Procesor 8086 - wczesny przykład nowych tendencji w organizacji procesora

według T.Jamrógiewicza

organizacja procesora 8086

Ze struktury mikroprocesora 8086 dają się wyodrębnić dwa układy, pracujące w dużej mierze niezależnie:

- jednostkę wykonawczą,
- jednostkę sprzężenia z magistralą.

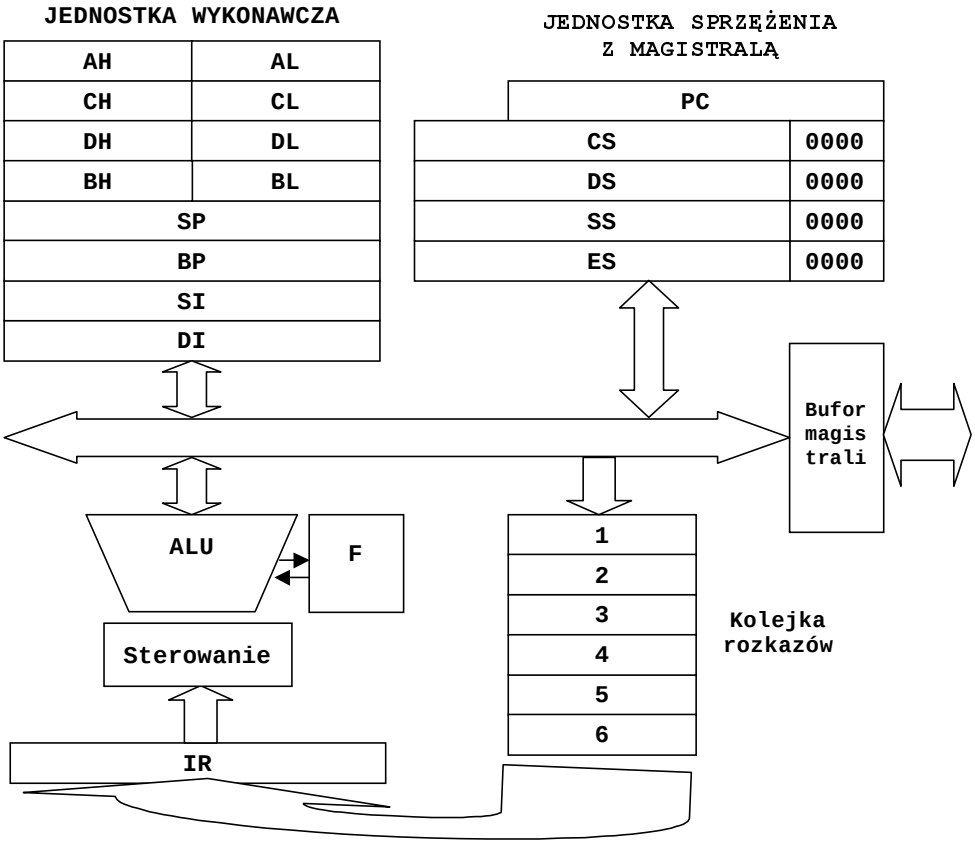
Jednostka sprzężenia z magistralą: Procesor komunikuje się z otoczeniem za pośrednictwem układu sprzężenia z magistralą. Układ ten realizuje wszystkie operacje dotyczące zewnętrznej magistrali procesora. Magistrala jest wspólną drogą, po której przesyłane są sygnały pomiędzy procesorem a pamięcią operacyjną i rejestrami sterowników urządzeń zewnętrznych. Układ sterowania magistralą zawiera sumator oraz rejestry służące do obliczania 20.bitowego adresu fizycznego. Dane przesyłane są po 16.bitowej magistrali danych. Magistrala adresowa jest 20.bitowa, co pozwala na bezpośrednie adresowanie pamięci operacyjnej o pojemności 1MB.

