

**Gebze Technical University**  
**Computer Engineering**

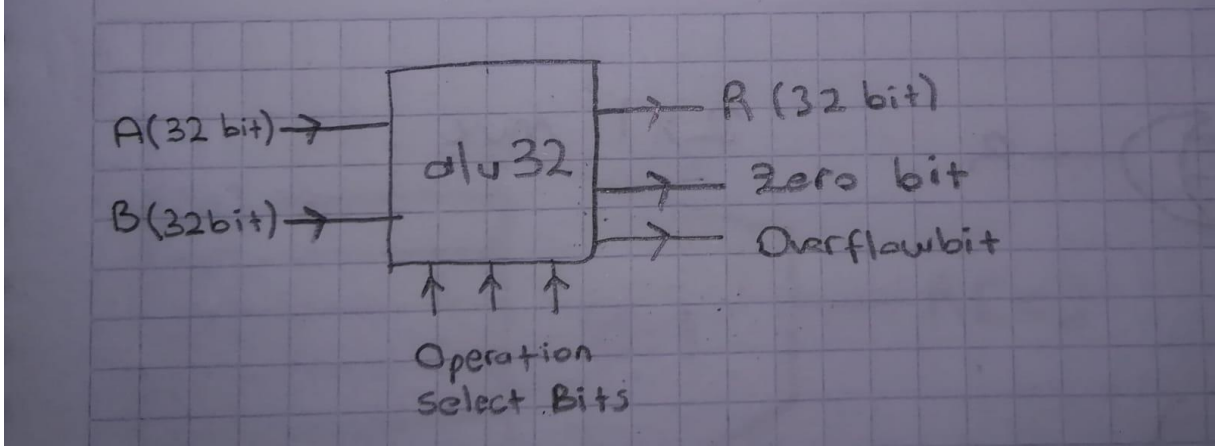
**CSE 331 - 2018**

**HOMEWORK 2 REPORT**

**SERKAN SORMAN**  
**151044057**

Course Assistant: Fatma Nur Esirci

**Module alu32:** Input olarak iki adet 32 bit sayı ve hangi işlemin yapılacağını belirten 3 adet sinyal bitini alır. Gelen 3 sinyal bitine göre seçilen işlemi yapar. Output olarak sonuç, Zero biti ve Addition, Substraction işlemleri sonucu oluşan Overflow bitini verir.

