

ZESTAW INSTRUKCJI PROCESORÓW SERII 65XX

W zestawie znajduje się 56 kodów mnemonicznych instrukcji, które są wykonywane przez procesor 65XX.

Każda z instrukcji opisana jest na kilka sposobów:

- co wykonuje procesor realizując instrukcję
- podstawowe użycie instrukcji przy programowaniu w ML
- jakich sposobów adresowania można użyć
- jak oddziałuje na rejestr wskaźników (FLAG)
- kod dziesiętny i hexadecymalny instrukcji
- ilość zajmowanych bajtów

ADC

Dodaje bajt z pamięci do bajtu w akumulatorze i ustawia wskaźnik przeniesienia C (Carry). Wskaźnik przeniesienia jest ustawiany gdy wynik dodawania jest większy od FF (255). Wynik pozostaje w akumulatorze.

GŁÓWNE ZASTOSOWANIA:

Dodaje dwie liczby do siebie. Gdy C był ustawiony w momencie wykonania ADC to wynik będzie większy o 1 od sumy dodawanych liczb (bit C jest dodawany do wyniku). Przed rozpoczęciem dodawania należy więc użyć instrukcji CLC (CLear Carry). Ale ta właściwość pozwala na łańcuchowe dodawanie bajtów bez dalszego użycia CLC.

Inne wskaźniki, na które oddziałuje ADC:

- wskaźnik przepełnienia V (oVerflow). Ten wskaźnik rzadko interesuje programistów. Wskazuje on jedynie, że wynik przekroczył 7F (127).
- wskaźnik zera Z (Zero). Ustawienie tego wskaźnika wskazuje, że wynik = 0.
- wskaźnik znaku N (Negative) wskazuje, że bit 7 wyniku jest równy 1. Dzięki temu można manipulować bajtami przyjmując bit znaku (+ lub -) do wykonywania działań arytmetycznych w kodzie znak modulo. W tym przypadku każda komórka może przechowywać maksymalną liczbę 7F (127), ponieważ siódmy bit jest użyty jako wskaźnik znaku. Tego typu arytmetyki używają instrukcje skoku warunkowego (B - Branch).

Instrukcja ADC może być użyta po wykonaniu instrukcji SED, która ustawia procesor w dziesiętny tryb pracy.

Oto przykład:

SED
CLC
LDA #75
ADC #\\$05
CLD

*wynik będzie 80
zawsze kasuj tryb dziesiętny natychmiast po zakończeniu w nim
działań*

W rzeczywistości tryb dziesiętny nie jest tak atrakcyjny dla programisty jak to wygląda na pierwszy rzut oka. Asembler pozwala na pracę na liczbach dziesiętnych bez wprowadzania procesora w tryb dziesiętny. Po prostu opuść znak \$ przed liczbą, a zostanie ona potraktowana jako liczba dziesiętna.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Natychmiastowy	ADC #15	\$69 / 105	2
Strony zerowej	ADC 15	\$65 / 101	2
Strony zerowej względem X	ADC 15,X	\$75 / 117	2
Bezpośredni	ADC 1500	\$6D / 109	3
Bezpośredni względem X	ADC 1500,X	\$7D / 125	3
Bezpośredni względem Y	ADC 1500,Y	\$79 / 121	3
Pośredni strony zerowej	ADC (15)	\$72 / 114	2
Pośredni względem X	ADC (15,X)	\$61 / 97	2
Pośredni względem Y	ADC (15),Y	\$71 / 113	2

ADC oddziałuje na wskaźniki: N, Z, C, V

AND

Wykonuje iloczyn logiczny bajtu w pamięci z bajtem w akumulatorze. Wynik pozostaje w akumulatorze. Wszystkie bity w obu bajtach są porównywane i jeśli obydwa bity o tej samej wadze są " 1 " to odpowiedni bit wyniku będzie równy " 1 ". W pozostałych przypadkach w wyniku będzie " 0 ".

GŁÓWNE ZASTOSOWANIA

W większości przypadków AND jest używane do selektywnego gaszenia bitów. Przypuśćmy, że chcesz zgasić bit 0 i 5 w liczbie \$80 (128). W zapisie binarnym liczbę tę przedstawiamy jako 01111111. Należy stworzyć "maskę" tak, by bity 0 i 5 były zgaszone, a pozostałe zostawiamy bez zmian. Mamy więc następującą sytuację:

Akumulator	0 1 1 1 1 1 1 1 (\$80)
Maska	1 1 0 1 1 1 1 0 (\$DE)

Wykonanie AND spowoduje wyzerowanie tylko tych bitów, które w masce są zgaszone. Zauważ, że nie jest istotna wartość pozostałych bitów w akumulatorze, bowiem wartość żadnego z nich w iloczynie z " 1 " nie zmienia się.

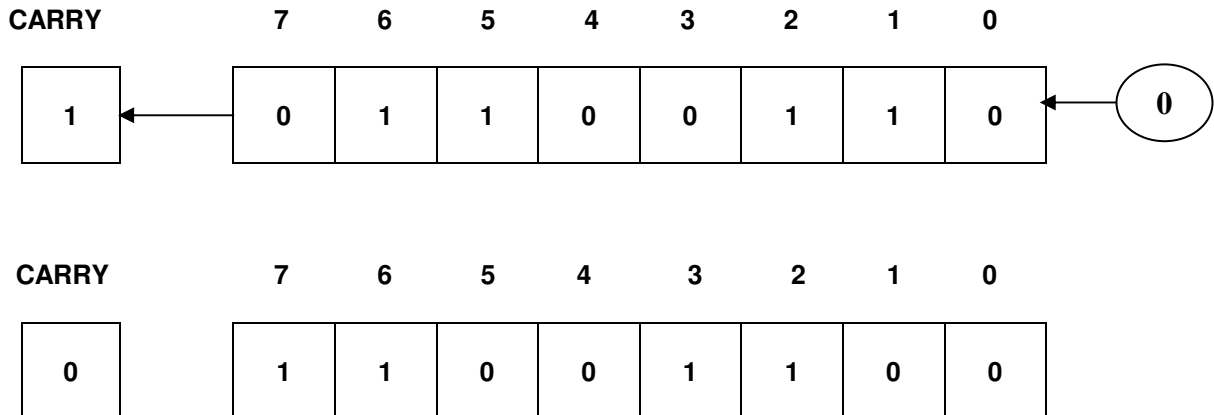
TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	AND #15	\$29 / 41	2
Strony zerowej	AND 15	\$25 / 37	2
Strony zerowej względem X	AND 15,X	\$35 / 53	2
Bezpośredni	AND 1500	\$2D / 45	3
Bezpośredni względem X	AND 1500,X	\$3D / 61	3
Bezpośredni względem Y	AND 1500,Y	\$39 / 57	3
Pośredni strony zerowej	AND (15)	\$32 / 50	2
Pośredni względem X	AND (15,X)	\$21 / 33	2
Pośredni względem Y	AND (15),Y	\$31 / 49	2

AND oddziałuje na wskaźniki: N, Z

ASL

Przesuwa bity w bajcie o jeden w lewo. Bajt może być umieszczony w akumulatorze albo w pamięci, zależnie od trybu adresowania. ASL przesunęła bit 7 do wskaźnika przeniesienia, a na pozycję bitu 0 wstawia wartość " 0 ".



