Notatki do wykładu z architektury komputerów

Marcin Peczarski

Instytut Informatyki
Uniwersytet Warszawski

6 lutego 2007

Architektura a organizacja komputera

Architektura komputera, według Stallingsa [1], to te atrybuty komputera, które są widzialne dla programisty i mają bezpośredni wpływ na logiczne wykonanie programu. Natomiast *organizacja komputera* jest to sposób realizacji architektury. Przykładami atrybutów architektury są: lista rozkazów, sposób reprezentacji liczb, metody adresowania argumentów. Przykładami atrybutów organizacji są: sposób realizacji instrukcji, rozwiązania sprzętowe niewidoczne dla programisty, technologie wykonania poszczególnych podzespołów. Z powyższymi pojęciami wiąże się pojęcie *kompatybilności*. Architektura jest czymś trwalszym niż organizacja. Ta sama architektura może być oferowana w różnych modelach komputerów, różniących się organizacją, która z kolei determinuje np. wydajność i cenę. Niezmienna architektura może być oferowana przez producenta przez wiele lat, umożliwiając łatwe przenoszenie istniejącego już oprogramowania. Natomiast kolejne modele komputerów mogą mieć zupełnie inną organizację, która może zmieniać się dowolnie często wraz z rozwojem technologii. Zgodnie z powyższymi definicjami, a nieco wbrew tytułowi, wykład ten poświęcony jest zarówno architekturze, jak i organizacji systemów komputerowych.

1 Podstawy techniki cyfrowej

1.1 Układy cyfrowe

Sygnałem nazywamy funkcję opisującą zmiany wielkości fizycznej w czasie. *Sygnałem analogowym* nazywamy sygnał przyjmujący nieskończoną (wg innej definicji nieprzeliczalną) liczbę wartości, na ogół z ciągłego przedziału. *Sygnałem dyskretnym* (nazywanym też: *ziarnistym*, *skwantowanym*, *cyfrowym*) nazywamy sygnał przyjmujący skończoną (wg innej definicji przeliczalną) liczbę wartości. *Układami analogowymi* nazywamy urządzenia przetwarzające sygnały analogowe. *Układami cyfrowymi* nazywamy urządzenia przetwarzające sygnały dyskretne. Przymiotnik *cyfrowy* pochodzi zapewne stąd, że wartościom sygnałów dyskretnych na ogół przypisujemy cyfry w pewnym systemie liczenia, na ogół dwójkowym. Należy pamiętać, że podział na urządzenia analogowe i cyfrowe odzwierciedla sposób interpretacji sygnałów. W układach fizycznych sygnały dyskretne są reprezentowane przez sygnały analogowe. Jeśli uwzględnić zjawiska kwantowe, to sygnały analogowe przyjmują tylko wartości dyskretne.

Wśród układów cyfrowych dominujące znaczenie mają *układy binarne* (*dwójkowe*), w których sygnały przyjmują tylko dwie wartości. Te dwie wartości oznaczane są cyframi 0 i 1. Wartości sygnałów binarnych są, na ogół, reprezentowane przez dwa poziomy potencjału elektrycznego. Są to *poziom wysoki* oznaczany literą H i *poziom niski* oznaczany literą L. Wartość potencjału elektrycznego reprezentującego poziom wysoki jest zawsze większa niż wartość potencjału elektrycznego reprezentującego poziom niski. W konwencji logicznej dodatniej poziomowi wysokiemu przypisuje się cyfrę 1, a poziomowi niskiemu cyfrę 0. W konwencji logicznej ujemnej jest odwrotnie, poziomowi wysokiemu przypisuje się cyfrę 0, a poziomowi niskiemu cyfrę 1.

1.2 Bramki logiczne

Podstawowe bramki logiczne: AND, NAND, OR, NOR, EX-OR, NOT.

Pomocnicze bramki cyfrowe: bramka transmisyjna, bufor trójstanowy.

Prawa de Morgana dla bramek.

1.3 Układy kombinacyjne

Układem kombinacyjnym nazywamy układ cyfrowy, w którym sygnały wyjściowe są jednoznacznie określone przez aktualne wartości sygnałów wejściowych. Układ kombinacyjny o n wejściach i m wyjściach jest w pełni opisany przez podanie funkcji przełączającej $f\colon X \to Y$, gdzie $X \subseteq \{0,1\}^n$ jest zbiorem dopuszczalnych stanów wejść, a $Y \subseteq \{0,1\}^m$ jest zbiorem możliwych stanów wyjść. Jeśli $X = \{0,1\}^n$, to funkcja przełączająca jest *zupełna*.

System funkcjonalnie pełny to zestaw typów bramek, z których można zbudować dowolny układ kombinacyjny. Przykłady najważniejszych systemów bramek funkcjonalnie pełnych:

- AND, OR, NOT;
- NAND;
- NOR.

Przykłady układów kombinacyjnych: sumator, multiplekser, demultiplekser.

Hazard w układach kombinacyjnych.

1.4 Układy sekwencyjne

Układem sekwencyjnym nazywamy układ cyfrowy, w którym sygnały wyjściowe zależą nie tylko od wartości sygnałów wejściowych w danej chwili, ale również od wartości sygnałów wejściowych w przeszłości. Układ sekwencyjny o n wejściach i m wyjściach można symbolicznie opisać jako piątkę $\langle Q, X, Y, \delta, \lambda\rangle$, gdzie:

- Q jest zbiorem stanów wewnętrznych, na ogół $Q \subseteq \{0,1\}^p$;
- $X \subseteq \{0,1\}^n$ jest zbiorem dopuszczalnych stanów wejść;
- $Y \subseteq \{0,1\}^m$ jest zbiorem możliwych stanów wyjść;
- $\delta\colon Q \times X \to Q$ jest funkcją przejść;
- $\lambda\colon Q \times X \to Y$ jest funkcją wyjść.

Układ sekwencyjny jest często nazywany *deterministycznym automatem skończonym*.

Ogólny schemat budowy układu sekwencyjnego.

Graf przejść układu sekwencyjnego.

Układ sekwencyjny nazywamy *asynchronicznym*, gdy spełniony jest warunek

$$\forall \mathbf{x} \in X\, \forall \mathbf{q}, \mathbf{s} \in Q\, (\delta(\mathbf{s}, \mathbf{x}) = \mathbf{q} \Rightarrow \delta(\mathbf{q}, \mathbf{x}) = \mathbf{q})$$

Ten warunek gwarantuje stabilność i oznacza, że w układzie asynchronicznym zmiana stanu wewnętrznego lub stanu wyjść może nastąpić jedynie pod wpływem zmiany stanu wejść.

Przykład układu asynchronicznego: przerzutnik RS.

Hazard w układach asynchronicznych.

Układ sekwencjny nazywamy *synchronicznym*, gdy zmiana stanu wewnętrznego i stanu wyjść jest synchronizowana *sygnałem taktującym*, zwanym też *sygnałem zegara*. Impulsy taktujące dzielą czas na odcinki zwane *taktami*, które wyznaczają dyskretny czas $t \in \mathbb{Z}$ i mamy $\mathbf{q}_{t+1} = \delta(\mathbf{q}_t, \mathbf{x}_t)$ oraz $\mathbf{y}_t = \lambda(\mathbf{q}_t, \mathbf{x}_t)$

Układ synchroniczny może zmieniać stan wewnętrzny i stan wyjść pod wpływem samego tylko sygnału taktującego bez zmiany stanu wejść. Moment zmiany wyznacza poziom sygnału zegarowego – *wyzwalanie poziomem*, częściej jednak zmiana następuje przy zmianie poziomu sygnału zegarowego z niskiego na wysoki – *wyzwalane zboczem narastającym* lub z wysokiego na niski – *wyzwalane zboczem opadającym*. Układ synchroniczny można uważać za szczególny przypadek układu asynchronicznego, jeśli uwzględnić sygnał zegara jako jeden z sygnałów wejściowych.

Przykład układu synchronicznego: przerzutnik D.

1.5 Aspekty technologiczne

Projektowanie i wytwarzanie układów cyfrowych podlega pewnym ograniczeniom:

- fan-in – maksymalna liczba wejść pojedynczej bramki;
- fan-out – maksymalna liczba wejść bramek, które można podłączyć do jednego wyjścia bramki;
- margines zakłóceń;
- czas propagacji – determinuje maksymalną częstotliwość przełączania;
- pobierana moc.

Polepszenie jednego z powyższych parametrów zwykle wiąże się z pogorszeniem innego.

Moc P pobieraną przez układ cyfrowy w zależności od napięcia zasilania U i częstotliwości przełączania (taktowania) f można wyrazić wzorem

$$P = (G + Cf)U^2.$$

Ze względu na wartości stałych G i C technologie wytwarzania układów cyfrowych można podzielić na trzy kategorie:

- $G > 0$, $C > 0$ – np. bardzo popularne w ubiegłym wieku układy TTL;
- $G > 0$, $C \approx 0$ – np. ciągle jeszcze używane w specyficznych zastosowaniach układy ECL;
- $G \approx 0$, $C > 0$ – np. dominujące obecnie układy CMOS.

Technologie realizacji bramek zwykło się porównywać na podstawie czasów propagacji t_p i mocy zasilania P_b pojedynczej bramki oraz iloczynu tych wielkości. Iloczyn $t_\mathrm{p}P_\mathrm{b}$ ma wymiar energii i jest adekwatny do porówywania technologii, dla których stała G dominuje nad stałą C (stała C ma zaniedbywalny wpływ na pobieraną moc). Jeśli tak nie jest, to należy określić częstotliwość, dla której dokonywane jest porównanie. Jeśli stała C dominuje nad stałą G, to lepszym kryterium porównawczym jest energia jednego cyklu przełączania bramki CU^2.

