

## **Rejestry:**

AX	akumulator
BX, CX, DX	
PC	'Program Counter': licznik zawierający adres komórki pamięci z wykonywaną właśnie operacją.

## **Polecenia:**

ADD $x$	dodaj $x$ do wartości w rejestrze AX.
ADD @ $m$	dodaj wartość zapisaną w komórce pamięci $m$ do wartości w rejestrze AX.
ADD $rr$	dodaj wartość zapisaną w rejestrze $rr$ do wartości w rejestrze AX; $rr : \{ AX, BX, CX, DX \}$
ADD @ $rr$	dodaj wartość zapisaną w komórce pamięci $m$ , gdzie $m$ jest zapisane w rejestrze $rr$ , do wartości w rejestrze AX.
AND	koniunkcja; patrz ADD.
CMC ?	zmień wartość flagi CA (jeśli flaga = 0 ustaw 1; jeśli flaga = 1 ustaw 0).
CME ?	zmień wartość flagi EV.
CMO ?	zmień wartość flagi OV.
CMS ?	zmień wartość flagi SI.
CMZ ?	zmień wartość flagi ZE.
DEC $rr$	odejmij 1 od wartości w rejestrze $rr$ .
DIV	podziel; patrz ADD.
HLT	zatrzymaj działanie programu.
IN $b$	wczytaj wartość komórki $b$ portu wyjściowego do rejestru AX.
INC $rr$	dodaj 1 do wartości w rejestrze $rr$ .
JA?	
JBE?	
JC $m$	przeskocz do komórki pamięci $m$ jeżeli CA = 1.
JE $m$	przeskocz do komórki pamięci $m$ jeżeli EV = 1.
JG?	> ?
JGE?	≥ ?
JL?	< ?
JLE?	≤ ?
JMP $m$	przeskocz do komórki pamięci $m$ .
JNC $m$	przeskocz do komórki pamięci $m$ jeżeli CA = 0.
JNE $m$	przeskocz do komórki pamięci $m$ jeżeli EV = 0.
JNO $m$	przeskocz do komórki pamięci $m$ jeżeli nie nastąpiło przepełnienie w rejestrze AX (OV = 0).
JNS $m$	przeskocz do komórki pamięci $m$ jeżeli wartość w rejestrze AX ≥ 0 (SI = 0).
JNZ $m$	przeskocz do komórki pamięci $m$ jeżeli wartość w rejestrze AX ≠ 0 (ZE = 0).
JO $m$	przeskocz do komórki pamięci $m$ jeżeli nastąpiło przepełnienie w rejestrze AX (OV = 1).
JS $m$	przeskocz do komórki pamięci $m$ jeżeli wartość w rejestrze AX < 0 (SI = 1).
JZ $m$	przeskocz do komórki pamięci $m$ jeżeli wartość w rejestrze AX = 0 (ZE = 1).
LOAD $x$	załaduj $x$ do rejestru AX.
LOAD @ $m$	załaduj wartość zapisaną w komórce pamięci $m$ do rejestru AX.
MOD	moduło; patrz ADD.
MUL	pomnóż; patrz ADD.
NEG	zmiana znaku wartości w rejestrze AX.
NOP	polecenie, które nic nie robi.
NOT	negacja wartości w rejestrze AX.
OR	alternatywa; patrz ADD.
OUT $b$	wypisz wartość rejestru AX do komórki $b$ portu wyjściowego.
STORE $rr$	zapisz wartość z rejestru AX do rejestru $rr$ .
STORE @ $m$	zapisz wartość z rejestru AX do komórki pamięci $m$ .
SUB	odejmij; patrz ADD.
XOR	alternatywa wykluczająca; patrz ADD.

## **Flagi:**

CA?	(carry?)
SI	'SIGN': 1 gdy AX < 0
ZE	'ZERO': 1 gdy AX = 0
OV	'OVERFLOW':
EV?	(parity?)