GŁÓWNE ZASTOSOWANIA

ASL pozwala na pomnożenie liczby przez 2. Liczby większe od \$FF (255) można przetwarzać używając ASL łącznie z instrukcją ROL.

Inne zastosowanie: przesuwanie czterech młodszych bitów w bajcie (jednostka czterobitowa to półbajt lub nibel) na pozycje czterech starszych bitów. Młodsze bity zastąpione są przez zera. Przesunięcie takie uzyskamy przez czterokrotne użycie ASL.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Akumulatora	ASL	\$0A / 10	1
Strony zerowej	ASL 15	\$06 / 06	2
Strony zerowej względem X	ASL 15,X	\$16 / 22	2
Bezpośredni	ASL 1500	\$0E / 14	3
Bezpośredni względem X	ASL 1500,X	\$1E / 30	3

ASL oddziałuje na wskaźniki: N, Z, C

BCC

Gdy wskaźnik C jest skasowany - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu, czyli wtedy, kiedy drugi składnik jest mniejszy od pierwszego.

PRZYKŁADY:

LDA #150	lub	LDA #22
CMP #149		SBC #15
BCC 20		BCC 148

W obydwu sekwencjach wyzerowany zostaje wskaźnik C i wykonany jest skok. W przypadku lewej sekwencji skok jest wykonywany wprzód o 20 bajtów, w prawej w tył o 20 bajtów (w liczbie 148, która jest większa od 127, ustawiony jest bit 7, co oznacza jednoczesne ustawienie wskaźnika N, a więc liczba jest ujemna).

GŁÓWNE ZASTOSOWANIA:

Do testowania wyników instrukcji CMP, ADC i innych wpływających na wskaźnik przeniesienia. Również struktury typu IF - THEN lub ON - GOTO mogą posługiwać się instrukcją BCC. Działanie tej instrukcji jest podobne do instrukcji ">" w Basicu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BCC adres	\$90 / 144	2

BCC nie oddziałuje na rejestr wskaźników.

BCS

Gdy wskaźnik przeniesienia jest ustawiony - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

PRZYKŁADY:

LDA #150	lub	LDA #22
CMP #249		SBC #85
BCS 20		BCS 148

W obydwu sekwencjach ustawiony zostaje wskaźnik C i wykonany jest skok. W przypadku lewej sekwencji skok jest wykonywany wprzód o 20 bajtów, w prawej w tył o 20 bajtów (w liczbie 148, która jest większa od 127, ustawiony jest bit 7, co oznacza jednoczesne ustawienie wskaźnika N, a więc liczba jest ujemna).

GŁÓWNE ZASTOSOWANIA:

Do testowania wyników instrukcji CMP, ADC i innych wpływających na wskaźnik przeniesienia.

Również struktury typu IF - THEN lub ON - GOTO mogą posługiwać się instrukcją BCS. Działanie tej instrukcji jest podobne do instrukcji "<" w Basicu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BCS adres	\$B0 / 176	2

BCS nie oddziałuje na rejestr wskaźników.

BEQ

Gdy wskaźnik zera Z jest ustawiony - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

PRZYKŁADY:

LDA #150	lub	LDA #22
CMP #150		SBC #22
BEQ 20		BEQ 148

W obydwu sekwencjach ustawiony zostaje wskaźnik Z i wykonany jest skok. W przypadku lewej sekwencji skok jest wykonywany wprzód o 20 bajtów, w prawej w tył o 20 bajtów (w liczbie 148, która jest większa od 127, ustawiony jest bit 7, co oznacza jednoczesne ustawienie wskaźnika N, a więc liczba jest ujemna).

GŁÓWNE ZASTOSOWANIA:

Do testowania wyników instrukcji LDA, ADC i innych wpływających na wskaźnik zera. Również struktury typu IF - THEN lub ON - GOTO mogą posługiwać się instrukcją BCS. Działanie tej instrukcji jest podobne do instrukcji " = " w Basicu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BEQ adres	\$F0 / 240	2

BEQ nie oddziałuje na rejestr wskaźników.

BIT

Porównuje bity bajtu umieszczonego w pamięci z bitami bajtu umieszczonego w akumulatorze. Instrukcja nie oddziałuje na bajty na których wykonuje operację, a jedynie ustawia wskaźniki:

- wskaźnik zera Z jest ustawiany tak jakby wykonywana była instrukcja AND
 - wskaźnik przepelnienia V otrzymuje wartość szóstego bitu wyniku
 - wskaźnik znaku N otrzymuje wartość siódmego bitu wyniku
- Wynik nie jest nigdzie zapamiętywany.

GŁÓWNE ZASTOSOWANIA:

Pomimo, że instrukcja BIT posiada dużą zaletę w postaci nie oddziaływania na testowane liczby, to jest dość rzadko używany, ponieważ nie posiada adresowania natychmiastowego. W tym przypadku można użyć innych instrukcji testujących, np. CMP lub AND.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Strony zerowej	BIT 15	\$24 / 36	2
Bezpośredni	BIT 1500	\$2C / 44	3

BIT oddziałuje na wskaźniki: N, Z, V

BMI

Gdy wskaźnik znaku N jest ustawiony - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

PRZYKŁAD: LDA #128 lub LDA #\$FF

W obydwu działaniach ustawiony zostaje wskaźnik N , co oznacza, że obecna liczba jest ujemna (jeśli używasz arytmetyki znak modulo), lub znak z klawiatury wprowadzony został z klawiszem SHIFT bądź jako słowo kluczowe.

GŁÓWNE ZASTOSOWANIA:

Testowanie dla rozpoznania słów kluczowych w Basicu, znaków ASCII (z SHIFT) czy też symboli graficznych, i wreszcie do rozpoznania znaku liczby w arytmetyce znak modulo.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BMI adres	\$30 / 48	2

BMI nie oddziałuje na rejestr wskaźników.

