АРИФМЕТИЧНО-ЛОГІЧНІ ОПЕРАЦІІ

ADD R (A)← (A)+(R) До вмісту акумулятора додати вміст регістра

R →{A,B,C,D,E,H,L,M} при цьому результат занести до акумулятора.

ADD A (A)← (A)+(A)

ADD B (A)← (A)+(B)

ADD M (A) ←(A)+{(HL)} До вмісту акумулятора додати вміст комірки

адреса якої вказана в HL парі.

ADI D8 (A)← (A)+D8 До вмісту акумулятора безпосередньо додати

число.

ADC R (A)← (A)+(R)+(Cy) До вмісту акумулятора додати вміст регістра

та ознаку переноса .

ADC B (A) ←(A)+(B)+(Cy) До вмісту акумулятора додати вміст регістра

В та ознаку переноса.

ADC M (A)← (A)+{(HL)}+(Cy) До вмісту акумулятора додати вміст комірки

адреса якої вказана в HL парі та ознаку переносу

ACI D8 (A) ←(A)+D8+(Cy) До вмісту акумулятора безпосередньо додати

число та ознаку переносу.

SUB R (A)← (A)-(R) Від вмісту акумулятора відняти вміст регістру.

SUB B (A) ←(A)-(B) Від вмісту акумулятора відняти вміст регістру В

SUB M (A) ←(A)-{(HL)} Від вмісту акумулятора відняти вміст комірки

пам’іті адреса якої знаходиться в парі HL.

SUI D8 (A) ←(A) –D8 Від вмісту акумулятора безпосередньо відняти

число.

SBB R (A)← (A)-(R)-(Cy) Від вмісту акумулятора відняти вміст регістру та ознаку переносу.

SBB B (A)← (A)-(B)-(Cy) Від вмісту акумулятора відняти вміст регістру В

та ознаку переносу.

SBB M (A) ←(A)-{(HL)}-(Cy) Від вмісту акумулятора відняти вміст комірки

Памяті адреса якої знаходиться в HL парі.

SBI D8 (A)← (A)-D8-(Cy) Від вмісту акумулятора безпосередньо відняти

число та ознаку переноса.

КОМАНДИ ЗБІЛЬШЕННЯ ЧИ ЗМЕНШЕННЯ ВМІСТУ РЕГІСТРУ ЧИ ПАРИ НА1

Інкриментування та та декрементування

INR (R)← (R)+1 до вмісту регістру додати одиницю.

1. →{A,B,C,D,E,H,L,M}

INR A (A)← (A) +1 до вмісту акумулятора додати одиницю.

INR B (B) ←(B) + 1 до вмісту регістра В додати 1.

INR M {(HL) } ←{ (HL)}+1 до вмісту комірки памяті адреса якої вказана

в HL парі додати 1.

DCR R (R)← (R)-1 від вмісту регістра відняти 1.

(R)→{ A,B,C,D,E,H,L,M}

DCR A (A) ←(A)-1 від вмісту акумулятора відняти 1.

DCR B (B) ←(B)-1 від вмісту регістра В відняти 1.

DCR M {(HL)}← {(HL)}-1 від вмісту комірки памяті адреса якої вказана в HL відняти 1.

