FETCH & DECODE & EXECUTE

T0: AR ← PC ; PC DEGERI AR'YE AKTARILIR.

T1: IR ←

M[AR], ; MEMORY'DEN ADRES DEGERINI AL.

PC ← PC + 1 ; PC İÇİNDEKİ DEGER BIR ARTTIRILIR.

T2: D0, ..., D7 ←

DECODE IR(12-14), ; OPCODE DECODE EDILIR

AR ← IR(0-11), ; IR ADDRESS DEGERI AR'YE AKTARILIR

I ← IR(15) ; MSB BİTİ I'YA AKTARILIR.

INSTRUCTION REGISTER [I (15) XXX (12-14) YYYYYYYYYYYY (0-11)]

I: DOLAYLI VEYA DOĞRUDAN ADRESIN BELIRTILDIGI BIT

X: OPCODE

Y: ADDRESS

DETERMINE THE TYPE OF INSTRUCTION

[I (15) XXX (12-14) YYYYYYYYYYY (0-11)]

XXX (OPCODE) DECODE EDILIR:

OPCODE 3 GİRİŞE KARŞILIK, 2^3 = 8 CIKIS VERIR

OPCODE

000 - 110 ARASINDA ISE, MEMORY REFERENCE INSTRUCTION.

111 ISE:

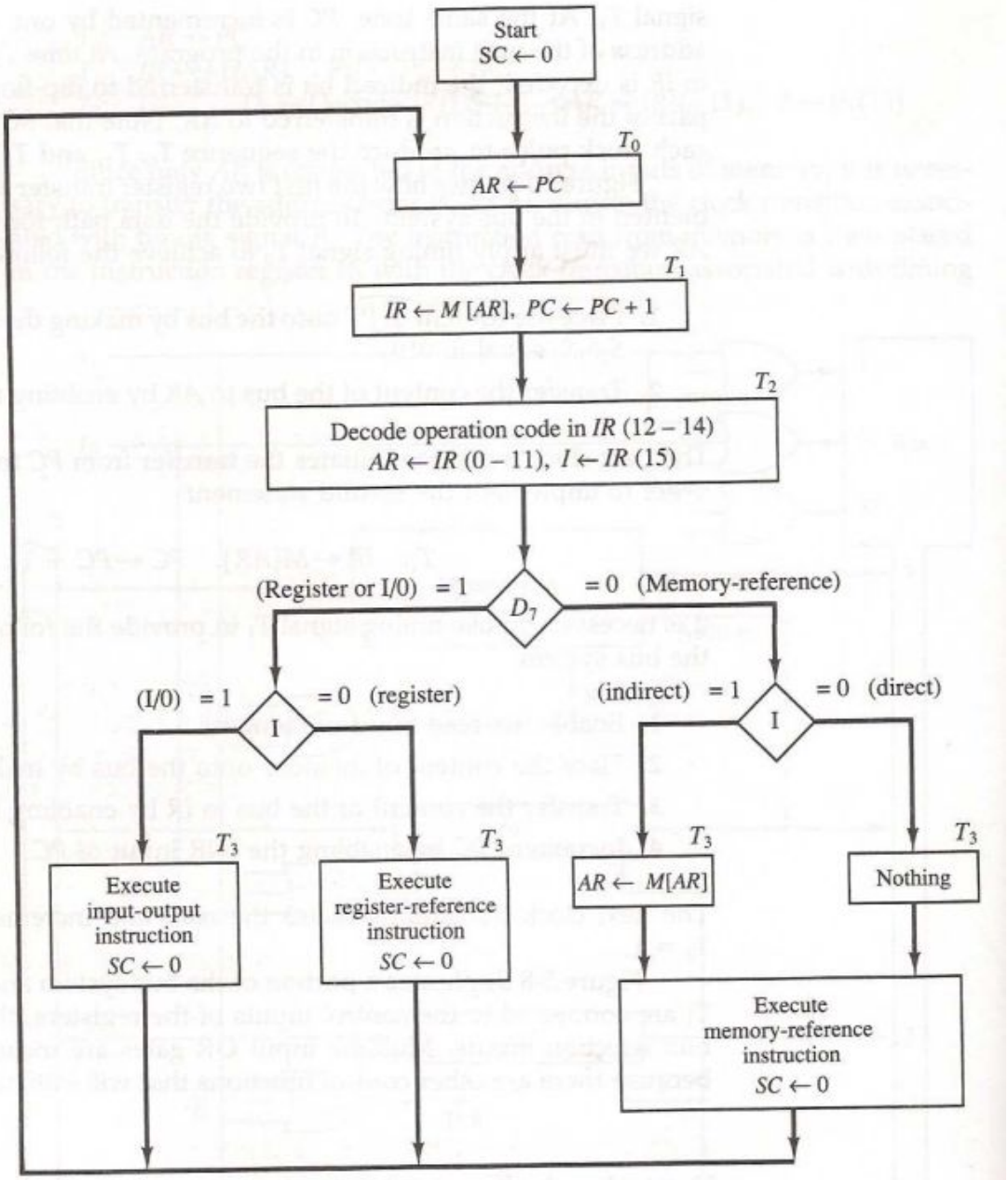
I 1 ISE, I/O INSTRUCTION.

I 0 ISE, REGISTER REFERENCE INSTRUCTION .

```
; =============== fetch, decode and execute cycle ===============
_start:
        AR load PC                      ; LOAD PC TO AR
        IR load M[AR]                   ; FETCH AR FROM MEMORY
                                          AND LOAD TO IR
        PC inc 0X01                     ; INCREASE PC

        decode IR[12-14]                ; DECODE OPCODE
        AR load IR[0-11]                ; LOAD LSB IN IR TO AR
        I load IR[15]                   ; LOAD I IN IR TO I FLIP-FLOP

                                        ; d7 IS MSB BIT
        if (d7 is 0X00):                ; MEMORY REFERENCE
                if (I is 0X00):         ; DIRECT
                        goto _execute   ; EXECUTE
                        SC load 0X00    ; LOAD 0 TO SC
                if (I is 0X01):         ; INDIRECT
                        AR load M[AR]   ; FETCH AR FROM MEMORY
                                          AND LOAD TO AR
                        goto _execute   ; EXECUTE
                        SC load 0X00    ; LOAD 0 TO SC
        else:                           ; REGISTER or I/O REFERENCE
                if (I is 0X00):         ; REGISTER REFERENCE
                        goto _execute   ; EXECUTE
                        SC load 0X00    ; LOAD 0 TO SC
                elif (I is 0X01):       ; I/O REFERENCE
                        goto _execute   ; EXECUTE
                        SC load 0X00    ; LOAD 0 TO SC
        goto _start                     ; AGAIN
```


Start
SC ← 0
T0
AR ← PC
T1
IR ← M [AR], PC ← PC + 1
T2
Decode operation code in IR (12 – 14)
AR ← IR (0 – 11), I ← IR (15)
(Register or I/0) = 1
D7
= 0 (Memory-reference)
(I/0) = 1
I
= 0 (register)
(indirect) = 1
I
= 0 (direct)
T3
Execute input-output instruction
SC ← 0
T3
Execute register-reference instruction
SC ← 0
T3
AR ← M[AR]
T3
Nothing
Execute memory-reference instruction
SC ← 0