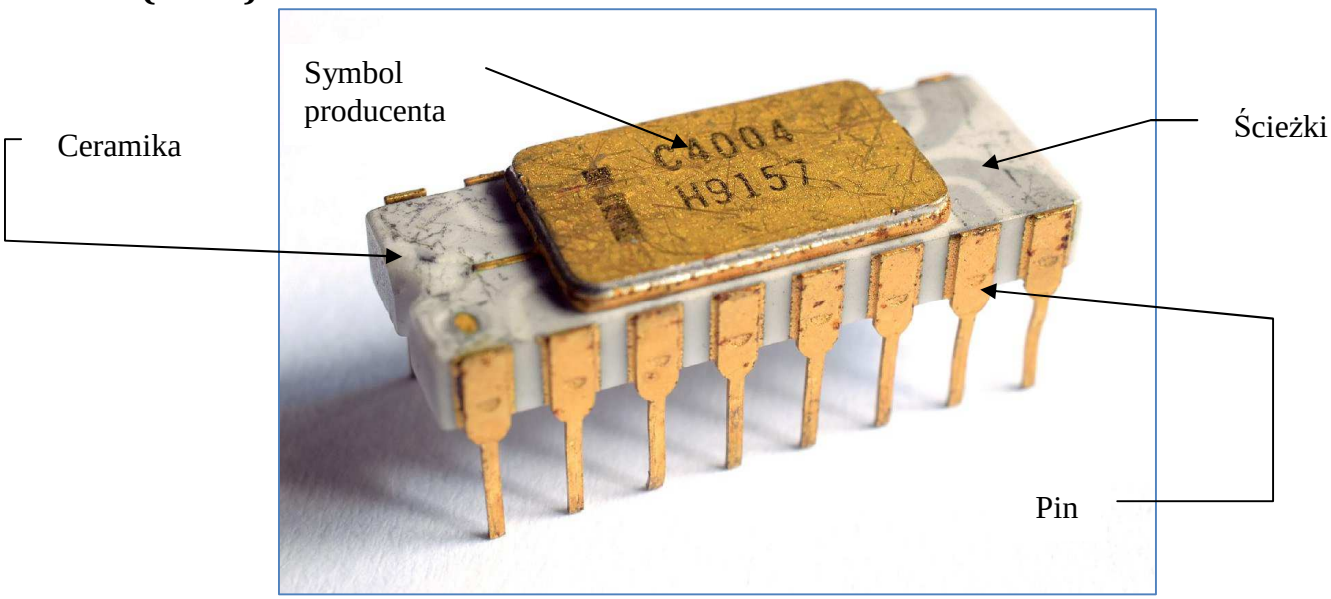
Wer. 2 (2020)



Rys. Przykładowy układ scalony mikroprocesora

Główne źródła

http://www.cpu-galerie.de/

Krzysztof Wojtuszkiewicz. Urządzenia Techniki Komputerowej. Cz. 1. Jak działa komputer ?[strony 158-159]

https://www.dell.com/support/article/pl-pl/sln32259/what-is-dynamic-execution-in-intel-pentium-ii-processors?lang=en

Tematyka wykładu

Wstęp – Wprowadzenie
Nowe podstawowe pojęcia związane z procesorami (Basic CPU parameters and technologies)
Lista rozkazów procesora (Instruction Set)
RISC
CISC
MMX
Rejestr procesora

Wstęp - Wprowadzenie

1). Przypomnienie
Masz system /model von Neumanna/ o następujących danych
Liczba rejestrów 4-bitowych wynosi 16
Szerokość magistrali danych procesora wynosi 4 bity
Szerokość magistrali adresowej procesora wynosi 4 bity
Oblicz jego przestrzeń adresową, wynik podaj jako liczbę komórek /4 bitowych/

wynik = 2^4 = 16

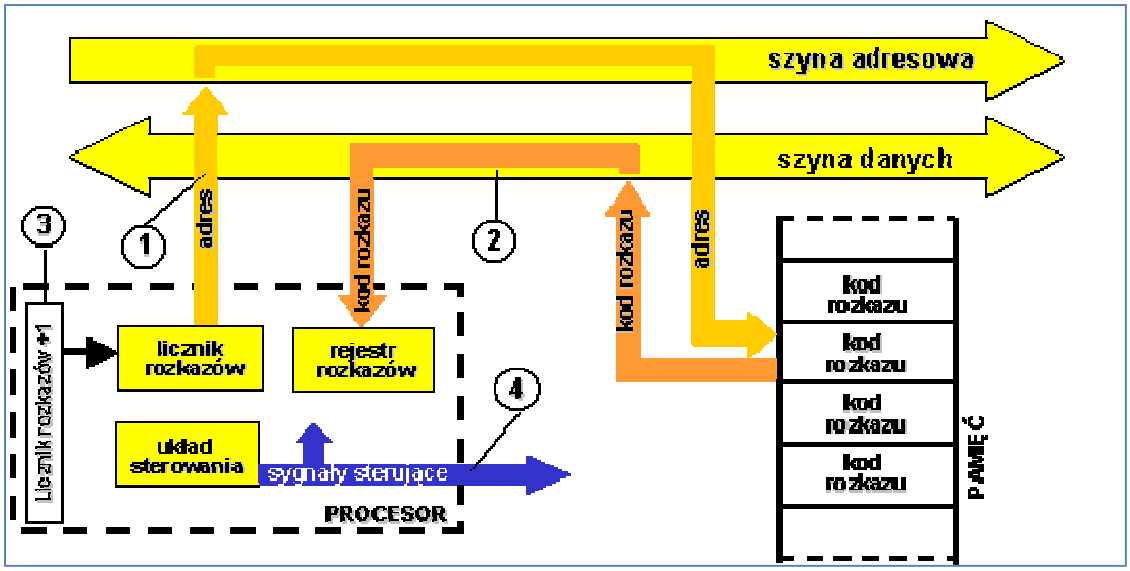
Masz system /model von Neumanna/ o następujących danych
Liczba rejestrów 8-bitowych wynosi 7
Szerokość magistrali danych procesora wynosi 8 bitów
Szerokość magistrali adresowej procesora wynosi 12 bitów
Oblicz jego przestrzeń adresową, wynik podaj jako liczbę komórek /8 bitowych/

wynik = 2^12 = 2^2 * 2^10 = 4K

Masz system /model von Neumanna/ o następujących danych
Liczba rejestrów 64-bitowych wynosi 32
Szerokość magistrali danych procesora wynosi 64 bitów
Szerokość magistrali adresowej procesora wynosi 32 bity
Oblicz jego przestrzeń adresową, wynik podaj jako liczbę bajtów /lub wielokrotności bajtów/

wynik = (2^32)*64/8 [B]= (2^32) * 8 [B]= (2^10 * 2^10 * 2^10 * 2^2) * 8 [B]= (1G * 4) * 8 [B] = 32 GB