Energia elektryczna pobierana przez elektroniczny układ cyfrowy jest prawie w całości zamieniana na ciepło. W danych katalogowych producenci podają maksymalną temperaturę obudowy T_c (lub struktury półprzewodnikowej), przy której układ może pracować poprawnie. Temperatura T_c, temperatura otoczenia T_o i wydzielana w układzie moc P są związane wzorem

$$T_\mathrm{c} = T_\mathrm{o} + R_\mathrm{th}P,$$

gdzie R_th jest rezystancją termiczną obudowa-otoczenie (lub struktura-otoczenia), wyrażaną w $\mathrm{K} \cdot \mathrm{W}^{-1}$. Jako temperaturę T_o należy przyjąć temperaturę wewnątrz obudowy komputera. Temperatura ta jest na ogół wyższa od temperatury panującej na jej zewnątrz. We współczesnych komputerach niewielka część ciepła wydzielanego w jego układach zostaje odprowadzona przez promieniowanie lub naturalną konwekcję. Większość musi zostać odprowadzona przez wymuszenie przepływu czynnika chłodzącego, najczęściej powietrza lub wody. Wydzielana moc P w W, szybkość przepływu czynnika chłodzącego q w $\mathrm{kg} \cdot \mathrm{s}^{-1}$, ciepło właściwe czynnika chłodzącego (przy stałym ciśnieniu) c_p w $\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}$ i przyrost temperatury czynnika chłodzącego ΔT w K związane są zależnością

$$P = qc_p\Delta T.$$

Za pomocą tego wzoru można obliczyć przyrost temparatury wewnątrz obudowy komputera, jak również w serwerowni. Należy wtedy jako P przyjąć łączną moc wszystkich urządzeń znajdujacych się w pomieszczeniu, a jako q wydajność układu wentylacji lub klimatyzacji. Ciepło właściwe powietrza wynosi $1{,}01 \cdot 10^3$ $\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}$, a wody $4{,}18 \cdot 10^3$ $\mathrm{J} \cdot \mathrm{kg}^{-1} \cdot \mathrm{K}^{-1}$. Szybkość przepływu można zamienić z $\mathrm{kg} \cdot \mathrm{s}^{-1}$ na $\mathrm{m}^3 \cdot \mathrm{s}^{-1}$, uwzględniając gęstość czynnika chłodzącego. Dla wody wynosi ona w przybliżeniu 10^3 $\mathrm{kg} \cdot \mathrm{m}^{-3}$ i w niewielkim stopniu zależy od temperatury, natomiast dla powietrza zależy od temperatury i ciśnienia, ale dla obliczeń przybliżonych można przyjąć, że wynosi nieco ponad 1 $\mathrm{kg} \cdot \mathrm{m}^{-3}$, co jest łatwo zapamiętać.

Podsumowując, w celu zapewnienia właściwych warunków pracy układu stosuje się następujące rozwiązania:

- radiator zmniejsza R_th;
- specjalna pasta termoprzewodząca między obudową a radiatorem zmniejsza R_th;

- wentylator zmniejsza T_o;
- chłodzenie cieczą może drastycznie zmniejszyć T_o.

Zwiększanie częstotliwości taktowania układu na ogół nieuchronnie prowadzi do zwiększania ilości wydzielanego w nim ciepła. Można temu przeciwdziałać, np. zmniejszając napięcie zasilania. Niestety zmniejszanie napięcia zasilania powoduje wzrost czasu propagacji, co powoduje ograniczenie maksymalnej częstotliwości taktowania. Zmniejszanie napięcia zasilania zmniejsza też margines zakłóceń, co czyni układ cyfrowy mniej odpornym na zakłócenia elektromagnetyczne. Znalezienie właściwego kompromisu jest jednym z istotnych problemów w technologii układów cyfrowych.

1.6 Prawo Moore'a

W 1965 roku Gordon Moore sformułował prawo dotyczące tempa rozwoju mikroelektroniki: *Liczba elementów, które można umieścić w układzie scalonym, minimalizując koszt produkcji na jeden element, rośnie wykładniczo w czasie*. Prawo to nadal obowiązuje. Czas potrzebny na podwojenie liczby elementów szacuje się obecnie na 18 do 24 miesięcy.

1.7 Układy wrażliwe na ładunki elektrostatyczne

Większość produkowanych obecnie układów scalonych jest wrażliwa na ładunki elektrostatyczne (ang. ESD – electrostatic sensitive devices). Uszkodzenia powstające na skutek niekontrolowanego gromadzenia się i rozładowywania ładunków elektrostatycznych są bardzo trudne do zlokalizowania, gdyż błędne działanie układu może wystąpić długo po zaistnieniu przyczyny uszkodzenia i może objawiać się tylko od czasu do czasu. ESD i moduły je zawierające są oznaczane przez umieszczenie na opakowaniu specjalnego symbolu.

Opakowania przeznaczone do przechowywania i przenoszenia ESD są wykonywane z tworzyw ułatwiających łagodne odprowadzanie ładunków elektrostatycznych. Należy przestrzegać następujących zaleceń:

- Powierzchnia stołu, na którym wykonuje się montaż i demontaż ESD, powinna być pokryta tworzywem przewodzącym, ale nie metalowa. Warstwa przewodząca powinna być uziemiona.
- Masy wszystkich urządzeń znajdujących się na stole powinny być połączone galwanicznie z powierzchnią przewodzącą stołu.
- ESD nie wolno przenosić bez odpowiedniego opakowania.
- Przed wyjęciem ESD z opakowania lub włożeniem ESD do opakowania należy wyrównać potencjały przez dotknięcie opakowania do warstwy przewodzącej stołu.
- Osoba pracująca z ESD powinna mieć na ręce opaskę odprowadzającą ładunki elektrostatyczne. Ze względów bezpieczeństwa opaska powinna być połączona z uziemieniem przez dużą rezystancję.
- W pomieszczeniach, gdzie pracuje się z ESD, należy stosować antystatyczne wykładziny podłogowe.

2 Przegląd architektur

2.1 Model von Neumanna a model współczesnego komputera

John von Neumann opublikował w 1945 propozycję opracowania nowego komputera. Zasadniczą nowością była koncepcja *przechowywania programu w pamięci*. Komputer miał składać się z czterech bloków funkcjonalnych:

- pamięć przechowująca program do wykonania i dane dla niego;
- jednostka arytmetyczno-logiczna zawierająca rejestry AC (accumulator), MQ (multiplier-quotier), MBR (memory buffer register);
- jednostka sterująca zawierająca licznik programu PC (program counter), rejestr adresowy MAR (memory address register) i pomocnicze rejestry IBR (instruction buffer register), IR (instruction register);
- urządzenia wejścia-wyjścia.

Składniki współczesnego komputera:

- (mikro)procesor, zawierający co najmniej jedną jednostkę arytmetyczno-logiczną, jednostkę sterująca i rejestry;
- pamięć operacyjna;
- urządzenia wejścia-wyjścia (klawiatura, mysz, karta graficzna, pamięci dyskowe itp.);
- układ bezpośredniego dostępu do pamięci (ang. DMA – direct memory access);
- układ przerwań.

Urządzenia wejścia-wyjścia same też mogą być skomplikwanymi układami mikroprocesorowymi i zawierać w swoim wnętrzu mikroprocesor, pamięć operacyjną, układy wejścia-wyjścia itd. Układ DMA jest specjalizowanym procesorem odciążającym procesor główny przy operacjach przesyłania dużych bloków danych między pamięcią operacyjną a układami wejścia-wyjścia. Komunikacja pomiędzy poszczególnymi składnikami odbywa się za pomocą szyn (zwanych też magistralami): danych, adresowych, sygnałów sterujących.

2.2 Architektury typu Princeton i Harward

Architektura typu Princeton posiada wspólną hierarchię pamięci programu i danych, jak opisano to w modelu von Neumanna. *Architektura typu Harward* polega na rozdzieleniu pamięci programu od pamięci danych. Stosowana jest dla zwiększenia wydajności w pamięciach podręcznych oraz w systemach wbudowanych (np. sterownikach urządzeń AGD, gdzie kod programu nie zmienia się przez całe życie urządzenia lub zmienia się rzadko).

2.3 Klasyfikacja Flynna

Klasyfikacja Flynna ma obecnie znaczenie historyczne, niemniej skróty w niej zdefiniowane są w powszechnym użyciu i należy je znać. Poniższa tabela przedstawia rozszerzoną klasyfikację Flynna – oryginalna nie zawiera kolumny 0 i wiersza 0.

		liczba strumieni danych		
		0	1	> 1
liczba strumieni instrukcji	0		NISD	NIMD
	1	automat	SISD	SIMD
	> 1	automat	MISD	MIMD

Skróty oznaczają odpowiednio: NI – no instruction, SI – single instruction, MI – multiple instruction, SD – single data, MD – multiple data. Maszyny typy NISD i NIMD nie są obecnie konstruowane, ale samo podejście *sterowania przepływem danych* (ang. dataflow) jest używane do modelowania procesów informacyjnych. Model SISD to klasyczna maszyna von Neumanna. Model SIMD to np. procesory wektorowe, które zostaną omówione w dalszej części wykładu. Trudno jest wskazać jakieś praktyczne zastosowanie modelu MISD. Model MIMD to różnego rodzaju architektury wieloprocesorowe. Urządzenia bez strumieni danych nie są komputerami – można tu umieścić niektóre automaty.

2.4 Architektury wieloprocesorowe

Podział architektur wieloprocesorowych ze względu na stopień powiązania:

- superkomputer, ang. massively parallel processing;
- klaster, grono, ang. cluster;
- konstelacja, ang. constellation;
- siatka, ang. grid.

