

Architektura hardwardzka:

- Rozdzielanie pamięci danych od pamięci programu (2 oddzielne szyny)
- Przechowywanie zmiennych w pamięci dynamicznej, a kodu programu w pamięci statycznej (bo 2 oddzielne szyny)
- Posiadanie oddzielnych magistral: pamięci danych i pamięci algorytmu
- Prostsza budowa (w stosunku do von Neumanna)
- Większa szybkość.
- Często wykorzystywana przy dostępie procesora do pamięci cache.

Architektura von Neumana:

- utrzymywanie w pamięci komputera zarówno programu, jak i danych, które są przechowywane
- w kodzie dwójkowym, a ich przetwarzanie odbywa się w arytmometrze.

Według tej koncepcji komputer składa się z 3 podstawowych części:

- procesor – z wydzieloną częścią sterującą oraz częścią arytmetyczno-logiczną (ALU)
- pamięć – dane i instrukcje są przechowywane we wspólnej pamięci w postaci binarnej
- urządzenia wejścia/wyjścia

ALU - jednostka arytmetyczno -logiczna układ cyfrowy będący jedną z głównych części procesora prowadząca proste operacje arytmetyczne (dodawanie, odejmowanie) oraz operacje logiczne (AND, OR, NOT, XOR). Zazwyczaj ALU ma dwa wejścia dla pary argumentów i jedno wyjście dla wyniku.

CPU - (Central Processing Unit) - urządzenie cyfrowe sekwencyjne potrafiące pobierać dane z pamięci, interpretować je i wykonywać jako rozkazy

- Bez analizy zawartości pamięci nie można jednoznacznie określić, czy dany blok pamięci zawiera kod programu czy nie (wynika to z budowy maszyny. W pamięci dane i instrukcje są wspólnie przechowywane)
- Zorganizowana jest pamięć operacyjna komputerów PC (praktycznie wszystkie komputery są budowane według pomysłu von Neumanna)

Zamiana BIN <-> HEX

BIN -> HEX

* dzielisz (cała liczba)

29	1
14	0
7	1
3	1
1	1
0	1

BIN -> DEC

* rozpisz
* dodaj

BIN -> HEX (Algo리즘)

* zapisz

0110	1110	1010
------	------	------

* podziel

* mierze dodaj zero

* wypisz **8421**

* zamień litery

np 10 = A
14 = E itp.
6 = 6

HEX -> BIN

* rozpisz

B4F

8421 8421 8421

* po dodaniu tych cyfry suma

1011 0100 1111

BIN -> OCT (8)

* rozpisz tylko

421

OCT -> BIN

247

417 417 417

010 100 111

Flagi:

- S (sign flag)** – flaga znaku, wynik pozytywny gdy ujemny. (dla wyniku -1 przyjmie wartość 1)
- C (carry flag)** – flaga przeniesienia, **wynik operacji zawiera się w większej niż dostępna liczbie bitów.** (np. dla 4bitowego wyrażenia, gdzie zakres wynosi 0..15 wyjdzie nam 16, to **C=1**) **NIE INTERPRETUJE ZNAKU, MA WYWALONE NA MINUSY!!**
- Z (zero flag)** – flaga zera, **pozytywne jeśli wynik = 0**
- O (overflow)** – flaga przepełnienia, podobna do flagi przeniesienia, jednak **odnosi się do operacji ze znakiem.** (zakres <-8,7> dla 4bitowego słowa)

Przesunięcie zawartości rejestru eax w lewo o 2 bity spowoduje:

mnożeniu wartości w rejestrze przez 4

Przesuwamy dwa w lewo, więc mamy
(0001 = 0100) - "I pyk z jedyinki zrobiła nam się czwóreczka" ~Maciej

Cechy pamięci SDRAM (rodzaj pamięci DRAM)

- Pracuje **synchronicznie** z magistralą systemową, w przeciwstwie do innych DRAM, które **asynchronicznie**
- Cykliczne odświeżanie zawartości
- Podział na banki
- Podział adresu na **adres rzędu** i **adres kolumny**
- Możliwość **łatwej alokacji pamięci**

Licznik rozkazów procesora (PC-program counter = IP-instruction pointer)

- rejestr, przechowuje informacje o tym, **w którym obecnie miejscu** sekwencji znajduje się procesor.
- **Przechowuje on adres kolejnej instrukcji** (w starszych modelach procesorów terażniejszej).
- Przez modyfikacje tego rejestru implementuje się skoki, skoki warunkowe, pętle i podprogramy.