CYKL ROZKAZOWY – PRZYPOMNIENIE

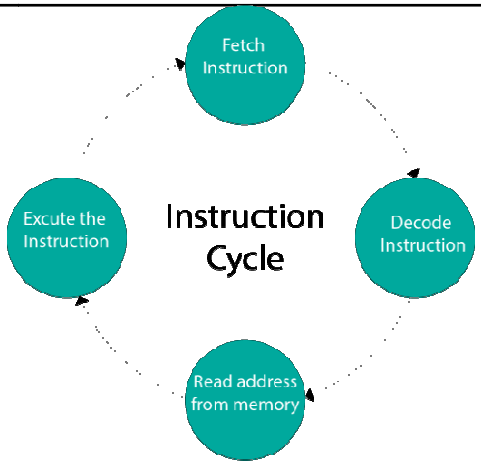


Rys. 1. Uproszczony cykl rozkazowy mikroprocesora

Cykl rozkazy procesora /prosty, bez pipeliningu, bez Multi-thredingu, bez Hyper-thredingu / składa się z faz:

- FETCH (POBRANIE ROZKAZU)
- EXECUTE (DEKODOWANIE I WYKOWANIE ROZKAZU)

Założenie: wykonywana jest instrukcja zapisu lub odczytu danej do/z pamięci



Rys. 2. Przykładowy uproszczony cykl rozkazowy

Rys. 3. Przykładowy uproszczony cykl rozkazowy

Powyższy cykl jest modelem uproszczonym, bowiem w rzeczywistości procesory używają zaawansowanych technik przetwarzania rozkazów, aby podwyższyć wydajność wykonywanych procesów.

2). Podstawowe nowe pojęcia

- Lista rozkazów
- RISC oraz CISC
- Proces i jego kontekst
- Multipleksowanie adresów w procesorze
- Prefetching
- Pipelining
- Dynamic Execution Microarchitecture
- Hyper-Threding Technology
- MMX

LISTA ROZKAZÓW – co to jest ?

Definicja

Lista rozkazów (Instruction Set) to po prostu zbiór rozkazów w języku asemblera, które potrafi wykonywać procesor !

https://en.wikipedia.org/wiki/X86_instruction_listings

Każdy procesor ma osobną listę rozkazów opisaną w jego dokumentacji.

Przykład 1 /fragment/:

INSTRUCTION	INSTRUCTION FORMAT				SEMANTICS
	15 – 12	11 - 8	7 - 4	3 - 0	
ADD Rt, Rs1,Rs2	0	R target	R source1	R source2	Rt ← Rs1 + Rs2 ; Inz ; Icv
SUB Rt, Rs1,Rs2	1	R target	R source1	R source2	Rt ← Rs1 - Rs2 ; Inz ; Icv
AND Rt, Rs1,Rs2	2	R target	R source1	R source2	Rt ← Rs1 *and* Rs2 ; Inz
OR Rt, Rs1,Rs2	3	R target	R source1	R source2	Rt ← Rs1 *or* Rs2 ; Inz
XOR Rt, Rs1,Rs2	4	R target	R source1	R source2	Rt ← Rs1 *xor* Rs2 ; Inz
ADDI Rt, *cte8*	5	R target	Constant		Rt ← Rt + ("00000000" & constant) ; Inz ; Icv
SUBI Rt, *cte8*	6	R target	Constant		Rt ← Rt - ("00000000" & constant) ; Inz ; Icv
LDL Rt, *cte8*	7	R target	Constant		Rt ← RtH & constant
LDH Rt, *cte8*	8	R target	Constant		Rt ← constant & RtL

Rys. 4. Przykład 1 /fragment/:

Mnemonic / Opcode / Operands / Description

INC	$0	<tar>[4R,12R,Mx] <tar-aop>	Increment contents of register or memory location[4,5]
DEC	$1	<tar>[4R,12R,Mx] <tar-aop>	Decrement contents of register or memory location[4,5]
ADDC	$2	<src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop>	Add contents of <src> and <tar> (with carry); store result in <tar>[1,2,3,4,5]
SUBB	$3	<src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop>	Subtract contents of <src> from <tar> (with borrow); store result in <tar>[1,2,3,4,5]
ROLC	$4	<tar>[4R,Mx] <tar-aop>	Rotate contents of <tar> left through the carry (C) status bit/flag[4,5]
RORC	$5	<tar>[4R,Mx] <tar-aop>	Rotate contents of <tar> right through the carry (C) status bit/flag[4,5]
AND	$6	<src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop>	AND contents of <src> and <tar>; store result in <tar>[1,2,3,4,5]
OR	$7	<src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop>	OR contents of <src> and <tar>; store result in <tar>[1,2,3,4,5]
XOR	$8	<src>[CV,4R,Mx] <tar>[4R,Mx] <src-aop> <tar-aop>	XOR contents of <src> and <tar>; store result in <tar>[1,2,3,4,5]
CMP	$9	<src1>[CV,4R,Mx] <src2>[4R,Mx] <src1-aop> <src2-aop>	Compare contents of <src> and <tar>; update C and Z status bits/flags
PUSH	$A	<src>[CV,4R,12R,Mx] <src-aop>	Push contents of <src> onto the stack[1,2,8,9]
POP	$B	<tar>[4R,12R,Mx] <tar-aop>	Pop value from stack into <tar>[4,5,10,11]
JMP	$C	<0/1 #sb> <tar-aop>	Jump to location[12]
JSR	$D	<0/1 #sb> <tar-aop>	Jump to subroutine[12]
NOP	$E	--	No operation (don't do anything furiously)
MOV	$F	<src>[CV,4R,12R,Mx] <tar>[4R,12R,Mx] <src-aop> <tar-aop>	Move (copy) the contents of <src> (the source) to <tar> (the target)[1,2,3,4,5,6,7]

<src> = Source
<tar> = Target (destination)
<src-aop> = Additional operand (none if source is a 4R/12R register)
<tar-aop> = Additional operand (none if target is a 4R/12R register)

0/1 = 1-bit logic 0 or 1 value
#sb = 3-bit value specifying status bit to test (0-7)

4R = One of the 4-bit registers (R0-R5, S0-S1)
12R = One of the 12-bit registers (PC, SP, IX, IV, TA)
CV = Constant value
Mx = Either the MD or MX virtual registers.
If MD, the address (operand) is used directly.
If MX, the address (operand) is first added to the contents of the index register (IX)

Rys. 5. Przykład 2 /fragment/:

Przykład możliwości procesorów:

procesor 4004 ma możliwość wykonania 46 różnych rozkazów.
procesor 8008 ma możliwość wykonania 48 różnych rozkazów.
procesor 8080 ma możliwość wykonania 72 różnych rozkazów.

RISC oraz CISC – co to jest ?

Wstęp

Według architektury CISC były tworzone pierwsze procesory x86, które wyposażano w pełny zestaw instrukcji mający im zapewnić wykonanie każdego polecenia użytkownika (a konkretnie programu). Z czasem okazało się jednak, że w 80% wypadków było wykorzystywanych tylko 20% dostępnych instrukcji, a pozostałe tylko sporadycznie. Zaowocowało to bardziej zaawansowaną architekturą o nazwie RISC. Współczesne procesory montowane w pecetach, np. Pentium czy Athlon, bazują na architekturze typu CISC, jednak przetwarzają rozkazy x86 na proste mikropolecenia, pracujące według idei RISC. Można więc powiedzieć, że obsługują architekturę mieszaną.