BNE

Gdy wskaźnik zera Z jest skasowany - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu. Oznacza to skok gdy wynik ostatniej operacji nie jest zerem.

PRZYKŁADY:

LDA #150	lub	LDA #22
CMP #249		SBC #85
BNE 20		BNE 148

W obydwu sekwencjach skasowany zostaje wskaźnik Z i wykonany jest skok. W przypadku lewej sekwencji skok jest wykonywany wprzód o 20 bajtów, w prawej w tył o 20 bajtów (w liczbie 148, która jest większa od 127, ustawiony jest bit 7, co oznacza jednocześnie ustawienie wskaźnika N, a więc liczba jest ujemna).

GŁÓWNE ZASTOSOWANIA:

Jest odwrotnością instrukcji BEQ. BNE oznacza skok, gdy porównywane są niejednakowe bajty. Służy do testowania wyników instrukcji CMP, ADC i innych wpływających na wskaźnik zera. Również struktury typu IF - THEN lub ON - GOTO mogą posługiwać się instrukcją BNE. Działanie tej instrukcji jest podobne do instrukcji " < > " w Basicu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BNE adres	\$D0 / 208	2

BEQ nie oddziałuje na rejestr wskaźników.

BPL

Gdy wskaźnik znaku N jest skasowany - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

PRZYKŁAD: LDA #12 lub LDA #\$3F

W obydwu działaniach skasowany zostaje wskaźnik N , co oznacza, że obecna liczba jest dodatnia (jeśli używasz arytmetyki znak modulo), lub znak z klawiatury wprowadzony został bez klawisza SHIFT (mała litera).

GŁÓWNE ZASTOSOWANIA:

Jest odwrotnością instrukcji BMI. Służy do testowania wyników instrukcji LDA, ADC i innych wpływających na wskaźnik znaku. Również struktury typu IF - THEN lub ON - GOTO mogą posługiwać się instrukcją BPL.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BPL adres	\$10 / 16	2

BPL nie oddziałuje na rejestr wskaźników

BRK

Powoduje przerwanie działania programu, które nie może być zamaskowane (zablokowane) przez ustawienie maski przerwania I w rejestrze stanu.

Jeśli w pamięci komputera znajduje się wektor przerwania, to wskazuje on adres programu obsługi przerwania (gdy taki program jest w pamięci).

Licznik programy zawiera adres BRK+2.

GŁÓWNE ZASTOSOWANIA:

Podczas uruchamiania programu w ML. W kodzie programu umieszczasz instrukcje BRK, które pozwalają uchronić komputer od zawieszenia w trakcie wykonywania procedur w miejscach niepewnych. Gdy w czasie wykonywania programu procesor napotka BRK - przerwie działanie i uruchomi monitor, w którym możesz sprawdzić zawartość rejestrów, tablic lub wartość zmiennych i czy są zgodne z oczekiwaniami, po czym uruchamiasz program od punktu przerwania.

Instrukcja BRK ma identyczne działanie jak instrukcja STOP w Basicu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	BRK	\$00 / 0	1

BRK ustawia wskaźnik B.

BVC

Gdy wskaźnik przepelnienia V jest skasowany - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

GŁÓWNE ZASTOSOWANIA:

W praktyce mało programistów używa arytmetyki znak modulo, gdzie siódmy bit jest przeznaczony do wskazywania znaku liczby. Wskaźnik V ma za zadanie powiadomić Cię, że wynik dodawania dwóch liczb przekroczył wartość 127, co spowodowało, że bit siódmy został ustawiony mimo, że wynik jest nadal dodatni ale nastąpiło przepelnienie.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BVC adres	\$50 / 80	2

BVC nie oddziałuje na rejestr wskaźników.

BVS

Gdy wskaźnik przepelnienia V jest ustawiony - skacze do 127 bajtów w przód lub do 128 bajtów w tył od aktualnego adresu.

GŁÓWNE ZASTOSOWANIA:

Brak. Patrz BVC, którego odwrotnością jest BVS.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Względny	BVS adres	\$70 / 112	2

BVC nie oddziałuje na rejestr wskaźników.

CLC

Kasuje wskaźnik przeniesienia C.

GŁÓWNE ZASTOSOWANIA:

Zawsze używany przed ADC. Jeśli ma być seria dodawań (dodawanie wielobajtowe) , tylko pierwsze dodawanie jest poprzedzone przez CLC, bowiem dalsza informacja z wskaźnika przeniesienia będzie niezbędna w następnych krokach.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	CLC	\$18 / 24	1

CLC kasuje wskaźnik przeniesienia C.

CLD

Kasuje wskaźnik trybu dziesiętnego.

GŁÓWNE ZASTOSOWANIA:

Zaleca się wykonywanie CLD w czasie inicjacji programu aby uniknąć niekontrolowanej pracy w trybie dziesiętnym. Jest to bardzo rzadki błąd, lecz tym bardziej trudny do wykrycia.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	CLD	\$D8 / 216	1

CLD kasuje wskaźnik trybu dziesiętnego

CLI

Kasuje wskaźnik przerwania I.

GŁÓWNE ZASTOSOWANIA:

Przywrócenie normalnego trybu przerwania po jego czasowym zawieszeniu. Dalsze informacje w opisie instrukcji SET

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	CLI	\$58 / 88	1

CLI kasuje wskaźnik maski przerwania I.

CLV

Kasuje wskaźnik przepelnienia V.

GŁÓWNE ZASTOSOWANIA:

Brak. Patrz BVC.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	CLV	\$B8 / 184	1

CLV kasuje wskaźnik przepelnienia V.

CMP

Porównuje bajt w pamięci z bajtem w akumulatorze. Jest w rzeczywistości odejmowaniem bajtu w pamięci od bajtu w akumulatorze.

GŁÓWNE ZASTOSOWANIA:

Jest to centralna instrukcja w strukturach typu IF - THEN czy ON - GOTO. W powiązaniu z instrukcjami skoków względnych (np. BEQ) pozwala na podjęcie decyzji o skierowaniu na alternatywne ścieżki w zależności od wyniku porównania.

CMP ustawia lub kasuje wskaźniki Z, N, i C. Wtedy może zostać wykonany skok zależny od stanu tych wskaźników.

PRZYKŁAD 1

LDA #15

Skasuje wskaźnik N (siódmy bit skasowany) i skasuje wskaźnik Z (wynik nierówny zeru) , natomiast nie oddziałuje na wskaźnik C. Więc wykonanie

LDA #15
BPL cel

spowoduje wykonanie skoku. Tak więc CMP nie zawsze jest niezbędne, ale wstawienie go nie spowoduje nigdy błędu.