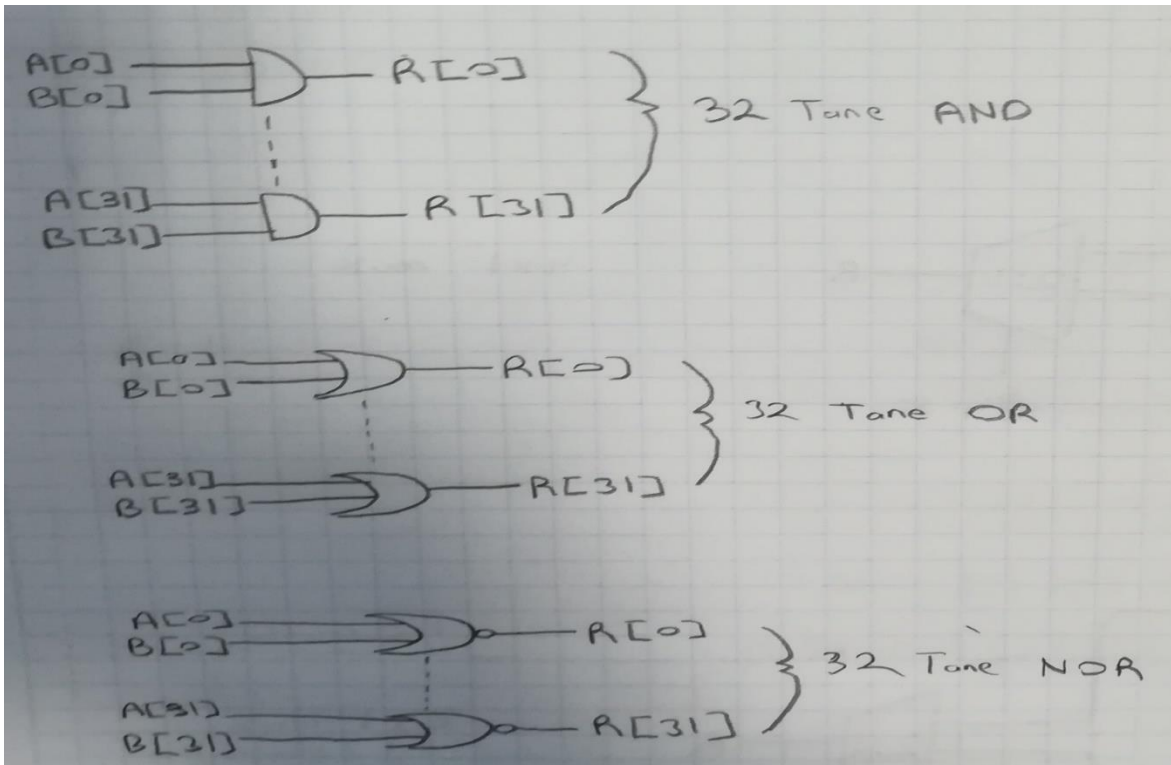


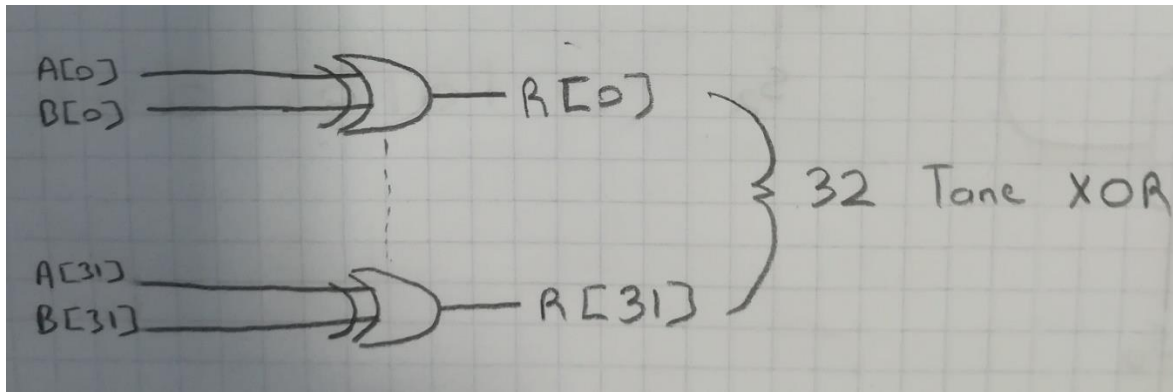
**Module andop:** Input olarak iki adet 32 bit sayı alır. Output olarak iki sayının tüm bitlerinin andlenmiş halini verir.