Architektura CISC

Cechy architektury **CISC** (Complex Instruction Set Computers) to

- Duża liczba rozkazów (100-200 instrukcji)
- Złożone i specjalistyczne rozkazy
- Mała optymalizacja – rozkazy potrzebują dużej liczby cykli procesora do wykonania
- Duża liczba trybów adresowania (od 5 do 20)
- Do pamięci może się odwoływać bezpośrednio duża liczba rozkazów
- Mniejsza częstotliwość taktowania procesora niż w architekturze RISC
- W jednej instrukcji może zostać wykonanych kilka operacji niskiego poziomu (np. pobranie z pamięci, operację arytmetyczną, zapisanie do pamięci)
- Mikroprogramowalna jednostka sterująca

Architektura RISC

RISC (Reduced Instruction Set Computers) to architektura procesorów z zredukowaną liczbą rozkazów (kilkadziesiąt instrukcji), wykonywanych w jednym cyklu zegara. Cechy architektury **RISC** to:

- Redukcja trybów adresowania
- Łatwe do zdekodowania, stałej długości (32 bity) formaty rozkazów
- Ograniczenie komunikacji między pamięcią a procesorem. Do przesyłania danych pomiędzy pamięcią a rejestrami służą instrukcje LOAD (załaduj z pamięci) oraz STORE (zapisz do pamięci). Pozostałe instrukcje operują tylko na rejestrach wewnętrznych procesora
- Zwiększenie liczby rejestrów co ma wpływ na zmniejszenie liczby odwołań do pamięci
- Jednostka sterująca realizowana układowo

CISC and RISC architecture Microcontrollers:

CISC Processors	***RISC Processors***
Complex Instruction Set Computer	Reduced Instruction Set Computer
When an MCU supports many addressing modes for arithmetic and logical instructions and for memory accesses and data transfer instructions, the MCU is said to of CISC architecture.	When an MCU has an instruction set that supports one or two addressing modes for arithmetic and logical instructions and few for memory accesses and data transfer instructions, the MCU is said to of RISC architecture
Large number of complex instructions	Small number of instructions
Instructions are of variable number of bytes	Instructions are of fixed number of bytes
Instructions take varying amounts of time for execution	Instructions take fixed amount of time for execution

Rys. 6. Porównanie CISC i RISC

RISC vs. CISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

Rys.7. Porównanie CISC i RISC

Proces i jego kontekst

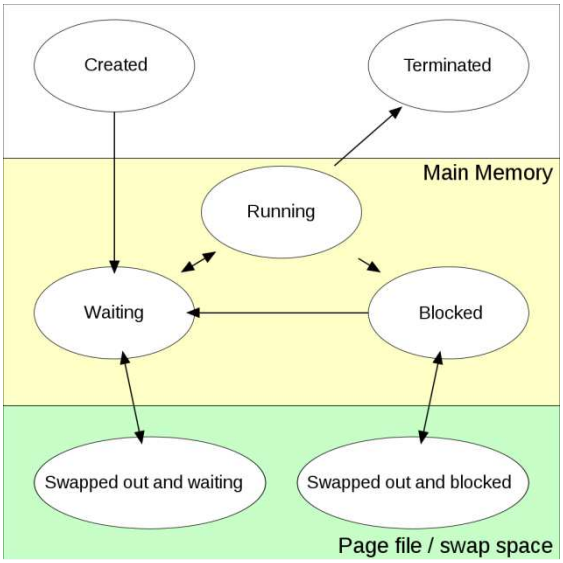
Proces ? Co to jest ?
Gdy uruchamiasz jakąś aplikację /program/ w systemie operacyjnym, to znaczy że ładujesz jej kod /rozkazy+dane/ z dysku do pamięci, w celu wykonania określonych czynności.

(w uproszczeniu) Proces to ciąg rozkazów załadowanych do pamięci RAM wykonywanych przez procesor.

W systemie operacyjnym /w pamięci operacyjnej/ pojawia się proces, który musi być wykonany przez procesor oraz powstaje tzw. kontekst procesu, czyli dodatkowe określające parametry tego procesu.

Przykładowe parametry procesu: identyfikator, stan procesu, czas startu, priorytet procesu, uprawnienia procesu, początkowy adres kodu, zakres adresów dla danych, dla stosu itd. itd.

Każdy administrator systemu operacyjnego może sprawdzać stan procesu, wpływać na niektóre jego parametry, np. może 'utworzyć nowy proces', 'zabić proces', 'wstrzymać proces', 'uruchomić ponownie proces', 'zamknąć proces'. /***polecam lekturę dot. systemu linux***/



Rys. 8. Przykładowy graf stanów procesu w systemie operacyjnym

Creating – tworzenie procesu
Terminating – zakończenie procesu
Running – proces jest wykonywany przez procesor
Waiting- proces oczekuje na obsługę przez procesor
Blocked – proces zablokowany
Swapped …. - /opis dla zaawansowanych, nie wymagany na tym poziomie edukacji/

W wielkim uproszczeniu każdy program uruchomiony w systemie operacyjnym

Multipleksowanie adresów w procesorze

Pierwotnie procesory używały jednego przewodu, (pinu, bitu) dla każdego bitu adresu.
Na przykład 4-bitowa magistrala adresowa miała 4 fizyczne przewody tworzące magistralę.
Ile mogła teoretycznie zaadresować komórek pamięci ? Oczywiście 2^4 = 16.

W celu zwiększenia przestrzeni adresowej systemu mikroprocesorowego, bez zwiększania szerokości magistrali adresowej /koszt implementacji ścieżek/, zastosowano metodę polegającą na wysyłaniu np. adresu 12-bitowego w 3 porcjach 4-bitowych (patrz procesor 4004). W ten sposób procesor wykorzystywał

szynę danych do adresowania. Oczywiście nie mógł jednocześnie przesyłać danej i adresu (wykorzystywał jedną wspólną magistralę dla adresów i danych), co spowalniało jego wydajność komunikacyjną.

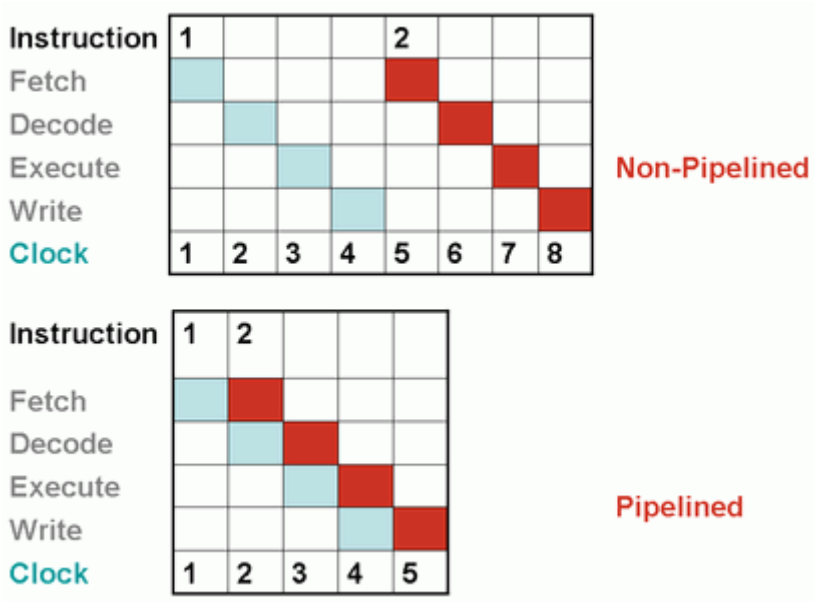
Pipelining

Pipelining to proces (technika) przetwarzania instrukcji przez procesor w tzw. potokach procesora.

