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.

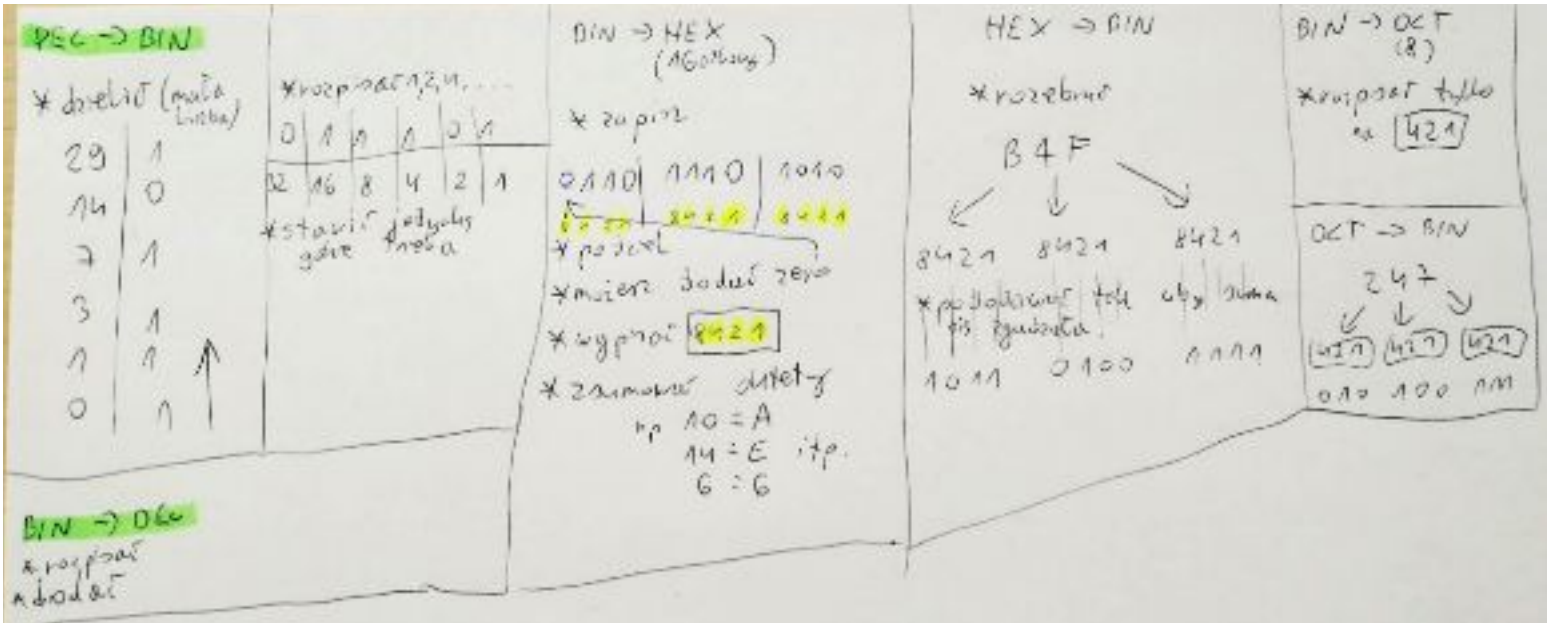
Według tej koncepcji komputer składa się z **3 podstawowych części:**
• procesor – z wydzieloną częścią sterującą oraz częścią arytmetyczno-logiczną (ALU)
• pamięć – dane i instrukcje są przechowywane we wspólnej pamięci w postaci binarnej
• urządzenia wejścia/wyjścia

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



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 dostepna liczbie bitow**. (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 jesli wynik = 0**
O (overflow) – flaga przepełnienia, podobna do flagi przeniesienia, jednak **odnosi się do operacji ze znakiem. (zakres <-8,7> dla 4bitowego słowa)**

Przesunięcie zawartoći 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 jedynki zrobiła nam się czwóreczka" ~Maciej*

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 teraźniejszej).
- Przez modyfikacje tego rejestru implementuje się skoki, skoki warunkowe, pętle i podprogramy.