**Module orop:** Input olarak iki adet 32 bit sayı alır. Output olarak iki sayının tüm bitlerinin orlanmış halini verir.

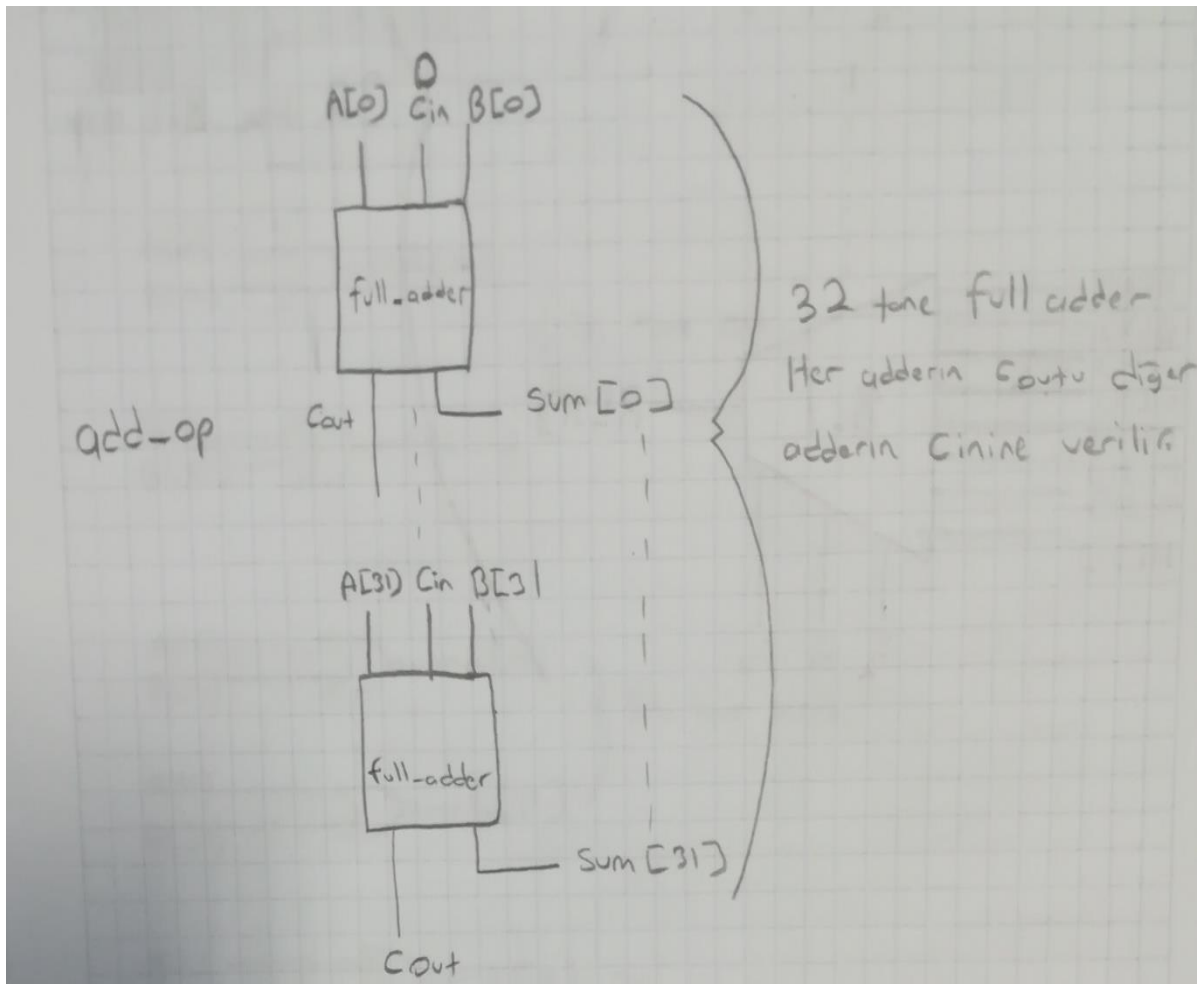
**Module xorop:** Input olarak iki adet 32 bit sayı alır. Output olarak iki sayının tüm bitlerinin xorlanmış halini verir.

**Module norop:** Input olarak iki adet 32 bit sayı alır. Output olarak iki sayının tüm bitlerinin norlanmış halini verir.

