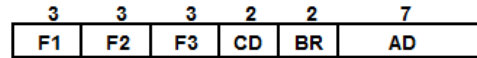


Microinstruction Format



F1, F2, F3: Microoperation fields
 CD: Condition for branching
 BR: Branch field
 AD: Address field

F1	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC + DR, E \leftarrow Cout$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0-10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0-10) \leftarrow PC$	PCTDR

F3	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow AC'$	COM
011	$AC \leftarrow cil AC$	CIL
100	$AC \leftarrow cir AC$	CIR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserved	

BR	Symbol	Function
00	JMP	$CAR \leftarrow AD$ if condition = 1 $CAR \leftarrow CAR + 1$ if condition = 0
01	CALL	$CAR \leftarrow AD, SBR \leftarrow CAR + 1$ if condition = 1 $CAR \leftarrow CAR + 1$ if condition = 0
10	RET	$CAR \leftarrow SBR$ (Return from subroutine)
11	MAP	$CAR(2-5) \leftarrow DR(11-14), CAR(0,1,6) \leftarrow 0$

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	DR(15)	I	Indirect address bit
10	$E = 1$	E	Carry bit
11	$AC = 0$	Z	Zero value in AC



Ad Soyad: _____ Numara: _____ İmza: _____

SORU 1:

Şekilde gösterilen mimaride ADD komutu (Opkod = 1100_B) bellekten okunacak 32 bitlik bir sayı ile (16 bitlik) AC'yi toplayacak ve sonucu tekrar bellekteki 32 bitlik alana yazacak şekilde yeniden düzenlenecektir.

Bellek 16 bitlik kelimelerden oluşmaktadır. Dolayısıyla bellekteki düşük adreste 32-bitlik sayının düşük manalı 16 biti, onu izleyen yüksek adreste ise 32 bitlik sayının yüksek manalı 16 biti bulunduğu varsayılacaktır. İşlem sonucu oluşabilecek elde biti E flip-flopunda saklanacaktır.

Bahsedilen yeni ADD komutunu gerçeklemek için gerekli mikroprogramı verilen tablolardaki sembolik mikrokomutları kullanarak yazınız. Derste kullanılan mapping yönteminin kullanıldığını varsayın.

Örnek: Doğrudan adresleme ile ADD 120 işletimi yürütülmeden hemen önce AC=0x8000, M[120]=0x9000 ve M[121]=0xFFFF değerleri bulunduğunu varsayalım. Yeni ADD komutu yürütüldükten sonra (0xFFFF9000 + 0x00008000=0x100001000) M[121]←0x0000, M[120]←0x1000 ve E←1 değerini alacaktır.

	M[121] 0xFFFF	M[120] 0x9000
		AC 0x8000
+		
E	M[121]	M[120]
1	0x0000	0x1000

SORU 2:

Soru 1 'de yazdığınız mikroprogramı verilen tabloları kullanarak ikili düzene çeviriniz. Her komut için bellek adresini ve mikrokomutun ikili düzendeki karşılığını veriniz.

Örnek:

Adress	Komut
xxxxxx	xxxxxxx
xxxxxx	xxxxxxx

SORU 1 - CEVAP:

```

ORG 0
NOP      I      CALL  INDIRECT
READ     U      JMP   NEXT
ADD      U      JMP   NEXT
DRTAC    U      JMP   LBL1  # Mimariye göre AC belleğe bağlı değil
                                # Dolayısıyla AC'nin içeriğini önce DR'ye gönderip
                                # DR üzerinden belleğe yazmamız gerekiyor.

```

Derste gösterilen mapping gereği her komut için
4 kelimelik yer bulunmaktadır. Dolayısı ile belleğin
40h adresinden sonra bir yere atlamamız gerekmektedir
40h (64) adresinde ve sonrasında FETCH ve INDIRECT alt
mikroprogramları bulunduğu için 60h'ye atlanabilir.

```

ORG 60h
LBL1: WRITE, INCAR I      JMP   NEXT
      CLRAC, READ  E      JMP   LBL2
      NOP          U      JMP   LBL3
LBL2: INCAC        U      JMP   NEXT
LBL3: ADD          U      JMP   NEXT
      DRTAC        U      JMP   NEXT
      WRITE        U      JMP   FETCH

```

SORU 2 - CEVAP:

Etiket	Adres
INDIRECT	1000100 (dersteki değer)
LBL1	1100000

Adres	Veri
0000000	0x002C4
0000001	0x10002
0000010	0x20003
0000011	0x80060
1100000	0xE3A61
1100001	0x50463
1100010	0x00064
1100011	0x60064
1100100	0x20065

1100101	0x80066
1100110	0xE0040