Jednostka wykonawcza: Jej głównym zadaniem jest dekodowanie oraz wykonywanie kolejnych rozkazów opuszczających, zapełnianą przez układ sprzężenia z magistralą, tzw. kolejkę rozkazów. Wykonanie rozkazów odbywa się przy udziale jednostki arytmetyczno-logicznej **ALU** (aritmetic logic unit), związanego z nią rejestru znaczników (flags), rejestrów arytmetycznych ogólnego przeznaczenia oraz programowo niedostępnych rejestrów roboczych (chwilowych).

Jednostka arytmetyczno-logiczna. Jest ona tą częścią procesora, która wykonuje operacje arytmetyczne, takie jak dodawanie i odejmowanie słów danych, a także elementarne operacje logiczne: sumowanie (OR), mnożenie (AND) i sumowanie modulo 2 (XOR).

Rejestry. Rejestr jest częścią pamięci wewnętrznej procesora o niewielkiej (liczonej w bitach) pojemności (w odniesieniu do rejestru częściej mówimy o jego długości). Jest to zwykle układ bistabilnych obwodów elektrycznych (przerzutników) służący do przechowywania informacji. Pewne rejestry procesora są związane z określonymi operacjami a wykonanie niektórych rozkazów jest związane z określonymi rejestrami. Rejestry są na ogół szybsze niż układy pamięci operacyjnej. W porównaniu z liczbą komórek pamięci ich liczba jest niewielka. Stąd w części adresowej rozkazu niewielka liczba bitów jest potrzebna dla wskazania rejestru. Główne zastosowanie rejestrów polega na przechowywaniu adresów lub danych przed lub w trakcie ich przetwarzania. Rejestry mogą być używane do modyfikacji adresów, pamiętania adresów powrotu z podprogramu, jako liczniki rozkazów, akumulatory pomocnicze lub małe pamięci podręczne.

Na rysunku poniżej został przedstawiony schemat blokowy procesora 8086:



Rysunek 1. Budowa procesora 8086.

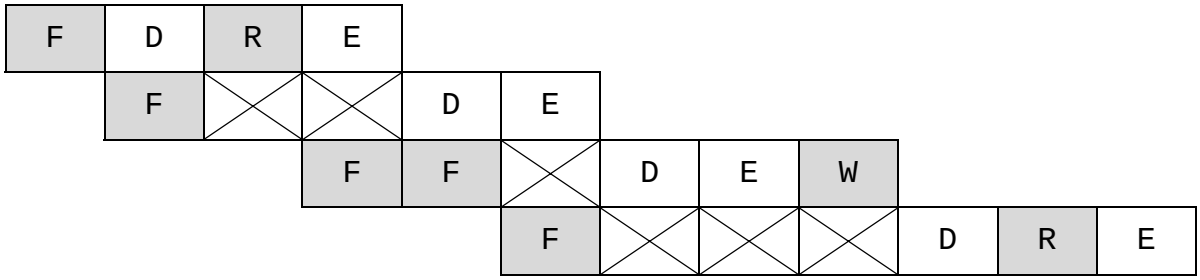
cykl pracy procesora 8086

Wykonanie kolejnych rozkazów procesora związane jest z sekwencją pewnych stałych kroków zwanych fazami cyklu. W zależności od rodzaju rozkazu mogą wystąpić maksymalnie cztery spośród pięciu faz cyklu:

- **pobranie** (fetch) - kod rozkazu zostaje pobrany z pamięci do zestawu rejestrów tworzących kolejkę rozkazów skąd trafi do rejestru rozkazów IR procesora;
- **dekodowanie** (decode) – w tej fazie rozkaz jest dekodowany i ustalane są, jeśli wymaga tego typ rozkazu, połączenia arytmometru z rejestrami, które mają dostarczyć argumentów operacji;
- ***odczyt*** (read) - jeżeli rozkaz wymaga argumentu z pamięci operacyjnej, to zostaje obliczony jego adres, a następnie argument jest odczytywany i wprowadzany do procesora;
- **wykonanie** (execute) – w tej fazie operacja zdefiniowana rozkazem jest wykonywana;
- ***zapis*** (write) - jeżeli wynik operacji ma być umieszczony w pamięci operacyjnej, to zostaje obliczony adres i wykonany zapis do pamięci.