Podział architektur wieloprocesorowych ze względu na dostęp do pamięci:

- procesor ma dostęp tylko do pamięci lokalnej;
- SMP (Symmetric Multi-Processing), UMA (Uniform Memory Access, Uniform Memory Architecture) – wszystkie procesory mają równoprawny dostęp do wspólnej pamięci;
- NUMA (Non-Uniform Memory Access, Non-Uniform Memory Architecture) – dostęp do pamięci lokalnej jest bardziej efektywny niż dostęp do pamięci innych procesorów;
- COMA (Cache Only Memory Architecture) – pamięć lokalna pełni funkcję pamięci podręcznej dla wszystkich procesorów;
- architektury mieszane.

3 Reprezentacja danych

3.1 Liczby całkowite

Rozważmy ciąg n-bitowy

$$b_{n-1}b_{n-2}\ldots b_2b_1b_0.$$

Ciąg ten może oznaczać różne liczby w zależności od przyjętego sposobu kodowania:

naturalny kod binarny (NKB)	$\sum_{i=0}^{n-1} b_i 2^i;$
moduł ze znakiem (MZ)	$(-1)^{b_{n-1}} \cdot \sum_{i=0}^{n-2} b_i 2^i;$
uzupełnieniowy do dwójki (U2)	$-b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i;$
uzupełnieniowy do jedynek (U1)	$\sum_{i=0}^{n-2} (b_i - b_{n-1}) 2^i;$
spolaryzowany (ang. biased)	$-B + \sum_{i=0}^{n-1} b_i 2^i;$
BCD (ang. binary coded decimal) dla n podzielnego przez 4	$\sum_{j=0}^{n/4-1} \sum_{i=0}^{3} b_{i+4j} 2^i 10^j.$

Poniższa tabela zawiera przykłady dla $n = 8$.

$b_7\ldots b_1b_0$	NKB	MZ	U2	U1	$B = 127$	BCD
00000000	0	0	0	0	−127	0
00000001	1	1	1	1	−126	1
01111110	126	126	126	126	−1	–
01111111	127	127	127	127	0	–
10000000	128	0	−128	−127	1	80
10000001	129	−1	−127	−126	2	81
11111110	254	−126	−2	−1	127	–
11111111	255	−127	−1	0	128	–

3.2 Liczby zmiennopozycyjne

Obecnie najczęściej stosowana jest norma binarnej arytmetyki zmiennopozycyjnej IEEE-754. Definiuje ona m.in. 32-bitowy format pojedynczy i 64-bitowy format podwójny. W formacie pojedynczym bity 0–22 tworzą część ułamkową mantysy M, bity 23–30 tworzą wykładnik W, bit 31 koduje znak S, a przesunięcie wykładnika (polaryzacja) wynosi $B = 127$. W formacie podwójnym bity 0–51 tworzą część ułamkową mantysy M, bity 52–62 tworzą wykładnik W, bit 63 koduje znak S, a przesunięcie wykładnika (polaryzacja) wynosi $B = 1023$. Wartość liczby można określić za pomocą poniższych reguł.

pole wykładnika	pole mantysy	rodzaj liczb	wartość
$1 \leqslant W \leqslant 11\ldots10_2$		liczby znormalizowane	$(-1)^S \cdot 1{,}M \cdot 2^{W-B}$
$W = 0$	$M \neq 0$	liczby zdenormalizowane	$(-1)^S \cdot 0{,}M \cdot 2^{1-B}$
$W = 0$	$M = 0$	zera	0
$W = 11\ldots11_2$	$M = 0$	nieskończoności	$(-1)^S \cdot \infty$
$W = 11\ldots11_2$	$M \neq 0$	nieliczby aktywne i pasywne, wartość nieokreślona	nan, NaN

3.3 Napisy

Do najczęściej stosowanych metod kodowania napisów należą:

- ASCII (American Standard Code for Information Interchange) i jego rozszerzenia np. ISO 8859-1, ISO 8859-2, ..., cp1250 itd.;
- EBCDIC (Extended Binary Coded Decimal Interchange Code);
- UTF-8 (8-bit Unicode Transformation Format).

UTF-8 jest sposobem kodowania znaków zdefiniowanych w normach Unicode (np. ISO/IEC 10646 Universal Multiple-Octet Coded Character Set).

zakres znaków (szesnastkowo)	liczba bitów	sekwencja oktetów UTF-8
0–7F	7	0xxxxxxx
80–7FF	11	110xxxxx 10xxxxxx
800–FFFF	16	1110xxxx 10xxxxxx 10xxxxxx
10000–10FFFF	21	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

Dokładniejszy opis można znaleźć w RFC 3629. Zauważmy, że oktety o wartościach szesnastkowych C0, C1, F5 do FF nie występują w tym kodowaniu.

3.4 Porządek bajtów

W przyrodzie występują maszyny:

- cienkokońcówkowe (ang. little-endian);
- grubokońcówkowe (ang. big-endian);
- dwukońcówkowe (ang. bi-endian).

3.5 Porządek bitów

Bity mogą być numerowane od najmniej znaczącego do najbardziej znaczącego lub odwrotnie. Numeracja bitów może zaczynać się od 0 albo od 1. Kolejność bitów jest szczególnie istotna przy szeregowym przesyłaniu danych.

3.6 Wyrównywanie danych

Wyrównywanie danych stosuje się w celu zwiększenia wydajności. Dane reprezentujące typ prosty są umieszczane pod adresem podzielnym przez jego rozmiar. Struktury są wyrównywane w taki sposób, żeby mogły być elementami tablicy. Pola struktury są wyrównywane zgodnie z ich rozmiarami.

3.7 Przedroski w informatyce

Przedrostki kilo, mega, giga, tera oznaczają odpowiednio wielokrotności 10^3, 10^6, 10^9, 10^{12} jednostek podstawowych. W informatyce mogą one też oznaczać odpowiednio 2^{10}, 2^{20}, 2^{30}, 2^{40}. Zdarza się też mieszanie potęg 2 i 10. Obowiązują pewne zwyczajowe reguły:

- Wielkość pamięci określa się, stosując potęgi 2.

- Przepływność określa się, stosując na ogół potęgi 10: w modemie 56 kb/s to 56000 b/s, w Ethernecie 100 Mb/s to 10^8 b/s, ale np. przepływność łącza oznaczanego E1 (czasem S2) często określa się jako 2 Mb/s, podczas gdy rzeczywista wartość wynosi 2048000 b/s i poprawnie należałoby napisać 2,048 Mb/s.
- Pojemności dysków twardych określa się, stosując potęgi 10.
- Przy określaniu pojemności innych pamięci masowych panuje bałagan: dla dyskietki 1,44 MB mega oznacza $10^3 \cdot 2^{10}$, dla płyty CD mega jest bliższe 2^{20}, dla płyty DVD giga jest bliższe 10^9.

4 Mikroprocesor

4.1 Długość słowa, rozmiar szyny danych i szyny adresowej

Słowo maszynowe to podstawowa porcja przetwarzania informacji w systemach komputerowych, określona przez rozmiar rejestrów procesora. Jest to jeden z podstawowych parametrów zarówno komputera jak i procesora, tzw. „bitowość". Większość operacji wewnątrz procesora oraz przesyłanie danych między procesorem a innymi składnikami systemu komputerowego są wykonywane na słowach (przynajmniej tak widzi to programista). Aby zwiększyć elestyczność oprogramowania, procesor zwykle umożliwia też wykonywanie operacji na argumentach, których rozmiar jest podwielokrotnością lub wielokrotnością rozmiaru słowa. Obecnie słowo maszynowe ma zwykle rozmiar 8, 16, 32 lub 64 bity. W przeszłości konstruowano również komputery o innych długościach słowa maszynowego, przykładowo: 4, 12, 24, 36, 39, 40 bitów.

Rozmiar szyny danych, podawany w bitach, określa w jakich maksymalnie porcjach dane mogą być przesyłane z i do pamięci operacyjnej. Wąskim gardłem jest zwłaszcza szybkość przesyłania danych i kodu programu z pamięci do procesora. Stąd obserwuje się tendencję do stosowania coraz szerszych szyn danych.

Rozmiar szyny adresowej, również podawany w bitach, określa maksymalny rozmiar fizycznej pamięci operacyjnej, jaka może być adresowana.

Poniższa tabela zawiera zestawienie mikroprocesorów o różnych długościach słowa maszynowego. Zestawienie to świadczy o pewnej umowności tego pojęcia i braku jego związku z rozmiarami szyn danych i adresowej.

długość słowa	procesor	rozmiary argumentów operacji (rejestrów)	szerokość szyny danych	szerokość szyny adresowej	fizyczna przestrzeń adresowa
8	8080	8, (16)	8	16	64 kB
8	Z80	8, 16	8	16	64 kB
16	8088	8, 16	8	20	1 MB
16	8086	8, 16	16	20	1 MB
16	68000	8, 16, 32	16	24	16 MB
32	386SX	8, 16, 32	16	24	16 MB
32	386DX	8, 16, 32	32	32	4 GB
32	P4	8, 16, 32, 64, 80, 128	64	36	64 GB
64	P4 EM64T	8, 16, 32, 64, 80, 128	64	36	64 GB
64	PowerPC 620	8, 16, 32, 64	64, 128	40	1 TB
64	Opteron	8, 16, 32, 64, 80, 128	64, 128	40	1 TB

4.2 Rejestry

Możemy wyróżnić następujące rodzaje rejestrów:

- danych – służą tylko do przechowywania argumentów i wyników operacji arytmetyczno-logicznych;
- adresowe – służą do przechowywania adresów i do operacji arytmetycznych na adresach;
- ogólnego przeznaczenia – mogą pełnić rolę zarówno rejestrów danych, jak i adresowych;
- specjalizowane – pełnią ściśle określoną funkcję, np. akumulator, czy wskaźnik stosu;
- stanu, znaczników – przechowuje jednobitowe znaczniki stanu i sterowania procesora;
- licznik programu – zawiera adres bieżącej instrukcji, która ma być wykonana;

- zmiennopozycyjne – służą do przechowywania liczb w formacie zmiennopozycyjnym;
- wektorowe – przechowują wektor (tablicę) wartości;
- segmentowe – służą do implementacji segmentowanego modelu pamięci.