Standard U2

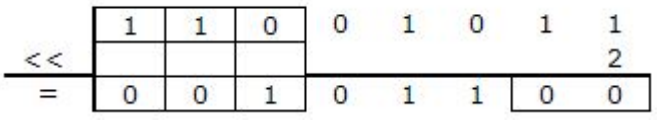
W standardzie U2 **najstarszy bit** świadczy o znaku liczby. (1=liczba ujemna, 0=liczba dodatnia)
Jako, że na miejscu najstarszego bitu mamy znak, to **zakres naszej liczby to <-128,127>.**

O co chodzi?
1101 to 1*(-8) + 1 * 4 + 1*2 +1*1
Chodzi o to ze normalnie rozpisujesz po kolei od prawej 1,2,4,8,16,32 okragle liczby, ale najstarszy bit jest na minusie :)

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) **tutaj jest jeden na początku więć jest to ujemna - - -**
0111 1111(U2) =127 (1+2+4+8+16+32+64) **tutaj jest zero na początku więc dodatnia +++**

Negacja (NOT) wszystkich bitów? Nic prostszego .
1. Zamieniasz wszystkie zera na jedynki
2. Dodajesz binarną jedynkę (...0001)

Przesuwanie bitów: tj.
2 w lewo: Fizycznie biore chwytam za ten ciąg i przesuwam go w lewo, pojawiaja mi sie 2 dziury po prawej i uzupelniam je zerami



Przesunięcie cykliczne (ROTACJA) ~ to samo tylko zamiast uzupelniac zerami uzupelniasz tym co wyleciało z drugiej storny

Zapis liczby zmiennoprzecinkowej, za pomocą całkowitych

R =S*M*P^W
S-znak M-znormalizowana mantysa, ułamek P-podstawa systemu liczbowego W-wykladnik

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 wartośći

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**

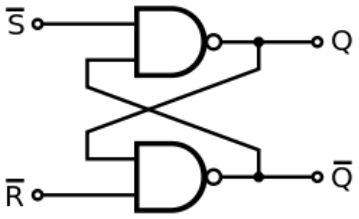
inne:
- Odpowiednnia organizacja danych w pamięci RAM komputera
- Lokalność czasowa -częstotliwość odwołań do określonych danych i kodu programu
- Czas dostępu do głównej pamięci operacyjnej

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 wejśćiowego / wyjściowego**
- małe ilości danych, do których musi być zapewniony ciągły dostęp.



Może to być element rejestru, element portu w/w, element pamieci 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 pamięći
- Posiadają wiele jednostek wykonawczych o zróżnicowanej funkcjonalnośći
- Posiadają zintegrowany mostek północny
- NIE POSIADAJĄ pamięci cache L3 odrebnej dla kazdego rdzynia
- NIE MOGĄ równolegle wykonywać kilkunastu instrukcji na jednym rdzyniu

Procesory CISC:
- Wiele różnych trybów adresowania
- Obecność instrukcji, które **UPRASZCZAJĄ** ręczne programowaniu w assemblerku **<3**

System przerwań x86 obejmuje:
- Przerwania **niemaskowalne**
- Przerwania **sprzętowe**: zewnętrzne i wewnętrzne(wyjątki) oraz **programowe**
- NIE OBEJMUJE przerwań 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 kliku 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 przustowośći:
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 dokładnośći 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 rdzynia dostępna jest cała przestrzeń adresowa
- Komputer z wieloma procesorami uniwersalnymi, dostosowanymi do wykonywania różnych zadań

- czas dostępu do pamięći operacyjnej **NIE ZALEŻY** od adresu konkretnej komórki i numeru rdzynia CPU