Inną ważną decyzją przy skokach są sytuacje " > " i " < ". W tym wypadku używa się instrukcji BCS i BCC aby testować wskaźnik przeniesienia C. Tu trzeba pamiętać o kolejności porównywanych liczb - bajt w pamięci jest porównywany z bajtem w akumulatorze. Niech wartość bajtu akumulatora wynosi A , zaś w pamięci P:

Gdy $P \leq A$ - w wyniku CMP skasowany zostanie C i BCC wykona skok
Gdy $P > A$ - w wyniku CMP zostanie ustawiony C i BCS wykona skok

PRZYKŁAD 2

Chcesz się dowiedzieć, czy liczba w akumulatorze jest mniejsza od \$40 to wystarczy napisać:

CMP #\$41
BCC adres

Wskaźnik przeniesienia C jest zerowany, gdy bajt pamięci jest mniejszy lub równy bajtowi w akumulatorze, dlatego szukając liczby mniejszej od \$40 wykonaliśmy porównanie z \$41.

Innym wskaźnikiem na który oddziałuje CMP jest wskaźnik znaku N. Jego użycie jest ograniczone, bo N informuje tylko o stanie siódmego bitu. BPL wykonuje skok gdy $N=1$, BMI gdy $N=0$. Jednakże bit siódmy informuje o tym, czy liczba jest mniejsza od 128 i można posłużyć się tą informacją do rozpoznania znaków ASCII, słów kluczowych Basicu lub przeszukiwania baz danych.

Ponieważ LDA i wiele innych instrukcji oddziałuje na wskaźnik N - można w wielu wypadkach stosować BPL i BMI bez konieczności uprzedniego wykonania CMP.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	CMP #15	\$C9 / 201	2
Strony zerowej	CMP 15	\$C5 / 197	2
Strony zerowej względem X	CMP 15,X	\$D5 / 213	2
Bezpośredni	CMP 1500	\$CD / 205	3
Bezpośredni względem X	CMP 1500,X	\$DD / 221	3
Bezpośredni względem Y	CMP 1500,Y	\$D9 / 217	3
Pośredni strony zerowej	CMP (15)	\$D2 / 210	2
Pośredni względem X	CMP (15,X)	\$C1 / 193	2
Pośredni względem Y	CMP (15),Y	\$D1 / 209	2

CMP oddziałuje na wskaźniki: N, Z, C .

CPX

Porównuje bajt w pamięci z bajtem w rejestrze X. Jest odpowiednikiem CMP dla rejestru X. Jest w rzeczywistości odejmowaniem bajtu w pamięci od bajtu w rejestrze X i działanie jego różni się od CMP umiejscowieniem bajtu.

GŁÓWNE ZASTOSOWANIA:

Wskaźniki, warunki i cele wykonania instrukcji CPX są takie jak w CMP.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	CPX #15	\$E0 / 224	2
Strony zerowej	CPX 15	\$E4 / 228	2
Bezpośredni	CPX 1500	\$EC / 236	3

CPX oddziałuje na wskaźniki: N, Z, C.

CPY

Porównuje bajt w pamięci z bajtem w rejestrze Y. Jest odpowiednikiem CMP dla rejestru Y. Jest w rzeczywistości odejmowaniem bajtu w pamięci od bajtu w rejestrze Y i działanie jego różni się od CMP umiejscowieniem bajtu.

GŁÓWNE ZASTOSOWANIA:

Wskaźniki, warunki i cele wykonania instrukcji CPY są takie jak w CMP.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	CPY #15	\$C0 / 192	2
Strony zerowej	CPY 15	\$C4 / 196	2
Bezpośredni	CPY 1500	\$CC / 204	3

CPY oddziałuje na wskaźniki: N, Z, C.

DEC

Zmniejsza o 1 wartość bajtu w pamięci.

GŁÓWNE ZASTOSOWANIA:

Bardzo użyteczna alternatywa dla SBC, prostsza i krótsza. Mimo, że nie oddziałuje na wskaźnik przeniesienia C - DEC może zmniejszać liczby dwubajtowe.

Innym zastosowaniem dla instrukcji DEC jest sterowanie wskaźnikiem pamięci, gdy zajęte są rejestry X i Y.

PRZYKŁAD

Możesz zdefiniować np. komórkę o adresie \$600 jako licznik dla pętli programowej:

```
LOOP STA $8000
      DEC $600
      BEQ END
      JMP LOOP
```

Ta struktura będzie zapamiętywała akumulator w komórce \$8000, aż zawartość komórki \$600 zostanie zmniejszona do zera. Imituje to instrukcje DEX i DEY i pozwala ustawić wiele pętli zagnieżdżonych jedna w drugiej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Strony zerowej	DEC 15	\$C6 / 198	2
Strony zerowej względem X	DEC 15,X	\$DE / 214	2
Bezpośredni	DEC 1500	\$CE / 206	3
Bezpośredni względem X	DEC 1500,X	\$DE / 222	3

DEC oddziałuje na wskaźniki: N, Z

DEX

Zmniejsza o 1 zawartość rejestru X.

GŁÓWNE ZASTOSOWANIA:

Używany jako licznik wewnątrz pętli programowej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	DEX	\$CA / 202	1

DEX oddziałuje na wskaźniki N, Z .

DEY

Zmniejsza o 1 zawartość rejestru Y.

GŁÓWNE ZASTOSOWANIA:

Używany jako licznik wewnątrz pętli programowej. Przez zastosowanie adresowania względem Y jest częściej stosowany od DEX

PRZYKŁAD

Powszechnie stosowany sposób wyprowadzania znaków na ekran. Znaki są umieszczone w buforze zaczynającym się od adresu \$5000 i tworzą wiadomość zakończoną liczbą 0.

```
LDY #0
LOOP LDA 5000,Y
      BEQ END
      STA EKRAM
      INY
      JMP LOOP
END
```

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	DEY	\$88 / 136	1

DEY oddziałuje na wskaźniki N, Z .

EOR

Wykonuje operacje "EXOR" na bajcie z pamięci i w akumulatorze. Każdy bit bajtu z pamięci jest porównywany z odpowiadającym mu bitem bajtu w akumulatorze i jeśli obydwa bity są różne to na tym miejscu w akumulatorze pojawi się 1, w przeciwnym wypadku 0. Instrukcja EOR oddziałuje tylko na bity w akumulatorze.

GŁÓWNE ZASTOSOWANIA:

EOR nie ma zbyt wielu zastosowań. Jego główną wartością jest możliwość selektywnego negowania bitów w akumulatorze.