Ponadto mogą występować różnego rodzaju rejestry, które nie są widoczne dla programisty lub są widoczne tylko w trybie uprzywilejowanym. Są to rejestry służące do:

- zarządzania pamięcią;
- debugowania.

Ze względu na liczbę i sposób używania rejestrów możemy wyróżnić następujące architektury:

- bezrejestrowe – Rejestry nie przechowują danych, służą wyłącznie do przechowywania adresów. Każda operacja na danych wymaga dostępu do pamięci.
- minimalny zestaw rejestrów – Są to najczęściej akumulator, wskaźnik stosu, licznik programu i rejestr adresowy. Przy operacjach dwuargumentowych jeden argument jest w pamięci, a drugi argument i wynik są zapamiętywane w akumulatorze. Przykładem była architektura komputera IAS zaproponowana przez von Neumanna.
- mały zestaw rejestrów specjalizowanych – Na ogół jest to kilka rejestrów o sztywno określonej funkcji. Przykładem jest architektura x86 w trybie 16-bitowym, posiadająca następujące rejestry: AX, DX, CX, BX, SP, BP, SI, DI, IP, FLAGS.
- mały zestaw rejestrów uniwersalnych – Podobna liczba rejestrów jak wyżej, ale każdy rejestr może wystąpić w dowolnej funkcji. Architektura x86 w trybie 32-bitowym (IA-32) z rejestrami EAX, EDX, ECX, EBX, ESP, EBP, ESI, EDI, EIP, EFLAGS jest prawie dobrym przykładem.
- duży zestaw rejestrów uniwersalnych – Typowo występuje od 32 do 128 rejestrów. Przykładem jest architektura Cray X1E.
- bufor wierzchołka stosu – Umożliwia przechowywanie jednej do kilku ramek stosu w szybkiej pamięci rejestrowej i wykonanie procedury bez odwołań do pamięci.
- stosowy zestaw rejestrów – Operacje wykonywane są na wierzchołku stosu. Struktura ta jest przystosowana do implementacji odwrotnej notacji polskiej. Przykładem jest jednostka zmiennopozycyjna x87.

Jak zobaczymy później, liczba i rodzaj fizycznie istniejących w procesorze rejestrów może się różnić od tego, co widzi programista.

4.3 Kod maszynowy a Asembler

Kod maszynowy jest binarnym sposobem opisu rozkazów procesora. Każdy typ procesora ma swój unikalny kod maszynowy. Programowanie przez bezpośrednie użycie kodu maszynowego jest niewygodne. Dlatego wymyślono *Asembler*, który jest językiem programowania niskiego poziomu. Jedno polecenie Asemblera odpowiada zwykle jednemu rozkazowi maszynowemu. Każda architektura ma swój unikalny Asembler. Nazwą *asembler* określa się też program tłumaczący Asembler na kod maszynowy.

4.4 Lista instrukcji

Instrukcje procesora możemy podzielić na aplikacyjne, które mogą być zawsze wykonane i systemowe, które mogą być wykonane tylko w trybie uprzywilejowanym. O trybie uprzywilejowanym będzie mowa w dalszej części. Instrukcje aplikacyjne możemy podzilić na:

- instrukcje przesyłania danych;
- instrukcje arytmetyczne;
- operacje bitowe;

- operacje na napisach;
- instrukcje sterujące wykonaniem programu.

Instrukcje sterujące wykonaniem programu, czyli instrukcje warunkowe, mogą być instrukcjami skoku, warunkowego przypisania lub warunkowego wykonania jakiejś operacji. Instrukcje warunkowego wykonania operacji umożliwiają implementację bez skoków krótkich sekwencji if-else, co może wydatnie poprawić szybkość działania programu. Instrukcje warunkowe są realizowane wg jednego z następujących modeli:

- ze znacznikami – Operacja warunkowa jest realizowana dwufazowo, pierwsza instrukcja modyfikuje jednobitowy znacznik lub kilka jednobitowych znaczników, druga instrukcja wykonuje skok warunkowy w zależności od wartości znacznika lub znaczników. Model ten umożliwia rozdzielenie w czasie wykonania tych instrukcji, druga instrukcja nie musi występować w kodzie programu bezpośrednio po pierwszej. Przykładem tego modelu są architektury x86, IA-32, x86-64.
- bez znaczników – Jedna instrukcja sprawdza warunek i w zależności od jego wyniku wykonuje skok. Model ten jest popularny w architekturach RISC.
- z predykatami – Predykat jest uogólnionym znacznikiem, mogącym przechowywać wartość logiczną wcześniej obliczonego warunku. Instrukcje warunkowe specyfikują numer predykatu. Model ten został zaimplementowany w architekturze IA-64.

Można wyróżnić dwa style realizacji dwuargumentowych operacji arytmetycznych (dodawanie, mnożenie itp.):

- jeden z argumentów źródłowych jest jednocześnie argumentem wynikowym,
  np. `add r1, r2` oznacza r1 := r1 + r2 lub r2 := r1 + r2;
- argument wynikowy może być różny od argumentów źródłowych,
  np. `add r1, r2, r3` oznacza r1 := r2 + r3 lub r3 := r1 + r2.

4.5 Tryby adresowania argumentów

Tryb adresowania określa sposób specyfikacji argumentu operacji.

Nazwa i opis	Typowe oznaczenia	Przykłady w IA-32
natychmiastowy (argument w kodzie instrukcji)	`7`	`mov dword [1000], 7`
bezpośredni (adres argumentu w kodzie instrukcji)	`[1000]`	`mov dword [1000], 7`
pośredni (adres argumentu w pamięci)	`[[1000]]`	nie występuje
rejestrowy bezpośredni (argument w rejestrze)	r1	`mov eax, [ebx]`
rejestrowy pośredni (adres argumentu w rejestrze)	`[r1]`	`mov eax, [ebx]`
indeksowy	`[r1+8]`, `[r1+r2]`	`mov eax, [ebx+4*ecx+8]`
stosowy (względem wartości wskaźnika stosu)	`[sp]`, `[sp+4]`	`push eax; pop eax;` `mov [esp+2*ecx+4], eax`
względny (względem wartości licznika rozkazów)	`ip+8`, `pc+9`	`jc przesunięcie`
rejestrowy pośredni z preinkrementacją	`[+r1]`	
rejestrowy pośredni z postinkrementacją	`[r1+]`	`movs; pop eax`
rejestrowy pośredni z predekrementacją	`[-r1]`	`loop; push eax`
rejestrowy pośredni z postdekrementacją	`[r1-]`	`movs`

W zasadzie w każdym mikroprocesorze występują tryby adresowania natychmiastowy, rejestrowy bezpośredni i rejestrowy pośredni. W większości współczesnych mikroprocesorów występuje jakiś wariant trybu indeksowego. Tryb względny też występuje w wielu mikroprocesorach, ale głównie w instrukcjach skoków.

4.6 Cykl pracy mikroprocesora

Mikroprocesory są cyfrowymi układami synchronicznymi. Ich praca jest synchronizowana sygnałem zegarowym. Mikroprocesor, z punktu widzenia programisty, wykonuje zaprogramowane instrukcje w sposób sekwencyjny.

Wykonywanie programu przez pierwsze mikroprocesory można podzielić na następujące cykle. *Cykl rozkazowy* (ang. instruction cycle) jest okresem czasu potrzebnym do pobrania i wykonania pojedynczej instrukcji. Cykl

ten, zależnie od instrukcji, składa się z jednego do kilku *cykli procesora* (ang. machine cycle). Każdy odczyt lub zapis pamięci wymaga jednego cyklu procesora. Każdy cykl procesora składa się z kilku *taktów zegara* (ang. clock period). Wykonanie instrukcji może zabrać dodatkowy cykl procesora lub może być zawarte w jednym z cykli dostępu do pamięci.

We współczesnych mikroprocesorach występuje wyraźne oddzielenie fazy komunikacji z pamięcią od fazy wykonania instrukcji, tak aby np. możliwe było pobieranie kolejnych instrukcji przed zakończeniem wykonania poprzednich. Mamy *cykl współpracy z pamięcią*, składający się z jednego do kilku taktów zegara oraz *cykl wykonania instrukcji*, składający się z jednego do nawet kilkuset cykli zegara. Przy czym częstotliwości obu zegarów mogą być różne.

4.7 Pomiar wydajności

Wydajność procesorów na ogół mierzy się za pomocą następujących jednostek:

- IPS (instructions per second) – obecnie zwykle używa się wielokrotności MIPS = $10^6 \cdot$ IPS;
- FLOPS (floating point operations per second) – obecnie zwykle używa się wielokrotności MFLOPS = $10^6 \cdot$ FLOPS, GFLOPS = $10^9 \cdot$ FLOPS, TFLOPS = $10^{12} \cdot$ FLOPS itd.

4.8 Metody zwiększania wydajności

Naturalny pomysł zwiększania wydajności obliczeniowej procesora polegający na zwiększaniu częstotliwości taktowania jest ograniczony przez różnego rodzaju bariery technologiczne:

- wydzielanie ciepła;
- czas propagacji sygnałów w układach cyfrowych.

