

Dokument ten jest dodatkiem do prezentacji zawierającej treść wykładu trzeciego. Zostały w nim dokładniej omówione najważniejsze zagadnienia przedstawione na prezentacji.

Zawartość

Budowa procesora	1
Jednostka arytmetyczno-logiczna (ALU)	2
Blok sterujący procesora	3
Rejestry procesora	4
Architektura CISC i RISC	4

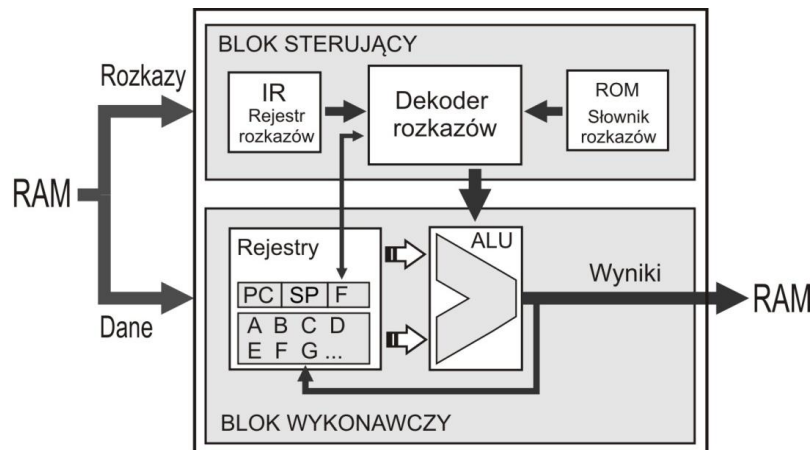
Budowa procesora

Budowę procesora opisywać można na wielu poziomach abstrakcji. Programiści korzystający z języków wysokiego poziomu traktują procesor, a często właściwie cały komputer, jako „czarną skrzynkę” realizującą zadane jej polecenia. Z tego punktu widzenia procesory różnią się właściwie tylko wydajnością.

Programiści korzystający z języków niskiego poziomu (Assemblerów) traktują procesor jako zestaw rejestrów, czyli komórek pamięci do których zapisywane są dane i w których pojawiają się wyniki. Sposób, w jaki procesor przetwarza dane i dochodzi do właściwych wyników również w tym przypadku nie jest istotny. Podejście takie sprawdza się w praktyce, aby jednak zrozumieć jak działają poszczególne elementy komputera PC, jak ze sobą współpracują i wymieniają dane należy nieco dokładniej przyjrzeć się budowie procesora i zasadzie jego działania.

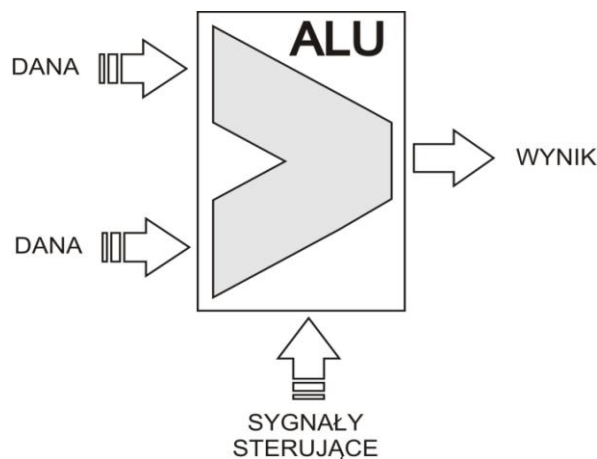
Wewnątrz każdego procesora wyróżnić można kilka podzespołów wykonujących ściśle określone zadania i współpracujących ze sobą w trakcie realizacji programu. Są nimi jednostka arytmetyczno logiczna (ALU), która wraz z zespołem rejestrów stanowi blok wykonawczy oraz dekodery rozkazów, rejestr rozkazów i słownik rozkazów składające się na blok sterujący procesora (rys...).

19 marca 2020



Jednostka arytmetyczno-logiczna (ALU)

Najważniejszym elementem procesora jest ALU – jednostka arytmetyczno-logiczna (ang. Arithmetic Logic Unit) nazywana też niekiedy arytmetrem. Jest to uniwersalny układ cyfrowy, w którym wykonywane są operacje arytmetyczne i logiczne na dostarczanych do niego liczbach. Dane pobierane są z pamięci operacyjnej lub rejestrów, a o tym, jaka operacja zostanie na nich wykonana decydują sygnały sterujące. Jednostkę arytmetyczno-logiczną wraz z zespołem rejestrów nazywamy blokiem wykonawczym procesora.



ALU można wyobrazić sobie, jako zestaw wielu prostych układów elektroniki cyfrowej, z których każdy wykonuje pojedynczą operację arytmetyczną lub logiczną. Sygnały sterujące uaktywniają taką kombinację tych układów, która jest potrzebna w danej chwili, do wykonania aktualnie przetwarzanego rozkazu lub jego części. Zmiana sygnałów sterujących powoduje uaktywnienie nowej kombinacji układów i przełączenie się ALU inna operację.