PRZYKŁAD

Jeśli chcesz zmienić wartość szóstego bitu w jakimś bajcie to wykonujesz:

```
LDA BYTE  
EOR #$40  
STA BYTE
```

Sekwencja ta ustawi bit szósty gdy był zerem i skasuje gdy był jedynką.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	EOR #15	\$49 / 73	2
Strony zerowej	EOR 15	\$45 / 69	2
Strony zerowej względem X	EOR 15,X	\$55 / 85	2
Bezpośredni	EOR 1500	\$4D / 77	3
Bezpośredni względem X	EOR 1500,X	\$5D / 93	3
Bezpośredni względem Y	EOR 1500,Y	\$59 / 89	3
Pośredni strony zerowej	EOR (15)	\$52 / 82	2
Pośredni względem X	EOR (15,X)	\$41 / 65	2
Pośredni względem Y	EOR (15,Y)	\$51 / 81	2

EOR oddziałuje na wskaźniki: N, Z .

INC

Zwiększa o 1 wartość bajtu w pamięci.

GŁÓWNE ZASTOSOWANIA:

INC używane jest dokładnie tak, jak DEC, z tym, że INC odlicza w górę.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Strony zerowej	NC 15	\$E6 / 230	2
Strony zerowej względem X	INC 15,X	\$F6 / 246	2
Bezpośredni	INC 1500	\$EE / 238	3
Bezpośredni względem X	INC 1500,X	\$FE / 254	3

INC oddziałuje na wskaźniki: N, Z

INX

Zwiększa o 1 zawartość rejestru X.

GŁÓWNE ZASTOSOWANIA:

Używany jako licznik wewnątrz pętli programowej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	INX	\$E8 / 232	1

DEX oddziałuje na wskaźniki N, Z .

INY

Zwiększa o 1 zawartość rejestru Y.

GŁÓWNE ZASTOSOWANIA:

Używany jako licznik wewnątrz pętli programowej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	INY	\$C8 / 200	1

DEY oddziałuje na wskaźniki N, Z .

JMP

Wykonuje skok do dowolnej lokacji w pamięci.

GŁÓWNE ZASTOSOWANIA:

Skoki o dalekim zasięgu. Jest równoważna instrukcji GOTO w Basicu. Argument instrukcji przepisywany jest do licznika programu i od tego adresu jest kontynuowany program.

Nie jest zalecane wykonywanie skoków pośrednich z uwagi na błędy występujące w przypadku zlokalizowania wskaźnika na granicy strony.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Bezpośredni	JMP 1500	\$4C / 76	3
Pośredni	JMP (1500)	\$6C / 108	3

JMP nie oddziałuje na rejestr wskaźników .

JSR

Skok do podprogramu w dowolnym miejscu pamięci. Licznik programu jest zwiększany o trzy i zapamiętany na stosie. Najbliższa instrukcja RTS ściągnie ten adres ze stosu i program będzie kontynuowany od następnej po JSR instrukcji .

GŁÓWNE ZASTOSOWANIA:

Jest to dokładny odpowiednik instrukcji GOSUB w Basicu. Jest bardzo często używaną instrukcją pozwalającą na przekazanie sterowania przez np. Basic.

Ponieważ JSR przed przejściem do podprogramu wrzuca prawidłowy adres na stos, to jeśli zdecydujesz nie wracać do punktu wywołania - musisz zdjąć dwa bajty ze stosu przez dwukrotne użycie instrukcji PLA a następnie JMP. Wyjście "bokiem " z podprogramu bez sekwencji PLA PLA może spowodować łatwo przepełnienie stosu i zawieszenie komputera.

Im bardziej rozbudowany jest program tym więcej jest w nim podprogramów, a więc też skoków typu JSR.

TRYBY ADRESOWANIA

Tryb adresowania	Format nemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Bezpośredni	JSR 1500	\$20 / 32	3

JSR nie oddziałuje na rejestr wskaźników .

LDA

Kopiuje bajt z pamięci do akumulatora w celu dalszej obróbki.

GŁÓWNE ZASTOSOWANIA:

Akumulator to najbardziej zapracowane miejsce w komputerze. Tędy przechodzą wszystkie bajty z i do urządzeń zewnętrznych. Ponieważ akumulator różni się od rejestrów X i Y jest on używany inaczej niż pozostałe rejestry. Akumulator nie może być użyty jako wskaźnik (gdyż nie istnieje instrukcja INA) - wykorzystywany jest więc jako główny rejestr tymczasowy zapamiętujący bajty podczas obrabiania ich w programie.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychniastowy	LDA #15	\$A9 / 169	2
Strony zerowej	DA 15	\$A5 / 165	2
Strony zerowej względem X	LDA 15,X	\$B5 / 181	2
Bezpośredni	LDA 1500	\$AD / 173	3
Bezpośredni względem X	LDA 1500,X	\$BD / 189	3
Bezpośredni względem Y	LDA 1500,Y	\$B9 / 185	3
Pośredni	LDA (15)	\$B2 / 178	2
Pośredni względem X	LDA (15,X)	\$A1 / 161	2
Pośredni względem Y	LDA (15),Y	\$B1 / 177	2

LDA oddziałuje na wskaźniki: N, Z .

LDX

Kopiuje bajt z pamięci do rejestru X.

GŁÓWNE ZASTOSOWANIA:

Wskaźnik pętli programowych.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	LDX #15	\$A2 / 162	2
Strony zerowej	DX 15	\$A6 / 166	2
Strony zerowej względem Y	LDX 15,Y	\$B6 / 182	2
Bezpośredni	LDX 1500	\$AE / 174	3
Bezpośredni względem Y	LDX 1500,Y	\$BE / 190	3

LDX oddziałuje na wskaźniki: N, Z .

LDY

Kopiuje bajt z pamięci do rejestru Y.

GŁÓWNE ZASTOSOWANIA:

Wskaźnik pętli programowych.