Zwiększenie wydajności uzyskuje się, stosując następujące rozwiązania:

- pobieranie instrukcji na zakładkę;
- kolejka (bufor) instrukcji;
- przetwarzanie potokowe;
- zrównoleglanie wykonywania instrukcji.

Kolejny istotny problem hamujący wzrost wydajności systemów komputerowych jest spowodowany nienadążaniem wzrostu szybkość pracy pamięci za wzrostem szybkości pracy procesorów. Uwidacznia się to w postaci różnych częstotliwości taktowania i różnych napięć zasilających układy wewnętrzne procesora i jego układy współpracy z szynami. Możliwe rozwiązania tego problemu to:

- stosowanie wielopoziomowych pamięci podręcznych, początkowo zewnętrznych, a obecnie wykonywanych w jednym układzie scalonym z procesorem;
- zwiększanie szerokości szyny danych;
- konstrukcja pamięci umożliwiających pobieranie danych co jeden cykl zegara, a nawet dwa razy w jednym cyklu zegara.

4.9 Przetwarzanie potokowe

Zasadnicza idea *przetwarzania potokowego* polega na rozłożeniu wykonania pojedynczej instrukcji na ciąg prostych etapów, z których każdy może być wykonany w jednym cyklu zegara. Wykonywanie instrukcji przypomina taśmę produkcyjną. W każdym cyklu zegara instrukcje przemieszczają się wzdłuż potoku do następnego etapu. Mimo, że wykonanie jednej instrukcji może wymagać wielu etapów, to w każdym cyklu rozpoczyna się i kończy wykonywanie jednej instrukcji. W praktyce taki wyidealizowany model nie może być zrealizowany. Powodem jest występowanie zależności między danymi, zależności sterowania i zależności zasobów. Ponadto poszczególne

instrukcje mogą wymagać różnej liczby etapów. Trzeba też uwzględnić obsługę sytuacji wyjątkowych. Czas wykonania instrukcji przy przetwarzaniu potokowym jest dłuższy niż przy jej wykonaniu w sposób sekwencyjny, ale średni czas wykonania jednej instrukcji przy wykonywaniu dostatecznie długiego ciągu instrukcji jest krótszy.

Rozróżniamy następujące typy zależności między danymi:

- RAW – odczyt po zapisie

  load-use
  ```
  mov eax, 7
  add ebx, eax
  ```
  define-use
  ```
  add eax, 3
  mul ebx
  ```

- WAR – zapis po odczycie
  ```
  add eax, ebx
  mov ebx, 4
  ```

- WAW – zapis po zapisie
  ```
  push eax
  sub esp, 16
  ```
  (w tym przykładzie występuje też zależność RAW)

Zależności typu RAW usuwa się przez *data bypassing* i *data forwarding*. Zależności typu WAR i WAW są zależnościami fałszywymi. Usuwa się je przez *przemianowanie rejestrów*. Fizycznych rejestrów jest więcej, niż widzi programista i są one przydzielane w miarę potrzeby.

Zależności sterowania rozwiązuje się przez:

- pobieranie docelowego rozkazu z wyprzedzeniem;
- równoległe przetwarzanie obu gałęzi programu;
- predykcja skoków (przewidywanie rozgałęzień):
  - zawsze następuje skok,
  - nigdy nie następuje skok,
  - decyduje kod instrukcji (kompilator),
  - jak przy ostatnim wykonaniu,
  - tablica historii skoków;
- bufor pętli;
- opóźnione rozgałęzianie.

Przy przetwarzaniu potokowym stosuje się podział skomplikowanych instrukcji na *mikrooperacje* (ang. μops). Mikroperacje jest łatwiej potokować.

4.10 Architektury superskalarna i wektorowa, procesory z bardzo długim słowem instrukcji

Przetwarzanie potokowe umożliwia zbliżenie się do granicy wykonywania średnio jednej (prostej) instrukcji (mikrooperacji) w jednym cyklu zegara. Dalsze przyspieszenie możliwe jest przez zrównoleglenie wykonywania instrukcji. Zrównoleglenie wymaga zwiększenia liczby potoków i jednostek wykonawczych (ALU – arithmetic-logic unit, AGU – address generator unit, FPU – floating point unit, LSU – load-store unit itp.). Poszczególne potoki mogą być specjalizowane. Można wyróżnić następujące sposoby wykorzystania tej równoległości (wielopotokowości):

- architektura superskalarna – zrównoleglenie na poziomie wykonania – układy wewnątrz procesora decydują o zrównolegleniu (hyper threading też tu podpada);
- architektura wektorowa (znana też jako SIMD – single instruction multiple data) – zrównoleglenie na poziomie algorytmu – nie wszystkie algorytmy dają się efektywnie zwektoryzować;
- procesory z bardzo długim słowem instrukcji (VLIW – very long instruction word) – zrównoleglenie na poziomie kompilatora – pojedyncza instrukcja opisuje, co mają robić poszczególne jednostki wykonawcze.

Przy wielu potokach mamy do czynienia z następującymi zagadnieniami:

- wydawanie instrukcji w innej kolejności niż zapisano w programie – scheduler;
- kończenie instrukcji w innej kolejności niż zapisano w programie – re-order buffer.

4.11 Architektury RISC i CISC

RISC – Reduced Instruction Set Computers	CISC – Complex Instruction Set Computers
Zawierają ograniczony, prosty zbiór instrukcji.	Występują skomplikowane instrukcje wspierające języki wysokiego poziomu.
Zawierają dużą liczbę uniwersalnych rejestrów.	Zawierają małą liczbą rejestrów i/lub rejestry specjalizowane.
Instrukcje arytmetyczno-logiczne wykonywane są na rejestrach.	Instrukcje arytmetyczno-logiczne mogą pobierać argumenty z pamięci i umieszczać wynik w pamięci.
Kody instrukcji są stałej długości, typowo 4 bajty, i mają stałe rozmieszczenie pól, co ułatwia dekodowanie.	Kody instrukcji mają zmienną długość, typowo od jednego do kilkunastu bajtów. Występuje prefiksowanie instrukcji utrudniające dekodowanie.
Posiadają małą liczbę trybów adresowania.	Posiadają dużą liczbę trybów adresowania.
Dozwolone jest tylko adresowanie wyrównane.	Dozwolone jest adresowanie niewyrównane.

4.12 System przerwań

- Przerwania sprzętowe
  - maskowalne
  - niemaskowalne
- Przerwania programowe
- Praca krokowa
- Wyjątki
- Tablica przerwań

4.13 Sprzętowe wsparcie dla systemów operacyjnych

- Translacja adresów
  - segmentacja
  - stronicowanie, prosta i odwrotna tablica stron
- Poziomy ochrony
  - wewnętrzny, nadzorcy, uprzywilejowany
  - zewnętrzny, aplikacji
- Wywoływanie usług systemu operacyjnego
  - przerwania
  - specjalna instrukcja (syscall)

4.14 Urządzenia wejścia-wyjścia

- Metody adresowania urządzeń wejścia-wyjścia
- Transmisja procesorowa (PIO) i bezpośredni dostęp do pamięci (DMA)
- Sterowniki urządzeń

5 Pamięci

Hierachia pamięci:

- rejestry procesora;
- pamięć podręczna, do trzech poziomów;
- pamięć operacyjna, pamięć główna;
- pamięć wirtualna, zrealizowana w oparciu o zewnętrzną pamięć masową;
- system plików, zewnętrzna pamięć masowa;
- nośniki wymienne, zasoby sieciowe.

W górę tej hierarchii rośnie szybkość, a w dół rośnie pojemność.

5.1 Pamięci półprzewodnikowe

Pamięci półprzewodnikowe można podzielić z uwzględnieniem wielu kryteriów:

- trwałość przechowywanej informacji
  - ulotne
  - nieulotne
- spsób adresowania
  - adresowanie poszczególnych bajtów lub słów
  - bezadresowe (np. pamięci FIFO)
  - adresowanie zawartością, asocjacyjne
- liczba magistral
  - jednobramowe, jednodostępne
  - wielobramowe, wielodostępne
- sposób wprowadzania i wyprowadzania informacji
  - równoległe
  - szeregowe
  - szeregowo-równoległe
- rodzaj synchronizacji
  - asynchroniczne
  - synchroniczne
- organizacja wewnętrzna
  - liczba matryc
  - rozmiar matrycy
  - liczba bitów pamiętana w pojedynczej komórce

Parametry użytkowe pamięci:

- pojemność
- szybkość dostarczania danych
- koszt na jeden bit

- ziarnistość – minimalna wielkość pamięci, o którą można zwiększyć pamięć systemu
- organizacja (zewnętrzna) – liczba bitów w kości razy liczba kości w module razy liczba adresów w kości razy liczba modułów
- czas dostępu – najmniejszy przedział czasu potrzebny do odczytania lub zapisania pojedynczej porcji (słowa) informacji
- czas cyklu – najmniejszy przedział czasu między początkiem dwóch kolejnych operacji dostępu do pamięci

5.1.1 Pamięci nieulotne

Pamięci nieulotne zachowują zawartość po wyłączeniu zasilania. W pamięciach nieulotnych czas zapisu informacji jest dłuższy od czasu odczytu, czasem nawet o kilka rzędów wielkości. Pamięci nieulotne dzieli się ze względu na sposób zapisu informacji oraz możliwość jej zmiany:

- MROM, ROM (mask programmable read only memory) – zawartość jest ustalana w czasie produkcji;
- PROM (programmable read only memory) – zawartość może być zaprogramowana jednokrotnie przez użytkownika przez przepalenie połączeń wewnętrznych;
- UV-EPROM, EPROM (erasable programmable read only memory) – zawartość może być wielokrotnie programowana przez użytkownika, programowanie przez umieszczenie ładunku elektrycznego w izolowanej bramce tranzystora, kasowanie za pomocą światła ultrafioletowego;
- OTPROM (one time programmable read only memory) – wersja UV-EPROM w taniej obudowie bez okienka do kasowania, może być zaprogramowana tylko jednokrotnie;
- EEPROM (electrically erasable programmable read only memory) – zawartość *pojedynczej komórki* może być zmieniana za pomocą sygnałów elektrycznych;
- FLASH – zawartość *całego sektora* może być zmieniana za pomocą sygnałów elektrycznych.

