanály),
- sběrnice HyperTransport s propustností až 6,4 GB/s,
- Prodloužená pipeline, vylepšený TLB a předpověď větvení kódu.

Opteron tedy ve svém jádře obsahuje integrovaný paměťový řadič. Většina současných procesorů přistupuje k hlavní paměti přes North Bridge, což je jedna součást čipové sady umístěná na základní desce. Spojení mezi procesorem a North Bridge musí být realizováno pomocí sběrnic, které zpomalují tok dat a zvyšují zpoždění (latenci). Protože má ale Opteron řadič paměti přímo v jádře, snižuje se tak latence a přístup je výrazně rychlejší, údajně až o 20%.



Obr. 7.10 procesor Opteron obsahující řadič paměti

Délka pipeline pro celočíselné operace		
Stupeň	Architektura K7	Architektura K8
1	Fetch	Fetch 1
2	Scan	Fetch 2
3	Align 1	Pick
4	Align 2	Decode 1
5	EDEC	Decode 2
6	IDEQ/Rename	Pack
7	Schedule	Pack/Decode
8	AGU/ALU	Dispatch
9	L1 Address Generation	Schedule
10	Data Cache	AGU/ALU
11		Data Cache 1
12		Data Cache 2

Obr. 7.11 12ti stupňová pipeline pro celočíselné operace

Athlon 64, Athlon 64-FX, AMD Athlon 64 X2 Dual-Core

Procesor Athlon 64 vznikl v roce 2003. Jedná se 64bitový procesor běžící na frekvenci 2 GHz a výš. Procesor obsahuje 105,9 miliónů tranzistorů. Procesor se instaluje do patice Socket 939 nebo Socket 754 a nyní AM2. Základem procesoru jsou tři páry ALU a AGU jednotek (jednotky pro výpočet celých čísel) a dvě výpočetní FPU jednotky plus jedna FPU jednotka na ukládání dat. Tedy totéž, co u předchozí architektury K7. Vylepšení tkví právě ve výpočetních jednotkách.

Prvním vylepšením je osmipoložkový řadič instrukcí namísto šestipoložkového. Schedule řadí instrukce za sebou v takovém pořadí, aby bylo možné co nejvíce vytížit výpočetní jednotku, tedy spočítat co nejvíce za co nejkratší dobu. U Athlonu jsou tři Schedulery po šesti položkách, v Athlonu 64 jsou tři po osmi.

Dalším zlepšením je rychlost násobení celých čísel. Zatímco v Athlonu XP trvalo násobení 32bit. registrů čtyři cykly, Athlonu 64 to trvá jen tři. Čtyři cykly trvá násobení 64bitových registrů. Procesor Athlon 64 se vyráběl nejprve 130 nm technologií (jádra ClawHammer a NewCastle – rozdíl ve velikosti L2 cache), poté 90 nm technologií (Winchester, San Diego, Orleans, Venice) a připraven je také přechod na 65 nm.

Procesor AMD Athlon 64 X2 Dual-Core vznikl v roce 2005. Jedná se 64bitový procesor běžící na frekvenci označované jako 4000+ a vyšší. Procesor se instaluje do patice Socket 939 a nyní AM2. Dual-core (v překladu "dvoujádrový") je takový procesor, který obsahuje dvě samostatné výpočetní části (jádra). Což v reálu znamená, že se PC teoreticky chová jako by mělo dva procesory.

Hlavním tahákem K8 procesorů AMD je podpora 64 bitů. Abychom mohli hovořit o 64bitovém procesoru, musí tento splňovat následující požadavky: adresace 64bit. paměťového rozsahu a výpočty s 64 bit. celými čísly v ALU jednotkách. První požadavek znamená, že procesor by měl umět adresovat 2 na šedesátou čtvrtou bytů RAM operační paměti, tj. 18446744073709551616 bytů, to je zatím ale zbytečné. Proto procesory Opteron a Athlon 64 umí adresovat „pouze“ 1024 GByte, tj. 1 TeraByte. Reálná použitelnost je ale omezena adresovací schopností paměťového řadiče, kdy jeden řadič zvládne 4 GByte - Opteron a Athlon 64FX tedy umí 8 GByte, klasický Athlon 64 pak pouze 4 GByte. Větší množství, než jsou tyto, připadá v úvahu pouze ve víceprocesorových systémech (pro které ale Athlony 64 nejsou určeny).

Athlon 64-FX je 64bitový procesor běžící na frekvenci 2,2 GHz a výš. Procesor obsahuje 105,9 miliónů tranzistorů. Procesor se instaluje do patice Socket 940. Procesor Athlon 64 FX je určen pro výkonné pracovní stanice a náročné uživatele. Má stejné vlastnosti jako Opteron. Jen s tím rozdílem, že Opteron má tři Hyper Transport sběrnice, zatímco Athlon 64 FX pouze jednu, takže nelze použít ve víceprocesorových konfiguracích. Procesory Athlon 64 FX jsou vyráběny zpočátku 0,13 mikronovou SOI technologií – jádro SledgeHammer, ClawHammer, později 90 nm technologií – jádro Toledo, SanDiego a Windsor. Procesor má 128 kB L1 (64 kB pro data + 64 kB pro instrukce) cache a 1 MB L2 cache.



Obr. 7.12 Procesor Athlon 64 FX

Verze jádra pro dual-core AMD jsou následující:

	Manchester 90 nm	Toledo 90 nm	Windsor 90 nm	Brisbane 65 nm
L1-Cache	64 + 64 kB (Data + instrukce na každé jádro)			
L2-Cache	512 kB	512 nebo 1024 kB	256, 512 nebo 1024 kB	512 kB
Technologie	MMX, Extended 3DNow!, SSE, SSE2, SSE3, AMD64, Cool'n'Quiet, NX Bit		MMX, Extended 3DNow!, SSE, SSE2, SSE3, AMD64, Cool'n'Quiet, NX Bit, AMD Virtualization	
Socket	Socket 939		Socket AM2	
VCore	1.35 V - 1.4 V	1.35 V - 1.4 V	1.25 V - 1.35 V	1.25 V - 1.35 V
Power Consumption (TDP)	89 - 110 Watt	89 Watt: 3800+, 4200+ a 4400+ 110 Watt: 4400+, 4600+ a 4800+	35 Watt 3800+, 65 Watt 3600+ až 4800+ 89 Watt 3800+ až 5600+ 125 Watt 6000+	65 Watt
Uvedení na trh	Srpen 2005	Duben 2005	Květen 2006	Prosinec 2006
Frekvence	2000 - 2400 MHz	2000-2400 MHz	2000-3000MHz	1900-2600MHz
P-rating	3800+: 2000 MHz 4200+: 2200 MHz 4600+: 2400 MHz	3800+: 2000 MHz 4200+: 2200 MHz 4400+: 2200 MHz 4600+: 2400 MHz 4800+: 2400 MHz	3600+: 2000 MHz 3800+: 2000 MHz 4200+: 2200 MHz 4600+: 2400 MHz 5000+: 2600 MHz 5400+: 2800 MHz 4000+: 2000 MHz 4400+: 2200 MHz 4800+: 2400 MHz 5200+: 2600 MHz 5600+: 2800 MHz 6000+: 3000 MHz	3600+:1900MHz 4000+:2100MHz 4400+:2300MHz 4800+:2500MHz 5000+:2600MHz

Tab 7.2 Verze CPU AMD Athlon 64 X2

7.5 DualCore a současnost (rok 2007)

Na konci roku 2006 Intel ohlásil revoluční změnu u svých procesorů.



Pojem k zapamatování: Rok 2006-7 - zánik Net Burst a Pentia 4

Pentium 4 s technologií NetBurst jsou již mimo hru. Společně s touto technologií zaniká vývoj Pentia M a Xeonu. Všechny procesory se vyprodají a nahradí je nová technologie Intel Core.

~~Pentium 4~~

~~Pentium M~~

~~XEON~~

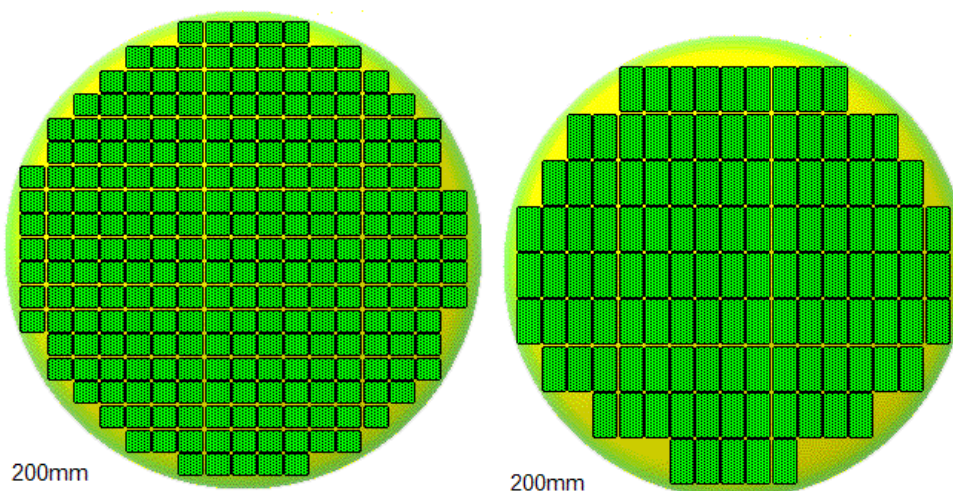
Intel oznámil, že nové jádro pronikne do všech segmentů – sjednotí se architektura, která ponese jméno zcela nové generace Intel Core pro PC, mobilní i serverové řešení. Byly vyvinuty jádra *Conroe*, *Woodcrest* a *Merom*.



Pojem k zapamatování: Intel Core – návrat ke kořenům

Intel Core = obdoba Pentia M (vycházející z Pentia III) + širší paralelismus + vyšší frekvence + více zdrojů pro provedení výpočtů + některá další vylepšení.

Dual-core (v překladu "dvoujádrový") je takový procesor, který obsahuje dvě samostatné výpočetní části (jádra). Hlavním přínosem dual-core procesorů je schopnost zpracovávat dva thready současně. Dvoujádrový procesor se tak v podstatě chová jako plnohodnotný počítač s dvěma procesory. Tím, že snížíme frekvenci, můžeme zároveň snížit i napětí. Pokles o 200 MHz může dnes přinést např. o 0,1V nižší napětí, což je skoro 10 procent. Snížení napětí o 10 procent zredukuje spotřebu dokonce o 20 procent. Když tímto způsobem snížíme spotřebu jednoho jádra např. na 60W, můžeme přidat jádro druhé. Dohromady se dostaneme na stejnou spotřebu jako u jednojádrového čipu, výkon však bude podstatně lepší. U Intelu pak platí, že: $2 \times 3,2 \text{ GHz (125 A)} > 1 \times 3,8 \text{ GHz (119 A)}$. Zpočátku této výrobní technologie však neuvidíme tak vysoké frekvence. Jedním z nejvýznamnějších problémů dvoujádrových procesorů pro zákazníka je cena. Jestliže jednojádrové Pentium 4 se rozkládá na 112 mm^2 a má 125 miliónů tranzistorů, pak dvoujádrové Pentium D zabere 206 mm^2 a má 230 miliónů tranzistorů. To je opravdu hodně. Na 200 mm wafer se tak Pentii 4 vejde odhadem 253, zatímco Pentii D jen 119.



Obr. 7.13 Srovnání waferu Single a Dual Core

V roce 2005 a začátkem roku 2006 jsou uváděné dvoujádrové procesory monolitické. To znamená, že dvě jádra jsou vyráběna spolu na jednom křemíku. Pentium D Presler řady 9xx v polovině roku 2006 vyroben na 65nm technologii, již chápeme jako dva samostatné čipy. Jednou z nejvíce problematických věcí u dual-core procesorů, co se týká výkonu, je L2 cache. Zásadní problém je ve sdílení obsahu. Existují v podstatě dva přístupy - separátní L2 cache pro každé jádro a sdílená L2 cache pro obě jádra. Separátní cache má výhodu rychlejší odezvy, avšak data změněná jedním jádrem musí být přenesena do jádra druhého. Naproti tomu u sdílené cache jsou změněná data ihned k dispozici druhému jádru, avšak sdílení cache má velkou nevýhodu v pomalejší odezvě.

Model	Frekvence	Násobič	FSB	L2 cache
E6700	2667 MHz	10x	10066 MHz	4 MB
E6600	2400 MHz	9x	10066 MHz	4 MB
E6400	2133 MHz	8x	10066 MHz	2 MB
E6300	1866 MHz	7x	10066 MHz	2 MB

Tab. 7.3 Stolní PC – jádro Conroe

Fakt, že zpočátku neuvídíme tak vysoké frekvence je důsledek toho, že firma Intel vsadila na spotřebu. Typické spotřeby nových Intel Core procesorů při zatížení (TDP) jsou na hodnotě 65W. Intel tudíž zaútočil na konkurenci, v místech kde po dlouhá léta zvyšování frekvencí jasně prohrával s konkurenčním AMD. AMD je v současnosti u těchto parametrů CPU na hodnotě 110W, přesto, že se jedná o max. spotřebu, neboť TDP, jak víme, jsou chápány u Intelu a AMD poněkud odlišně. Pro procesory Intel Core jsou určeny čipsety i965: P965 a G965 (integrované grafické jádro). Nejdůležitější změnou oproti současnému stavu je podpora nových pamětí DDR2-800.

Model	Frekvence	Násobič	FSB	L2
5160	3000 MHz	9x	1333 MHz	4 MB
5150	2667 MHz	8x	1333 MHz	4 MB
5140	2333 MHz	7x	1333 MHz	4 MB
5130	2000 MHz	6x	1333 MHz	4 MB
5120	1867MHz	7x	1066 MHz	4 MB
5110	1600 MHz	6x	1066 MHz	4 MB

Tab. 7.4 Servery – jádro Woodcrest

Nová patice u serverových procesorů LGA 771- piny se přestěhují do patice základní desky – zvýší se FSB na 333 MHz! QDR = 1333 MHz, FSB navýší přenosovou rychlost až na 10,6 GB/s, což je teoretické maximum dvoukanálového řadiče DDR2-667, a to je příležitost pro DDR2-800.

Architektura Intel Core sebou přináší řadu revolučních novinek, například dekodování až pěti instrukcí za takt. Instrukce, ve kterých jsou napsány programy, jsou poskytnuty z paměti cache a dekodovány na velmi malé mikroinstrukce. To se děje proto, aby jednotlivé mikroinstrukce byly paralelně prováděny ve výpočetních jednotkách. Intel Core má k dispozici čtyři instrukční dekodéry. Tři jsou určeny pro jednoduché instrukce, které se dekodují do jedné mikroinstrukce. Jeden dekodér je určen pro komplexní instrukce, které se přetvoří do až čtyřech mikroinstrukcí. Celkově tak dekodování umí dodat do bufferu až sedm microOPS za takt. To ovšem uměla i předchozí architektura P6. Novinkou však je, že jednoduché dekodéry nyní umí zpracovat i instrukce, které dříve musely procházet dekodérem komplexním. Jedná se především o instrukce pracující s pamětí a SSE instrukce.

Hlavní novinkou je tzv. fúzování makroinstrukcí, to znamená, že dojde k dekodování programu tak, aby se dvě instrukce dekodovaly na jednu mikroinstrukci, která však zastoupí obě dvě! Takto sloučené

instrukce pak převezme jeden dekodér a vytvoří z nich jednu microOP. Za jeden hodinový cyklus je možné takto sloučit jeden pár instrukcí a toto sloučení může provést libovolný ze čtyřech dekodérů.

Čtyři dekodéry tak mohou dekódovat až pět instrukcí (tři dekodéry jednu a jeden dvě fúzované). Výstupem dvou fúzovaných instrukcí je jedna microOP. Všechny ostatní dnešní procesory umí dekódovat pouze tři instrukce za takt. Další technikou, tentokrát převzatou z architektury mobilního Pentia, je fúzování mikrooperací. Tento proces je obdobný jako u makrooperací popsaných výše. Samozřejmostí architektury je Out-of-Order zpracování, které znamená, že instrukce jsou zpracovávány v jiném pořadí, než je program napsán. Zásadní změnou oproti předchozím architektuám jsou 64bitové jednotky pro počítání s celými čísly (ALU). Dalším vylepšením jsou plně 128bit jednotky pro SSE operace. Architektura Core tak zpracovává vektorově 128 bitů s využitím 128bitového registru najednou na jednu mikroinstrukci. Dalším doplněním instrukční sady bude u architektury Intel Core sada SSE4.



Další zdroje

1. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
2. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
3. Nygrýn P., Bábiček R. *444 tipů a triků pro nákup počítače a komponent*, Brno, Computer Press 2006, počet stran: 136. ISBN 80-251-0928-3.
4. Mueller, S. *Osobní počítač – Hardware, upgrade, opravy*, Brno, Computer Press 2006, počet stran: 888. ISBN 80-7226-796-5.
5. Procesor – Elektronická učebnice [online] dostupný z http://zsprazska.oknet.cz/navody/arch_PC/TEXTY/PROCESOR.HTML
6. The BalusC Server - Hardware - CPU Types - Intel [online] dostupný z <http://balusc.xs4all.nl/srv/har-cpu-int-p1.php>
7. Intel Pentium Technical documentation [online] dostupný z <http://www.intel.com/design/Pentium4/documentation.htm>
8. Advanced Micro Devices, AMD – Global Provider of Innovative Microprocessor Solutions [online] dostupný z <http://www.amd.com/us-en/>
9. Wikipedia otevřená encyklopedie - kapitola CPU [online] dostupný z <http://en.wikipedia.org/wiki/Cpu>



Shrnutí kapitoly - pokračování

Novinkou procesoru Pentium jsou dvě větve pro zpracování instrukcí tzv. superskalární technologie. Doposud byla pro zpracování pouze jedna větev, takže CPU mohl v jednom cyklu zpracovávat pouze jednu instrukci. Jedna z větví pro vykonávání instrukcí se nazývá *u* větev a druhá *v* větev. Primární větví je větev *u*, která je schopná vykonávat všechny instrukce jak s celými čísly, tak s desetinnými. Proces provádění dvou různých instrukcí současně ve dvou větvích se nazývá párování. Procesor také obsahuje zásobník BTB (*Branch Target Buffer*), který umožňuje předpovídat větvení. Jeho účelem je zabránit zpoždění ve vykonávání instrukcí v kterékoliv větvi, jenž by mohlo nastat v okamžiku načítání instrukcí, ukazujících na nesouvisející místa v paměti. BTB se pokouší takové větvení předpovědět a dopředu načíst potřebné instrukce. CPU má dvě oddělené paměti cache. Obvody řadiče cache a samotné cache jsou integrovány přímo do čipu. Obě paměti cache jsou rozděleny do dvou asociativních sad. Současně každá cache má svůj zásobník

pro překlad adres nazvaný TLB (*Translation Lookaside Buffer*).

Hlavní novinkou procesorů páté generace byla superskalární architektura a podpora MMX. U procesoru šesté generace to jsou dynamické vykonávání instrukcí, architektura DIB a vylepšení superskalární architektury. Procesorem sedmé generace Athlon firma AMD vyrovnala vývojový náskok Intelu. Jedná se o zcela nový čip, který je následníkem řady K6. První procesory Athlon byly taktovány od 550 MHz po 1 GHz. Výrobní technologie zpočátku na 0,25 mikronech vystřídala později technologie 0,18 mikronu a nakonec u procesoru Athlon XP dokonce 0,13 mikronu. Zajímavým faktem je použití mědi při výrobě, čímž firma AMD předstihla Intel. Procesory sedmé generace lze umístit do doby 1999 – 2005, kdy architekturu K7 vystřídala K8 s 64 bitovými procesory Opteron a Athlon 64. Firma AMD dosáhla největšího úspěchu a dobyla výborné pozice na trhu s procesory sedmé generace, konkrétně s procesorem AMD Athlon XP.

Výrobní technologie procesoru Pentium 4 nese název NetBurst a je Intelem propagována a vyvíjena až do roku 2006. Novinkou této architektury byla podpora procesorové sběrnice 400 a 533 MHz představující přenosové rychlosti 3,2 či 4,26 GB/s, ALU jednotky běží na dvounásobku rychlosti jádra, instrukce tak lze provést za polovinu cyklu. Výrobní technologie byla nejprve 0,18 mikronu, 42 milionů tranzistorů na 217 mm² (jádro Williamette), později 0,13 mikronu 55 milionů tranzistorů na 131 mm² (jádro Northwood). První procesor vyráběný technologií 0,09 mikronu, neboli 90 nm byl Prescott. Byl uveden na trh 1.2.2004, přinesl nejen vyšší frekvence, ale i větší cache a různé úpravy v pipeline. Prescott je bezpochyby největší úprava architektury P4 od jejího uvedení.

Prakticky již v roce 2000 avizovalo AMD novou technologii, která má být obsažena v procesorech osmé generace. Jde o podporu 64-bitových aplikací při zachování instrukční sady x86, tedy se současnou podporou 32-bitového kódu. Tato technologie, o které se dosud mluvilo jako o x86-64, získala oficiální název AMD64. Společnost nemusela budovat pro procesory K8 novou architekturu, ale stačilo přidat novou instrukční sadu, která dovolí provozovat i 64-bitový kód. Opteron ale dokáže díky AMD64 využít až 1024 GB fyzické (40-bit) a 256 TB virtuální paměti (48-bit).

Na konci roku 2006 Intel ohlásil revoluční změnu u svých procesorů. Pentium 4 s technologií NetBurst jsou již mimo hru. Společně s touto technologií zaniká vývoj Pentia M a Xeonu. Všechny procesory se vyprodají a nahradí je nová technologie Intel Core. Dual-core (v překladu "dvoujádrový") je takový procesor, který obsahuje dvě samostatné výpočetní části (jádra). Hlavním přínosem dual-core procesorů je schopnost zpracovávat dva thready současně. Dvoujádrový procesor se tak v podstatě chová jako plnohodnotný počítač se dvěma procesory. V roce 2005 a začátkem roku 2006 jsou uváděné dvoujádrové procesory monolitické. To znamená, že dvě jádra jsou vyráběna spolu na jednom křemíku. Pentium D Presler řady 9xx v polovině roku 2006 vyroben na 65nm technologii, již chápeme jako dva samostatné čipy. Jednou z nejvíce problematických věcí u dual-core procesorů, co se týká výkonu, je L2 cache. Zásadní problém je ve sdílení obsahu. Existují v podstatě dva přístupy: separátní L2 cache pro každé jádro a sdílená L2 cache pro obě jádra.

Architektura Intel Core sebou přináší řadu revolučních novinek, například dekodování až pěti instrukcí za takt. Instrukce, ve kterých jsou napsány programy, jsou poskytnuty z paměti cache a dekodovány na velmi malé mikroinstrukce. To se děje proto, aby jednotlivé mikroinstrukce byly paralelně prováděny ve výpočetních jednotkách. Intel Core má k dispozici čtyři instrukční dekodéry. Tři jsou určeny pro jednoduché instrukce, které se dekodují do jedné mikroinstrukce. Jeden dekodér je určen pro komplexní instrukce, které se přetvoří do až čtyřech mikroinstrukcí. Celkově tak dekodování umí dodat do bufferu až sedm mikroOPs za takt. Novinkou je, že jednoduché dekodéry nyní umí zpracovat i instrukce, které dříve musely procházet dekodérem komplexním. Jedná se především o instrukce pracující s pamětí a SSE instrukce.

**Úkol k řešení 7.1**

Zakreslete blokové schéma procesoru Pentium.

**Kontrolní otázka 7.1**

Popište blokové schéma procesoru Pentium.

**Kontrolní otázka 7.2**

Definujte frekvence na kterých jsou taktovány jednotlivé generace procesorů.

**Kontrolní otázka 7.3**

Popište architekturu NetBurst.

**Kontrolní otázka 7.4**

Co je to instrukční pipeline?

**Kontrolní otázka 7.5**

V čem se liší nová architektura Intel Core od předchozích?

8. Funkce a vlastnosti sběrnic



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- definovat pojem sběrnice a jejich základní parametry
- rozdělit sběrnice v počítači podle určité typologie
- popsat vývoj lokálních sběrnic
- charakterizovat sběrnici pro grafické rozhraní a jejich jednotlivé verze
- porovnat jednotlivé sběrnice podle datové šířky a přenosové rychlosti
- popsat vývoj nové sběrnice PCI express



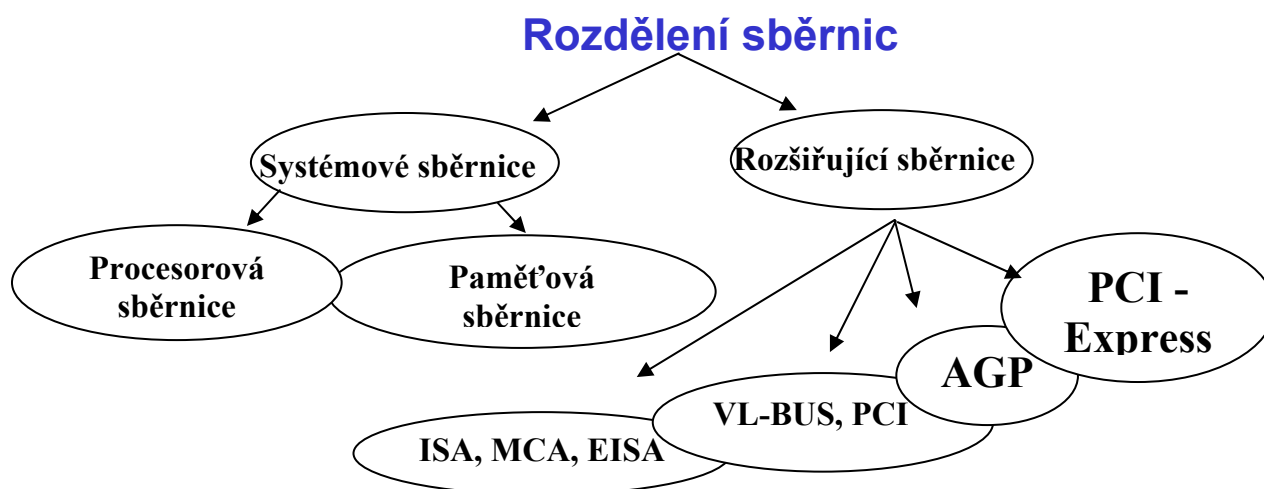
Výklad

8.1 Rozdělení sběrnic PC

Sběrnici označujeme jakoukoliv skupinu vodičů, která v systému přenese data z jedné části systému do druhé. Sběrnice jsou v počítači uspořádány hierarchicky, to znamená, že v systému nejpomalejší sběrnice (nižší úroveň) je vždy zapojena do rychlejší o úroveň výše. Na tyto sběrnice jsou poté připojovány zařízení, podle svého typu a rychlosti. Tato zařízení komunikují se systémem prostřednictvím těchto sběrnic a čipovou sadou základní desky, která vykonává v tomto případě funkci mostu mezi sběrnicemi. Mezi základní parametry každé sběrnice patří:

Parametr	Význam	Jednotka
Šířka přenosu	Počet bitů, které lze přenést po sběrnici v jeden okamžik	Bit
Frekvence	Maximální frekvence se kterou může sběrnice pracovat	Hz
Propustnost	Počet bytů přenesených za jednotku času	B/s

Tab. 8.1 Základní parametry sběrnic



Obr. 8.1 Souhrnné rozdělení sběrnic



Pojem k zapamatování: definice pojmu sběrnice

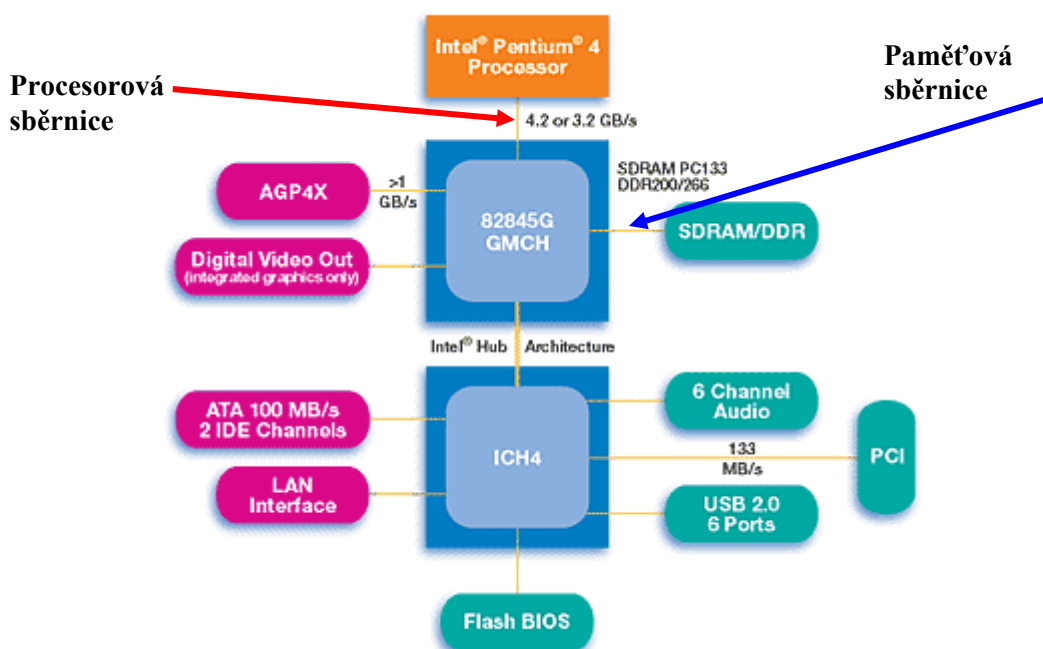
Pod pojmem sběrnice obecně rozumíme soustavu vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače. Pomocí těchto vodičů mezi sebou jednotlivé části počítače komunikují a přenášejí data.

- **Procesorová sběrnice**

Tato sběrnice slouží k přenosu dat mezi procesorem a čipovou sadou základní desky, přesněji mezi procesorem a obvodem *North Bridge* či rozbočovačem řadiče paměti čipové sady. Sběrnice pracuje na plné rychlosti základní desky (vnější rychlosti procesoru).

- **Paměťová sběrnice**

Tato sběrnice se využívá pro přenosy dat mezi procesorem a systémovou pamětí RAM. Sběrnice je připojena k obvodu *North Bridge* či řadiči rozbočovače paměti čipové sady a její rychlost závisí na typu paměťových modulů, které je čipová sada schopna podporovat.



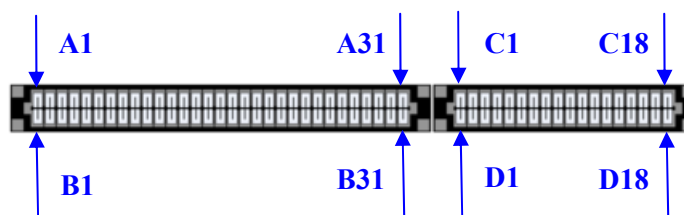
Obr. 8.2 Schéma čipové sady

S postupujícím vývojem počítačů bylo zřejmé, že původní sběrnice PC bus již svými možnostmi nedostačuje a může degradovat výkon celého počítače. S příchodem procesoru 80286 se tedy objevuje nový typ sběrnice označovaný jako ISA (*Industry Standard Architecture*), popř. AT bus. Tento typ rozšiřující sběrnice je vyroben s 16bitovou datovou sběrnici a 24bitovou adresovou sběrnici.

8.2 Lokální sběrnice

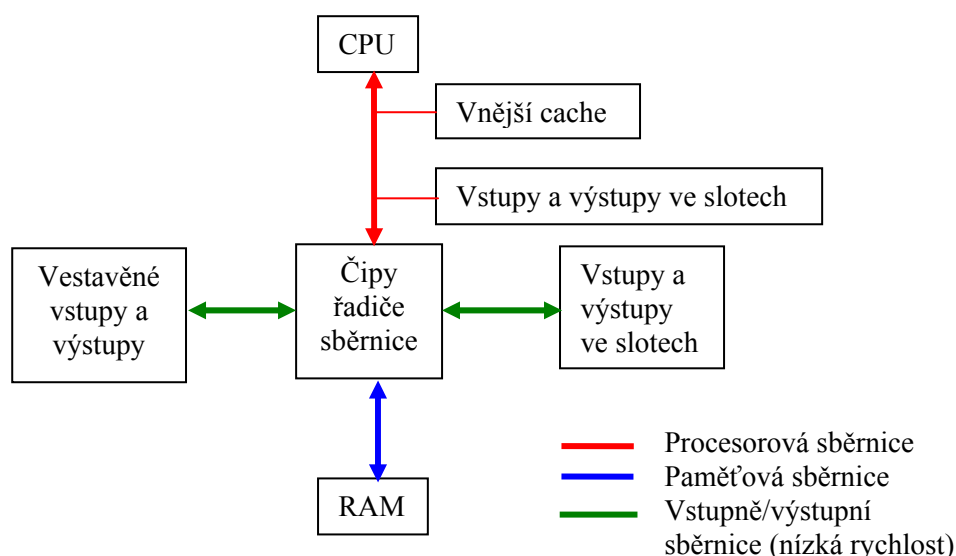
Sběrnice ISA je podobně jako předchozí *PC bus* navržena tak, aby plně odpovídala možnostem procesoru 80286. ISA dodržuje plnou zpětnou kompatibilitu s předešlou sběrnici PC bus. To znamená, že uživatel může používat přídatné karty určené pro PC bus i v počítači se sběrnici ISA. Kompatibilitu je dosaženo tak, že se stará 62 vodičová sběrnice rozšířila o dalších 36 vodičů a odpovídající slot se rozšířil o další konektor umístěný v jedné řadě hned za starším 8bitovým slotem pro PC bus. Takto vznikl nový 16bitový slot, který je umístěn na sběrnici ISA. Sběrnici ISA charakterizují následující vlastnosti:

- šířka přenosu 16 bitů, tj. během jedné operace je možné po sběrnici přenášet nejvýše 16bitovou informaci,
- rychlost 16 bitové verze byla standardizována na 8,33 MHz,
- k jednomu přenosu dat je zapotřebí 2 – 8 cyklů,
- pro přenos adresy vymezeno 24 vodičů odpovídajících 24bitové adresové sběrnici procesoru 80286,
- další 4 DMA kanály (DMA4 – DMA7),
- dalších 5 vodičů pro úroveň přerušení IRQ (IRQ10 – IRQ12, IRQ14 a IRQ15). Zbývající úrovně IRQ nepotřebují svou linku na sběrnici, protože jsou zapojeny následovně:
 - IRQ8 je spojeno přímo se systémovými hodinami,
 - IRQ9 je propojeno s IRQ2,
 - IRQ13 je určeno pro numerický koprocessor.



Obr. 8.3 98 pinový ISA Slot

Sběrnice typu ISA, MCA a EISA mají společný rys: poměrně malou rychlost. Omezení rychlosti je dědictvím z dob, kdy I/O sběrnice pracovaly na stejné rychlosti jako procesorová sběrnice.



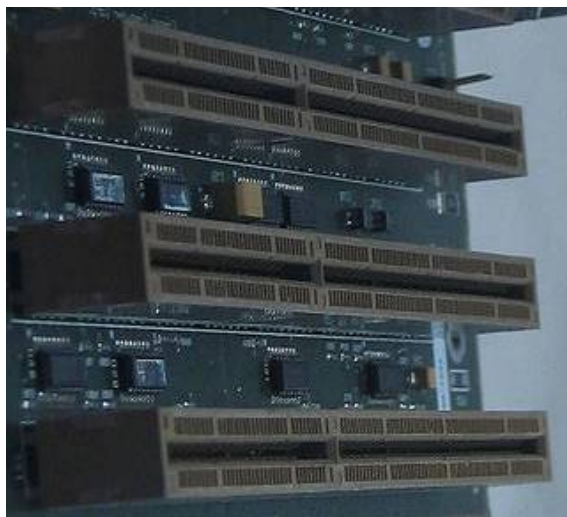
Obr. 8.4 Koncepční schéma PC se znázorněním sběrnic

Sběrnice MCA (*MicroChannel Architecture*) je typem sběrnice, který byl vyvinutý pro novou řadu počítačů firmy IBM s označením IBM PS/2. Hlavním cílem IBM bylo zrychlit přenos dat uvnitř počítače a snížit hladinu šumu na sběrnici.

Obrovskou nevýhodou a patrně i důvodem, proč se sběrnice MCA nerozšířila, je její nekompatibilita s ISA a to, že počítače PS/2 neměly osazenu pro zpětnou kompatibilitu i sběrnici ISA. Sběrnice MCA dovoluje běh s frekvencí 10 MHz a dovoluje přenášet data po 16 i 32 bitech. Jedná se tedy o rychlejší sběrnici s šířkou přenosu 32 bitů. Kromě toho má MCA i tzv. proudový režim, ve kterém dokáže současně přenášet 64 bitů. Šířka adresové části je v závislosti na procesoru počítače 24 bitů (pro 80286) nebo 32 bitů (pro 80386).

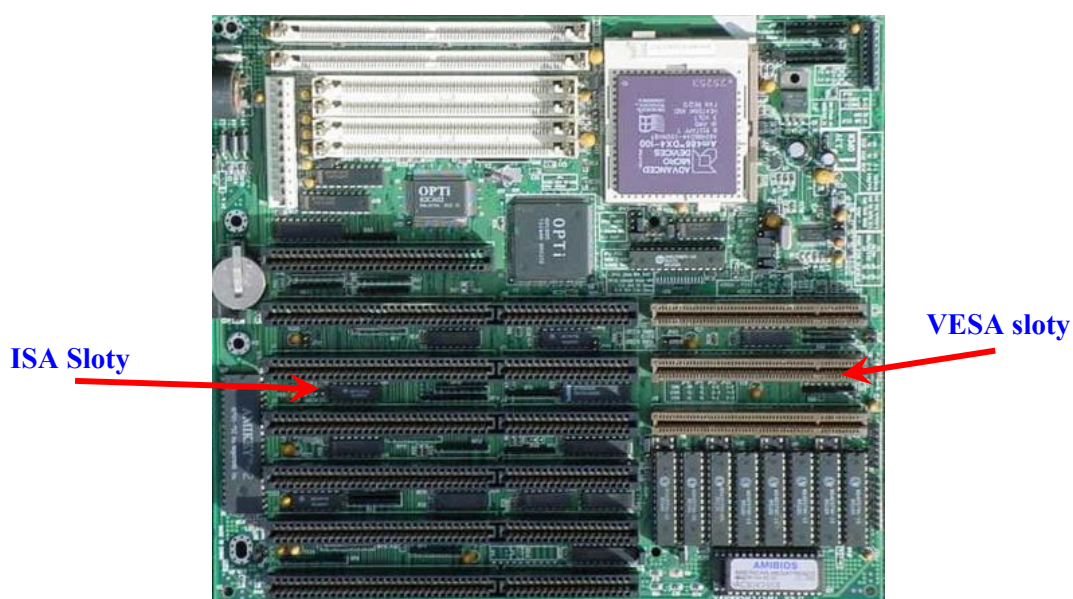
Sběrnice EISA (*Extended Industry Standard Architecture*) byla vyrobena 9 firmami (AST Research, Compaq, Epson, Hewlett Packard, NEC, Olivetti, Tandy, Wyse a Zenith) jako odpověď na sběrnici MCA. Záměrem bylo poskytnout sběrnici s vyšším výkonem, ale takovou, která by byla kompatibilní se sběrnici ISA. EISA byla uvedena na trh v roce 1989 a její základní vlastnosti jsou:

- šířka toku dat je 32 bitů,
- šířka adresy je 32 bitů,
- dovoluje programové nastavování přídatných desek,
- pracuje s frekvencí 8 MHz (z důvodů kompatibility s ISA),
- dovoluje busmastering.



Obr. 8.5 Sloty sběrnic EISA

Sběrnice VL bus (*VESA Local Bus*) byla navržena v roce 1992 konsorciem VESA (*Video Electronic Standards Association*) a jedná se o klasickou lokální sběrnici. Šířka přenosu dat i adresy je 32 bitů. VL bus podporuje maximálně 3 přídatné sloty. Čím vyšší je počet karet zasunutých na sběrnici VL bus, tím nižší je maximální frekvence, se kterou může sběrnice pracovat. Teoretická mez VL busu je 50 MHz.



Obr. 8.6 Základní deska se sběrnici VL bus

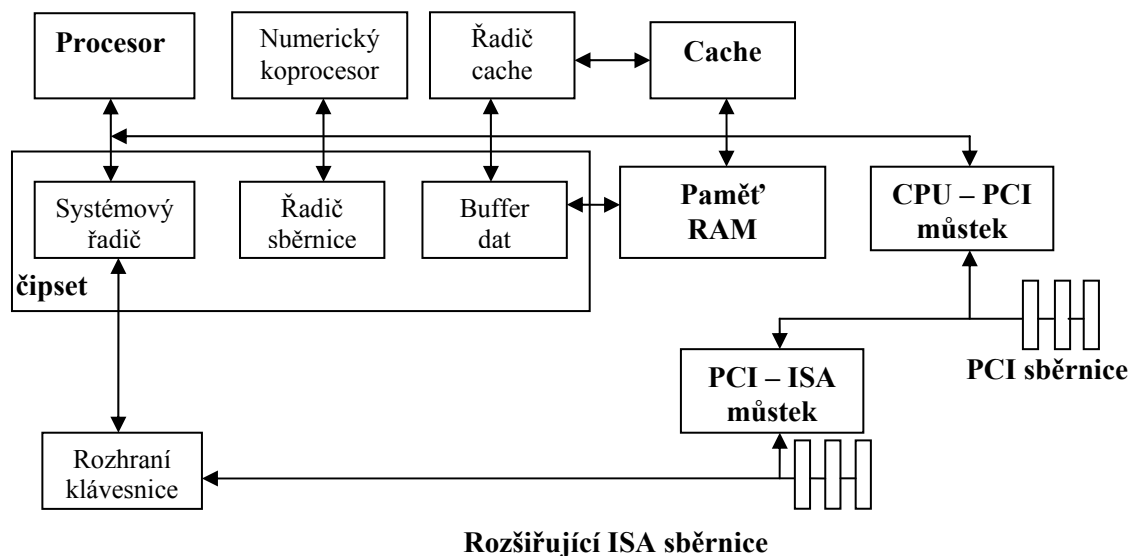
Prakticky je možné, aby pracovala s frekvencí 33 MHz při třech osazených přídavných kartách. Sběrnice umožňovala přenosovou rychlost až 133 MB/s, avšak bylo ji obtížné zaimplementovat do systému tak, aby splňoval kompatibilitu s různými procesory a tak po zveřejnění standardu sběrnice PCI sběrnice VL Bus zmizela z trhu.

Konektor VL busu se nachází v jedné řadě za 16bitovým konektorem ISA a má 2×58 vývodů. Sběrnice VL bus byla používána zejména u počítačů s procesorem 80486 a prvních počítačů s procesorem Pentium. Mezi nevýhody VL busu (oproti PCI) patří silná procesorová závislost způsobená přímým zapojením slotů VL busu na systémovou sběrnici. Tato závislost nedovoluje prakticky použít VL bus v jiném počítači, než je počítač s procesorem Intel nebo kompatibilním. Další nevýhodou je její práce s úrovněmi logiky TTL (5 V), zatímco nové procesory pracují s napětím 3,3 V a nižším. Sběrnice VL bus je vyráběna na základní desce vždy spolu se sběrnici ISA, protože při své práci využívá některých jejích signálů. Touto skutečností je zajištěna její kompatibilita s ISA, ale i její neoddělitelnost od sběrnice ISA.

PCI

V roce 1992 byl poprvé uveřejněn standard sběrnice PCI (*Peripheral Component Interconnect*). Tento standard byl postupně upravován, v roce 1993 na verzi PCI 2.0, která definuje již rozměry rozšiřujících karet. Verze 2.1 v roce 1995 definuje provoz na dvojnásobné frekvenci a standard 2.2 v roce 1999 se zabývá mechanickými částmi. Sběrnice se neustále vyvíjí a vycházejí z ní i další standardy PCI-X, Mini-PCI a také sériová nejnovější sběrnice PCI Express definovaná od roku 2002. PCI již není klasickou lokální sběrnici jako VL bus, ale je k systémové sběrnici připojena přes tzv. mezisběrníkový můstek. Toto řešení s sebou nese následující výhody:

- možnost použití sběrnice PCI i v jiných počítačích než jsou PC (např. Macintosh, DEC),
- můstek dovoluje provádět přizpůsobování napěťových úrovní.



Obr. 8.7 Blokové schéma s mezisběrníkovým můstkem

Sběrnice PCI má standardně šířku přenosu dat 32 bitů. Existuje však i verze PCI, která je 64bitová. Tato rozšířená verze se používá zejména pro řadiče diskových polí. Maximální frekvence, se kterou může tato sběrnice standardně pracovat, je 33 MHz, čímž je zajištěna propustnost sběrnice 132 MB/s (32 bitů) nebo 264 MB/s (64 bitů). Kromě sběrnice PCI, která pracuje s rychlostí 33 MHz, je vyráběna i rychlejší verze, která umožňuje pracovat při frekvenci 66 MHz.

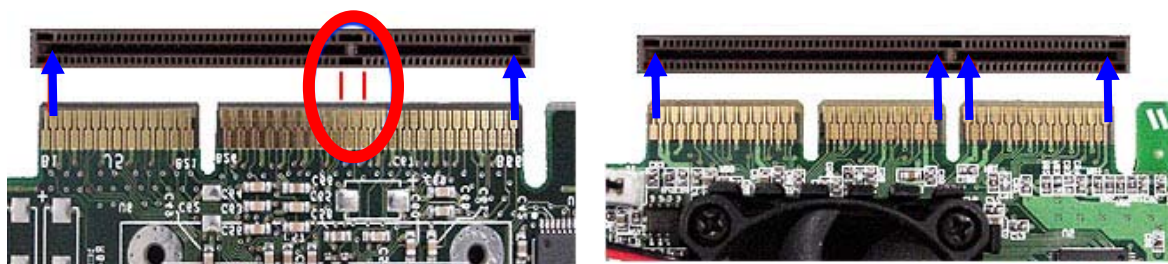
Podobně jako MCA a EISA a narozdíl od VL busu podporuje PCI busmastering, což umožňuje vyšší výkon počítače, protože přenosy řízené CPU vyžadují spoustu času. PCI dále podporuje standard *Plug & Play* vyvinutý výrobcí hardwaru téhož roku 1992, ten dovoluje velmi snadnou konfiguraci desek pro PCI, popř. jejich automatickou konfiguraci bez zásahu uživatele. Sběrnice PCI je používána u počítačů s procesory Pentium a vyššími.

8.3 AGP Port

Když se na konci roku 1996 objevila čipová sada i440LX, zbavila se sběrnice PCI největší zátěže. Objevil se dedikovaný slot **AGP** (*Accelerated Graphics Port*) taktovaný základní frekvencí 66MHz. Tehdy se jednalo o vskutku revoluční řešení, které vyhovovalo všem. Kvůli tomu, že se nejedná o sběrnici ale o jeden port, nemusí se grafická karta dělit o propustnost se zbytkem periferií. Hlavním důvodem přechodu mimo sběrnici PCI nebylo původně další navyšování grafického výkonu, ale zajištění přímého přístupu grafické kartě do paměti a zrychlení komunikace s procesorem. V průběhu doby se objevilo několik typů AGP, seřazených v tabulce 8.2. Index za AGP udává, kolik dat je možné přenést při jednom taktu sběrnice. V portu AGP není možné provozovat všechny typy grafických karet, známým faktem je nekompatibilita grafických karet AGP 1x a 2x s novými sloty AGP 4x a 8x. Důležitým faktorem v oblasti verzí AGP je fakt, že bychom si neměli plést verzi AGP multiplikátoru s verzí AGP specifikace. Specifikace AGP 1.0 podporuje verze multiplikátoru 1x a 2x na signálech o velikosti napětí 3,3 V. Specifikace AGP 2.0 podporuje již verzi AGP 4x a předchozí verze, avšak na napěťové úrovni 1,5 V. Poslední specifikace AGP 3.0 již podporuje pouze multiplikátory 4x a 8x na napětí 0,8 V.