Dzięki kolejce rozkazów jednostka wykonawcza oraz jednostka sprzężenia z magistralą mogą pracować niezależnie i jest możliwe równoczesne wykonywanie dwóch faz należących do cykli kolejnych rozkazów. Zwiększa to szybkość pracy całego systemu.

W mikroprocesorze 8086 pojemność kolejki wynosi 6 bajtów. Jeśli w kolejce zwolni się miejsce na co najmniej dwa bajty, a jednostka wykonawcza nie żąda dostępu do magistrali, to jednostka sprzężenia z magistralą wykonuje samodzielnie cykle pobrania, ustawiając w kolejce rozkazów kolejne bajty kodu programu. Operacja ta może przebiegać równocześnie z inną fazą cyklu, w której jednostka wykonawcza procesora nie odwołuje się do pamięci. Jeżeli jednostka wykonawcza zażąda dostępu do magistrali w czasie, gdy jednostka sprzężenia z magistralą dokonuje pobrania, to faza pobrania zostanie dokończona, a potem uwzględnione żądanie układu wykonawczego.



F **- pobranie** (fetch) *pracuje* *jednostka sprzężenia z magistralą*,
D - **dekodowanie** (decode),
R - **odczyt** (read),
E - **wykonanie** (execute),
W - **zapis** (write).

- oczekiwanie

Rysunek 2. Przykładowa realizacja 4 kolejnych rozkazów w procesorze 8086.

Powyżej przedstawiono przepływ rozkazów ilustrujący równoległą pracę jednostka wykonawczej i jednostki sprzężenia z magistralą. Zacieniowano klatki ilustrujące fazy cyklu, w których procesor odwołuje się do pamięci. Trzeci z kolei rozkaz ma podwójną długość, stąd dwie fazy pobrania.

W przypadku zdekodowania przez układ wykonawczy efektywnego rozkazu skoku, wprowadzone po nim do kolejki rozkazy nie są tymi, które w tym momencie powinny być wykonywane. W celu zapewnienia poprawnej kolejności wykonywania instrukcji programu muszą zostać usunięte z kolejki. Kolejne będą pobierane spod nowego adresu, obliczonego w trakcie wykonywania instrukcji skoku.

Podstawowy cykl roboczy jednostki wykonawczej procesora rozpoczyna się kiedy rozkaz po opuszczeniu kolejki rozkazów trafia do **rejestru rozkazów** IR gdzie jest dekodowany. Układy dekodujące bloku sterowania (rozkazami) inicjują czynności wynikające z interpretacji kodu instrukcji.

rejestry procesora 8086

Rejestry te są wykorzystywane jako rejestry uniwersalne do przechowywania danych i wykonywania różnych operacji (np. arytmetycznych bądź logicznych), ale jednocześnie pełnią pewne funkcje specjalne odpowiadające ich nazwom:

AH	AL	akumulator AX
BH	BL	rejestr bazowy BX
CH	CL	rejestr zliczający CX
DH	DL	rejestr danych DX
F		rejestr znaczników
SP		wskaźnik stosu
BP		wskaźnik bazy
SI		rejestr indeksowy źródła
DI		rejestr indeksowy przeznaczenia
CS		rejestr segmentowy programu
DS		rejestr segmentowy danych
SS		rejestr segmentowy stosu
ES		rejestr segmentowy dodatkowy
PC		licznik rozkazów

Rejestry podstawowe

AX *(accumulator)* - akumulator

BX *(basis register)* - rejestr bazowy

CX *(count register)* - rejestr zliczający

DX *(data register)* - rejestr danych.

Są to rejestry 16.bitowe. Przez niektóre rozkazy mogą być traktowane jako pary rejestrów 8.bitowych z niezależnym dostępem do części mniej (*low*) oraz bardziej znaczącej (*high)*. Odpowiednie rejestry przyjmują wtedy nazwy: AL, AH, BL, BH, CL, CH, DL, DH. Taka organizacja rejestrów pozwala na wykonywanie w prosty sposób zarówno operacji 8.bitowych, jak i 16.bitowych.