5.1.2 Pamięci o dostępie swobodnym

Omawiane w tym podrozdziale układy RAM (ang. random access memory) są pamięciami ulotnymi, gdyż tracą zawartość po wyłączeniu zasilania. Czasy zapisu i odczytu w tych pamięciach są jednakowe. Pamięci o dostępie swobodnym są wytwarzane w dwóch technologiach:

- pamięci statyczne (ang. SRAM);
- pamięci dynamiczne (ang. DRAM).

Na ustalonym etapie rozwoju technologii układy SRAM są szybsze od układów DRAM, ale mają większy koszt na jeden bit. Wynika to z faktu, że w układach SRAM potrzeba 6 tranzystorów (oraz połączenia między nimi) do zapamiętania jednego bitu, a w układach DRAM potrzebny jest jeden tranzystor i jeden kondensator. Dlatego układy DRAM są stosowane do konstrukcji pamięci głównych komputerów, a układy SRAM do konstrukcji pamięci podręcznych. W układach SRAM raz zapisna informacja pozostaje niezmieniona aż do zapisania nowej informacji lub wyłączenia zasilania. Natomiast w układach DRAM zapisana informacja ulega samozniszczeniu po pewnym czasie i w każdym procesie odczytu. Stąd w układach DRAM informacja musi być cyklicznie odświeżana. Współcześnie produkowane DRAM zawierają wbudowany układ odświeżania.

Wzrost szybkości pamięci DRAM nie nadąża za wzrostem szybkości procesorów. Wymusza to stosowanie zaawansowanych rozwiązań układowych:

- FPM DRAM (fast page mode DRAM), EDO DRAM (extended data out DRAM);
- SDRAM (synchronous DRAM);
- VRAM (video RAM), SGRAM (synchronous graphisc RAM);
- DDR SDRAM (double data rate DRAM), DDR2 SDRAM (double data rate two SDRAM);
- Rambus DRAM.

5.1.3 Pamięci podręczne

Sposób odwzorowania adresów pamięci wyższego poziomu na adresy pamięci podręcznej:

- bezpośredni, 1-skojarzeniowy;
- sekcyjno-skojarzeniowy, n-skojarzeniowy ($n > 1$);
- skojarzeniowy, w pełni skojarzeniowy, asocjacyjny.

Zjawisko migotania pamięci podręcznej.

Przy odwzorowaniu innym niż bezpośrednie stosuje się następujące algorytmy zastępowania:

- LRU (least recently used, najdawniej ostatnio używany) – zastępowany jest ten blok, który pozostawał w pamięci podręcznej najdłużej bez odwoływania się do niego;
- FIFO (first in first out, pierwszy wchodzi pierwszy wychodzi) – zastępowany jest ten blok, który pozostawał w pamięci podręcznej najdłużej;
- LFU (least frequently used, najrzadziej używany) – zastępowany jest ten blok, do którego było najmniej odwołań;
- zastępowany jest losowo wybrany blok.

Sposób uzgadniania zawartości pamięci wyższego poziomu z pamięcią podręczną podczas zapisu:

- zapis jednoczesny (ang. write through);
- zapis opóźniony (ang. copy/write back);
- zapis *inclusive*;
- zapis *exclusive*.

Algorytmy zapewniania zgodności pamięci podręcznych:

- MESI;
- MSI;
- MOSI;
- MOESI.

Poszczególne litery w nazwie algorytmu oznaczają dopuszczalny stan, w którym może znaleźć się blok (linia) pamięci podręcznej:

- M – modified – mam ważną kopię bloku, który został zmodyfikowany;
- O – owned – jestem właścicielem, ponoszą odpowiedzialność za wszystkie odwołania do tego bloku;
- E – exclusive – mam ważną kopię bloku, który nie został zmodyfikowany;
- S – shared – wszyscy mają kopię tego bloku;
- I – invalid – blok zawiera nieważne dane.

5.2 Pamięci masowe

Pamięci masowe nazywane są też pamięciami zewnętrznymi. Pamięć masowa ma większą pojemność i dłuższy czas dostępu niż pamięć operacyjna. Z punktu widzenia procesora pamięć masowa jest urządzeniem wejścia-wyjścia. Oznacza to, że procesor nie posiada instrukcji bezpośrednio operujących zawartością tej pamięci, tak jak zawartością pamięci operacyjnej. Inną wspólną cechą pamięci masowych, odróżniającą je od pamięci operacyjnych, jest operowanie nie na pojedynczych bajtach a na blokach, sektorach danych o typowym rozmiarze od 512 bajtów do kilkudziesięci kilobajtów. Z uwagi na stosowane technologie wyróżniamy obecnie następujące rodzaje pamięci masowych:

- magnetyczne: dyskowe (dyski twarde, dyskietki) i taśmowe;
- optyczne:
  - Compact Disc: CD−ROM, CD−R, CD−RW;
  - Digital Video/Versatile Disc: DVD−ROM, DVD−R, DVD−RW, DVD+R, DVD+RW, DVD−RAM;
  - Blu-ray Disc: BD;
  - High Definition DVD: HD DVD;
- magnetyczno-optyczne.

Ze względu na sposób dostępu, do pamięci zewnętrznych (masowych) należałoby też zaliczyć karty pamięci FLASH (pen drive, flash drive, USB key), które są pamięciami półprzewodnikowymi.

Budowa dysku twardego:

- dyski wykonywane są z aluminium lub szkła i pokrywane materiałem magnetycznym;
- w jednej hermetycznej obudowie może być umieszczonych na wspólnej osi od jednego do kilku wirujących dysków;
- zapis i odczyt wykonywane są za pomocą zespołu głowic poruszających się w przybliżeniu wzdłuż promienia dysków;
- do przechowywania informacji może być wykorzystana jedna lub obie strony każdego dysku, łączna liczba wykorzystanych stron jest równa liczbie głowic;
- dane rozmieszczone są na koncentrycznych ścieżkach (cylindrach);
- każda ścieżka podzielona jest na sektory, liczba sektorów na ścieżce może być stała lub zmienna;
- format sektora:
  - przerwa;
  - identyfikacja – synchronizacja, nr ścieżki, nr sektora, korekcja błędów;
  - przerwa;
  - dane – synchronizacja, właściwe dane, korekcja błędów;
  - przerwa;
- czas dostępu to suma czasu koniecznego do przesunięcia głowicy, czyli znalezienia odpowiedniej ścieżki, czasu związanego z opóźnieniem obrotowym i czasu potrzebnego do przesłania danych;
- stosuje się przeplot dla skrócenia czasów dostępu.

Zapis i odczyt informacji na dyskach optycznych wykonywany jest za pomocą lasera. W dyskach typu ROM informacja zapisywana jest przez wykonywanie za pomocą matrycy zagłębień (ang. pit) w stosunku do otaczającej powierzchni materiału (ang. land). W dyskach zapisywalnych wykorzystuje się zjawisko zmiany własności optycznych nośnika wskutek naświetlania laserem o zmiennej mocy promieniowania. Moc promieniowania używanego do zapisu jest większa niż moc używana do odczytu. Wprowadzenie kolejnej technologii dysków optycznych wiązało się z opracowaniem tanich laserów o coraz krótszej długości fali promieniowania: 780 nm dla CD, 650 nm dla DVD, 405 nm dla BD i HD DVD. Na większości dysków optycznych stosuje się jedną spiralną ścieżkę. Wyjątkiem jest np. DVD−RAM posiadający organizację podobną do dysków twardych, czyli koncentryczne

ścieżki podzielone na sektory o jednakowym rozmiarze kątowym. Stosowanie jednej spiralnej ścieżki zwiększa gęstość zapisu kosztem zwiększenia czasu dostępu.

Odczyt informacji z dysków magnetyczno-optycznych wykonywany jest za pomocą lasera. Wykorzystuje się zjawisko zmiany własności optycznych ośrodka magnetycznego w zależności od kierunku namagnesowania. Do zapisu informacji wykorzystuje się przejście między ferromagnetykiem a paramagnetykiem w temperaurze Curie. Laser podgrzewa lokalnie warstwę magnetyczną powyżej temperatury Curie, a głowica magnetyczna wytwarza odpowiednie pole magnetyczne. Obniżenie temperatury poniżej punktu Curie powoduje utrwalenie ustawienia domen magnetycznych w nośniku. Na dyskach magnetyczno-optycznych stosuje się organizację podobną do stosowanej na dyskach twardych.

W pamięciach masowych stosuje się następujące metody zapisu informacji oraz metody ich zapisywania lub odczytywania:

- CAV (Constant Angular Velocity), stała prędkość kątowa, stosowana w dyskach twardych oraz wielu komputerowych napędach nośników optycznych dla zmniejszenia czasu dostępu;
- CLV (Constant Linear Velocity), stała prędkość liniowa, zastosowana np. w pierwotnej specyfikacji CD (1,2 m/s);
- ZCLV (Zoned CLV), dysk podzielony jest na strefy o stałych prędkościach liniowych, zastosowana np. w specyfikacji DVD;
- CAA (Constant Angular Acceleration), odmiana CLV, w której prędkość kątowa zmienia się krokowo ze stałym przyspieszeniem (opóźnieniem), stosowana w odtwarzaczach CD w celu wyeliminowania pewnych artefaktów powodowanych przez metodę CLV.