Standard U2

W standardzie U2 **najstarszy bit** świadczy o znaku liczby. (1=liczba ujemna, 0=liczba dodatnia)
Jako, że na miejscu najstarszego bitu mamy znak, to **zakres naszej liczby to <-128,127>**.
1000 0000=-128 (najstarsza jedynka traktowana i jako znak, i jako liczba (-2^7). Nie jest to zero, ponieważ 0= 0000 0000, a w systemie tym występuje tylko jedno zero)
1000 0001(U2) = -127 (-128+1)
0111 1111(U2) =127 (1+2+4+8+16+32+64)

Zapis liczby zmiennoprzecinkowej, za pomocą ca łkowitych

R =S*M*P^W

S-znak M-znormalizowana mantysa, ułamek P-podstawa systemu liczbowego W-wykładnik

Technika przemianowania rejestrów (register renaming)

- używana w celu uniknięcia niepotrzebnego szeregowego wykonania instrukcji narzuconego przez wykorzystanie tych samych rejestrów procesora przez następujące po sobie instrukcje.
- Powodem powstawania hazardów jest wielokrotne używanie tych samych rejestrów do przechowywania różnych wartości

Dostępne hazardy:

- RAW** - read after write - NIE USUWA
- WAR** - write after read - OK
- WAW** - write after write - OK

Rozwiązuje to poniższe: (usuwa te hazardy)

- A)WaR** - nie musimy zapisywać nic w tym rejestrze po odczycie, bo specjalnie po to użyliśmy kolejnego rejestru
- c)WaW** - skoro dodaliśmy nowy rejestr na przechowywanie danych, to zapisany wcześniej rejestr nie jest nadpisany kolejnym
- d)War i WaW**

Pamięć cache: korzyści

najważniejsze:

- **zmniejszenie czasu dostępu**
- **wbudowany algorytm inteligentnego przechowywania danych = lepsze wykorzystanie**
- **lokalność odniesień w kodzie programu**

inne:

- Odpowiednna organizacja danych w pamięci RAM komputera
- Lokalność czasowa -częstotliwość odwołań do określonych danych i kodu programu
- Czas dostępu do głównej pamięci operacyjnej

Cykl rozkazowy procesora: sekwencja cykli maszynowych

1. Pobranie rozkazu

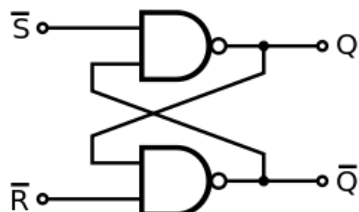
2. (Inkrementacja licznika rozkazów)
3. Dekodowanie rozkazu
4. Wykonanie rozkazu
5. Zapis wyniku i/lub ustawienie flag

Procesor jest układem sekwencyjnym

Jak najbardziej

Przerzutnik może być traktowany jako

- Podstawowy element funkcjonalny pamięci rejestru: może przechowywać **najmniejszą jednostkę informacji**
- **Element portu wejściowego / wyjściowego**
- małe ilości danych, do których musi być zapewniony ciągły dostęp.



Może to być element rejestru, element portu w/w, element pamięci statycznej RAM (**nie może być dynamicznej RAM**)

Wyjątek "Page Fault", czyli tzw. błąd stronicowania,

jest zgłaszany przez procesor wtedy, kiedy nastąpi nieprawidłowe odwołanie do jakiegoś liniowego adresu pamięci. Nieprawidłowe oznacza, że strona wyznaczana przez dany adres nie jest obecna.

Przerwanie lub żądanie przerwania to sygnał powodujący zmianę przepływu sterowania, niezależnie od aktualnie wykonywanego programu. Pojawienie się przerwania powoduje wstrzymanie aktualnie wykonywanego programu i wykonanie przez procesor kodu procedury obsługi przerwania.

-Tak więc ze względu na wystąpienie błędu polegającego na nieprawidłowym odwołaniu do adresu pamięci, do procesora wysyłany jest sygnał żądania przerwania. (ODP: zgłoszenie wewnętrznego przerwania CPU-wyjątku)

Najnowsze procki zgodne z x86 (i5, i7 itp.)