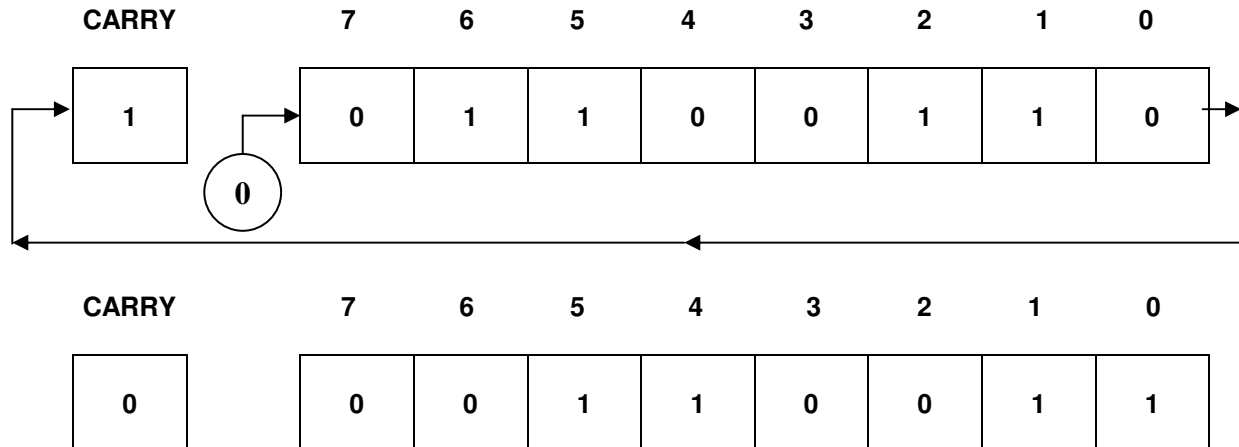
TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Natychmiastowy	LDY #15	\$A0 / 160	2
Strony zerowej	DY 15	\$A4 / 164	2
Strony zerowej względem Y	LDY 15,Y	\$B4 / 180	2
Bezpośredni	LDY 1500	\$AC / 172	3
Bezpośredni względem Y	LDY 1500,Y	\$BC / 188	3

LDY oddziałuje na wskaźniki: N, Z .

LSR

Przesuwa bity w bajcie o jeden w prawo. Bajt może być umieszczony w akumulatorze albo w pamięci, zależnie od trybu adresowania. LSR przesuwają bit 0 do wskaźnika przeniesienia, a na pozycję bitu 7 wstawia wartość " 0 ".



GŁÓWNE ZASTOSOWANIA

LSR pozwala na podzielenie liczby przez 2. Liczby większe od \$FF (255) można przetwarzać używając LSR łącznie z instrukcją ROR.

Inne zastosowanie: przesuwanie czterech starszych bitów w bajcie (jednostka czterobitowa to półbajt lub nibel) na pozycje czterech młodszych bitów. Starsze bity zastąpione są przez zera. Przesunięcie takie uzyskamy przez czterokrotne użycie LSR.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Akumulatora	LSR	\$4A / 74	1
Strony zerowej	LSR 15	\$46 / 70	2
Strony zerowej względem X	LSR 15,X	\$56 / 86	2
Bezpośredni	LSR 1500	\$4E / 78	3
Bezpośredni względem X	LSR 1500,X	\$5E / 94	3

LSR oddziałuje na wskaźniki: N, Z, C

NOP

Nic nie robi oprócz zwiększenia licznika programu o 1.

GŁÓWNE ZASTOSOWANIA:

W czasie uruchamiania pozwala na skasowanie pułapek BRK poprzez zastąpienie ich przez NOP. Trzy kolejne NOP mogą zamaskować JSR i adres - można wtedy obserwować, jak działa program bez tego podprogramu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	NOP	\$EA / 234	1

NOP nie oddziałuje na rejestr wskaźników .

ORA

Wykonuje sumę logiczną bajtu w pamięci z bajtem w akumulatorze. Wynik pozostaje w akumulatorze. Wszystkie bity w obu bajtach są porównywane i jeśli obydwa bity o tej samej wadze są " 0 " to odpowiedni bit wyniku będzie równy " 0 ". W pozostałych przypadkach w wyniku będzie " 1 ".

GŁÓWNE ZASTOSOWANIA

W większości przypadków ORA jest używane do selektywnego ustawiania bitów. Przypuśćmy, że chcesz ustawić bit 0 i 5 w liczbie \$5E (94). W zapisie binarnym liczbę tę przedstawiamy jako 01011110. Należy stworzyć "maskę" tak, by bity 0 i 5 były zgaszone, a pozostałe zostawiamy bez zmian. Mamy więc następującą sytuację:

Akumulator 0 1 0 1 1 1 1 0 (\$5E)

Maska 0 1 1 1 1 1 1 1 (\$80)

Wykonanie ORA spowoduje ustawienie tylko bitów 0 i 5 , które w akumulatorze są zgaszone a w masce zapalone. Bit 7 nie zmieni stanu bo w akumulatorze i masce wartość bitu jest zero.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX /DEC	Ilość bajtów
Natychmiastowy	ORA #15	\$09 / 9	2
Strony zerowej	ORA 15	\$05 / 5	2
Strony zerowej względem X	ORA 15,X	\$15 / 21	2
Bezpośredni	ORA 1500	\$0D / 13	3
Bezpośredni względem X	ORA 1500,X	\$1D / 29	3
Bezpośredni względem Y	ORA 1500,Y	\$19 / 25	3
Pośredni strony zerowej	ORA (15)	\$12 / 18	2
Pośredni względem X	ORA (15,X)	\$01 / 1	2
Pośredni względem Y	ORA (15,Y)	\$11 / 17	2

ORA oddziałuje na wskaźniki: N, Z

PHA

Zawartość akumulatora zapisuje na stosie.

GŁÓWNE ZASTOSOWANIA:

Do czasowego (bardzo krótkiego) przechowania bajtu z akumulatora. Jeśli jesteś w jakimś podprogramie i chcesz zachować bajt na krótki czas, możesz wrzucić go na stos przez PHA. Pamiętaj jednak, aby wycofać go do akumulatora za pomocą instrukcji PLA jeszcze przed wykonaniem RTS, bo inaczej komputer pobierze błędny adres powrotu. Wszystkie adresy powrotu są przechowywane na stosie.

Bezpieczniej jest przechować wartość w jakiejś komórce pamięci instrukcją STA .

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	PHA	\$48 / 72	1

PHA nie oddziałuje na rejestr wskaźników.

PHP

Kopiuje zawartość rejestru wskaźników na stos.

GŁÓWNE ZASTOSOWANIA:

Do czasowego (bardzo krótkiego) przechowania zawartości rejestru wskaźników. Jeśli jesteś w jakimś podprogramie i chcesz zabezpieczyć wszystkie warunki na krótki czas, możesz wrzucić zawartość rejestru wskaźników na stos przez PHP. Pamiętaj jednak, aby wycofać go do rejestru za pomocą instrukcji PLP jeszcze przed wykonaniem RTS, bo inaczej komputer pobierze błędny adres powrotu. Wszystkie adresy powrotu są przechowywane na stosie.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	PHP	\$08 / 8	1

PHP nie oddziałuje na rejestr wskaźników.