Nie tłumaczymy tego słowa jako rura lub rurociąg! /czasem można znaleźć takie sformułowania w polskiej wersji Wikipedii/. Najlepiej jak będziemy używać oryginalnego słowa „**pipeline**” ewentualnie **potok**!

Technika lub proces składający się z wielu kolejnych, na przemiennie nakładających się wewnętrznych cykli rozkazowych.

Poniżej przedstawiono porównanie kolejności wykonywania faz : Fetch, Decode, Execute, Write w procesorze bez techniki pipelining oraz z bez techniką pipelining. Przykład zawiera instrukcje (rozkazy) nr 1 oraz nr 2.



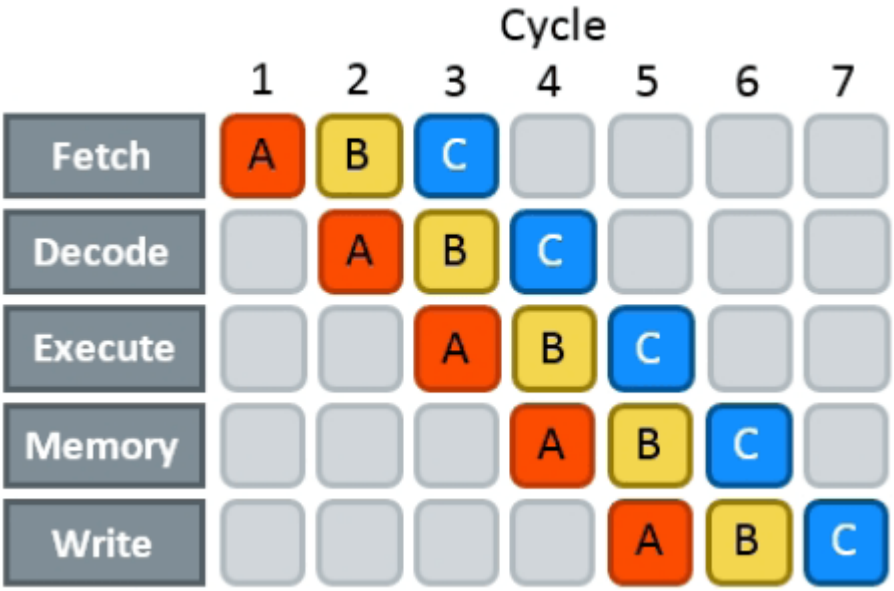
Rys. 9. Przykładowy cykl rozkazowy bez techniki pipelining oraz z techniką pipelining.

W pierwszym przypadku instrukcje (rozkazy) nr 1 oraz nr 2 wykowywane są w czasie 8 taktów zegarowych, a w przypadku drugim w 5 taktach zegarowych. Uzyskano skrócenie czasu wykonania obu instrukcji o 3 takty zegarowe.

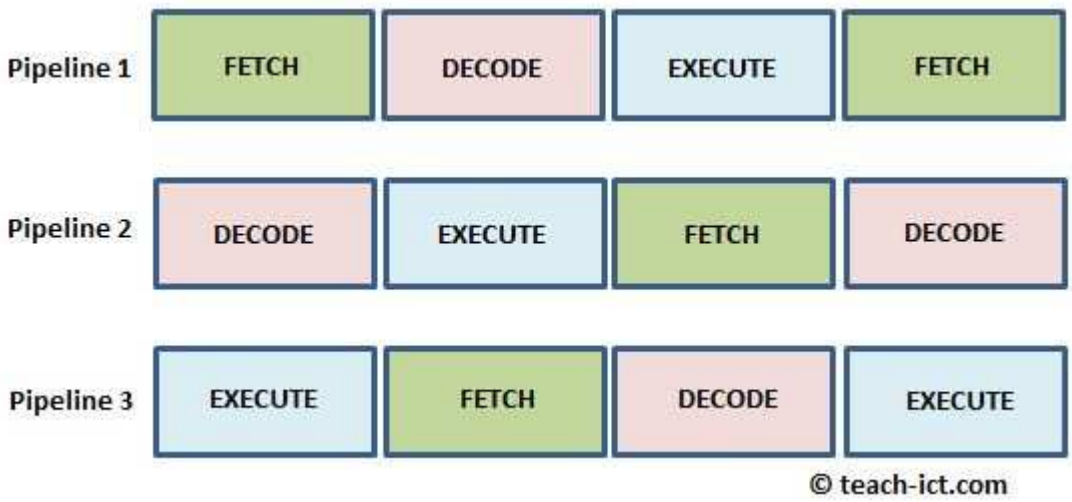
Założenie : do procesora zostały wysłane instrukcje A, B, C.
Kolejne cykle na rysunku to:

1. Najpierw jest pobrana instrukcja A.
2. Jednocześnie jest pobierana instrukcja B i dekodowana instrukcja A
3. Jednocześnie jest pobierana instrukcja C i dekodowana instrukcja B i wykonywana instrukcja A
4. Jednocześnie jest wykonywana operacja na pamięci (A) oraz dekodowana instrukcja C i wykonywana instrukcja B
5. Jednocześnie jest wykonywana operacja na pamięci (B i A) oraz wykonywana instrukcja C
6. Jednocześnie jest wykonywana operacja na pamięci (B i C)

7. Wykonywana operacja na pamięci (C)



Rys. 10. Demonstracja techniki pipelining dla trzech instrukcji A, B, C.



Rys. 11. Demonstracja techniki pipelining dla trzech potoków.
/źródło rysunku: http://teach-ict.com/

Krótki opis:
Założenie : do procesora zostały wysłane instrukcje A, B, C.
W procesorze jednocześnie w potoku 3 wykonuje się instrukcja (np. A) jakaś już zdekodowana instrukcja, a w potoku 2 dekoduje się inna instrukcja (np. B), a w potoku 1 następuje pobranie nowej instrukcji (np. C).

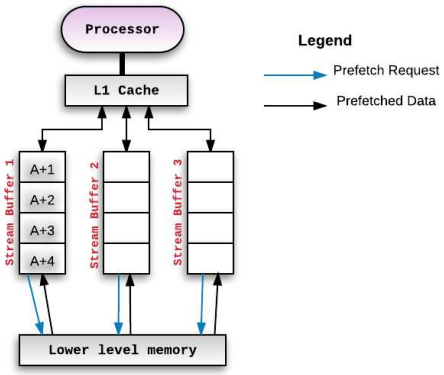
Spostrzeżenie : procesor wykonuje jednocześnie : pobranie instrukcji, dekodowanie instrukcji, wykonanie instrukcji /w uproszczeniu wyjaśniającym tą technikę/.

Prefetching

Prefetching to technika używana w procesorach, pozwala zwiększyć wydajność pobierania instrukcji (rozkazów).
Działanie jej polega na przenoszeniu rozkazów (instrukcji) z pamięci wolniejszej do pamięci szybszej.