INX rp (rp)← (rp)+1 Вміст регістрової пари стає більшим на одиницю

(rp) ←{ BC,DE,HL,SP}

**DCX rp (rp) ←(rp)-1 Вміст регістрової пари стає меншим на 1.**

**DAD (rp) {(HL)}←{(HL)}+(rp) До вмісту HL пари додати вміст регістрової**

**(rp)←{ BC,DE,HL,SP} пари.**

**DAD B {(HL)}← {(HL)}+(BC) До вмісту HL пари додати вміст регістрової**

**пари ВС.**

**DAD SP {(HL)}← {(HL)}+(SP) До вмісту HL пари додати вміст стека.**

**DAA двійково-десятична корекція.**

**ANA R (A)← (A)^(R) Ця команда змінює ознаку переносу на 0.**

**R ←{A,B,C,D,E,H,L,M } Вміст акумулятора разом з вмістом регістра**

**Підлягає логічній операції “і”.**

**ANA M (A)← (A)^{(HL)} Вміст акумулятора разом з вмістом комірки памяті адреса якої вказана в HL парі підлягає**

**Логічній операції “ і“.**

**ANI D8 (A) ←(A)^D8 Вміст акумулятора разом з безпосереднім**

**числом підлягають логічній операції “і”.**

XRA R (A)← (A) ⊕ (R) Вміст акумулятора з вмістом регістра підлягають

R →{A,B,C,D,E,H,L,} логічній операціїї виключаюче “або”.

XRI D8 (A) (A) ⊕ D8 **Вміст акумулятора збезпосереднім числом**

**підлягають логічній операціїї виключаюче**

“або**”**

**ORA R (A)← (A)v(R) Вміст акумулятора з вмістом регістра**

**R→{ A,B,C,D,E,H.L,M} підлягають логічній операції “чи”.**

**ORI D8 (A)← (A)vD8 Вміст акумулятора збезпосереднім числом**

**Підлягають логічній операції “чи”.**

**Команди порівняння**

**CMP R (A)← (A)- (R) Від вмісту акумулятора відняти вміст регістра,**

**(R) →{A,B,C,D,E,H,L,M} при цьому вміст акумулятора не змінюється,**

**однак змінюється стан флага z.**

**Якщо А=r ,то z 1 ;A>r ,тоz=0 ;A<r,тоСy =1.**

**CMP M (A) ←(A) – {(HL)} Від вмісту акумулятора відняти вміст комірки**

**пам’яті адреса якої вказана в HL парі,при цьому**

**вміст акумулятора не змінюється,змінюється**

**тільки стан флага z.**

**CPI D8 (A) ←(A) –D8 порівняти вміст аумулятора з безпосереднім числом.**

**СMA (A)← (A) Вміст акумулятора підлягає інверсії.**

**RLC Вміст акумулятора зсувається на одну**

**позицію в ліво ,молодший біт D0 і Сy**

**приймають значення витисненого біта D7 .**

**RRC Вміст акумулятора зсувається на одну позицію в**

**право .При цьому старший біт D7 та Сy приймають**

**значення витисненого біта D0 .**

**RAL Циклічний зсув в ліво вмісту акумулятора . В молодшому біті D0 відображається вміст Сy , а в Сy - D7 .**

**RAR Циклічний зсув в право вмісту акумулятора .**

**В старшому біті D7 відображається вміст Сy ,**

**а в Сy - D0 .**

**STC У флаг переносу всановлюється 1 .**

**СMC Вміст акумулятора підлягає інверсії .**

**EI Дозволити преривання .**

**DI Заборонити преривання .**

**Команда безумовного переходу**

**JMP addr При її виконанні відбувається перехід обчислень по вказаному адресу незалежно від**

**результатів попередніх обчислень .**

**Команди умовного переходу**

**JZ addr Якщо вміст регістра**

**попередніх обчислень рівний 0 , то управління пере-**

**дається по вказаному адресу, якщо не рівний 0 , то**

**керування передається слідуючій за JZ команді .**

**JNZ addr Якщо вміст регістра попередніх обчислень не рівний**

**0 ,то управління передається по вказаному адресу .**

**JC addr Якщо С рівний 0 ,то управління передається по**

**вказаному адресу .**

**JNC addr Якщо С y не рівний 0 ,то управління передається по**

**JNC addr Якщо С y не рівний 0 ,то управління передається по**

**вказаному адресу .**

**JPO addr Якщо якщо кількість одиниць не парна ,то**

**управління передається по вказаному адресу .**

**JPE addr Якщо якщо кількість одиниць парна ,то**

**управління передається по вказаному адресу .**

**JP addr Перехід по адресу ,якщо у флазі регістра признаку S**

**знаходиться 0 .**

**JM addr****Перехід по адресу ,якщо у флазі регістра признаку S**

**знаходиться 1 .**