Typ AGP	Rok uvedení standardu	Napětí na sběrnici	Propustnost
AGP 1x	1996	3,3 V	266 MB/s
AGP 2x	1996	3,3 V	533 MB/s
AGP 4x	1998	1,5 V	1066 MB/s
AGP 8x	2000	0,8 V	2133 MB/s

Tab. 8.2 Propustnost jednotlivých verzí AGP

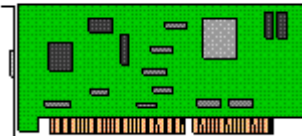
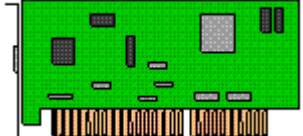
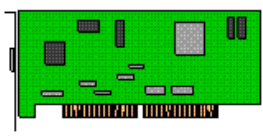
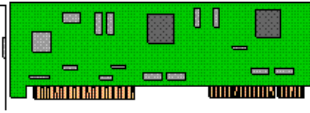
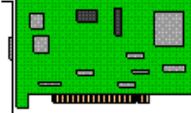


Obr. 8.8 Nekompatibilita rozhraní je zajištěna klíčovými výřezy

Typ sběrnice	Šířka v bitech	Rychlost v MHz	Přenos dat za cyklus	Přenosová rychlost MB/s
ISA 8bitová	8	8,33	1/2	4,17
ISA 16bitová	16	8,33	1/2	8,33
Řadič disk. mech.	1	0,5	1	0,0625
EISA	32	8,33	1	33
VLBUS	32	33	1	133
PCI	32	33	1	133
PCI 66MHz	32	66	1	266
PCI 64 bitová	64	66	1	533
AGP	32	66	1	266
AGP 2x	32	66	2	533

AGP 4x	32	66	4	1066
AGP 8x	32	66	8	2133
RS232	1	0,1152	1/10	0,01152
USB 1.1	1	12	1	1,5
USB 2.0	1	480	1	60
IEEE1394 S200	1	200	1	25
Ultra ATA 33	16	8,33	2	33
Ultra ATA 66	16	16,67	2	66
Ultra ATA 100	16	25	2	100
Ultra ATA 133	16	33	2	133
SATA 150	1	1500	1	150
SATA 300	1	3000	1	300
SCSI	8	5	1	5
SCSI Ultra	8	20	1	20
SCSI Ultra 2	8	40	1	40
SCSI Ultra 3 (160)	16	40	2	80 (160)
Paměť FPM	64	22	1	177
Paměť EDO	64	33	1	266
Paměť SDRAM PC100	64	100	1	800
Paměť DDR PC3200	64	200	2	3200
Paměť RDRAM 8500	64	533	2	8533
33 MHz FSB 486	32	33	1	133
66 MHz FSB P I	64	66	1	533
100 MHz FSB PII	64	100	1	800
400 MHz FSB P4	64	100	4	3200
533 MHz FSB P 4	64	133	4	4266

Tab. 8. 3 Srovnání parametrů sběrnic a rozhraní v PC

Souhrnný přehled sběrnic		
ISA	EISA	MCA
		
8, 16 bitů	32 bitů	32 bitů
8-10 MHz	8-10 MHz	5-20 MHz
VLBUS	PCI	AGP
		
32 bitů	32, 64 bitů	32 bitů
40 MHz	33 MHz	66 – 533 Mhz

Obr. 8.9 Přehled a parametry sběrnic

8.4 PCI Express

Jak už to v oblasti počítačů bývá, každá technologie se dostane dříve či později na hranici svých možností a poté musí přijít jiná. Obdobně je tomu i s lokální sběrnicí PCI. Sběrnice PCI byla používána jako hlavní sběrnice pro připojení periferii u osobních počítačů typu PC více než deset let. V roce 1993, kdy byl zveřejněn standard 2.0 s již pevně definovanými parametry rozšiřujících karet, se téměř okamžitě prosadila jako průmyslový standard a ukončila používání sběrnic ISA, VLB (*Vesa Local Bus*), EISA i sběrnici MCA (*MicroChannel Architecture*) od IBM. Nejvýše taktovaná verze měla propustnost 133 MB/s, což s rezervou pokrývalo tehdejší potřeby. Největšími spotřebiteli propustnosti jsou samozřejmě grafické karty, a tak brzy sběrnice přestala vyhovovat a grafická karta se přesunula na port AGP. Ten ovšem také nelze donekonečna zrychlovat, a tak se vývojáři hardwaru začali poohlížet po dalších možnostech. Dobrou možností se zpočátku jevila sběrnice PCI-X, která zdvojnásobila frekvenci na 133 MHz s přenosovou rychlostí přes 1 GB/s a ve verzi 2.0 dokonce 266 MHz a 2,15 GB/s. Při vysokých taktech se však objevují problémy s implementací sběrnice, signál je náchylný na šum a zkreslení, hrozí přeslechy (tzv. *crosstalk*), dochází k problému s vedením jednotlivých drah sběrnice, což při vyřešení problémů vede k vyšší ceně základních desek. Pokud potřebujeme více zásuvných karet, musíme použít více řadičů. Je evidentní, že PCI-X není řešením vhodným pro běžné počítače (u serveru vyšší cena a omezený počet slotů nevadí). A tak začal vznikat požadavek na novou sběrnici označovanou jako **3GIO** (*Third Generation Input and Output*) třetí generace vstupně/výstupních zařízení. Požadavky na architekturu 3GIO:

- Sjedení přístupu pro více platform (desktop, mobilní zařízení, servery, pracovní stanice).
- Podpora různých druhů připojení (čip-čip, karta-karta přes konektor nebo kabel).
- Nové mechanické uspořádání.
- Zachování současného softwarového modelu:
 - Detekce a konfigurace PCI Express zařízení pomocí současných mechanismů.
 - Bootování existujících operačních systémů bez nutnosti modifikovat SW.
 - Podpora existujících driverů zařízení.
 - Podpora nových funkcí - vyžaduje úpravu konfiguračního mechanismu.
- Výkon:
 - Nízká režie na vlastní komunikaci a rychlá odezva.
 - Vysoká přenosová rychlost na pin, z toho plynoucí omezení počtu pinů konektoru.
 - Škálovatelná šířka přenosové trasy.
- Speciální vlastnosti:
 - Podpora různých datových typů a pravidel pro přenosy dat.
 - PowerManagement a úsporné režimy práce.
 - Identifikovat podporované režimy pro dané zařízení.
 - Přechod zařízení do daného stavu.
 - Identifikace současného stavu zařízení.
 - Generování wakeup sekvencí pro zdroje.
 - Postupný náběh napájecích napětí.
- HotPlug a HotSwap:
 - Zachování podpory současných zařízení typu HotPlug a HotSwap.
 - Zavedení ryziho standardu HotPlug a HotSwap bez pomocných signálů.
 - Jeden softwarový model pro všechny platformy.
- Datová integrita:
 - Podpora detekce chyb na linkové vrstvě jak v datech tak i v paketech.
- Oprava chyb:
 - Podpora současné verze PCI.
 - Rozšířené generování chyb pro jejich lepší lokalizaci.
- Nezávislost na typu přenosového média a napěťových úrovních.
- Podpora jednoduchého testovacího mechanismu.

Omezením architektury PCI je nejednoznačné definování velikosti bufferu, takže velikost přenášeného bloku dat není omezená a může tak docházet k odpojování zařízení od sběrnice, PCI zařízení tak může vkládat velký počet čekacích stavů. Přenos dat je omezen pouze na transakce v délce 32/64 bitů a na sběrnici může probíhat pouze jedna transakce jen jedním směrem. Paměťově orientované operace vyžadují spolupráci s cache procesoru, aby byla zajištěna koherence dat, což má za následek snížení výkonu CPU a PCI.

Sběrnice PCI Express se významně odlišuje od svých předchůdců, protože mění celkovou filozofii pohledu na sběrnice tak, jak je známe do dnešních dnů. Obvyklá představa sběrnice vycházela z propojení několika zařízení sdílející stejnou skupinou vodičů (paralelní nebo sériové). Některé typy sběrnic byly doplněny o vybrané vodiče sloužící pro dekódování daného zařízení, informaci o jeho stavu a synchronizaci hodinovým signálem. Model fyzické vrstvy sběrnice PCI Express vychází spíše ze síťové architektury než z architektury PCI nebo PCI-X. Podobnost architektury PCI Express najdeme především v dělení vrstev, což připomíná síťový model ISO-OSI. PCI Express je sběrnici typu point-to-point, která přenáší data po paketech. Charakter přenosu dat je tedy sériový, data jsou enkodována 8b/10b. Základní přenosová rychlost každé linky (jsou dvě, pro každý směr jedna) je 2,5Gb/s. Základní sběrnice x1 dokáže data přenášet tak (data na vstupu je "rozloží" a pak zase "složí"), jakoby putovaly po paralelní virtuální sběrnici o volitelné šíři: 1, 2, 4, 12, 16 a 32 bitů. **Jako náhrada 32bit PCI tedy stačí pouze PCIe 1x.**



Pojem k zapamatování: PCI Express

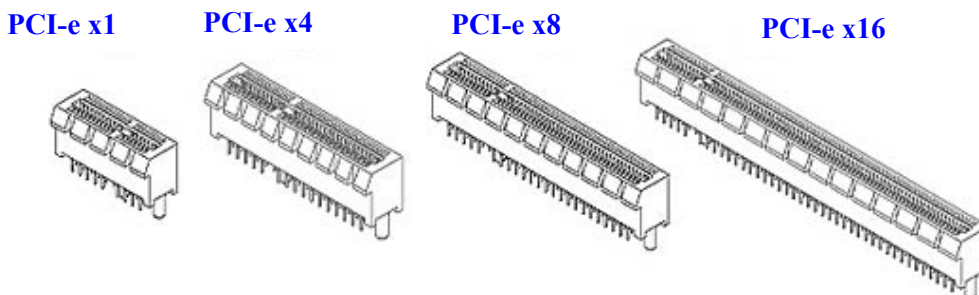
Sběrnici PCI Express v konkrétním počítači tedy můžeme vnímat jako soustavu několika nezávislých obousměrných propojení typu 1:1, které dokážou data přenášet tak, jako by tyto sběrnice byly široké 32 bitů. V základní verzi máme k dispozici sběrnici o propustnosti 250 MB/s v každém směru (PCI Express x1), což je prakticky dvojnásobná propustnost sběrnice PCI.

Nesmíme také zapomenout, že v případě klasické sběrnice PCI je propustnost celé sběrnice sdílena veškerými připojenými periferiemi, zatímco PCIe má celé pásmo pro sebe. Situace kdy jedno zařízení bude omezovat jiné, by tedy neměla nastat.

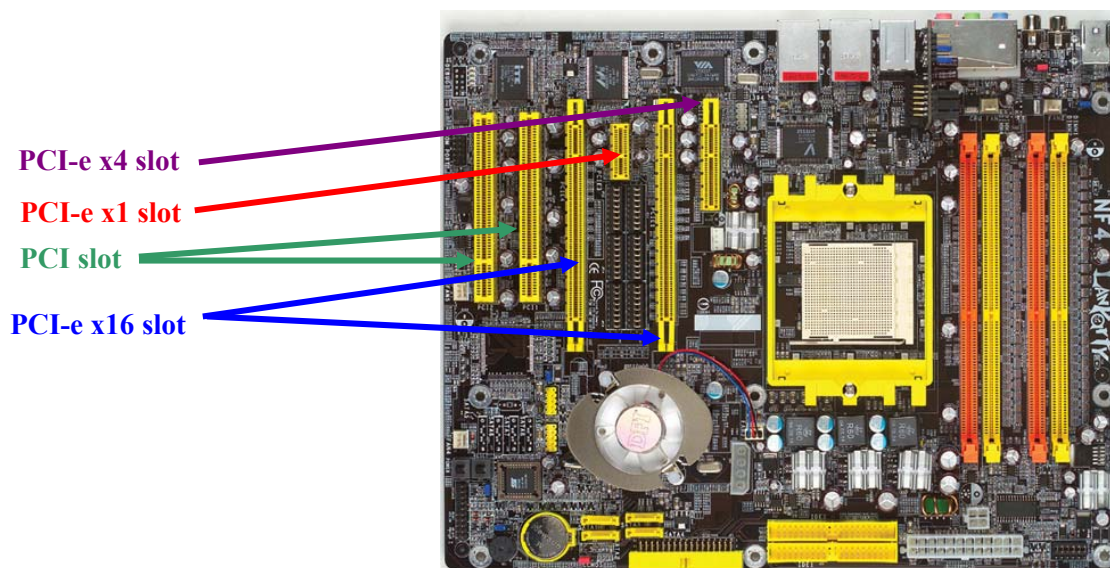
Typ PCI Express	Propustnost
PCI-e x1	250 MB/s
PCI-e x2	500 MB/s
PCI-e x4	1000 MB/s
PCI-e x8	2000 MB/s
PCI-e x16	4000 MB/s
PCI-e x32	8000 MB/s

Tab. 8.4 Propustnost jednotlivých variant PCI Express

Definovanými sloty jsou x1, x4, x8 a x16:



Obr. 8.10 Varianty jednotlivých PCI-x slotů



Obr. 8.11 Základní deska s variantami slotů sběrnice PCI a PCI Express



Další zdroje

1. Gook, M. *Hardwarová rozhraní*, Brno, Computer Press 2006, počet stran: 464. ISBN 80-251-1019-2.
2. Horák, J. *Bios a Setup*, Brno, Computer Press 2004, počet stran: 132. ISBN 80-251-0148-7.
3. Aspinwall, J. *IRQ, DMA a I/O*, Brno, Computer Press 2000, počet stran: 288. ISBN 80-722-6264-5.
4. Vlach, J, Vlachová V. *Počítačová rozhraní*, Praha, Ben 2002, počet stran: 176. ISBN 80-7300-010-5.
5. PCI Specifications Documents [online] dostupný z <http://members.datafast.net.au/dft0802/specs.htm>
6. IBM RedBooks [online] dostupný z <http://www.redbooks.ibm.com/abstracts/tips0456.html>
7. PCI Express Technology Dell whitepaper. [online] dostupný z http://www.dell.com/downloads/global/vectors/2004_pciexpress.pdf
8. How Stuff Works server - How PCI Express works. [online] dostupný z <http://computer.howstuffworks.com/pci-express.htm>
9. How Stuff Works server - How AGP works. [online] dostupný z <http://computer.howstuffworks.com/agp.htm>



Shrnutí kapitoly

Sběrnici označujeme jakoukoliv skupinu vodičů, která v systému přenese data z jedné části systému do druhé. Sběrnice jsou v počítači uspořádány hierarchicky, to znamená, že v systému nejpomalejší sběrnice (nižší úroveň) je vždy zapojena do rychlejší o úroveň výše. Na tyto sběrnice jsou poté připojovány zařízení podle svého typu a rychlosti. Mezi základní parametry každé sběrnice patří:

Šířka přenosu, frekvence a datová propustnost. Procesorová sběrnice slouží k přenosu dat mezi procesorem a čipovou sadou základní desky, přesněji mezi procesorem a obvodem *North Bridge* či rozbočovačem řadiče paměti čipové sady. Paměťová sběrnice se využívá pro přenosy dat mezi procesorem a systémovou pamětí RAM.

Sběrnice typu ISA, MCA a EISA mají společný rys: poměrně malou rychlost. Omezení rychlosti je dědictvím z dob, kdy I/O sběrnice pracovaly na stejné rychlosti jako procesorová sběrnice.

PCI již není klasickou lokální sběrnici jako VL bus, ale je k systémové sběrnici připojena přes tzv. mezisběrníkový můstek. Toto řešení s sebou nese následující výhody jako možnost použití sběrnice PCI i v jiných počítačích než jsou PC (např. Macintosh, DEC) a možnost provádět přizpůsobování napětíových úrovní.

Sběrnice PCI má standardně šířku přenosu dat 32 bitů. Existuje však i verze PCI, která je 64bitová. Tato rozšířená verze se používá zejména pro řadiče diskových polí.

Maximální frekvence, se kterou může tato sběrnice standardně pracovat, je 33 MHz, čímž je zajištěna propustnost sběrnice 132 MB/s (32 bitů) nebo 264 MB/s (64 bitů). Kromě sběrnice PCI, která pracuje s rychlostí 33 MHz, je vyráběna i rychlejší verze, která umožňuje pracovat při frekvenci 66 MHz.

V roce 1996 se na trhu objevil slot AGP (*Accelerated Graphics Port*) taktovaný základní frekvencí 66 MHz. Specifikace AGP 1.0 podporuje verze multiplikátoru 1x a 2x na signálech o velikosti napětí 3,3 V. Specifikace AGP 2.0 podporuje již verzi AGP 4x a předchozí verze, avšak na napětíové úrovni 1,5 V. Poslední specifikace AGP 3.0 již podporuje pouze multiplikátory 4x a 8x na napětí 0,8 V.

Omezením architektury PCI je nejednoznačné definování velikosti bufferu, takže velikost přenášeného bloku dat není omezená a může tak docházet k odpojování zařízení od sběrnice, PCI zařízení tak může vkládat velký počet čekacích stavů. Přenos dat je omezen pouze na transakce v délce 32/64 bitů a na sběrnici může probíhat pouze jedna transakce jen jedním směrem. A tak začal vznikat požadavek na novou sběrnici označovanou jako 3GIO (*Third Generation Input and Output*) třetí generace vstupně/výstupních zařízení. Touto sběrnici je PCI Express, můžeme ji vnímat jako soustavu několika nezávislých obousměrných propojení typu 1:1 které dokážou data přenášet tak, jako by tyto sběrnice byly široké 32 bitů. V základní verzi máme k dispozici sběrnici o propustnosti 250 MB/s v každém směru (PCI Express x1), což je prakticky dvojnásobná propustnost sběrnice PCI. Propustnost PCI-e verze x16 dosahuje dokonce 4000 MB/s.



Kontrolní otázka 8.1

Uveďte základní parametry sběrnice.



Kontrolní otázka 8.2

Obecně rozdělte sběrnice v PC.



Kontrolní otázka 8.3

Proveďte rozbor vývoje lokálních sběrnice.



Kontrolní otázka 8.4

Definujte hlavní charakteristiku sběrnice PCI.



Kontrolní otázka 8.5

Charakterizujte sběrnici AGP.



Kontrolní otázka 8.6

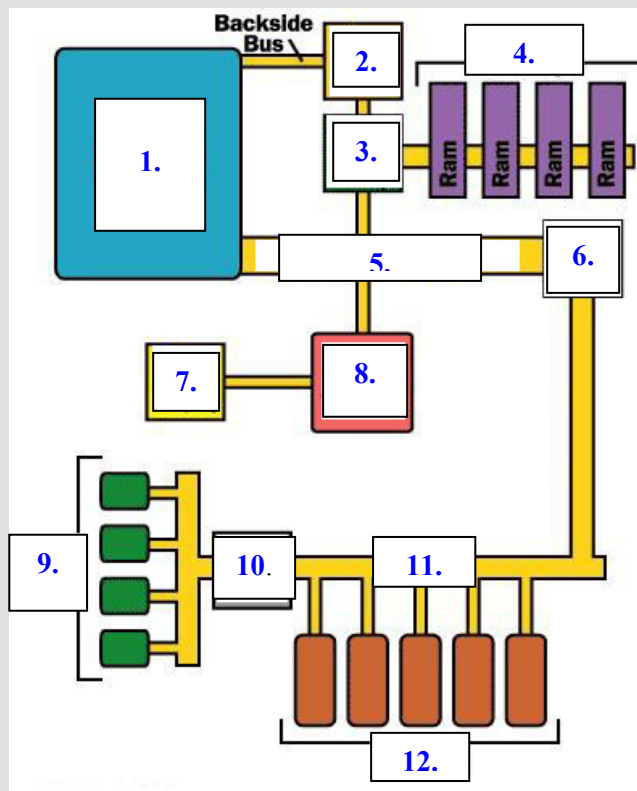
Charakterizujte sběrnici PCI Express.



Úkol k řešení 8.1

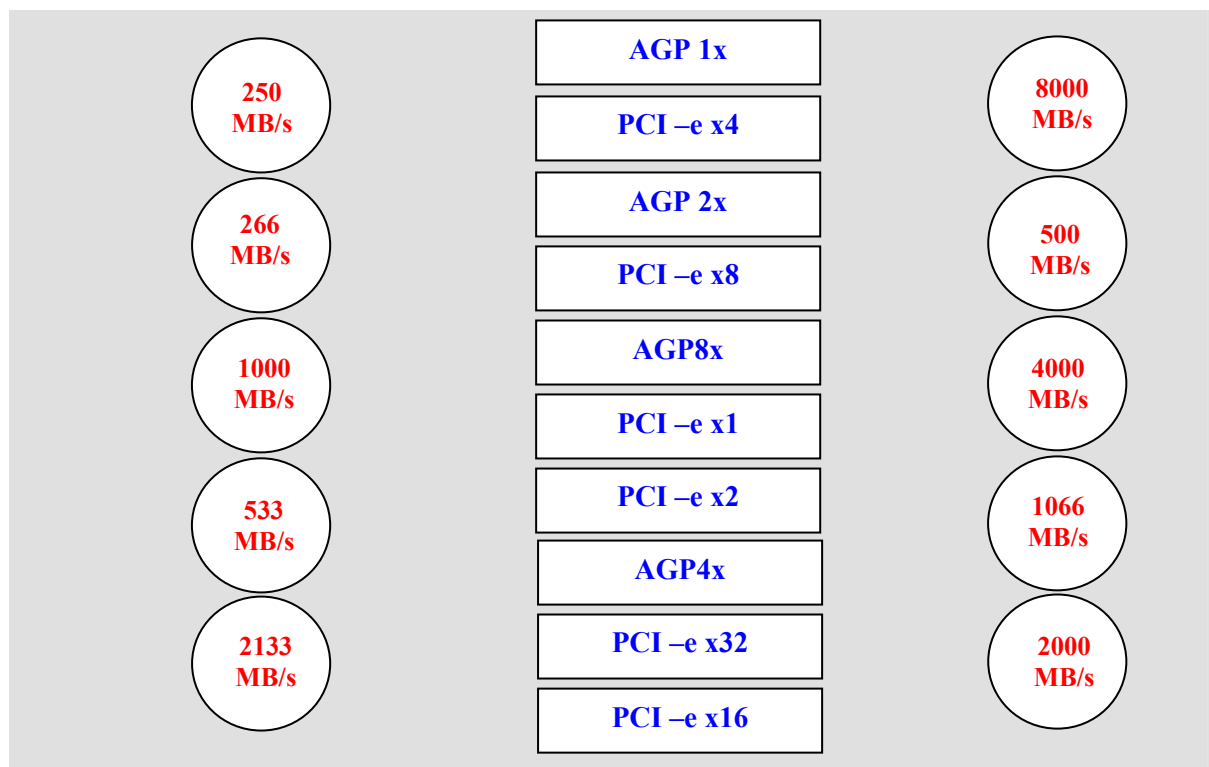
Doplňte blokové schéma PC se sběrníci PCI.

- A) AGP čipset
- B) Grafická karta
- C) Systémová paměť
- D) ISA sloty
- E) PCI sloty
- F) CPU
- G) Sběrníkový můstek
- H) Sběrníkový můstek
- I) Cache
- J) Řadič paměti
- K) FSB
- L) PCI sběrnice



Úkol k řešení 8.2

Přiřaďte propustnosti konkrétním sběrnicím AGP a PCI Express.



9. Vstupně/výstupní porty



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- charakterizovat sběrnici USB a rozhraní WireFire
- popsat sériový port, typy zařízení a přístup k sériovému portu
- pomocí definovaných funkcí naprogramovat úlohu v jazyce C na propojení dvou PC pomocí sériového a paralelního portu
- charakterizovat paralelní port, jeho jednotlivé verze, standardní použití a propojení



Výklad

Úkolem této kapitoly je seznámení studentů se sériovou a paralelní komunikací pomocí portů PC. Nacházíme se v devátém týdnu semestru a po procvičení úloh z jazyka C nastal vhodný čas na procvičení jazyka na praktických úlohách. Uvidíme, že i zdánlivě náročný projekt propojení dvou počítačů pomocí portu a naprogramování řízení jejich vzájemné komunikace můžeme zvládnout s dosaženými zkušenostmi, které jsme s jazykem C doposud nabyli. S pomocí některých předem připravených funkcí budete schopni tento projekt úspěšně zvládnout. Řešený příklad obdobnou úlohu zadání projektů řeší a je přiložen na CD k učebnímu textu. Na přiloženém CD jsou varianty příkladů pro semestrální projekt.

9.1 Sériový versus paralelní přenos

Sběrnice se ve svém nejjednodušším rozdělení mohou dělit na sériové a paralelní. O různých typech sběrnic jsme již hovořili mnohokrát v předchozích kapitolách. Popsali jsme paralelní architektury, které používají více vodičů (8, 16, 32...). V kapitole věnované rozhraním pevných disků se budeme věnovat také paralelním rozhraním IDE a SCSI. Do popředí se však stále více dostávají rozhraní sériová jako náhrada zmíněných paralelních, např. SATA (viz. Kapitola 11). Při stejné frekvenci je samozřejmě paralelní rozhraní rychlejší, neboť po sériové sběrnici je v jednom časovém okamžiku přenášen jen jeden bit. Podstatným faktem ale je možnost zrychlování sběrnice, tady paralelní naráží na své technologické omezení a maxima, protože dochází k problémům s kmitáním signálů a jejich zpožděním. Tyto problémy jsou dány fyzikálními vlastnostmi vodičů. Se zvyšující frekvencí sběrnice roste vzájemná interference signálů. Vyšší počet vodičů tak vyžaduje nutnou ochranu proti interferenci a cena takových vedení je dalším limitujícím faktorem. Naopak sériová komunikace nemá problém s výrazným zvýšením frekvence. Srovnáme-li maximální rychlost paralelního portu *EPP/ECP* 2 MB/s zhruba s rychlostí *USB* sběrnice verze 2.0 s podporou rychlosti až 60 MB/s je jasné, které porty v osobních počítačích zvítězily.

9.2 USB a FireWire port

První verze standardu byla uvedena v roce 1996. Ihned následovala modifikace 1.1. Tento standard podporuje přenosovou rychlost 12 Mb/s, což převedeno na rychlost v bytech odpovídá 1,5 MB/s. V roce 2000 byl uveden standard USB 2.0, který je 40krát rychlejší než jeho předchůdce. Sběrnice

umožňuje připojit až 127 zařízení, přičemž je založena na hvězdicové topologii. Základem je USB hub neboli rozbočovač, který je buďto interní součástí počítače anebo může figurovat jako externí zařízení. Z fyzikálního hlediska přenosu jsou data přenášena po sběrnici metodou NRZI (*Non Return Zero Invert*). To znamená, že u jednotlivých bitů 0 a 1 reprezentovaných opačnou hodnotou napětí nedochází k žádnému návratu k nulovému napětí. Jednička je tedy přenášena bez jakékoliv změny napětí a nula je prezentována změnou napětí.

Alternativou USB je použití standardu IEEE 1394 (*FireWire*), který dosahuje rychlosti 400 Mb/s a navrhovanou další rychlostí je 800 Mb/s. Na rozdíl od USB není ale tak rozšířen. Důvodem je existence USB 2.0 standardu, který je rychlejší než současná 400 megabitová varianta. Avšak v nedávné minulosti byl standard velice oblíbeným a prakticky jedinou variantou pro propojování digitálních videokamer a vysokorychlostních zařízení. Obliba tohoto standardu u výrobců digitálních komponent přetrvává a se standardem *FireWire* se musí stále počítat. Zvláště pak proto, že jeho navrhované další rychlosti dosahují až 1 GB/s, zatímco USB se zvyšováním rychlosti zatím nepočítá, spíše se standard bude orientovat na jiné technologické novinky jako např. *wireless USB*. Pro rozhraní *FireWire* platí oficiální označení standardu IEEE-1394 z roku 1995. Tento příhodnější název je z dílny firmy Apple. Jiným názvem *i.Link* ho označuje firma Sony.



Obr. 9.1 Konektor USB a FireWire



Korespondenční úkol – další parametry USB a FireWire

Vyhledejte maximální délky kabelu mezi dvěma zařízeními u standardu USB 1.1, 2.0 a FireWire.

Vyhledejte maximální počet připojitelných zařízení standardu IEEE – 1394.



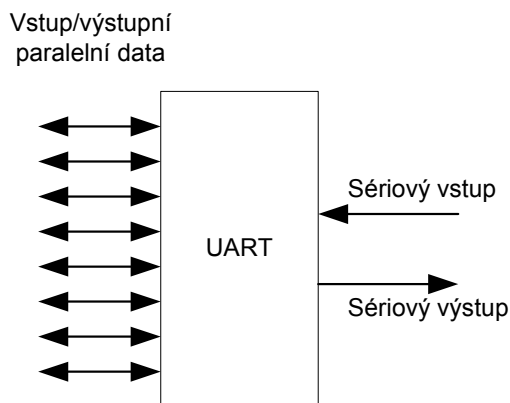
Korespondenční úkol – připojování zařízení

Mohou být zařízení USB připojována za chodu? A jak je to se zařízeními připojenými rozhraním FireWire?

9.3 Sériový port

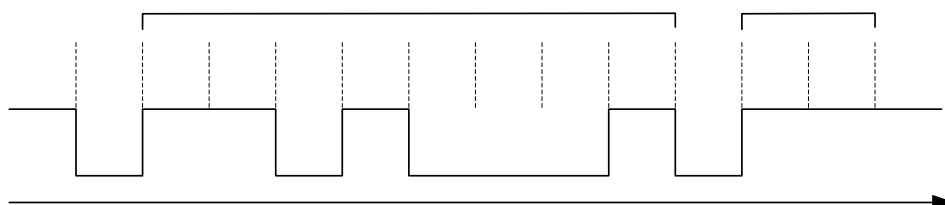
Jedno z vůbec nejstarších rozhraní je sériové rozhraní RS 232. Poprvé se objevilo již v roce 1960. Rozhraní RS232, také označované jako V.24 (podle normy CCITT) nebo DIN66020, je běžně používáno v personálních počítačích pro universální připojení jednoduchých a relativně pomalých zařízení jako jsou myši, klávesnice, nebo plotery.

RS232 je typickým představitelem asynchronní sériové komunikace. Přenos dat se provádí postupně po jednotlivých bitech – tedy sériově, na rozdíl od souběžného přenosu u paralelního portu. Pro nejjednodušší propojení dvou komunikujících sériových zařízení proto stačí dva vodiče – jeden pro příjem, druhý pro vysílání. Sériové rozhraní je realizováno speciálním obvodem (dříve 8252, dnes 16750). Jedním z nejdůležitějších úkolů obvodu je převod paralelních dat, nejčastěji 8bitového bajtu, na sériový tok bitů a naopak. Schematicky lze proces znázornit následujícím způsobem (obr. 9.2):



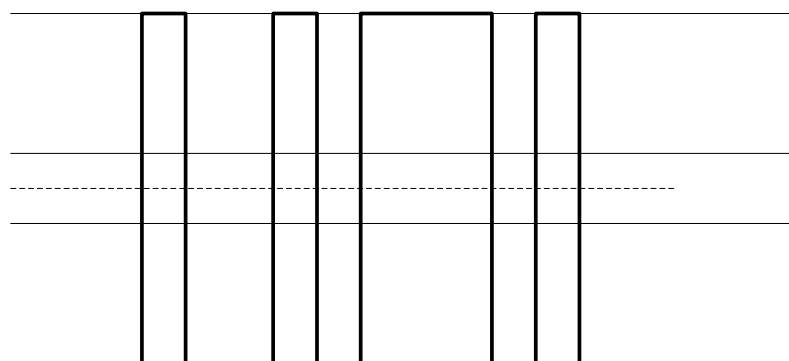
Obr. 9.2 Funkce převodníku UART

Sériovou komunikaci často realizujeme asynchronní formou. Znamená to, že data nejsou synchronizovány žádným hodinovým signálem udávajícím příchod nových dat a jednotlivých bitů. Obě zařízení musí být pro úspěšnou komunikaci nastaveny na ekvivalentní přenosovou rychlost udávanou v bitech za sekundu [bps]. Stejný takt obou zařízení však nezaručí korektní interpretaci příchozího signálu. Zajištění synchronizace se provede přidáním speciálních úvodních a ukončovacích bitů k vlastním datům. Každý vyslaný bajt v délce 5-8 bitů je doplněn tzv. **Start bitem** a **Stop bity** (1, 1,5 nebo 2). Sekvenci bitů lze navíc doplnit o paritní bit – sudý nebo lichý. Paritní bit umožňuje provést jednoduchou kontrolu přenesených informací. Sudá parita nabývá hodnotu 1 v případě, že data obsahují sudý počet bitů o hodnotě 1, lichá pak nabude hodnoty pro lichý počet jedničkových bitů. Ukázku signálu, který vznikne vysláním osmibitové hodnoty 0x0D1 = 11010001 se dvěma stopbity a lichou paritou, vidíme na obr. 9.3.



Obr. 9.3 Serializovaný signál

Napětíové úrovně signálů rozhraní RS232 se však nepohybují ve standardních TTL hodnotách (0-5V). Pro větší odolnost proti rušení a k dosažení větších vzdáleností mezi komunikujícími zařízeními jsou tyto úrovně převedeny na napětí ± 12 V. Signál z obrázku 9.3 bude ve skutečnosti na vstup/výstupních vodičích vypadat takto:



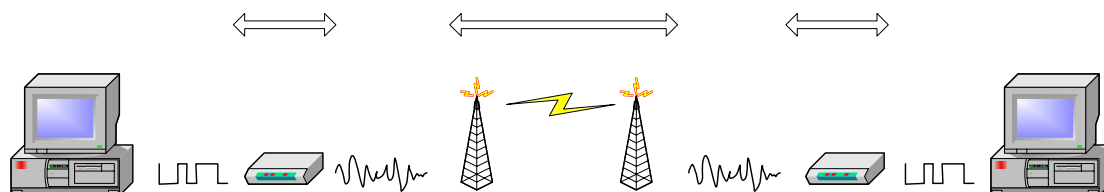
Obr. 9.4 Signál sériového portu v napětíových úrovních rozhraní RS232

9.4 Typy sériových zařízení

Systémy komunikující po rozhraní RS232 lze rozdělit obecně do dvou kategorií:

- DTE (Data Terminal Ready) zařízení – zařízení je zdrojem a příjemcem dat.
- DCE (Data Communication Equipment) zařízení – zařízení vytváří komunikační kanál mezi dvěma DTE zařízeními.

Mezi typická DTE zařízení patří personální počítač nebo tiskárna. Nejběžnějším DCE zařízením je telefonní modem. Konfigurace s DTE a DCE zařízeními je na obr. 9.5.



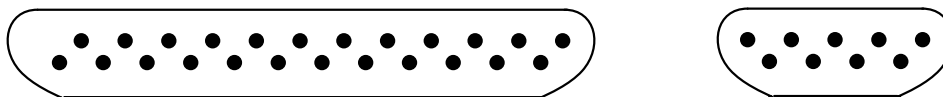
Obr. 9.5. Typy koncových zařízení

Aby mohla spolu jednotlivá zařízení komunikovat, musí být k dispozici sada doplňujících signálů, souhrnně nazývaných signály řízení modemu. Jejich účelem je hardwarová synchronizace komunikujících stran. Celkově je v sériovém rozhraní RS232 definováno 9 různých signálů. Jejich seznam je uveden v tabulce 9.1.

Signál	Význam
TxD	Transmit Data – signál se používá pro vysílání dat
RxD	Receive Data – signál se používá pro příjem dat
RTS	Request to Send – DTE zařízení signalizuje tímto vodičem, že chce začít vysílat data.
CTS	Clear to Send – DCE zařízení signalizuje tímto vodičem připravenost k vyslání dat. Signál se také používá jako odpověď na RTS.
DTR	Data Terminal Ready – signál je používán DTE zařízením k oznámení, že zařízení je připojeno a připraveno začít komunikaci.
DSR	Data Set Ready – jde o protisignál k DTR a je používán DCE zařízením k oznámení, že je připraveno ke komunikaci.
DCD	Data Carrier Detect – signál je využíván DCE zařízením a indikuje DTE zařízení, modem přijal „nosný signál“ a očekává se příjem dat.
RI	Ring Indicator – signálem DCE zařízení oznamuje DTE zařízení, že zvoní telefon připojený k modemu (DCE zařízení).
GND	Signal Ground – signálová zem. Vodič musí být vždy mezi zařízeními propojen.

Tab. 9.1 Pojmenování a význam signálů RS232 rozhraní

Prakticky jsou signály rozhraní nejčastěji přístupné pomocí Cannon konektorů s 25 nebo 9 piny. Zapojení obou variant vidíme na obr. 9.6 a v tabulce 9.2.



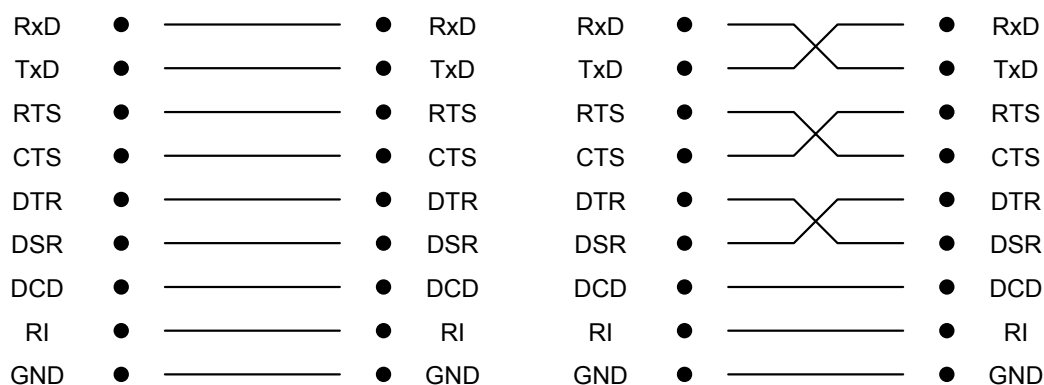
Obr. 9. 6 Číslování pinů konektoru Cannon 25 a Cannon 9

Signál	Cannon 25	Cannon 9	Signál	Cannon 25	Cannon 9
TxD	2	3	<i>DSR</i>	6	6
RxD	3	2	<i>DTR</i>	20	4
CTS	5	8	<i>DCD</i>	8	1
RTS	4	7	<i>RI</i>	22	9
GND	7	5			

14

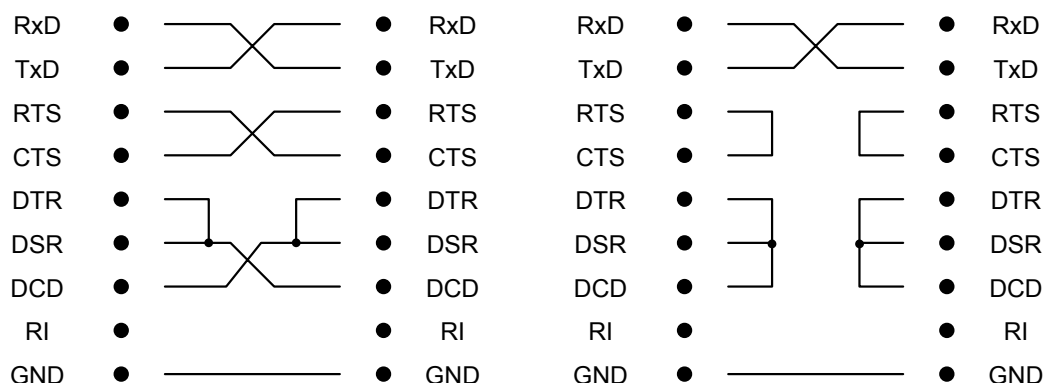
Tab. 9.2 Umístění signálů RS232 na konektorech Cannon

Konstrukce propojovacího kabelu sériových zařízení závisí jak na typu zařízení (DTE-DTE, DTE-DCE), tak na požadavcích programového vybavení. Na obrázku 9.7 vidíme několik možných typů propojení.

**DTE****DCE****DTE****DTE**

Způsob propojení zařízení typu DTE se
zařízením typu DCE (např. počítač-modem)

Propojení dvou zařízení typu DTE

**DTE(PC)****DTE(PC)****DTE****DTE**

Propojení typu NULL modem

Tří-vodičové propojení

Obr. 9.7 Způsoby propojení dvou zařízení

Pro přenášení dat mezi dvěma počítači se nejčastěji používá propojení nazvané NULL modem, u kterého jsou vzájemně propojeny signály DSR a DTR na jednom počítači a zároveň připojeny na signál DCD. Uvedenou konfigurací je vyjádřeno, že počítače jsou připojeny a inicializovány a je detekováno spojení (virtuální nosná frekvence modemu). Pro započetí komunikace je nutné u obou počítačů nastavit signály RTS a DTR do logické úrovně 1. V opačném případě nebude možné komunikaci uskutečnit.

9.5 Synchronizace v asynchronní komunikaci

Jak bylo uvedeno v úvodu, komunikace po sériové lince je implicitně asynchronní. Znamená to, že pro realizaci oboustranné komunikace není nutné provést propojení více než třemi vodiči (příjem, vysílání a signálová zem – viz. obr. 9.7). Příjemní strana by však musela velice rychle přijímaná data zpracovávat. Pokud by se tak nestalo, byla by přepsána nově přichozími. Prakticky existují dvě řešení uvedeného problému – řízení komunikace pomocí doplňkových signálů nebo vytvoření vyrovnávací paměti v obvodu sériového rozhraní. Obě metody dnes nacházejí své uplatnění. Je-li použito doplňkových signálů, signálů řízení modemu (viz. tabulka 9.1), dochází k tzv. hardwarovému řízení komunikace (hardware handshaking). Vedle něj existuje i tzv. softwarové řízení komunikace (software handshaking), které vkládá do komunikace speciální znaky nahrazující funkci jednotlivých pomocných signálů. Tyto speciální znaky bývají označovány zkratkami XOn a XOff.

Zároveň lze na současných obvodech 16750 zapnout vyrovnávací paměť (FIFO frontu) až do hloubky 64 bajtů. Do této paměti jsou ukládány přichozí data, pokud je aplikace není schopna neprodleně vyzvednout. Při naplnění paměti jsou nejstarší hodnoty nenávratně ztraceny.

9.6 Přístup k sériovému portu

Přístup k sériovému portu je ve všech operačních systémech rodiny Microsoft Windows zcela identický a nevyžaduje žádné doplňkové ovladače zařízení. V operačních systémech třídy NT (ale i Windows 2000 nebo XP) sice nemáme k registrům sériového portu přímý přístup, ale to není v daném případě překážkou. Všechny operační systémy totiž nabízejí programátorům identickou sadu vysokoúrovňových funkcí, které je oprostují od nutnosti hluboké znalosti obvodu řadiče sériového portu a nízkourovňového programování s jeho jednotlivými registry. Na rozdíl od práce se sériovým portem není nutné připojit k projektu žádný další zdrojový soubor, ale pouze vložit odpovídající systémový hlavičkový soubor. Ten se v prostředí 32-bitových Windows a jazyka MS Visual C++ jmenuje Windows.h.

I přes relativní jednoduchost práce se sériovým portem v prostředí MS Windows, programování s množinou komunikačních funkcí vyžaduje docela hlubokou znalost jednotlivých parametrů, konstant a interních principů fungování operačního systému. Pro zjednodušení práce na projektu byla vytvořena sada několika „užitečných funkcí“, které lze s výhodou využít v konkrétních úlohách. Následující řádky obsahují příklad komunikace dvou počítačů po sériové lince. Příklad je realizován dvěma programy nazvanými *ttyKlient* a *ttyServer*. Oba příklady používají stejnou sadu již zmíněných funkcí a přidávají vlastní aplikační logiku ve funkcích main.



K této kapitole je přiřazen ukázkový příklad

Příložené CD obsahuje plné znění ukázkového programu *ttyServer*.



ukázkový příklad klienta

Následující výpis obsahuje pouze aplikační logiku umístěnou do funkce main, neboť zbývající množina funkcí je identické s programem ttyServer.

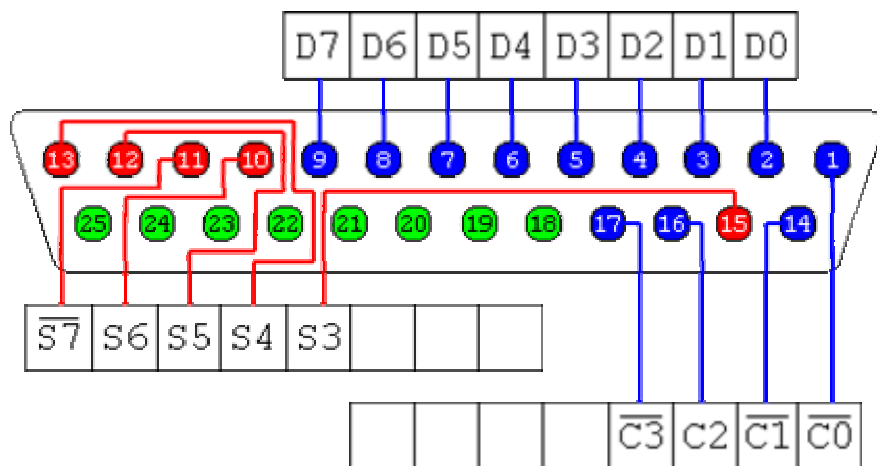
```
int main()
{
    int character;
    short cByteRead, i;
    BYTE buffer[255];

    // Otevru port
    if(PortInitialize("COM1", 9600, NOPARITY, ONESTOPBIT) == FALSE)
        return 1;
    // Ctu znaky dokud nezadam EOF
    do
    {
        character = getch();           // jeden znak z klavesnice
        PortWrite(character);          // a poslu jej na port
        cByteRead = PortRead(buffer);  // prectu data z portu
        for(i=0; i<cByteRead; i++)     // a vsechny je vypisu na obrazovku
            putchar(buffer[i]);
    } while(character != EOF);         // dokud neni EOF

    PortClose(hPort);                 // Uzavrenim koncim.
    return 0;
}
```

9.7 Paralelní port

Standard IBM LPT byl vytvořen ke komunikaci s tiskárnou „na míru“ této problematice po dohodě s výrobcí tiskáren. Jedná se o rozhraní určené pouze ke komunikaci s tiskárnou a nikdo nepočítal s jiným použitím. Tomu odpovídá i hardwarové provedení a zapojení. Standardní paralelní port je připojen na 25pinový konektor CANNON v provedení zásuvka.



Obr. 9.8 Paralelní port

Obsahuje:

- osm datových vodičů D0 až D7 umístěných na pinech 2 až 9,
- čtyři stavové výstupní,
- pět vstupních vodičů.

Ačkoliv plánované propojení PC s tiskárnou bylo původně jediným cílem využití paralelního portu, v průběhu dalších let byla vyvinuta řada zařízení, která paralelní port využívala. Původně měl být paralelní port jednosměrný. Později byly vyvinuty porty pro obousměrnou komunikaci jak vysílání dat z PC, tak i jejich příjem. Paralelní port byl tedy využit i pro vzájemné propojení počítačů. Standard, který definuje fyzické vlastnosti paralelního rozhraní, se jmenuje IEEE-1284 a jeho konečná verze byla schválena v roce 1994. Z pohledu komunikace definuje standard pět režimů. Kombinace těchto režimů určuje typ paralelního portu. Standardní paralelní port (SPP) byl původně určen pouze pro vysílání dat. Později byl však upraven, aby byl data schopen i přijímat. Součástí počítačů IBM PS/2 již byl oboustranný paralelní port. Směr přenosu dat určoval nově zavedený stavový bit. V roce 1991 byl vyvinut standard EPP (*Enhanced Parallel port*) firmami Intel, Xircom a Zenith Data Systems. O rok později byl ustanoven standard ECP (*Enhanced Capabilities port*) firmami Microsoft a HP. Některá paralelní zařízení vyžadují pro svou práci nastavení příslušného portu. To provedeme v programu Setup systémového BIOSu.

Režim	Směr	Přenosová rychlost
4 bitový Nibble	Pouze vstup	50 kB/s
8 bitový znakový	Pouze vstup	150 kB/s
Kompatibilní	Pouze výstup	150 kB/s
EPP	Vstup i výstup	500 kB/s – 2 MB/s
ECP	Vstup i výstup	500 kB/s – 2 MB/s

Tab. 9.3 Režimy paralelních portů podle standardu IEEE – 1284

Typ portu	Režim vstupu	Režim výstupu
SPP	Nibble	Kompatibilní
Obousměrný	Znakový	Kompatibilní
EPP	EPP	EPP
ECP	ECP	ECP

Tab. 9.4 Typy paralelních portů

9.8 Standardní použití paralelního portu – spojení s tiskárnou

Ve standardním provozu s tiskárnou čeká počítač na tiskárnu, až oznámí, že je připravena přijmout další znak. Potom vyšle na datové vodiče bity specifikující tento znak, oznámí pulsem na pinu STROBE, že na datových vodičích je připraven další znak, a opět čeká na tiskárnu. V nejlepším případě je možné, aby během doby, kdy tiskárna zpracovává přijatý znak, běžel hlavní program. Toho je dosaženo pomocí využití hardwarového přerušení na pinu /ACK (pokud je toto přerušení povoleno v odpovídajícím registru). Standardní paralelní port používá hardwarové přerušení IRQ 5 nebo IRQ 7. V jednom počítači umožňuje BIOS až čtyři rozhraní pro tiskárnu, tak řečená LPT (Line PrinTer). Pro první dvě rozhraní jsou vyhrazeny základní adresy 378H (LPT1) a 278H (LPT2). Po startu počítače prohledává BIOS uvedené adresy v daném pořadí a nalezne-li na nich paralelní port, přiřadí mu číslo LPT od 1 do 4. Teoreticky by měl port LPT1 používat IRQ7 a LPT2 IRQ5, ale není to nutné. Na tomto místě je třeba poznamenat, že všechna LPT porty mohou klidně používat stejné IRQ zároveň se zvukovou kartou nebo modemem. Pokud nebudou tato zařízení používána najednou, mělo by teoreticky všechno fungovat. Jde však o velice nevhodnou konfiguraci a vždy je výhodnější mít pro každé rozhraní samostatné IRQ. Na všech externích rozšiřujících deskách portů jsou parametry (adresa a přerušení) nastavitelné přepínači. Pokud se jedná o porty integrované na základní desce, je možné nastavovat parametry v BIOS SETUP počítače.

Číslo pinu	Směr	Označení	Význam
1	Výstupní	/Strobe (STR)	Potvrzení výstupních dat
2	Výstupní	Data0 (D0)	Výstupní data bit 0
3	Výstupní	Data1 (D1)	Výstupní data bit 1
4	Výstupní	Data2 (D2)	Výstupní data bit 2
5	Výstupní	Data3 (D3)	Výstupní data bit 3
6	Výstupní	Data4 (D4)	Výstupní data bit 4
7	Výstupní	Data5 (D5)	Výstupní data bit 5
8	Výstupní	Data6 (D6)	Výstupní data bit 6
9	Výstupní	Data7 (D7)	Výstupní data bit 7
10	Vstupní	/ACK	Potvrzení přijetí dat
11	Vstupní	Busy	Tiskárna je zaneprázdněná
12	Vstupní	Paper Empty (PE)	Tiskárně došel papír
13	Vstupní	Select (SEL)	Tiskárna je on-line
14	Výstupní	/Auto Feed (AF)	Odstránkování
15	Vstupní	/Error (ERR)	Tiskárna detekovala poruchu
16	Výstupní	Initialize Printer (IP)	Inicializace (SW reset) tiskárny
17	Výstupní	/Select Input (SI)	Počítač vybere tiskárnu jako výstupní zařízení

Tab. 9.5 Význam jednotlivých signálů portu

9.9 Programování s paralelním portem

Každý paralelní port obsahuje tři osmi-bitové registry. Všechny tři lze číst a do dvou z nich zapisovat. Jednotlivé registry jsou mapovány na fyzické signály paralelního portu.

Název registru	Adresa registru (LPT1, LPT2, LPT3)
Datový registr	378H, 278H, 3BCH
Stavový registr	379H, 279H, 3BDH
Řídicí registr	37AH, 27AH, 3BEH

Význam jednotlivých bitů v registrech je dán v následujících tabulkách následovaných významem operací čtení z zápisu. Šedivě označené bity jsou na vstupu nebo výstupu negovány. Tzn., že logické hodnota 0 bude reprezentována jako 1 a naopak.

Datový registr

Pin	9 (D7)	8 (D6)	7 (D5)	6 (D4)	5 (D3)	4 (D2)	3 (D1)	2 (D0)
bit	7	6	5	4	3	2	1	0

Operace zápisu má za následek přenesení hodnot jednotlivých bitů registru na odpovídající výstupní vodiče paralelního portu (data).

Operace čtení přečte naposled zapsaná data do registru. Operace se běžně nepoužívá a slouží k ověření korektní práce výstupního datového registru.

Stavový registr

Pin	11 (Busy)	10 (ACK)	12 (PE)	13 (SEL)	15 (ERR)	-	-	-
bit	7	6	5	4	3	2	1	0

Operace zápisu na tomto registru není definována a nemá význam ji provádět.

Operace čtení má za následek přečtení aktuálních hodnot vstupních vodičů paralelního portu (stav). Význam má pouze nejvyšších 5 bitů. Hodnota dolních třech není jednoznačně definována.

Řídicí registr

Pin	-	-	-	IRQ	17 (SI)	16 (IP)	14 (AF)	1 (STR)
bit	7	6	5	4	3	2	1	0

Operace zápisu má za následek přenesení hodnot nižších 4 bitů registru na odpovídající výstupní vodiče paralelního portu (řízení). Pátý bit povoluje generování přerušení od paralelního portu.

Operace čtení přečte naposled zapsaná data do registru. Operace se běžně nepoužívá a slouží k ověření korektní práce výstupního řídicího registru.

Paralelní port lze použít k přímému spojení počítačů bez nutnosti osazení síťové karty a instalování speciálního síťového softwaru. Pomocí paralelního portu lze přenášet data v průměru 3x až 4x rychleji než pomocí sériového rozhraní. Komunikační kabel má však výrazně omezenou délku. Vzhledem k tomu, že veškerá komunikace probíhá pomocí napětových úrovní 5V a nikoli proudovou smyčkou, jsou kabely delší než 1m velmi náchylné k rušení. Přesto lze po dodržení určitých zásad používat až 10 m dlouhé kabely pro komunikaci po paralelním portu.

Při propojení dvou počítačů lze použít pro přenos pouze 5 datových kabelů, neboť máme k dispozici pouze 5 vstupních (stavových) signálů. Standardní propojení se provádí pomocí tzv. Laplink kabelu, jehož schéma vidíme na předchozím obrázku.

9.10 Přístup k paralelnímu portu

V operačním systému Microsoft Windows XP (ale i Windows 2000 nebo dříve NT) nelze k registrům paralelního portu přistupovat přímo. Operační systém brání hardware před přímým „útokem“ programového vybavení. Abychom mohli s porty přece jen pracovat, byl na katedře ATŘ vytvořen speciální systémový program (ovladač zařízení – device driver), který nám paralelní porty zpřístupní.