Każdy typ jednostki arytmetyczno logicznej ma skończony zestaw operacji, jakie może wykonać. Zestaw ten jest różny, zależnie od stopnia skomplikowania ALU oraz jednostki sterującej, jednak zawsze musi on być funkcjonalnie pełny.

Funkcjonalnie pełny zestaw instrukcji to taki, za pomocą którego można zrealizować dowolny algorytm przetwarzania danych. Nie oznacza to jednak, że musi on zawierać wszystkich możliwe operacje matematyczne i logiczne. Każda skomplikowana operacja może zostać zastąpiona sekwencją operacji prostszych. Na przykład mnożenie zastąpić można wielokrotnym dodawaniem.

$$5 \cdot 4 = 5 + 5 + 5 + 5$$

Mogą więc istnieć (i w początkowych fazach rozwoju procesorów rzeczywiście istniały) procesory, które są w pełni funkcjonalne pomimo, że ich jednostki arytmetyczno-logiczne nie potrafią wykonać instrukcji mnożenia.

Każda lista rozkazów zawiera kilka grup działań występujących w różnych wersjach niemal w każdym komputerze są to:

- przesłania,
- działania arytmetyczne,
- działania logiczne,
- przesunięcia,
- sterowanie przebiegiem programu, przesłania wejścia-wyjścia, działania zmiennopozycyjne, działania na argumentach upakowanych.

ALU nie posiada pamięci ani urządzeń umożliwiających współpracę z pamięcią RAM. Współpracuje z zestawem rejestrów.

Blok sterujący procesora

Należy w tym miejscu wyraźnie powiedzieć, że sygnały sterujące jednostką arytmetyczno-logiczną to nie to samo, co polecenia programu. Realizacja pojedynczego rozkazu programu wymagać może podania całej sekwencji sygnałów sterujących do jednostki arytmetyczno-logicznej (mówimy wtedy o sterowaniu mikroprogramowym). Przetwarzaniem poleceń programu – rozkazów asemblera dla danego procesora – na wewnętrzne sygnały sterujące zajmuje się blok sterujący. W jego skład wchodzi:

- rejestr rozkazów (IR), w którym przechowywany jest kod aktualnie wykonywanego rozkazu;
- dekodery rozkazów, którego zadaniem jest rozpoznanie pobranego z pamięci operacyjnej rozkazu i wygenerowanie na jego podstawie sekwencji sygnałów sterujących dla ALU oraz pozostałych podzespołów procesora;
- pamięć ROM zawierająca słownik rozkazów (nie należy mylić z pamięcią ROM umieszczoną na płycie głównej).

Najważniejszym elementem bloku sterującego jest dekodery rozkazów. Pobiera on kod operacji zapisany w rejestrze IR i na jego podstawie generuje sygnały sterujące blokiem wykonawczym. Dekodery rozkazów korzysta z wbudowanej w procesor pamięci ROM. Zawiera ona słownik rozkazów pozwalający przetłumaczyć standardo-

we (wspólne dla całej rodziny procesorów) polecenia języka maszynowego na wewnętrzne sygnały sterujące konkretnego modelu procesora.

Rejestry procesora

Jednostka arytmetyczno-logiczna współpracuje z zestawem rejestrów. Rejestry są wbudowanymi bezpośrednio w strukturę procesora komórkami pamięci. W odróżnieniu od pamięci RAM rejestry identyfikowane są nie poprzez adres, lecz nazwę (odwołując się do wartości rejestrów podajemy jego nazwę).

Niektóre z rejestrów przechowują informacje ważne dla bieżącej pracy procesora. Do kategorii tej należą między innymi:

- IR – rejestr rozkazów – przechowuje kod aktualnie wykonywanego rozkazu,
- PC – licznik rozkazów – przechowuje adres rozkazu, który będzie wykonany w następnej kolejności,
- SP – wskaźnik stosu – zawiera wierzchołek stosu, czyli adres ostatniego elementu odłożonego na stos
- F – rejestr flag – rejestr opisujący i kontrolujący stan procesora.

Drugi rodzaj rejestrów to rejestry ogólnego przeznaczenia,

Architektura CISC i RISC

CISC

Procesory stosowane w mikrokomputerach i wczesnych komputerach osobistych były stosunkowo proste. Posiadały niewielkie listy rozkazów zawierające tylko proste operacje arytmetyczne oraz podstawowe operacje logiczne niezbędne do sterowania programem.

Najpopularniejszym, stosowanym wtedy językiem programowania był assembler – język symboliczny niskiego poziomu maksymalnie zbliżony do kodu maszynowego (binarnych kodów poleceń zapisanych w pamięci operacyjnej). Assembler to właściwie cała grupa języków programowania. W odróżnieniu od języków wyższego poziomu takich jak np. C++, Pascal, czy Basic, jest on powiązany z typem procesora. Komendy dostępne w assemblerze są w dużej mierze odwzorowaniem listy rozkazów procesora.