Uwaga, w nośnikach optycznych należy wyraźnie odróżnić opisaną w specyfikacji metodę zapisu (przechowywania) informacji na nośniku od stosowanej przez napęd metody jej zapisu lub odczytu. Dysk CD, którego specyfikacja określa stałą prędkość liniową zapisu informacji, może być odczytywany (i zapisywany) ze stałą prędkością kątową, co oznacza, że szybkość odczytu (zapisu) jest zmienna w funkcji promienia.

5.3 RAID

Patrz np. [1] lub [9].

6 Magistrale i interfejsy

Składniki komputera komunikują się za pomocą *magistral*, nazywanych też *szynami* (ang. bus). Komputer może zawierać wiele magistral, które mogą być połączone hierarchicznie. Magistrala może składać się z wielu równoległych linii sygnałowych. Każdą linią mogą być przesyłane sygnały binarne. Do magistrali może być podłączonych wiele urządzeń. Tylko jedno z tych urządzeń może sterować liniami magistrali w danym momencie.

Linie magistralowe mogą być:

- dedykowane – trwale przypisane do jednej funkcji,
- multipleksowane – funkcja jest określana przez stan linii sterujących – multipleksowanie czasowe.

Podłączenie wielu urządzeń do magistrali wymaga zastosowania arbitrażu, który może być:

- centralny,
- rozproszony.

Ze względu na sposób koordynacji czasowej magistrale dzielimy na:

- synchroniczne,
- asynchroniczne.

Urządzenia wchodzące w skład systemu komputerowego komunikują się za pomocą *interfejsów*. Pojęcie interfejsu bywa używane zamiennie z pojęciem *łącza*. Należy rozróżniać pewną subtelną różnicę między *łączem* a *złączem*.

Interfejsy (łącza) dzielimy na:

- szeregowe,
- równoległe.

Interfejsy mogą być:

- dwupunktowe (ang. point-to-point) – łączą dwa urządzenia,
- wielopunktowe (ang. multi-point) – łączą wiele urządzeń, są często magistralami.

Tryby pracy interfejsów:

- simplex – transmisja jednokierunkowa,
- half duplex – transmisja dwukierunkowa naprzemienna,
- full duplex – transmisja dwukierunkowa jednoczesna.

6.1 Magistrala systemowa

We współczesnych komputerach magistrala systemowa łączy procesor(y) z pamięcią główną i sterownikami innych magistral (np. PCI). Z każdym modelem procesora związana jest specyficzna dla niego magistrala systemowa. Magistrala systemowa zawiera:

- linie danych,
- linie adresowe,
- linie sterowania.

6.2 PCI

Magistrala PCI (Peripheral Component Interconnect) wyparła poprzednio używane magistrale ISA (Industry Standard Architecture), EISA (Extended ISA), VLB (VESA Local Bus – Video Electronics Standards Association Local Bus) i MCA (Micro Channel Architecture).

Specyfikacja PCI określa wymiary złącza, charakterystyki elektryczne i protokół. Magistrala PCI zawiera m.in. następujące linie sygnałowe:

- systemowe
  - CLK – sygnał zegara
  - RST – reset
- adresów i danych
  - AD – multipleksowane linie adresów i danych
  - C/BE – multipleksowane linie rozkazów magistralowych i zezwolenia bajtów
- arbitrażowe – każde urządzenie ma własną parę tych linii
  - REQ – żądanie dostępu do magistrali
  - GNT – przyznanie dostępu do magistrali
- sterowania interfejsem
  - FRAME – linia aktywowana przez urządzenie, które uzyskało dostęp do magistrali i stało się urządzeniem nadrzędnym, wskazuje początek i koniec transakcji

  - IRDY – linia gotowości urządzenia nadrzędnego, podczas odczytu wskazuje, że urządzenie jest gotowe przyjąć dane, a podczas zapisu, że dane na liniach AD są ważne
  - TRDY – linia gotowości urządzenia podrzędnego, podczas odczytu wskazuje, że dane na liniach AD są ważne, a podczas zapisu, że urządzenie jest gotowe przyjąć dane
- zgłaszania przerwań – INTA, INTB, INTC, INTD

Pierwotna specyfikacja PCI określała:

- częstotliwość taktowania na 33 MHz,
- szerokość szyny danych na 32 lub 64 bity,
- 32-bitową przestrzeń adresową,
- 256 bajtową przestrzeń konfiguracyjną,
- zasilanie napięciem 3,3 V lub 5 V.

Integralną częścią protokołu PCI jest funkcjonalność *plug and play*. Oprogramowanie rozpoznaje urządzenia podłączone do magistrali PCI i przydziela im obszar(y) adresów pamięci, obszar(y) adresów wejścia-wyjścia i numer przerwania. Urządzenie PCI może też zawierać opcjonalnie ROM zawierający kod wykonywalny (ang. firmware).

Rozwój standardu PCI:

- PCI 2.2 zwiększa częstotliwość taktowania do 66 MHz;
- PCI-X specyfikuje częstotliwości taktowania 133 MHz;
- PCI-X 2.0 specyfikuje częstotliwości taktowania 266 MHz i 533 MHz, rozszerza przestrzeń konfiguracyjną do 4096 bajtów, dodaje 16-bitową wersję interfejsu i zasilanie 1,5V;
- Mini PCI to wersja PCI 2.2 do użytku w laptopach;
- Cardbus to wersja PCMCIA specyfikacji PCI (33 MHz i 32-bity);
- AGP (Accelerated Graphics Port, Advanced Graphics Port) – interfejs dwupunktowy dedykowany do kart graficznych;
- PCI Express.

6.3 SCSI

SCSI (Small Computer System Interface) jest intefejsem służącym do podłączania: dysków twardych, napędów taśmowych, skanerów, drukarek, napędów CD/DVD itp.

Fazy pracy magistrali SCSI:

- wolna – żadne urządzenie nie korzysta z magistrali,
- arbitraż – przejęcie sterowania magistralą przez jedno urządzenie,
- wybór – inicjator wybiera adresata,
- powtórny wybór – ponowne połączenie się przez adresata z inicjatorem w celu zakończenia uprzodnio rozpoczętej przez inicjatora lecz zawieszonej przez adresata operacji,
- fazy transferu informacji:
  - rozkaz,
  - dane,
  - stan,
  - komunikat.

SCSI umożliwia transfer asynchroniczny: wystawienie danych sygnalizowane jest sygnałem REQ, a odczytanie potwierdzane sygnałem ACK. W fazie transferu danych możliwe jest wynegecjowanie trybu synchronicznego: wystawienie kolejnych słów sygnalizowane jest sygnałem REQ, który pełni funkcję sygnału zegarowego, a potwierdzenie odczytania musi pojawić się w określonym czasie, ale nie musi następować po każdym słowie.

Magistrala (szyna) SCSI musi być zakończona z obu stron terminatorami. Terminatory zapobiegają powstawaniu odbić sygnału na końcu linii magistralowej, co mogłoby prowadzić do przekłamań transmisji. Terminatory mogą być pasywne (rezystory) lub aktywne (stabilizator napięcia) i mogą być zewnętrzne lub wbudowane w urządzenie.

Rozwój standardu SCSI:

- SCSI (SCSI-1) – 8 bitów, 5 MHz, 5 MB/s;
- Fast SCSI (SCSI-2) – 8 bitów, 10 MHz, 10 MB/s;
- Wide SCSI (SCSI-2) – 16 bitów, 10 MHz, 20 MB/s;
- Ultra SCSI (SCSI-3) – 8 bitów, 20 MHz, 20 MB/s;
- Ultra Wide SCSI (SCSI-3) – 16 bitów, 20 MHz, 40 MB/s;
- Ultra-2 SCSI – 8 bitów, 40 MHz, 40 MB/s;
- Ultra-2 Wide SCSI – 16 bitów, 40 MHz, 80 MB/s;
- Ultra-160 SCSI – 16 bitów, 40 MHz DDR, 160 MB/s;
- Ultra-320 SCSI – 16 bitów, 80 MHz DDR, 320 MB/s;
- Ultra-640 SCSI – 16 bitów, 160 MHz DDR, 640 MB/s.

Przyszłość SCSI:

- iSCSI – umożliwia używanie rozkazów SCSI przez sieć TCP/IP,
- Serial SCSI – szeregowa wersja interfejsu.

6.4 ATA

ATA (Advanced Technology Attachment) jest interfejsem służącym do podłączania pamięci masowych. Inną spotykaną nazwą tego standardu jest IDE (Integrated Drive Electronics). Pierwotnie interfejs był przeznaczony tylko do współpracy z dyskami twardymi. Rozszerzenie standardu nazywane ATAPI (ATA Packet Interface) obejmuje obsługę napędów dysków optycznych, napędów taśmowych i różnego rodzaju napędów dyskietek o dużej pojemności.

Przyszłość ATA:

- SATA (Serial ATA) – całkowicie nowy, szeregowy interfejs do urządzeń dyskowych.

6.5 RS-232

RS-232 jest jednym z najstarszych i nadal jeszcze używanych interfejsów szeregowych. Pełna asynchroniczna wersja tego interfejsu specyfikuje 9 drutów (8 sygnałów i masa). Jest też wersja specyfikująca 24 druty, mogąca pracować również w trybie synchronicznym i nazywana V.24. W najprostszym przypadku wystaczy użycie tylko 3 drutów i taka wersja zostanie omówiona. Te 3 druty to:

- TD (Transmitted Data) – dane nadawane,
- RD (Received Data) – dane odbierane,
- GND (Ground) – masa.