Ovladač definuje tři důležité funkce:

```
void WritePortB(unsigned long adresa, unsigned char data);
```

Funkce provede zápis jednoho bajtu uloženého v proměnné data na port daný proměnnou adresa.

```
unsigned char ReadPortB(unsigned long adresa);
```

Funkce přečte z portu určeného adresou jeden byte a vrátí jej jako návratovou hodnotu.

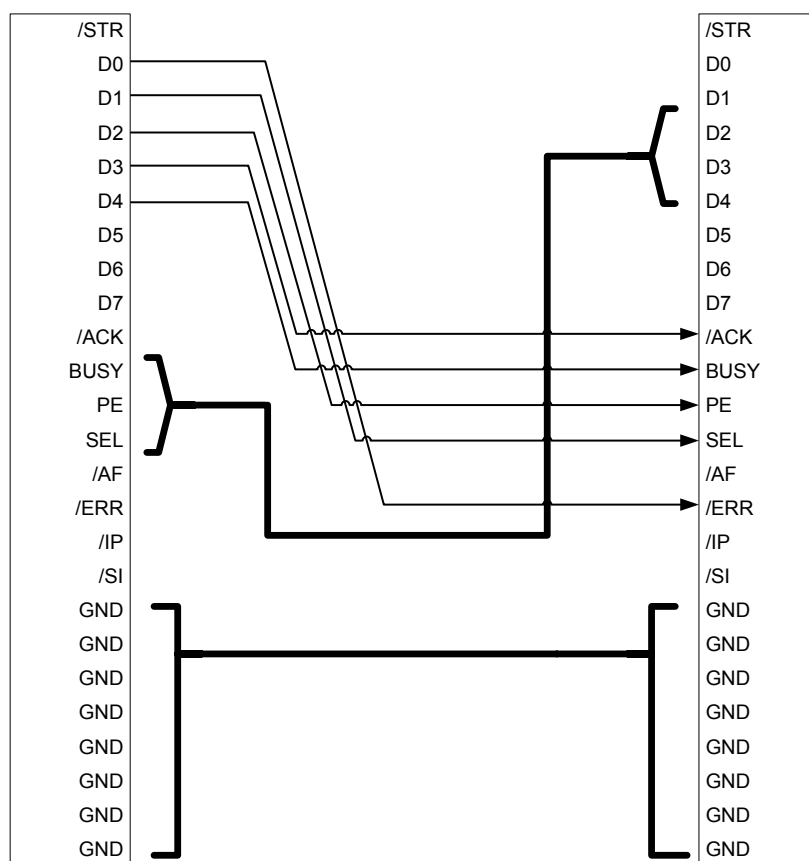
```
void Wait(unsigned long milisekund);
```

Funkce přeruší (pozastaví) běh programu na zadaný počet milisekund.

Abychom mohli uvedené funkce použít, musíme do projektu aplikace vložit následující soubory:

- DirectIO.h
- DirectIO.cpp
- IOCtrl.h

Soubory musí být umístěny i do aplikačního adresáře.



Obr. 9.9 Propojení paralelního portu



Příklad použití funkcí

```
#include <stdio.h>
#include "DirectIO.h"

void main()
{
    int TimeOut;
    int Port;
    char Znak;
```

```

do
{
    printf("Zadejte adresu paralelniho portu (0x278/0x378):");
    scanf("%x",&Port);
} while(Port!=0x278 && Port!=0x378);

printf("Zadejte velikost casove periody vysilani T=");
scanf("%d",&TimeOut);

printf("Vysilam cisla:\n");
Znak = 0;
do
{
    WritePortB(Port,Znak);
    Wait(TimeOut);

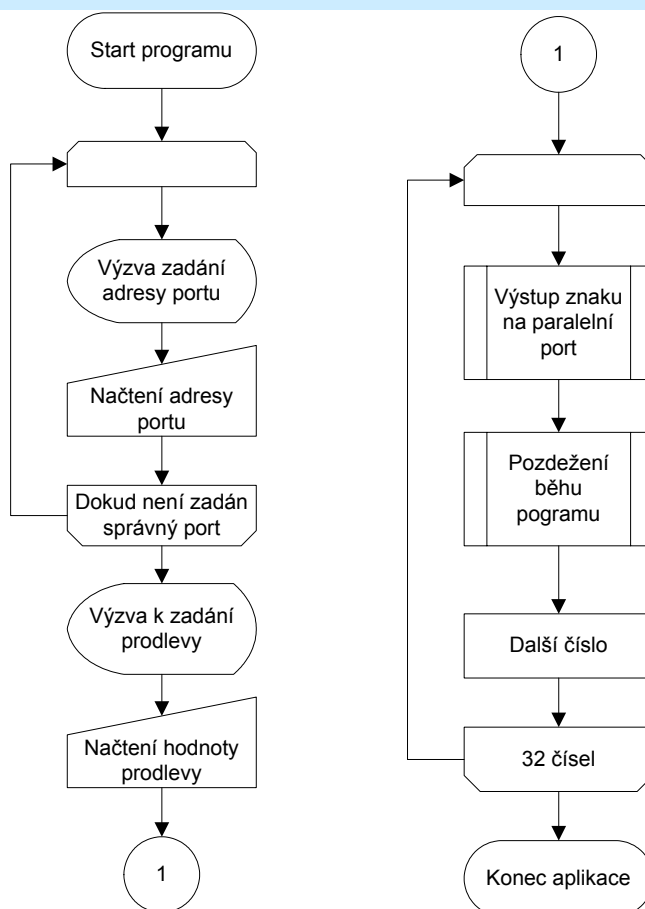
    printf("%c",Znak);
    ++Znak;
} while(Znak != 32);
}

```

Program nejdříve od uživatele načte adresu paralelního portu a zkontroluje validitu zadané hodnoty. Dále je načtena velikost prodlevy mezi vysláním jednotlivých dat. Závěrem na paralelní port postupně zapíše hodnoty od 0 do 31. Logiku aplikace lze vyjádřit vývojovým diagramem.



Příklad popisu programu vývojovým diagramem





Korespondenční úkol – připojitelná zařízení

Jaká zařízení kromě tiskáren mohou být připojeny k paralelnímu portu? Vyhledejte zařízení nejen z oblasti domácích PC ale i z průmyslové praxe.



Další zdroje

1. Vlach, J, Vlachová V. *Počítačová rozhraní*, Praha, Ben 2002, počet stran: 176. ISBN 80-7300-010-5.
2. Mueller, S. *Osobní počítač – Hardware, upgrade, opravy*, Brno, Computer Press 2006, počet stran: 888. ISBN 80-7226-796-5.
3. Gook, M. *Hardwarová rozhraní*, Brno, Computer Press 2006, počet stran: 464. ISBN 80-251-1019-2.
4. USB Official site [online] dostupný z <http://www.usb.org>
5. Beyond Logic – Serial interface server [online] dostupný z <http://www.beyondlogic.org>
6. HW server představuje RS-232 [online] dostupný z <http://rs232.hw.cz/>



Shrnutí kapitoly

USB standard 1.1 podporuje přenosovou rychlost 12 Mb/s, což převedeno na rychlost v bytech odpovídá 1,5 MB/s. V roce 2000 byl uveden standard USB 2.0, který je 40 krát rychlejší než jeho předchůdce. Sběrnice umožňuje připojit až 127 zařízení, přičemž je založena na hvězdicové topologii. Základem je USB hub neboli rozbočovač, který je buďto interní součástí počítače anebo může figurovat jako externí zařízení. Alternativou USB je použití standardu IEEE 1394 (FireWire), který dosahuje rychlosti 400 Mb/s a navrhovanou další rychlostí je 800 Mbit/s. Na rozdíl od USB není ale tak rozšířen. Důvodem je existence USB 2.0 standardu, který je rychlejší než současná 400 megabitová varianta.

Typickým představitelem sériové komunikace je rozhraní RS232. Přenos dat se provádí postupně po jednotlivých bitech – tedy sériově, na rozdíl od souběžného přenosu u paralelního portu. Pro nejjednodušší propojení dvou komunikujících sériových zařízení proto stačí dva vodiče – jeden pro příjem, druhý pro vysílání. Sériové rozhraní je realizováno speciálním obvodem (dříve 8252, dnes 16750). Jedním z nejdůležitějších úkolů obvodu je převod paralelních dat, nejčastěji 8-mi bitového bajtu, na sériový tok bitů a naopak. Sériovou komunikaci často realizujeme asynchronní formou. Znamená to, že data nejsou synchronizovány žádným hodinovým signálem udávajícím příchod nových dat a jednotlivých bitů. Obě zařízení musí být pro úspěšnou komunikaci nastaveny na ekvivalentní přenosovou rychlost, udávanou v bitech za sekundu [bps]. Napětíové úrovně signálů rozhraní RS232 se však nepohybují ve standardních TTL hodnotách (0 – 5 V). Pro větší odolnost proti rušení a k dosažení větších vzdáleností mezi komunikujícími zařízeními jsou tyto úrovně převedeny na napětí ± 12 V. Přístup k sériovému portu je ve všech operačních systémech rodiny Microsoft Windows zcela identický a nevyžaduje žádné doplňkové ovladače zařízení.

Standard IBM LPT byl vytvořen ke komunikaci s tiskárnou „na míru“ této problematice

po dohodě s výrobcí tiskáren. Jedná se o rozhraní určené pouze ke komunikaci s tiskárnou, a nikdo původně nepočítal s jiným použitím. Původně měl být paralelní port jednosměrný. Později byly vyvinuty porty pro obousměrnou komunikaci, jak vysílání dat z PC, tak i jejich příjem. Paralelní port byl tedy využit i pro vzájemné propojení počítačů. Standard, který definuje fyzické vlastnosti paralelního rozhraní, se jmenuje IEEE-1284 a jeho konečná verze byla schválena v roce 1994. Z pohledu komunikace definuje standard pět režimů. Kombinace těchto režimů určuje typ paralelního portu.



Úkol k řešení 9.1 – komunikace dvou PC po sériovém portu

Na počítači A (server) spustíte program, který očekává přijetí 8+1+3 znaků, udávajících jméno souboru. Dále následuje sekvence bajtů s obsahem souboru. Soubor je na počítači A nejdříve vytvořen v aktuálním adresáři a následovně naplněn přijatou sekvencí bajtů. Na počítači B (klient) se spustí program, který na základě dotazu získá jméno souboru, jehož obsah pošle počítači A. Na závěr oba počítače vypíší počet vyslaných/přijatých bajtů souboru a jeho název.

Výsledný program odevzdejte společně s protokolem, který obsahuje:

1. Zadání,
2. vývojový diagram,
3. okomentovaný zdrojový text,
4. závěr.

Pro řešení semestrálního projektu bude mít každá dvojice studentů upravenou verzi svého zadání.



Úkol k řešení 9.2 – komunikace dvou PC po paralelním portu

Počítač A čeká na přijetí dvou typů dotazů realizovaných zvolenými kódy. Dotaz 1 – kolik je hodin?, dotaz 2 – jaké je datum? Na základě přijetí dotazu počítač A odpoví textovým řetězcem s patřičnými informacemi. Počítač B nejprve dotaz zadá a následovně přijme odpověď. Odpověď také zobrazí na obrazovce. Uživatel může zadat na počítači B libovolný počet dotazů. Po ukončení aplikace B se automaticky ukončí i aplikace A.

Pro řešení semestrálního projektu bude mít každá dvojice studentů upravenou verzi svého zadání.



Kontrolní otázka 9.1

Jaké typy bloků používáme při tvorbě vývojových diagramů? Použijte tyto bloky při tvorbě vývojového diagramu v semestrálním projektu.



Kontrolní otázka 9.2

Jaké signály jsou využity při komunikaci po sériovém portu?



Kontrolní otázka 9.3

Popište obecně paralelní port, jeho typy, konektory a komunikaci přes paralelní port.



Kontrolní otázka 9.4

Co je to UART a jaká je jeho funkce?



Kontrolní otázka 9.5

Uved'te přenosové rychlosti jednotlivých rozhraní a srovnejte je s ostatními počítačovými sběrnici.



Kontrolní otázka 9.6

Charakterizujte obecně sběrnici USB.

10. Paměti



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- rozdělit paměti dle různých kritérií
- charakterizovat vlastnosti a typy polovodičových pamětí
- charakterizovat vlastnosti pamětí SRAM a DRAM
- popsat vývoj paměťových čipů v počítači
- definovat fyzické uspořádání jednotlivých paměťových modulů



Výklad

Paměť je hardwarové zařízení, které umožňuje uložit obsah informace, uchovat ji po požadovanou dobu a znovu ji získat pro další použití. Těmto operacím říkáme zápis do paměti a čtení z paměti. Informace je vyjádřena jako číselná hodnota, jejíž jednotkou je jeden bit. Na paměti se můžeme podívat z různých úhlů pohledu. Z pohledu fyzikálního složení se paměti dělí na magnetické, optické, magnetooptické a polovodičové. Dále můžeme paměti charakterizovat dvojicemi přívlastků: statické vs. dynamické nebo napěťově závislé vs. nezávislé. Existuje mnoho dalších hledisek dělení pamětí, např. podle přístupu, rychlosti, podle hardwarového provedení paměťových modulů a v neposlední řadě také podle schopnosti zápisu.

10.1 Rozdělení pamětí

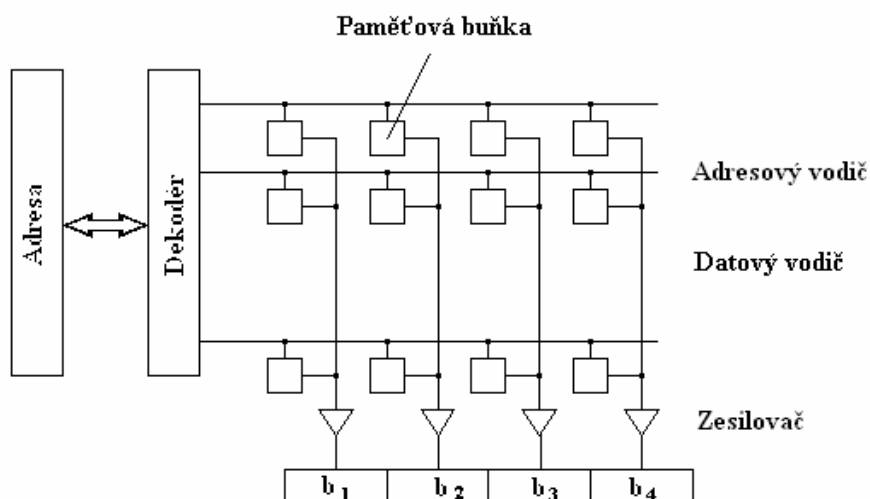
Paměť je základním prvkem jakéhokoliv počítače. Paměti můžeme rozdělit do několika skupin, jak už jsme uvedli v úvodu. Počítačové paměti si rozdělíme takto:

- vnitřní
 - o ROM (PROM, EPROM, EEPROM, Flash),
 - o RAM (DRAM – *dynamic RAM*, SRAM – *static RAM*),
- vnější (magnetické, magneticko optické, optické, polovodičové).

Nejprve si představíme polovodičové paměti, abychom pochopili princip uložení informace v paměti a její čtení. Poté si představíme charakteristiku čipů SRAM a DRAM a seznámíme se s paměťovými moduly pro současné počítače. Popíšeme si vývoj jednotlivých modulů hierarchicky od počítačů s procesory x86 až po současnost.

10.2 Polovodičové paměti

Tyto paměti jsou založeny na logických obvodech, které jsou vytvořeny na polovodičových čípech. Rozlišujeme paměti typu ROM (*Read Only Memory*) a RAM (*Random Access Memory*). Paměť se skládá z adresových vstupů, datových vstupů a výstupů, řídicích vstupů a napájecích vstupů. Základními parametry charakterizující paměť jsou kapacita (maximální velikost uložených dat), přístupová doba (doba od požadavku po zpřístupnění dat) a přenosová rychlost (doba čtení a zápisu dat).

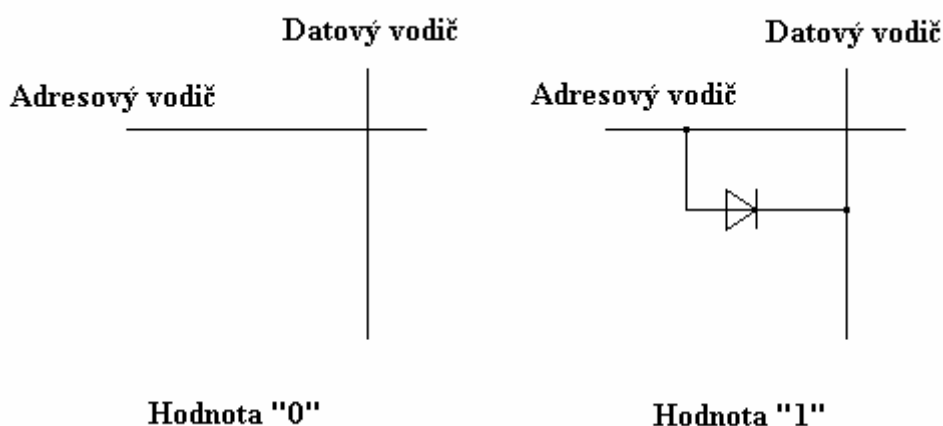


Obr. 10.1 Struktura paměti

Paměť ROM

Jedná se o paměť nezávislou na zdroji energie. Tyto paměti již dle názvu jsou určeny pouze ke čtení. Jejich obsah je pevně dán již při výrobě paměti. Tyto paměti jsou využívány v dnešních počítačích zejména k uložení tzv. *BIOSu* resp. tohoto programu (neplést s uložením nastavení). To je uloženo v paměti *CMOS*. Paměťová buňka je realizovaná pomocí polovodičové diody viz. Obr. 8.2 nebo pomocí tranzistoru. Paměť statická, energeticky nezávislá znamená fakt, že data se v ní uchovávají i bez trvalého poskytnutí přísunu energie. Informace jsou do těchto pamětí pevně zapsány při jejich výrobě. Rozlišujeme několik typů těchto pamětí:

- PROM,
- EPROM,
- EEPROM,
- Flash.

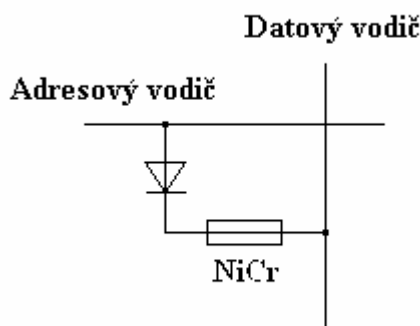


Obr. 10.2 Struktura ROM – polovodičová dioda

Paměť PROM

PROM (*Programmable ROM*). Jedná se o paměť typu ROM, kterou lze pomocí programátoru jednou naprogramovat. Je to paměť statická, energeticky nezávislá. Tato paměť má stejnou strukturu jako

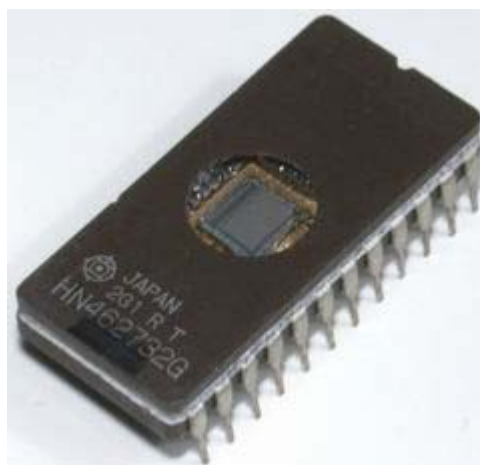
paměť ROM s polovodičovými diodami. Neobsahuje po vyrobení žádnou pevnou informaci a je až na uživateli, aby provedl příslušný zápis informace. Tento zápis je možné provést pouze jednou a poté již paměť slouží stejně jako paměť ROM. Oproti paměti typu ROM naprogramovanou při výrobě obsahuje navíc tavnou pojistku (obr. 8.3). Tato pojistka se při zápisu vyšší hodnotou elektrického proudu přetaví a proto již není možno znovu zapisovat.



Obr. 10.3 Princip PROM

Paměť EPROM

EPROM (*Erasable PROM*). Opět je základ paměti realizován jako paměť ROM resp. PROM. Na rozdíl od těchto pamětí do této paměti lze dle názvu data mazat a opětovně zapisovat. Je to paměť statická, energeticky nezávislá. Paměť se maže UV zářením a zapisuje se do ní elektrickým signálem. Tyto paměti jsou realizovány pomocí speciálních unipolárních tranzistorů, které jsou schopny na svém přechodu udržet elektrický náboj po dobu až několika let. Tento náboj lze vymazat právě působením UV záření přes průhledný střed pouzdra již zmíněným UV zářením (obr. 8.4).



Obr. 10.4 Paměť EPROM

Paměť EEPROM

EEPROM (*Electrically Erasable PROM*). Struktura je znovu postavena na základě ROM technologie. Na rozdíl od EPROM zde mazání probíhá elektrickým signálem (obr. 8.5), což je mnohem rychlejší. Umožňuje až několik tisíc přepisů resp. mazání. Je to paměť statická, energeticky nezávislá. Založena je na tranzistorech vyrobených technologií MNOS (*Metal Nitrid Oxide Semiconductor*). Jedná se o tranzistory, na jejichž řídicí elektrodě je nanesena vrstva nitridu křemíku a pod ní je umístěna tenká vrstva oxidu křemičitého.



Obr. 10.5 EEPROM čip

Paměť Flash

Je obdobou paměti EEPROM. Mazání je realizováno elektrickou cestou. Na rozdíl od EEPROM není potřeba speciálního programovacího zařízení. Tato paměť lze přepsat přímo v počítači. Jedná se o paměť statickou, energeticky nezávislou. Paměť je vnitřně organizována po blocích a na rozdíl od paměti typu EEPROM, lze programovat každý blok samostatně.

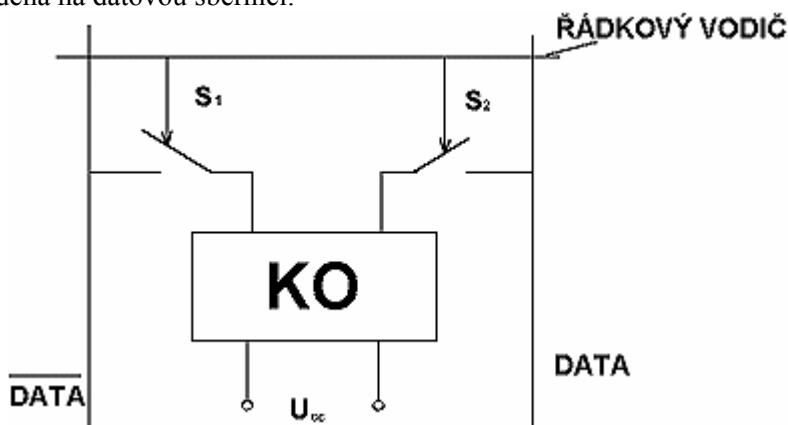
10.3 Paměťové čipy RAM

Paměť RAM (*Random Access Memory*) je na rozdíl od ROM energeticky závislá. To znamená, že po odpojení zdroje energie se tato paměť vymaže. Paměti jsou určeny pro zápis i čtení. Rozlišujeme dva typy těchto pamětí a to:

- Statické (SRAM),
- Dynamické (DRAM).

Paměť SRAM

SRAM (*Static Random Access Memory*). Paměťové buňky jsou tvořeny bistabilním klopným obvodem. Jednotlivé paměťové buňky bývají uspořádány do matice, jejich výběr je prováděn přes řádkové a sloupcové vodiče. Přes výběr řádků je zapsaná informace připojena na datové vodiče. Adresace paměťové matice je zajišťována přes dekodér řádků a dekodér sloupců. Data jsou v budících obvodech a přivedena na datovou sběrnici.



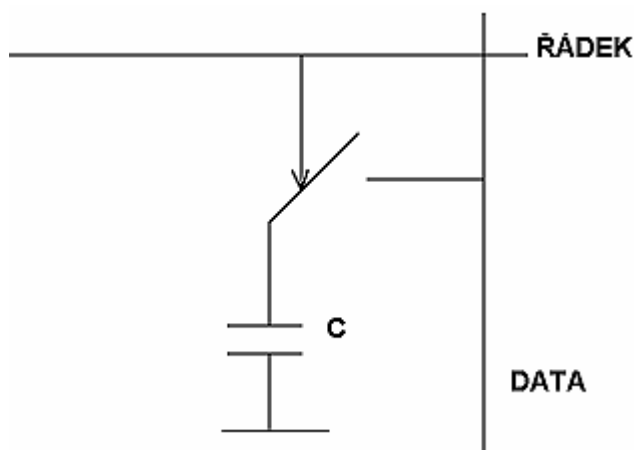
Obr. 10.6 Struktura SRAM

Tyto paměti jsou dobré vzhledem ke svým nízkým přístupovým dobám, které jsou pod 0,5 ns. Jsou rychlejší avšak mají nižší hustotu než čipy DRAM. To znamená, že čipy mají větší velikost a nižší kapacitu. Proto našly uplatnění hlavně jako paměť typu cache. U těchto pamětí není nutná obnova informace jako u DRAM, protože je paměťová buňka tvořena šesti tranzistory a nejsou použity kondenzátory.

Paměť DRAM

DRAM (*Dynamic Random Access Memory*). Paměťové buňky jsou zde realizovány pomocí kondenzátorů.

Z tohoto důvodu je zde nutná pravidelná obnova uložených informací neboť kondenzátory se vybíjejí díky svodovým odporům i v době, kdy je paměť připojena ke zdroji elektrického napájení. Obnova je prováděna několiksetkrát za sekundu. Napětí v kondenzátorech odpovídá logické 0 nebo 1. Buňka paměti DRAM je velmi jednoduchá a dovoluje vysokou integraci a nízké výrobní náklady. I když mají vyšší přístupové doby než paměti SRAM, jsou mnohem více používány, neboť jejich cena je mnohem nižší.



Obr. 10.7 Struktura paměťové buňky DRAM

Paměti DRAM jsou konstruovány jako matice, v nichž se jedna paměťová buňka zpřístupňuje pomocí dvou dekodérů. Řadič operační paměti adresu rozdělí na dvě části, z nichž každá je přivedena na vstup samostatnému dekodéru (jeden dekodér vybere řádek a druhý sloupec). Obvody operačních pamětí pak bývají realizovány jako matice.

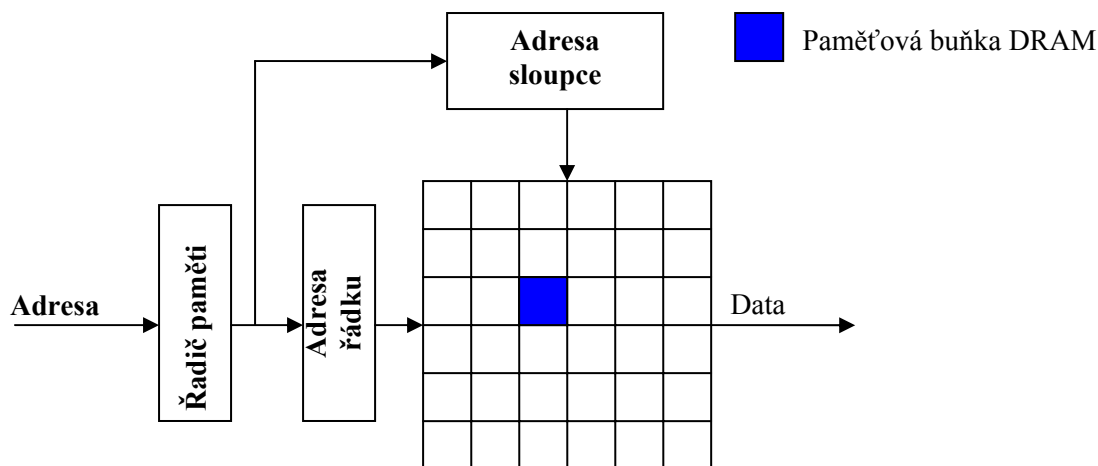


Pojem k zapamatování: Hustota integrace paměti DRAM

Oproti pamětem SRAM mají paměti DRAM obrovskou hustotu integrace. Paměťová buňka je tvořena dvojicí kondenzátor – tranzistor. Z návrhu paměťového čipu tak vyplývá, že paměťový modul s kapacitou 1024 MB obsahuje 1024 milionů dvojic kondenzátor – tranzistor.

V našem výkladu o hardwarových komponentách, konkrétně procesorech a čipových sadách se často vyskytuje letopočet 1998. Nejinak tomu je i u pamětí. V tomto roce přestaly stačit paměťové moduly s přístupovou dobou 60 nanosekund, které odpovídaly taktování 16,7 MHz. Na trh se velmi rychle rozšířily paměti SDRAM taktovaných 66, 100 a 133 MHz.

V roce 2001 se opět posunula rychlost až na 266 MHz, následovanou hodnotami 333 a 400 MHz. Paměťové moduly RIMM byly schopny dosáhnout až 1066 MHz, ale pro svou cenu nebyly příliš rozšířeny. Dnešní paměťové moduly DDR2 SDRAM však již tuto hranici také dosáhly.



Obr. 10.8 Matice paměti DRAM



Pojem k zapamatování: Přístupová doba vs. taktovací frekvence

Kritériem posuzování rychlosti paměti je její přístupová doba. Tato hodnota je vyjádřena v nanosekundách. V oblasti procesorů, sběrnic a pamětí se rychlost vyjadřuje taktovací frekvencí v MHz. Proto je nutné najít vztah mezi délkou cyklu v nanosekundách a frekvencí v MHz.

MHz	ns	MHz	ns	MHz	ns	MHz	ns	MHz	ns
60	17	200	5,0	466	2,1	800	1,3	1500	0,71
66	15	266	3,8	500	2,0	1000	1,0	2000	0,53
100	10	300	3,3	533	1,9	1100	0,91	2500	0,40
133	7,5	333	3,0	600	1,7	1200	0,88	3000	0,33
166	6,0	400	2,5	667	1,5	1300	0,83	3500	0,29

Tab. 10.1 Délka cyklu v nanosekundách pro příslušnou frekvenci



Pojem k zapamatování: Zpoždění (*latency*)

Po obdržení požadavku na paměťovou operaci je nejprve vyžádaný bit paměti vyhledán, až poté je proveden samotný přenos. Zpoždění – *latence* paměti je doba potřebná pro vyhledání konkrétního bitu.

DRAM paměti existuje několik základních typů vývojově navazujících:

- **FPM RAM**

U tohoto typu paměti je využit proces nazývaný stránkování (*paging*). Využívá klasických DRAM pamětí. Běžná paměť typu DRAM pracuje na principu vyhledávání adres řádků a sloupců, ve kterých je umístěna hledaná paměťová buňka. U režimu stránkování je paměť rozdělena na několik stránek o

velikosti 512 bajtů až několik kilobajtů. Proces vyhledávání se urychluje tím, že stránkování zachová adresu řádky a změní pouze adresu sloupce. Toto je samozřejmě využitelné pouze v případě, že se následná hledaná buňka nachází na stejné stránce. Pokud není na stejné stránce, vyhledávání se opět zpomaluje.

Rychlost paměťového modulu byla dále zvýšena režimem *burst mode* vycházejícího z předpokladu načtení dat v sousedních buňkách, a to až v dalších třech vedle sebe sousedících, bez jakýchkoliv čekacích stavů. Režim přístupu DRAM tak byl vyjádřen posloupností x-y-y-y, kde x je čas pro načtení dat z první buňky včetně zmiňované latence a y je čas potřebný pro načtení dat z další buňky již bez zpoždění (latence). U klasických DRAM modulů byl *burst mode* definován posloupností 5-5-5-5, neboť bylo nutné pokaždé nastavit adresu sloupce i řádku. U paměti FMP adresa řádku zůstává stejná po celou dobu přístupu k datům tohoto řádku, a proto je možné hovořit o snížení potřebných cyklů na 5-3-3-3.

U těchto pamětí byl také využíván proces prokládání, což znamenalo použití dvou stejných paměťových modulů. Princip prokládání pracuje na principu, že zatímco řadič paměti načítá data z určité buňky v prvním modulu, začíná zároveň vyhledávat buňku následující v druhém paměťovém modulu, ze které poté začne načítat data a vyhledávat, tudíž bude opět v prvním modulu. Využívají modulů SIMM.

- **EDO RAM**

EDO (*Extended Data Out*). Poprvé se tyto paměti objevily v roce 1995. Vychází z modulů FPM. Princip prokládání u těchto modulů spočívá opět v principu načítání. Zatímco řadič paměti načítá data z určité buňky, začíná zároveň vyhledávat buňku následující. Výhodou u těchto pamětí bylo to, že nemusely při prokládání využívat dva paměťové moduly. Jedná se doslova o zlepšený princip FPM využívající algoritmy k překrývání časování mezi dvěma po sobě následujícími přístupy do paměti. Využívají stále modulů SIMM. Burst mode časování těchto modulů bylo sníženo na 5-2-2-2, což pro čtyři přenosy dat vyžaduje namísto 14 cyklů u paměti FPM cyklů 11 a to je teoretické zrychlení časování paměti o více než 20 procent.

- **BEDO RAM**

BEDO (*Burst Extended Data Output RAM*). Tyto paměti čtou data v dávkách a mají vlastní adresový čítač. Nejsou vhodné pro rychlosti větší než 66 MHz. Využívaly modulů SIMM a jejich neuplatnění spočívalo v nasazení již prvních SDRAM modulů na trh.

- **SDRAM**

SDRAM (*Synchronous Dynamic RAM*). Tyto paměti využívají pro svou práci modulů DIMM. Paměti SDRAM jsou synchronní tzn. vstupní a výstupní signály pracují synchronizovaně s kmitočtem základní desky. Tyto paměti jsou schopny pracovat dle normy PC100 a PC133. Což v praxi znamená, že pro moduly o rychlosti 100 MHz byly používány čipy s přístupovou dobou 8 ns a pro 133 MHz to je přístupová doba 7,5 ns. V burst mode režimu jsou tyto paměti schopny dodávat data každý hodinový cyklus, zápis časování tak vypadá takto: 5-1-1-1. SDRAM paměti byly vyráběny s frekvencemi:

- PC66: pro systémovou sběrnici s taktem 66 MHz
- PC100: pro systémovou sběrnici s taktem 100 MHz
- PC133: pro systémovou sběrnici s taktem 133 MHz

Paměťové moduly PC133 jsou přes svůj vývojově starší věk stále na trhu ve variantách 256 a 512 MB. To jen dokazuje jaký náskok má v praxi vývoj posledních novinek oproti hlavnímu proudu v oblasti hardwaru. Je třeba respektovat poptávku běžných uživatelů, kteří neobměňují hardware tak často jako počítačová fanouška.

- **DDR SDRAM**

DDR SDRAM (*Dual Data Rate SDRAM*). Tyto paměti dokáží přenášet dvě informace v jednom cyklu, čímž se zvyšuje jejich rychlost. Používají moduly vycházející z modulů DIMM označenými jako DDR DIMM a jejich napájecí napětí je 2,5 V. Tyto paměti v burst mode režimu ještě zrychlily časování a nyní jsou již data dodávána za půlku hodinového cyklu. Časování je tedy 5-1/2-1/2-1/2.

Typ paměti	Označení	Frekvence systémové sběrnice	Přenosová rychlost
DDR 200	PC1600	100 MHz	1,600 GB/s
DDR 266	PC2100	133 MHz	2,133 GB/s
DDR 333	PC2700	166 MHz	2,667 GB/s
DDR 400	PC3200	200 MHz	3,200 GB/s

Tab. 10.2 Základní typy paměťových modulů DDR SDRAM

Kromě výše uvedených pamětí DDR SDRAM jsou vyráběny i typy umožňující práci při vyšší frekvenci:

PC3500 (DDR433), PC3600 (DDR444), PC3700 (DDR466), PC4000 (DDR500) nebo PC4300 (DDR533).

- **DDR2 SDRAM**

Jedná se o novější standard vycházející z pamětí DDR SDRAM, poskytují dvojnásobnou přenosovou rychlost oproti DDR SDRAM. Data jsou čtena (zapisována) s nástupnou i sestupnou hranou hodinového signálu obdobně jako u DDR SDRAM. Poskytují dvojnásobnou přenosovou rychlost oproti DDR SDRAM a mají asi o 50% menší spotřebu el. energie. Jejich napájecí napětí je 1,8 V oproti 2,5 V u DDR SDRAM modulů. Dosažení vyšší přenosové rychlosti je založeno na skutečnosti, že jádro paměťového obvodu (pracující na frekvenci 100 MHz) může při každém čtecím cyklu předvybrat další 4 bity z paměťové matice a uložit je do V/V bufferů. Paměťové moduly DDR2 SDRAM a DDR SDRAM nejsou vzájemně kompatibilní.

Typ paměti	Označení	Frekvence systémové sběrnice	Přenosová rychlost
DDR 2 - 400	PC2 - 3200	200 MHz	3,200 GB/s
DDR 2 - 533	PC2 - 4300	266 MHz	4,264 GB/s
DDR 2 - 667	PC2 - 5300	333 MHz	5,336 GB/s
DDR 2 - 800	PC2 - 6400	400 MHz	6,400 GB/s

Tab. 10.3 Základní typy paměťových modulů DDR2 SDRAM

Někteří výrobci se tohoto označení nedrží a název čipu spojují přímo s typem paměti, dostáváme se tak k názvům PC533, PC667, PC800. V současné době (rok 2007) pronikají na trh paměťové moduly s označením PC900 a dokonce PC1066, což odpovídá označení PC8500 a dokonce PC1160 (PC9280). Číslo tak nyní udává pracovní frekvenci a dostáváme se s taktováním paměti nad 1 GHz. Některé společnosti mají prototypy pamětí DDR3, avšak na jejich standardizaci si ještě musíme počkat.



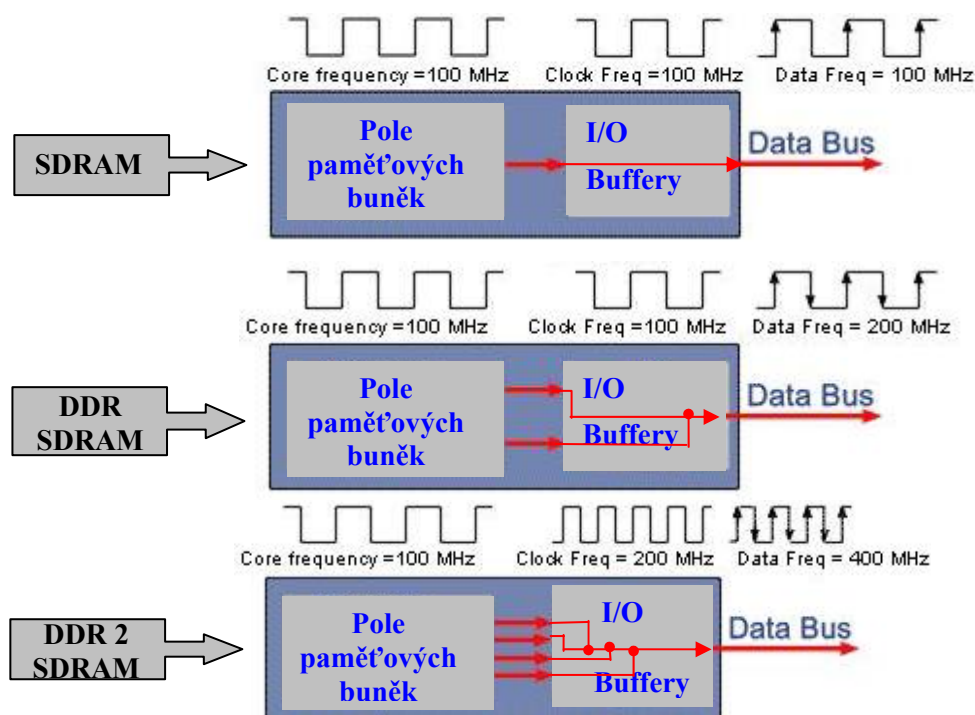
Korespondenční úkol – paměťové moduly SDRAM

V souvislosti s vývojem paměťových modulů vyhledejte a prostudujte parametry a vlastnosti paměťových modulů typu DDR3.

- **Dual Channel DDR**

Nejedná se o nový typ paměti, ale o zdokonalení architektury využívající paměti DDR SDRAM. Pracují na stejném principu jako DDR, ale pro práci využívají dva kanály, a proto jsou data přenášena po 128 bitech (64 bitů pro každý kanál). Tato technologie vyžaduje párování paměťových modulů. Tyto moduly musí mít stejné parametry (nejlépe je kupovat najednou a od stejného výrobce), pokud

ovšem nevlastníte čipovou sadu s technologií *Intel Flex Memory*, která umožňuje použití různých kapacit pamětí při zachování dvoukanalového režimu. Pro využití architektury *Dual Channel DDR* je tedy zapotřebí použití čipové sady podporující *Dual Channel DDR*.



Obr. 10.9 Srovnání taktování paměťových modulů technologie SDRAM

- **RDRAM**

RDRAM (*Rambus DRAM*). Tyto paměti jsou velice rychlé, ale nekompatibilní s SDRAM. Jako DDR SDRAM pracují s dvojnásobným přenosem dat. Na rozdíl od všech předešlých pamětí, které mají sběrnici širokou 64 bitů, tyto paměti jí původně měly širokou pouze 16 bitů, takže musely využívat až čtyřnásobnou frekvenci (800 MHz v základní verzi), aby se srovnaly s SDRAM. Tyto paměti musely pracovat se dvěma připojenými moduly. Novější verze dokáží pracovat i s jedním modulem. Tyto paměti využívají moduly typu *RIMM*.

V předchozích kapitolách jsme nastínili princip vyhledávání paměťového místa. Nejprve je vybrána příslušná banka paměti, poté dochází k vyhledání řádku a sloupce. K vyhledání požadovaných dat potřebují paměti znát provedení následujících příkazů, které jsou charakterizovány příslušnou dobou. Tato doba je uváděna v hodinových cyklech *T*. Ve specifikacích pamětí se pak dovídáme číselné parametry udávané právě v těchto hodinových cyklech.

Operace	Příslušný čas	Význam
RAS	tRAS	Row Address Strobe - Vyhledání řádku
RCD	tRCD	RAS to CAS Delay- Prodleva mezi vyhledáním řádku a sloupce
CAS L	tCL	Column Access Strobe Latency – Prodleva při vyhledání sloupce
RAS Precharge	tRP	Row Precharge - Obnova dat

Tab. 10.4 Označení paměťových operací a časových intervalů

Po vybrání správného banku paměti je nezbytné vyhledat příslušný řádek. Řadič paměti vyvolá stav paměti **Active** příkazem RAS – *Row Address Strobe*. Tato operace zabere více hodinových cyklů a po

jejich dokončení bude řádek paměti vyhledán. Poté dochází k časové prodlevě mezi hledáním řádku a sloupce (*RAS to CAS Delay*). Následuje vyhledání sloupce vyžadující čas *TCL* neboli *Column Address Strobe Latency Time*, na který je kladen důraz ve specifikacích paměti. Po tomto časovém úseku jsou již data přečtena.



Pojem k zapamatování: Časování paměti

Ve specifikacích paměťových modulů je uváděná posloupnost doby jednotlivých příkazů v hodinových cyklech **T**.

Jedná se o posloupnost **tRAS – tRP – tRCD – tCL / T**, často udávaná v opačném pořadí, tak aby **tCL** byl první údaj a nejdelší doba **tRAS** na konci.

10.4 Fyzické uspořádání paměti (paměťové moduly)

Fyzickým uspořádáním paměti se rozumí osazení paměťových čipů do jednotlivých modulů. Paměťový modul se skládá z těchto čipů a působí navenek jako jeden čip o velké kapacitě. Celková velikost paměti, kterou máme možnost použít, je omezena procesorem a čipovou sadou základní desky. Faktem je, že adresovatelnost paměti procesorem není limitujícím faktorem, neboť již procesory Pentium II a III umožnili podporu až 4 GB RAM, což bylo na svou dobu v praxi nerealizováno. Čipové sady však již limitující byly, neboť podporovaly použití daleko menší velikosti paměti. Proto se také čipové sady vyvíjely rychleji než procesory.

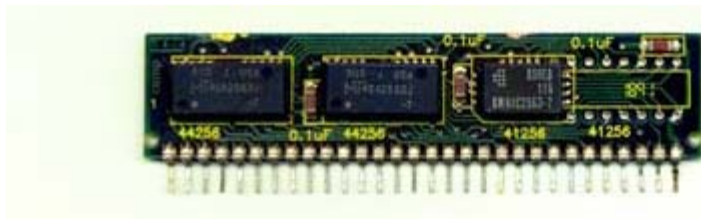
V minulosti byly paměťové čipy přímo zasazeny na základní desku. Toto řešení bylo nahrazeno paměťmi, kde byly paměťové čipy umístěny na malé destičce neboli paměťovém modulu. Tyto moduly se vyráběly v několika fyzických uspořádáních. Prvním tímto modulem byl modul SIPP.



Obr. 10.10 Paměťový modul

- **SIPP**

SIPP (*Single In-Line Pin Package*). Jedná se o moduly podobné modulům 30-pin SIMM. Moduly SIPP (obr. 8.11) mají stejný počet stejně rozmístěných vývodů. Jediný rozdíl je ve tvaru vývodů, které jsou tvořeny úzkými piny.



Obr. 10.11 Modul SIPP

- **SIMM**

Dlouhou dobu, prakticky až do roku 1997, byly využívány moduly SIMM. SIMM (*Single In Line Memory Module*). Tyto moduly byly ve svých starších verzích 30 pinové. Později je nahradily moduly, které měly pinů 72 (viz. obr. 8.12). 30 pinové moduly měly šířku přenosu dat 8 bitů (+ 1 bit pro podporu parity) a 72pinové byly 32bitové (až 4 paritní bity). Kapacita 30pinových byla až 4 MB a u 70pinových se dosahovalo kapacity až 32 MB.



Obr. 10.12 Modul SIMM 72 pin



Pojem k zapamatování: Kontrola parity a ECC kód

Moduly s kontrolou parity detekují logické chyby pomocí paritního bitu. Ten je kontrolou osmi po sobě jdoucích bitů, tedy 1 bytu. Kontrola parity na chybu upozorní, nikoliv opravuje. Kód ECC (*Error Corecting Code*) je naopak schopen následné opravy jednoho bitu. Umožňuje i detekci dvou chybných bitů, avšak bez opravy. Je však ověřeno že s 98% jsou logické chyby tvořeny pouze chybou jednoho bitu.

- **DIMM**

V počítačích s procesory Pentium Pro a Pentium II se začaly využívat paměťové moduly DIMM (*Dual In Line Memory Module*). Tento modul má již 168 vývodů, datová šířka je 64 bitů bez kontroly parity nebo 72 bitů s paritní kontrolou či podporou ECC. Jak již z názvu vyplývá, datové vývody jsou po obou stranách, přičemž počet pinů na každé straně modulu je jiný. Standardně pracuje s napětím 3,3 V. Existovaly moduly i pro 5 V pro jiné typy počítačů, avšak vzájemná záměna není povolena. Moduly DIMM používají odlišný způsob detekce paměti v systémové desce oproti modulům SIMM. Detekce je realizovaná umístěním malé paměti EEPROM či flash ROM, ve které jsou uložena data o daném paměťovém modulu. Při detekci přítomnosti pak jsou tato data čtena speciálními vývody a poskytnuta čipové sadě. Paměťové moduly DIMM jsou na první pohled rozeznatelné dvěma klíčovými výřezy na straně vývodu.



Obr. 10.13 Modul DIMM 168 pin

- **DDR a DDR2 DIMM**

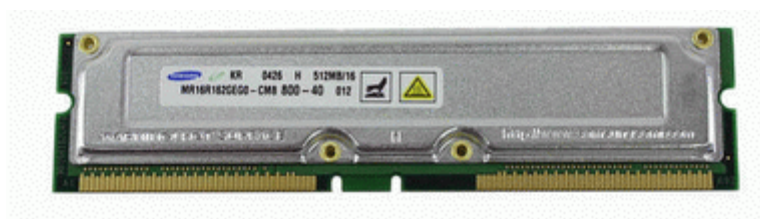
DDR SDRAM (*double-data-rate synchronous dynamic random access memory*) je typ paměti, který vývojově následoval klasický paměťový modul DIMM SDRAM. Dosahuje vyššího výkonu než předchozí typ SDRAM tím, že přenáší data na obou koncích hodinového signálu. Tento přístup zvyšuje efektivní výkon téměř dvakrát bez nutnosti zvyšování frekvence sběrnice. Tento modul má 184 pinů a jeho maximální kapacita dosáhla 1024 MB. Napájecí napětí je 2,5 V. Rozměr paměťového modulu se zvětšil a byl odstraněn jeden klíčový výřez. Modul DDR2 má 240 pinů a není tedy kompatibilní s modulem DDR.



Obr. 10.14 Modul DDR DIMM 184 pin (vlevo) a DDR2 240 pin (vpravo)

- **RIMM**

Modul RIMM se díky své ceně neprosadil a byl zcela vytlačen moduly DDR SDRAM. Každý modul RIMM má 184 pinů symetricky rozdělených na obě strany. Napájecí napětí je 2,5 V. Informace o paměti má modul uložen v SPD zařízení tvořené pamětí flash ROM.



Obr. 10.15 Modul RIMM



Další zdroje

1. Minasi, M. *Velký Průvodce Hardwarem*, Praha, Grada 2003, počet stran: 768. ISBN 80-247-0273-8.
2. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
3. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
4. How Stuff Works server How works RAM [online] dostupný z <http://computer.howstuffworks.com/ram.htm>.
5. Oficiální stránky standardu Jedec [online] dostupný z <http://www.jedec.org/>
6. PC Guide Server: DRAM Technologies [online] dostupný z http://www.pcguides.com/ref/ram/tech_SDRAM.htm
7. Vývoj modulů DRAM a operační paměti – Svět Hardware [online] dostupný z http://www.svethardware.cz/art_doc-53D8F3993772ECFBC1257205005DA285.html



Shrnutí kapitoly

Paměť je základním prvkem jakéhokoliv počítače. Paměti můžeme rozdělit do několika skupin, mimo jiné takto: vnitřní - ROM (PROM, EPROM, EEPROM, Flash), RAM (DRAM – dynamic RAM, SRAM – static RAM) a vnější (magnetické, magneticko optické, optické, polovodičové).

Polovodičové paměti jsou založeny na logických obvodech, které jsou vytvořeny na polovodičových čípech. Rozlišujeme paměti typu ROM (Read Only Memory) a RAM (Random Access Memory). Paměť se skládá z adresových vstupů, datových vstupů a výstupů, řídicích vstupů a napájecích vstupů. Základními parametry charakterizující paměť jsou kapacita (maximální velikost uložených dat), přístupová doba (doba od požadavku po zpřístupnění dat) a přenosová rychlost (doba čtení a zápisu dat).

Paměť RAM (Random Access Memory) je na rozdíl od ROM energeticky závislá. To znamená, že po odpojení zdroje energie se tato paměť vymaže. Tyto paměti jsou určeny pro zápis i čtení. Rozlišujeme dva typy těchto pamětí a to: Statické (SRAM) a dynamické (DRAM).

Paměťové buňky SRAM (Static Random Access Memory) jsou tvořeny bistabilním klopným obvodem. Jednotlivé paměťové buňky bývají uspořádány do matice, jejich výběr je prováděn přes řádkové a sloupcové vodiče. Přes výběr řádků je zapsaná informace připojena na datové vodiče. Adresace paměťové matice je zajišťována přes dekodér řádků a dekodér sloupců. Data jsou v budících obvodech a přivedena na datovou sběrnici.