PLA

Przenosi bajt ze szczytu stosu do akumulatora.

GŁÓWNE ZASTOSOWANIA:

Przywracanie stanu sprzed momentu wykonania PHA tak, by na szczycie stosu były właściwe adresy powrotów z podprogramów, co jest podstawowym celem stosu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	PLA	\$68 / 104	1

PLA oddziałuje na wskaźniki N, Z.

PLP

Przenosi bajt ze szczytu stosu do rejestru wskaźników.

GŁÓWNE ZASTOSOWANIA:

Przywracanie stanu sprzed momentu wykonania PHP tak, by na szczycie stosu były właściwe adresy powrotów z podprogramów, co jest podstawowym celem stosu.

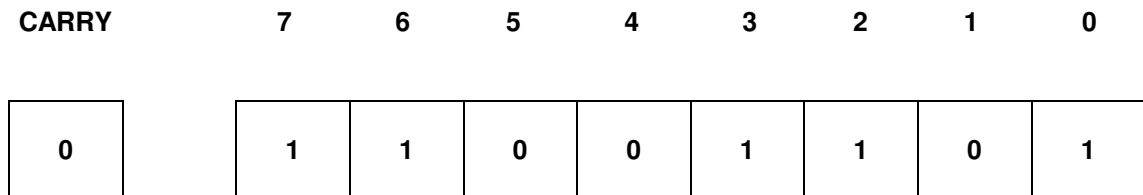
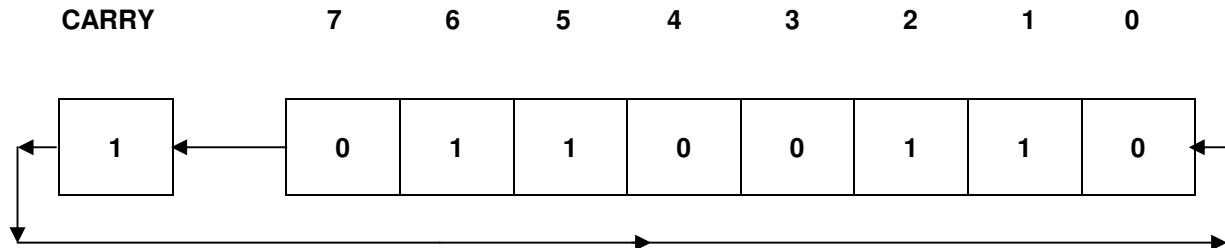
TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	PLP	\$28 / 40	1

PLP oddziałuje na wszystkie wskaźniki.

ROL

Obraca w lewo bity w akumulatorze lub bajcie pamięci. Do bitu 0 przeniesiona zostaje zawartość wskaźnika przeniesienia C, bit 7 na miejsce C a pozostałe bity o jedno miejsce w lewo.



GŁÓWNE ZASTOSOWANIA:

Do mnożenia bajtu przez dwa. ROL można użyć wraz z ASL do mnożenia liczb wielobajtowych, ponieważ ROL przesunął wskaźnik C do bitu zerowego. Jeśli ASL ustawiłoby wskaźnik C, to zostanie on przeniesiony do bitu zerowego starszego bajtu liczby.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Akumulator	ROL	\$2A / 42	1
Strona zerowa	ROL 15	\$26 / 38	2
Strona zerowa względem X	ROL 15,X	\$36 / 54	2
Bezpośrednie	ROL 1500	\$2E / 46	3
Bezpośrednie względem X	ROL 1500,X	\$3E / 62	3

ROL oddziałuje na wskaźniki: N, Z, C .

ROR

Obraca w prawo bity w akumulatorze lub bajcie pamięci. Do bitu 7 przeniesiona zostaje zawartość wskaźnika przeniesienia C, bit 0 na miejsce C a pozostałe bity o jedno miejsce w prawo.



GŁÓWNE ZASTOSOWANIA:

Do dzielenia bajtu przez dwa. ROR można użyć wraz z LSR do mnożenia liczb wielobajtowych, ponieważ ROR przesuwaa wskaźnik C do bitu siódmego. Jeśli LSR ustawiłoby wskaźnik C, to zostanie on przeniesiony do bitu zerowego starszego bajtu liczby.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Akumulator	ROR	\$6A / 106	1
Strona zerowa	ROR 15	\$66 / 102	2
Strona zerowa względem X	ROR 15,X	\$76 / 113	2
Bezpośrednie	ROR 1500	\$CE / 110	3
Bezpośrednie względem X	ROR 1500,X	\$7E / 126	3

ROR oddziałuje na wskaźniki: N, Z, C .

RTI

Powrót z przerwania.

GŁÓWNE ZASTOSOWANIA:

Brak. Możesz chcieć własne procedury do procedur przerwania w komputerze (patrz opis SEI), ale sam nie wytworzysz rzeczywistego przerwania. W konsekwencji nie możesz wykonać powrotu z przerwania, którego sam nie stworzyłeś.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	RTI	\$40 / 64	1

RTI oddziałuje na wszystkie wskaźniki, są one bowiem odzyskiwane ze stosu.

RTS

Powrót z podprogramu wywołanego przez instrukcję JSR .

GŁÓWNE ZASTOSOWANIA:

Automatycznie ściąga ze szczytu stosu dwa bajty i umieszcza je w liczniku programu PC. To przywraca główny ciąg programu przerwany przez JSR (która przed przejściem do wykonania podprogramu umieściła zawartość licznika programu na stosie).

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	RTS	\$60 / 96	1

RTS nie oddziałuje na rejestr wskaźników.

SBC

Odejmuje bajt w pamięci od bajtu w akumulatorze; w razie konieczności wykonuje "pożyczkę". Jeśli miała miejsce pożyczka - wskaźnik przeniesienia jest zerowany, a więc zawsze przed SBC trzeba wykonać SEC, czyli ustawić wskaźnik C.

GŁÓWNE ZASTOSOWANIA:

Odejmowanie liczb.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Natychmiastowe	SBC #15	\$E9 / 233	2
Strona zerowa	SBC 15	\$E5 / 229	2
Strona zerowa względem X	SBC 15,X	\$F5 / 245	2
Bezpośrednie	SBC 1500	\$ED / 237	3
Bezpośrednie względem X	SBC 1500,X	\$FD / 253	3
Bezpośrednie względem Y	SBC 1500,Y	\$F9 / 249	3
Pośrednie strony zerowej	SBC (15)	\$F2 / 242	2
Pośrednie względem X	SBC (15,X)	\$E1 / 225	2
Pośrednie względem Y	SBC (15),Y	\$F1 / 241	2

SBC oddziałuje na wskaźniki: N, Z, C, V

SEC

Ustawia wskaźnik przeniesienia C.

