

5 + (-1) = 4      00101 + 11111 = 100100 (görüldüğü gibi işlemin sonucu 6 bitlik bir değer çıktı oysaki değerlerimiz 5 bit idi. Bu durumda overflow yani taşma olmuştur. Aslında Sonuç : 00100 dür.)

b) 5 – 11 **Overflow ( taşma) olmamıştır.**

**Two's complement Notation için 5 in gösterimi: 00101**

Eğer değerimiz negatifse pozitif halinin değerlerini ters çevirip 1 ekliyorduk. Yeni oluşan gösterim negatif halidir. Mesela two's complement notation da -11 in gösterimi için önce 11 i bulmalıyız.

**Two's complement Notation için 11 in gösterimi: 01011**

**01011 => 10100 (öncelikle değerlerin zıttını aldık.) => 10101 (daha sonra oluşan değere 1 ekledim.)**

**Two's complement Notation için -11 in gösterimi: 10101**

**5 + (-11) = -6                      00101 + 10101 = 11010 (görüldüğü gibi işlemin sonucu 5 bitlik bir değer çıktı.**

**Bu durumda overflow yani taşma olmamıştır.)**

4. Aşağıdaki işlemleri gerçekleştiriniz.

a) 01001011 AND 10101011 **11110110**

01001011                      **AND için eğer 2 değerden biri 0 ise sonuç 0 dir. Eğer her ikisi de 1 ise sonuç 1 dir.**  
10101011  
AND -----  
**00001011**

b) 01001011 OR 10101011

01001011                      **OR için eğer 2 değerden biri 1 ise sonuç 1 dir. Eğer her ikisi de 0 ise sonuç 0 dir.**  
10101011  
OR -----  
**11101011**

c) 01001011 XOR 10101011

01001011                      **XOR için eğer 2 değer birbirlerinden farklı ise sonuç 1 dir. Eğer her iki değer eşit ise sonuç 0 dir.**  
10101011  
XOR -----  
**11100000**

5. Aşağıdakiler verilenler, kitabınızdaki kılavuza (Appendix C) göre makine diliyle yazılmış komutlardır (instruction). Her komutun ne olduğunu bulunuz.

a) 7123

**OR the bit patterns in registers 2 and 3 and place the result in register 1. = Register 2 ve Register 1 bit modellerine OR uygula daha sonra çıkan sonucu Register 1 e yaz.**

b) 2BCD

**LOAD the register B with the bit pattern CD. = CD değerini B register na yükle.**

6. Kitabınızdaki kılavuza (Appendix C) göre A0 adresli belleğin içindeki değer ilk dört bitini A1 adresli belleğin içindeki değer son dört biti ile birleştirip sonuçta 16 bitlik bir değer elde eden ve bu değeri A2 adresli belleğe yazan bir programı çeviri dilinde (assembly language) yazınız.

**İlk olarak A0 ve A1 adresli belleğe Register dan 8 bitlik bir değer yüklenmelidir. Bundan dolayı önce registerlara değerler yüklenmeli daha sonra bu değerler registerdan main memory aktarılmalıdır.**

**LOAD R2,34      (A0 ' a yüklenecek 8 bitlik herhangi bir değeri Register 2 ye yazdım.)**

**LOAD R3,56      (A1 ' e yüklenecek 8 bitlik herhangi bir değeri Register 3 ye yazdım.)**

**Daha sonra Register dan bu değerleri Main Memory deki A0 ve A1 adreslerine yazmam gerekli.**

**STORE R2,[160]    ( 160 A0 'ın memory deki adresidir. R2 deki değeri A0 da depoladım.)**

**STORE R3,[161]    ( 161 A1 'ın memory deki adresidir. R2 deki değeri A1 de depoladım.)**

**Bu şekilde A0 ve A1 de 8 er bitlik değerler bulunmaktadır. Aslında sorunun çözümü şimdi başlıyor.**

A0 ın ilk 4 bitini almak için son 4 bit in 0 olması gerekir. Aynı şekilde A1 in son 4 bitini almak için ilk 4 bitinin 0 olması gerekir. Çünkü bu şekilde bu değerleri toplayabilir ve 8 bitlik değeri oluşturup Main Memory deki A2 adresine yazabilirim.

**A0 'ın son 4 bitini sıfırlamak için:**

11110000 ile A0 daki değeri AND leyerek son 4 biti sıfırlayabilirim. Ama önce 11110000 i herhangi bir register'a yükleyip A0 İLE AND işlemine sokmam lazım.

Örneğin: 10111101  
11110000  
AND -----  
10110000

10110000 'in Hexadecimal karşılığı 120'dir. Bunu register yazalım.

**LOAD R4,120**

**AND R6,R2,R4** ( Register 2 ile register 4 ü and leyip register 6 ya yazdım.)

**A1'in ilk 4 bitini sıfırlamak için:**

00001111 ile A1 deki değerleri AND leyerek ilk 4 bitini sıfırlayabilirim. Ama önce 00001111 i herhangi bir register a yükleyip A1 ile AND işlemi sokmam lazım.

Örneğin: 10111101  
00001111  
AND -----  
00001101

00001101 'in Hexadecimal karşılığı 15'dir. Bunu register yazalım.

**LOAD R5,15**

**AND R7,R3,R5** ( Register 3 ile register 5 i and leyip register 7 ye yazdım.)

Artık A0 ın ilk 4 biti R6 'da A1 in son 4 biti R7 de bulunuyor. R6 ile R7 yi toplayıp herhangi bir register a yazıp onu da A2 ye store edebilirim. [162] main memory de A2 ye denk gelir.

**ADDI R8,R7,R6**

**STORE R8,[162]**

---

**LOAD R2,34**

**LOAD R3,56**

**STORE R2,[160]**

**STORE R3,[161]**

**LOAD R4,120**

**AND R6,R2,R4**

**LOAD R5,15**

**AND R7,R3,R5**

**ADDI R8,R7,R6**

**STORE R8,[162]**

**halt**