Paměťové buňky DRAM (Dynamic Random Access Memory) jsou zde realizovány pomocí kondenzátorů. Z tohoto důvodu je zde nutná pravidelná obnova uložených informací, neboť kondenzátory se vybíjejí díky svodovým odporům i v době, kdy je paměť připojena ke zdroji elektrického napájení. Obnova je prováděna několikrát za sekundu. Napětí v kondenzátorech odpovídá logické 0 nebo 1. Buňka paměti DRAM je velmi jednoduchá a dovoluje vysokou integraci a nízké výrobní náklady. I když mají vyšší přístupové doby než paměti SRAM, jsou mnohem více používány, neboť jejich cena je mnohem nižší.

Kritériem posuzování rychlosti paměti je její přístupová doba. Tato hodnota je vyjádřena v

nanosekundách. V oblasti procesorů, sběrnic a pamětí se rychlost vyjadřuje taktovací frekvencí v MHz. Proto je nutné najít vztah mezi délkou cyklu v nanosekundách a frekvencí v MHz.

Po obdržení požadavku na paměťovou operaci je nejprve vyžádaný bit paměti vyhledán, až poté je proveden samotný přenos. Zpoždění – latence paměti je doba potřebná pro vyhledání konkrétního bitu.

DRAM pamětí existuje několik základních typů vývojově navazujících: FPM RAM, EDO RAM, BEDO RAM, SDRAM a její následovníci DDR a DDR2 SDRAM.

DDR SDRAM (*double-data-rate synchronous dynamic random access memory*) je typ pamětí, který vývojově následoval klasický paměťový modul DIMM SDRAM. Dosahuje vyššího výkonu než předchozí typ SDRAM tím, že přenáší data na obou koncích hodinového signálu. Tento přístup zvyšuje efektivní výkon téměř dvakrát bez nutnosti zvyšování frekvence sběrnice. Tento modul má 184 pinů a jeho maximální kapacita dosáhla 1024 MB. Napájecí napětí je 2,5 V. Rozměr paměťového modulu se zvětšil a byl odstraněn jeden klíčový výřez. Modul DDR2 má 240 pinů a není tedy kompatibilní s modulem DDR. Ve specifikacích paměťových modulů je uváděná posloupnost doby jednotlivých příkazů v hodinových cyklech T. Jedná se o posloupnost tRAS – tRP – tRCD – tCL / T, často udávaná v opačném pořadí, tak aby tCL byl první údaj a nejdelší doba tRAS na konci.



Kontrolní otázka 10.1

Definujte možné způsoby rozdělení paměti.



Kontrolní otázka 10.2

Zakreslete paměťovou buňku ROM, PROM, SRAM, DRAM.



Kontrolní otázka 10.3

Popište vývojové verze pamětí DRAM v PC.



Kontrolní otázka 10.4

Definujte pojem časování paměti a s tím související pojmy a označení paměťových operací.



Kontrolní otázka 10.5

Jaké paměťové moduly fyzického uspořádání znáte?



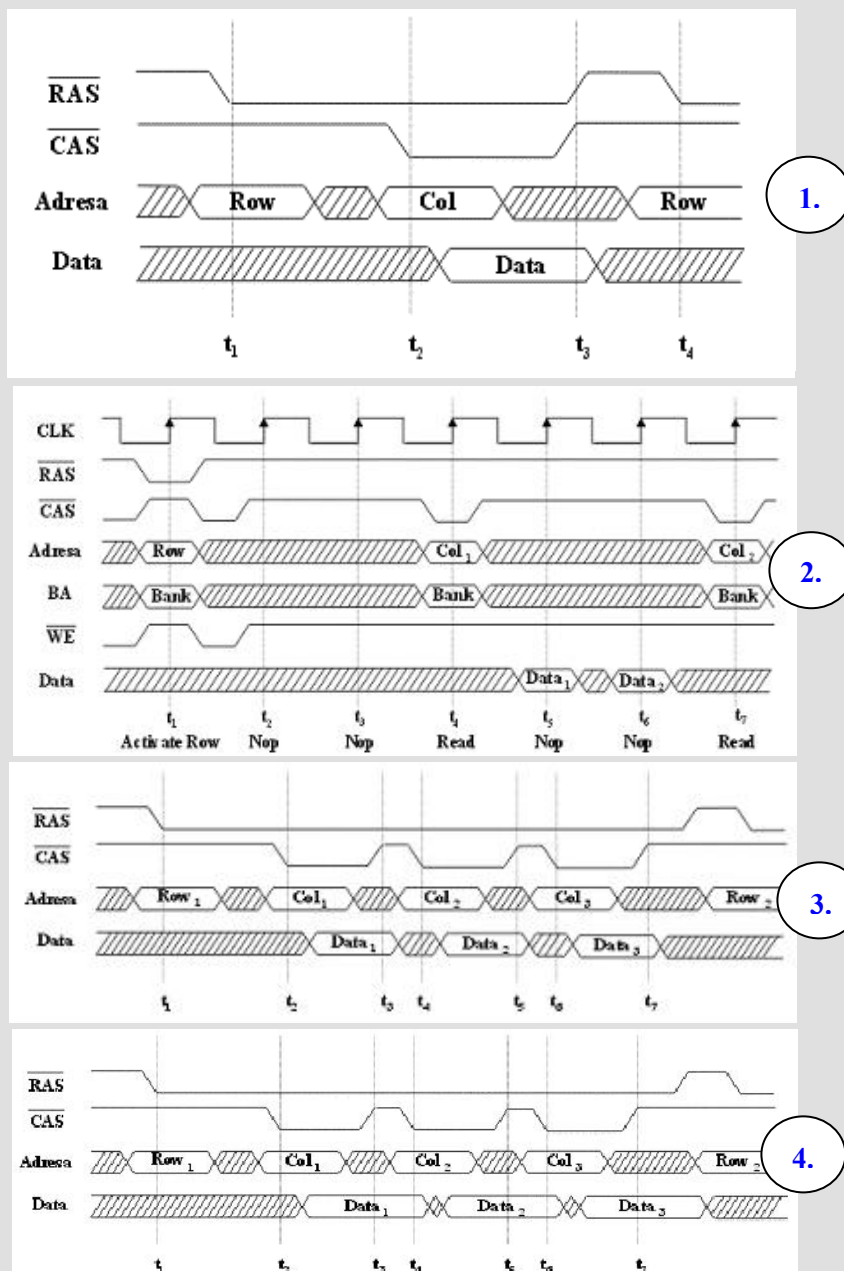
Kontrolní otázka 10.6

Co je to kontrola parity, kód ECC, pojem latence a přístupová doba paměti?



Úkol k řešení 10.1

Přiřaďte jednotlivá časování k typu paměti.



A.) původní DRAM B.) FPM RAM C.) EDO RAM D.) SDRAM

11. Pevné disky, rozhraní a standardy



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- popsat vývoj rozhraní pevných disků
- definovat jednotlivé verze rozhraní ATA
- rozdělit jednotlivé standardy podle přenosové rychlosti
- vysvětlit odstranění problémů s omezením kapacity disku při adresování
- popsat složení pevného disku
- charakterizovat pojem geometrie pevného disku
- vysvětlit podstatu souborových systémů



Výklad

11.1 Vývoj rozhraní pevných disků

Definovat rozhraní pevného disku můžeme jako zařízení, která zprostředkovávají komunikaci mezi pevným diskem a ostatními částmi počítače. Rozhraní pevného disku určuje způsob komunikace a tím i typ disku (popř. jiného zařízení, tj. např. mechaniky CDROM, DVDROM, ZIP, páskové jednotky, atd.), který je možné k němu připojit. První ve větší míře používané rozhraní pevných disků pro počítače PC bylo rozhraní ST506, vyrobeno firmou *Shugart Technologies* (dnešní Seagate). Toto rozhraní dokázalo pracovat s disky, které měly maximálně 16 hlav a bylo možné k němu připojit maximálně dva disky. Jednalo se o rozhraní, které bylo poměrně náchylné na rušení a vyžadovalo tedy co možná nejkratší a kvalitní kabeláž. ST506 bylo s disky spojeno dvěma kabely: 20 žilovým kabelem pro přenos dat (pro každý disk zvláštní kabel) a 34žilovým kabelem pro přenos řídicích informací (společný pro oba disky). Rozhraní mělo komplikovanou komunikaci s diskem. Rozhraní totiž neumí přikázat disku, aby vystavil hlavy na nějaký konkrétní cylindr. Je možné vysílat pouze příkazy pro přesunutí hlav na následující popř. předcházející cylindr. Rozhraní také nedisponuje žádnou funkcí pro programové zjištění informací o geometrii připojených pevných disků.

Začátkem 80. let vzniká jako snaha o standardní rozhraní pro připojování periférií rozhraní ESDI (*Enhanced Small Device Interface*), které by nahradilo rozhraní ST506. Jedná se o výrazně zlepšené rozhraní ST506, u kterého jsou data přenášena sériově a řídicí informace paralelně. Ke zlepšení oproti svému předchůdci patří: podpora disků, které mohou mít až 256 hlav, dovoluje podstatně vyšší přenosovou rychlost dat a to až 24 Mb/s. Disková jednotka může zasílat informace o své konfiguraci a je možné programově zjistit informace o geometrii pevného disku. Dekódování informací je prováděno přímo na desce pevného disku, což snižuje náchylnost na rušení a dovoluje použití delších propojovacích kabelů. Rozhraní ESDI teoreticky dovoluje připojit i jiná zařízení, než jsou pevné disky. Tato zařízení se však nikdy ve větší míře nevyráběla. Rozhraní ESDI zachovává stejnou kabeláž jako rozhraní ST506 a dovoluje také připojit maximálně dvě zařízení.

V roce 1986 bylo navrženo firmami Western Digital a Compaq rozhraní IDE (*Integrated Device Electronics*). Zkratka IDE je poměrně obecná a označuje kteroukoliv mechaniku s vestavěným řadičem, rozhraní IDE se dnes častěji označuje jako ATA (*AT Attachment*).

Základní charakteristiku rozhraní tvoří následující vlastnosti:

- Zapojení diskových jednotek IDE (ATA) se provádí pomocí 40vodičového kabelu (zpočátku).
- Existují i verze (používané nejčastěji u pevných disků pro přenosné počítače), které používají kabel s 44 vodiči. V rámci tohoto 44vodičového kabelu je vedeno i napájení pevného disku.
- Novější verze pracující s protokoly Ultra-DMA 4 až Ultra-DMA 6 vyžadují použití 80vodičového kabelu.
- 80vodičový kabel minimalizuje vlivy okolního rušení a přeslechy mezi vodiči.



Obr. 11.1 Logic board pevného disku

- Přenos dat mezi zařízením (HDD, CD-ROM, atd.) a operační pamětí může být realizován pomocí dvou různých režimů PIO a DMA. U režimu PIO - *Programmed (Processor) Input Output* je přenos dat řízen procesorem. Ve srovnání s režimem DMA (*Direct Memory Access*) generuje větší zátěž procesoru, neboť veškerá přenášená data procházejí přes procesor. DMA režim využívá speciální obvod (*DMA controller*), který je schopen zprostředkovat přenos dat bez účasti procesoru.

PIO režimy pro standard ATA	
PIO 0	maximální přenosová rychlost je 2 – 3,3 MB/s, definován již na ATA 1
PIO 1	maximální přenosová rychlost je 5,22 MB/s, definován již na ATA 1
PIO 2	maximální přenosová rychlost je 8,33 MB/s, definován již na ATA 1
PIO 3	pro VL bus a PCI maximální přenosová rychlost je 11,1 MB/s, ATA 2
PIO 4	maximální přenosová rychlost je 16,6 MB/s, ATA 2
PIO 5	maximální přenosová rychlost je 22 MB/s, podporován BIOSem základních desek, ale výrobci disků nebyl implementován
Režimy DMA před standardy Ultra DMA	
DMA 0	maximální přenosová rychlost je 2,08 MB/s
DMA 1	maximální přenosová rychlost je 4,17 MB/s
DMA 2	maximální přenosová rychlost je 8,33 MB/s
DMA Multiword 0	maximální přenosová rychlost je 4,17 MB/s
DMA Multiword 1	maximální přenosová rychlost je 13,3 MB/s
DMA Multitword 2	maximální přenosová rychlost je 16,6 – 22 MB/s

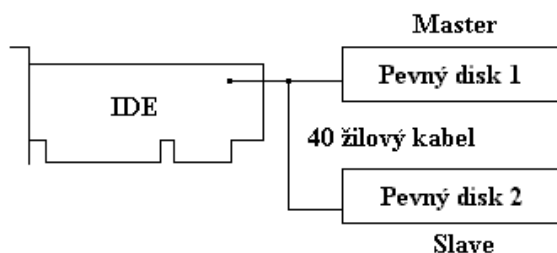
Tab. 11.1 PIO a DMA režimy prvních ATA standardů

Verze rozhraní ATA bychom mohli rozdělit podle určení typu sběrnice:

- XT IDE:
 - určeno pro 8bitovou sběrnici PC bus,
 - data jsou přenášena po 8 bitech,
- MCA IDE:
 - určeno pro 16bitovou sběrnici MCA (počítače IBM PS/2),
 - data jsou přenášena po 16 bitech,
- ATA IDE:
 - určeno pro 16bitovou sběrnici ISA, popř. pro další typy rozšiřujících sběrnic (VL-bus, PCI),
 - data jsou přenášena po 16 bitech,
- SATA (Serial ATA).

11.2 Verze rozhraní ATA

Rozhraní ATA 1 bylo používáno od roku 1986, ale standardizováno až v roce 1994. Je všeobecně označováno jako IDE (ATA IDE). Rozhraní dovoluje programově zjistit informace o geometrii pevného disku. Používá konektory a vodiče se 40 (resp. 44) vývody a dovoluje připojit maximálně dvě zařízení. Rozhraní ATA 1 podporuje pouze připojování pevných disků, jiná zařízení (např. mechaniky CD-ROM, páskové jednotky apod.) nebyly oficiálně podporovány.



Obr. 11.2 Master – Slave připojení disků rozhraní ATA 1

Rozhraní ATA 1 je charakterizováno následujícími vlastnostmi:

- podporuje přenosy v režimech:
 - PIO 0 až PIO 2,
 - Single-word DMA 0 až Single-word DMA 2,
 - Multi-word DMA 0.
- maximální přenosová rychlost je 8,33 MB/s.
- Rozhraní ATA-1 používá pro adresaci dat na pevném disku schéma, které vychází z omezení rozhraní ATA a z omezení BIOSu:
 - 4 bity: adresace povrchu max. 16 povrchů,
 - 10 bitů: adresace cylindru max. 1024 cylindrů,
 - 6 bitů: adresace sektoru max. 64 sektorů.
- Použití tohoto schématu má za následek omezení kapacity pevného disku na 512 MB.

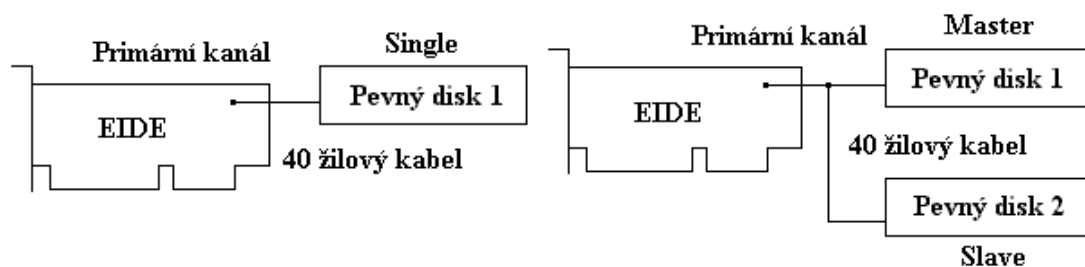
Disky připojované k rozhraním ATA je nutné pomocí propojek (jumperů) nastavit jako:

1. Single: reprezentuje nastavení disku, který je v počítači zapojen jako jediný (tedy zároveň hlavní) - nastavení Single bývá u většiny disků shodné s nastavením Master.
2. Master: reprezentuje nastavení hlavního (řídícího) disku, který je zapojen společně s druhým (podřízeným) diskem.
3. Slave: reprezentuje nastavení podřízeného disku, který je zapojen společně s dalším (hlavním) diskem.

4. Cable Select: reprezentuje nastavení, kdy hlavní a podřízený disk jsou identifikovány podle svého zapojení na kabelu, vyžaduje použití speciálního kabelu, který dovoluje rozlišit řídící a podřízený disk.

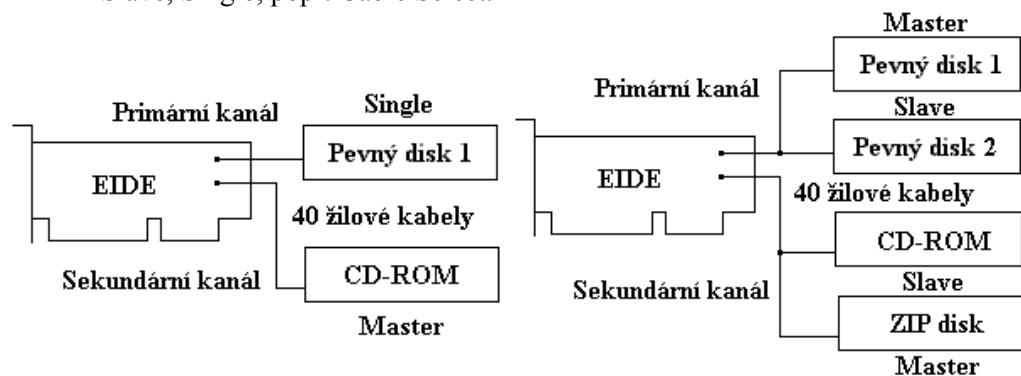
Rozhraní **ATA2** bylo standardizováno v roce 1996. Ve své době bylo také nazýváno Fast-ATA, Fast-ATA 2 nebo EIDE - Enhanced IDE. Označení EIDE bylo firmou Western Digital používáno i pro následující verze rozhraní ATA. Rozhraní je navrženo tak, aby jej bylo možné využít jako obecné rozhraní mezi hostitelským systémem (počítačem) a jakýmkoliv zařízením pro ukládání dat. V rámci rozhraní ATA-2 je často integrováno i rozhraní odpovídající (v této době ještě samostatnému) standardu ATAPI (*AT Attachment Packet Interface*). Standard ATAPI poskytoval jednotné rozhraní pro připojování zařízení jako jsou např.: mechaniky CD-ROM, CD-R, CD-RW, jednotky ZIP, LS-120, páskové jednotky. Rozhraní ATA2 charakterizují následující vlastnosti:

- Podporuje přenosy dat v rychlejších režimech:
 - PIO 0 až PIO 4,
 - Single-word DMA 0 až Single-word DMA 2,
 - Multi-word DMA 0 až Multi-word DMA 2.
- Maximální přenosová rychlost rozhraní ATA-2 je 16,67 MB/s.
- Umožňuje připojit až čtyři zařízení.
- Při práci s pevným diskem používá adresovací metodu LBA - Linear (Logical) Block Address.



Obr. 11.3 Připojení jednoho disku v režimu single nebo dvou disků v režimu Master - Slave

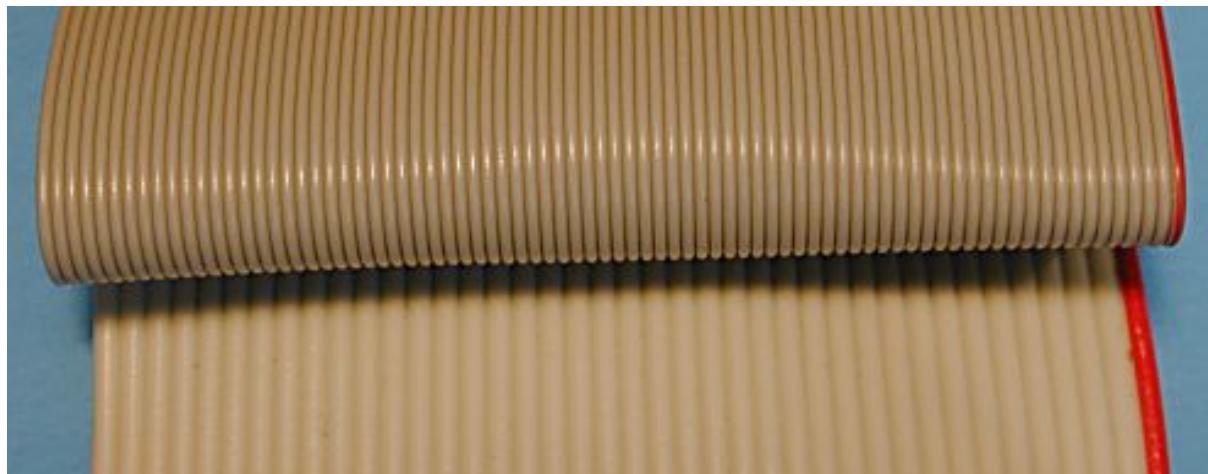
- Zařízení, která jsou připojena k rozhraní ATA-2 a vyšším, lze zapojit na dva kanály: primární (primary channel) a sekundární (secondary channel).
- Na každý kanál je možné připojit maximálně dvě zařízení pomocí 40žilového kabelu, který je shodný s kabelem rozhraní ATA-1.
- Na obou kanálech je potom u jednotlivých zařízení nutné nastavit propojky do pozic Master, Slave, Single, popř. Cable Select.



Obr. 11.4 Připojení více zařízení na primární a sekundární kanál

V roce 1997 bylo standardizováno rozhraní **ATA 3**, o rok později přišlo ATA 4. Rozhraní ATA 3 přináší pouze drobná vylepšení (oproti rozhraní ATA-2), bývá rovněž označováno termínem EIDE. Hlavní novinkou je přidání podpory pro technologii S.M.A.R.T.

ATA 4 bylo označováno také jako ATA/ATAPI-4 nebo Ultra-ATA 33. Součástí rozhraní se stává i dříve používaný (samostatný) standard ATAPI. Kromě již dříve používaných přenosových režimů jsou podporovány i nové režimy Ultra-DMA 0 až Ultra-DMA 2 přesto, že režimy Ultra-DMA 0 a Ultra-DMA 1 nebyly výrobci nikdy implementovány. Režim Ultra-DMA 2 je podle své přenosové rychlosti (33 MB/s) také označován jako Ultra-DMA 33 nebo Ultra-ATA 33. Rozhraní ATA 4 definuje nový propojovací kabel s 80 vodiči.



Obr. 11.5 Rozdíl 80vodičového kabelu oproti 40vodičovému



Pojem k zapamatování: Použití 80 vodičového kabelu rozhraní ATA4 a vyšší.

Použití 80vodičového kabelu u rozhraní ATA 4 zatím není vyžadováno, ale je doporučováno. Tato skutečnost je vynucena vyšší přenosovou rychlostí režimu Ultra DMA 2 a tím i větší náchylností k chybám (šumu).

Chyby přenosu mohou být zapříčiněny okolním elektromagnetickým rušením nebo přeslechy mezi jednotlivými vodiči. Konektory použité na těchto kabelech (odpovídajících discích) jsou opět 40vývodové a jejich zapojení je shodné se zapojením dřívějších disků. Zbývajících 40 vodičů v 80žilovém kabelu je určeno pro uzemňující signál. Tyto vodiče jsou rozmístěny mezi původními 40 vodiči a nejsou vyvedeny na konektor. Nových 40 vodičů tedy slouží pouze jako stínění, čímž zamezují rušení a přeslechům. Datové přenosy jsou zabezpečeny pomocí CRC - *Cyclical Redundancy Checking*.

V roce 1999 byl zveřejněn standard ATA 5, který je nazýván také ATA/ATAPI-5 nebo Ultra-ATA 66. Přináší podporu nových přenosových režimů Ultra-DMA 3 a Ultra-DMA 4. Režim Ultra-DMA 4 je podle své přenosové rychlosti (66 MB/s) také označován jako Ultra-DMA 66 nebo Ultra-ATA 66. Rozhraní dovoluje automatickou detekci kabelu se 40, resp. 80 vodiči. Provoz režimů s vyšší přenosovou rychlostí než 33 MB/s (Ultra-DMA 3 a Ultra-DMA 4) již vyžaduje použití nového 80vodičového kabelu.

80vodičový kabel je také vyžadován pro přenosové režimy Ultra-DMA 5 a Ultra-DMA 6, které jsou používané u následujících verzí rozhraní ATA

Rozhraní ATA 6 bylo zveřejněno v roce 2000 a ještě po dlouhé dva roky upravováno, než bylo definitivně standardizováno. Bylo nazýváno také jako ATA/ATAPI-6 nebo Ultra-ATA 100. Podporuje

nový přenosový režim Ultra-DMA 5 (Ultra-DMA 100, Ultra-ATA 100) s maximální přenosovou rychlostí 100 MB/s. Rozšiřuje režim LBA, který místo 28bitové adresy používá adresu 48bitovou, max. velikost adresace pevného disku je tak 144 PB (Petabyte). Charakteristikou posledního standardu ATA 7 vyvíjeného od 2001 až do 2005 je přenosový režim Ultra-DMA 6 (Ultra-DMA 133, Ultra-ATA 133) o rychlosti 133 MB/s. Rozhraní ATA se svou verzí 7 dostává na hranici technologických možností, hlavně v oblasti časování signálů. Paralelní režim již dále za stejných podmínek nelze donekonečna zrychlovat a tak se přechází na sériové rozhraní označované SATA. Ve vývoji je sice ještě rozhraní ATA 8, avšak pouze pro hybridní disky. Výrobci čipových sad již ignorují rozhraní ATA 7, neboť pro maximální využití ATA disků dostačuje režim Ultra DMA 5 předchozího standardu. V současnosti se tedy spíše setkáváme s výběrem možností rozhraní ATA 6 či přechodu na sériové SATA rozhraní.

Proužek na vodiči označuje
signál pinu 1



Klíč na konektoru proti zasunutí
V nesprávné poloze

Obr. 11.6 Datový kabel rozhraní ATA

Příkazy rozhraní ATA

ATA IDE bylo navrženo podle původního řadiče firmy IBM u počítačů AT. Všechny disky standardu ATA musí podporovat všech 8 příkazů řadiče. Nejdůležitějším příkazem je *Identify Drive* využívaný biosem při automatické detekci disku. Na základě tohoto příkazu se zjistí následující údaje:

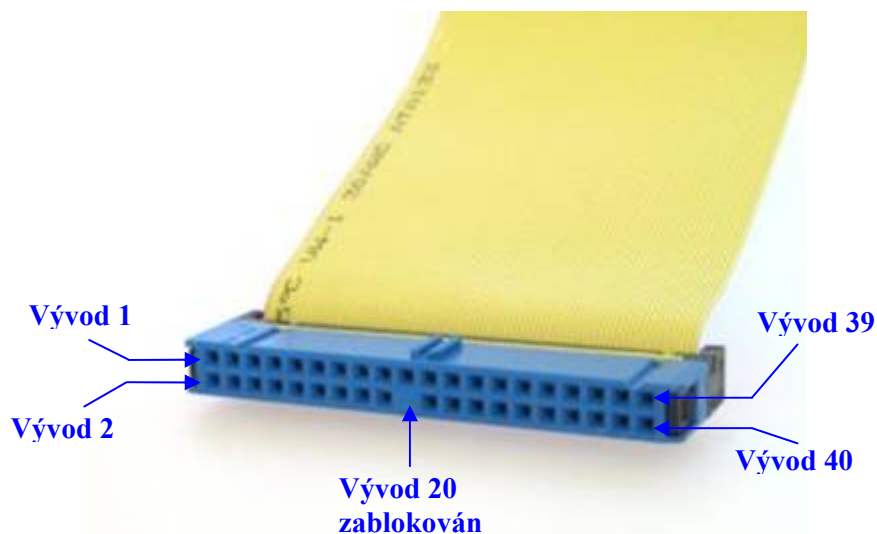
- Počet logických adresových bloků při použití režimu LBA.
- Počet fyzických cylindrů, hlav a sektorů při použití režimu P-CHS.
- Počet logických cylindrů, hlav a sektorů při použití režimu L-CHS.
- Přenosové režimy a rychlosti podporované diskem.
- Výrobce a typ disku.
- Verze firmwaru.
- Výrobní číslo.
- Typ a velikost zásobníku, info o mezipaměti.



Pojem k zapamatování: Omezení kapacity disků ATA

Rozhraní ATA ve verzi 1 až 5 omezovaly velikost disku na 136,9 GB. Kombinací s verzí biosu se kapacita omezovala na 528 MB nebo 8,4 GB.

Původní ovladače disků byly volány softwarovým přerušením INT 13h. Konkrétní adresa sektoru byla určena číslem cylindru, hlavy a tohoto sektoru. Zmíněné adresování CHS navrhla firma IBM pro své počítače řady PC XT. Omezení tak platí pro všechny BIOSy a jejich parametry vidíme v tabulce 11.2.



Obr. 11.7 Konektor rozhraní ATA

Maximální adresování disku biosem bylo (jak vidíme v tabulce) 8,4 GB. Omezení rozhraním ATA je větší a to až zmiňovaných 136,9 GB. V čem je tedy problém? Ve skutečnosti se uvedená omezení Biosu a rozhraní ATA zkombinuje a ve výsledné kombinaci tak musíme počítat vždy s menším počtem bitů vyhrazeným pro vyjádření cylindrů, hlav a sektorů.

Pole	Velikost	Max. hodnota	Rozsah využitelných hodnot
Omezení CHS			
Cylindr	10 bitů	1024	0 – 1023
Hlava	8 bitů	256	0 - 255
Sektor	6 bitů	64	1 - 63
Omezení rozhraní ATA			
Cylindr	16 bitů	65636	0-65635
Hlava	4 bity	16	0-15
Sektor	8 bitů	256	1-255
Parametr	Bios	Rozhraní ATA	Výsledná kombinace
Kombinovaná omezení			
Cylindr	1024	65636	1024
Hlava	256	16	16
Sektor	63	255	63
Počet sektorů	16 515 072	267 386 880	1 032 192
Kapacita	8,4 GB	136,9 GB	528 MB

Tab. 11.2 Kombinací omezení ATa a biosu se dostáváme na hranici 528 MB



Kontrolní otázka: Jakým způsobem jsme se v předchozí tabulce dostali k výsledné kapacitě disku?

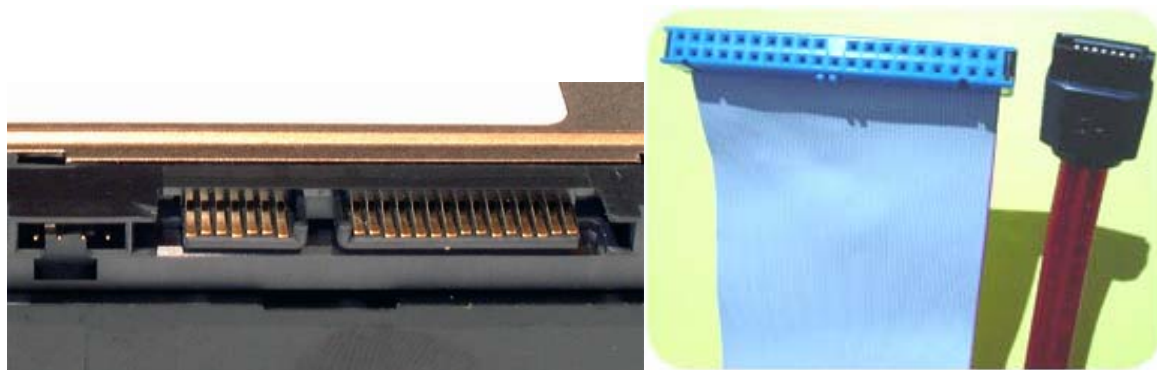
V roce 1994 bylo navrženo řešení překročení hranice 528 MB. Výsledkem je překlad CHS parametrů maximálních hodnot rozhraní ATA na maximální hodnoty biosu a naopak. Jedná se o geometrický překlad adres LBA (*Logical Block Adress*). Druhým způsobem převodu byl geometrický přesun bitů. Fyzické adresy disku P-CHS se překládají na adresy biosu L-CHS. Výhodnější, bezproblémový překlad však byl LBA. V důsledku překladu je výsledné omezení dané pouze biosem 8,4 GB.

Překročení omezení 8,4 GB bylo přijato standardem v roce 1996. Jednalo se o rozšíření příkazů softwarového přerušení INT13H na 64 bitů, 2^{64} tedy umožňuje adresovat až 9,44 ZB. To je však rozšíření biosu. Rozhraní ATA v konkrétních verzích ve své době podporuje maximální počet sektorů 28bitovým adresováním, 2^{28} sektorů adresuje 137,44 GB. Tak velký diskový prostor byl ve své době nepředstavitelně velký a tak na několik let problém s omezením disku ustal. V roce 2001 se začaly na trhu objevovat pevné disky, které se k této kapacitě začaly přibližovat. Proto bylo nutné daná omezení opět řešit. Výsledkem bylo rozšíření 48 bitového adresování u standardu ATA-6. Kapacita, která je tímto adresováním hraniční, je až 144 PB(petabytů). Podle Mooreova zákona o zdvojnásobování kapacity disků v rozmezí 18 – 24 měsíců by se problém dal očekávat po roce 2030, problém s biosem v rozmezí let 2055 až 2070. Nutno podotknout, že Mooreův zákon byl vysloven jen pro zvyšování kapacity za současného zachování technologie.

11.3 Rozhraní Serial ATA

Rozhraní Serial ATA (SATA) využívá pro přenos informací sériovou sběrnici. Podobně jako u rozhraní vyšších standardů ATA je možné k rozhraní SATA připojovat i jiná zařízení než jsou pevné disky (CD-ROM, DVD, ZIP). Každé zařízení je k rozhraní SATA připojeno vlastním kabelem, tj. není nutné provádět nastavení *Single, Master, Slave, Cable Select*.

Konkrétní rozhraní může integrovat několik kanálů (např. 2, 4, 8), pomocí nichž lze jednotlivá zařízení připojovat. Propojovací kabel obsahuje 7 vodičů a může být dlouhý až 1 metr (u ATA může být kabel dlouhý cca 45 cm). Takovýto kabel umožňuje připojení disku na větší vzdálenost (např. ve větší skříni), lepší manipulaci s kabelem a také lepší proudění vzduchu, tj. lepší chlazení celého počítače. Přenosy jsou zabezpečeny pomocí CRC. Rozhraní SATA podporuje tzv. *Hot Plug-In*, tj. připojování (odpojování) zařízení i za chodu počítače. Rozhraní SATA je zpětně kompatibilní s rozhraními ATA, existují adaptéry, pomocí nichž lze připojit ATA zařízení k rozhraní SATA.



Obr. 11.8 Konektory a kabel rozhraní SATA



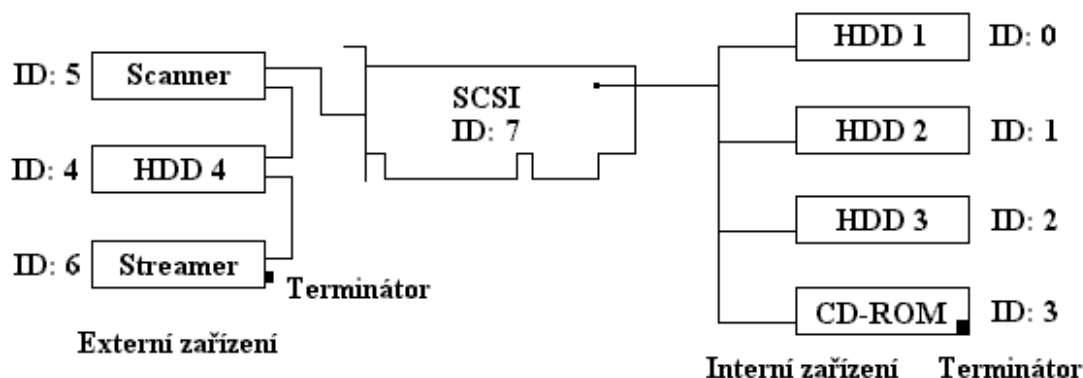
Pojem k zapamatování: Přenosové rychlosti SATA

Pro rozhraní SATA máme definované v současnosti tyto standardní přenosové rychlosti: SATA I: 150 MB/s, SATA II: 300 MB/s, SATA III: 600 MB/s.

Šířka rozhraní SATA je tedy 1 bit. Odolnost proti elektromagnetickému rušení je zajištěno přenosem shodného signálu s opačnou polaritou na sousedních vodičích. Fyzický přenos dat je realizován diferenciálním přenosem NRZ (*Not Return to Zero*). Pro možné zvýšení spolehlivosti při vysokých rychlostech je zavedeno kódování/dekódování dat, které nepovoluje přenos více než čtyř stejných logických hodnot (0 nebo 1) za sebou. V jednom desetibitovém znaku se rovněž nevyskytuje více než šest a méně než 4 logické hodnoty. Tím je zajištěn rovnoměrný tok změn napětí.

11.4 Rozhraní SCSI

Cílem rozhraní SCSI (*Small Computer Systems Interface*) bylo vytvořit standardní rozhraní poskytující sběrnici pro připojení dalších zařízení. První specifikace SCSI-1 byla uveřejněna v roce 1986. Tato specifikace definuje 50vodičovou sběrnici, ke které je možné připojit interní i externí zařízení. SCSI sběrnice musí být na posledních (krajních) zařízeních ukončena tzv. terminátory (zakončovacími odpory). Terminátory realizují impedanční přizpůsobení sběrnice a zabraňují tak odrazům signálů od konce vedení. V opačném případě by se signály odrážely od konce vedení, tím by docházelo k vzájemnému skládání signálů a vzniku stojatých vln. Nepřítomnost terminátorů by tedy měla za následek nefunkčnost celé SCSI sběrnice. Jedna SCSI sběrnice podporuje 8 nebo 16 (podle typu standardu) zařízení. Každé zařízení má přiřazeno své SCSI ID. Jedno ID je přiřazeno vždy řadiči sběrnice, to znamená že maximální počet připojitelných zařízení je 7 resp. 15. Přehled jednotlivých verzí SCSI rozhraní se svými vlastnostmi je uveden v tabulce 11.3.



Obr. 11.9 SCSI rozhraní dovoluje připojit až 7 nebo 15 zařízení (podle typu)

Standard	Technologie	Označení	Rychlost (MHz)	Šířka (bity)	Max. počet zařízení	Přenosová rychlost (MB/s)
SCSI-1	Async	Asynchronous	5	8	7	4
SCSI-1	Fast-5	Synchronous	5	8	7	5
SCSI-2	Fast-5/Wide	Wide	5	16	15	10
SCSI-2	Fast-10	Fast	10	8	7	10
SCSI-2	Fast-10/Wide	Fast/Wide	10	16	15	20
SPI(SCSI-3)	Fast-20	Ultra	20	8	7	20
SPI(SCSI-3)	Fast-20/Wide	Ultra/Wide	20	16	7	40
SPI-2(SCSI-3)	Fast-40	Ultra2	40	8	7	40
SPI-2(SCSI-3)	Fast-40/Wide	Ultra2/Wide	40	16	15	80
SPI-3(SCSI-3)	Fast-80DT	Ultra3	40	16	15	160
SPI-4(SCSI-3)	Fast-160DT	Ultra4	80	16	15	320
SPI-5(SCSI-3)	Fast-320DT	Ultra5	160	16	15	640

Tab. 11.3 Verze standardu SCSI rozhraní a jejich parametry

Rozhraní se SCSI je určeno zejména pro počítače, na které je kladena velká zátěž (počítačové servery). Je používáno i u počítačů jiných, než jsou počítače PC (např. Macintosh, Sun, SGI), může být osazeno i v počítači, který již obsahuje rozhraní ATA, resp. SATA a může obsahovat více kanálů, které dovolují připojit zařízení k více sběrnici a zvýšit tak počet maximálně připojitelných zařízení. Standardy SCSI mají různé přenosové rychlosti dané svým vývojem, Ultra4 podporuje přenosové rychlosti až 320 MB/s, Ultra5 ve vývoji až 640 MB/s.

11.5 Základní součásti pevných disků

Tato kapitola popisuje součásti pevných disků a to ze dvou pohledů: z pohledu konstrukce disků a se samotného použití pro ukládání dat, kterému říkáme geometrie disku. Z pohledu konstrukce se disk skládá z následujících částí:

- **Plotny disku**

Velikost: průměr disků použitých ke konstrukci pevného disku např.: 2", 2^{1/2}", 3^{1/2}", 5^{1/4}" ale i 1".

- **Hlavy pro čtení a zápis**

Obvykle jedna hlava pro každou stranu plotny, hlavy jsou spojeny a zavěšeny na jednom pohyblivém závěsu.

- **Pohon hlav**

Obsahuje mechanismus vystavení hlav, což je mechanismus, pomocí kterého se vystavují čtecí (zapisovací) hlavy na patřičný cylindr. Tento mechanismus může být realizován pomocí krokového motorku - u starších pevných disků (dlouhá přístupová doba, velká citlivost) nebo elektromagnetu – u novějších (moderních) pevných disků.

- **Vzduchové filtry**

Rozlišujeme dva typy filtrů: necirkulační a barometrický. Filtry jsou umístěny uvnitř disku a jsou tedy navrženy tak, aby nebyla nutná jejich výměna.

- **Pohon ploten disku**

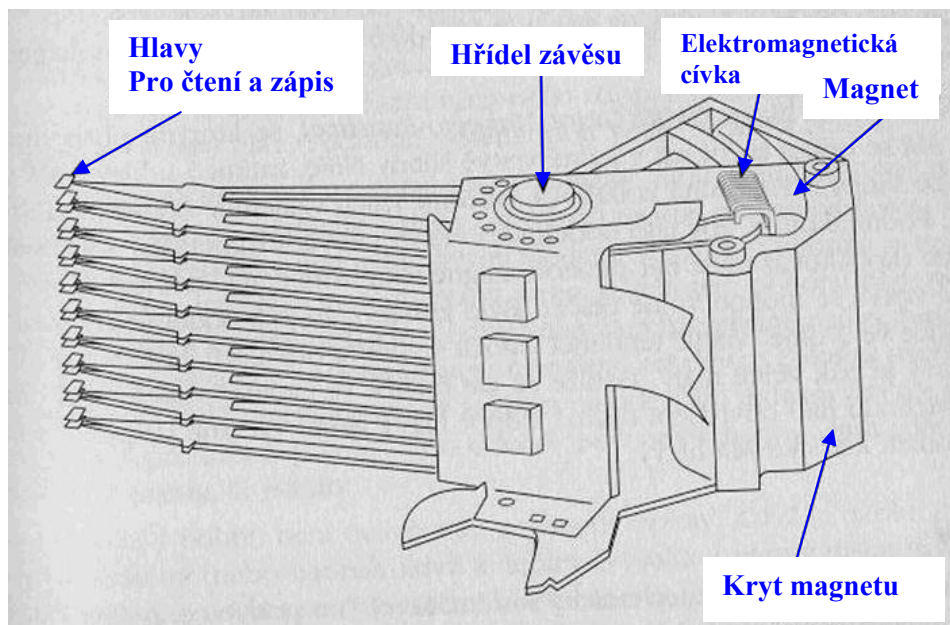
Je tvořen motorem, vždy připojen přímo k hřídeli (žádné převody), nutností je, aby neprodukoval žádné vibrace. To je řešeno pomocí kapalinových ložisek s viskozni mazací tekutinou.

- **Řídicí deska**

Logic board obsahuje elektroniku pro řízení polohování hlav, pohon disku, hlav a předávání dat pro řadič disku.

- **Kabely a konektory**

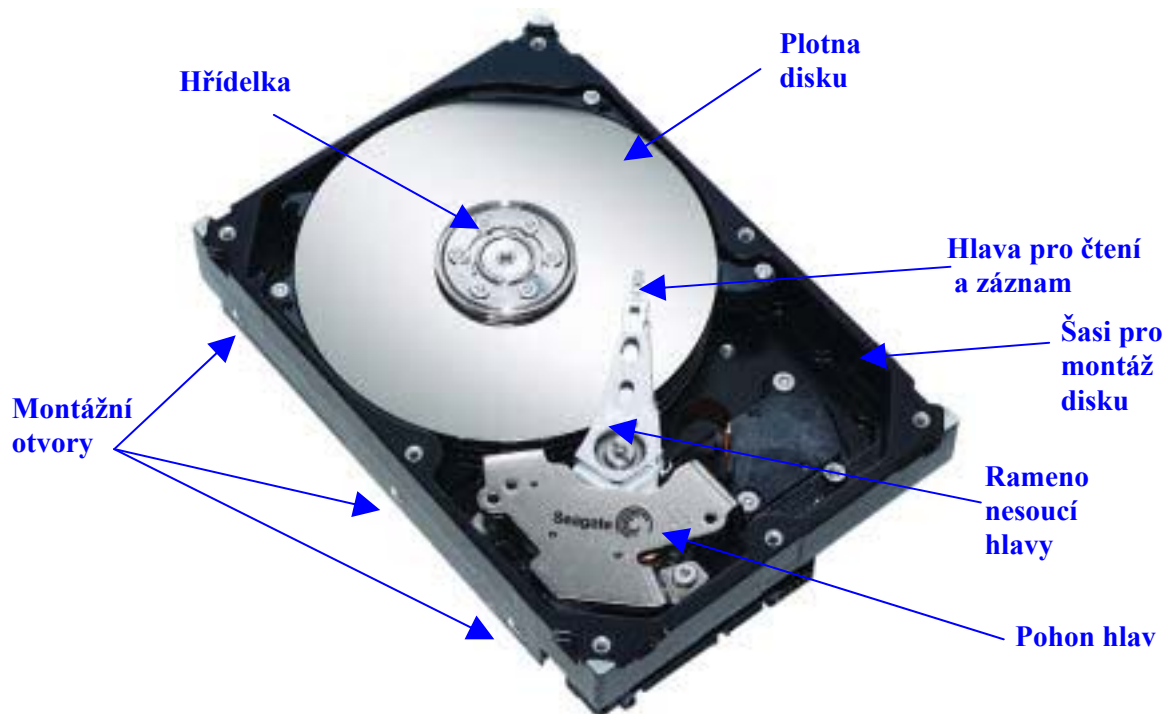
Pevné disky mají konektory pro připojení k systému. Jedná se o datový, napájecí a uzemňovací konektor. Dalšími součástmi jsou konfigurační prvky, spojky a přepínače.



Obr. 11.10 Hlavy a pohon disku

Definice pevného disku

Jak bylo uvedeno výše, disk obsahuje pevné plotny diskového tvaru, na rozdíl od disket se tyto plotny nemohou ohýbat – proto termín pevný disk. Základem disku je tedy několik ploten (umístěných nad sebou) což jsou otáčející se disky. Každá plotna má dvě strany, na které se ukládají data. Plotny se otáčejí stejnými otáčkami. Tyto disky se otáčejí po celou dobu, kdy je pevný disk připojen ke zdroji elektrického napájení, díky tomuto otáčení se v okolí disků vytváří tenká vzduchová vrstva. Nad těmito disky se pohybují hlavy a ukládají a čtou data do soustředných prstenců nazývaných stopy (tracks), stopy jsou rozděleny na segmenty označené jako sektory, z nichž každý pojme 512 Bytů dat. Všechny hlavy jsou připevněny k jednomu závěsu hlav. Vzdálenost hlav od disku je asi 0,3 až 0,6 mikronu. Kapacita je množství informací, které lze na pevný disk uložit a bývá v rozmezí 10 MB - 400 GB. Přístupová doba je doba, která je nutná k vystavení čtecích/zapisovacích hlav na požadovaný cylinder, obvykle je v rozmezí 3,6 - 65 ms. Obvyklá rychlost otáčení je dnes 7200 ot/min nebo dokonce 10000 ot/min, dříve 5400 ot/min nebo 4500 ot/min. Je možné se setkat ale i s 12000 a 15000 ot/min. Přenosová rychlost (*Data transfer rate*) určuje počet bytů, které je možné z disku přenést za jednu sekundu, jedná se o rychlosti 700 kB/s - řádově desítky MB/s. Cache paměť pevného disku je realizována jako paměť typu DRAM a její velikost je až 16 MB.



Obr 11.11 Součásti pevného disku



Pojem k zapamatování: podpora pro technologii S.M.A.R.T.

Technologie S.M.A.R.T (*Self Monitoring Analysis And Reporting Technology*) je standardem umožňujícím předpovídání poruch pevných disků.

Disk sám sleduje určité své parametry a vlastnosti, jejichž změna může indikovat blížící se poruchu.

Technologie S.M.A.R.T. umožňuje uživatele informovat o běžně nepozorovatelných problémech při práci pevného disku, např.:

- chybné čtení (chybný zápis) sektoru,
- kolísání rychlosti otáček,
- teplota uvnitř pevného disku,
- počet realokovaných (vadných sektorů),
- doba provozu disku,
- počet zapnutí pevného disku a další.

Uživatel je tímto upozorňován, že by měl provést zálohu dat (výměnu pevného disku) ještě dříve, než dojde k havárii disku a tím i ztrátě dat. Mezi sledované parametry patří také výška vznášení hlav nad povrchem ploten, počet pokusů o roztočení ploten, počet opakovaných pokusů kalibrace disku a jiné. Některé parametry jednotliví výrobci neuveřejňují a jsou před uživateli utajeny.



Pojem k zapamatování: Geometrie disku

Základními součástmi každého disku jsou tyto pojmy pojmenované často termínem geometrie disku. Jedná se o hlavy disku, cylindry, stopy a sektory.

Geometrie disku udává hodnoty následujících parametrů:

Hlavy disku (heads):

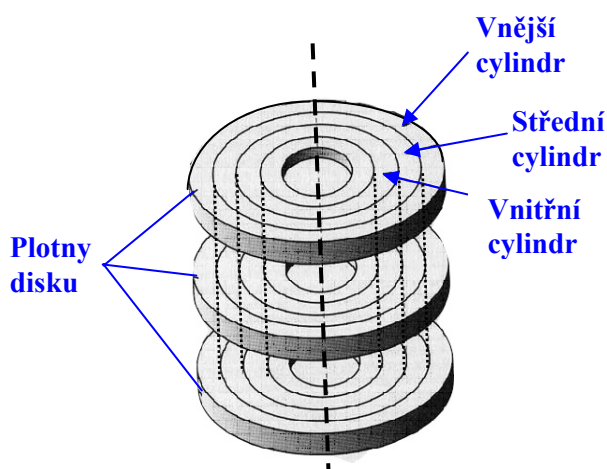
Počet čtecích (zapisovacích) hlav pevného disku. Tento počet je shodný s počtem aktivních ploch, na které se provádí záznam. Většinou každý jednotlivý disk má dvě aktivní plochy a k nim příslušné čtecí (zapisovací) hlavy.

• Stopy disku (tracks):

Počet stop na každé aktivní ploše disku. Stopy disku bývají číslovány od nuly, přičemž číslo nula je číslo vnější stopy disku.

• Cylindry disku (cylinders):

Počet cylindrů pevného disku. Tento počet je shodný s počtem stop, takže i číslování je shodné s číslováním stop. Z tohoto důvodu bývá u pevného disku většinou udáván pouze počet cylindrů.