- Posiadają zintegrowany kontroler pamięci
- Posiadają wiele jednostek wykonawczych o zróżnicowanej funkcjonalności
- Posiadają zintegrowany mostek północny
- **NIE POSIADAJĄ** pamięci cache L3 oddzielnej dla każdego rdzenia
- **NIE MOGĄ** równolegle wykonywać kilkunastu instrukcji na jednym rdzeniu

Procesory CISC:

- Wiele różnych trybów adresowania
- Obecność instrukcji, które **UPRASZCZAJĄ** ręczne programowanie w assemblerku <3

System przerwania x86 obejmuje:

- Przerwanie **niemaskowalne**
- Przerwanie **sprzętowe**: zewnętrzne i wewnętrzne(wyjątki) oraz **programowe**
- **NIE OBEJMUJE** przerwania chronionych, wirtualnych, modulowanych fazą sygnału zegarowego

Przerwanie aby uruchomić system calls w Linuxie to przerwanie:

Programowe 80h

Procki ARM/Cores stosowane obecnie na szeroką skalę np. w prockach do urządzeń przenośnych mają architekturę typu RISC - najbardziej wydajne = najbardziej popularne

ALU - jednostka arytmetyczno -logiczna układ cyfrowy będący jedną z głównych części procesora

- proste operacje arytmetyczne (dodawanie, odejmowanie)
- operacje logiczne (AND, OR, NOT, XOR).
- sterowaniem skoków warunkowych.

Zazwyczaj ALU ma dwa wejścia dla pary argumentów i jedno wyjście dla wyniku.

DMA DIRECT MEMORY ACCESS

- ma za zadanie **odciążyć procesor główny od przesyłania danych** (np. z urządzenia wejściowego do pamięci). Procesor może w tym czasie zająć się innymi działaniami, wykonując kod programu pobrany uprzednio z pamięci RAM do pamięci podręcznej.
- Specjalizowane układy wspomagające DMA (np. te spotykane w PC), potrafią kopiować obszary pamięci dużo szybciej niż uczyniłby to programowo procesor główny.

ODP:wprowadzono to po to żbby

- a) **procesor nie musiał czynnie zajmować się transferem danych (I/O-pamięć)**
- B) **maksymalizować przepustowość urządzenia I/O-pamięć**

Przetwarzanie potokowe:

sposób przetwarzania rozkazów w procesorach, w którym do przetwarzania zastosowano potok podzielony na etapy. W przeciwieństwie do klasycznego (sekwencyjnego) sposobu przetwarzania rozkazów gdzie każdy jest pobierany, dekodowany, wykonywany oddzielnie, w przetwarzaniu potokowym w procesorze następuje **jednoczesne przetwarzanie kilku rozkazów jednocześnie**. Każdy z rozkazów znajduje się w danym cyklu maszynowym w innym etapie przetwarzania i dzięki temu wyniki operacji mogą być produkowane niemal w każdym cyklu.

Wiążą się z nim:

- **hazardy danych**
- **konflikty w dostępie do różnych jednostek wykonawczych procesora**
- **spekulacyjne wykonywanie instrukcji warunkowych/skoków**

Zapis binarny wykorzystujący bit znaku np float, znak-moduł - problemy:

- podwójne zero +0 -0
- NaN (not a number)
- symbol nieoznaczony
- **NIE MA PROBLEMU z wyjątkami od cpu**

```
-----
cmp eax,ebx
je wynik
```

je musi mieć flagę Z=1 (oraz S=0) bo różnica ta (wynik cmp) musi byc równa zero

Uszeregować wg największej przustowości:

1. PCI EXPRESS x16 3.0
2. Seria ata
3. USB 2.0
4. PS/2

W architekturze x86 adresy portów I/O

znajdują się w przestrzeni adresowej urządzeń I/O

FMA (alternatywna nazwa: FMAD)

to *zabezpieczona* (ang. *fused*) operacja typu **pomnóż-i-dodaj wykonywana w jednym kroku**, podczas którego wykonuje się tylko jedno zaokrąglenie. Oznacza to, że w instrukcji $a = b + c * d$ operacje mnożenia i dodawania wykonywane są dokładnie (bez żadnych zaokrągleń) i dopiero wynik dodawania zaokrąglany jest tak, by mógł być zapisany w a .