Instrukcje push i pop

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

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 rpwoadzone jednocześnie do pamięci głównej i podręcznej
- Dzięki temu dane są zawsze aktualne w pamięći głównej
- Jakikolwiek inny moduł [procesor - pamięć podręczna] może monitorować przesyłanie do pamięci głónej w celu zchowania spójnośći pamięći 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/

1) Info o L1,L2 itp z pamieci cache
2) O prockach cisc / risc
3) O pamieciach sdram itp
4) Cos tam o tych U2

DYNAMIC MULTIPLE ISSUE
- Dokonuje wiele insturkcji naraz
- Celem jest uniknięcie przestojów
- Wyniki są zapisywane do pamięći i rejestrów log
- Skomplikowany niestety układ, a co za tym idzie zajmuje sporo miejsca i zużywa w luj energii

Hierarchia pamięci: http://cygnus.tele.pw.edu.pl/olek/doc/syko/www/rozdzial4.html
- na wyższym poziomie mamy pamięći szybsze, ale droższe na bit
- na wyższym poziomie stosuje sie w mniejszych ilośćiach, tam gdzie częstość odwoływania przez sytsem jest większa
- te na niższym poziomie się uzywa czesciej, ale tam gdzie czestotliwosc odwoływania sie przez system jest mniejsza

Rejestry CPU - przechowuja argumenty operacji wykoanywanych przez procek, jest to pamiec statyczna, zaawartosc znika po wylaczeniu zasilania procka. Ma małą pojemność, obejmuje kilkanaście rejestrów 16,32,64 bitowych, ale jest SUPER SZYBKA!

Pamięć podręczna pierwszego poziomu L1 - ma pojemność kilkadziesiąt KB (np. Pentium III - 32KB, AMD Athlon - 128KB, Pentium 4 - 20KB) - przetrzymuje dane najczesciej uzywan eprzez procesor. Wspomaga komunikacje procka z pamiecia, zapewnia krotki czas dostepu(kilka nanosekund) do danych ktore w niej są, zapis nastpeuje wtedy z szybkoscia zegara procesora, nie ma stanów czekania. Pamięć ta jest asocjacyjną pamięćią SRAM zintegrowaną z prockiem

Pamięć podręczna drugiego poziomu - pamięć RAM o pojemnośći 256 KB lub 512 KB, teraz czasem jest zintegrowana z prockiem. Wspomaga prace pamieci podrecznej pierwszego poziomu. Odwoluje sie do niej procek jesli nie znajdzie info w pamieci pierwszego poziomu. (nastpeujwe to po magistrali lokalnej, w trybie seryjnym tzw. Burst mode. STANY OCZEKIWANIA wystepuja tylko i wylacznie jesli nie znajdzie tam sie czego sie chce i trzeba sie odwolac do pamieci głównej.

Pamięć ROM - read only memory - pamięć stała o swobodnym dostępie, przechowuje dane i kody mikroprogramów jendostki sterującej i programów systemowych. Obejmuje kilkaset KB. czas dostepu to kilkadziesią nanosekund.

Pamięć główna RAM - random access memory - pamiec operacyjna systemu. Przechowuje dani i kody programów. Ulotna pamięć dynamiczna o swobodnym dostępie DRAM. stanowi ogniwo między informacją zapisaną na dysku a prockiem i pamięciami

podręcznymi. Im więćej programów jest przechowywane w RAMIE a nie na dysku tym komp szybszy. Rezerwuje się w niej specjalny bufor zwany dyskową pamięcią podręczną służący do zbierania danych, które maja byz zapisane na dysku. Dzieki lokalnośći wiekszosci odniesien do dysku zmniejsza sie liczbe dostepow do dysku. Obecnie są 128MB do 1GB pamięci zintegrowanych w sepcjalnych bankach SIMM DIMM DDR i RIMM. Zapewiony jest niski czas dostepu przy duzych pojemnosciach jest problemem. Ponadto DRAM wymaga odświeżania. Czas potrzebny na odswiezenie charakteryzuje jej szybkość (ok 50-70 ns)

Pamięć dyskowa - pamiec nieulotna, charakteryzuje sie dostepem bezposrednim, bez koniecznosci przchodzenia przez wszytskie lokacje od początku. Czas dostępu do milisekundy.