Występują dwa typy urządzeń:

- DTE (Data Terminal Equipment) – np. komputer, terminal,
- DCE (Data Communication Equipment, Data Circuit-terminating Equipment) – np. modem.

Oznaczenia sygnałów TD i RD odnoszą się do urządzenia typu DTE. W urządzeniu DCE linia TD służy do odbioru sygnałów, a linia RD do nadawania sygnałów. Występują więc dwa typy kabli:

- prosty – łączący urządzenie DTE z urządzeniem DCE, linia TD w DTE jest połączona z linią TD w DCE, a linia RD w DTE jest połączona z linią RD w DCE;
- skrzyżowany – łączący dwa urządzenia DTE, linia TD jednego urządzenia jest połączona z linią RD drugiego urządzenia.

Sygnalizacja odbywa się za pomocą napięcia o wartości bezwzględnej od 3 V do 25 V w stosunku do masy. Typowe napięcia sygnalizacyjne to: 5 V, 10 V, 12 V, 15 V. Używane są dwa poziomy napięcia:

- napięcie ujemne to sygnał mark, logiczna 1, stan off;
- napięcie dodatnie to sygnał space, logiczne 0, stan on.

Transmisja odbywa się w następujący sposób:

- 1 bit startowy, space, czyli logiczne 0;
- 5 lub 8 bitów danych, najpierw najmniej znaczący (LSB);
- opcjonalny bit parzystości, łączna liczba jedynek może być parzysta (bit even) lub nieparzysta (bit odd);
- 1 bit lub 1,5 bita lub 2 bity stopu, mark, czyli logiczna jedynka.

Pomiędzy kolejnymi transmisjami mogą występować dowolnie długie przerwy. Stosowane są prędkości transmisji od 50 bitów/s do 76800 bitów/s, a czasem więcej.

6.6 USB

USB (Universal Serial Bus) jest uniwersalnym interfejsem szeregowym. Aktualna wersja specyfikacji to 2.0.

Interfejs elektryczny składa się z 4 przewodów:

- VBUS – napięcie zasilające (nominalnie +5 V),
- GND – masa,
- D+, D− – para przewodów sygnalizacyjnych (skrętka), sygnalizacja różnicowa, naprzemienna (half duplex).

Zdefiniowane są trzy prędkości transmisji:

- high-speed – 480 Mb/s,
- full-speed – 12 Mb/s,
- low-speed – 1,5 Mb/s.

Bity kodowane są za pomocą kodu NRZI (ang. non return to zero inverted). Jedynka reprezentowana jest przez brak zmiany poziomu sygnału, zero reprezentowane jest przez zmianę (inwersję) poziomu sygnału. W celu zapewnienia synchronizacji stosowane jest *nadziewanie bitami*. Po każdych kolejnych sześciu jedynkach wstawiane jest dodatkowe zero. Odebranie siedmiu kolejnych jedynek traktowane jest jako błąd transmisji. Dane transmitowane są w pakietach. Początek pakietu wyznacza sekwencja synchronizacyjna (ang. sync pattern). Koniec pakietu wyznacza sekwencja EOP (ang. end of packet). Bajty transmitowane są w porządku little-endian, a bity w obrębie bajtu w porządku od najmłodszego (LSB) do najstarszego (MSB).

Standard USB definiuje dwa typy urządzeń:

- hub,
- function.

USB ma topologię drzewa. Taka konfiguracja wymuszana jest przez kształt złączy. Rozgałęzienia tworzą urządzenia typu hub. Liśćmi są urządzenia typu function. W korzeniu znajduje się *host controller* i *root hub*. Maksymalna wysokość drzewa wynosi 7, włączając w to korzeń i liście.

Urządzenia podłączane do USB mogą mieć własne autonomiczne zasilanie albo mogą być zasilane z interfejsu. Protokół USB przewiduje mechanizm zarządzania zasilaniem (ang. power management) obejmujący np. możliwość przejścia w stan uśpienia (ang. suspend) i powrotu do normalnej pracy (ang. resume).

Podłączanie i odłączanie urządzeń (TO DO).

USB umożliwia też połączenie ze sobą dwóch „równorzędnych" urządzeń, np. dwóch komputerów. Łączone urządzenia muszą obsługiwać protokół hosta i urządzenia typu function (ang. dual role devices). Do negocjacji, które z urządzeń pełni w danym momencie funkcję hosta, służy *host negotiation protocol*.

6.7 Inne interfejsy szeregowe

W przyrodzie występują setki różnego rodzaju interfejsów szeregowych. Omówmy dwa obecnie dość powszechne.

FireWire to nazwa handlowa opracowanego przez Apple interfejsu szeregowego. Standard został opisany w IEEE-1394 i IEEE-1394b. W sprzęcie audio występują jego modyfikacje oznaczane jako i.Link lub DV. Szybkość transmisji wynosi od 100 Mb/s do 400 Mb/s (3200 Mb/s w przypadku IEEE-1394b). Interfejs zawiera 6 przewodów: dwie skrętki do transmisji w dwu kierunkach oraz zasilanie i masę. Interfejs umożliwia zasilanie podłączonych do niego urządzeń.

DVI (Digital Visual Interface) służy do przesyłania danych z karty graficznej do monitora. Występują trzy rodzaje złącz:

- DVI-D – do przesyłania sygnałów cyfrowych,
- DVI-I – do przesyłania sygnałów cyfrowych i analogowych,
- DVI-A – do przesyłania tylko sygnałów analogowych (rzadko używane).

Złącze DVI-D zawiera 24 końcówki, trzy rzędy po 8 sygnałów. Interfejs zawiera 6 jednokierunkowych, szeregowych łączy cyfrowych. Standardowo wykorzystywane są 3 łącza, każde do przesyłania informacji o jednej z barw podstawowych. W wersji Dual Link do przesyłania każdego z kolorów używa się dwóch interfejsów, podwajając tym samym przepływność. Dodatkowo interfejs zawiera łącze umożliwiające przesyłanie informacji z monitora, np. w celu identyfikacji modelu i parametrów. Złącze DVI-I posiada dodatkowo końcówki do przesyłania sygnałów analogowych kompatybilnych z występującymi w używanym dawniej złączu D-Sub (służącym wtedy do podłączania monitorów CRT). Są to sygnały trzech kolorów (czerwony, zielony, niebieski) oraz sygnał synchronizacji poziomej (Hsync, Vsync przesyłany jest na końcówce „cyfrowej" nr 8).

7 Przykłady architektur

Na liście Top 500 [8] publikowane jest dwa razy do roku zestawienie największych komputerów na świecie. Omówimy te najpopularniejsze i te najbardziej egzotyczne architektury z tej listy. Oprócz superkomputerów, mikroprocesory występują obecnie w prawie każdym urządzeniu elektrycznym. Omówimy też i takie architektury.

7.1 IA-32, x86-64, EM64T

Wyczerpujące informacje można znaleźć na stronach internetowych [5] i [7] w następujących manualach:

- IA-32 Intel Architecture Software Developer's Manual
  - Volume 1: Basic Architecture
  - Volume 2A: Instruction Set Reference, A-M

  - Volume 2B: Instruction Set Reference, N-Z
  - Volume 3A: System Programming Guide, Part 1
  - Volume 3B: System Programming Guide, Part 2
- AMD64 Architecture Programmer's Manual
  - Volume 1, Application Programming
  - Volume 2, System Programming
  - Volume 3, General-Purpose and System Instructions
  - Volume 4, 128-Bit Media Instructions
  - Volume 5, 64-Bit Media and x87 Floating-Point Instructions

7.2 Power

TO DO

7.3 Cray X1E

Więcej informacji można znaleźć na stronie internetowej [6] w następujących pozycjach:

- Cray X1E Datasheet
- Cray X1 Series System Overview
- Optimizing Applications on Cray X1 Series Systems
- Cray Assembly Language (CAL) for Cray Systems Reference Manual

Firma Cray oferuje też computery o bardziej tradycyjnej architekturze. Dla porównania patrz:

- Cray XT3 Datasheet
- Cray XD1 Datasheet

7.4 Mikrokomputery jednoukładowe

TO DO

7.5 Mikroprocesory do zastosowań specjalnych

Procesory graficzne i sygnałowe – TO DO

Literatura

[1] W. Stallings, *Organizacja i architektura systemu komputerowego*, zawiera większość materiału omawianego na wykładzie.

[2] J. Baranowski, B. Kalinowski, Z. Nosal, *Układy elektroniczne, część III, układy i systemy cyfrowe*, pozycja nadprogramowa.

[3] W. Traczyk, *Układy cyfrowe, podstawy teoretyczne i metody syntezy*, pozycja nadprogramowa.

[4] R. Goczyński, M. Tuszyński, *Mikroprocesory 80286, 80386 i i486*, opisuje nieco archaiczne procesory, ale stanowi dobre wprowadzenie do poznania architektury IA-32.

[5] AMD, http://www.amd.com, zawiera wyczerpujące informacje na temat procesorów firmy AMD.

[6] Cray, http://www.cray.com, zawiera wyczerpujące informacje na temat komputerów firmy Cray.

[7] Intel, http://www.intel.com, zawiera wyczerpujące informacje na temat procesorów firmy Intel.

[8] Lista Top500, http://www.top500.org, publikuje dane o największych komputerach na świecie.

[9] Wikipedia, http://en.wikipedia.org, całkiem dużo informacji, czasem dość ogólnych, ale przystępnie podanych.