FMAD **przyspiesza i poprawia dokładność** wielu algorytmów numerycznych, w których wyznacza się sumy iloczynów:

Zwiększenie dokładności następuje poprzez **redukcję liczby zaokrągleń pośrednich wyników** w działaniu $a*b+c$ (**tylko raz zaokrągla**)!!!!

Komputer z prockiem wielordzyniowym: (np. i7)

- To system wielprocesorowy o **pamięci wspólnej (NIE rozproszonej!!)**
- Dla każdego rdzyna dostępna jest cała przestrzeń adresowa
- Komputer z wieloma procesorami uniwersalnymi, dostosowanymi do wykonywania różnych zadań
- czas dostępu do pamięci operacyjnej **NIE ZALEŻY** od adresu konkretnej komórki i numeru rdzyna CPU

Instrukcje push i pop

- związane są z obsługą stosu
- związane są z transferem danych do i z pamięci
- modyfikują wartość wskaźnika stosu
- **NIE MODYFIKUJĄ ramki stosu**
- **NIE MODYFIKUJĄ znaczników procesora**

Instrukcje Logical shift

Shift left i shift right

shlR, 1 (przesuwa wartosc rejestru R o 1 w lewo)

shrR, 1 (j/w ale w prawo)

Inne pytania:

Bramka realizująca funkcję eXOR może znaleźć

zastosowanie:

- a) Przy generowaniu/sprawdzaniu bitu parzystości
- b) Jako „sterowana” zewnętrznym sygnałem bramka negacji
- c) W komparatorze (dekoderze równości argumentów)
- d) W generowaniu flagi overflow
- e) W generowaniu flagi Carry(przeniesienie) **NIE**
- f) W sprawdzeniu poprawności działania arytmetycznych w kodzie U2 **NIE**

W procesorach Pentium4, Core2, i7 zastosowano

wielowątkowość typu:

- a) Fine-grained **NIE**
- b) Coarse-grained, **NIE**
- c) SMT(simultaneous MultiThreading)

Algorytmy zapisu:

Algorytm Write-Through: (zapis jednoczesny)

- Wszystkie operacje zapisu są rprowadzone jednocześnie do pamięci głównej i podręcznej
- Dzięki temu dane są zawsze aktualne w pamięci głównej
- Jakikolwiek inny moduł [procesor - pamięć podręczna] może monitorować przesyłanie do pamięci głównej w celu zchowania spójności pamięci podręcznej
- WADA: generuje znaczny przepływ danych do pamięci i może powodować wystąpienie wąskich gardeł

Algorytm Write-Back: (zapis opóźniony)

- minimalizuje ilość operacji zapisu do pamięci.
- aktualizuje się tylko pamięć podręczną.
- dane do pamięci głównej zapisuje się dopiero wtedy, gdy nie wystarcza już miejsca w pamięci podręcznej na ich dalsze magazynowanie.
- problem jest z tym że zawsze część pamięci głównej jest nieaktualna
- przez to dostęp przez I/O jest możliwy tylko za pomocą pamięci podręcznej

WAŻNE: Powoduje problemy ze spójnością danych w pamięci cache i RAM

Algorytmy zapisu http://kik.pcz.pl/soold/mainpage/subject22_2/chapt2.html

Big endian - od najbardziej znaczących bajtów, tak jest w starych prockach

Little endian -od najmniej znaczących, tak jest juz w nowszych intelach, amdkach, itp. Mimo ze kalkulator windowsowy pokaze wg big endian to procek juz woli sobie liczyc w little endian

O bajcie ~~ . Bajt jest najmniejszą jednostką obsługiwaną przez procesor. Coś jak z taksówką, ośmioosobowym mikrobusem. Nawet jeśli chcemy pojechać sami, to i tak siedem siedzeń pojedzie pustych. Spróbujmy poprosić taksówkarza żeby wymontował te zbędne i że zapłacimy tylko 1/8 ceny. Tak samo zareagował by procesor poproszony o przetworzenie jednego bitu.

<http://digitalforensics.pl/podstawy-analizy-danych/bajt/>