Dla wielu rozkazów szczególną funkcję pełni akumulator **AX** (lub AL). Jest to rejestr, w którym umieszczony jeden z argumentów

operacji arytmetycznej po wykonaniu rozkazu może być zamieniony przez wynik operacji. Rozkazy wykorzystujące rejestr akumulatora (w porównaniu z rozkazami wykorzystującymi inne rejestry) mają zwarty kod, a czas ich wykonania jest krótszy.

Rejestr bazowy **BX** służy do przechowywania adresu bazowego. Jego zawartość przez niektóre rozkazy może być potraktowana jako odniesienie przy obliczaniu adresów podczas wykonywania programu komputera.

CX - rejestr zliczający może być wykorzystywany jako licznik sterujący wykonywaniem pętli programowych i do operacji na łańcuchach (danych traktowanych jako ciąg słów).

Rejestr **DX** jest stosowany (zwykle łącznie z AX) jako starsza część słowa danych podczas wykonywania operacji mnożenia i dzielenia. Kiedy mnożymy dwie wartości 16.bitowe do zapamiętania 32.bitowego wyniku potrzebne są dwa rejestry DX (dla starszej części wyniku) i AX (dla młodszej). Z kolei w przypadku dzielenia w rejestrach tych umieszczana jest 32.bitowa dzielna.

Rejestry wskaźnikowe i indeksowe

Mikroprocesor 8086 posiada dwa rejestry wskaźnikowe i dwa indeksowe. Wszystkie są rejestrami 16.bitowymi. Przechowywane w nich dane mogą być użyte jako argumenty większości rozkazów arytmetycznych i logicznych. Jednak ich nazwy są związane z rolą jaką rejestry te pełnią podczas adresowania pamięci operacyjnej.

SP (stack pointer) - wskaźnik stosu

BP (base pointer) - wskaźnik bazy

SI (source index register) - rejestr indeksowy źródła

DI (destination index register) - rejestr indeksowy przeznaczenia

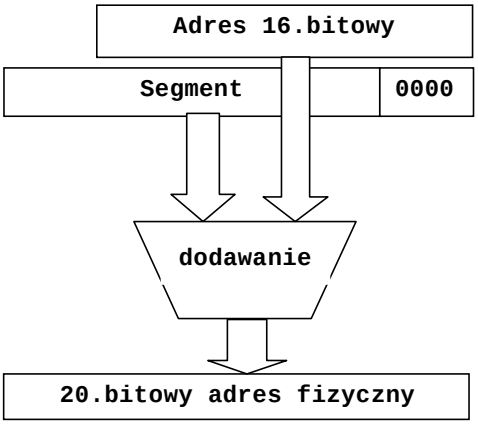
SP jest to rejestr wskaźnikowy stosowany do adresowania danych w obrębie wydzielonego obszaru pamięci, zwanego segmentem stosu. Rejestr SP jest modyfikowany przy standardowych operacjach prowadzonych z wykorzystaniem stosu tak, że zawiera adres ostatnio zapamiętanej danej.

Rejestr **BP** podobnie jak rejestr bazowy BX może dostarczać jeden ze składników (adresu bazowego) tzw. adresu efektywnego EA (effective address). Adresem efektywnym nazywamy, wyznaczany podczas wykonania niektórych instrukcji, adres argumentu odczytywanego z pamięci lub adres komórki, w której zapisywany jest wynik. Adres efektywny może być sumą informacji adresowej zawartej w kodzie instrukcji oraz zawartości rejestrów bazowego i indeksowego. Wskaźnik BP bywa też wykorzystywany podczas operacji niestandardowych np. przy pobieraniu parametrów przekazywanych przez stos.

Oba **rejestry indeksowe** mogą wchodzić w skład sumy dającej w wyniku adres efektywny, z tym że po wykonaniu rozkazu wykorzystującego indeksowany tryb adresacji ich zawartość zostaje automatycznie zwiększona lub zmniejszona (o jeden lub dwa). W ten sposób adres argumentu podczas wykonywania tej samej instrukcji programu podlega automodyfikacji.