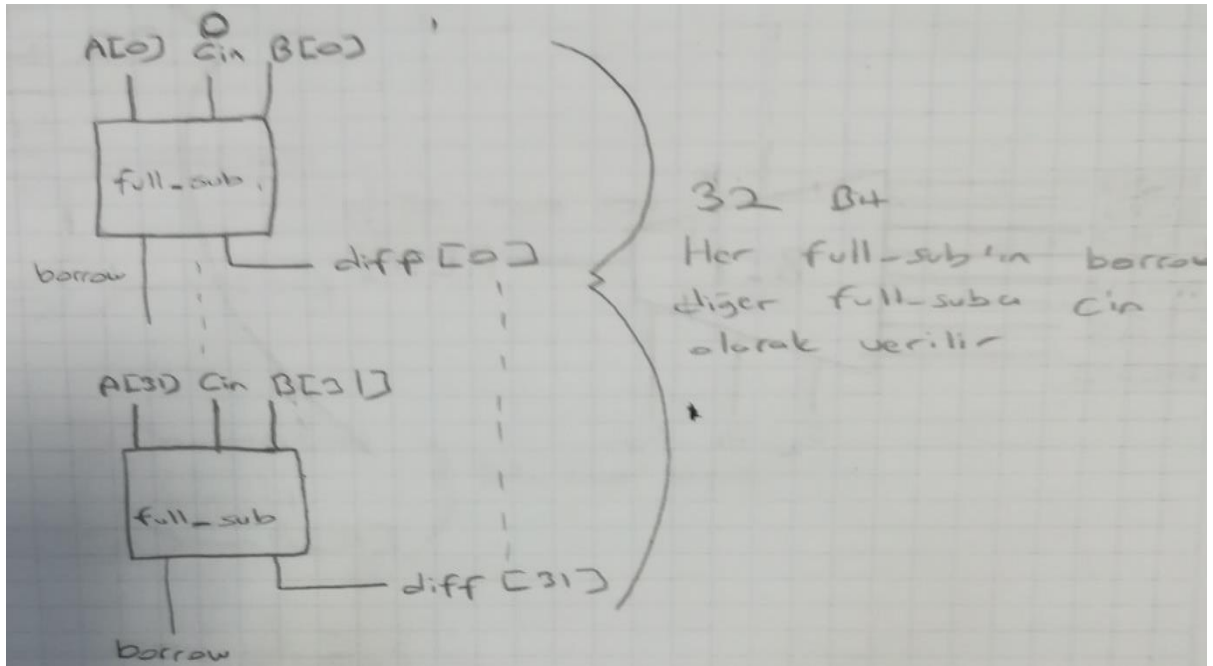




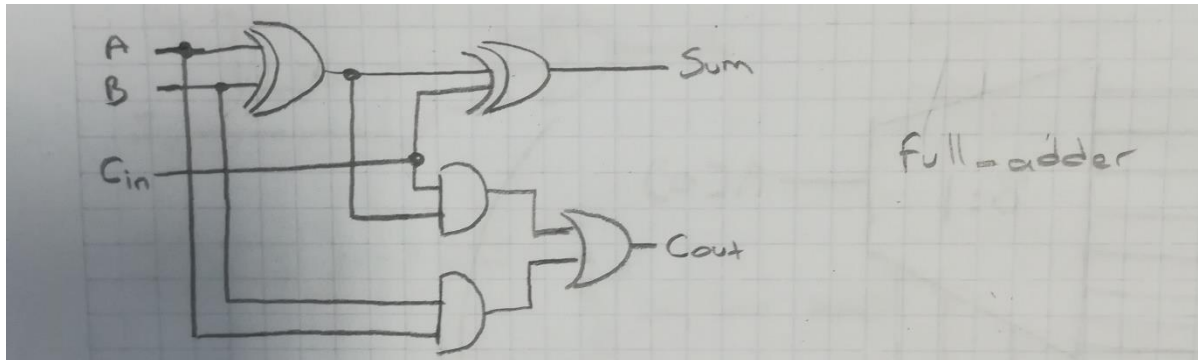
**Module addop:** Input olarak iki adet 32 bit sayı alır ve bu sayılar birer bit olarak full\_addera yollar. Output olarak carry out ve iki sayının toplamını verir.



