

ROZKAZY W KODZIE

MOV r1, r2- Przesył danych między dwoma wskazanymi rejestrami r2 -> r1

MVI r, dane- Dotyczy przesłania danych zawartych w drugim bajcie instrukcji do wskazanego rejestru

Skok warunkowy o kodzie CA, odnosi się do warunków zakodowanych na bitach CCC zawartych w pierwszym bajcie rozkazu

LXI H - Dotyczy przesyłu danych zawartych w drugim i trzecim bajcie do pary rejestrów wskazanych rozkazem

RST 1/2/3 - Oznacza programowe wywołanie jednej z ośmiu lokacji restartów

Kod:

RST	0	1	2	3	4	5	6	7
kod H	C7	CF	D7	DF	E7	EF	F7	FF

CALL - Jest to rozkaz bezwarunkowy o podobnym działaniu jak skok JMP adr, tylko że w tym przypadku CALL następuje przekazanie do stosu zawartości licznika programu, czego nie wykonuje JMP. Wykonuje te same operacje jak przy żądaniu przerwania.

CPI 0 -

JZ - Skok warunkowy o kodzie CA, odnosi się do warunków zakodowanych na bitach CCC zawartych w pierwszym bajcie rozkazu

JC- Skok warunkowy o kodzie DA, odnosi się do warunków zakodowanych na bitach CCC zawartych w pierwszym bajcie rozkazu

STS - ustawianie flagi przesunięcia jako 1

ADD r - Oznacza dodanie do zawartości akumulatora zawartości wskazanego rejestru (r) gdzie r jest lokalizowany kodem bitów S

DCR r - Oznacza zmniejszenie zawartości wskazanego rejestru r o 1

JZN - Skok warunkowy o kodzie C2, odnosi się do warunków zakodowanych na bitach CCC zawartych w pierwszym bajcie rozkazu

JMP - Oznacza bezwarunkowy skok do komórki pamięci adresowanej drugim i trzecim bajtem rozkazu

INR r - Oznacza zwiększenie zawartości wskazanego rejestru r o 1

ADI 48 - Oznacza dodanie do zawartości akumulatora drugiego bajtu rozkazu

- **SUI** dane - A <- (A) - (bajt 2) Kod rozkazu 06H

HLT - Jest rozkazem zatrzymywania realizacji programu wywołując stan czekania TwH.

Wyjście z tego stanu jest tylko możliwe poprzez przerwanie (lub przez Reset)

SUB r - Oznacza odejmowanie od zawartości akumulatora zawartości wskazanego rejestru

SBB r - Oznacza odejmowanie od zawartości akumulatora zawartości wskazanego rejestru z uwzględnieniem pożyczki