Rejestry segmentowe

Również te rejestry są wykorzystywane do adresowania pamięci operacyjnej. Pamięć o maksymalnej wielkości 1 MB (20 bitów adresu) jest dzielona na logiczne segmenty nie większe niż 64kB (16 bitów adresu). Poszczególne segmenty są identyfikowane przy pomocy adresów bazowych przechowywanych w czterech wyspecjalizowanych rejestrach segmentowych: CS, DS, ES i SS.



Rysunek 3. Sumator do obliczania 20.bitowego adresu fizycznego.

W fazie pobrania procesor 8086 adresuje pamięć przy pomocy dwóch rejestrów: licznika rozkazów (PC) i rejestru segmentowego programu (CS). Wystawiany wtedy na magistralę 20.bitowy adres fizyczny **FA** obliczany jest następująco:

$$FA = PC + 16 \cdot CS$$

Jeśli podczas wykonywania instrukcji programu trzeba odwołać się do pamięci operacyjnej to domyślna wartość adresu fizycznego wynosi:

$$FA = EA + 16 \cdot DS$$

Tak więc rejestry segmentowe zawierają adresy początkowe segmentów (można je zmieniać z dokładnością do 16 bajtów):

- **CS** (code segment register) - rejestr segmentowy programu wskazuje segment programu, z którego aktualnie są pobierane kolejne rozkazy do wykonania;
  - adresy wewnątrz segmentu są, w fazie pobrania kodu instrukcji, generowane przez 16.bitowy licznik rozkazów PC.
- **DS** (data segment register) - rejestr segmentowy danych wskazuje segment, w którym są zapamiętane zmienne używane w programie;
  - w fazie wykonania, kiedy zachodzi potrzeba odczytania z pamięci operacyjnej argumentu instrukcji, 16-bitowy adres efektywny - obowiązujący wewnątrz segmentu - jest wyliczany zgodnie z przyjętym dla danej instrukcji trybem adresowania,
  - jeżeli wynik operacji ma być umieszczony w pamięci operacyjnej, to zgodnie z zapisanym w instrukcji trybem

adresowania, zostaje obliczony adres efektywny, który wskaże miejsce zapisu do pamięci w obrębie segmentu.

- **ES** (extra segment register) - rejestr segmentowy dodatkowy wskazuje położenie dodatkowego segmentu danych;
- **SS** (stack segment register) - rejestr segmentowy stosu wskazuje segment pamięci, w którym pamięć może być adresowana za pośrednictwem wskaźnika stosu SP;
  - jeśli odczyt operandu lub zapis wyniku realizowany jest na stosie to 16.bitowy adres jest brany ze wskaźnika stosu SP.

W tabeli poniżej zestawiono rejestry uczestniczące w generowaniu 20.bitowego adresu fizycznego:

Typ operacji	Segment domyślny	Inne możliwe segmenty	Adres 16.bitowy
Pobranie instrukcji	CS	żaden	PC
Operacja na stosie	SS	żaden	SP
Adres źródłowy łańcucha	DS	CS,ES,SS	SI
Adres docelowy łańcucha	ES	żaden	DI
BP jako rejestr bazowy	SS	CS,ES,DS	dowolny EA
BX jako rejestr bazowy	DS	CS,ES,SS	dowolny EA
SI lub DI jako indeks	DS	CS,ES,SS	dowolny EA
Inna zmienna w pamięci	DS	CS,ES,SS	dowolny EA

Licznik rozkazów

Rejestr PC (program counter) łącznie z rejestrem segmentowym CS adresuje kolejne rozkazy przeznaczone do wykonania. PC wskazuje adres liczony względem początku segmentu programu. Jako rejestr 16.bitowy może przyjąć maksymalną wartość $2^{16}-1$. Dodanie jedynki zeruje cały rejestr, który wskazuje wtedy początek segmentu programu.

Znaczniki

Każdy znacznik (flag) jest bitem w rejestrze znaczników, który wskazuje czy wystąpił określony stan. Jedne instrukcje w wyniku swojego działania ustawiają znaczniki inne pozwalają badać ich stan.