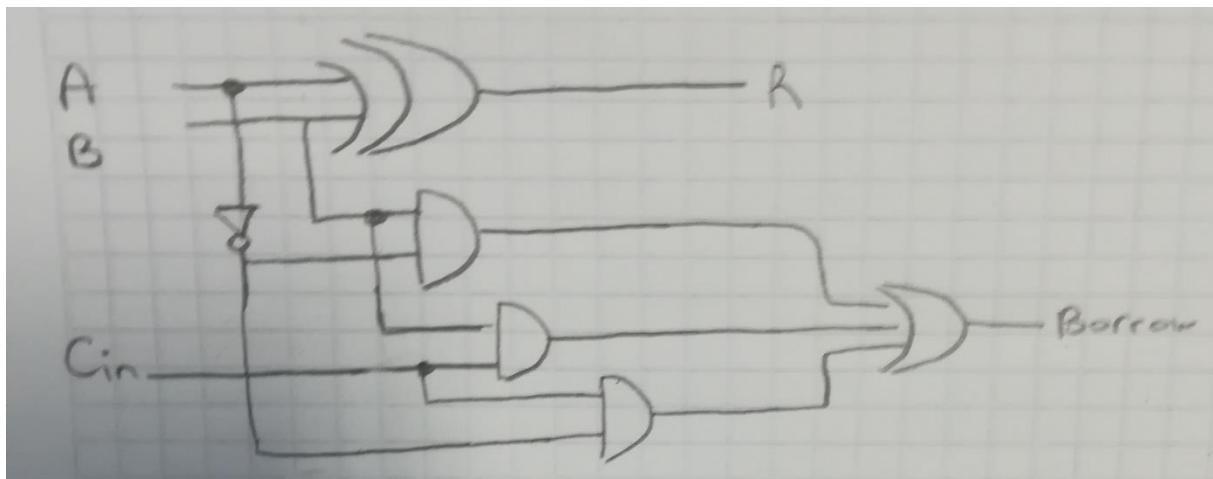
**Module subop:** Input olarak iki adet 32 bit sayı alır ve bu sayılar birer bit olarak full\_suba yollarılır. Output olarak borrow ve iki sayının farkını verir.



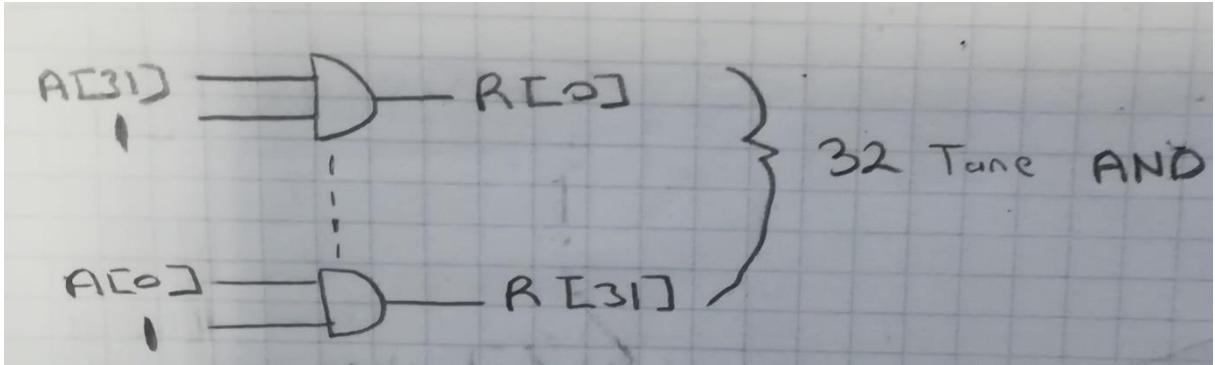
**Module full\_adder:** Input olarak iki adet 1 bit sayı ve 1 bit carry in alır. Output olarak carry out ve iki bitin toplamını verir.