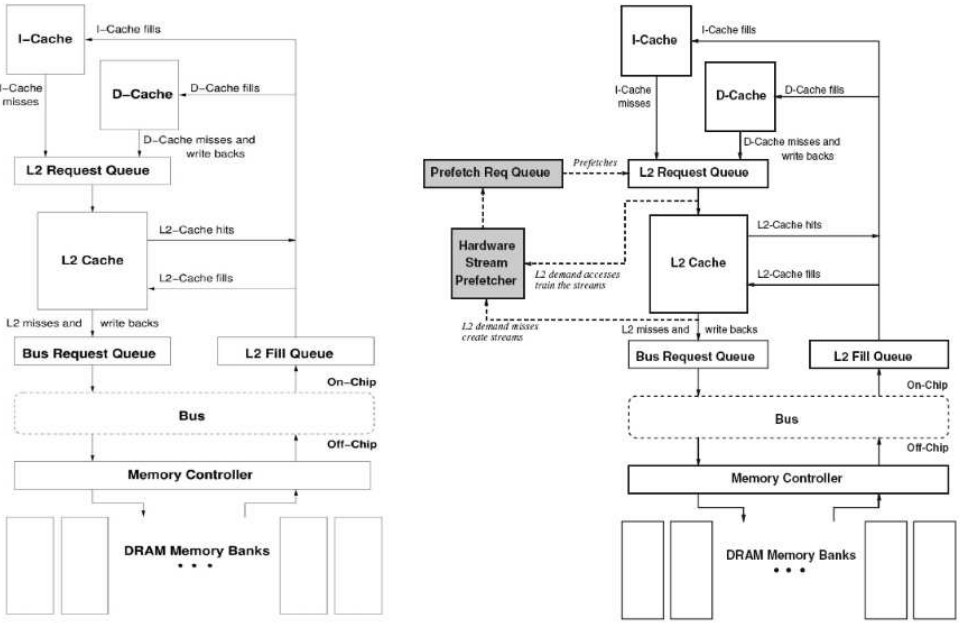
W ten sposób w czasie gdy procesor wykonuje jakiś rozkaz to jednocześnie są przygotowywane dla niego następne rozkazy w pamięci znajdującej się blisko niego, nazywanej jako pamięć podręczna /szybszej niż pamięć RAM/. Później zastosowano tą samą technikę także do pobierania danych.

Przykład



Rys. 12.

Lower level memory – pamięć wolniejsza (RAM), Prefetched Request - żądanie pobrania rozkazu, Prefetched Data – pobranie wstępne danych.



Rys. 13. System pamięci z zastosowaniem prefetch

Technologia Dynamic Execution w procesorze Intel

W procesorze Intel Pentium Pro zastosowano technologię o nazwie Dynamic Execution w celu polepszenia wydajności przetwarzania danych przez procesor.

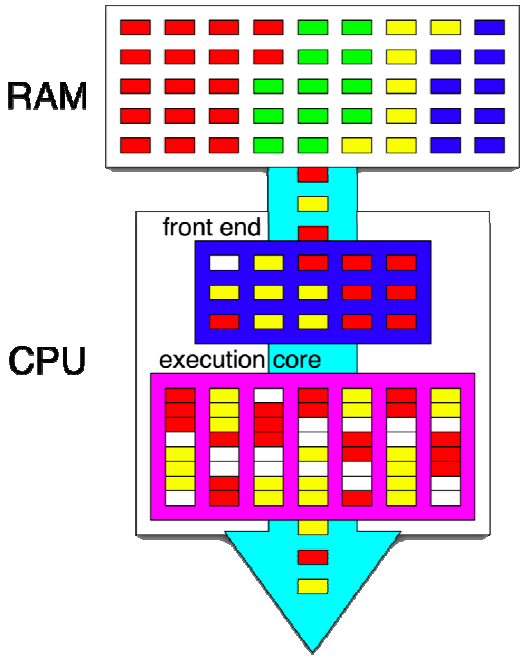
Technologia składa się z trzech technik przetwarzania, które zwiększają szybkość wykonywania instrukcji:

- **Multiple Branch Prediction**
- **Data Flow Analysis**
- **Speculative Execution**

/szczegółowe wyjaśnienie tych technik wykracza poza program nauczania przewidziany w tym dziale/

Hyper-Threding Technology /wielowątkowość procesów/

Technologia oficjalnie nazwana jako **Hyper-Threading Technology** lub **HT Technology** lub **HTT** lub **HT**. Wprowadzona do Intel Pentium 4.



Rys. 14. HT

Rys. 15. Pentium 4

Przykład dla dwóch rdzeni /wielowątkowość/
Dla każdego fizycznie obecnego rdzenia procesora system operacyjny adresuje dwa wirtualne (logiczne) rdzenie i dzieli obciążenie między nimi.

Główną funkcją wielowątkowości jest zwiększenie liczby niezależnych instrukcji w potoku, czyli wykonuje wiele instrukcji równolegle na oddzielnych rdzeniach.

W przypadku HTT jeden rdzeń fizyczny w systemie operacyjnym widziany jest jako dwa procesory, co umożliwia jednoczesne wykonywanie /programowanie/ dwóch procesów na rdzeń. Ponadto dwa lub więcej

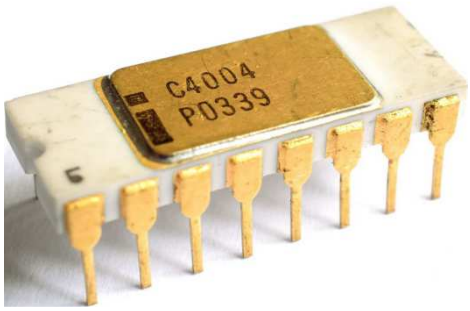
procesów może korzystać z tych samych zasobów, czyli jeśli zasoby dla jednego procesu nie są dostępne, inny proces może być kontynuowany, jeśli jego zasoby są dostępne.

Wielowątkowość a propos systemu operacyjnego (ang. *multithreading*) – cecha systemu operacyjnego, dzięki której w ramach jednego procesu może być wykonywanych kilka zadań nazywanych wątkami. Nowe zadania to kolejne ciągi instrukcji realizowane do pewnego stopnia niezależnie. Wszystkie wątki (zadania) w ramach tego samego procesu współdzielą tą samą wirtualną przestrzeń adresową zawierającą kod programu i jego dane.

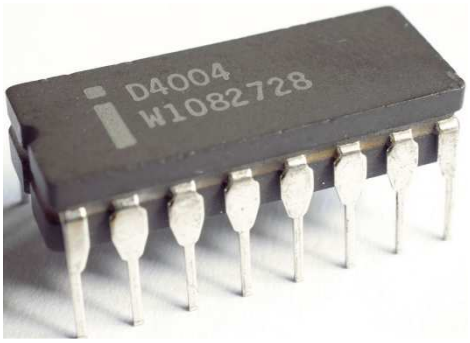
Główne cechy, podstawowe parametry i schematy blokowe procesorów

Procesor Intel 4004

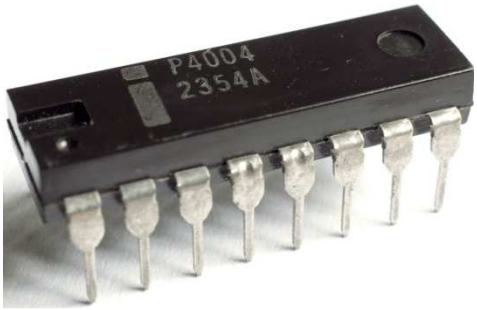
Pierwszy komercyjny mikroprocesor, wyprodukowany w roku 1971, przez firmę Intel, oznaczony jako **4004**.