Procesor 8086 wykorzystuje dziewięć znaczników, które przechowywane są w rejestrze znaczników, na pozycjach wyspecyfikowanych poniżej:

nr bitu	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	X	X	X	X	V	D	I	T	S	Z	X	AC	X	P	X	C

X - bit nie wykorzystany

Znaczniki arytmetyczne: informujące o pewnych cechach otrzymanego wyniku po wykonaniu operacji arytmetycznej bądź logicznej. Istnieje zestaw instrukcji skoków warunkowych pozwalający badać stan tych znaczników. Dzięki temu w zależności od stanu pojedynczych znaczników lub ich logicznej kombinacji można zmieniać przebieg realizowanego programu.

C (carry) **-** przeniesienie

P (parity) - parzystość

AC (auxiliary carry) **-** przeniesienie pomocnicze

Z (zero) - zero

S (sign) - znak

V (overflow) **-** nadmiar (przepełnienie)

Znaczniki kontrolne ustawiane lub zerowane programowo celem wymuszania odpowiedniego sposobu pracy procesora.

T (trap) - praca krokowa (pułapka)

I (interrupt enable) **-** zezwolenie na przerwanie

D (direction) **-** znacznik kierunku automodyfikacji adresu operandu (inkrementacja lub dekrementacja rejestru indeksowego).

Znacznik pracy krokowej T ustawiony w stan 1 powoduje wprowadzenie procesora w tryb pracy krokowej (simple step modus) umożliwiający po każdym wykonanym rozkazie wygenerowanie przerwania (single step interrupt) i przejście do specjalnych procedur obsługi (np. programów uruchomieniowych). Wyzerowanie znacznika T powoduje powrót procesora do normalnej pracy.

Znacznik zezwolenia na przerwanie I ustawiony w stan 1 powoduje odblokowanie systemu przerwań procesora. Zewnętrzne przerwania maskowane mogą przerwać realizację wykonywalnego aktualnie programu i uruchomienie procedury obsługi przerwania. Wyzerowanie znacznika powoduje, że tego typu przerwania są przez procesor ignorowane.

Znacznik kierunku D jest uwzględniany przy wykonywaniu działań na łańcuchach (ciągach słów). Jeżeli ma wartość 1, to przetwarzanie łańcuchów odbywa się w kierunku rosnących adresów (zawartość rejestru indeksowego jest zwiększana), jeżeli jest równy 0 - przy malejących adresach.

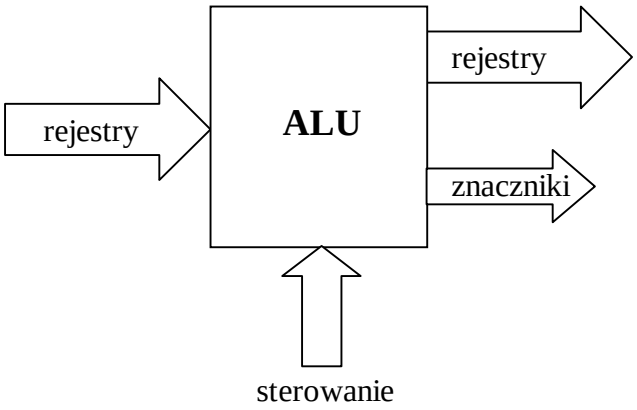
jednostka arytmetyczno-logiczna

W tej części jednostki wykonawczej procesora skupione są układy wykonujące operacje arytmetyczne i logiczne. Argumentami mogą być bajty (8 bitów) lub słowa (16.bitowe).

Źródłem argumentów operacji wykonywanych przez ALU może być:

- dostępny programowo rejestr procesora,
- komórka pamięci,
- operand bezpośredni zapisany w wykonywanej instrukcji programu.

W fazie wykonania dane wejściowe dla jednostki arytmetyczno-logicznej znajdują się w rejestrach (te odczytane z pamięci operacyjnej pamiętane są w rejestrach chwilowych) a wyniki kierowane są do rejestrów i znaczników arytmetycznych.



Rysunek 4. Wejścia i wyjścia jednostki arytmetyczno-logicznej.