Programowanie procesorów dysponujących dużym, rozbudowanym zestawem rozkazów było łatwiejsze. Fakt ten stanowił jedną z przyczyn tendencji do tworzenia procesorów o coraz to bardziej rozbudowanych listach rozkazów. Procesory stawały się coraz szybsze, a ich programowanie łatwiejsze. Assembler upodabniał się coraz bardziej do języków wysokiego poziomu.

Procesory o rozbudowanych listach rozkazów określamy jako zbudowane w oparciu o architekturę CISC (*ang. Complex Instruction Set Computers*).

Charakterystycznymi cechami architektury CISC są:

- duża liczba skomplikowanych rozkazów,
- skomplikowana jednostka sterująca procesora,
- różne formaty (długości) rozkazów,
- duża liczba sposobów adresowania pamięci,
- możliwość bezpośredniego operowania na danych zawartych w pamięci operacyjnej.

Architektura CISC bazowała na sterowaniu mikroprogramowym. Pojedynczy, złożony rozkaz rozbijany był wewnątrz procesora na kilka prostszych rozkazów, a jego wykonanie trwało kilka lub nawet kilkanaście taktów zegara. Sterowanie mikroprogramowe wymagało wyposażenia procesora w skomplikowaną jednostkę sterującą.

RISC

Termin RISC (*ang. Reduced Instruction Set Computers*) oznacza architekturę procesorów posiadających listę rozkazów ograniczoną do niewielu prostych, lecz bardzo szybko wykonywanych komend. Cechami charakterystycznymi architektury RISC są:

- krótka lista prostych rozkazów,
- prosta i bardzo szybka jednostka sterująca,
- ujednolicone formaty rozkazów (rozказы o równych długościach)
- niewielka liczba trybów adresowania,
- brak możliwości odwoływania się bezpośrednio do pamięci operacyjnej – dane muszą być najpierw przepisane do rejestrów procesora.

Zredukowanie listy rozkazów i uproszczenie budowy procesora wydaje się krokiem wstecz. Zastosowanie architektury RISC umożliwiło jednak konstruowanie znacznie wydajniejszych procesorów.

Prostsza konstrukcja jednostki sterującej ułatwiła dalsze zwiększanie taktowania procesorów. Zmiany w organizacji rozkazów (szczególnie ujednolicenie ich długości) pozwoliły na zastosowanie nowych technik podnoszenia wydajności takich jak wielopotokowość oraz superskalarność.

Szybkość przetwarzania instrukcji w technologii RISC osiągnięto poprzez sterowanie mikroukładowe. W procesorze dysponującym niewielką liczbą prostych rozkazów nie zachodzi, w przeciwieństwie do architektury CISC, konieczność rozbijania ich na prostsze operacje składowe. Realizacją każdego rozkazu może zajmować się wyspecjalizowany układ logiczny bloku wykonawczego.

Procesor zbudowany w technologii RISC wykonuje w krótkim czasie bardzo dużo prostych rozkazów. Rozkazy te (wraz z ich argumentami) muszą być dostarczane w odpowiednim tempie. Wymaga to dużej przepustowości magistrali łączącej procesor z pamięcią operacyjną. Problem ten nie występował w technologii CISC, gdzie pojedynczy rozkaz był wczytany i wykonywany przez kilkanaście tak-

tów zegara. Procesory RISC są natomiast w stanie pobrać więcej niż jeden rozkaz w pojedynczym taktie zegara (tak, to możliwe – patrz superskalarność).

Dla zapewnienia odpowiednio szybkiej komunikacji z pamięciom RAM współczesne procesory wyposażone są w dwa lub trzy poziomy pamięci podręcznej CACHE (L1, L2 i L3).

Tabela: Porównanie architektury CISC i RISC.

	Architektura	
	CISC	RISC
Liczba rozkazów:	duża	mała
Format (długość) rozkazów:	zmienna	stała
Rodzaj sterowania jednostką wykonawczą:	mikroprogramowe	mikroukładowe
Budowa jednostki sterującej:	skomplikowana	prosta
Konieczność stosowania wysokowydajnych magistral pamięci:	NIE	TAK
Konieczność stosowania pamięci Cache:	NIE	TAK
Możliwość przetwarzania potokowego:	NIE	TAK
Możliwość przetwarzania superskalarne-go:	NIE	TAK

Tabela prezentuje porównanie obu omawianych tu koncepcji architektury procesora. W praktyce trudno jednak znaleźć procesory całkowicie spełniające założenia którejkolwiek z nich. Mówimy raczej o pewnych cechach architektury CISC i RISC, którymi charakteryzuje się procesor.

Współczesne procesory rodziny x86 produkowane przez firmy Intel oraz AMD mają listy rozkazów dużo dłuższe, niż starsze procesory wykonane w technologii CISC. Jednocześnie procesory te posiadają większość kluczowych cech technologii RISC (mała liczba trybów adresowania, podział pamięci *Cache* na blok danych i blok instrukcji, możliwość pracy potokowej i superskalarnej).

Procesory te łączą cechy obu typów architektury. Długa lista rozkazów zapewnia zgodność ze starszym oprogramowaniem, a szybkie, oparte na koncepcji RISC, przetwarzanie programu wewnątrz procesora zwiększa jego moc obliczeniową.