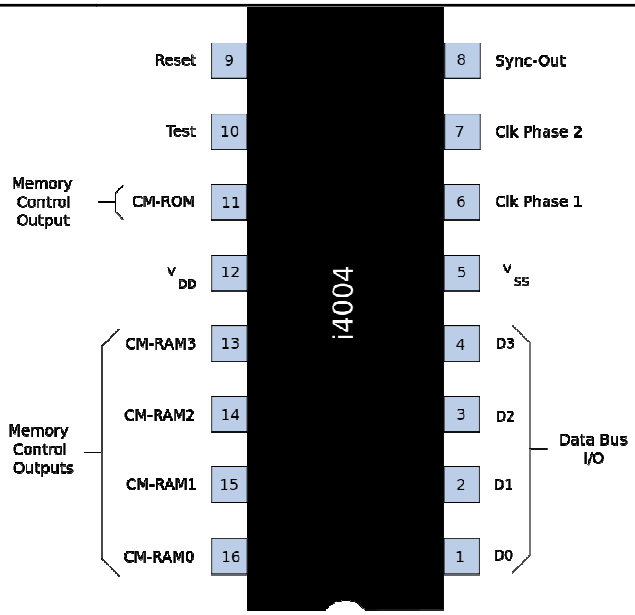
Rys. 16.Wariant podstawowy C4004 (ceramiczny)



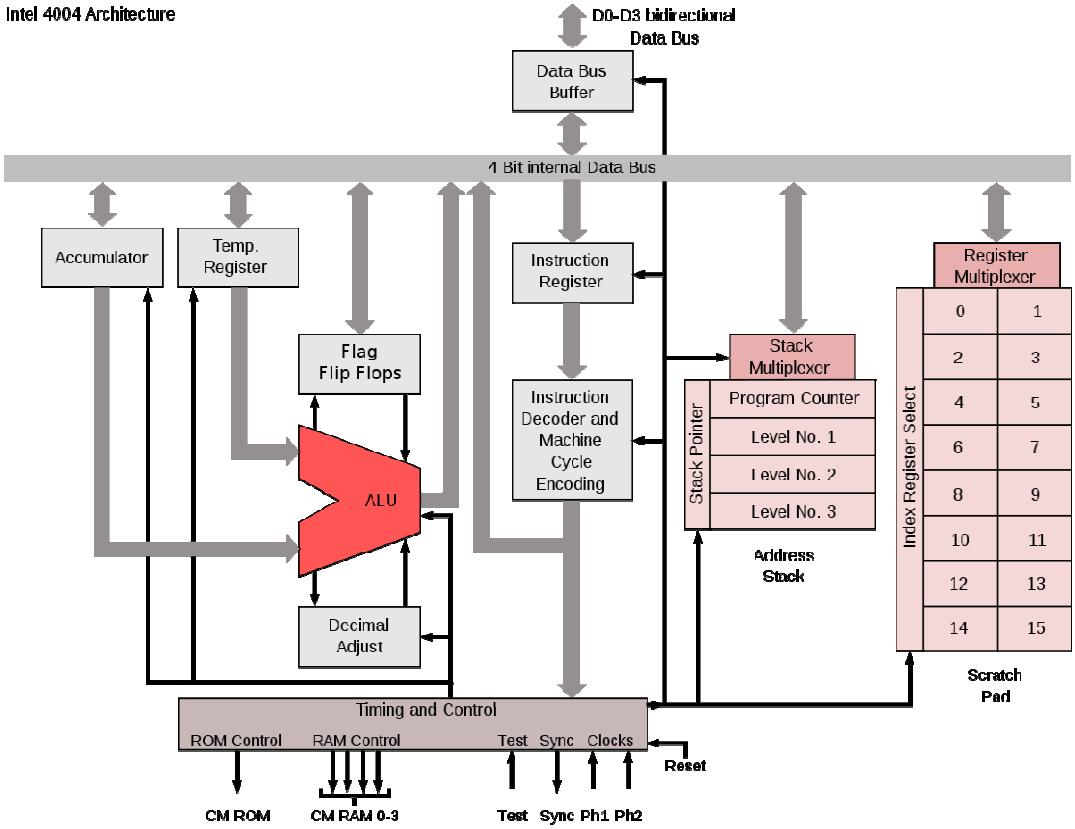
Rys. 17.Wariant D4004 (ceramiczny)



Rys. 18.Wariant P4004 (plastikowy)



Rys. 19. i4004 - rozkład pinów (styków)



Rys. 20. 1 i4004 – schemat blokowy

Główne parametry i cechy podstawowe i4004

Symbol procesora	4004
Liczba rozkazów	46
Liczba rejestrów 4-bitowych ogólnego przeznaczenia	16
Szerokość magistrali danych procesora	4
Przestrzeń adresowa	4 KB
Maksymalna częstotliwość taktowania	740 kHz
Główne napięcie zasilania	15V
Liczba tranzystorów	2300

Liczba rozkazów wykonywanych w 1 sekundzie	92000

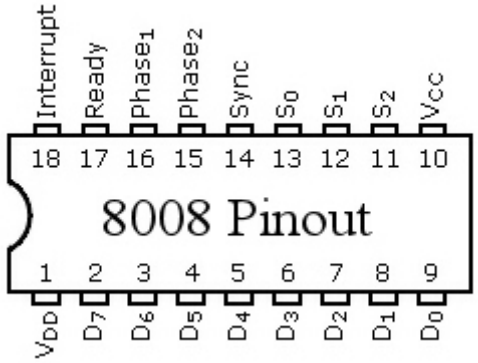
http://www.cpu-galerie.de/

Liczba rozkazów 46
Liczba rejestrów 4-bitowych 16
Szerokość magistrali danych procesora 4 bity
Przestrzeń adresowa 4 KB /12 bit /
Maksymalna częstotliwość taktowania 740 kHz
Główne napięcie zasilania: 15V
Liczba tranzystorów 2300
Liczba rozkazów wykonywanych w 1 sekundzie: 92000