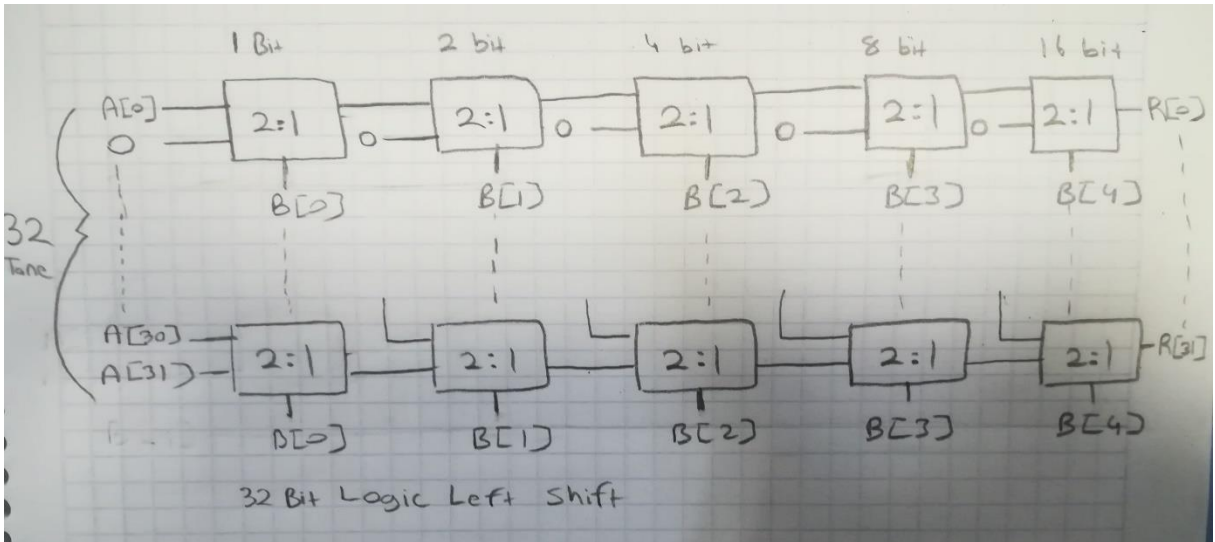
**Module full\_sub:** Input olarak iki adet 1 bit sayı ve 1 bit carry in alır. Output olarak borrow ve iki bitin farkını verir.



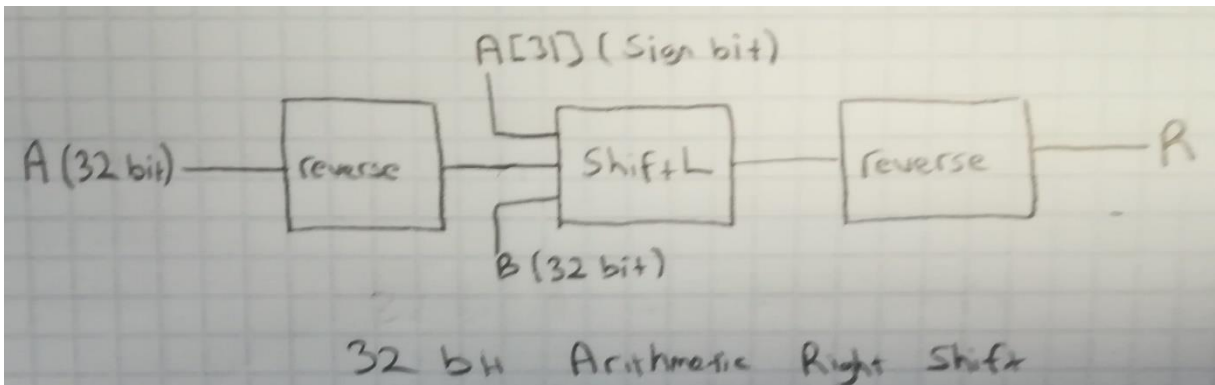
**Module reverse:** Input olarak verilen 32 bit sayının ters çevrilmiş halini output olarak verir.