Obr 11.12 Plotny disku a znázorněné cylindry

- **Přistávací zóna (landing zone):**

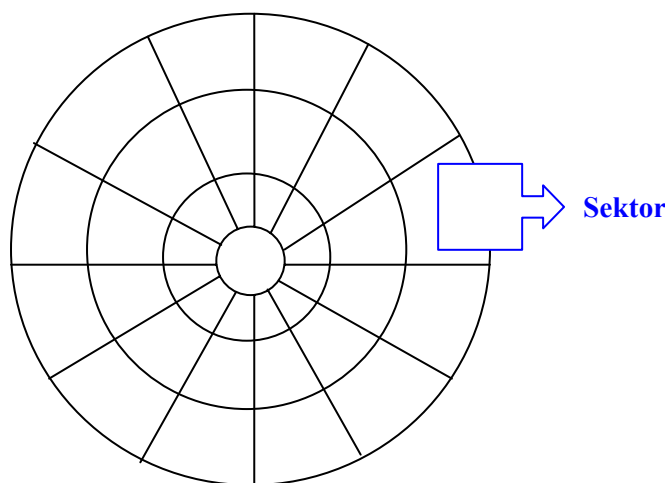
Číslo stopy (cylindru), která slouží jako přistávací zóna pro čtecí/zapisovací hlavy.

- **Sektory (sectors):**

Počet sektorů, na které je rozdělena každá stopa. U většiny starších pevných disků je podobně jako u pružných disků počet sektorů na všech stopách stejný. Tento způsob do jisté míry plýtvá médiem, protože vnější stopy jsou delší a tudíž by se na ně mohlo umístit více sektorů.

Stopy a sektory

Různé disky mají různý počet sektorů ve stopě. Diskety 8 až 36 sektorů ve stopě, disky 900 (i více) sektorů. Z důvodu kompatibility se staršími biosy a ovladači si disky provádějí interní logický překlad na 63 log. sektorů. Sektory se číslují od 1, zatímco hlavy a cylindry od 0. Úvodní částí sektoru je hlavička *Header* identifikující začátek sektoru a obsahující jeho číslo. Zakončení sektoru se nazývá *Trailer* a využívá se pro kontrolní součet. Sektor má tudíž ve skutečnosti 571 bytů, ale pro data pouze 512 bytů.



Obr 11.13 Zobrazení povrchu disku bez zonálního zápisu

Zápis (čtení) na (z) pevný disk probíhá na magnetickou vrstvu ve třech krocích:

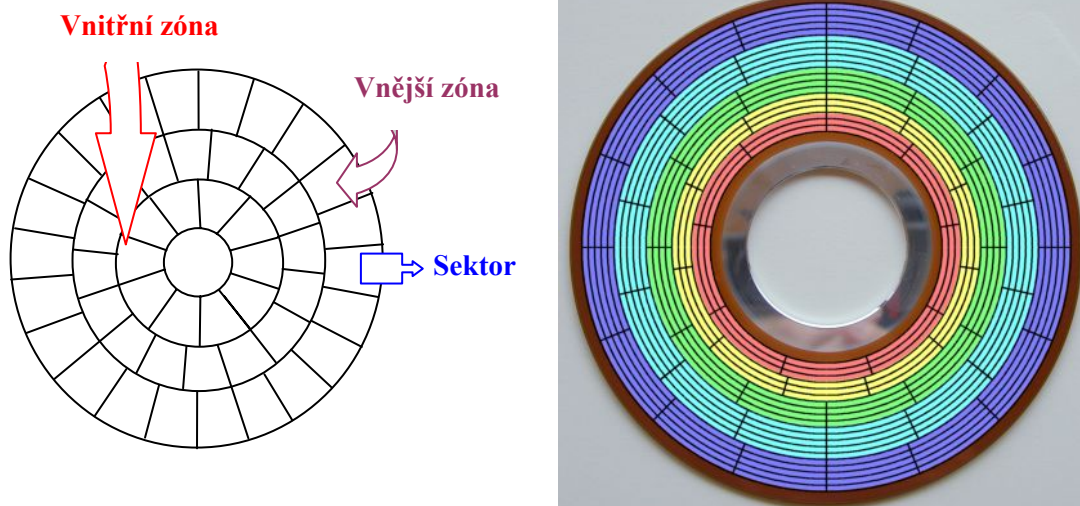
1. Vystavení zapisovacích (čtecích) hlav na příslušný cylindr - pomocí krokového motorku (dříve) nebo elektromagnetu (dnes).
2. Pootočení disků na patřičný sektor.
3. Zápis (načtení) dat.

Nejprve je zaplněn celý 1. cylindr, potom 2. cylindr a tak dále až po poslední cylindr. Tento způsob dovoluje, aby se čtecí (zapisovací) hlavy podílely na čtení (zápisu) paralelně.

Ukládání dat po jednotlivých discích by bylo podstatně pomalejší, protože v daném okamžiku by vždy mohla pracovat právě jedna hlava. V případě novějších pevných disků se používá tzv. zonální zápis označovaný jako ZBR (*Zone Bit Recording*). Jedná se o metodu zápisu na pevný disk, která dovoluje umístit na vnější stopy pevného disku větší počet sektorů než na stopy vnitřní. Celý disk je rozdělen do několika zón, z nichž každá má rozdílný počet sektorů na jednu stopu. ZBR tedy lépe využívá záznamové médium, ale způsobuje podstatně složitější přístup k datům. Sektory bývají číslovány od jedničky.

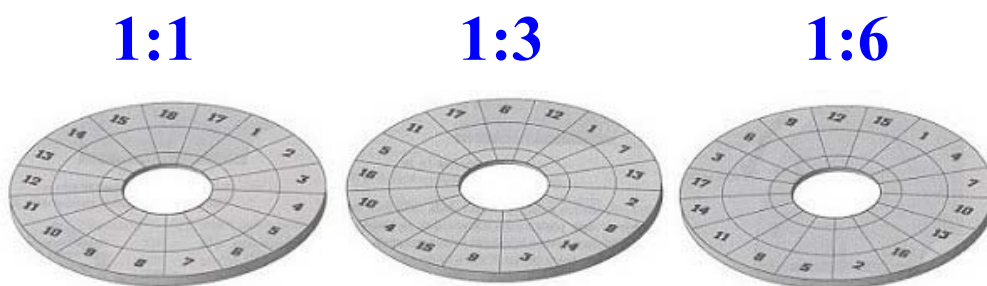
Při vypnutí počítače (a tím i pevného disku) se pevný disk přestává otáčet. Tím přestává existovat tenká vrstva, na které se pohybují čtecí (zapisovací) hlavy, a vzniká riziko jejich pádu na disky. Je nezbytné, aby čtecí (zapisovací) hlavy byly přemístěny do zóny, která je speciálně uzpůsobena k jejich přistání. U starších pevných disků bylo nutné vždy před vypnutím počítače provést pomocí nějakého programu tzv. **zaparkování diskových hlav**, tj. jejich přemístění na patřičné místo. Nové pevné disky

již využívají tzv. **autopark**, který je založen na tom, že po vypnutí pevného disku se pevný disk ještě chvíli setrvačností otáčí a tím vyrobí dostatek energie nutné pro přemístění hlav do parkovací zóny. Pro tuto parkovací zónu bývá většinou vyčleněna nejvnitřnější stopa disku, protože je na ní nejnižší rychlost.



Obr 11.14 Zonální zápis (ZBR recording)

Technika, při které nejsou data zapisována (a posléze čtena) do za sebou následujících sektorů, ale jsou během jedné otáčky disku zapisována vždy do každého n -tého sektoru, se nazývá **faktor prokládání** – *interleaving*. Faktor prokládání je označen poměrem **1:n**, kde číslo n je voleno tak, aby po přečtení a zpracování dat z jednoho sektoru byla čtecí (zapisovací) hlava nad dalším požadovaným sektorem. Rychlost otáčení pevného disku je poměrně vysoká, může se stát, že poté, co je přečten (zapsán) jeden sektor a data jsou předána dále, dojde k pootočení disků, takže čtecí (zapisovací) hlavy se nenacházejí nad následujícím sektorem. Nyní by tedy bylo nutné čekat další otáčku, než čtecí (zapisovací) hlavy budou nad požadovaným sektorem a pak by se situace znovu opakovala. V současnosti ovšem tento termín ztratil význam, neboť faktor prokládání současných disků již dosáhl hodnoty **1:1**.



Obr. 11.15 Faktor prokládání pevného disku

Formátování disku

Před prvním použitím disku je nutné disk naformátovat. Každý disk se formátuje dvěma způsoby, procesem nízkoúrovňového a vysokoúrovňového formátování. Tyto činnosti probíhají odděleně, navíc se mezi tyto operace může provést dělení disku na jednotlivé oddíly. Tato možnost dovoluje nainstalovat na disk více operačních systémů.



Pojem k zapamatování: Proces nízkourovňového formátování:

Při nízkourovňovém formátování jsou stopy na disku rozděleny na počet sektorů. Jsou vytvořeny hlavičky a zakončení každého sektoru. Každý sektor je pak vyplněn standardní hodnotou, např. F6H.

Všechny moderní disky používají zónový záznam.

Vysokourovňové formátování je v podstatě vytvoření tabulky s obsahem disku. Operační systémy MS Dos a Windows mohou pomocí příkazu *FORMAT* provést oba procesy formátování současně pouze při formátování disket. V případě formátování disku provádí tento příkaz pouze vysokourovňové formátování.

Rychlost přenosu dat závisí na zóně, ve které se hlavy nacházejí. Vnější zóna má samozřejmě více sektorů a protože rychlost otáčení je pro všechny zóny stejná, rychlost přenosu dat na vnější zóně je největší. Ustálená přenosová rychlost se pak rovná rozpětí mezi max. a min. hodnotou a určuje přenosovou rychlost média.



Pojem k zapamatování: Přenosová rychlost pevných disků:

Na základě parametrů HDD lze určit jeho maximální přenosovou rychlost:

$$v_{\max} = (\text{Psec} \cdot 512 \cdot \text{Pot}) / (60 \cdot 2^{20}) \text{ [MB/s]}$$

Psec značí počet sektorů na jednu stopu,
Pot značí počet otáček HDD za minutu,
512 je kapacita jednoho sektoru (v bytech).

Poznámka: u soudobých disků (s technikou ZBR) je nutné použít počet sektorů v krajní vnější zóně



Řešený příklad : Určení rychlosti disku

Pokud bude mít disk dvě plotny = 4 hlavy; 63 sektorů

a bude mít ve specifikaci 5400 ot/min:

$$v_{\max} = (4 \times 63 \times 512 \times 5400) / (60 \cdot 2^{20})$$

$$v_{\max} = 11,074 \text{ MB/s}$$

11.6 Struktura souborového systému

Souborové systémy jsou založeny na alokační tabulce souborů **FAT** (*File Allocation Table*), obsahující údaje o datech, uložených v každém sektoru na disku.

Jednotlivé varianty FAT:

- FAT12. Používá se na všech svazcích o kapacitě menší než 16 MB (např. na disketách),
- FAT16. Používá se na svazcích o velikosti 16 MB až 2 GB,
- FAT32. Lze využít na svazcích o kapacitě 512 MB až 2 TB.

Souborové systémy FAT16 byly součástí verzí operačních systémů MSDOS a Windows 3.11, FAT32 se stal součástí Windows 95, 98, Me a 2000. Windows XP využívají souborový systém NTFS (*NT File System*), OS/2 podporuje HPFS (*High Performance File System*).

Operační systém vytváří na disku několik základních struktur shodných pro všechny operační systémy podporující daný souborový systém. V případě souborového systému FAT se jedná o tyto struktury:

- Hlavní spouštěcí záznam a spouštěcí záznam rozšířeného oddílu.
- Spouštěcí záznam svazku.
- Kořenový adresář.
- Alokační tabulka souborů.
- Clustery (jednotlivé alokační jednotky v datové oblasti).
- Diagnostické cylindry pro čtení a zápis.

Na HDD nalezneme všechny uvedené struktury, zatímco na disketě chybí hlavní spouštěcí záznam a diagnostické cylindry. Hlavní spouštěcí záznam je pouze jeden, zatímco každý oddíl či svazek disku má svůj spouštěcí záznam svazku, svůj kořenový adresář, svou alokační tabulku a clustery. Všechny tyto struktury jsou vytvářeny programem FDISK.

MBR

Je vždy uložen v prvním sektoru celého disku (cylindr 0, hlava 0, sektor 1) a je tvořen těmito daty:

- Hlavní tabulkou rozdělení disku (Master Partition Table).

Obsahuje seznam všech oddílů vytvořených na daném disku a informace o spouštěcích záznamech jednotlivých oddílů.

- Hlavní spouštěcí kód. Malý program prováděný BIOSem systému, jeho hlavním účelem je předání řízení systému tomu oddílu, který je v tabulce označen jako aktivní (tj. ze kterého má být spouštěn operační systém).

VBR

Se nachází v prvním sektoru kterékoliv části disku, označené jako oddíl či svazek. Skládá se také ze dvou částí:

- Blok parametrů disku.

Obsahuje specifické parametry týkající se daného svazku: např. velikost, počet sektorů, velikost clusteru a název svazku.

- Spouštěcí kód svazku.

Je to program, který zahájí proces spouštění operačního systému. V případě systémů řady Windows9x/Me se jedná o soubor IO.SYS. Spouštěcí záznam svazku aktivního oddílu je načítán hlavním spouštěcím záznamem disku; v případě disket je načítán ROM BIOSem. Tímto okamžikem spouštěcí záznam svazku přebírá řízení systému. Provede se několik testů a následně se pokusí o načtení prvního souboru operačního systému (tedy io.sys u win9x/Me).

Root – kořenový adresář

Je poměrně jednoduchá databáze, obsahuje informace o souborech uložených v oddíle. Každý záznam této db má délku 32 bytů. V adresáři jsou uloženy prakticky všechny informace, které jsou operačnímu systému o daném souboru známy:

- Název souboru a jeho extenze.
- Byte s atributy souboru.
- Datum a čas poslední změny.
- Velikost souboru.
- Odkaz na počáteční cluster (číslo clusteru, ve kterém soubor začíná).

FAT – File Allocation Table

Tato tabulka popisuje přiřazení každého clusteru v oddílu, tj. datová část oddílu má pro každý cluster jeden záznam v této tabulce. Přitom základní jednotkou pro ukládání dat je sektor. Aby však FAT nemusela popisovat každý sektor oddílu, alokuje se prostor na disku po určitých skupinách sektorů nazývaných alokační jednotky či clustery.

Velikost clusteru (tj. počet sektorů tvořící cluster) je určen programem FDISK v průběhu vytváření oddílu, základním hlediskem je přitom velikost oddílu.



Pojem k zapamatování: Cluster

Cluster je nejmenším prostorem, který může být obsazen souborem; jinými slovy, bude-li soubor o jediný byte větší než cluster, zabere cluster dva.

Cluster je nejmenší alokační jednotkou diskového prostoru, kterou operační systém může využít pro zápis dat nějakého souboru. Cluster je vždy celým násobkem sektorů, mající velikost 512 Bytů.

Tabulka FAT16 je schopna pojmout až $2^{16} = 65536$ 16bitových záznamů - každý záznam ukazuje na určitý cluster. Velikost diskového oddílu FAT je historicky omezena 2^{16} clusterů. Velikost clusteru je od 0,5 kB do 32 kB. Diskový oddíl FAT16 může mít tedy maximální velikost nejvýše 2 GB. U velkých disků se toto omezení řešilo vytvořením více diskových oddílů.

Sektorů na cluster	Velikost Clusteru	Max. velikost diskového oddílu
1	512 B	32 MB
2	1 kB	64 MB
4	2 kB	128 MB
8	4 kB	256 MB
16	8 kB	512 MB
32	16 kB	1 GB
64	32 kB	2 GB

Tab. 11.4 Parametry tabulky FAT 16

Je dobré si uvědomit, že se stoupající velikostí disku roste i minimální velikost spotřebovaná k uložení i velmi malého souboru. Např. i soubor o velikosti 10 bajtů zabere na 2 GB disku 32 kB skutečného místa (toto neefektivně využití kapacity se říká *slack*).

Systém souborů FAT32 přináší kromě zvýšení maximální velikosti oddílu na 2 TB a neomezenému počtu kořenových záznamů ještě tu výhodu, že narozdíl od FAT16 používá dvě tabulky FAT (hlavní a záložní). Systém souborů je pak lépe zabezpečen proti nekonzistenci. Zvýšení počtu alokačních záznamů umožňuje používat velké disky v celku (neplatí limit 2GB na jednu *partition*), a to navíc při menší velikosti clusterů, což přispívá k lepšímu využití diskové kapacity.

Sektorů na cluster	Velikost Clusteru	Max. velikost diskového oddílu
1	512 B	32 – 256 MB
8	4 kB	256 MB – 8 GB
16	8 kB	8 - 16 GB
32	16 kB	16 – 32 GB
64	32 kB	32 GB – 2 TB

Tab. 11.5 Parametry tabulky FAT 32

Souborový systém NTFS

Byl navržen pro Windows NT, které také podporují nebo podporovaly HPFS (navržený pro OS/2) a FAT (navržený pro DOS), ale tyto souborové systémy mají omezené možnosti a nesplňovaly podmínky kladené potřebami na systém.

Těmito nedostatky jsou zejména omezená velikost disku a bezpečnost. Rovněž FAT32 podporovaný Windows 2000 tyto požadavky nesplňuje. NTFS byl navržen na konci 80 let při současně zahájeném vývoji NT a uveden v roce 1993 a lze říct, že byl navržen, aby splňoval nejen současné, ale i předpokládané požadavky.

Seznamy řízení libovolného přístupu DACL (*Discretionary access control lists*) a seznamy řízení systémového přístupu SACL (*system access control lists*) řídí, kdo vykoná akci se souborem a zajistí její zaznamenání.

FAT používá pro pojmenování souborů 8bitové ASCII kódování. V NTFS je použito 16bitové kódování Unicode, což umožňuje pojmenovat soubor v libovolném jazyce.

FAT nemá žádné opatření proti chybám na disku. Při pádu systému se systémové struktury mohou stát nekonzistentní, což může způsobit ztrátu dat. NTFS má naopak zabudováno transakční zaznamenávání akcí, proto se může při pádu systému pokusit obnovit data při jejich minimální ztrátě.

K uživatelským datům jsou přidružena tzv. metadata, ve kterých jsou informace o organizaci dat na disku. Jestliže inicializujete NTFS disk, NTFS vytvoří 12 souborů s metadaty. Tyto soubory nejsou normálně viditelné z nějakého souborového manažeru.

MFT Jméno	Záznam	Popis
\$MFT	0	Master File Table
\$MFTMIRR	1	Kopie prvních 16-ti záznamů \$MFT
\$LOGFILE	2	Transakční logovací soubor
\$VOLUME	3	Sériové číslo svazku, čas vytvoření
\$ATTRDEF	4	Definice atributů
.	5	Kořenový adresář disku
\$BITMAP	6	Mapa použití clusterů
\$BOOT	7	Boot record jednotky
\$BADCLUST	8	Seznam špatných clusterů na disku
\$QUOTA	9	Uživatelské kvóty
\$UPCASE	10	Přidělení velkých znaků k malým
\$EXTEND	11	Různá rozšíření: kvóty, data identifikátory objektů
-	12-15	Rezervováno pro využití v budoucnu

Tab 11.6 Metadata uložená v Master-File Table

Podobně jako v souborovém systému *FAT* je hlavní částí *file allocation table*, tak v *NTFS* je hlavní částí *MFT*, protože udržuje informace o rozložení všech souborů, adresářů i metadat na disku.

MFT je rozdělena na jednotky, které se nazývají záznamy. V jednom nebo více MFT záznamech NTFS ukládá *metadata*, která popisují vlastnosti souboru nebo adresáře (bezpečnostní nastavení, atributy) a jeho umístění na disku.

Protože *MFT* je také soubor, je i on zaznamenán v *MFT*. Uložení informací v těchto záznamech umožňuje, aby *MFT* mohla růst nebo zmenšovat se. Rychlost přístupu do *MFT* hraje rozhodující roli v celkovém výkonu *NTFS*, proto se *NTFS* snaží tento přístup maximálně zrychlit.



Pojem k zapamatování: limity souborového systému NTFS

Cluster má 64bitovou adresu, což umožňuje velikost diskového oddílu 16 EB (což odpovídá přibližně 17×10^9 TB). Soubor může obsahovat 255 znaků a počet souborů který je schopen souborový systém uložit je $4\,294\,967\,295 (2^{32}-1)$.

Protože MFT je soubor, který může růst a zmenšovat se, může být fragmentovaný (rozdělený do více částí). Tato fragmentace vzniká, protože NTFS nemůže přidělit souvislý prostor pro MFT, protože velikost MFT nelze dopředu určit.

Přístup k souboru je nejrychlejší, jestliže celý soubor je v jednom souvislém bloku clusterů, ale fragmentovaná MFT může způsobit opakované čtení záznamu, což snižuje výkon. Aby se předcházelo fragmentaci, NTFS vytváří oblasti clusterů okolo MFT, ve kterých zabráňuje ukládání jiných souborů a adresářů. Tato oblast (MFT-Zóna) snižuje pravděpodobnost fragmentace MFT. NTFS uvolní tuto oblast pouze v případě, jestliže je na disku málo volného prostoru.



Další zdroje

1. Mueller, S. *Osobní počítač – Hardware, upgrade, opravy*, Brno, Computer Press 2006, počet stran: 888. ISBN 80-7226-796-5.
2. Gook, M. *Hardwarová rozhraní*, Brno, Computer Press 2006, počet stran: 464. ISBN 80-251-1019-2.
3. Nygrýn P., Bábíček R. *444 tipů a triků pro nákup počítače a komponent*, Brno, Computer Press 2006, počet stran: 136. ISBN 80-251-0928-3.
4. Horák, J. *Bios a Setup*, Brno, Computer Press 2004, počet stran: 132. ISBN 80-251-0148-7.
5. Aspinwall, J. *IRQ, DMA a I/O*, Brno, Computer Press 2000, počet stran: 288. ISBN 80-722-6264-5.
6. Vlach, J, Vlachová V. *Počítačová rozhraní*, Praha, Ben 2002, počet stran: 176. ISBN 80-7300-010-5.
7. Serial ATA International organization server [online] dostupný z <http://www.sata-io.org/>
8. Overview and History of the IDE/ATA Interface [online] dostupný z <http://www.pcguides.com/ref/hdd/if/ide/over-c.html>



Shrnutí kapitoly

V roce 1986 bylo navrženo firmami Western Digital a Compaq rozhraní IDE (*Integrated Device Electronics*). Zkratka IDE je poměrně obecná a označuje kteroukoliv mechaniku s vestavěným řadičem, rozhraní IDE se dnes častěji označuje jako ATA (*AT Attachment*). Verzi rozhraní ATA bylo během vývoje standardizováno celkem 7, od původního rozhraní, kdy bylo možné připojit pouze 512 MB disk až po rozhraní ATA7, po roce 2000 s přenosovou rychlostí 133 MB/s. Paralelní režim již dále za stejných podmínek nelze donekonečna zrychlovat a tak se přechází na sériové rozhraní označované SATA. Pro rozhraní SATA máme definované v současnosti tyto standardní přenosové rychlosti: SATA I: 150 MB/s, SATA II: 300 MB/s, SATA III: 600 MB/s.

Pevný disk obsahuje plotny diskového tvaru, na rozdíl od disket se tyto plotny nemohou

ohýbat – proto termín pevný disk. Základem disku je tedy několik ploten (umístěných nad sebou) což jsou otáčející se disky. Každá plotna má dvě strany, na které se ukládají data. Plotny se otáčejí stejnými otáčkami. Tyto disky se otáčejí po celou dobu, kdy je pevný disk připojen ke zdroji elektrického napájení, díky tomuto otáčení se v okolí disků vytváří tenká vzduchová vrstva. Nad těmito disky se pohybují hlavy a ukládají a čtou data do soustředných prstenců nazývaných stopy (tracks), stopy jsou rozděleny na segmenty označené jako sektory, z nichž každý pojme 512 bytů dat. Všechny hlavy jsou připevněny k jednomu závěsu hlav. Vzdálenost hlav od disku je asi 0,3 až 0,6 mikronu.

Přístupová doba je doba, která je nutná k vystavení čtecích/zapisovacích hlav na požadovaný cylindr, je v rozmezí 3,6 - 65 ms. Obvyklá rychlost otáčení je dnes 7200 ot/min nebo dokonce 10000 ot/min, dříve 5400 ot/min nebo 4500 ot/min. Je možné se setkat ale i s 12000 a 15000 ot/min. Přenosová rychlost (*Data transfer rate*) určuje počet bytů, které je možné z disku přenést za jednu sekundu, jedná se o rychlosti 700 kB/s až řádově desítky MB/s. Cache paměť pevného disku je realizována jako paměť typu DRAM a její velikost je až 16 MB.

Při nízkourovňovém formátování jsou stopy na disku rozděleny na počet sektorů. Jsou vytvořeny hlavičky a zakončení každého sektoru. Každý sektor je pak vyplněn standardní hodnotou, např. F6H.

Všechny moderní disky používají zónový záznam. Rychlost přenosu dat závisí na zóně ve které se hlavy nacházejí. Vnější zóna má samozřejmě více sektorů a protože rychlost otáčení je pro všechny zóny stejná, rychlost přenosu dat na vnější zóně je největší. Ustálená přenosová rychlost se pak rovná rozpětí mezi max. a min. hodnotou a určuje přenosovou rychlost média.

Souborové systémy jsou založeny na alokační tabulce souborů FAT (*File Allocation Table*), obsahující údaje o datech uložených v každém sektoru na disku. Operační systém vytváří na disku několik základních struktur shodných pro všechny operační systémy podporující daný souborový systém. V případě souborového systému FAT se jedná o tyto struktury: hlavní spouštěcí záznam a spouštěcí záznam rozšířeného oddílu, spouštěcí záznam svazku, kořenový adresář, alokační tabulka souborů, cluster (jednotlivé alokační jednotky v datové oblasti) a diagnostické cylindry pro čtení a zápis.

Cluster je nejmenším prostorem, který může být obsazen souborem; jinými slovy, bude-li soubor o jediný byte větší než cluster, zabere cluster dva.

Cluster je nejmenší alokační jednotkou diskového prostoru, kterou operační systém může využít pro zápis dat nějakého souboru. Cluster je vždy celým násobkem sektorů, mající velikost 512 Bytů.

Podobně jako v souborovém systému *FAT* je hlavní částí *file allocation table*, tak v *NTFS* je hlavní částí *MFT*, protože udržuje informace o rozložení všech souborů, adresářů i metadat na disku.

MFT je rozdělena na jednotky, které se nazývají záznamy. V jednom nebo více *MFT* záznamech *NTFS* ukládá *metadata*, která popisují vlastnosti souboru nebo adresáře (bezpečnostní nastavení, atributy) a jeho umístění na disku. Cluster má 64 bitovou adresu, což umožňuje velikost diskového oddílu 16 EB (což odpovídá přibližně 17×10^9 TB). Soubor může obsahovat 255 znaků a počet souborů, který je schopen souborový systém uložit, je $4\,294\,967\,295$ ($2^{32}-1$).



Úkol k řešení 11.1

1. Určete přenosovou rychlost pevného disku o rychlosti 7200 ot/min, který má 4 hlavy (63 sektorů).

**Úkol k řešení 11.2**

Vytvořte tabulku srovnání propustnosti jednotlivých verzí rozhraní ATA, SATA, SCSI. Porovnejte s rychlostí dalších vybraných počítačových sběrnic.

**Kontrolní otázka 11.1**

Definujte vývoj rozhraní pevných disků.

**Kontrolní otázka 11.2**

Vysvětlete pojmy PIO režim a DMA.

**Kontrolní otázka 11.3**

Jak se vyvíjelo rozhraní ATA?

**Kontrolní otázka 11.4**

Vysvětlete, s jakými omezeními velikosti disku musel vývoj počítat.

**Kontrolní otázka 11.5**

Definujte součásti pevných disků.

**Kontrolní otázka 11.6**

Co je to geometrie pevného disku?

**Kontrolní otázka 11.7**

Definujte pojmy stopy a sektory.

**Kontrolní otázka 11.8**

Popište souborové systémy pevných disků.

12. Grafický subsystém počítače – grafická karta a monitor



Čas ke studiu: 3 hodiny



Cíl Po prostudování této kapitoly budete umět

- popsat vývoj standardu grafických adaptérů
- popsat podrobněji součásti grafického adaptéru
- definovat vývoj paměťových čipů grafické karty a vypočítat potřebnou paměť pro různá rozlišení a množství barev
- Popsat princip zobrazení na CRT monitoru
- Charakterizovat LCD monitor, princip zobrazení a typy výrobních technologií



Výklad

Grafická karta je jednou z komponent, jejíž vývoj dosáhl největších změn. V rozboru jednotlivých komponent jsme si popsali procesory, čipové sady, paměti. Ale vždyť tohle všechno grafický akcelerátor obsahuje také! Grafický procesor je součástí čipové sady, avšak paměťové čipy jsou téměř totožné jako u systémové paměti, tedy co se týká výrobní technologie. Z celkového uspořádání grafického adaptéru je patrné, jak složité a matematicky náročné jsou v dnešní době grafické aplikace. Přesto výrobci čipových sad základních desek nezahálají a dokáží nabídnout poměrně kvalitní integrované grafické čipy, takže mnoho z nás vlastně grafickou kartu ani nepotřebuje! Přesto je důležité si grafickou kartu podrobněji představit. Druhá část kapitoly bude věnována rozboru výstupního zařízení pro zobrazení. Ještě v nedávné minulosti jsme měli možnost připojit k PC pouze klasický CRT monitor, před několika lety se poměr LCD a CRT začal vyrovnávat a je otázkou několika dalších let, kdy CRT bude pouhým nepatrným zlomkem na trhu s monitory.

12.1 Standardy grafických adaptérů

Grafická karta je adaptér, jenž tvoří rozhraní mezi počítačem a monitorem, odesílá do monitoru signály, které vytvoří na obrazovce požadovaný obraz. Grafický adaptér podobně jako ostatní hardwarové komponenty také prošel svým vývojem standardů. Pomineme-li grafické adaptéry pro monochromatické zobrazení, začneme tak od standardu *CGA (Color Graphics Adapter)* navrženého firmou IBM v roce 1981. Tento standard umožnil maximální rozlišení 640 x 200 pixelů při čtyřbitové barevné hloubce, což znamená 16 barev. Firma IBM v podstatě od počátku hrála hlavní roli při tvorbě standardů a tak se vyvinul v roce 1984 standard *EGA (Enhanced graphics adapter)*, jenž je vlastně vylepšený CGA standard pro rozlišení 640 x 350 pixelů při 16 barvách. Přímou pro své počítače řady PS/2 vyvinula IBM další tři standardy v roce 1987: *VGA (Video Graphics Array)*, *MCGA (MultiColor Graphics Array)* a *851*, přičemž pouze standard VGA se uchytil a zbývající dva zanikly. Standard VGA zvýšil maximální počet horizontálních pixelů až na 720 a vertikálních na 480, což zvýšilo možné rozlišení na 640 x 480 při 16ti barvách. Při nižším rozlišení umožňoval současně zobrazit až 256 barev z palety možností 262144 barev. Obnovovací frekvence standardu VGA dosahovala až 70 Hz.

Po uvedení standardu XGA (*eXtended Graphics Array*), které je charakterizováno rozlišením 800 x 600 při 16bitové barevné hloubce resp. 1024 x 768 při 8 bitové barevné hloubce firmou IBM v roce

1990, se ostatní výrobci vzepřeli a začali pracovat na vývoji levnějších variant grafických adaptérů při vyšším rozlišení. Tyto adaptéry měly jednoduché označení *SVGA* (*Super VGA*). Vývoj aplikací při absenci standardu byl velice obtížný, z čehož vyplynula nutnost standardizace SVGA. Standard SVGA podporuje celou řadu neustále se zvyšujících rozlišení při 24bitové barevné hloubce. Pojmu barevná hloubka se budeme ještě věnovat v podkapitole Paměť grafické karty.

12.2 Grafický akceleračtor

Pojem grafický akceleračtor již můžeme v dnešní době ztotožnit s pojmem grafická karta, neboť dnešní na trhu běžné grafické karty již splňují požadavky na zpracování určitých náročných operací, 3D scén, podporu funkcí DirectX či OpenGL. Náročné grafické operace zpracovává procesor grafické karty, který je propojen s videopamětí pomocí dnes již 64, 128 či 256 bitové sběrnice.



Obr. 12.1 Grafický akceleračtor s aktivním chlazením se sběrnici AGP 8x (vlevo), s pasivním chlazením se sběrnici PCI Express (vpravo)

Na obrázku 12.1 vidíme představitele dnešní standardní grafické karty. Grafická karta se mimo jiné skládá z těchto hlavních součástí:

Video ROM BIOS: Obdobně jako je tomu u biosu systému na základní desce, tak i grafický adaptér má svůj vlastní BIOS nezbytný pro funkčnost grafické karty a mimo jiné obsahuje instrukce pro rozhraní mezi grafickým adaptérem a systémem. Je uložen v paměťovém čipu typu ROM, obdobně jako u základní desky může být verze ROM paměti Flash-ROM či EEPROM a BIOS tak může být aktualizován.

Grafický procesor: Řídí činnosti grafické karty, ovládá všechny elementy spojené s vykreslováním pixelů na obrazovku. Je součástí čipové sady grafického adaptéru a podobně jako je tomu u čipových sad základních desek i u grafických karet platí, že grafické adaptéry mají obdobný výkon, ovladače a podobné vlastnosti, mají-li stejnou čipovou sadu. Výrobci grafických adaptérů je mnoho, ale výrobu grafických čipových sad ovládli výrobci ATI a N-VIDIA se svými čipy Radeon resp. GeForce.

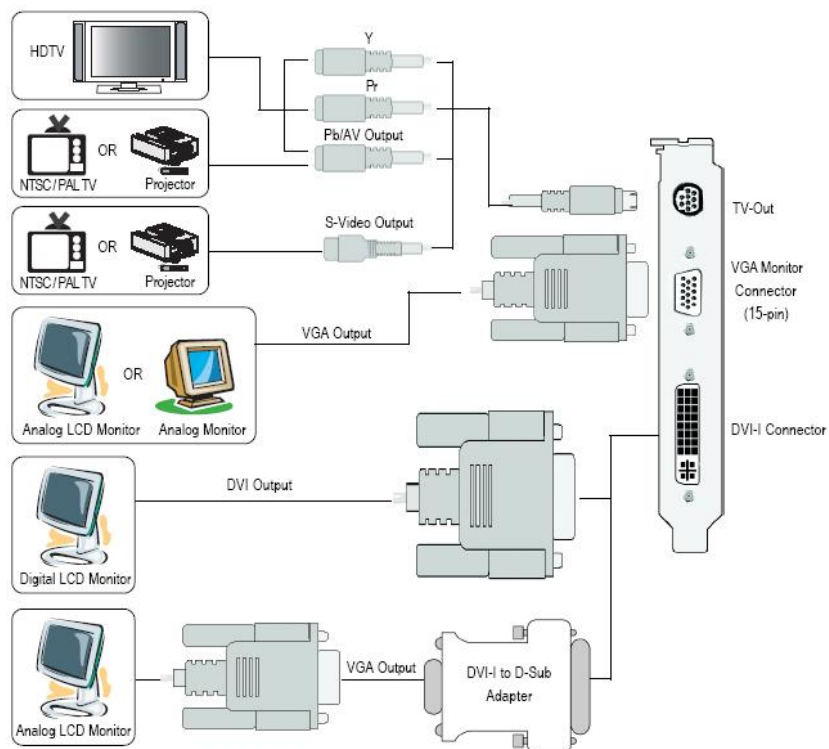
Paměť grafického adaptéru: Uchovává informace, ze kterých grafický procesor vytváří obraz. Paměti grafického adaptéru je věnována následující kapitola.

Konvertor digitálního signálu RAMDAC: Jedná se o obvod integrovaný přímo do čipu grafického adaptéru, který převádí digitální obraz vypočítaný grafickým procesorem na analogový signál monitoru. Rychlost konvertorů se udává v MHz a z části na něm závisí obnovovací frekvence monitoru.

Rozhraní podporované sběrnice: Požadavkem je, aby grafická karta byla připojena vždy k nějaké systémové sběrnici. V minulosti to byla sběrnice ISA, později sběrnice PCI a v nedávné minulosti AGP (*Accelerated Graphics Port*). Ale i sběrnice AGP má svá technologická omezení, a proto

sběrnice AGP 8x je posledním, i když stále poměrně hojně využívaným rozhraním grafické karty. Díky technologickému omezení tohoto rozhraní (není možné ho dále rozvíjet - zrychlovat) bylo nutné počítat s rozhraním novým, které bylo představeno na počátku roku 2005, kdy se začaly vyrábět grafické adaptéry na sběrnici PCI Express a tímto směrem se bude ubírat i jejich další vývoj. Více je k systémovým sběrnicím uvedeno v kapitole 8.

Výstupní konektor: Zajišťuje výstup výsledného zobrazení na monitor či jiné zařízení. Následující obrázek popisuje dnes již standardní výstup grafického adaptéru, jež je vybaven standardním 15ti pinovým konektorem pro výstup na analogový monitor CRT nebo LCD, dále pak DVI-I konektorem (Digital Video Interface) s 20ti vývody pro výstup na digitální monitor LCD a také S – video konektor pro výstup na TV. Na obrázku 12.2 vidíme možnosti propojení výstupu grafické karty.



Obr. 12.2 Možnosti připojení na výstup grafické karty

12.3 Paměť grafické karty

Současné grafické karty jsou vybaveny svou vlastní pamětí pro ukládání jednotlivých obrazů v době samotného zpracování grafickým procesorem. V současné době jsou paměťové čipy grafických karet typu DDR SDRAM. V minulosti byly grafické karty vybavovány čipy obdobné technologie jako paměťové moduly RAM. Jednalo se o FPM RAM (*Fast Page Mode RAM*) u dnes již zastaralé sběrnice ISA, EDO RAM (*Extended Data Out RAM*) pro sběrnice PCI a SDRAM obdobné paměťovým modulům DIMM pro počátky rozhraní AGP. Při vývoji technologie pamětí grafických karet však byly pokusy i o speciální paměťové moduly, které se od systémové paměti lišily. Jednalo se o paměti typů VRAM (Video RAM) a WRAM (*Window RAM*), které by se daly charakterizovat jako dual port paměti s možností současného čtení a zápisu. Tyto paměti vystřídala již zmíněná technologie SDRAM současně s technologií SGRAM (*Synchronous Graphics RAM*), jež byla vybavena velice rychlou paměťovou sběrnicí, jež se synchronizovala s rychlostí sběrnice základní desky, obdobně jako SDRAM. Čipy SGRAM byly navrženy oproti SDRAM s možností zápisu velkých bloků dat najednou, což využívaly náročnější grafické operace a 3D instrukce využívajících tzv. Z zásobníku. Tyto čipy byly výkonné, ale jejich negativní stránkou byla vysoká cena. Možnosti výroby čipů za nižší ceny, jak už tomu v oblasti hardwaru bývá, často rozhodnou o nepokračování životnosti konkrétní

technologie, a tak se čipy typu SGRAM přestaly vyrábět. Rozhodl o tom přechod na současné paměťové čipy typu DDR SDRAM, které zdvojnásobují přenosovou rychlost přenosem dat na počáteční i koncové hraně pulsu. Nyní se již bude jednat o obdobné zvyšování frekvencí, jako tomu je u paměťových modulů systémové paměti (DDR2, DDR3).

Velikost paměti určuje maximální možné rozlišení a počty barev, jež můžeme na monitoru zobrazit. Pro určitá rozlišení (i vysoká) nám postačí i méně paměti, než je v běžných adaptérech k dispozici. Z toho důvodu nemusí zvětšení paměti vést nutně ke zvětšení výkonu grafického adaptéru, potažmo celého systému. Běžné kancelářské a uživatelské aplikace nejsou samozřejmě tak náročné jako grafické 3D aplikace, CAD nástroje a počítačové hry. Potřebnou velikost paměti pro zobrazení určitého rozlišení s příslušným počtem barev určíme podle následujícího vztahu:



Pojem k zapamatování: Výpočet potřebné kapacity paměti

$$\text{Kapacita potřebné paměti} = H_p \times V_p \times B_p$$

Kde H_p určuje počet pixelů v horizontálním směru, V_p určuje počet pixelů ve vertikálním směru a B_p určuje počet bytů pro zobrazení v příslušné barevné hloubce.



Řešený příklad : Výpočet potřebné kapacity paměti

Pro rozlišení 1280 x 1024 v režimu True Color

Vynásobením pixelů dostáváme 1 310 720 obrazových bodů.

Režim True Color představuje 16,7 miliónů barev,

což je mocnina 2^{24} , barevná hloubka je 24 bitů,

takže potřebujeme 3 byty pro zobrazení každého pixelu.

Z toho plyne $1\,310\,720 \times 3 = 3\,932\,160$ bytů.

Výsledkem je tedy potřeba **3,75 MB** paměti.

Z výše uvedeného příkladu je zřejmé, že na práci v rozlišení 1280x1024 v režimu True Color potřebujeme 3,75 MB paměti. V tomto režimu jsme mohli tedy v nedávné minulosti pracovat až po uvedení grafického čipu o velikosti 4 MB. Pro práci v 3D grafických aplikacích musíme samozřejmě počítat s větší potřebou paměti a to na hloubce Z – bufferu. Při 24bitové hloubce Z – bufferu bychom tedy při uvedeném rozlišení z příkladu potřebovali 11,25 MB paměti pro práci ve 3D. Dnešní grafické adaptéry obsahují běžně 128 nebo 256 MB a některé dokonce 512 MB.

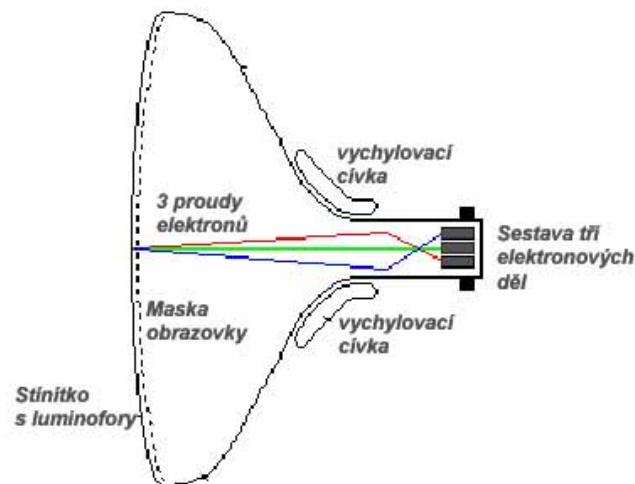


Korespondenční úkol – výpočet rozlišení

Jakým výpočtem jsme se v předchozím příkladu dostali od čísla 3 932 160 k výsledku 3,75?

12.4 Charakteristika CRT monitoru

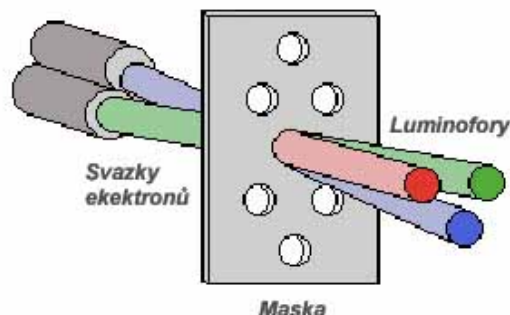
Monitor CRT (*Cathode Ray Tube*) je vyroben na základě stejné technologie jako televizor a před nástupem LCD panelu byl v podstatě zobrazovacím standardem u stolních počítačů. CRT obsahuje elektronku, která je vzduchoprázdná. Na konci elektronky se nachází soustava tří elektronových děl a na opačném rozšířeném konci obrazovka pokryta vrstvou luminoforů, které přeměňují kinetickou energii na energii světelnou. Luminofory jsou ve třech základních barvách RGB, zatímco proudy elektronů jsou samozřejmě bezbarvé a až po dopadu na příslušné luminofory dojde k rozsvícení bodu odpovídající barvy.



Obr. 12.3 Obrazovka CRT monitoru

12.5 Princip zobrazení na CRT monitoru

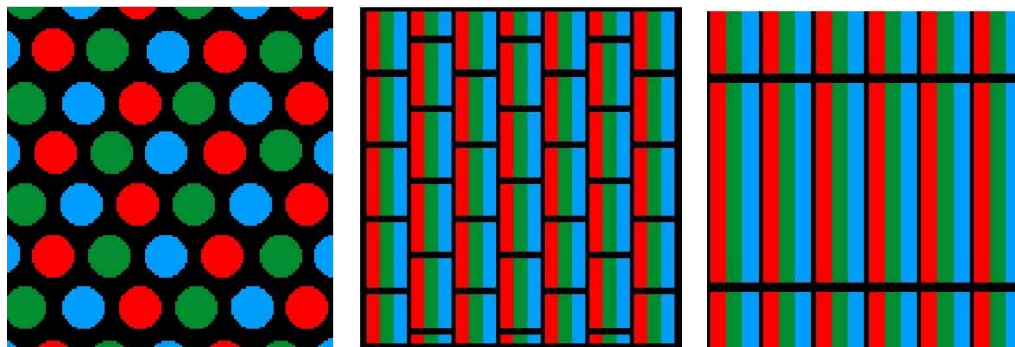
Elektronové dělo emituje po zahřátí proud elektronů o vysoké rychlosti. Elektronové svazky jsou vychylovány pomocí vychylovacích cívek tak, aby opisovaly shora dolů a zleva doprava jednotlivé řádky obrazovky. Vychylovací a zaostřovací cívky se nacházejí podél elektronky. Na vnitřní straně obrazovky je nanesená stínící maska děrovaná nebo štěrbínová (podle typu obrazovky), která slouží k seskupení RGB spektra do jednoho bodu viz. obr. 12.4. Maska je v podstatě mříž, která propustí pouze úzký svazek elektronů.



Obr. 12.4 Průchod elektronových svazků maskou obrazovky

12.6 Typy obrazovek a parametry monitorů

Podle tvarů otvorů masky a nanesení luminoforů se dělí obrazovky na tři základní typy: *Delta*, *Inline* a *Trinitron* – propagována zejména firmou Sony. U obrazovek typu Delta jsou otvory v masce a taktéž luminofory na stínítku kruhové a uspořádány do tvaru velkého písmene delta. Jejich nevýhodou je horší kvalita obrazu a vypouklá obrazovka. Druhý typ obrazovky Inline má otvory v masce obdélníkového tvaru a jednotlivé luminofory nanesené v řadě vedle sebe. Masky obrazovky Trinitron je tvořena svislými pásy, které je ovšem nutno přichytit stabilizačními dráty, natažené napříč mřížkou, které ji mají za úkol vyztužit. Obrazovky Trinitron mají jasnější obraz, i když při podrobném prohlížení obrazu lze spatřit stabilizační dráty.



Obr. 12.5 Typy obrazovek (zleva): Delta, Inline a Trinitron

V průběhu pohybu elektronového paprsku dopadá proud elektronů na luminofory. Body se velmi rychle rozsvítí ale také velice rychle pohasínají. Z tohoto důvodu musí být obraz neustále obnovován, navíc pohasínání a rozsvěcování luminoforů způsobuje nepříjemné blikání obrazu. Obnovovací frekvence je tedy nejdůležitějším parametrem monitoru. Rozlišujeme dva typy obnovovací frekvence:



Pojem k zapamatování: Obnovovací frekvence CRT monitoru

Horizontální frekvence – řádkový kmitočet, měří se v kHz a udává rychlost s jakou paprsek přejíždí z levé strany obrazovky na pravou.

Vertikální frekvence – obnovovací kmitočet obrazu, měří se v Hz a udává počet obrazů překreslených za jednu sekundu.

Obecně platí, že čím vyšší tyto frekvence jsou, tím kvalitnějším a stabilnějším obrazem obrazovka disponuje.

Dalšími parametry, podle kterého jsou CRT monitory vybírány, jsou maximální a doporučené rozlišení, šířka pásma a samozřejmě velikost úhlopříčky. Dnešní nejčastěji používané CRT monitory mají úhlopříčku 17 či 19 palců.



Korespondenční úkol – Poruchy CRT obrazu

Vinou nepřesnosti vychylovacích cívek, popřípadě další závadou monitoru je zapříčiněn fakt, že elektronové svazky nevypisují přesný obdélník obrazu, ale nějaký jiný obrazec, který vznikl zkreslením tohoto obdélníku. Vyhledejte na internetu popřípadě v jiných zdrojích možné geometrické poruchy obrazu.

12.7 Charakteristika LCD monitoru

LCD (*Liquid crystal display*) monitory jsou zobrazovací jednotky, které při své činnosti využívají technologii kapalných (tekutých) krystalů. Z hlediska podstaty struktury si dokážeme představit skupenství pevné látky, jehož molekuly nemění svou pozici a skupenství kapalně, jehož molekuly se volně pohybují. Skupenství tekutého krystalu si tedy představme s oběma vlastnostmi: pevné látky, kde osy molekul jsou orientovány stejným směrem i kapaliny, jejichž molekuly se mohou volně pohybovat. Molekuly krystalů jsou organické látky, které mají schopnost se shlukovat do geometrických obrazců různých vrstev a tvarů, svislé i válcovité sloupce. Odborně se tyto varianty tekutých krystalů dělí na nematické, cholesterické a smectické. Pro konstrukci LCD monitorů se používají tekuté krystaly nematické, jejichž molekuly mají podlouhlý tyčovitý tvar. Z praktického hlediska je důležité, že pravidelná struktura umožňuje s molekulami tekutých krystalů různě otáčet působením elektromagnetického pole, měnit přísun světla a pomocí polarizačních a barevných filtrů vytvářen výsledné barevné odstíny na monitoru. Než si popíšeme samotný princip a současné technologie zobrazení, vyjmenujme si základní parametry LCD monitorů, které víceméně hrají roli při výběru LCD panelu. Mnoho uživatelů se řídí při výběru převážně cenou, nyní si uvedme od čeho se bude odvíjet:

Úhlopříčka: Viditelná plocha je velkou výhodou LCD oproti CRT monitorům. Můžeme proto srovnávat zhruba 17ti palcové LCD s 19ti palcovou obrazovkou CRT. S klesající cenou se právě 17ti a 19ti palcové LCD stávají v současnosti (*) standardem.

Doba odezvy: Je měřena v milisekundách a udává dobu potřebnou pro změnu barvy pixelu z černé na bílou a zpět na černou. Doba odezvy se neustále zrychluje, avšak spotřebitel může být klamán výrobcem pomocí marketingového triku. A to z toho důvodu, že přechod z černé barvy na bílou se v reálu příliš nevyskytuje, proto výrobci uvádějí přechod z jednoho odstínu šedé do druhého. A protože různé barevné přechody trvají různou dobu, výrobce obvykle uvede ve specifikaci tu nejkratší. V současnosti (*) jsou LCD monitory uváděny na trh s dobou odezvy 2 – 16 ms.

Kontrast: Jedná se o svítivost monitoru při zobrazení pouze černé a pouze bílé barvy. Tyto hodnoty jsou měřeny v luxech a dány do poměru. Současné LCD panely mají kontrastní poměr od 700:1 až po 2000:1. Kontrastní poměr je jedna z nevýhod technologie LCD oproti plasmovým obrazovkám. V současnosti (*) ale firma Samsung dokázala vyrobit LCD obrazovky s kontrastním poměrem až 5000:1, což se už dostává na poloviční hodnotu plasmové obrazovky 10000:1.

Jas: Jas monitoru se udává v jednotkách cd/m^2 . Měří se svítivost monitoru při zobrazení bílé barvy. Dnešní LCD panely disponují touto hodnotou okolo 250 cd/m^2 .

Doporučené rozlišení: U LCD panelu nemá příliš cenu měnit rozlišení monitoru, neboť to je dáno výrobcem podle počtu bodů a jim příslušných řídicích tranzistorů. Typická rozlišení jsou 1280 x 1024 pro 17ti a 19ti palcové LCD, 1440 x 900 pro širokoúhlé 19ti palcové LCD, 1600 x 1200 pro 20ti palcové LCD, 1680 x 1050 pro 20ti palcové širokoúhlé LCD a mnoho dalších.