6 spośród 9 znaczników ustawianych jest przez ALU:

Znacznik przeniesienia C przyjmuje wartość 1 wówczas, gdy na skutek wykonanego działania nastąpiło przeniesienie z najbardziej znaczącego bitu na zewnątrz (np. przy dodawaniu) lub też nastąpiła pożyczka z zewnątrz do bitu najbardziej znaczącego (np. przy odejmowaniu). W przeciwnym przypadku znacznik jest zerowany.

Znacznik parzystości P przyjmuje wartość 1 wówczas, gdy w wyniku wykonanego działania liczba bitów o wartości 1 w mniej znaczącym bajcie wyniku jest parzysta. Znacznik jest zerowany wówczas, gdy liczba ta jest nieparzysta.

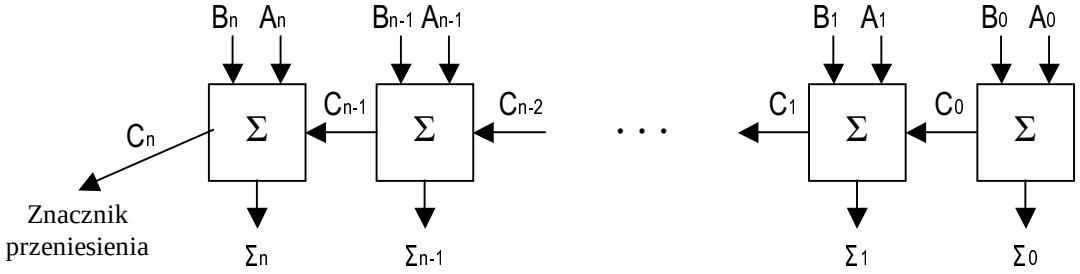
Znacznik przeniesienia pomocniczego AC przyjmuje wartość 1 wówczas, gdy nastąpiło przeniesienie z bitu 3 na 4 lub pożyczka z bitu 4 na 3. W przeciwnym przypadku wskaźnik jest zerowany. Wskaźnik AC jest wykorzystywany przy działaniach na liczbach w kodzie BCD.

Znacznik zera Z przyjmuje wartość 1 wówczas, gdy wynik działania jest równy zero, w przeciwnym przypadku jest zerowany.

Znacznik znaku S przyjmuje wartość 1 wówczas, gdy najbardziej znaczący bit w otrzymanym wyniku jest równy 1, w przeciwnym przypadku jest zerowany. Stan znacznika S jest zatem zgodny z bitem na pozycji znaku w kodzie U2.

Znacznik nadmiaru V przyjmuje wartość 1 wówczas, gdy (podczas dodawania) wystąpiło przeniesienie na pozycję bitu znaku ale nie było przeniesienia z bitu znaku do znacznika C (tzn. C=0) albo odwrotnie - wystąpiło przeniesienie z bitu znaku (tzn. C=1) ale nie było przeniesienia na pozycję bitu znaku. W pozostałych dwóch przypadkach (oba przeniesienia równe zero i oba przeniesienia równe jeden) znacznik ten jest zerowany. Stan znacznika V jest istotny przy działaniach na liczbach zapisanych w kodzie U2.

Podstawowym układem jednostki arytmetyczno-logicznej jest sumator. Dodawanie naturalnych liczb binarnych (podobnie jak dodawanie liczb dziesiętnych) polega na sumowaniu bitów na poszczególnych pozycjach (poczynając od najmniej znaczącej) z uwzględnieniem wartości przeniesienia z poprzedniej pozycji. Operacja jednopozycyjnego dodawania sprowadza się do dodania dwóch bitów i przeniesienia a jej rezultatem jest bit wyniku i bit przeniesienia na następną pozycję.



Rysunek 5. Sumator wielobitowy

Przedstawiony na rysunku sumator wielobitowy zbudowany jest z identycznych modułów sumatorów jednopozycyjnych. Poniżej została przedstawiona tablica prawdy definiująca działanie sumatora jednopozycyjnego:

A_i	B_i	C_{i-1}	Σ_n	C_i
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1