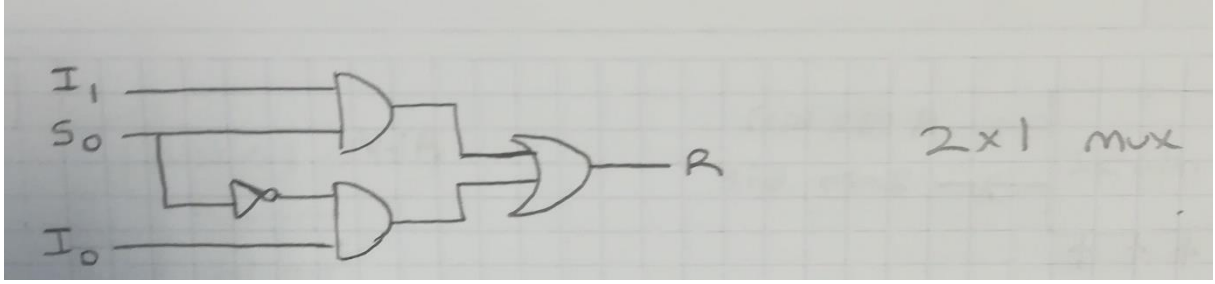
**Module ShiftL:** Input olarak Logic Left Shift edilecek 32 bit bir sayı, Shift edilecek basamak sayısını belirten ikinci bir 32 bit sayı ve Shift işleminde kaydırılan bitlerin yerine konacak biti belirten bit alınır (Logic left shift yapıldığında 0 yollar). Output olarak verilen 32 bit sayının Left Logic Shift edilmiş hali verilir.



**Module ShiftR:** Input olarak Arithmetic Right Shift edilecek 32 bit bir sayı, Shift edilecek basamak sayısını belirten ikinci bir 32 bit sayı alır. Verilen sayı reverse edilip Logic Left Shift yapılır (Kaydırılan bitlerin yerine konması için sign biti yollar). Ardından tekrar reverse edilir ve Output olarak verilir.