Pozorovací úhel: Je určen pro horizontální a vertikální směr, např.: 176°/176° a udává maximální pozorovací úhel, který bychom neměli překročit, pokud chceme dobře vidět obraz na monitoru. Po překročení těchto úhlů je mnohdy vyloučené sledovat jakýkoliv obraz.

Spotřeba: Dnešní spotřeba 19ti palcového LCD panelu se pohybuje mezi 35 – 50 Watty. Spotřeba LCD je často kladně hodnocena při porovnání s plasmovými obrazovkami na trhu s TV. Zde je ovšem nutné uvést fakt, že plasmové TV jsou vyráběny s úhlopříčkami nad 100 cm a jejich spotřeba je uváděna okolo 270 W. Ale pokud bychom vzali v úvahu LCD s obdobnou úhlopříčkou, budeme opět u spotřeby vysoko nad 200 W

Vybavení monitoru: Ve prospěch určitého typu při výběru LCD panelu může v některých případech hrát roli další vybavení jako USB Hub, vestavěné reproduktory, sluchátkový výstup či další.

Výrobní technologie: Mnoho uživatelů nad tímto pojmem vůbec neuvažuje a bere LCD jako jednu výrobní technologii. Avšak samotný princip výrobní technologie přímo předurčuje výše uvedené vlastnosti jako doba odezvy, pozorovací úhel či kvalitu barev. Ale v případě, že se zajímá alespoň

o tyto údaje, nemusí být pro něj výrobní technologie podstatná. V následující kapitole si je v krátkosti představíme. Jedná se o výrobní technologie *TN*, *IPS*, *NVA* a jejich další verze.

Pozn: (*): počátek roku 2007

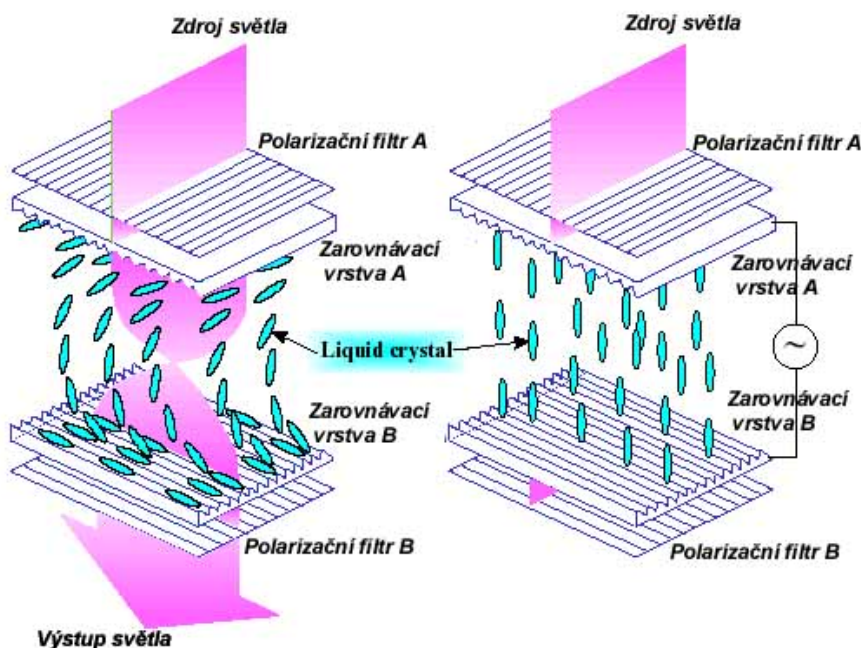
12.8 Princip zobrazení na LCD monitoru

Pro získání obrazového výstupu potřebujeme zajistit světelný zdroj. Ten je zajištěn světelnými katodami a odraženým světlem. Světlo je rozkládáno do barevných spekter tří složek – RGB. Body obrazu jsou ovládány příslušnými tranzistory a jsou ohraničeny dvěma polarizačními filtry, barevným filtrem a vyrovnávacími vrstvami. Tranzistory kontrolují příslušné napětí mezi vyrovnávacími vrstvami a elektrické pole působí na změnu natočení tekutých krystalů, čímž vzniká výsledný barevný odstín.



Pojem k zapamatování: Počet tranzistorů příslušného LCD panelu

Každý obrazový bod, resp. Pixel je složen ze tří subpixelů, jeden pro každou složku RGB spektra. Nová výrobní technologie S-IPS potřebuje dokonce dva tranzistory na subpixel. Každý subpixel je ovládán jedním tranzistorem. Z toho plyne, že při rozlišení 1600 x 1200 obsahuje LCD panel 5 760 000, resp. 11 520 000 tranzistorů.



Obr. 12.6 Princip LCD zobrazení tekutými krystaly

Obecně LCD display s aktivní maticí je označován zkratkou TFT (*Thin Film Transistor*), což vyjadřuje zapouzdření tranzistorů do tenké pružné fólie. V souvislosti s výrobním postupem se setkáme se zkratkami A-Si, resp. P-Si. Jedná se o výrobní postup, jakým silikonem je obrazovka tvořena. A-Si znamená amorfni silikon, jímž se vyráběly LCD panely zpočátku, pro svoji dostupnější cenu a možnost výroby při nižších teplotách. Postupně byl vynalezen další nízkoteplotní postup P-Si a je také využíván.

Výrobních technologií, jak už bylo řečeno výše, je několik rozdělených do tří kategorií. První kategorií a zároveň původní technologií je TN (*Twisted Nematic*). Ta se vyznačovala obdélníkovým tvarem pixelu, který zapříčinil různé pozorovací úhly ve vertikálním a horizontálním směru. Molekuly

tekutých krystalů technologie TN se mění ze šroubovitého uspořádání do přímého. Nová technologie označovaná B-TN vychází z původní TN, pouze pixel již je čtvercový, a tak jsou pozorovací úhly shodné v obou směrech, přesto jsou stále nízké, a tak tato technologie je prosazována v levnějším segmentu trhu. Druhou kategorií tvoří technologie MVA (*Multi-Domain Vertical Aligment*) vyvinutá firmou Fujitsu Siemens a PVA (*Patterned Vertical Aligment*) vyvinutá firmou Samsung. Pixely jsou čtvercové a symetrické, takže pozorovací úhly jsou naprosto stejné. Molekuly již nejsou uspořádány do šroubovice jako u technologie TN, nýbrž do stromovitého uspořádání, které při natáčení celého uspořádání disponuje s rychlejší odezvou. Další výhodou je bezpochyby větší kontrast, neboť při průchodu světla polarizačními filtry dochází k menším ztrátám. Barevné podání se liší u jednotlivých verzí této technologie (technologie je zdokonalována - *Prem.MVA*, *S-MVA*, *S-PVA*, *A-MVA*), protože původní technologie disponují šesti bity na barvu, některé odvozené technologie osmi bity. Z toho důvodu se barvy mohou jevit u starších verzí jako u technologie TN. Třetí a poslední kategorií je technologie IPS (*In-Plane Switching*), která byla pro svou velice dlouhou odezvu zdokonalena S-IPS a AS-IPS (*Advanced Super*). Tato kategorie poskytuje nejlepší barevné podání ze všech technologií vůbec. Avšak tato technologie má i své nevýhody, potřebuje silné podsvícení, neboť některé nejdál od elektrody umístěné molekuly nejsou zcela natočeny a propouštějí méně světla. To má za následek i menší jas a kontrast, což ovšem nemusí vždy vadit. Pozorovací úhly jsou velké a jak už bylo řečeno, barevné podání je vynikající. Avšak za to budou muset zákazníci počítat s vyšší cenou.



Další zdroje

1. Aspinwall, J. *Osobní počítač na maximum*, Brno, Computer Press 2006, počet stran: 256. ISBN 80-251-1136-9.
2. Minasi, M. *Velký Průvodce Hardwarem*, Praha, Grada 2003, počet stran: 768. ISBN 80-247-0273-8.
3. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
4. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
5. wikipedia otevřená encyklopedie, kapitola CRT [online] dostupný z http://en.wikipedia.org/wiki/Cathode_ray_tube
6. Server LCD monitory [online] dostupný z <http://www.lcdmonitor.cz/>
7. Plazma vs. LCD server [online] dostupný z <http://www.plazma-lcd.cz/cz>



Shrnutí kapitoly

Grafická karta je adaptér, jenž tvoří rozhraní mezi počítačem a monitorem, odesílá do monitoru signály, které vytvoří na obrazovce požadovaný obraz. Během vývoje byla ustanovena celá řada standardů od CGA, EGA, VGA, XGA až po SVGA.

Grafický procesor řídí činnosti grafické karty, ovládá všechny elementy spojené s vykreslováním pixelů na obrazovku. Konvertor digitálního signálu RAMDAC je obvod integrovaný přímo do čipu grafického adaptéru, který převádí digitální obraz vypočítaný grafickým procesorem na analogový signál monitoru. Rychlost konvertorů se udává v MHz a z části na něm závisí obnovovací frekvence monitoru. Obdobně jako je tomu u biosu systému na základní desce, tak i grafický adaptér má svůj vlastní BIOS nezbytný pro funkčnost grafické karty a mimo jiné obsahuje instrukce pro rozhraní mezi grafickým adaptérem a systémem. Je uložen v paměťovém čipu typu ROM, obdobně jako u základní desky může být verze ROM paměti Flash-ROM či EEPROM a BIOS tak může být

aktualizován.

V současné době jsou paměťové čipy grafických karet typu DDR SDRAM. V minulosti byly grafické karty vybavovány čipy obdobné technologie jako paměťové moduly RAM. Velikost paměti určuje maximální možné rozlišení a počty barev, jež můžeme na monitoru zobrazit. Pro určitá rozlišení (i vysoká) nám postačí i méně paměti, než je v běžných adaptérech k dispozici. Z toho důvodu nemusí zvětšení paměti vést nutně ke zvětšení výkonu grafického adaptéru, potažmo celého systému.

Kapacita potřebné paměti = $H_p \times V_p \times B_p$, kde H_p určuje počet pixelů v horizontálním směru, V_p určuje počet pixelů ve vertikálním směru a B_p určuje počet bytů pro zobrazení v příslušné barevné hloubce.

Monitor CRT (Cathode Ray Tube) je vyroben na základě stejné technologie jako televizor a před nástupem LCD panelu byl v podstatě zobrazovacím standardem u stolních počítačů. CRT monitory obsahují elektronové dělo, které emituje po zahřátí proud elektronů o vysoké rychlosti. Elektronové svazky jsou vychylovány pomocí vychylovacích cívek tak, aby opisovaly shora dolů a zleva doprava jednotlivé řádky obrazovky. Vychylovací a zaostřovací cívky se nacházejí podél elektronky. Na vnitřní straně obrazovky je nanesená stínící maska děrovaná nebo šterbinová (podle typu obrazovky), která slouží k seskupení RGB spektra do jednoho bodu. Masky je v podstatě mříž, která propustí pouze úzký svazek elektronů.

Horizontální frekvence je řádkový kmitočet, měří se v kHz a udává rychlost, s jakou paprsek přejíždí z levé strany obrazovky na pravou.

Vertikální frekvence je obnovovací kmitočet obrazu, měří se v Hz a udává počet obrazů překreslených za jednu sekundu.

LCD (*Liquid crystal display*) monitory jsou zobrazovací jednotky, které při své činnosti využívají technologii kapalných (tekutých) krystalů. Pro získání obrazového výstupu potřebujeme zajistit světelný zdroj. Ten je zajištěn světelnými katodami a odraženým světlem. Světlo je rozkládáno do barevných spekter tří složek – RGB. Body obrazu jsou ovládány příslušnými tranzistory a jsou ohraničeny dvěma polarizačními filtry, barevným filtrem a zarovňovacími vrstvami. Tranzistory kontrolují příslušné napětí mezi vyrovnávacími vrstvami a elektrické pole působí na změnu natočení tekutých krystalů, čímž vzniká výsledný barevný odstín. Každý obrazový bod, resp. pixel je složen ze tří subpixelů, jeden pro každou složku RGB spektra. Nová výrobní technologie S-IPS potřebuje dokonce dva tranzistory na subpixel. Každý subpixel je ovládán jedním tranzistorem. Z toho plyne, že při rozlišení 1600 x 1200 obsahuje LCD panel 5 760 000 resp. 11 520 000 tranzistorů. Obecně LCD display s aktivní maticí je označován zkratkou TFT (*Thin Film Transistor*), což vyjadřuje zapouzdření tranzistorů do tenké pružné fólie.

Výrobních technologií je několik a jsou rozděleny do tří kategorií. První kategorií a zároveň původní technologií je TN (*Twisted Nematic*). Druhou kategorií tvoří technologie MVA (*Multi-Domain Vertical Alignment*) vyvinutá firmou Fujitsu Siemens a PVA (*Patterned Vertical Alignment*) vyvinutá firmou Samsung. Třetí a poslední kategorií je technologie IPS (*In-Plane Switching*), která byla pro svou velice dlouhou odezvu zdokonalena na S-IPS, AS-IPS (*Advanced Super*).



Úkol k řešení 12.1

Vypočtete potřebnou velikost paměti v režimu True color pro rozlišení 1600 x 1200.



Kontrolní otázka 12.1

Vyjmenujte vývojové standardy grafických adaptérů.



Kontrolní otázka 12.2

Popište součásti grafického adaptéru.



Kontrolní otázka 12.3

Jaké typy paměti používají grafické adaptéry?



Kontrolní otázka 12.4

Popište princip zobrazení CRT monitoru.



Kontrolní otázka 12.5

Co určuje horizontální a vertikální frekvence ?



Kontrolní otázka 12.6

Jaké technologie výroby LCD panelu znáte ?

13. Optická média, mechaniky CD a DVD



Čas ke studiu: 3 hodiny



Cíl Po prostudování této kapitoly budete umět

- definovat základní pojmy v oblasti optických médií
- charakterizovat důležité pojmy spojené se čtením a zápisem optických médií
- stručně popsat vývoj standardů CD
- popsat proces výroby CD disku
- popsat strukturu zapisovatelného média
- vyjmenovat hlavní technologické rozdíly CD a DVD disku
- popsat činnost CD mechaniky
- charakterizovat novou technologii Blu-Ray



Výklad

Dnes bereme jako samozřejmost, že náš osobní počítač, popřípadě notebook je vybaven mechanikou pro zápis DVD disku. Také struktura datového média, možnost ukládat na disky zvukové i datové stopy, vytvářet multimediální CD, to jsou fakta nad kterými se běžný uživatel nepozastavuje. Málokdo si uvědomuje, že aby tohle všechno bylo možné, muselo být vydáno od doby vzniku prvního CD disku a mechaniky celá řada standardů. Následující kapitola nám podá přehled vývoje těchto standardů, vysvětlí rozdíl mezi továrním a tzv. vypalovaným diskem a vysvětlí další pojmy z oblasti optického záznamu dat.

13.1 Čtení a záznam dat z magneto-optického disku

Čtení je založeno na Kerrově efektu (elektro-optický dvojlom) - provádí se laserovým paprskem nižší intenzity, z optického disku je prováděno laserovým paprskem, který dopadá na médium a odráží se od něj. Následně jsou snímány jeho vlastnosti (např. intenzita, stáčení roviny polarizovaného světla). Naopak záznam je prováděn do magnetické vrstvy za současného působení laserového paprsku vysoké intenzity. Povrch média se zahřeje na Curiovu teplotu (dojde k jeho změně z feromagnetického materiálu na materiál paramagnetický, který má velmi malou koercitivní sílu).

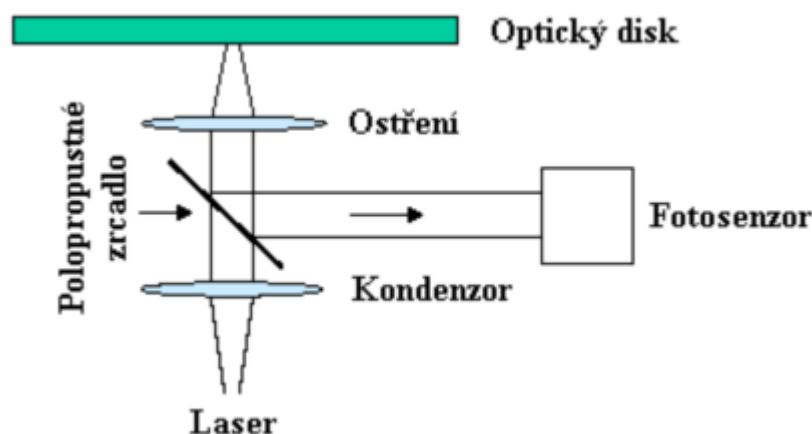


Pojem k zapamatování: Curiova teplota, koercivita

Curiova teplota je teplotní bod, při kterém může být materiál magnetizován ve velmi slabém magnetickém poli.

Koercivita neboli koercivní síla je schopnost permanentního magnetu odolávat demagnetizaci externím magnetickým polem a také svým vlastním demagnetizačním polem.

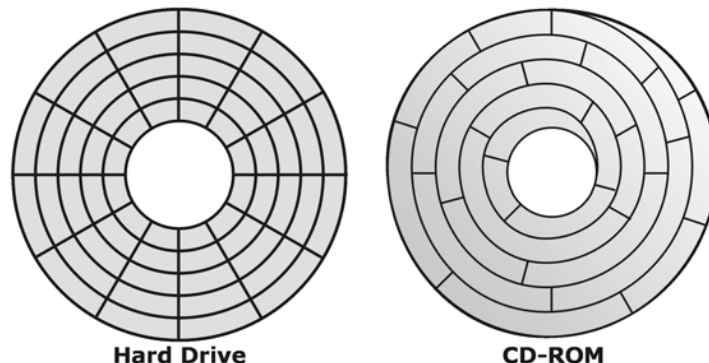
Magnetickým polem malé intenzity se změní magnetická orientace záznamového materiálu. Po ochladnutí zahřátého místa zůstává záznam zachován. Záznam se provádí ve dvou fázích: na dané místo se zaznamenají samé nuly, na patřičná místa se zaznamenají jedničky.



Obr. 13.1 Čtení z optického disku

13.2 disk CD

CD-ROM (Compact Disk-Read Only Memory). Vývoj začal v roce 1978. Pod tímto názvem jsou známy jak záznamová média, tak mechaniky pro jejich čtení. Dále jsou známy rozšířená média, která vycházejí z technologie CD-ROM. Mluvíme o médiích CD-R (Compact Disk-Recordable) a CD-RW (Compact Disk-ReWritable). Na rozdíl od pevných disků jsou data zapisována do spirály (viz. obr. 13.2), která má počátek u středu disku a pokračuje k jeho okraji. Na této spirále jsou tzv. pity, resp. prohlubně a landy, které určují, zda se paprsek laseru odrazí či rozptýlí. Podle toho je určeno, zda se jedná o 0 nebo 1. Jejich kapacita je max. 700 MB nebo 80 minut audio záznamu.



Obr. 13.2 Srovnání HDD A CD ROM

Data jsou tedy uložena ve spirále jako posloupnost pitů a landů. Hloubka pitu je 0,125 mikronu, šířka 0,6 mikronu, závity spirály jsou vzdáleny o 1,6 mikronu a celková délka dosahuje 5,77 km, u 80 minutového CD je to 6,2 km. Laserový paprsek je ostřen na land, od kterého se odráží s vyšší intenzitou než od pitu, kam dopadá mírně rozostřen. Čtení dat tedy probíhá v závislosti na intenzitě odraženého paprsku od média.



Pojem k zapamatování: Pity a landy (Jamky a pevniny)

Jednotlivé pity a landy jsou interpretovány takto:

1 - změna z pitu na land nebo z landu na pit; 0 - setrvalý stav (pit nebo land)

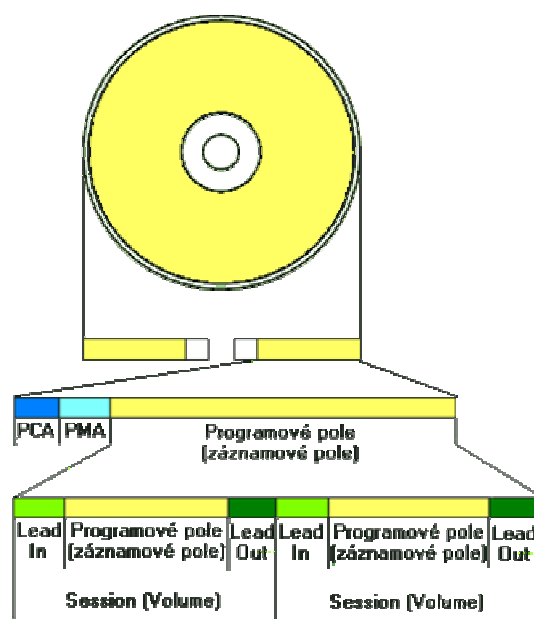
Vzhledem k omezené citlivosti fotosenzoru je nezbytné, aby mezi dvěma jedničkami (změna z pitu na land nebo naopak) byly vždy alespoň dvě nuly (dvě po sobě rychle následující změny intenzity odraženého paprsku nelze spolehlivě rozpoznat). Naopak je rovněž nutné zabezpečit, aby posloupnost po sobě následujících nul nebyla nikdy delší než deset, protože jinak by došlo ke ztrátě synchronizace čtecí jednotky a nebylo by možné určit jejich přesný počet.

Výše uvedené požadavky jsou realizovány pomocí speciálního kódování EFM (*Eight To Fourteen Modulation*), které každému bytu přiřazuje jednoznačně 14bitový vzorek. Kódování EFM splňuje vytyčené požadavky pouze uvnitř jednoho vzorku, nikoliv však mezi po sobě následujícími vzorky. Mezi každé dva byty (po překódování 14 bitové vzorky) se tedy ještě zapisují 3 bity, jejichž hodnota nemá žádný informační význam. Tyto tři bity mají hodnotu závislou na přilehlých 14bitových vzorcích a mají za úkol zaručit, že ani mezi těmito vzorky nebude porušeno pravidlo: mezi dvěma jedničkami minimálně dvě nuly a maximálně deset nul.

Struktura CD: Hustota stop je 625 závitů na milimetr, celkem 22188 závitů na 74 minutové médium.

Disk je rozdělen na 6 základních částí:

1. Oblast pro uchycení média HCA (*Hub Clamping Area*).
2. Oblast kalibrace výkonu PCA (*Power Calibration Area*) pouze pro CDR, CDRW, určeno pro test optimálního výkonu laseru.
3. Oblast programové paměti PMA (*Program Memory Area*) pouze pro CDR, CDRW pro uložení TOC (*Table of Contents* – tabulka obsahu) v době kdy ještě není ukončena záznamová sekce.
4. Zaváděcí oblast Lead-In. Počáteční adresy a délky všech stop, i sekce.
5. Oblast pro data - Program Data Area.
6. Ukončovací oblast Lead-Out. Žádná konkrétní data, pouze značka.



Obr. 13.3 Struktura datového CD

13.3 Stručná historie

Vývoj CD ROM se datuje do roku 1978, kdy společnosti Philips a Sony vyvinuly společně CD nosič a mechaniku – Philips se specializoval na médium, Sony především na převodníky a obvody mechaniky. Bylo také nutné definovat standard pro formáty médií, aby byly čitelné ve všech mechanikách různých výrobců. Tyto standardy byly označovány pod pojmem kniha s příslušným barevným označením. Prvním standardem byl v roce 1982 definován Standard Red Book. Z tohoto standardu vzešly všechny následující standardy a definuje základní charakteristiky CD ROM, mimo jiné dvanácticentimetrový průměr nosiče, tak aby se na něj vešla celá Devátá symfonie Ludwiga van Beethowena. Následovaly další standardy :1984 Yellow Book i pro ukládání dat, 1986 Green Book pro uložení zvuku i videa na jeden disk, 1989 CDRom XA – Microsoft kombinace předchozích standardů, zvuk i video na CD na PC, 1996 Orange Book, která pro CDR, CDRW definuje DAO, TAO, PW (*packet writing*) a 1995 Blue Book, tedy CD Extra, což znamená dvě sekce, první zvuková a druhá datová, tak aby to četly jak stolní tak PC mechaniky.

Také souborové systémy na CD discích bylo nutné standardizovat a tak se v roce 1985 představitelé firem sešli v hotelu Sierra v Lake Tahoe, kde vznikl první standard High Sierra, definován tak, aby všechna CD byla přečtena ve všech mechanikách. V roce 1988 následoval standard ISO 9660, je doplnění Sierry tak, aby se dalo číst téměř ve všech OS: DOS, Unix, Mac. V názvech souborů

umožňoval použít pouze znaky a-z, 0-9 a podtržítka při maximální délce 8 znaků. Adresář byl definován také na maximální délku 8 znaků a hloubku 8 úrovní. V dalších letech se standard ISO 9660 rozšířil o Level 2, resp. Level 3, umožnil tak použití 30 znaků pro soubor, byla odstraněna podmínka spojitosti souborů a další. V roce 1995 navrhla firma Microsoft formát Joliet, především pro použití v operačním systému Windows 95. (Pzn: Win95 = Chicago, Joliet je městečko poblíž Chicaga, ve kterém byl zatčen hrdina filmu The Blues Brothers) Standard rozšiřoval použití názvu souborů a adresářů na 64 znaků, znakovou sadu Unicode, adresáře mohou obsahovat extenze a zanořovat se více než do 8 úrovní. Standard také umožňoval použití více sekcí na jedno médium. Dalšími nově vyvinutými standardy jsou UDF pro paketový zápis, ten podporuje 255 znaků v názvech souborů, Macintosh HFS pro Mac OS, který je nečitelný v PC, a Rock Ridge z roku 1994, jenž rozšiřuje ISO 9660 o vlastnosti unixových souborových systémů.

13.4 Výroba CD disku

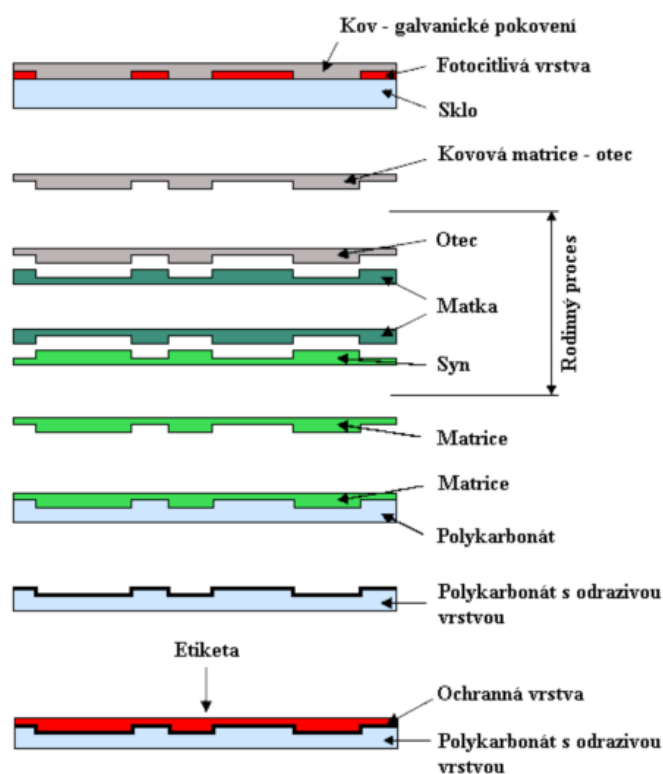


Pojem k zapamatování: Tovární výroba CD disku

Originální média se vyrábějí lisováním, tzv. rodinným procesem, kdy je vyrobena kovová matrice pro vyražení obrazu do polykarbonátu.

CD tovární disky jsou vyráběny lisováním, nikoliv vypalováním, tzv. rodinným procesem. Tento proces by se dal shrnout do následujících devíti kroků:

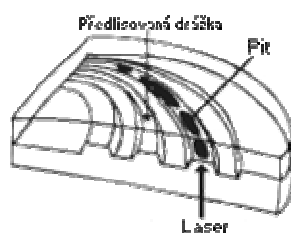
1. Pokrytí disku fotorezistentním materiálem.
2. Laserový záznam dat do fotorezistentní vrstvy.
3. Výroba vzorového záznamu – vznik jamek po polížení vrstvy hydroxidem sodným a následným rozpuštěním ozářených míst.
4. Galvanické pokovování – kovová matrice je nazvaná otec.
5. Oddělení matrice od skleněného vzorového disku.
6. Výroba médií 18g, roztavený polykarbonát, 350 st.Celsia, 138 Mpa, výroba 1 disku za 2 sekundy.
7. Pokovení tenkou vrstvou hliníku pro zajištění odrazivosti.
8. Zalití disku ochranným lakem – ochrana před oxidací.
9. Dokončení média – etiketa.



Obr. 13.4 Tovární výroba disku rodinným procesem

13.5 CDR a CDR-W disky

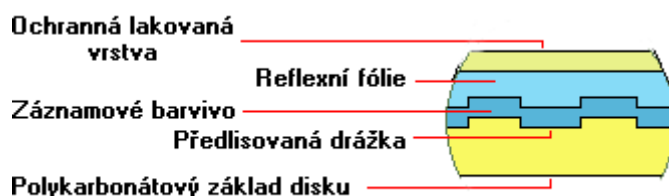
Princip čtení dat z CD-R média je stejný jako u továrního lisovaného média, laserové světlo je různým způsobem odráženo od jamek a pevnin. Rozdílná je však již struktura samotného disku. Jak už bylo řečeno, lisovaná média mají data uložena v polykarbonátové vrstvě ve spirálovité stopě z jamek a pevnin. U nového CD-R média je při výrobě vytvořena vystouplá drážka (vylisovaná *pregroove* stopa), do které jsou jamky vypáleny CD-R mechanikou.



Obr. 13.5 Vylisovaná stopa pro vypalování jamek do CD-R média

Záznamová vrstva je tvořena organickým barvivem: cyanine: zelená, phthalocyanine: zlatá, azo: modrá. Záznam je prováděn laserovým paprskem vyšší intenzity, tento paprsek spálí organické barvivo, které pak již nepropouští světlo a nemůže tedy dojít k jeho odrazu od odrazivé vrstvy. Tímto se vytvoří ekvivalenty jednotlivých pitů a landů, což dovoluje, aby zaznamenané CD-R médium bylo čteno v běžné CD-ROM mechanice. Při záznamu na CD-R je nutné dbát na to, aby proud dat posílaný do CD-R mechaniky byl kontinuální, jinak dojde ke zničení média.

Při záznamu dat se používá stejná vlnová délka jako při čtení, tedy 780 nm avšak výkon laseru je zvětšen na desetinásobek. V místě dopadu se organické barvivo zahřívá na teplotu 280 - 300 stupňů a stává se neprůhledným.



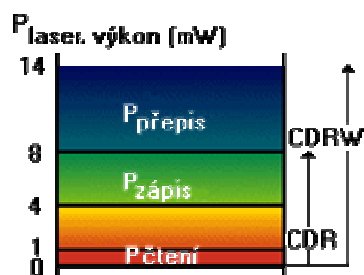
Obr. 13.6 Struktura CD-R média

Přepisovatelné médium CD-RW obsahuje také polykarbonátový základ s předlisovanou stopou, ovšem obsahuje více vrstev. Druhou vrstvu tvoří dielektrický materiál (izolant) za níž následuje záznamová vrstva, ta je opět překryta dielektrikem, teprve poté je disk pokryt hliníkovou odrazovou vrstvou.

Záznam se provádí na principu změny fáze záznamové vrstvy: krystalická odráží více světla, zatímco amorfni odráží méně světla.

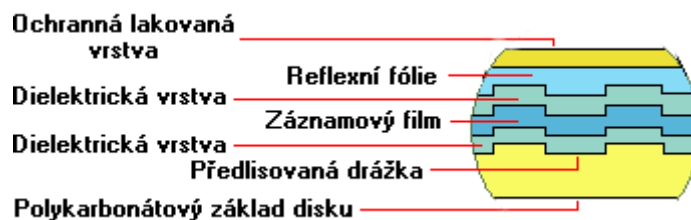
Záznamová vrstva má polykrystalickou strukturu, při záznamu dat se laser přepíná mezi dvěma výkonovými nastaveními P-write a P-Erase, P-write používá vyšší výkon a zahřeje se až na 500 až 700 stupňů, záznamová vrstva ztratí polykrystalickou strukturu. Druhé výkonové nastavení laseru zahřívá materiál pouze na teplotu okolo 200 stupňů, záznamová vrstva měkne a vrací se zpět polykrystalická struktura.

CD-RW mechanika tedy pracuje se třemi intenzitami laseru: PW (PWrite): používá se pro zápis amorfni části, PE (PErase): používá se pro zápis krystalické části a PR (PRead): používá se pro čtení a společně s PWrite pro zápis amorfni části.



Obr. 13.7 Tři intenzity laseru CD-RW mechaniky

Stopa zaznamenaného média je pak tvořena částmi s amorfni fází a částmi s krystalickou fází, které opět vytvářejí ekvivalenty pitů a landů. Rozdíl intenzit paprsků odražených od amorfni částí a od krystalických částí je však menší než v případě pitů a landů u CD-ROM (popř. částí se spáleným a nespáleným organickým barvivem u CD-R).



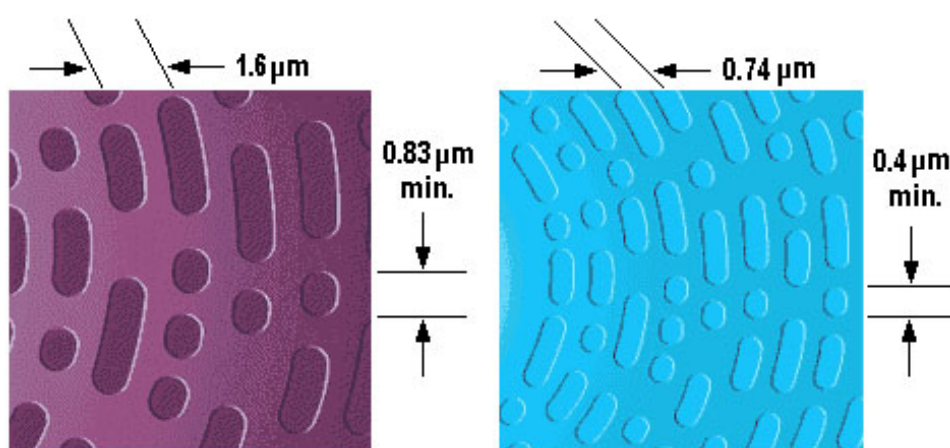
Obr. 13.8 Struktura CD-RW média

13.6 DVD disk

První DVD disky a mechaniky byly uvedeny do prodeje v roce 1997. V této době již bylo jasné, že kapacita CD médií je již nedostačující. DVD disk je podobně jako CD disk z polykarbonátového základu o průměru 120 mm a tloušťce 1,2 mm. Záznam je opět tvořen stopou s jamkami a pevninami. Na rozdíl od CD je však toto médium dvouvrstvé a může být i oboustranné, přičemž při výrobě jsou vrstvy lisovány odděleně a následně pak slepeny. Mimo tento rozdíl oproti CD disku jsou známy další faktory zvyšující kapacitu DVD média. Jedná se o následující:

- zkrácení délek jamek a pevnin (z 0,9 na 0,4 mikronu),
- zmenšení rozteče mezi závity (z 1,6 na 0,74 mikronu),
- mírné zvětšení plochy média využitelné pro záznam,
- zmenšení poměru redundance záznamu – z celkové velikosti sektoru je určeno více bytů pro data,
- zvýšení účinnosti modulace při záznamu, účinnosti kódu pro opravu chyb, atd.

Hlavním rozdílem vyplývajícím z těchto faktů je tedy kapacita média. Zmenšení velikosti jamek má za následek volbu laseru o kratší vlnové délce. Stopa je opět spirálovitá, její hustota je 1351 závitů na mm, tedy 49324 závitů na vrstvu, výsledná délka stopy je 11,84 km.



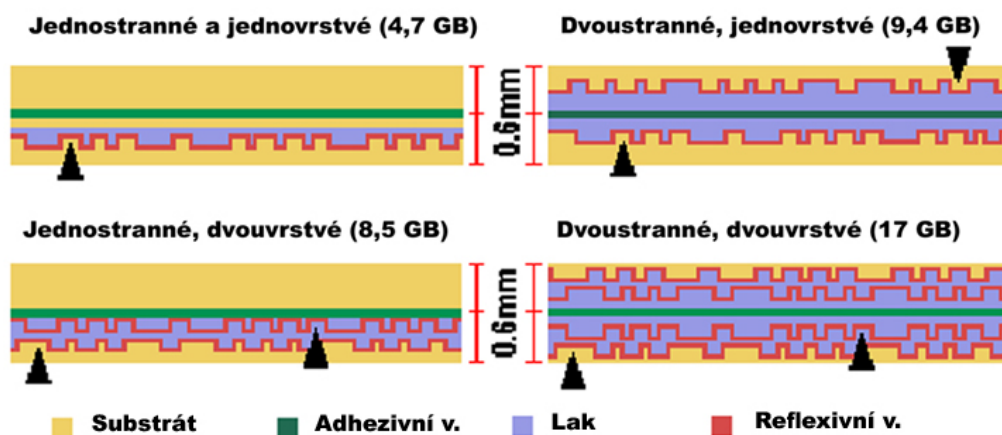
Obr. 13.9 Srovnání stop CD a DVD disku

Disk je rozdělen na 4 základní části: oblast uchycení disku (*HCA – Hub Camping Area*), zaváděcí oblast (*Lead-In*), oblast pro data (*Data Zone*) a ukončovací oblast (*Lead-Out*).

Parametr	CD-ROM	DVD
Průměr disku [mm]	120	120
Tloušťka disku [mm]	1,20	1,20
Tloušťka substrátu [mm]	1,20	0,60
Rozteč stop [μm]	1,60	0,74
Minimální velikost pitu [μm]	0,83	0,40
Vlnová délka laseru [nm]	780 (červená)	635 / 650 (oranžová)
Kapacita jedné vrstvy [GB]	0,65	4,70

Tab. 13.1 Srovnání parametrů CD a DVD disků

DVD disky se vyrábí ve 4 formátech: SS/SL (*Single Sided, Single Layer*) s kapacitou 4,7 GB, SS/DL (*Single Sided, Double Layer*) s kapacitou 8,5 GB, DS/SL (*Double Sided, Single Layer*) s kapacitou 9,4 GB a DS/DL (*Double Sided, Double Layer*) s kapacitou 17,0 GB.



Obr. 13.10 Struktura vrstev DVD disku

Zajímavým pojmem u DVD disků se kterým se u CD nesetkáme je ochrana RPC – *Regional Playback Control*. Protože se DVD disky významnou měrou začlenily do distribuce ve filmovém průmyslu, byla vymyšlena ochrana znemožňující použít DVD disk v mechanikách pro odlišná geografická pásma. Důvodem jsou především různá časová období uvedení filmů pro tato území, lišící se mnohdy až o mnoho měsíců. RPC ochrana rozděluje těchto 8 regionů: 1 USA a Kanada, 2 Japonsko a Jižní Afrika, 3 Asie, 4 Austrálie, 5 Evropa, 6 Čína, 7 je rezervována a číslo 8 je vyhrazeno pro letadla a lodě.

13.7 Princip činnosti CD mechaniky



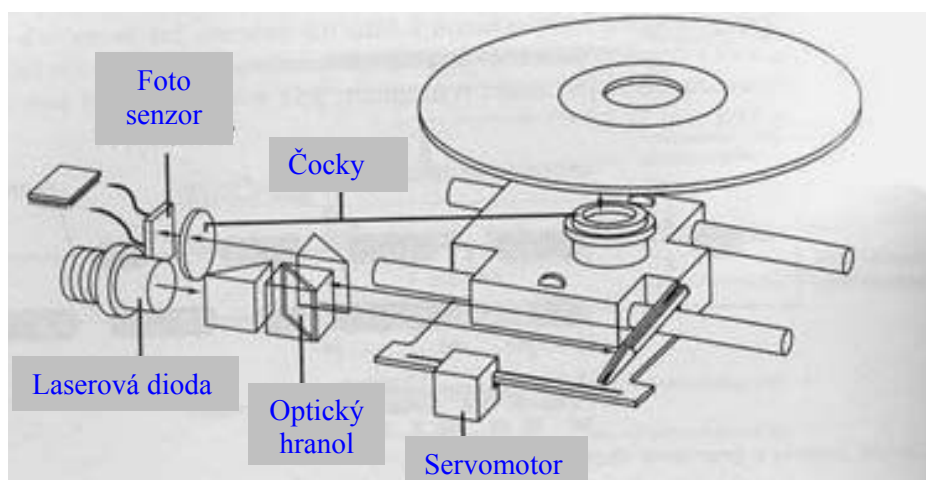
Pojem k zapamatování: Přenosová rychlost, přístupová doba

Přenosová rychlost: určuje, s jakou rychlostí mechanika čte soubory, je to ustálená rychlost, až poté co se mechanika roztočí na plnou rychlost otáčení a najede na požadované místo. **Pro CD je Single speed: 153,6 kB/s, pro DVD je 1,385 MB/s.** Od této rychlosti se počítají rychlosti mechanik s příslušným násobkem (16x, 32x, 48x, 52x).

Přístupová doba: stejně jako u HDD je to zpoždění mezi přijetím příkazu pro čtení a okamžikem načtení prvního bitu dat. Např. pro násobek rychlosti 24x je to 95 ms.

Mechaniky pracují podle následujících kroků:

1. Laserová dioda emituje laserový paprsek směrem k pohyblivému zrcátku.
2. Na základě příkazu z CPU přesune servomotor zrcátko pod stopu, z níž se mají číst data.
3. Po dopadu na jamky a pevniny se světlo láme a odráží zpět, je ostřeno čočkou pod médiem, odráží se od zrcátka na optický hranol.
4. Odražené světlo optický hranol směřuje na další čočku.
5. Poslední čočka směřuje odražené světlo na fotocitlivý senzor, ten převádí světelné impulzy na elektrické.
6. Elektrické impulzy jsou dekodovány mikroprocesorem a předány jako data.



Obr. 13.11 Princip činnosti CD mechaniky

Podle způsobu čtení datové spirály byly v minulosti CD-ROM mechaniky rozděleny do dvou skupin: CLV (Constant Linear Velocity): data jsou čtena konstantní lineární rychlostí rychlost otáček čteného média musí být neustále přizpůsobována tak, aby byla dodržena konstantní rychlost čtených dat (u single speed je to cca 1,3 m/s): u středu je rychlost otáček vyšší (500 ot/min), u kraje je rychlost otáček nižší (200 ot/min), na tomto principu pracují zejména starší mechaniky, protože je nutné neustále přizpůsobovat rychlost otáček, mají tyto mechaniky vyšší přístupovou dobu (cca 500 ms), udávaná rychlost čtení odpovídá rychlosti, se kterou je mechanika schopna číst data nezávisle na tom, kde se na médiu nacházejí.

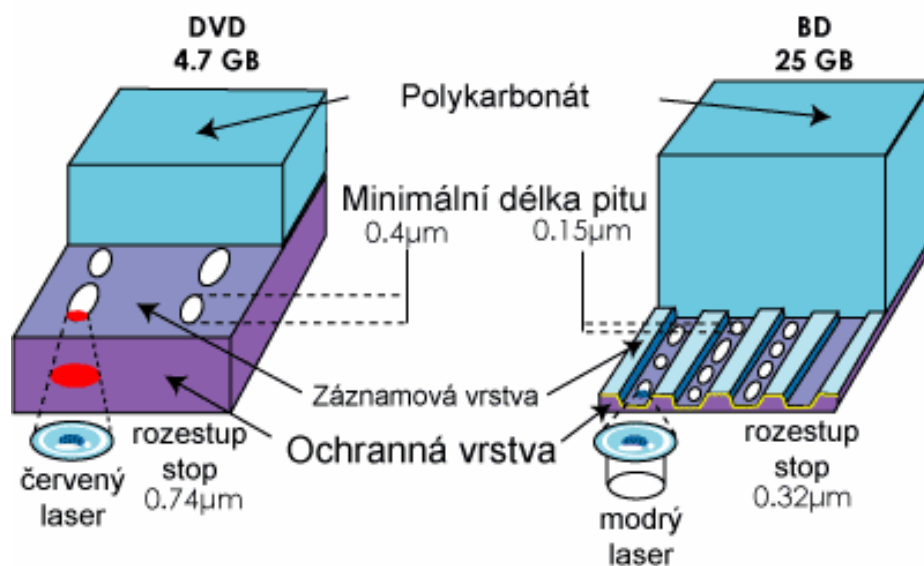
CAV (Constant Angular Velocity): data jsou čtena konstantní úhlovou rychlostí, rychlost otáček média je konstantní a není nutno ji přizpůsobovat v závislosti na tom, odkud se čtení provádí, na tomto principu pracují zejména dnešní moderní mechaniky s rychlostí vyšší než 12x, díky konstantní úhlové

rychlosti je možné, aby CAV mechaniky dosahovaly nižších přístupových dob (cca 100 ms) než je tomu u mechanik CLV, udávaná rychlost pak nevyjadřuje přenos dat, který je mechanika schopna poskytnout v libovolný okamžik, ale vyjadřuje maximální přenos, který nastává u okraje média (kde lineární rychlost je nejvyšší).

13.8 Blu-Ray - blízká budoucnost?

V současnosti se i DVD média dostávají do kategorie nedostačující médium pro svou kapacitu. To si samozřejmě uvědomili i přední světoví výrobci a tak v roce 2001 byl ohlášeno vznik optického média třetí generace ve spolupráci firem Matsushita a Sony. Uvedené velkokapacitní médium je určeno jak pro zaznamenávání video záznamů, tak pro informační technologie. O rok později (2002) vzniklo konsorcium Blu-ray Disk Association sdružující na 70 velkých firem zabývajících se spotřební elektronikou. Název vznikl z Blue-Ray (modrý paprsek), kvůli nemožnosti registrovat běžně užívaný název jako ochranou známku, registrované označení Blu-Ray vzniklo vypuštěním písmena *e*.

Základními vlastnostmi a vlastně i rozdíly oproti DVD jsou: vyšší vlnová délka 405 nm – kvůli užšímu paprsku a z toho plynoucí možnosti zhuštění záznamu. Pro zaostření paprsku o vyšší vlnové délce musí být čočka laseru co nejblíže k záznamové vrstvě média, médium má proto záznamovou vrstvu jen 0,1 mm pod povrchem média. Kapacita dosahuje 23,3, 25, 27 GB u jednovrstvých disků a dokonce až 50 nebo 57 GB u dvouvrstvých disků. Průměr a tloušťka disku zůstává na stejné hodnotě jako u DVD a CD disku. Blu-Ray disk využívá on-groove způsob zápisu, což znamená zápis na vrchol vodící stopy. Naproti tomu DVD používá zápis in-groove (zapisuje mezi vodící stopy). Rozestup vodících stop je 320 nm. Přenosová rychlost dosahuje 36 Mbps. *Groove* neboli vodící drážka navíc nemá tvar přesné spirály, ale je zakřivený do sinusoidy, aby bylo možné data lépe adresovat. Toto zakřivení se nazývá *wobble* a slouží jako orientace pro laser.



Obr. 13.12 Srovnání DVD a Blu-Ray disku

Závěrem krátkého představení můžeme říci, že médium a mechaniky Blu-Ray mají před sebou nejistou budoucnost na poli informačních technologií, kde je ohrožují holografická média, která by se měla dostat na trh v horizontu několika let. V oblasti video nosičů bude pro toto médium těžké porazit rozšířenost DVD nosičů. Jejich vyšší cena a náklady na pořízení zapisovací mechaniky zatím používání DVD nosičů příliš neohrožuje. Velmi mnoho si slibovali od Blu-Ray producenti filmů pro jejich údajně neprolomitelnou ochranu proti pirátským kopiím, avšak ta již byla prolomena a to ještě Blu-Ray technologie není všeobecně známou a rozšířenou technologií.



Další zdroje

1. Aspinwall, J. *Osobní počítač na maximum*, Brno, Computer Press 2006, počet stran: 256. ISBN 80-251-1136-9.
2. Minasi, M. *Velký Průvodce Hardwarem*, Praha, Grada 2003, počet stran: 768. ISBN 80-247-0273-8.
3. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
4. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
5. Understanding CD-R, CD-RW [online] dostupný z <http://www.osta.org/technology/cdqa.htm>
6. CD-R server - vše co se točí okolo vypalování CD a data storage [online] dostupný z <http://www.cdr.cz/>
7. Informační server nejen o optických technologiích [online] dostupný z <http://cdrhard.cz/>
8. Blu Ray Disk Association server [online] dostupný z <http://www.blu-raydisc.com/>



Shrnutí kapitoly

Čtení dat z magnetooptického disku je založeno na Kerrově efektu (elektro-optický dvojlom) - provádí se laserovým paprskem nižší intenzity, z optického disku je prováděno laserovým paprskem, který dopadá na médium a odráží se od něj, záznam je prováděn do magnetické vrstvy za současného působení laserového paprsku vysoké intenzity.

Data jsou na CD disku uložena ve spirále jako posloupnost pitů a landů. Hloubka pitu je 0,125 mikronu, šířka 0,6 mikronu, závity spirály jsou vzdáleny o 1,6 mikronu a celková délka dosahuje 5,77 km, u 80 minutového CD je to 6,2 km. Laserový paprsek je ostřen na land, od kterého se odráží s vyšší intenzitou než od pitu, kam dopadá mírně rozostřen. Čtení dat tedy probíhá v závislosti na intenzitě odraženého paprsku od média. CD tovární disky jsou vyráběny lisováním, nikoliv vypalováním, tzv. rodinným procesem. Tento proces by se dal shrnout do devíti kroků: Pokrytí disku fotorezistentním materiálem, laserový záznam dat do vytvořené fotorezistentní vrstvy, výroba vzorového záznamu, galvanické pokovování, oddělení matrice, výroba polykarbonátových médií, pokovení, zalití disku a dokončení média. Rozdílem DVD oproti CD, mimo možnost záznamu do dvou vrstev, je zkrácení délek jamek a pevnin (z 0,9 na 0,4 mikronu), zmenšení rozteče mezi závity (z 1,6 na 0,74 mikronu), mírné zvětšení plochy média využitelné pro záznam, zmenšení poměru redundance záznamu – z celkové velikosti sektoru je určeno více bytů pro data a zvýšení účinnosti modulace při záznamu. Přenosová rychlost určuje s jakou rychlostí mechanika čte soubory, je to ustálená rychlost až poté, co se mechanika roztočí na plnou rychlost otáčení a najede na požadované místo.

Pro CD mechaniku je základní rychlost tzv. Single speed: 153,6 kB/s, pro DVD je 1,385 MB/s. Od této rychlosti se počítají rychlosti mechanik s příslušným násobkem (16x, 32x, 48x, 52x). Přístupová doba je stejně jako u HDD zpoždění mezi přijetím příkazu pro čtení a okamžikem načtení prvního bitu dat. Např. pro násobek rychlosti 24x je to 95 ms. Média a mechaniky Blu-Ray mají před sebou nejistou budoucnost. V oblasti video nosičů bude pro toto médium těžké porazit rozšířenost DVD nosičů. Jejich vyšší cena a náklady na pořízení zapisovací mechaniky zatím používání DVD nosičů příliš neohrožuje.



Kontrolní otázka 13.1

Jakým způsobem probíhá čtení z CD disku?



Kontrolní otázka 13.2

Popište jednotlivé kroky výroby CD disku.



Kontrolní otázka 13.3

Charakterizujte zapisovatelné a přepisovatelné médium.



Kontrolní otázka 13.4

Vyjmenujte rozdíly CD oproti DVD.