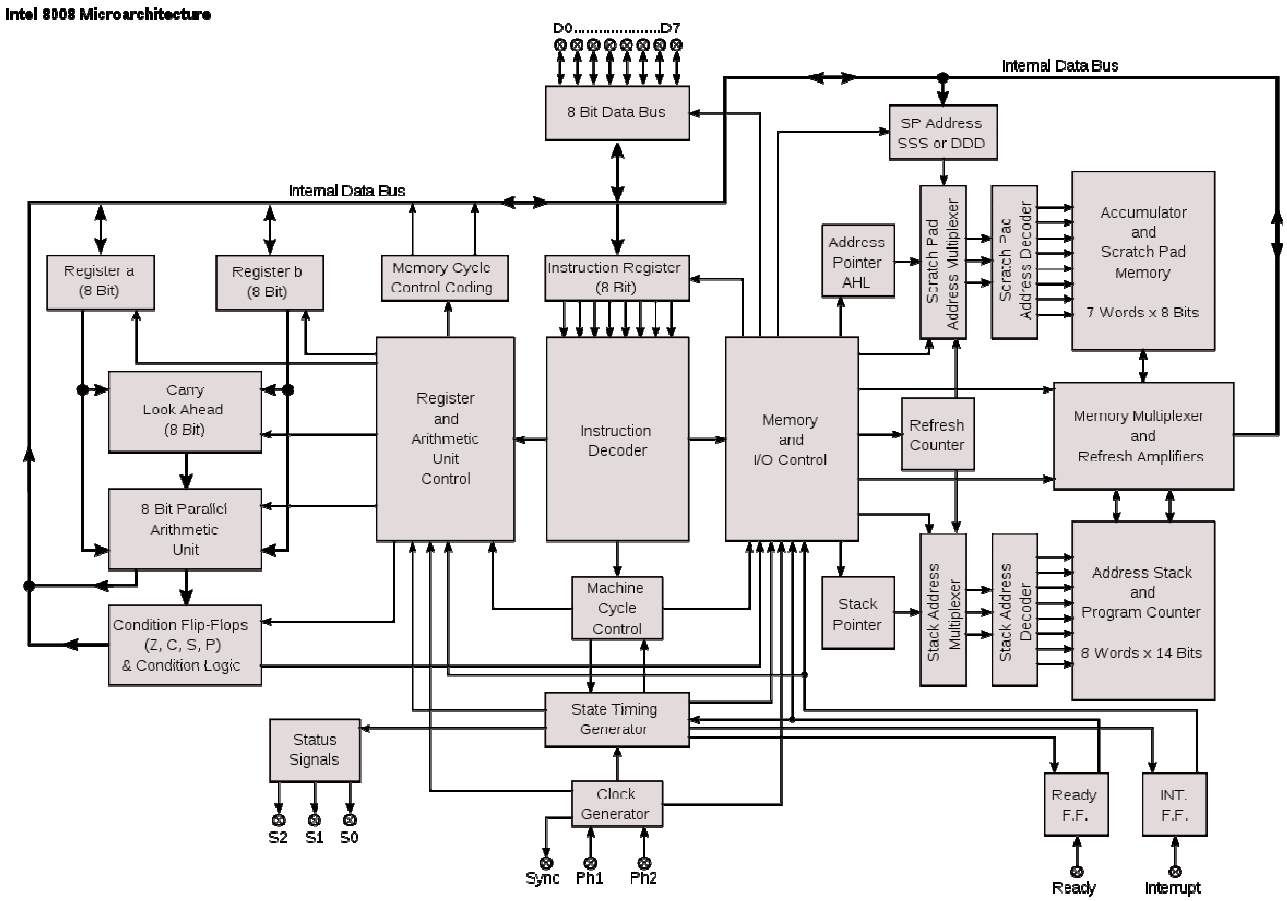
Podsumowanie cech architektury i4004

- 12-bitowe adresowanie
- 8-bitowe instrukcje
- 4-bitowe dane

Główne parametry i cechy podstawowe i8008



Rys. 21. i8008 - rozkład pinów (styków)



Rys. 22. i8008 – schemat blokowy

Symbol procesora	8008
Liczba rozkazów	48
Liczba rejestrów 8-bitowych ogólnego przeznaczenia	7
Szerokość magistrali danych procesora	8
Przestrzeń adresowa	64 KB
Maksymalna częstotliwość taktowania	800 kHz
Główne napięcie zasilania	5V
Liczba tranzystorów	3500
Liczba rozkazów wykonywanych w 1 sekundzie	160000

http://www.cpu-galerie.de/

Liczba rozkazów 48
Liczba rejestrów 7
Szerokość magistrali danych procesora 8
Przestrzeń adresowa 64 KB /16 bit/
Maksymalna częstotliwość taktowania 800 kHz
Napięcia zasilania: 5V
Liczba tranzystorów 3 500
Liczba rozkazów wykonywanych w 1 sekundzie: 160 000

