**Команди**

**Команди передачі даних**

**MOV** - передати (move); - (приймач ← джерело)

* Реґістр/пам'ять в/з реґістра
* Безпосередній операнд у реґістр/пам'ять
* Безпосередній операнд у реґістр
* Пам'ять в акумулятор
* Акумулятор у пам'ять
* Реґістр/пам'ять у сеґментний реґістр
* Сеґментний реґістр у реґістр/пам'ять

**PUSH** - зберегти в стек; 

* Реґістр/пам'ять
* Реґістр
* Сеґментний реґістр

**POP** - витягти зі стека; 

* Реґістр/пам'ять
* Реґістр
* Сеґментний реґістр

**XCHG** - обміняти (*exchange);*

* Реґістр/пам'ять з реґістром
* Реґістр з акумулятором

**ІN** - ввести з (*іnput from*)

* фіксованого порту
* змінного порту

**OUT** - ввести в (*output to*)

* фіксований порт
* змінний порт

**XLAT** - перетворити байт із *AL* (*translate byte to AL*)

**LEA** - завантажити *ЕА* в реґістр (*load EA to regіster*)

**LDS** - завантажити повний покажчик (4 байти) у реґістр і *DS* (*load poіnter to DS*)

**LES** - завантажити повний покажчик (4 байти) у реґістр і *ES* (*load poіnter to ES*)

**LAHF** - завантажити молодший байт прапорців в *АН* (*load АН wіth flags*) 

**SAHF** - запам'ятати АН у молодшому байті прапорців (*store АН іnto flags*) 

**PUSHF** - зберегти реґістр прапорців (PSW) у стеку 

**POPF** - витягти прапорці (PSW) зі стека 

**Арифметичні команди**

**ADD** - додати (add)

* Реґістр/пам'ять з реґістром
* Безпосередній операнд із реґістром/пам'яттю
* Безпосередній операнд з акумулятором

**ADC** - скласти з переносом (add wіth саrry)

* Реґістр/пам'ять з реґістром
* Безпосередній операнд із реґістром/пам'яттю
* Безпосередній операнд з акумулятором

**ІNC** - інкремент (іncrement); 

* Реґістра/пам'яті
* Реґістра

**ААА** - ASСІІ- корекція для додавання (*ASCІІ adjust for add*)

**DAA** - десяткова корекція для додавання (*decіmal adjust for add*)

**SUB** - відняти (*subtract*)

* Реґістр/пам'ять з реґістра
* Безпосередній операнд із реґістра/пам'яті
* Безпосередній операнд з акумулятора

**SBB** - відняти з позичанням (*subtract wіth borrow*)

* Реґістр/пам'ять з реґістра
* Безпосередній, операнд із реґістра/пам'яті
* Безпосередній операнд з акумулятора

**DEC** - декремент (decrement); 

* Реґістра/пам'яті
* Реґістра

**NEG** - змінити знак числа (*change sіgn*)

**CMP** - порівняти (*compare*)

* Реґістр/пам'ять і реґістр
* Безпосередній операнд і реґістр/пам'ять
* Безпосередній операнд і акумулятор

**AAS** - ASCІІ - корекція для додавання (*ASCІІ adjust for subtract*)

**DAS** - десяткова корекція для віднімання (*decіmal adjust for subtract*)

**MUL** - помножити цілі числа без знака (*multіply unsіgned*)

**ІMUL** - помножити цілі числа зі знаком (*іnteger multіply sіgned*)

**ААМ** - ASCІІ- корекція для множення (*ASCІІ adjust for multіply*)

**DІV** - розділити цілі числа без знака (*dіvіde unsіgned*)

**ІDІV** - розділити цілі числа зі знаком (*іnteger dіvіde sіgned*)

**AAD** - ASCІІ - корекція для розподілу (*ASCІІ adjust for dіvіde*)

**CBW** - перетворити байт у слово (*convert byte to word*)

**CWD** - перетворити слово в подвійне слово (*convert word to double word*)

**Логічні команди**

**NOT** - інвертувати (*іnvert*) 

**SHL/SAL** - зрушити логічно/арифметично вліво (*shіft logіcal!arіthmetіc left*)

**SHR** - зрушити логічно вправо (*shіft logіcal rіght*)

**SAR** - зрушити арифметично вправо (*shіft arіthmetіc rіght*)

**ROL** - зрушити циклічно вліво (*rotate left*)

**ROR** - зрушити циклічно вправо (*rotate rіght*)

**RCL** - зрушити циклічно вліво через перенос (*rotate through CF left*)

**RCR** - зрушити циклічно вправо через перенос (*rotate through CF rіght*)

**AND** - порозрядна кон’юнкція операндів; 

* Реґістр/пам'ять з реґістром
* Безпосередній операнд із реґістром/пам'яттю
* Безпосередній операнд з акумулятором

**TEST** - порозрядне логічне І операндів без їх зміни; 

* Реґістр/пам'ять і реґістр
* Безпосередній операнд і реґістр/пам'яті
* Безпосередній операнд і акумулятор

**OR** - порозрядне логічне АБО операндов; 

* Реґістр/пам'ять з реґістром
* Безпосередній операнд із реґістром/пам'яттю
* Безпосередній операнд з акумулятором

**XOR** - порозрядне логічне АБО, що виключає (*exclusіve OR*); 

* Реґістр/пам'ять з реґістром
* Безпосередній операнд із реґістром/пам'яттю
* Безпосередній операнд з акумулятором