Kontrolní otázka 13.5

Popište jakým způsobem pracuje CD ROM mechanika.



Kontrolní otázka 13.6

Jaké násobky MB/s používáme u vyjádření rychlosti CD a DVD mechanik?

14. Zvuk, audio hardware



Čas ke studiu: 2 hodiny



Cíl Po prostudování této kapitoly budete umět

- definovat základní pojmy v oblasti zvuku, jeho podstaty, komprese a kodeků
- charakterizovat vlastnosti zvukové karty
- popsat vývoj zvukové karty, její výstupy a kritéria hodnocení



Výklad

Žijeme v technologicky moderní době, kdy výrobci čipových sad již jsou schopni implementovat na základní desku rozhraní, které v nedávné době potřebovalo vlastní obslužnou komponentu (kartu), připojenou na rozšiřující sběrnici. A tak dnes mnoho uživatelů téměř neřeší výběr síťové karty, mnohdy i grafické karty, o zvuku ani nemluvě. V tomto případě vezmou za vděk integrovaným součástem základní desky a věnují pozornost výběru reproduktorů či sluchátek. To v dřívější době nebylo možné, počítač byl vybaven pouze *PC speakerem* s velmi omezenými možnostmi a kdo chtěl slyšet zvuk, musel připlatit obvykle velkou částku za nákup zvukové karty. V dnešní době jsou integrované zvukové čipy základních desek natolik kvalitní, že na běžnou každodenní práci i při využití multimediálních aplikací jsou plně postačující. To ovšem neznamena, že by zvukovým kartám bylo odzvoněné. Své o tom jistě vědí nároční uživatelé, ať už v oblasti her, či připojitelných MIDI (*Musical Instrument Digital Interface*) zařízení a další.

14.1 Zvuk a jeho podstata

Zvuk je šířen pomocí vibrací, které stlačují vzduch. Je tedy podélným mechanickým vlněním v látkovém prostředí, které vyvolává v lidském uchu sluchový vjem. Tyto zvukové vlny se ze svého zdroje šíří všemi směry rovnoměrně. Jakmile dorazí do lidského ucha, rozkmitají ušní bubínek a tento vjem je pak převeden na zvuk. Lidé však nejsou schopni slyšet zvuk o všech frekvencích. Frekvence tohoto pro člověka slyšitelného vlnění leží v rozsahu přibližně 16 Hz až 20 kHz. Například nejnížší tón píána má frekvenci 27 Hz, nejvyšší 4 kHz. Na omezené schopnosti lidského ucha vnímat určité frekvence je založena i komprese zvuku, při výsledné kompresi jsou vynechávány všechny zvuky nad určitou slyšitelnou frekvencí, u MP3 se uvádí $f = 15 \text{ kHz}$. Pro zvuk jsou charakteristické dvě jeho vlastnosti: výška a intenzita.



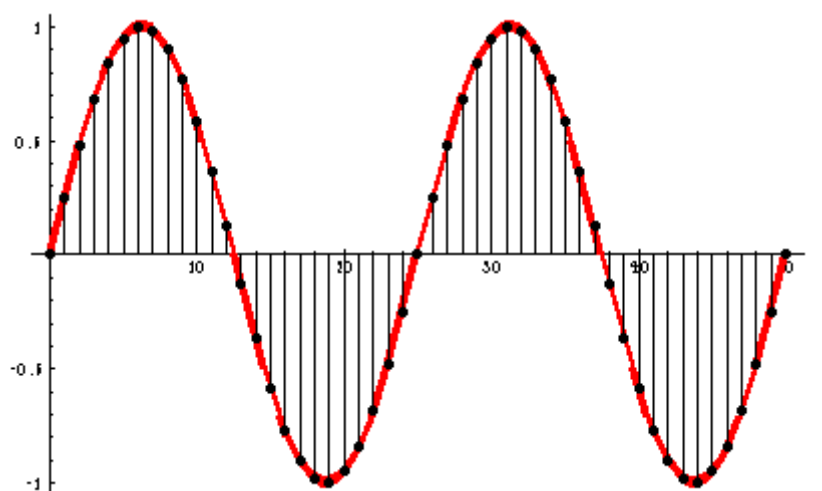
Pojem k zapamatování: Výška a intenzita zvuku

Výška zvuku je měřena v Hertzích (Hz) neboli cyklech za sekundu a je dána rychlostí šíření jednotlivých vibrací, kde jedním cyklem se miní úplná vibrace dopředu a dozadu. Počet Hertzů pak udává frekvenci tónu, čím vyšší je frekvence, tím vyšší je vnímaná výška tónu.

Intenzitu zvuku nazýváme amplitudou a definuje hlasitost, s jakou daný zvuk vnímáme.

Intenzita přímo závisí na síle jednotlivých vibrací, jímž se zvuk šíří. Udává se v decibelech (db), přičemž prostředí lesa dosahuje hlasitosti asi 20 db, průměrná ulice asi 70 db, práh bolesti je označován hranicí 130 db.

Zdroj zvuku je šířen analogovým signálem, který počítač není schopen uchovávat, je nutné tento signál převést na digitální. Tento proces převodu bývá na zvukové kartě prováděn pomocí převodníku ADC (*Analog to Digital Convertor*). Převod se uskutečňuje metodou zvanou vzorkování (*sampling*), která pracuje tak, že v každém časovém (pevně stanoveném) intervalu je zjištěn a zaznamenán aktuální stav signálu (tzv. vzorek – *sample*). Je zřejmé, že čím kratší je tento interval, tím vyšší je vzorkovací frekvence, tím více vzorků bude pořízeno a tím bude výsledný záznam kvalitnější.



Obr. 14.1 Půběh vzorkování

Při záznamu zvuku se využívá Shannonovy vzorkovací věty, která definuje:



Pojem k zapamatování: Výška a intenzita zvuku

Signál spojitý v čase je plně určen posloupností vzorků odebíraných ve stejných intervalech, je-li jejich frekvence větší než dvojnásobek nejvyšší frekvence v signálu.

Známým faktem je, že lidské ucho vnímá zvuky od frekvencí 16 Hz až do frekvencí 20 kHz, tudíž je zřejmé, že frekvence 44,1 kHz použitá pro CD kvalitu je plně dostačující.

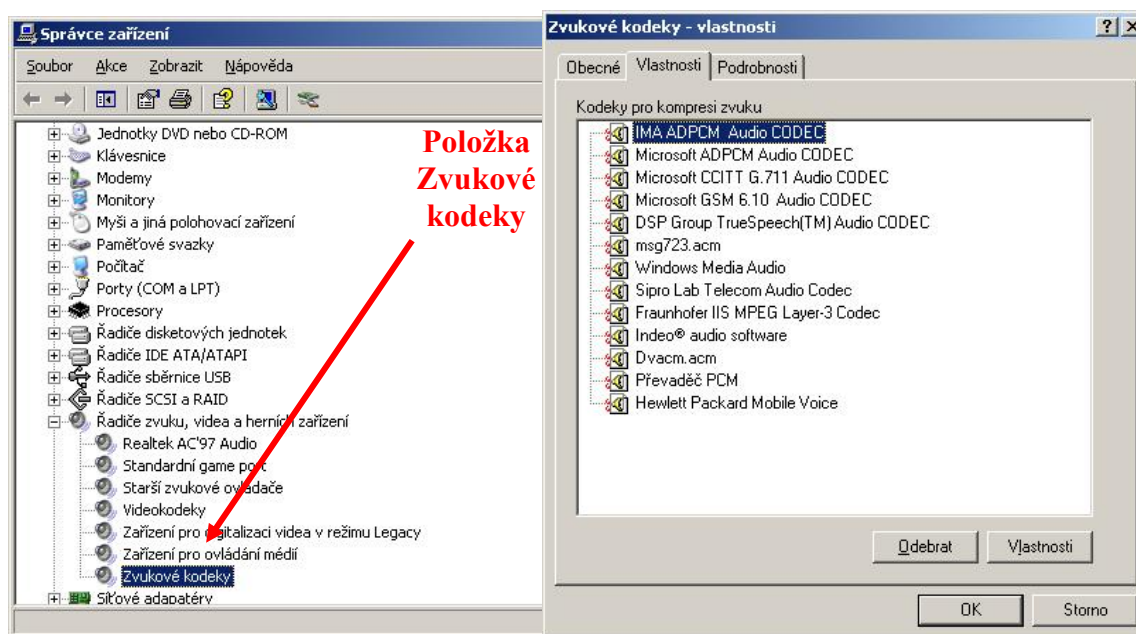
14.2 Komprese zvuku

Každý soudobý zvukový adaptér je schopen reprodukovat zvuk v CD kvalitě, ten je vzorkován s frekvencí 44,1 kHz. Při této rychlosti vzorkování však minuta záznamu jakéhokoliv zvuku může zabírat až 10 MB diskového prostoru. Vzhledem k této náročnosti je nutno zvuk komprimovat a tím zmenšit velikost souboru.

Komprese je v podstatě založena na odřezávání zvuků, které jsou pro lidské ucho neslyšitelné. Jak již bylo řečeno lidské ucho slyší v rozsahu 16 Hz až 20 kHz. Komprimováním jsou zvuky, které jsou za těmito hranicemi, odstraněny.

Před příchodem MPEG komprimací byl nejznámějším kompresním algoritmem pro zvuk ADPCM (*Adaptive Differential Pulse Code Modulation*). Podstatou algoritmu je komprese šestnáctibitových vzorků zvuku na čtyřbitové. Tato komprese má ale za následek ztrátu kvality zvuku. Při instalaci zvukových adaptérů jsou do systému přidány i některé kodeky zvuku (tj. programy, zajišťující kompresi a dekompresi zvuku). Kodek ADPCM je jedním z nich.

V současné době nejpoužívanější komprimací je MPEG (*Motion Picture Experts Group*) a z ní vycházející formát mp3. Algoritmy MPEG principiálně umožňují kompresi až 30:1. Při kompresi je reálný zvuk rozdělen na vzorky, kterým se říká framy (frames), kde se v předem definovaných frekvenčních pásmech zjišťují a vyhodnocují dominantní složky. Podle toho, zda se přitom bere ohled i na sousední framy, rozlišujeme metody na vrstvy (layers). Takto upravený zvuk se komprimuje do formátu MPEG.



Obr. 14.2 Zvukové kodeky nainstalované v PC

Tyto komprese se obecně označují jako ztrátové. MP3 používá kompresi MPEG Audio Layer 3. Při nastavení kompresoru na stereo, 128kB/s a 44,1 kHz vzorkovacího kmitočtu, však dostaneme soubor MP3 kvalitou poslechu srovnatelný s Audio CD, ale zabírající přibližně 10x méně místa na pevném disku.



Korespondenční úkol – standard MPEG

Standard MPEG se dělí na jednotlivé skupiny standardů MPEG 1 – MPEG 4, které charakterizují různé techniky kódování videa a audia. Vyhledejte základní vlastnosti těchto standardů v internetových zdrojích či v odborné literatuře.

14.3 Zvuková karta

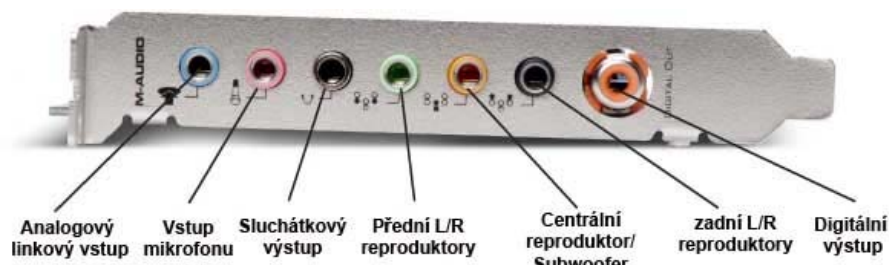
Historie zvukové karty sahá do roku 1987, kdy první zvukové karty byly připojeny pomocí ISA sběrnice a dokázaly reprodukovat zvuk pomocí modulace frekvence. První zvukové karty pracovaly s osmibitovým vzorkováním, umožňující rozlišit pouze 256 zvuků. To platí samozřejmě také pro první Sound Blaster a jeho několik nástupců. Sound Blaster je legendou v oblasti zvukových adaptérů a jako obchodní značka firmy Creative Labs je po desetiletí vyvíjen a zdokonalován až do dnešní podoby. V podstatě firma Creative Labs hraje roli giganta na poli zvukových adaptérů a konkurovat víceméně dokáže pouze firma Hercules. První Sound Blastery dokázaly zaznamenat zvuk nejprve v 13 kHz, později 22,05 kHz. Později začaly být dodávány zvukové adaptéry podporující 16 bitové vzorkování (65 536 zvuků) a karty dokázaly tvořit rozhraní i pro MIDI zařízení. Dnešní zvukové karty standardně využívají 24 bitové vzorkování (16,8 milionů zvuků). Jsou buďto interní, připojené na rozhraní PCI, nebo externí, připojené pomocí USB. Dále podporují Analog-Digital konverze během nahrávání v 8,

16 nebo 24-bit při frekvencích 8, 11,025, 16, 22,05, 24, 32, 44,1, 48 a 96 kHz 24-bit a Digital-Analog konverze během přehrávání s frekvencemi 8, 11,025, 16, 22,05, 24, 32, 44,1, 48 a 96kHz v režimu prostorového zvuku nebo až 192 kHz ve stereo režimu.



Obr. 14.3 Současná interní zvuková karta na PCI rozhraní

Součástí zvukových karet jsou také čipy pro dekódování prostorového zvuku. Prostorový zvuk je dekódován pro 5.1 někdy 6.1 či dokonce 7.1 kanálů. První číslo znamená počet reproduktorů, jednička za tečkou označuje subwoofer neboli speciálně upravený reproduktor pro nízké kmitočty basů. Mimo tyto výstupní kanály prostorového zvuku disponuje zvuková karta dalšími vstupy a výstupy (linkový vstup, mikrofon, sluchátka), které mohou být analogové nebo digitální, podle typu implementovaných obvodů a vybavení zvukového adaptéru. Zvukové adaptéry mohou být vybaveny SPDIF (*Sony/Philips Digital Interface*) rozhraním umožňujícím přímý vstup či výstup digitálního signálu, přitom fyzická vrstva zajišťující přenos digitálního signálu může být metalická ale i optická.



Obr. 14.4 Vstupy a výstupy 5.1 zvukové karty

14.4 Hodnocení zvukových adaptérů

Kvalita zvukových adaptérů se obvykle měří třemi kritérii:

- **Frekvenčním rozsahem.**
- **Celkovým harmonickým zkreslením.**
- **Odstupem signálu od šumu.**

Frekvenčním rozsahem se míní rozsah těch frekvencí, které je zvukový adaptér schopen rozpoznat či reprodukovat na slyšitelné úrovni. Platí, že čím je frekvenční rozsah vyšší, tím je zvukový adaptér lepší. Mnohé běžné zvukové adaptéry podporují rozsah 30 Hz až 20 kHz. Celkové harmonické zkreslení udává linearitu a přímost křivky, vyjadřující frekvenční rozsah zvukového adaptéru. V podstatě lze říci, že se jedná o přesnost reprodukce jednotlivých tónů. Jakákoliv nelineární část,

nacházející se na uvedené křivce, způsobí harmonické zkreslení. Proto platí, že čím je harmonické zkreslení udávané v procentech nižší, tím je adaptér lepší. Hlavním rozdílem mezi adaptéry různých výrobců je právě tento parametr. Odstup signálu od šumu (*Signal-to-Noise Ratio* – SnR nebo S/R) určuje sílu signálu vzhledem k nežádoucím zvukům (šumu). Čím je tento údaj udávaný v decibelech vyšší, tím je kvalitnější daný zvukový adaptér. Dnešní adaptéry disponují hodnotou S/R nad 100 db.



Korespondenční úkol – vysvětlení technických pojmů

V souvislosti s technologií dekódování prostorového zvuku a dalších pokročilých funkcí zvukových adaptérů se můžeme setkat s celou řadou technických pojmů a zkratk. Vyhledejte a vysvětlete následující pojmy: Dolby Digital, DTS, Dolby Digital Surround EX, THX, ASIO, CMSS, QSOUND, X-FI. Jako pomůcku vezměme v úvahu fakt, že se jedná především o registrované pojmy a obchodní značky firem zabývajících se výrobou zvukových adaptérů.



Další zdroje

1. Aspinwall, J. *Osobní počítač na maximum*, Brno, Computer Press 2006, počet stran: 256. ISBN 80-251-1136-9.
2. Minasi, M. *Velký Průvodce Hardwarem*, Praha, Grada 2003, počet stran: 768. ISBN 80-247-0273-8.
3. Messmer, H.P., Dembowski K. *Velká kniha hardware*, Praha, Computer Press 2005, počet stran: 1224. ISBN 80-251-0416-8.
4. Horák, J. *Hardware učebnice pro pokročilé*, Praha, Computer Press 2005, počet stran: 348. ISBN 80-251-0647-0.
5. wikipedia otevřená encyklopedie – kapitola Zvuk. [online] dostupný z <http://cs.wikipedia.org/wiki/Zvuk>
6. Multimedia svět hardware, článek zvukové karty a jejich vývoj. [online] dostupný z http://www.svethardware.cz/art_doc-BC77A7A4115424A5C125725D005327DĚ.html
7. Intel and Microsoft - PC 99 System Design Guide [online] dostupný z http://www-pc.uni-regensburg.de/hardware/TECHDOK/PC_99_1.pdf
8. Sound Blaster - Hear to Believe. Official server [online] dostupný z <http://www.soundblaster.com/>



Shrnutí kapitoly

Zvuk je šířen pomocí vibrací, které stlačují vzduch. Zvukové vlny se ze svého zdroje šíří všemi směry rovnoměrně. Jakmile dorazí do lidského ucha, rozkmitají ušní bubínky a tento vjem je pak převeden na zvuk. Frekvence pro člověka slyšitelného vlnění leží v rozsahu přibližně 16 Hz až 20 kHz. Pro zvuk jsou charakteristické dvě jeho vlastnosti: výška a intenzita. Výška zvuku je měřena v Hertzích (Hz) neboli cyklech za sekundu a je dána rychlostí šíření jednotlivých vibrací, kde jedním cyklem se miní úplná vibrace dopředu a dozadu. Počet hertzů pak udává frekvenci tónu, čím vyšší je frekvence, tím vyšší je vnímaná výška tónu. Intenzitu zvuku nazýváme amplitudou a definuje hlasitost,

s jakou daný zvuk vnímáme.

Každý soudobý zvukový adaptér je schopen reprodukovat zvuk v CD kvalitě, ten je vzorkován s frekvencí 44,1 kHz. Vzhledem ke kapacitní náročnosti je nutno zvuk komprimovat a tím zmenšit velikost souboru. Komprese je v podstatě založena na odřezávání zvuků, které jsou pro lidské ucho neslyšitelné. Lidské ucho slyší v rozsahu 16 Hz až 20 kHz. První zvukové karty pracovaly s osmibitovým vzorkováním, umožňující rozlišit pouze 256 zvuků. Dnešní zvukové karty standardně využívají 24bitové vzorkování (16,8 milionů zvuků).

Kvalita zvukových adaptérů se obvykle měří třemi kritérii: frekvenčním rozsahem, celkovým harmonickým zkreslením a odstupem signálu od šumu. Frekvenčním rozsahem se míní rozsah těch frekvencí, které je zvukový adaptér schopen rozpoznat či reprodukovat na slyšitelné úrovni. Celkové harmonické zkreslení udává linearitu a přímost křivky, vyjadřující frekvenční rozsah zvukového adaptéru. Odstup signálu od šumu (*Signal-to-Noise Ratio* – SnR nebo S/R) určuje sílu signálu vzhledem k nežádoucím zvukům (šumu). Čím je tento údaj udávaný v decibelech vyšší, tím je kvalitnější daný zvukový adaptér.



Úkol k řešení 14.1

Definujte barevné odlišení externích konektorů dle standardu Microsoft 99.

Pozn.: Barvy se netýkají pouze konektorů zvukové karty.



Úkol k řešení 14.2

Vyhledejte nainstalované zvukové kodeky ve vašem PC.



Kontrolní otázka 14.1

Jakým způsobem se šíří zvuk?



Kontrolní otázka 14.2

Definujte dvě základní vlastnosti zvuku.



Kontrolní otázka 14.3

Jaké frekvence je člověk schopen vnímat?



Kontrolní otázka 14.4

Co je to ztrátová komprese?



Kontrolní otázka 14.5

Popište zvukový adaptér.



Kontrolní otázka 14.6

Jaká kritéria určují kvalitu zvukových adaptérů?



Klíč k řešení – kapitola 1

Odpověď na kontrolní otázku 1.1

CPU je hlavní procesorová jednotka (*Central processor Unit*), která čte z paměti instrukce a zpracovává data podle programu. Paměť je zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat). Grafický akcelerátor je procesor grafické karty označený jako GPU (*Graphics Processing Unit*). Sběrnice je soustava vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače. Pomocí těchto vodičů mezi sebou jednotlivé části počítače komunikují a přenášejí data. Porty jsou sériové a paralelní rozhraní, které slouží k připojování periférií. Řadič je zařízení převádějící příkazy v symbolické formě (instrukce) na posloupnost signálů ovládajících připojené zařízení. Jedná se tedy o zařízení, které řídí činnost jiného zařízení. Rozhraní je překlad anglického interface a určuje jednotný styk mezi vstupně výstupními zařízeními.

Odpověď na kontrolní otázku 1.2

Do operační paměti se pomocí vstupního zařízení přes aritmeticko-logickou jednotku umístí program, který bude provádět výpočty. Následně se obdobným způsobem do operační paměti umístí data, se kterými bude program pracovat. Proběhne program, který provádí operace s daty, jednotlivé kroky programu provádí aritmeticko-logická jednotka. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Jednotlivé dílčí výsledky výpočtu jsou ukládány do operační paměti. Po skončení výpočtu jsou výsledky posílány přes aritmeticko-logickou jednotku na výstupní zařízení.

Harvardská architektura má na rozdíl od Von Neumannovy architektury oddělený paměťový prostor pro data a pro program. U Von Neumannovy koncepce je paměť pro data a pro program společná. Tímto způsobem se můžeme na instrukce dívat jako na data a během programu je můžeme měnit a ovlivňovat chování programu.

Odpověď na kontrolní otázku 1.3

Výrobci se musí dohodnout na variantách velikost základních desek, tak aby bylo možné tyto desky montovat do počítačových skříní různých výrobců. Desky stejného formátu tedy musí odpovídat předepsaným rozměrům, předepsaným způsobem upevnění a montování, musí mít stejné požadavky na napájení a podobně. Formátům základních desek se říká form faktory.

Odpověď na kontrolní otázku 1.4

Jedná se o Sockety 1 – 7 pro procesory 486, Pentium a AMD K 5, dále Slot 1 pro Pentium II a III, Slot A pro první AMD Athlony, Socket 370 pro Pentium III a Celeron, Socket 423, Socket 478 pro Pentium 4. V současné době LGA 775 pro všechny současné procesory Intel, Sockety 754 pro starší AMD Athlony XP, Sockety 939, 940 a AM2 pro současné procesory AMD.

Odpověď na kontrolní otázku 1.5

SECC (*Single Edge Contact Cartridge*) – typ pouzdra procesoru.

SEP (*Single Edge Processor*) varianta umístění procesoru bez pouzdra pro levnější varianty Celeron.

SPGA (*Staggered Pin Grid Array*) – typ pouzdra procesoru..

DMA: přímý přístup do paměti (*Direct memory access*). Je to schopnost sběrnice posílat data ze zařízení (pevný disk, mechanika CD, atd.) pomocí řadiče přímo do paměti bez účasti aritmeticko-logické jednotky (procesoru).

IRQ: (*Interrupt Request*) označuje signál, kterým požádá nějaké zařízení procesor o přerušení právě probíhajícího procesu za účelem obsluhy důležitější činnosti.

RISC: Architektura procesoru s redukovanou instrukční sadou. Procesor disponuje instrukční sadou složenou pouze s omezeného počtu jednoduchých instrukcí, které svou sekvencí nahrazují složité instrukce.

CISC: architektura procesoru s komplexní instrukční sadou (*Complex Instruction Set Computer*).

BIOS: (*Basic Input Output System*): programové vybavení uložené v paměti ROM (EPROM, EEPROM, Flash) zajišťující nejzákladnější funkce (např. zavedení OS).

GPU: procesor grafické karty (*Graphics Processor Unit*).

USB rozhraní: (*Universal serial Bus*) sériová sběrnice má již dvě vývojové verze, USB 1.1 (max. přenosová rychlost 12 Mbit/s) a USB 2.0 (480 Mbit/s, pokud je zařízení high-speed).



Klíč k řešení – kapitola 2

Odpověď na kontrolní otázku 2.1

Barevné značení psaného textu rozlišuje tři typy barev: černá označuje obecný a neznámý text, který nespadá do žádné kategorie, modrá označuje klíčové slova jazyka C a zelená komentáře jazyka C.

Odpověď na kontrolní otázku 2.2

Chyby, které nejsou syntaktickou chybou v programu, kompilátor neodhalí, vše je napsáno správně, avšak program se nechová jak má, v horším případě se chová podivně jen v některých ojedinělých případech. Na tyto chyby si musíme dát velký pozor.

Odpověď na kontrolní otázku 2.3

I pro jeden jediný a krátký zdrojový soubor musíme mít vytvořený projekt. Bez otevřeného aktivního projektu není možno kompilovat a spouštět programy.

Odpověď na kontrolní otázku 2.4

Záložka Header Files obsahuje hlavičkové soubory projektu. V těchto souborech jsou obsaženy hlavičky funkcí, které projekt využívá.



Klíč k řešení – kapitola 3

Odpověď na kontrolní otázku 3.1

Při klasickém krokování se program krokuje po řádcích, funkce je napsána na řádku a tak se krokováním provede celá. Proto, pokud chceme zjistit bezproblémové fungování těla funkce, musíme ji odkrokovat krokováním dovnitř funkce – klávesou F11.

Odpověď na kontrolní otázku 3.2

Nejjednodušší způsob jak zjistit hodnotu proměnné během ladění je ukázat na ni kurzorem.



Klíč k řešení – kapitola 4

Odpověď na kontrolní otázku 4.1

ANSI C je verze jazyka, která je dnes podporována téměř všemi rozšířenými překladači. Většina kódu psaného v současné době v C je založena na ANSI C. Jakýkoli program napsaný pouze ve standardním C je přeložitelný a spustitelný na jakékoli platformě, která odpovídá tomuto standardu.

Odpověď na kontrolní otázku 4.2

Soubory se zdrojovým kódem s příponou cpp chápeme jako překladové jednotky. Ty jsou vstupem pro překladač společně se svými využívanými hlavičkovými soubory. Překladač tyto soubory kompiluje a výsledkem překladu je objektový soubor pro každou překladovou jednotku. Objektový soubor má příponu obj a tuto vlastnost si nepleťme s pojmem objektově orientovaného programování. Objektový kód je pak spojovacím programem linkován společně s knihovními funkcemi jazyka C (přípona lib) a v případě, že je vše v pořádku a bez chyb, je vytvořen výsledný spustitelný exe soubor.

Odpověď na kontrolní otázku 4.3

Norma ANSI C tuto hodnotu omezuje na 31.

Odpověď na kontrolní otázku 4.4

Jazyk C je Case-sensitive, to znamená, že rozlišuje malá a velká písmena v názvech identifikátorů, proměnných, ale i klíčových slov.

Odpověď na kontrolní otázku 4.5

Ternární operátor je podmíněný operátor pracující se třemi operandy.

Odpověď na kontrolní otázku 4.6

Cyklus *while* vykoná tělo cyklu vícekrát, nebo vůbec, dokud je hodnota vyhodnocení výrazu pravdivá. Cyklus *for* se poněkud liší svou podobou od ostatních cyklů, především svou konstrukcí. Obsahuje ve svém těle tři výrazy: inicializační, podmíněný a opakovací. Inicializační výraz je vyhodnocen pouze jednou a to před vyhodnocením podmíněného výrazu. Cyklus *do* je jediným z cyklů, který zajišťuje alespoň jedno provedení těla cyklu. To je způsobeno tím, že výraz je testován až po průchodu tělem cyklu. Takže při nesplnění podmínky je tělo cyklu provedeno právě jednou.

Odpověď na kontrolní otázku 4.7

Preprocesor předchází překladač, zpracovává vstupní text, provádí v něm textové změny a jeho výstupem je opět text. Neprovádí tedy žádnou kontrolu syntaxe, nebo typovou kontrolu. Preprocesor zpracovává hlavičkové soubory, odbourává komentáře, umožňuje provádět podmíněný překlad zdrojového textu a rozvíjí tzv. makra. Jeho direktivy zvyšují přenositelnost kódu a zpřehledňují program.

Odpověď na kontrolní otázku 4.8

Každá funkce má svou definici a základních vlastností: je pojmenována a tímto jménem se volá v programu, může, ale nemusí mít parametry, jejichž prostřednictvím předáváme data do těla funkce ve které jsou následně zpracovány, má návratovou hodnotu obsahující výsledky zpracování v těle funkce.



Klíč k řešení – kapitola 5

Odpověď na kontrolní otázku 5.1

Sběrnice mezi CPU a severním mostem je FSB. Od této sběrnice se odvíjí frekvence CPU, AGP, paměti, samotných mostů čipsetu atd.

Odpověď na kontrolní otázku 5.2

U prvních PC vyráběných firmou IBM obsahovaly základní desky kromě procesoru mnoho dalších čipů, které z hlediska funkce představují čipovou sadu. Např: generátor hodin, řadič sběrnice, systémový časovač, řadič přerušení, řadič DMA, řadič klávesnice, atd., celkem až 100 obvodů. Po roce 1993 se čipové sady ubíraly nastoleným směrem, jenž definoval sady založené na architektuře North/South Bridge. Novější čipové sady jsou založené na rozbočovačích, přičemž North Bridge je nazýván rozbočovačem řadiče paměti MCH – Memory Controller Hub a čip South Bridge je označován pojmem rozbočovač řadiče vstupů a výstupů ICH – I/O Controller Hub.

Odpověď na kontrolní otázku 5.3

Čip North Bridge je nejdůležitější součástí základní desky a jediný obvod pracující na plné rychlosti sběrnice procesoru. Čip South Bridge byl odvozen od úkolu propojit sběrnice PCI s pomalejší sběrnicí ISA. Architektura využívá ještě dalšího čipu s názvem Super/IO čip, ten je oddělený a obvykle ho dodává jiný výrobce (např. National Semiconductor). Slouží k připojení běžných periférií k systému.

Od roku 1999 si nároky na rychlosti procesorových sběrnic a spojů mezi jednotlivými mosty vyžádaly změnu technologie. Oba nejdůležitější čipy, tedy rozbočovače MCH a ICH byly nyní propojeny sběrnicí s nižší šířkou, tedy menším počtem vodičů, což mělo pozitivní vliv na zrychlení technologie s menším rušením, signálovým šumem a počtem směřování vodičů na základní desce. MCH - Memory Controller Hub (dříve severní most) má na starost propojení čipové sady s procesorovou sběrnicí a

grafickou sběrnici. ICH - I/O Controller Hub (dříve jižní most) tvoří rozhraní mezi rozbočovačem, IDE řadičem pevných disků a sběrnici PCI a USB.

Odpověď na kontrolní otázku 5.4

DDR – *Double Data Rate*: dvojnásobná propustnost po sběrnici.

QDR – *Quad Data Rate*: čtyřnásobná propustnost 400, 533, 800, (1066 MHz).

SMP - *Symmetric Multiprocessing*, tedy podpora dvou procesorů.

MCH – *Memory Controller Hub*.

ICH – *I/O Controller Hub*.

DVI – *Digital Visual Interface* je rozhraní (tzv. dedikovaný spoj) pro propojení video zařízení s počítačem.

FSB - (*Front Side Bus*) nebo *System Bus* je fyzická obousměrná datová sběrnice, která přenáší veškeré informace mezi CPU (*Central Processing Unit*) a ostatními zařízeními uvnitř systému.

AGP - *Accelerated Graphics Port* (též *Advanced Graphics Port*), je vysokorychlostní „point-to-point“ kanál (nejde tedy o sběrnici v pravém slova smyslu) pro připojení grafické karty k základní desce počítače.

ECC kód - samoopravný kód pro detekci a opravu chyby (error-correcting code).

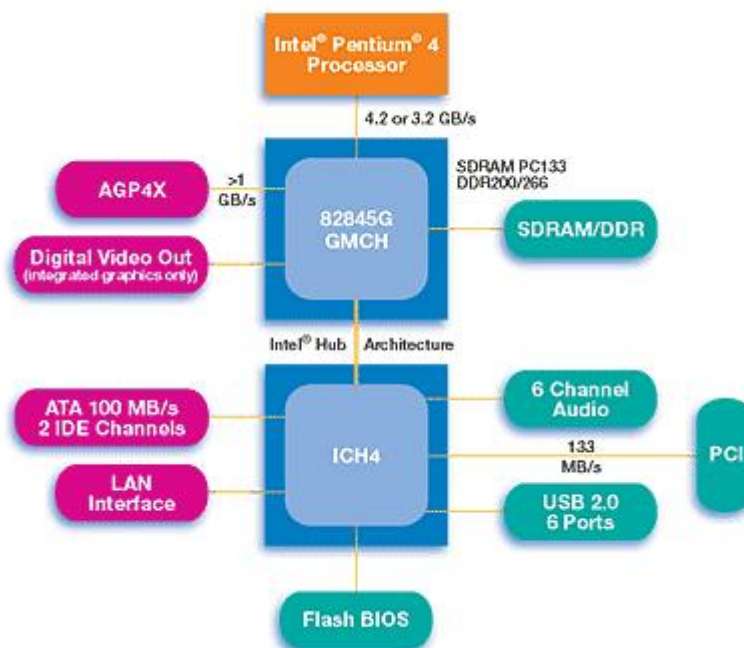
Odpověď na kontrolní otázku 5.5

Při výběru základní desky dnes rozhodují mnohé kritéria. Před samotným výběrem konkrétního typu někdy dochází k volbě form faktoru. Ne všichni uživatelé kupují standardní formát do provedení desktop či tower. Dále pak obvykle postupuje podle těchto bodů:

1. Procesor Intel vs. AMD, základní deska pro tyto procesory má odlišnou patici.
2. Podporovaná rychlost a typ paměti.
3. Integrovaná grafika vs. Grafický akcelerátor.
4. Počet dalších volných slotů (PCI, PCI-e).
5. Speciální vlastnosti základní desky.

Odpověď na kontrolní otázku 5.6

Schéma čipové sady můžeme zobrazit například takto:



Obr. 15.1 Schéma čipové sady



Klíč k řešení – kapitola 6

Odpověď na kontrolní otázku 6.1

PGA (*Pin Grid Array*) – používá se pro integrované obvody s velmi vysokým počtem součástek i vývodů. PGA je navrženo tak, aby mohlo být osazeno (a případně vyjmuto) z patice. Pouzdra vycházející z tohoto typu s různými obměnami a vylepšeními používá většina procesorů.

Pouzdro SPGA (*Staggered Pin Grid Array*) využilo mezer mezi pravidelnými sloupcemi a řadami pinů pouzdra pro umístění dalších vývodů.

DIP (*Dual In-line Package*) – stále se používá pouze pro malé stupně integrace a má dvojnásobný počet vývodů než SIP,

PPGA (*Plastic Pin Grid Array*) - varianty procesorů Celeron využívaly levnější plastické varianty pouzder.

SECC pouzdro typu Single Edge Contact Cartridge.

SEP (*Single Edge Processor*) typ osazení procesoru bez pouzdra pro levnější varianty Celeron.

SIMM - Single Inline Memory Module.

BGA (*Ball Grid Array*) - Pro mobilní procesory a také paměťové čipy. Místo klasických nožiček má vývody ve tvaru malých kuliček, a je tak uzpůsoben k přímé montáži na povrch plošného spoje.

LGA (*Land Grid Array*). Poslední variantou využívanou u současných procesorů INTEL.

Odpověď na kontrolní otázku 6.2

Prostřednictvím paměti cache, která pracuje na rychlosti procesoru, neboť je implementována v jeho jádře.

Odpověď na kontrolní otázku 6.3

Velikost datové sběrnice přímo nesouvisí s velikostí adresové sběrnice, návrháři procesorů mohou použít v podstatě libovolnou šířku. Avšak obecně platí, že čím má procesor větší šířku datové sběrnice, má i větší šířku adresové.

Odpověď na kontrolní otázku 6.4

- Vytěžený křemík je zbaven příměsí, roztaven, destilován a zformován do tyčí o čistotě 99,999% vhodných k výrobě polovodičů. Tyče jsou rozlámány na kusy a vloženy do elektrických pecí umožňujících tažení krystalů.
- Křemík je zahřát na teplotu 1370 stupňů.
- Jakmile je křemík roztaven, vloží se do něj malý zárodečný krystal, postupným tvářením se dosáhne toho, že se krystal rozšíří do požadovaného průměru.
- Vznikne křemíkový ingot o průměru 20 – 30 cm, délka ingotu je 1,5 metru, hmotnost stovky kg.
- Výsledný ingot je diamantovou pilou rozřezán na více než 1000 plátků o tloušťce 1 mm, každý plátek je pak vyleštěn.
- Následuje fotolitografie – proces výroby: vytvoření vrstvy oxidu křemičitého, vytvoření polovodičové vrstvy, nakonec metalizační vrstvy – spoje mezi tranzistory. (hliníková metalizační vrstva je nově nahrazena mědí (menší odpor).

Odpověď na kontrolní otázku 6.5

Běžné instrukce x86 a x87 jsou schopny pomocí jedné instrukce spočítat pouze jednu informaci (data). Dalším krokem vývoje a zrychlování práce procesoru umožnil typ instrukcí SIMD (*Single Instruction Multiple Data*). Jedna SIMD instrukce provádí výpočet na více datech současně. První SIMD sadou byla instrukční sada MMX (*MultiMedia eXtensions*). Tato sada byla opět celočíselnou sadou. Dnes ji podporují prakticky všechny procesory a je často vyžadována různými programy.

Instrukční sada uvedena v roce 1999 nesla název SSE (*Streaming SIMD Extensions*). Sada se ihned stala součástí procesorů Pentium III. Umožňuje práci s desetinnými čísly, jenž ale vyžadovala v procesoru speciální přídavnou jednotku, ta standardní stále využívala dle potřeby instrukce MMX. Alternativou firmy AMD k instrukční sadě SSE byla sada 3DNow! a později Enhanced 3DNow! Jedná se v podstatě o rozšíření MMX a SSE o nové instrukce, ty ale nejsou kompatibilní s instrukcemi SSE. V roce 2000 se sada SSE dočkala velkého rozšíření SSE zavedeného pro procesory Pentium 4. Jedná se o sadu SSE2, která obsahuje všechny instrukce MMX a SSE a navíc dalších až 144 instrukcí. Dalším rozšířením, již ne tak rozsáhlým, byla instrukční sada SSE3. Jedná se o rozšíření o 13 instrukcí oproti SSE2. Pohled do blízké budoucnosti odkrývá 45 nm technologii s jádrem Nehalem, což s sebou přinese novou instrukční sadu NNI neboli SSE4. AMD v převaze svých 64 bitových procesů oproti Intelu zavádí sadu AMD64. Jedná se o 64bitové rozšíření x86, původně vyvíjené pod označením x86-64. Intel reagoval na tuto sadu vypuštěním EM64T.

Odpověď na kontrolní otázku 6.6

Procesory mohou být posuzovány podle dvou hlavních kritérií: podle šířky sběrnic a podle jejich rychlosti. Pojem šířka sběrnice se používá k popisu tří různých parametrů CPU:

- šířka vnitřních registrů,
- šířka vstupních a výstupních sběrnic pro data,
- šířka adresové sběrnice.

Šířka datové sběrnice určuje počet datových bitů, které mohou být v průběhu jednoho cyklu přesunuty z/do procesoru. Šířka adresové sběrnice určuje maximální velikost RAM, kterou je schopen procesor využít.

Odpověď na kontrolní otázku 6.7

Chladiče jsou hodnoceny podle jejich schopnosti chladit, vyjádřené odolnosti vůči přenosu tepla. Jednotka je °C/W a platí, čím je hodnota nižší, tím je chladič lepší. Aktivní chladiče vyžadují vlastní napájení. Materiálem používaným pro konstrukci chladičů je převážně měď kombinovaná s hliníkem. Pro vysokou hustotu vyzářeného tepla má většina moderních chladičů alespoň měděné jádro. Dnešními charakteristickými vlastnostmi chladičů mimo možnost nasazení na konkrétní typ soketu jsou hmotnost, rozměr, hlučnost a počet otáček. Tyto parametry se pohybují dle ceny a kvality konkrétního chladiče takto: hmotnost 300 – 950 gramů, 1000 – 3000 ot./min a hlučnost 15 – 40 dB.

Kromě běžného chlazení vzduchem se v oblasti počítačů můžete setkat i s dalšími metodami. K těm běžnějším a také komerčně dostupným patří hlavně vodní chlazení, které je založeno na vodním okruhu, který odvádí teplo od procesoru protékáním speciálním blokem.



Klíč k řešení – kapitola 7

Odpověď na kontrolní otázku 7.1

Pentium I obsahuje dvě větve pro zpracování instrukcí, tzv. superskalární technologie. Jedna z větví pro vykonávání instrukcí se nazývá *u* větev a druhá *v* větev. Primární větví je větev *u*, která je schopna vykonávat všechny instrukce jak s celými čísly, tak s desetinnými. Naopak *v* větev vykonává některé instrukce s celými a jednoduché instrukce s desetinnými čísly.

Proces provádění dvou různých instrukcí současně ve dvou větvích se nazývá párování. Ne všechny instrukce mohou být párovány, pokud k takové situaci dojde, je použita pouze *u* větev.

Procesor také obsahuje zásobník BTB (*Branch Target Buffer*), který umožňuje předpovídat větvení. Jeho účelem je zabránit zpoždění ve vykonávání instrukcí v kterékoliv větvi, jenž by mohlo nastat v okamžiku načítání instrukcí ukazující na nesouvisející místa v paměti. BTB se pokouší takové větvení předpovědět a dopředu načíst potřebné instrukce.

Pentium má 32bitovou adresovou a 64bitovou datovou sběrnici. CPU má dvě oddělené paměti cache, každou o velikosti 8 kB. Obvody řadiče cache a samotné cache jsou integrovány přímo do čipu. Obě

paměti cache jsou rozděleny do dvou asociativních sad, každá z nich je rozdělena na řádky po 32 bytech. Současně každá cache má svůj zásobník pro překlad adres, nazvaný TLB (*Translation Lookaside Buffer*.) L2 cache o velikosti 512 kB je možno přidat na základní desku PC.

Odpověď na kontrolní otázku 7.2

Procesory Pentium I byly taktovány do 266 MHz. U procesoru Pentium II byla velikost paměti zvětšena na 512 kB. Procesor se vyráběl v těchto verzích taktovacích frekvencí: 233 – 266 – 300 – 333 – 350 – 400 a 450 MHz. Procesory Pentium III jsou taktovány od 400 po 1400 MHz s jádry Katmai (250 nm, 450-900 MHz), Coppermine (180 nm, 533-1133 MHz) a Tualatin (130 nm, 1000 – 1400 MHz). Procesory Pentium 4 s prvním jádrem Willamette byl taktován do 2 GHz, jádro Northwood do 3 GHz a další revize Prescott až 3,8 GHz. Touto mezí bylo ukončeno zrychlování frekvencí a technologie se začala ubírat jiným směrem.

Odpověď na kontrolní otázku 7.3

Hlavní rysy architektury NetBurst:

Hyper řetězová technologie (*Hyper Pipelined Technology*) - procesor P4 zdvojnásobuje hloubku řetězení do dvaceti úrovní, čímž výrazně zvýšil výkonnost a frekvenční možnosti procesoru.

- *Rapid Execution Engine* označuje fakt, že aritmetické logické jednotky procesoru (ALUs) pracují na dvojnásobné frekvenci jádra, což umožňuje provádět určité instrukce v polovičním taktu oproti jádrovému.

- Systémová sběrnice je taktována na 400 MHz. Má 128-bitové linky s 64-bitovými přístupy (32-bitové linky u předchozí generace).

- *Execution Trace Cache* - jedná se o vyspělou instrukční vyrovnávací paměť úrovně Level 1, která vyrovnává dekodované instrukce, čímž odstraňuje čekací dobu dekodéru u hlavního prováděcího cyklu.

Vyspělé dynamické provádění (*Advanced Dynamic Execution*)- Procesor Pentium 4 má extrémně účinný poruchový spekulativní prováděcí engine, který trvale zatěžuje výkonné jednotky. Nová je také zvýšená schopnost predikce větvení, která udržuje činnost procesoru ve správném programovém toku a snižuje chybu špatné predikce spojenou s hlubším řetězením.

- *Streaming SIMD Extension 2* - SSE2 rozšiřuje technologie MMX a SSE přidáním 144 nových instrukcí, včetně 128-bitových SIMD celočíselných aritmetických a 128-bitových SIMD instrukcí s pohyblivou desetinnou čárkou o dvojnásobné přesnosti, které zvyšují výkon v celé šíři spektra aplikací.

Odpověď na kontrolní otázku 7.4

Procesory rozkládají instrukce do několika fází. Jednotlivé fáze jsou součástí takzvané pipeline, každou instrukci je třeba načíst z vyrovnávací paměti, dekodovat do formy které rozumí procesor, dále pak instrukci vykonat a zapsat výsledek zpět do paměti. Fáze pipeline mohou mít různou délku. Jednotlivé procesory se liší délkou své instrukční pipeline. Procesor Pentium 4 s jádrem Prescott má dokonce 31 stupňovou instrukční pipeline. Předchozí verze procesoru P4 měli délku pipeline 20 stupňů.

Odpověď na kontrolní otázku 7.5

Architektura Intel Core sebou přináší řadu revolučních novinek, například dekodování až pěti instrukcí za takt. Instrukce, ve kterých jsou napsány programy, jsou poskytnuty z paměti cache a dekodovány na velmi malé mikroinstrukce. To se děje proto, aby jednotlivé mikroinstrukce byly paralelně prováděny ve výpočetních jednotkách. Intel Core má k dispozici čtyři instrukční dekodéry. Tři jsou určeny pro jednoduché instrukce, které se dekodují do jedné mikroinstrukce. Jeden dekodér je určen pro komplexní instrukce, které se přetvoří do až čtyřech mikroinstrukcí. Celkově tak dekodování umí dodat do bufferu až sedm microOPs za takt. Novinkou je, že jednoduché dekodéry nyní umí zpracovat i instrukce, které dříve musely procházet dekodérem komplexním. Jedná se především o instrukce pracující s pamětí a SSE instrukce.

Hlavní novinkou je tzv. fúzování makroinstrukcí, to znamená, že dojde k dekódování programu tak, aby se dvě instrukce dekodovaly na jednu mikroinstrukci, která však zastoupí obě dvě! Takto sloučené instrukce pak převezme jeden dekodér a vytvoří z nich jednu microOP. Za jeden hodinový cyklus je možné takto sloučit jeden pár instrukcí a toto sloučení může provést libovolný ze čtyřech dekodérů.

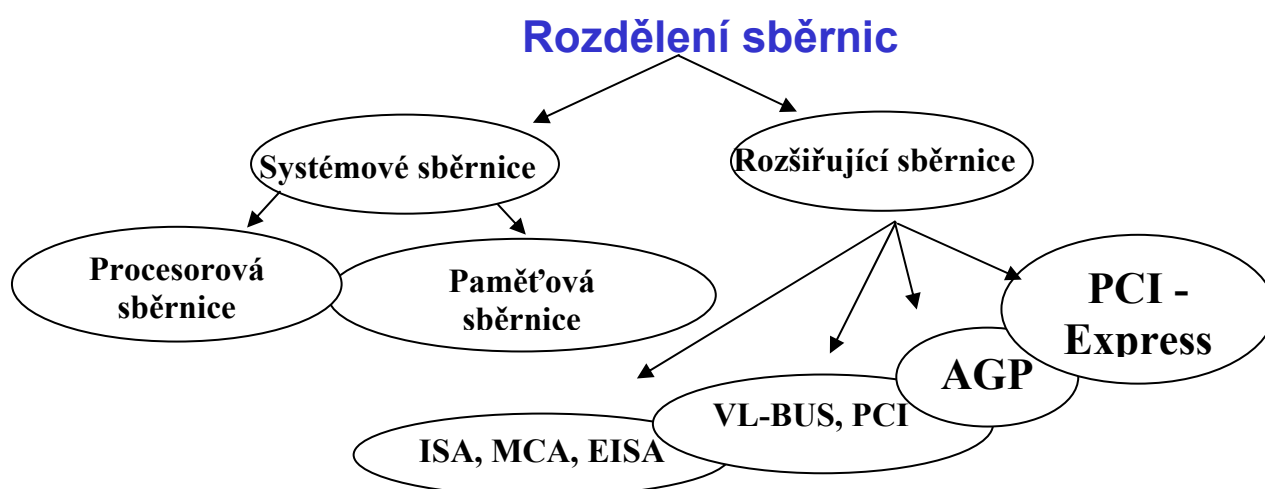


Klíč k řešení – kapitola 8

Odpověď na kontrolní otázku 8.1

Mezi základní parametry každé sběrnice patří: šířka přenosu, frekvence a propustnost. Šířka přenosu určuje počet bitů, které lze přenést po sběrnici v jeden okamžik, propustnost určuje počet bajtů přenesených za jednotku času.

Odpověď na kontrolní otázku 8.2



Obr. 15.2 Rozdělení sběrnic

Odpověď na kontrolní otázku 8.3

Sběrnice typu ISA, MCA a EISA mají společný rys: poměrně malou rychlost. Omezení rychlosti je dědictvím z dob, kdy I/O sběrnice pracovaly na stejné rychlosti jako procesorová sběrnice. PCI již není klasickou lokální sběrnici jako VL bus, ale je k systémové sběrnici připojena přes tzv. mezisběrníkový můstek. Toto řešení s sebou nese následující výhody jako možnost použití sběrnice PCI i v jiných počítačích než jsou PC (např. Macintosh, DEC) a možnost provádět přizpůsobování napěťových úrovní. Sběrnice PCI má standardně šířku přenosu dat 32 bitů. Existuje však i verze PCI, která je 64bitová. V roce 1996 se na trhu objevil slot AGP (*Accelerated Graphics Port*) taktovaný základní frekvencí 66 MHz. Specifikace AGP 1.0 podporuje verze multiplikátoru 1x a 2x na signálech o velikosti napětí 3.3 V. Specifikace AGP 2.0 podporuje již verzi AGP 4x a předchozí verze, avšak na napěťové úrovni 1,5 V. Poslední specifikace AGP 3.0 již podporuje pouze multiplikátory 4x a 8x na napětí 0,8 V.

Odpověď na kontrolní otázku 8.4

V roce 1992 byl poprvé uveřejněn standard sběrnice PCI (*Peripheral Component Interconnect*). Tento standard byl postupně upravován, v roce 1993 na verzi PCI 2.0, která definuje již rozměry rozšiřujících karet. Verze 2.1 v roce 1995 definuje provoz na dvojnásobné frekvenci a standard 2.2 v roce 1999 se zabývá mechanickými částmi. Sběrnice se neustále vyvíjí a vycházejí z ní i další standardy PCI-X, Mini-PCI a také sériová nejnovější sběrnice PCI Express, definovaná od roku 2002. PCI již není klasickou lokální sběrnici jako VL bus, ale je k systémové sběrnici připojena přes tzv. mezisběrníkový můstek. Sběrnice PCI má standardně šířku přenosu dat 32 bitů. Existuje však i verze

PCI, která je 64bitová. Tato rozšířená verze se používá zejména pro řadiče diskových polí. Maximální frekvence, se kterou může tato sběrnice standardně pracovat, je 33 MHz, čímž je zajištěna propustnost sběrnice 132 MB/s (32 bitů) nebo 264 MB/s (64 bitů). Kromě sběrnice PCI, která pracuje s rychlostí 33 MHz, je vyráběna i rychlejší verze, která umožňuje pracovat při frekvenci 66 MHz.