Pamięć wymienna wymaga interwencji operatora - to dlatego jej czas dostepu jest dlugi.

Rodzaje pamięci:

SRAM - static RAM

- Elementarna komórka **złożona z przerzutnika** (od 4 do 6 tranzystorów)
- Wewnętrznie tablica komórek n rzędów na m kolumn
- Szybki dostęp

WAŻNE: Nie wymaga odświeżania zawartości (dopóki jest zasilana)

- Mały pobór prądu gdy niezmieniana

DRAM - dynamic RAM

- Elementarna komórka złożona z **tranzystora i kondensatora**
- Stan określony **naładowaniem** lub **rozładowaniem** kondensatora
- Wewnętrznie tablica komórek n rzędów na m kolumn
- Adres dzielony na adres rzędu i adres kolumny

SDRAM (rodzaj pamięci DRAM)

- Pracuje **synchronicznie** z magistralą systemową, w przeciwnstwie do innych DRAM, które asynchronicznie
- **Cykliczne odświeżanie zawartośći**
- Podział na banki
- Podział adresu na **adres rzędu** i **adres kolumny**
- Możliwość **łatwej alokacji pamięci**
- Magistrale: danych, adresowa, sygnały sterujące
- Możliwość transferu danych blokami

Cykle pracy:

- ACT - aktywacja/kopia danych z danego banku i rzędu
- RD/WR - odczyt/zapis, transfer danych seriami
- PRE - precharge, zamknięcie rzędu, zapis do komórek
- Refresh - odświeżanie zawartości

2 typy procesorów:

CISC - complex instruction set computer

- Duża liczba złożonych i skomplikowanych rozkazów
- **Mało rejestrów ogólnego przeznaczenia / roboczych**
- Duża liczba trybów adresowania dostępna w rozkazach wewnętrznych (5-20)
- mało rejestrów roboczych
- zróżnicowane czasy wykonania rozkazów

RISC - reduced instruction set computer

- Tylko rozkazy szybkie, proste i najczęściej stosowane (do 128 rozkazów)
- **Proste tryby adresowania**, mała ich liczba, do 4
- **Dużo rejestrów ogólnego przeznaczenia / roboczych**
- zróżnicowane czasy wykonania rozkazów
- łatwiejsza optymalizacja
- Lepsze do przetwarzania potokowego i do programowania równoległego
- Pisanie programu wymaga więcej pracy (złożone rozkazy CISC muszą być zastąpione prostymi RISC :( )

Scalar processors represent a class of computer processors. A scalar processor processes only one data item at a time, with typical data items being integers or floating point numbers.[1] A scalar processor is classified as a SISD processor (Single Instructions, Single Data)

Procesory skalarne

- Procesują tylko jeden proces w jednym momencie, zwykle inty i floaty.
- Jest to procek SISD (single instructions single data)

Procesory SUPERskalarne

- Mogą wykonywać dwie lub wincyj instrukcji skalarnych jednocześnie

Maja jednak takie ograniczenia:

- Zależność danych(drugi rozkaz moze ptrzebowac dane od pierwszego i nie moze sie wykonac :( )
- Rozgałęzienia w sekwencji rozkazów
- Konfikt zasobów - rywalizowanie instrukcji o te same zasoby np. Pamięć

Przeusnięcia i rotacje:

shr- jest dla unsigned, wypelenienie zerami z lewej
shl - jest dka unsigned - wypelnienie zerami z prawej
sar - jest dla signed - wypelnienie od lewej bitem znaku zeby zachowac znak
sal - jest dla signed, wypelenienie zerami z prawej
ror / rol - rotacja, czyli też wypelnienie tym co wyleciało