GŁÓWNE ZASTOSOWANIA:

Ta instrukcja jest zawsze wykonywana przed SBC

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	SEC	\$38 / 56	1

SEC oddziałuje na wskaźnik C .

SED

Ustawia wskaźnik trybu dziesiętnego.

GŁÓWNE ZASTOSOWANIA:

Ta instrukcja wprowadza procesor 6502 w tryb arytmetyki dziesiętnej

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	SED	\$F8 / 248	1

SED oddziałuje na wskaźnik trybu dziesiętnego D .

SEI

Ustawia wskaźnik zakazu przerwania I w rejestrze wskaźników. Gdy ten wskaźnik jest ustawiony - procesor nie potwierdzi ani nie obsłuży żadnego przerwania (za wyjątkiem kilku przerwania niemaskowalnych, np. RESET)

GŁÓWNE ZASTOSOWANIA:

Blokowanie przerwania dla wykonania krytycznych procedur. Przykładem może być szybki transfer dużej ilości danych między obszarami pamięci lub urządzeń peryferyjnych. Zawsze tam, gdzie procedura ma krytyczne wartości czasowe (timingi) można zastosować SEI.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	SEI	\$78 / 120	1

SEI oddziałuje na wskaźnik blokady przerwania I .

STA

Kopiuje bajt z akumulatora do pamięci.

GŁÓWNE ZASTOSOWANIA:

Służy do wielu celów i jest jedną z najczęściej używanych instrukcji, głównie jednak do komunikacji z pamięcią za pośrednictwem akumulatora.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Strona zerowa	STA 15	\$85 / 133	2
Strona zerowa względem X	STA 15,X	\$95 / 149	2
Bezpośrednie	STA 1500	\$8D / 141	3
Bezpośrednie względem X	STA 1500,X	\$9D / 157	3
Bezpośrednie względem Y	STA 1500,Y	\$99 / 153	3
Pośrednie strony zerowej	STA (15)	\$92 /146	2
Pośrednie względem X	STA (15,X)	\$81 / 129	2
Pośrednie względem Y	STA (15,Y)	\$91 / 145	2

STA nie oddziałuje na rejestr wskaźników.

STX

Kopiuje bajt z rejestru X do pamięci.

GŁÓWNE ZASTOSOWANIA:

Służy do wielu celów i jest jedną z najczęściej używanych instrukcji, głównie jednak do komunikacji z pamięcią za pośrednictwem rejestru X.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Strona zerowa	STX 15	\$86 / 134	2
Strona zerowa względem Y	STX 15,Y	\$96 / 150	2
Bezpośrednie	STX 1500	\$8E / 142	3

STX nie oddziałuje na rejestr wskaźników.

STY

Kopiuje bajt z rejestru Y do pamięci.

GŁÓWNE ZASTOSOWANIA:

Służy do wielu celów i jest jedną z najczęściej używanych instrukcji, głównie jednak do komunikacji z pamięcią za pośrednictwem rejestru Y.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Strona zerowa	STY 15	\$84 / 132	2
Strona zerowa względem X	STY 15,X	\$94 / 148	2
Bezpośrednie	STY 1500	\$8C / 140	3

STY nie oddziałuje na rejestr wskaźników.

TAX

Przenosi bajt z akumulatora do rejestru X.

GŁÓWNE ZASTOSOWANIA:

Rzadko używany z uwagi na to, że rejestr X najczęściej pracuje jako licznik pętli programowej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TAX	\$AA / 170	1

TAX oddziałuje na wskaźniki N, Z. .

TAY

Przenosi bajt z akumulatora do rejestru Y.

GŁÓWNE ZASTOSOWANIA:

Rzadko używany z uwagi na to, że rejestr Y najczęściej pracuje jako licznik pętli programowej.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TAX	\$A8 / 168	1

TAY oddziałuje na wskaźniki N, Z.

TSX

Przenosi wskaźnik stosu SP do rejestru X.

GŁÓWNE ZASTOSOWANIA:

Wskaźnik stosu (Stack Point) jest bajtem, który wskazuje, w którym miejscu można wrzucić na stos nowy bajt, czyli gdzie jest wierzchołek stosu. Ponieważ stos to 256 bajtów pomiędzy \$100 a \$ 1FF. Jeśli do wartości SP dodamy \$100 otrzymamy adres wierzchołka stosu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TSX	\$BA / 186	1

TSX oddziałuje na wskaźniki N, Z. .

TXA

Przenosi bajt z rejestru X do akumulatora.

GŁÓWNE ZASTOSOWANIA:

Rzadko używany z uwagi na to, że rejestr X najczęściej pracuje jako licznik pętli programowej. Czasem można użyć wyniku licznika do jakichś obliczeń.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TXA	\$8A / 138	1

TXA oddziałuje na wskaźniki N, Z. .

TXS

Przenosi bajt z rejestru X do wskaźnika stosu SP.

GŁÓWNE ZASTOSOWANIA:

Rzadko używany. Pozwala na zmianę adresu wierzchołka stosu.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TXS	\$9A / 154	1

TXS nie oddziałuje na rejestr wskaźników .

TYA

Przenosi bajt z rejestru Y do akumulatora.

GŁÓWNE ZASTOSOWANIA:

Rzadko używany z uwagi na to, że rejestr Y najczęściej pracuje jako licznik pętli programowej. Czasem można użyć wyniku licznika do jakichś obliczeń.

TRYBY ADRESOWANIA

Tryb adresowania	Format mnemoniczny	Kod operacji HEX / DEC	Ilość bajtów
Wewnętrzny	TYA	\$98 / 152	1

TYA oddziałuje na wskaźniki N, Z. .

Procesor 6502 posiada następujące rejestry:

A - akumulator - rejestr ośmiobitowy
X - rejestr X - ośmiobitowy
Y - rejestr Y - ośmiobitowy
PC - licznik programu - szesnastobitowy
SP - wskaźnik stosu - ośmiobitowy
FLAG - rejestr wskaźników - ośmiobitowy

Znaczenie bitów rejestru FLAG:

0 - Carry - bit przeniesienia
1 - Zero - wynik operacji = 0
2 - Interrupt - wystąpiło żądanie przerwania
3 - Decimal - praca w trybie dziesiętnym
4 - Break - przerwa
5 - nie używany
6 - oVerflow - przepełnienie
7 - Negative - wynik operacji jest ujemny