Podobně jako MCA a EISA a narozdíl od VL busu podporuje PCI busmastering, což umožňuje vyšší výkon počítače, protože přenosy řízené CPU vyžadují spoustu času. PCI dále podporuje standard Plug & Play vyvinutý výrobcí hardwaru téhož roku 1992, ten dovoluje velmi snadnou konfiguraci desek pro PCI, popř. jejich automatickou konfiguraci bez zásahu uživatele. Sběrnice PCI je používána u počítačů s procesory Pentium a vyššími.

Odpověď na kontrolní otázku 8.5

AGP (*Accelerated Graphics Port*) taktovaný základní frekvencí 66 MHz. Hlavním důvodem přechodu mimo sběrnici PCI nebylo původně další navyšování grafického výkonu, ale zajištění přímého přístupu grafické kartě do paměti a zrychlení komunikace s procesorem. V průběhu doby se objevilo několik typů AGP seřazených v tabulce. Index za AGP udává, kolik dat je možné přenést při jednom taktu sběrnice. V portu AGP není možné provozovat všechny typy grafických karet, známým faktem je nekompatibilita grafických karet AGP 1x a 2x s novými sloty AGP 4x a 8x.

Typ AGP	Rok uvedení standardu	Napětí na sběrnici	Propustnost
AGP 1x	1996	3,3 V	266 MB/s
AGP 2x	1996	3,3 V	533 MB/s
AGP 4x	1998	1,5 V	1066 MB/s
AGP 8x	2000	0,8 V	2133 MB/s

Tab. 15.1 Propustnost jednotlivých verzí AGP

Odpověď na kontrolní otázku 8.6

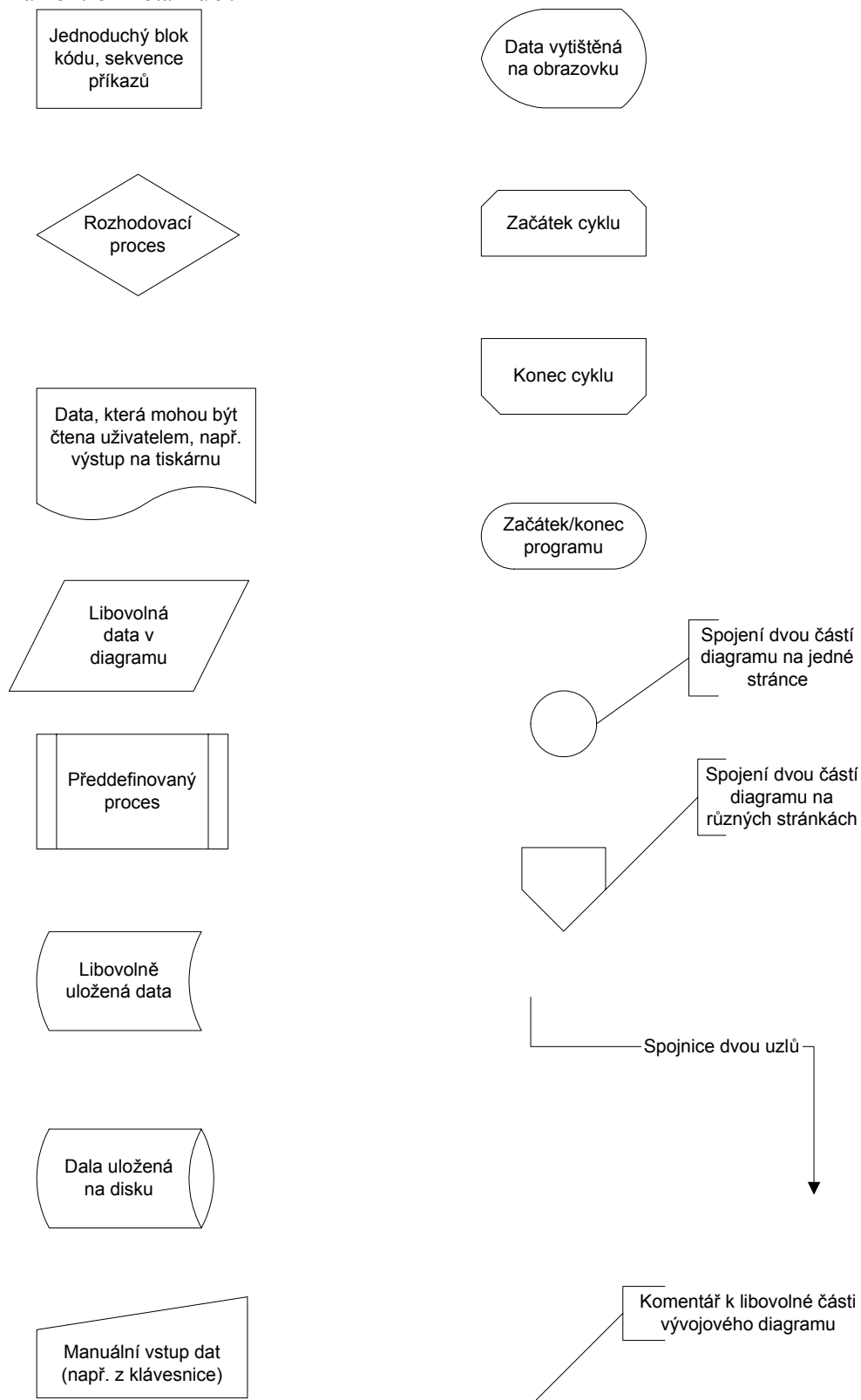
Sběrnice PCI Express se významně odlišuje od svých předchůdců, protože mění celkovou filozofii pohledu na sběrnice tak, jak je známe do dnešních dnů. Obvyklá představa sběrnice vycházela z propojení několika zařízení sdílející stejnou skupinou vodičů (paralelní nebo sériové). Některé typy sběrnic byly doplněny o vybrané vodiče sloužící pro dekódování daného zařízení, informaci o jeho stavu a synchronizaci hodinovým signálem. Model fyzické vrstvy sběrnice PCI Express vychází spíše ze síťové architektury než z architektury PCI nebo PCI-X, podobnost architektury PCI Express najdeme především v dělení vrstev, což připomíná síťový model ISO-OSI.

Sběrnici PCI Express v konkrétním počítači tedy můžeme vnímat jako soustavu několika nezávislých obousměrných propojení typu 1:1, které dokážou data přenášet tak, jako by tyto sběrnice byly široké 32 bitů. V základní verzi máme k dispozici sběrnici o propustnosti 250 MB/s v každém směru (PCI Express x1), což je prakticky dvojnásobná propustnost sběrnice PCI.



Klíč k řešení – kapitola 9

Odpověď na kontrolní otázku 9.1



Obr. 15.3 Bloky vývojových diagramů

Odpověď na kontrolní otázku 9.2

TxD	Transmit Data – signál se používá pro vysílání dat
RxD	Receive Data – signál se používá pro příjem dat
RTS	Request to Send – DTE zařízení signalizuje tímto vodičem, že chce začít vysílat data.
CTS	Clear to Send – DCE zařízení signalizuje tímto vodičem připravenost k vyslání dat. Signál se také používá jako odpověď na RTS.
DTR	Data Terminal Ready – signál je používán DTE zařízením k oznámení, že zařízení je připojeno a připraveno začít komunikaci.
DSR	Data Set Ready – jde o protisignál k DTR a je používán DCE zařízením k oznámení, že je připraveno ke komunikaci.
DCD	Data Carrier Detect – signál je využíván DCE zařízením a indikuje DTE zařízení, modem přijal „nosný signál“ a očekává se příjem dat.
RI	Ring Indicator – signálem DCE zařízení oznamuje DTE zařízení, že zvoní telefon připojený k modemu (DCE zařízení).
GND	Signal Ground – signálová zem. Vodič musí být vždy mezi zařízeními propojen.

Tab. 15.2 definované signály rozhraní RS232

Odpověď na kontrolní otázku 9.3

Původně měl být paralelní port jednosměrný. Později byly vyvinuty porty pro obousměrnou komunikaci, jak vysílání dat z PC, tak i jejich příjem. Paralelní port byl tedy využit i pro vzájemné propojení počítačů. Standard, který definuje fyzické vlastnosti paralelního rozhraní se jmenuje IEEE-1284 a jeho konečná verze byla schválena v roce 1994. Z pohledu komunikace definuje standard pět režimů. Kombinace těchto režimů určuje typ paralelního portu. Standardní paralelní port (SPP) byl původně určen pouze pro vysílání dat. Později byl však upraven aby byl data schopen i přijímat. Součástí počítačů IBM PS/2 již byl oboustranný paralelní port. Směr přenosu dat určoval nově zavedený stavový bit. V roce 1991 byl vyvinut standard EPP (*Enhanced Parallel port*) firmami Intel, Xircom a Zenith Data Systems. O rok později byl ustanoven standard ECP (*Enhanced Capabilities port*) firmami Microsoft a HP.

Odpověď na kontrolní otázku 9.4

Sériové rozhraní je realizováno speciálním obvodem UART (*universal asynchronous receiver transmitter*) (dříve 8252, dnes 16750). Jedním z nejdůležitějších úkolů obvodu je převod paralelních dat, nejčastěji 8-bitového bajtu, na sériový tok bitů a naopak.

Odpověď na kontrolní otázku 9.5

Typ sběrnice	Přenosová rychlost MB/s
ISA 8bitová	4,17
ISA 16bitová	8,33
Řadič disk. mech.	0,0625
EISA	33
VLBUS	133
PCI	133
PCI 66 MHz	266
PCI 64bitová	533

AGP	266
AGP 2x	533
AGP 4x	1066
AGP 8x	2133
RS232	0,01152
Paralelní 4bitový Nibble	0,050
Paralelní 8bitový	0,150
Paralelní kompatibilní	0,150
Paralelní EPP	0,5 – 2
Paralelní ECP	0,5 – 2
USB 1.1	1,5
USB 2.0	60
IEEE1394 200	25
IEEE1394 400	50
IEEE1394 800	100
Ultra ATA 33	33
Ultra ATA 66	66
Ultra ATA 100	100
Ultra ATA 133	133
SATA 150	150
SATA 300	300
SCSI	5
SCSI Ultra	20
SCSI Ultra 2	40
SCSI Ultra 3 (160)	80 (160)
Paměť FPM	177
Paměť EDO	266
Paměť SDRAM PC100	800
Paměť DDR PC3200	3200
Paměť RDRAM 8500	8533
33 MHz FSB 486	133
66 MHz FSB P I	533
100 MHz FSB PII	800
400 MHz FSB P4	3200
533 MHz FSB P 4	4266

Tab. 15.3 srovnání rychlostí rozhraní a sběrnic

Odpověď na kontrolní otázku 9.6

První verze standardu byla uvedena v roce 1996. Ihned následovala modifikace 1.1. Tento standard podporuje přenosovou rychlost 12 Mb/s, což převedeno na rychlost v bytech odpovídá 1,5 MB/s. V roce 2000 byl uveden standard USB 2.0, který je 40 krát rychlejší než jeho předchůdce. Sběrnice umožňuje připojit až 127 zařízení, přičemž je založena na hvězdicové topologii. Základem je USB hub neboli rozbočovač, který je buďto interní součástí počítače anebo může figurovat jako externí zařízení. Z fyzikálního hlediska přenosu jsou data přenášena po sběrnici metodou NRZI (*Non Return Zero Invert*). To znamená, že u jednotlivých bitů 0 a 1 reprezentovaných opačnou hodnotou napětí nedochází k žádnému návratu k nulovému napětí. Jednička je tedy přenášena bez jakékoliv změny napětí a nula je prezentována změnou napětí.



Klíč k řešení – kapitola 10

Odpověď na kontrolní otázku 10.1

Existuje mnoho hledisek dělení pamětí, např. podle přístupu, rychlosti, podle hardwarového provedení paměťových modulů a v neposlední řadě také podle schopnosti zápisu. Paměti můžeme tedy rozdělit do několika skupin, počítačové paměti si rozdělíme takto:

- vnitřní
 - ROM (PROM, EPROM, EEPROM, Flash),
 - RAM (DRAM – dynamic RAM, SRAM – static RAM),
- vnější (magnetické, magneticko optické, optické, polovodičové).

Odpověď na kontrolní otázku 10.2

Zakreslené buňky paměti najdete přímo v učebním textu, není nutné je zde znovu uvádět.

Odpověď na kontrolní otázku 10.3

DRAM pamětí existuje několik základních typů vývojově navazujících:

FPM RAM - U tohoto typu paměti je využit proces nazývaný stránkování (paging).

EDO RAM - (*Extended Data Out*). Poprvé se tyto paměti objevily v roce 1995. Vychází z modulů FPM. Princip prokládání u těchto modulů spočívá opět v principu načítání. Zatímco řadič paměti načítá data z určité buňky začíná zároveň vyhledávat buňku následující.

BEDO RAM - (*Burst Extended Data Output RAM*). Tyto paměti čtou data v dávkách a mají vlastní adresový čítač. Nejsou vhodné pro rychlosti větší než 66 MHz. Využívaly modulů SIMM a jejich neuplatnění spočívalo v nasazení již prvních SDRAM modulů na trh.

SDRAM - (*Synchronous Dynamic RAM*). Tyto paměti využívají pro svou práci modulů DIMM. Paměti SDRAM jsou synchronní, tzn. vstupní a výstupní signály pracují synchronizovaně s kmitočtem základní desky.

DDR SDRAM - (*Dual Data Rate SDRAM*). Tyto paměti dokáží přenášet dvě informace v jednu cyklu čímž se zvyšuje jejich rychlost.

DDR2 SDRAM - Jedná se o novější standard vycházející z pamětí DDR SDRAM, poskytují dvojnásobnou přenosovou rychlost oproti DDR SDRAM. Data jsou čtena (zapisována) s nástupnou i sestupnou hranou hodinového signálu obdobně jako u DDR SDRAM. Poskytují dvojnásobnou přenosovou rychlost oproti DDR SDRAM a mají asi o 50% menší spotřebu el. energie.

Odpověď na kontrolní otázku 10.4

Po vybrání správného banku paměti je nezbytné vyhledat příslušný řádek. Řadič paměti vyvolá stav paměti Active příkazem RAS – *Row Address Strobe*. Tato operace zabere více hodinových cyklů a po jejich dokončení bude řádek paměti vyhledán. Poté dochází k časové prodlevě mezi hledáním řádku a sloupce (RAS to CAS Delay). Následuje vyhledání sloupce vyžadující čas TCL neboli *Column Address Strobe Latency Time*, na který je kladen důraz ve specifikacích paměti. Po tomto časovém úseku jsou již data přečtena.

Odpověď na kontrolní otázku 10.5

SIPP (*Single In-Line Pin Package*). Jedná se o moduly podobné modulům 30-pin SIMM.

SIMM (*Single In Line Memory Module*). Tyto moduly byly ve svých starších verzích 30 pinové. Později je nahradily moduly, které měly pinů 72.

DIMM (*Dual In Line Memory Module*). Tento modul má již 168 vývodů, datová šířka je 64 bitů bez kontroly parity nebo 72 bitů s paritní kontrolou či podporou ECC.

DDR SDRAM (*Double-Data-Rate Synchronous Dynamic Random Access Memory*) je typ pamětí, který vývojově následoval klasický paměťový modul DIMM SDRAM. Dosahuje vyššího výkonu než předchozí typ SDRAM tím, že přenáší data na obou koncích hodinového signálu. Tento přístup

zvyšuje efektivní výkon téměř dvakrát bez nutnosti zvyšování frekvence sběrnice. Tento modul má 184 pinů a jeho maximální kapacita dosáhla 1024 MB.

Modul RIMM se díky své ceně neprosadil a byl zcela vytlačen moduly DDR SDRAM. Každý modul RIMM má 184 pinů symetricky rozdělených na obě strany.

Odpověď na kontrolní otázku 10.6

Moduly s kontrolou parity detekují logické chyby pomocí paritního bitu. Ten je kontrolou osmi po sobě jdoucích bitů, tedy jednoho bytu. Kontrola parity na chybu upozorní, nikoliv opravuje. Kód ECC (*Error Corecting Code*) je naopak schopen následné opravy jednoho bitu. Umožňuje i detekci dvou chybných bitů, avšak bez opravy. Je však ověřeno že s 98% jsou logické chyby tvořeny pouze chybou jednoho bitu.



Klíč k řešení – kapitola 11

Odpověď na kontrolní otázku 11.1

První ve větší míře používané rozhraní pevných disků pro počítače PC bylo rozhraní ST506, vyrobeno firmou Shugart Technologies (dnešní Seagate). Toto rozhraní dokázalo pracovat s disky, které měly maximálně 16 hlav a bylo možné k němu připojit maximálně dva disky. Začátkem 80. let vzniká jako snaha o standardní rozhraní pro připojování periférií rozhraní ESDI (Enhanced Small Device Interface), které by nahradilo rozhraní ST506. Jedná se o výrazně zlepšené rozhraní ST506, u kterého jsou data přenášena sériově a řídicí informace paralelně. Ke zlepšení oproti svému předchůdci patří: podpora disků, které mohou mít až 256 hlav, dovoluje podstatně vyšší přenosovou rychlost dat a to až 24 Mb/s. V roce 1986 bylo navrženo firmami Western Digital a Compaq rozhraní IDE (Integrated Device Electronics). Zkratka IDE je poměrně obecná a označuje kteroukoliv mechaniku s vestavěným řadičem, rozhraní IDE se dnes častěji označuje jako ATA (AT Attachment). Rozhraní ATA bylo standardizováno postupně v 7 verzích, poté následoval přechod na seriové rozhraní SATA. Odděleně od tohoto rozhraní bylo vyvíjeno rozhraní SCSI. První specifikace SCSI-1 byla uveřejněna v roce 1986. Rozhraní je neustále vyvíjeno a verze SPI-5 disponuje přenosovou rychlostí 640 MB/s.

Odpověď na kontrolní otázku 11.2

Přenos dat mezi zařízením (HDD, CD-ROM, atd.) a operační pamětí může být realizován pomocí dvou různých režimů PIO a DMA. U režimu PIO - Programmed (Processor) Input Output je přenos dat řízen procesorem. Ve srovnání s režimem DMA (*Direct Memory Access*) generuje větší zátěž procesoru neboť veškerá přenášená data procházejí přes procesor. DMA režim využívá speciální obvod (*DMA controller*), který je schopen zprostředkovat přenos dat bez účasti procesoru.

Odpověď na kontrolní otázku 11.3

Rozhraní ATA 1 bylo používáno od roku 1986, ale standardizováno až v roce 1994. Omezení kapacity pevného disku u prvního rozhraní je stanoveno na 512 MB. Rozhraní ATA 2 disponuje přenosovou rychlostí 16,67 MB/s a umožňuje připojit až čtyři zařízení. V roce 1997 bylo standardizováno rozhraní ATA 3, o rok později přišlo ATA 4. Rozhraní ATA 3 přináší pouze drobná vylepšení (oproti rozhraní ATA-2), bývá rovněž označováno termínem EIDE. Hlavní novinkou je přidání podpory pro technologii S.M.A.R.T. ATA 4 bylo označováno také jako ATA/ATAPI-4 nebo Ultra-ATA 33. Součástí rozhraní se stává i dříve používaný (samostatný) standard ATAPI. Rozhraní ATA 4 definuje nový propojovací kabel s 80 vodiči. V roce 1999 byl zveřejněn standard ATA 5, který je nazýván také ATA/ATAPI-5 nebo Ultra-ATA 66. Přináší podporu nových přenosových režimů Ultra-DMA 3 a Ultra-DMA 4. Režim Ultra-DMA 4 je podle své přenosové rychlosti (66 MB/s) také označován jako Ultra-DMA 66 nebo Ultra-ATA 66. Rozhraní dovoluje automatickou detekci kabelu se 40, resp. 80 vodiči. Provoz režimů s vyšší přenosovou rychlostí než 33 MB/s (Ultra-DMA 3 a Ultra-DMA 4) již vyžaduje použití nového 80vodičového kabelu. Rozhraní ATA 6 bylo zveřejněno v roce 2000 a ještě po dlouhé dva roky upravováno, než bylo definitivně standardizováno. Bylo nazýváno také jako ATA/ATAPI-6 nebo Ultra-ATA 100. Podporuje nový přenosový režim Ultra-DMA 5 (Ultra-DMA 100, Ultra-ATA 100) s maximální přenosovou rychlostí 100 MB/s. Rozšiřuje režim LBA, který

místo 28bitové adresy používá adresu 48bitovou, max. velikost adresace pevného disku je tak 144 PB (Petabyte). Charakteristikou posledního standardu ATA 7 vyvíjeného od 2001 až do 2005 je přenosový režim Ultra-DMA 6 (Ultra-DMA 133, Ultra-ATA 133) o rychlosti 133 MB/s. Rozhraní ATA se svou verzí 7 dostává na hranici technologických možností, hlavně v oblasti časování signálů.

Odpověď na kontrolní otázku 11.4

Rozhraní ATA ve verzích 1 až 5 omezovaly velikost disku na 136,9 GB. Kombinací s verzí biosu se kapacita omezovala na 528 MB nebo 8,4 GB. Překročení omezení 8,4 GB bylo přijato standardem v roce 1996. Jednalo se o rozšíření příkazů softwarového přerušení INT13H na 64 bitů, 2^{64} tedy umožňuje adresovat až 9,44 ZB. To je však rozšíření biosu. Rozhraní ATA v konkrétních verzích ve své době podporuje maximální počet sektorů 28 bitovým adresováním, 2^{28} sektorů adresuje 137,44 GB. Tak velký diskový prostor byl ve své době nepředstavitelně velký a tak na několik let problém s omezením disku ustal. V roce 2001 se začaly na trhu objevovat pevné disky, které se k této kapacitě začaly přibližovat. Proto bylo nutné daná omezení opět řešit. Výsledkem bylo rozšíření 48 bitového adresování u standardu ATA-6. Kapacita, která je tímto adresováním hraniční, je až 144 PB (petabytů).

Odpověď na kontrolní otázku 11.5

Z pohledu konstrukce se disk skládá z následujících částí: plotny disku, hlavy pro čtení a zápis, pohon hlav s mechanismem na vystavení hlav, vzduchové filtry, pohon plotny disku, řídicí deska a nezbytné kabely a konektory.

Odpověď na kontrolní otázku 11.6

Geometrie disku udává hodnoty následujících parametrů:

Hlavy disku (heads): Počet čtecích (zapisovacích) hlav pevného disku. Tento počet je shodný s počtem aktivních ploch, na které se provádí záznam. Většinou každý jednotlivý disk má dvě aktivní plochy a k nim příslušné čtecí (zapisovací) hlavy.

Stopy disku (tracks): Počet stop na každé aktivní ploše disku. Stopy disku bývají číslovány od nuly, přičemž číslo nula je číslo vnější stopy disku.

Cylindry disku (cylinders): Počet cylindrů pevného disku. Tento počet je shodný s počtem stop. Číslování cylindrů je shodné s číslováním stop. Z tohoto důvodu bývá u pevného disku většinou udáván pouze počet cylindrů.

Přistávací zóna (landing zone): Číslo stopy (cylindru), která slouží jako přistávací zóna pro čtecí/zapisovací hlavy.

Sektory (sectors): Počet sektorů, na které je rozdělena každá stopa. U většiny starších pevných disků je podobně jako u pružných disků počet sektorů na všech stopách stejný. Tento způsob do jisté míry plynule přechází, protože vnější stopy jsou delší a tudíž by se na ně mohlo umístit více sektorů.

Odpověď na kontrolní otázku 11.7

Různé disky mají různý počet sektorů ve stopě. Diskety 8 až 36 sektorů ve stopě, disky 900 (i více) sektorů. Z důvodu kompatibility se staršími biosy a ovladači si disky provádějí interní logický překlad na 63 log. sektorů. Sektory se číslovají od 1, zatímco hlavy a cylindry od 0. Úvodní částí sektoru je hlavička *Header* identifikující začátek sektoru a obsahující jeho číslo. Zakončení sektoru se nazývá *Trailer* a využívá se pro kontrolní součet. Sektor má tudíž ve skutečnosti 571 bytů, ale pro data pouze 512 Bytů.

Odpověď na kontrolní otázku 11.8

Souborové systémy jsou založeny na alokační tabulce souborů FAT (File Allocation Table), obsahující údaje o datech uložených v každém sektoru na disku.

Jednotlivé varianty FAT:

- FAT12. Používá se na všech svazcích o kapacitě menší než 16 MB (např. na disketách),
- FAT16. Používá se na svazcích o velikosti 16 MB až 2 GB,
- FAT32. Lze využít na svazcích o kapacitě 512 MB až 2 TB.

Souborový systém NTFS byl navržen pro Windows NT, které také podporují nebo podporovaly HPFS (navržený pro OS/2) a FAT (navržený pro DOS), ale tyto souborové systémy mají omezené možnosti

a nesplňovaly podmínky kladené potřebami na systém. Těmito nedostatky jsou zejména omezená velikost disku a bezpečnost. Rovněž FAT32 podporovaný Windows 2000 tyto požadavky nesplňuje. NTFS byl navržen na konci 80 let při současně zahájeném vývoji NT a uveden v roce 1993 a lze říct, že byl navržen, aby splňoval nejen současné, ale i předpokládané požadavky.



Klíč k řešení – kapitola 12

Odpověď na kontrolní otázku 12.1

Pomineme-li grafické adaptéry pro monochromatická zobrazení, začneme tak od standardu CGA (Color Graphics Adapter) navržený firmou IBM v roce 1981. Tento standard umožnil maximální rozlišení 640 x 200 pixelů při čtyřbitové barevné hloubce, což znamená 16 barev. V roce 1984 byl představen standard EGA (Enhanced graphics adapter), jenž je vlastně vylepšený CGA standard pro rozlišení 640 x 350 pixelů při 16 barvách. Přímou pro své počítače řady PS/2 vyvinula IBM další tři standardy v roce 1987: VGA (Video Graphics Array), MCGA (MultiColor Graphics Array) a 851, přičemž pouze standard VGA se uchytí a zbývající dva zanikly. Standard VGA zvýšil maximální počet horizontálních pixelů až na 720 a vertikálních na 480, což zvýšilo možné rozlišení na 640 x 480 při 16ti barvách. Při nižším rozlišení umožňoval současně zobrazit až 256 barev z palety možností 262144 barev. Obnovovací frekvence standardu VGA dosahovala až 70 Hz. Po uvedení standardu XGA (eXtended Graphics Array), které je charakterizováno rozlišením 800 x 600 při 16 bitové barevné hloubce resp. 1024 x 768 při 8bitové barevné hloubce, firmou IBM v roce 1990, se ostatní výrobci vzepřeli a začali pracovat na vývoji levnějších variant grafických adaptérů při vyšším rozlišení. Tyto adaptéry měly jednoduché označení SVGA (Super VGA). Vývoj aplikací při absenci standardu byl velice obtížný, z čehož vyplynula nutnost standardizace SVGA. Standard SVGA podporuje celou řadu neustále se zvyšujících rozlišení při 24bitové barevné hloubce.

Odpověď na kontrolní otázku 12.2

- Grafická karta se mimo jiné skládá z těchto hlavních součástí:
- Video ROM BIOS, podobně jako u základní desky.
- Grafický procesor: řídí činnosti grafické karty, je součástí čipové sady grafického adaptéru.
- Paměť grafického adaptéru: uchovává informace, ze kterých grafický procesor vytváří obraz.
- Konvertor digitálního signálu RAMDAC: jedná se o obvod integrovaný přímo do čipu grafického adaptéru, který převádí digitální obraz vypočítaný grafickým procesorem na analogový signál monitoru.
- Rozhraní podporované sběrnice a výstupní konektor: zajišťuje výstup výsledného zobrazení na monitor či jiné zařízení.

Odpověď na kontrolní otázku 12.3

V současné době jsou paměťové čipy grafických karet typu DDR SDRAM. V minulosti byly grafické karty vybavovány čipy obdobné technologie jako paměťové moduly RAM. Jednalo se o FPM RAM (Fast Page Mode RAM) u dnes již zastaralé sběrnice ISA, EDO RAM (Extended Data Out RAM) pro sběrnice PCI a SDRAM obdobné paměťovým modulům DIMM pro počátky rozhraní AGP. Při vývoji technologie pamětí grafických karet však byly pokusy i o speciální paměťové moduly, které se od systémové paměti lišily. Jednalo se o paměti typů VRAM (Video RAM) a WRAM (Window RAM), které by se daly charakterizovat jako dual port paměti s možností současného čtení a zápisu. Tyto paměti vystřídala již zmíněná technologie SDRAM současně s technologií SGRAM (Synchronous Graphics RAM), jež byla vybavena velice rychlou paměťovou sběrnicí, jenž se synchronizovala s rychlostí sběrnice základní desky obdobně jako SDRAM.

Odpověď na kontrolní otázku 12.4

Elektronové dělo emituje po zahřátí proud elektronů o vysoké rychlosti. Elektronové svazky jsou vychylovány pomocí vychylovacích cívek tak, aby opisovaly shora dolů a zleva doprava jednotlivé řádky obrazovky. Vychylovací a zaostřovací cívky se nacházejí podél elektronky. Na vnitřní straně obrazovky je nanesená stínící maska děrovaná nebo štrbinová (podle typu obrazovky), která slouží

k seskupení RGB spektra do jednoho bodu. Masky je v podstatě mříž, která propustí pouze úzký svazek elektronů.

Odpověď na kontrolní otázku 12.5

Horizontální frekvence – řádkový kmitočet, měří se v kHz a udává rychlost, s jakou paprsek přejíždí z levé strany obrazovky na pravou. Vertikální frekvence – obnovovací kmitočet obrazu, měří se v Hz a udává počet obrazů překreslených za jednu sekundu.

Odpověď na kontrolní otázku 12.6

Výrobní technologie dělíme do tří kategorií. První kategorií a zároveň původní technologii je TN (*Twisted Nematic*). Ta se vyznačovala obdélníkovým tvarem pixelu, který zapříčinil různé pozorovací úhly ve vertikálním a horizontálním směru. Druhou kategorií tvoří technologie MVA (*Multi-Domain Vertical Alignment*) vyvinutá firmou Fujitsu Siemens a PVA (*Patterned Vertical Alignment*) vyvinutá firmou Samsung. Pixely jsou čtvercové a symetrické, takže pozorovací úhly jsou naprosto stejné. Molekuly již nejsou uspořádány do šroubovice jako u technologie TN, nýbrž do stromovitěho uspořádání, které při natáčení celého uspořádání disponuje s rychlejší odezvou. Třetí a poslední kategorií je technologie IPS (*In-Plane Switching*), která byla pro svou velice dlouhou odezvu zdokonalena na S-IPS, AS-IPS (*Advanced Super*). Tato kategorie poskytuje nejlepší barevné podání ze všech technologií vůbec.



Klíč k řešení – kapitola 13

Odpověď na kontrolní otázku 13.1

Data jsou na CD disku uložena ve spirále jako posloupnost pitů a landů. Hloubka pitu je 0,125 mikronu, šířka 0,6 mikronu, závity spirály jsou vzdáleny o 1,6 mikronu a celková délka dosahuje 5,77 km, u 80ti minutového CD je to 6,2 km. Laserový paprsek je ostřen na land, od kterého se odráží s vyšší intenzitou než od pitu, kam dopadá mírně rozostřen. Čtení dat tedy probíhá v závislosti na intenzitě odraženého paprsku od média. Princip čtení dat z CD-R média stejně jako u továrního lisovaného média je tedy prováděn laserovým světlem, které je různým způsobem odraženo od jamek a pevnin.

Odpověď na kontrolní otázku 13.2

Originální média se vyrábějí lisováním, tzv. rodinným procesem, kdy je vyrobena kovová matrice pro vyražení obrazu do polykarbonátu.

Tento proces probíhá v devíti krocích:

1. Pokrytí disku fotorezistentním materiálem.
2. Laserový záznam dat do fotorezistentní vrstvy.
3. Výroba vzorového záznamu – vznik jamek po políci vrstvy hydroxidem sodným a následným rozpuštěním ozářených míst.
4. Galvanické pokovování – kovová matrice je nazývána otec.
5. Oddělení matrice od skleněného vzorového disku.
6. Výroba medií 18g, roztavený polykarbonát, 350 st.Celsia , 138 Mpa, výroba 1 disku za 2 sekundy.
7. Pokovení tenkou vrstvou hliníku pro zajištění odrazivosti.
8. Zalití disku ochranným lakem – ochrana před oxidací.
9. Dokončení média – etiketa.

Odpověď na kontrolní otázku 13.3

Záznam na CD-R je prováděn laserovým paprskem vyšší intenzity, tento paprsek spálí organické barvivo, které pak již nepropouští světlo a nemůže tedy dojít k jeho odrazu od odrazivé vrstvy. Tímto se vytvoří ekvivalenty jednotlivých pitů a landů, což dovoluje, aby zaznamenané CD-R médium bylo čteno v běžné CD-ROM mechanice.

Přepisovatelné médium CD-RW obsahuje také polykarbonátový základ s předlisovanou stopou, ovšem obsahuje více vrstev. Druhou vrstvu tvoří dielektrický materiál (izolant) za níž následuje záznamová vrstva, ta je opět překryta dielektrikem, teprve poté je disk pokryt hliníkovou odrazovou vrstvou. Záznam se provádí na principu změny fáze záznamové vrstvy: krystalická odráží více světla zatímco amorfni odráží méně světla.

Záznamová vrstva má polykrystalickou strukturu, při záznamu dat se laser přepíná mezi dvěma výkonovými nastaveními P-write a P-Erase, P-write používá vyšší výkon a zahřeje se až na 500 až 700 stupňů, záznamová vrstva ztratí polykrystalickou strukturu. Druhé výkonové nastavení laseru zahřívá materiál pouze na teplotu okolo 200 stupňů, záznamová vrstva měkne a vrací se zpět polykrystalická struktura.

Odpověď na kontrolní otázku 13.4

Na rozdíl od CD je DVD médium dvouvrstvé a může být i oboustranné, přičemž při výrobě jsou vrstvy lisovány odděleně a následně pak slepeny. Mimo tento rozdíl oproti CD disku jsou známy další faktory zvyšující kapacitu DVD média. Jedná se o následující: zkrácení délek jamek a pevnin (z 0,9 na 0,4 mikronu), zmenšení rozteče mezi závitů (z 1,6 na 0,74 mikronu), mírné zvětšení plochy média využitelné pro záznam, zmenšení poměru redundance záznamu – z celkové velikosti sektoru je určeno více bytů pro data, zvýšení účinnosti modulace při záznamu, účinnosti kódu pro opravu chyb, atd.

Hlavním rozdílem vyplývajícím z těchto faktů je tedy kapacita média. Zmenšení velikosti jamek má za následek volbu laseru o kratší vlnové délce. Stopa je opět spirálovitá, její hustota je 1351 závitů na mm, tedy 49324 závitů na vrstvu, výsledná délka stopy je 11,84 km.

Odpověď na kontrolní otázku 13.5

Mechaniky pracují podle následujících kroků:

1. Laserová dioda emituje laserový paprsek směrem k pohyblivému zrcátku.
2. Na základě příkazu z CPU přesune servomotor zrcátko pod stopu, z níž se mají číst data.
3. Po dopadu na jamky a pevniny se světlo láme a odráží zpět, je ostřeno čočkou pod médiem, odráží se od zrcátka na optický hranol.
4. Odražené světlo optický hranol směřuje na další čočku.
5. Poslední čočka směřuje odražené světlo na fotocitlivý senzor, ten převádí světelné impulzy na elektrické.
6. Elektrické impulzy jsou dekodovány mikroprocesorem a předány jako data.

Odpověď na kontrolní otázku 13.6

Přenosová rychlost určuje s jakou rychlostí mechanika čte soubory, jedná se o ustálenou rychlost, tedy až poté, co se mechanika roztočí na plnou rychlost otáčení a najede na požadované místo. Pro CD je Single speed: 153,6 kB/s, pro DVD je 1,385 MB/s. Od této rychlosti se počítají rychlosti mechanik s příslušným násobkem (16x, 32x, 48x, 52x).



Klíč k řešení – kapitola 14

Odpověď na kontrolní otázku 14.1

Zvuk je šířen pomocí vibrací, které stlačují vzduch. Je tedy podélným mechanickým vlněním v látkovém prostředí, které vyvolává v lidském uchu sluchový vjem. Tyto zvukové vlny se ze svého zdroje šíří všemi směry rovnoměrně. Jakmile dorazí do lidského ucha, rozkmitají ušní bubínky a tento vjem je pak převeden na zvuk.

Odpověď na kontrolní otázku 14.2

Pro zvuk jsou charakteristické dvě jeho vlastnosti: výška a intenzita. Výška zvuku je měřena v hertzech (Hz) neboli cyklech za sekundu a je dána rychlostí šíření jednotlivých vibrací, kde jedním cyklem se miní úplná vibrace dopředu a dozadu. Počet hertzů pak udává frekvenci tónu, čím vyšší je frekvence, tím vyšší je vnímaná výška tónu. Intenzitou zvuku nazýváme amplitudu a definuje hlasitost, s jakou daný zvuk vnímáme.

Odpověď na kontrolní otázku 14.3

Frekvence slyšitelného zvuku leží v rozsahu přibližně 16 Hz až 20 kHz.

Odpověď na kontrolní otázku 14.4

Ztrátová komprese se nejčastěji používá pro ukládání obrazových a zvukových záznamů. Pomocí speciálního algoritmu se zmenšuje objem dat na zlomek původní velikosti. Přitom se některé méně důležité informace ztrácejí a z vytvořených dat již nejdou zrekonstruovat. Přesto, že se část informace při ztrátové kompresi nevratně ztrácí, je tento způsob ukládání dat často velmi výhodný. Ztráta některých informací je totiž zcela vyvážena velmi výrazným zmenšením komprimovaných dat. Obvykle je tak určitá (malá) ztráta kvality vyvážena výraznou úsporou místa. Algoritmus ztrátové komprese má dvě podstatné části, jimiž jsou transformace původních dat a potlačení méně důležitých dat.

Odpověď na kontrolní otázku 14.5

Zvuková karta obsahuje zvukový čip, který provádí digitálně-analogový převod vygenerovaného digitálního záznamu. Tento signál je přiveden na výstup zvukové karty. Obsahuje také linkový vstup pro zdroj zvukového signálu. Třetím obvyklým konektorem je připojení mikrofону. Většina zvukových karet má také MIDI a GamePort konektor. Konektor MIDI slouží k připojení např. elektrických varhan, nebo jiného zdroje digitálního signálu. Pokud nahráváme z MIDI, tak nemusíme provádět vzorkování. Součástí zvukových karet jsou také čipy pro dekódování prostorového zvuku. Prostorový zvuk je dekódován pro 5.1 někdy 6.1 či dokonce 7.1 kanálů. První číslo znamená počet reproduktorů, jednička za tečkou označuje subwoofer neboli speciálně upravený reproduktor pro nízké kmitočty basů. Zvukové adaptéry mohou být vybaveny SPDIF (Sony/Philips Digital Interface) rozhraním umožňujícím přímý vstup či výstup digitálního signálu, přitom fyzická vrstva zajišťující přenos digitálního signálu může být metalická ale i optická.

Odpověď na kontrolní otázku 14.6

Kvalita zvukových adaptérů se měří třemi kritérii: frekvenčním rozsahem, celkovým harmonickým zkreslením a odstupem signálu od šumu.

Rejstřík

3

3DNow! · 98

A

Adaptér pevného disku · 8
ADPCM · 214
Adresová sběrnice · 94
Adresový operátor · 49
AGP · 132, 193
AMD K5 · 111
AMD K6 · 113
ANSI C · 41, 44, 75
Argumenty příkazového řádku · 72
architektura čipové sady · 79
aritmetické operátory · 47
AS-IPS · 199
ATA · 170, 172
ATA IDE · 172
ATAPI · 173
Athlon 64 · 120
Athlon 64 X2 · 120
Athlon XP · 114

B

Barevné značení textu · 21
Base frequency · 95
BEDO RAM · 161
BGA · 102
binární · 47, 75
BIOS · 8, 11, 147, 156, 175
Bitové operátory · 48
Blokové schéma procesoru Pentium · 110
Blu-Ray · 210
Breakpoints · 34, 39
Bulk Silicon · 100

C

Cable Select · 173
Cache · 8, 180
CAS · 164
Case-sensitive · 43
CAV · 209
CD-R · 203
CD-ROM · 203
CD-RW · 205
Celočíselné konstanty · 45
CGA · 191
CISC · 8
ClassView · 19
CLV · 209
const · 45

CPU · 8
CRC · 174
CRT · 195
Curiova teplota · 202
Cyklus do · 58
Cylindry · 181

Č

čipová sada · 80
Čipová sada 810 · 83
Čipová sada 820 · 83
čipová sada 845 · 85
čipová sada 850 · 85
čipová sada 865 · 86
čipová sada 915 · 86
Čipová sada 925 · 88
čipová sada 945 · 88
čipová sada 965 · 89
Čipová sada 975 · 89

D

Datová sběrnice · 94
DCD · 145
DCE · 143
DDR · 79, 166
DDR SDRAM · 161, 193
DDR2 · 166
DDR2 SDRAM · 162
Definice funkce · 63
Definice pevného disku · 180
definice pole · 66
definice symbolických konstant · 62
Deklarace funkce · 64
Delta obrazovka · 196
DIMM · 165, 166
DIP · 101
DirectX · 192
Direktiva preprocesoru · 61
DMA · 8, 129
DMA režim · 171
Doba odezvy · 197
Doporučené rozlišení · 197
DRAM · 159
DSR · 145
DTE · 143
DTR · 145
Dual Channel · 162
DualCore · 122
DVD · 207
Dynamické vykonávání instrukcí · 112

E

ECC kód · 165
ECP · 147

EDO RAM · 161, 193
EEPROM · 157, 158
EFM · 204
EGA · 191
EIDE · 174
EISA · 130
EM64T · 98
Enhanced 3DNow! · 98
EPP · 147
EPROM · 157
ESDI · 170
Execution Trace Cache · 116
eXtreme Performance · 115

F

FAT · 184, 186
fclose · 70
FC-PGA · 102
FDISK · 185, 186
FDIV · 111
feof · 72
fgetc · 71
FILE · 69
FileView · 19
FireWire · 140, 141
Firmware · 9
Flash · 158
float.h · 44
fopen · 69
for · 57
Form faktory · 11
formální parametry · 64
Formátování · 183
fotocitlivý senzor · 209
FPM RAM · 160, 193
fprintf · 71
fputs · 71
Frekvenčním rozsah · 216
Front Side Bus · 79
FSB · 13, 89
fscanf · 71
fseek · 72
Funkce · 63
Funkce standardního vstupu a výstupu · 50

G

Geometrie disku · 181
getc · 70
getchar · 51
gets · 51
grafická karta · 191
Grafický adaptér · 191
Grafický akcelerátor · 9, 192
Grafický procesor · 192

H

Hardware · 9
harmonické zkreslení · 217
HCA · 207
Hexadecimální konstanty · 46

Hlavy disku · 181
Hlavy pro čtení a zápis · 179
Horizontální frekvence · 196
Hot Plug-In · 177
Hustota integrace · 159
HyperThreading · 97
HyperTransport · 97

Ch

Chlazení procesoru · 104

I

i.Link · 141
IDE · 140, 170
identifikátor · 43, 75
IEEE1394 · 9
IEEE 1394 · 141
IEEE-1284 · 147
IEEE754 · 44
if · 54
if – else · 54
ICH · 82
Index (Rejstřík) · 26
Informace · 9
Inicializace ukazatele · 65
Inline obrazovka · 196
Instrukční sada · 97, 106
INTEL · 93
INTEL 8088 · 93
intenzita zvuku · 213
interleaving · 183
interpunkční znaky · 43, 75
IRQ · 9
ISA · 128
ISO 9660 · 204

J

Jamky a pevniny · 203
Jas · 197
jižní most · 79

K

Kapacita paměti · 194
Kerrův efekt · 202
klíčová slova · 43, 75
koercivita · 202
Kompilace · 22
Kompilace programu · 23, 27
Kompilátor · 24, 27
Komprese zvuku · 214
Konstanty · 45
Kontrast · 197
Kontrola parity · 165
kořenový adresář · 185
Krokování dovnitř funkcí · 31
Krokování programu · 30

L

L1 a L2 Cache · 97
 Ladění · 29
 landy · 203
 Latence · 95, 106
 LBA · 175, 176
 LCD · 197
 LGA 771 · 123
 L-CHS · 175
 LIF · 104
 limits.h · 44
 Logic board · 179
 logické operátory · 47
 Lokální sběrnice · 128
 luminofory · 196

M

mainboard · 10
 mainstream · 89
 MBR · 185
 MCA · 129
 MCA IDE · 172
 MCGA · 191
 metadata · 187
 MCH · 82
 MMX · 98
 MNOS · 157
 monitor · 191
 motherboard · 10
 MPEG · 214
 MVA · 199

N

Nápověda MSDN · 25
 návratová hodnota · 63
 NetBurst · 115
 nízkoúrovňové formátování · 184
 North Bridge · 79
 Northwood · 99
 NRZ · 177
 NRZI · 141
 NTFS · 185, 187

O

Odstranění zářezek · 33
 Odstup signálu od šumu · 217
 Okno Variables · 36
 Okno Watch · 36, 39
 Okno Workspace · 21, 27
 Oktalové konstanty · 45
 OpenGL · 192
 Operand · 47
 operátor · 47
 Operátor přiřazení · 48
 operátory · 43, 75
 Operátory dekrementace · 49
 Operátory inkrementace · 49
 Oprava vytvořených chyb · 24

Opteron · 118

P

Paměť · 9
 Paměť grafické karty · 193
 Paměť grafického adaptéru · 192
 Paměťová sběrnice · 128
 Paměťové čipy · 158
 Paralelní port · 146
 parametry monitorů · 196
 PCI · 131
 PCI – Express · 134
 PCI-X · 131
 Pentium 4 · 115
 Pentium D · 123
 Pentium II · 112
 Pentium III · 113
 PGA · 101, 109
 P-CHS · 175
 PIIX · 80
 pity · 203
 Pixel · 198
 Plotny disku · 179
 Podmíněný operátor · 49
 Podmíněný příkaz · 54
 Pohon hlav · 179
 Pole · 66
 Polovodičové paměti · 155
 polykarbonát · 205, 206
 Porty · 9
 Použití nápovědy · 25
 použití zářezek · 32
 Pozorovací úhel · 197
 Preprocesor · 60
 Princip zobrazení · 195
 printf · 51, 71
 Procesorová sběrnice · 128
 Procesory osmé generace · 118
 Procesory páté generace · 109
 Procesory sedmé generace · 114
 Procesory šesté generace · 112
 Projekt · 18, 27
 PROM · 156
 proměnné · 45
 Proměnné · 47
 Propustnost · 127
 Přenosová rychlost · 184
 Přepínač · 55
 Přepínání mezi soubory · 20
 přetypování výrazu · 49
 Příkaz skoku · 59
 Příkazy a bloky · 54
 Přistávací zóna · 182
 přístupová doba · 160
 puštění programu · 22
 putc · 70
 putchar · 50
 puts · 51
 PVA · 199

Q

QDR · 79

Quad Pumped Bus · 95, 106

R

RAM · 155, 158
 RAMDAC · 192
 RAS · 163
 RDRAM · 85, 163
 Reálné konstanty · 46
 Red Book · 204
 Registr · 9
 Rekurzivní funkce · 64
 relační operátory · 47
 RGB · 195, 198
 RIMM · 166
 RISC · 9
 ROM · 155, 156
 Root · 185
 Rozdělení sběrnice · 127
 RPC · 208
 RS 232 · 141
 RTS · 145

Ř

Řadič · 9
 Řetězce · 67
 Řetězcové konstanty · 46
 Řídicí struktury · 54

S

S.M.A.R.T · 180
 sběrnice · 128
 Sběrnice · 9
 Sběrnice procesorů · 93
 scanf · 51, 71
 SCSI · 140, 178
 SDRAM · 161, 193
 SEC 2 · 12
 SECC · 12
 Sektory · 182
 SEP · 13
 Serial ATA · 177
 Sériový port · 141
 severní most · 79
 SGRAM · 193
 Shugart Technologies · 170
 signed · 44, 75
 Silicon On Insulator · 100
 SIMD · 98
 SIMM · 103, 165
 SIO · 80
 SIP · 101
 SIPP · 164
 S-IPS · 199
 sizeof() · 49
 skutečné parametry · 64
 Sledování hodnot proměnných · 36
 SLOT A · 13
 Sloty · 12
 SMP · 80
 Socket 370 · 13

socket 478 · 86
 Socket 7 · 12
 Socket 754 · 14
 socket AM2 · 14
 Socket LG775 · 13
 sockety · 12
 Software · 9
 South Bridge · 79
 Southbridge · 87, 88
 SPDIF · 216
 SPGA · 12, 102, 109
 Spotřeba · 197
 SPP · 147
 Spuštění programu · 23
 SRAM · 158
 SSE · 98
 SSE2 · 98
 SSE3 · 98
 ST506 · 170
 Start bit · 142
 stdio.h · 42
 stínící maska · 195
 Stop bit · 142
 Stopy · 181
 Strained Silicon · 100
 Strained Silicon Directly On Insulator · 100
 Struktura CD · 204
 Struktura souborového systému · 184
 superskalární · 94
 superskalární technologie · 109
 SVGA · 192
 switch · 55
 symetrický multiprocessing · 82

Š

šířka sběrnice · 93, 106

T

taktovací frekvence · 160
 TDP · 105
 ternární · 47, 75
 TFT · 198
 Third Generation Input and Output · 134
 TN · 198
 Tovární výroba CD disku · 205
 Trinitron obrazovka · 196
 TTL · 142
 Typ struktura · 73
 Typ Unie · 74

U

UDF · 205
 Úhlopříčka · 197
 ukazatele · 66
 Ukazatele · 65
 Ultra-ATA · 174
 Ultra-DMA · 171, 174
 unární · 47, 75
 unsigned · 44, 75
 USB · 140

USB rozhraní · 9
Uživatelský datový typ · 73

V

VBR · 185
Vertikální frekvence · 196
VESA · 130
vícerozměrná pole · 67
Video ROM BIOS · 192
Visual C++ 6.0 · 18
VL bus · 130
Von Neumannovo schéma · 10
VRAM · 193
Výčtový typ · 73
Vyhledávání chyb · 24
výraz · 47
Výroba procesoru · 101
Vysokoúrovňové formátování · 184
Výstupní konektor · 193
Výška zvuku · 213
Vzduchové filtry · 179

W

while · 57
Williamette · 99
Win32 Console Application · 18, 27
Win32 Debug · 30

WRAM · 193

X

x86 · 98
x87 · 98
XGA · 191
XOff. · 145
XOn · 145
XT IDE: · 172

Z

Z-80 · 93
Základní desky · 10
Základní typy dat · 44
založení nového projektu · 18
Zarážka s podmínkou · 34
Zarážky · 32, 39
Zarážky (breakpointy) · 32
Zarovnání zdrojového textu · 21
zásobník BTB · 109
Zavření a obnovení oken projektu · 21
ZBR · 182
ZIF · 104
Znakové konstanty · 46
zpoždění · 160
Zvuková karta · 215