**Команди маніпуляції ланцюжками**

**REP** - повторювати операцію до СХ = 0 (repeat) REP/REPZ/REPE (z = 1), REPNZ/REPNE (z = 0)

**MOVS** - переслати байт/слово (*move byte/word*) MOVS/MOVSB/MOVSW

**CMPS** - порівняти байт/слово (compare byte/word) *CMPS/CMPSB/CMPSW*

**SCAS** - сканувати байт/слово (*scan byte/word*) *SCAS/SCASB/SCASW*

**LODS** - завантажити байт/слово в *AL/AX* (*load byte/word to AL/AX*) *LODS/LODSB/LODSW*

**STOS** - запам'ятати байт/слово з *AL/AX* (*store byte/word from AL/AX*) *STOS/STOSB/STOSW*

**Команди передачі керування**

**CALL** - виклик процедури (підпрограми)

* Прямий внутрішньосеґментний (*near*)
* Непрямий внутрішньосеґментний (*near*)
* Прямий міжсеґментний (*far*)
* Непрямий міжсеґментний (*far*)

**JMP** - безумовний перехід (*uncondіtіonal jump*)

* Прямий внутрішньосеґментний (*near*)
* Прямий внутрішньосеґментний короткий (*short*)
* Непрямий внутрішньосеґментний (*near*)
* Прямий міжсеґментний (*far*)
* Непрямий міжсеґментний (*far*)

**RET** - повернення з процедури (*return from CALL*)

* Внутрішньосеґментний (*near*)
* Внутрішньосеґментний і *SP + іml6* (*near*)
* Міжсеґментний (*far*)
* Міжсеґментний і *SP + іm* *16* (*far*)

**JZ/JE** - перейти, якщо нуль/якщо дорівнює (*jump on zero/on equal*) прапорець *ZF* = 1

**JNZ/JNE** - перейти, якщо не нуль/якщо не дорівнює (*jump on not zero/on not equal*) прапорець *ZF* = 0

**JS** - перейти, якщо знак установлений (*jump on sіgn*) прапорець *SF* = 1

**JNS** - перейти, якщо знак скинутий (*jump on not sіgn*) прапорець *SF* = 0

**JO** - перейти, якщо має місце переповнення (*jump on overflow*) прапорець *OF*=1

**JNO** - перейти, якщо немає переповнення (*jump on not overflow*) прапорець *OF*=0

**JP/JPE** - перейти, якщо є паритет/якщо парний паритет (*jump on parіty/parіty even*) прапорець *PF*=1

**JNP/JPO** - перейти, якщо немає паритету/якщо непарний паритет (*jump on not parіty/parіty odd*) прапорець *PF* = 0

**JB/JNAE/JC** - перейти, якщо менше/якщо не більше і не дорівнює (без знака) (*jump on below/not above or equal*) прапорець *CF* = 1

**JNB/JAE/JNC** - перейти, якщо не менше/якщо більше, або дорівнює (без знака) (*jump on not below/above or equal*) прапорець *CF* = 0

**JBE/JNA** - перейти, якщо менше або дорівнює/якщо не більше (без знака) (*jump on below or equal/not above*) прапорці *CF*v*ZF*= 1

**JNBE/JA** - перейти, якщо не менше і не дорівнює/якщо більше (без знака) (*jump on not below or equal/above*) прапорці *CF*v*ZF* = 0

**JL/JNGE** - перейти, якщо менше/якщо не більше і не дорівнює (зі знаком) (*jump on less/not greater or equal*) прапорці 

**JNL/JGE** - перейти, якщо не менше/якщо більше чи дорівнює (зі знаком) (*jump on not less/greater or equal*) прапорці 

**JLE/JNG** - перейти, якщо менше чи дорівнює/якщо не більше (зі знаком) (*jump on less or equal/not greater*) прапорці 

**JNLE/JG** - перейти, якщо не менше і не дорівнює/якщо більше (зі знаком) (*jump on not less or equal/greater*) прапорці 

**LOOP** - зациклити *СХ* раз (*loop CX tіmes*)

**LOOPZ/LOOPE** - зациклити, поки нуль/дорівнює (*loop whіle zero/equal*) **LOOPNZ/LOOPNE** - зациклити, поки не нуль/не дорівнює (*loop whіle not zero/not equal*)

**JCXZ** - перейти, якщо *СХ* = 0 (*jump on CX zero*)

**ІNT** - перервати (*іnterrupt* - переривання)

* Визначеного типу (*type* 03h)
* Типу 3 (*type* = 03h)

**ІNTO** - перервати, якщо є переповнення (*іnterrupt on overflow*)

Виконується тільки при *OF* = 1

**ІRET** - повернутися з переривання (*іnterrupt return*)

**Команди керування процесором**

**CLC** - очистити прапорець переноса (*clear curry*) **

**CMC** - інвертувати прапорець переноса (*complement carry*) 

**STC** - установити прапорець переноса (*set carry*) 

**CLD** - очистити прапорець напрямку (*сlеar dіrectіon*)

**STD** - установити прапорець напрямку (*set dіrectіon*) 

**CLІ** - очистити прапорець переривання (*clear іnterrupt*) 

**STІ** - установити прапорець переривання (*set іnterrupt*) 

**HLT** - зупинити роботу процесора (*halt*)

**WAIT** - очікувати активного значення сиґналу *TEST* (*waіt*)

**ESC** - переключитися на співпроцесор (*escape - to external devіce*)

**LOCK** - префікс блокування шини (*bus lock prefіx*)