Główne parametry i cechy podstawowe i8080

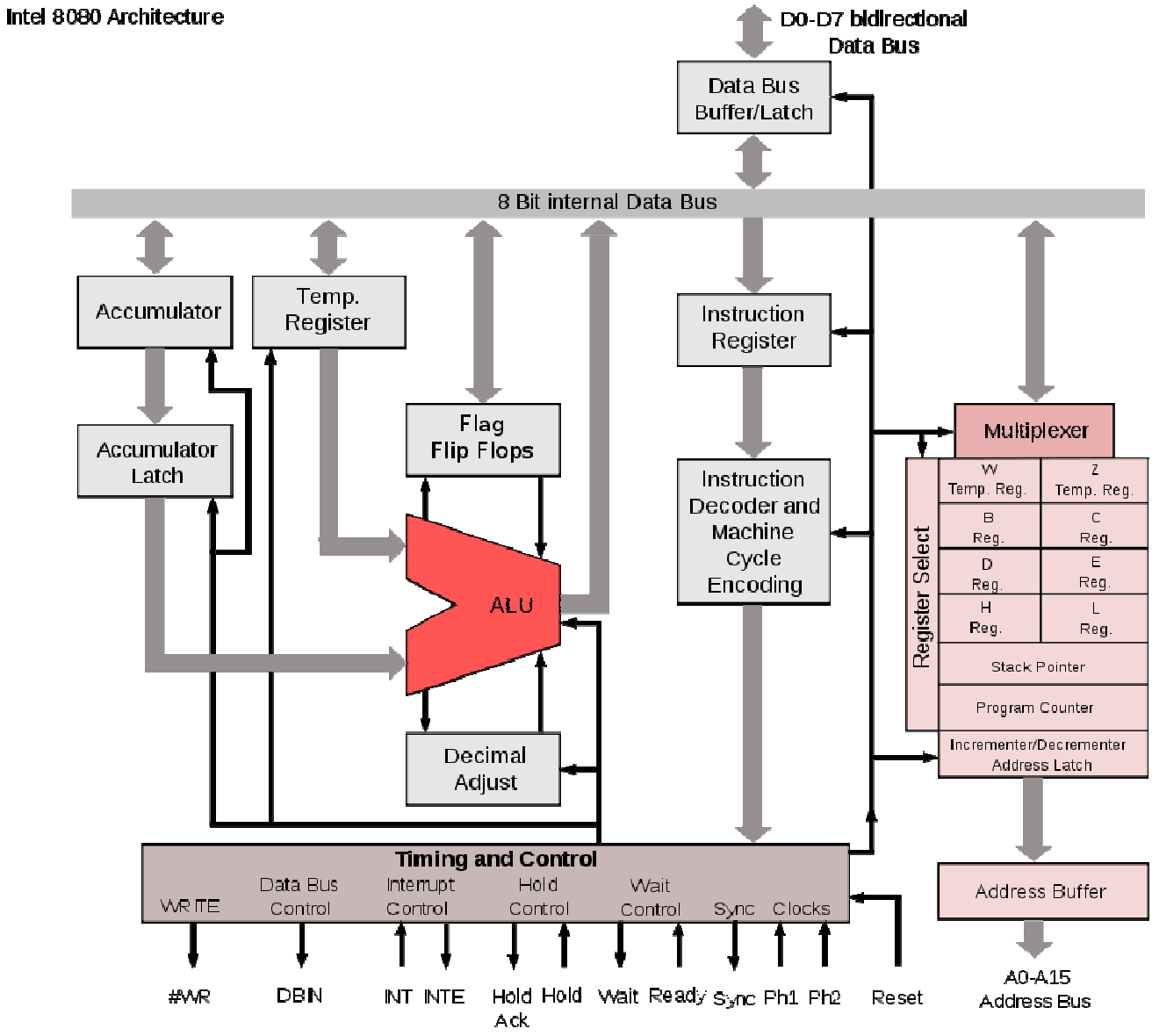
Rys. 23. Intel 8080A

Symbol procesora	8080
Liczba rozkazów	72
Liczba rejestrów 8-bitowych ogólnego przeznaczenia	8
Szerokość magistrali danych procesora	8
Przestrzeń adresowa	64KB
Maksymalna częstotliwość taktowania	2 MHz
Główne napięcie zasilania	5V
Liczba tranzystorów	6000

http://www.cpu-galerie.de/

Liczba rozkazów 72
Liczba rejestrów 8
Szerokość magistrali danych procesora 8
Przestrzeń adresowa 64 KB /16 bit/
Maksymalna częstotliwość taktowania 2 MHz
Napięcia zasilania: 5V
Liczba tranzystorów 6000

Rys. 24. Intel 8080A



Rys. 25. i8080 – schemat blokowy

MMX

MMX (ang. *Multimedia Extensions* lub *Matrix Math Extensions*) – zestaw 57 instrukcji dla procesorów Pentium i zgodnych z nim.

Instrukcje te wykonują między innymi operacje:

- wyświetlanie grafiki trójwymiarowej: przekształcenia geometryczne, cieniowanie, teksturowanie,
- dekodowanie obrazów JPEG i PNG,
- dekodowanie i kodowanie filmów MPEG,
- filtrowanie sygnałów: obrazów statycznych, filmów, dźwięku;
- wyświetlanie grafiki dwuwymiarowej (blue box, maskowanie, przezroczystość);

SSE

SSE (ang. *Streaming SIMD Extensions*) jest nazwą zestawu instrukcji wprowadzonego w 1999 roku po raz pierwszy w procesorach Pentium III firmy Intel. SSE daje przede wszystkim możliwość wykonywania działań zmiennoprzecinkowych na 4-elementowych wektorach liczb **pojedynczej precyzji** (48 rozkazów). Ponadto wprowadzono jedenaście nowych rozkazów stałoprzecinkowych w zestawie MMX, a także dano możliwość wskazywania, które dane powinny znaleźć się w pamięci podręcznej.

Rejestry procesora

Rejestr procesora to komórka pamięci o niewielkich rozmiarach (najczęściej 4/8/16/32/64/128/256 bitów)

Rejestry umieszczone są wewnątrz procesora i służące do przechowywania tymczasowych wyników obliczeń.

Rejestry procesora to najszybsze komórki pamięci.

Rejestry procesorów x86 ogólnego przeznaczenia

32-bitowe rejestry ogólnego przeznaczenia to:

- **EAX** – Accumulator (akumulator – jego pamięć wykorzystuje arytmometr; używa się go do przechowywania wyników wielu operacji)
- **EBX** – Base Register (rejestr bazowy – służy do adresowania)
- **ECX** – Counter Register (rejestr licznikowy – służy jako licznik w pętli)
- **EDX** – Data Register (rejestr danych – umożliwia przekaz/odbiór danych z portów wejścia/wyjścia)
- **ESP** – Stack Pointer (przechowuje wskaźnik wierzchołka stosu)
- **EBP** – Base Pointer (rejestr bazowy – służy do adresowania)
- **ESI** – Source Index (rejestr źródłowy – trzyma źródło łańcucha danych)
- **EDI** – Destination Index (rejestr przeznaczenia – przetrzymuje informacje o miejscu docelowym łańcucha danych)

Zestawienie wybranych procesorów rodziny x86

Procesor	Data powstania	Cz. zegara	L. tranzystorów	Rozmiar rejestrów	Przestrzeń adresowa	Pamięć cache	Architektura
8086	1978	8MHz	29 tys.	16GP	1MB	brak	prefetching
80286	1982	12,5MHz	134 tys.	16GP	16MB	Brak	pipelining
80386	1985	20MHz	275 tys.	32GP	4GB	zewnętrzna	pipelining
80486	1989	25MHz	1,2 mln	16GP 80FPU	4GB	L1 8KB	pipelining
Pentium	1993	60MHz	3,1 mln.	16GP 80FPU	4GB	L1 16KB	pipelining
Pentium Pro	1985	200MHz	5,5 mln.	16GP 80FPU	64GB	L1 16KB L2 256KB lub 512KB, 1MB	Dynamic Execution Microarchitecture
Pentium II	1997	266MHz	7 mln.	16GP 80FPU 64MMX	64GB	L1 16KB L2 512KB	Dynamic Execution Microarchitecture
Pentium III	1999	500MHz	8,2 mln.	16GP 80FPU 64MMX	64GB	L1 16KB L2 512KB	Dynamic Execution Microarchitecture
Pentium 4	2004	3,4GHz	125 mln.	16GP 80FPU 64MMX 128 XMM	64GB	L1 16KB L2 1MB	NetBurst Microarchitecture Hyper-Threding Technology
64-bit Xeon with 800 MHz System Bus	2004	6,6GHz	125 mln.	32, 64GP 80FPU 64MMX 128 XMM	64GB	L1 16KB L2 1MB L3 2MB	NetBurst Microarchitecture Hyper-Threding Technology Extended Memory 64 Technology
…	……	……	……	……	……	…….	……..

Źródło : Krzysztof Wojtuszkiewicz. Urządzenia Techniki Komputerowej. Cz. 1. Jak działa komputer ?[strony 158-159]

Podsumowanie - Przykładowe pytania sprawdzające do kartkówki /ale nie sugeruj się bo mogą być trudniejsze/

Podaj długość listy rozkazów dla procesora Y
Podaj liczbę rejestrów X-bitowych dla procesora Y
Podaj liczbę rejestrów ogólnego przeznaczenia (General Purpose Registers) dla procesora Y
Podaj szerokość magistrali danych procesora Y
Podaj maksymalną częstotliwość taktowania w procesorze Y
Podaj szerokość magistrali adresowej procesora Y
Podaj główne napięcie zasilania procesora Y
Podaj przestrzeń adresową dla procesora Y
Opisz cechy architektury CISC i RISC
Opisz technologię Pipelining
Opisz technologię HT

KARTKÓWKA!!!

WAŻNA UWAGA KOŃCOWA:
Dot. przedmiotów PODSSTAWY TECHNIKI KOMPUTEROWEJ, URZĄDZENIA TECHNIKI KOMPUTEROWEJ, PRACOWNIA TECHNIK KOMPUTEROWYCH, PRACOWNIA URZĄDZĘŃ TECHNIKI KOMPUTEROWEJ.

SYGNATURY /PTK,UTK,PRUTK,PRTK/