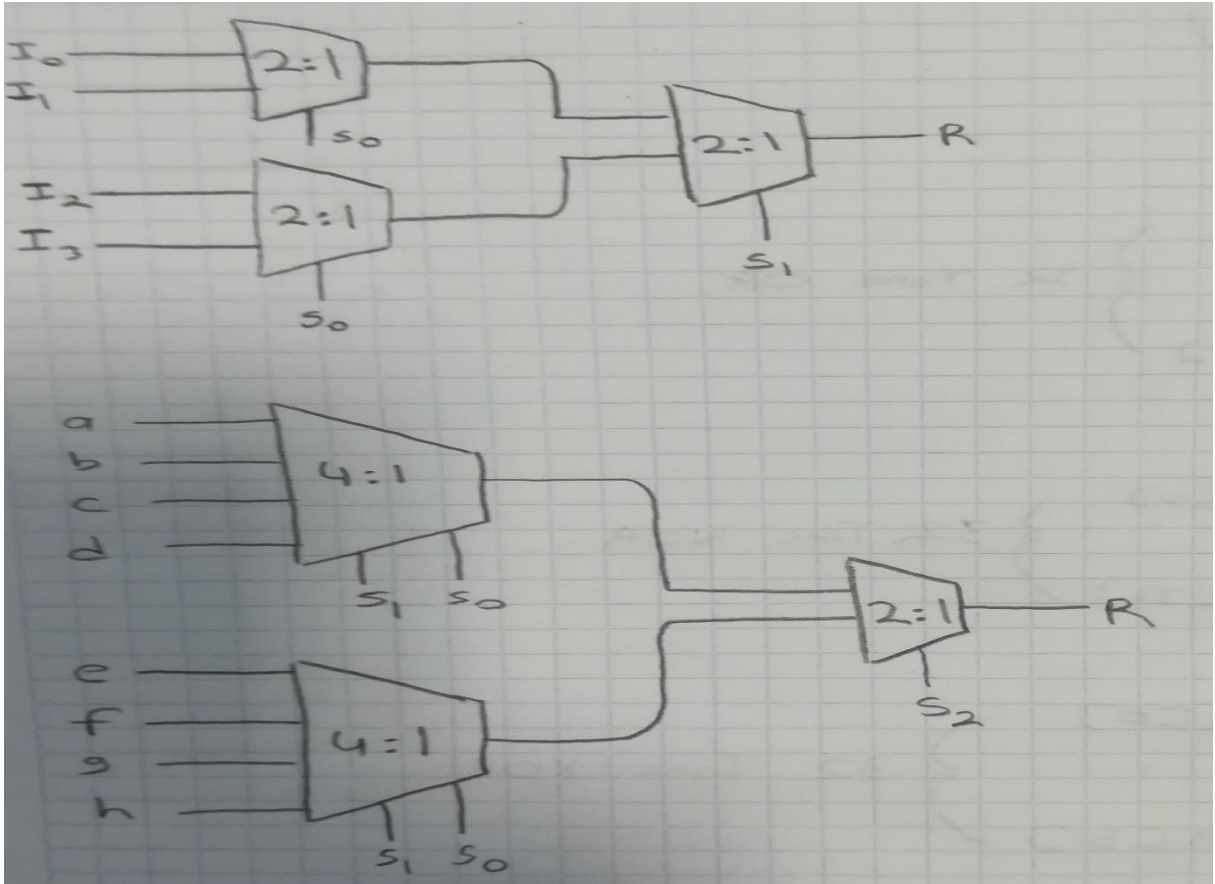


**Module mux2x1:** İki adet birer bit input ve bir seçici olarak inputlardan birini verir.



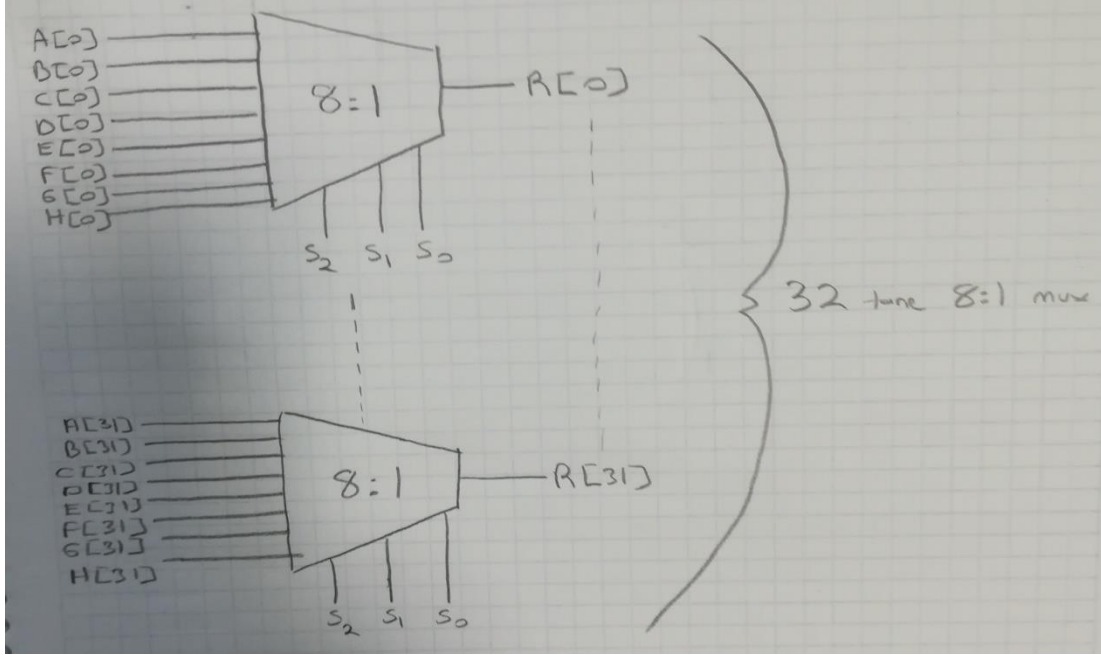
**Module mux4x1:** Dört adet birer bit input ve iki seçici olarak inputlardan birini verir. Üç adet 2:1 mux kullanılarak tasarlanmıştır.

**Module mux8x1:** Sekiz adet birer bit input ve üç seçici olarak inputlardan birini verir. İki adet 4:1 mux ve 1 adet 2:1 mux kullanılarak tasarlanmıştır.

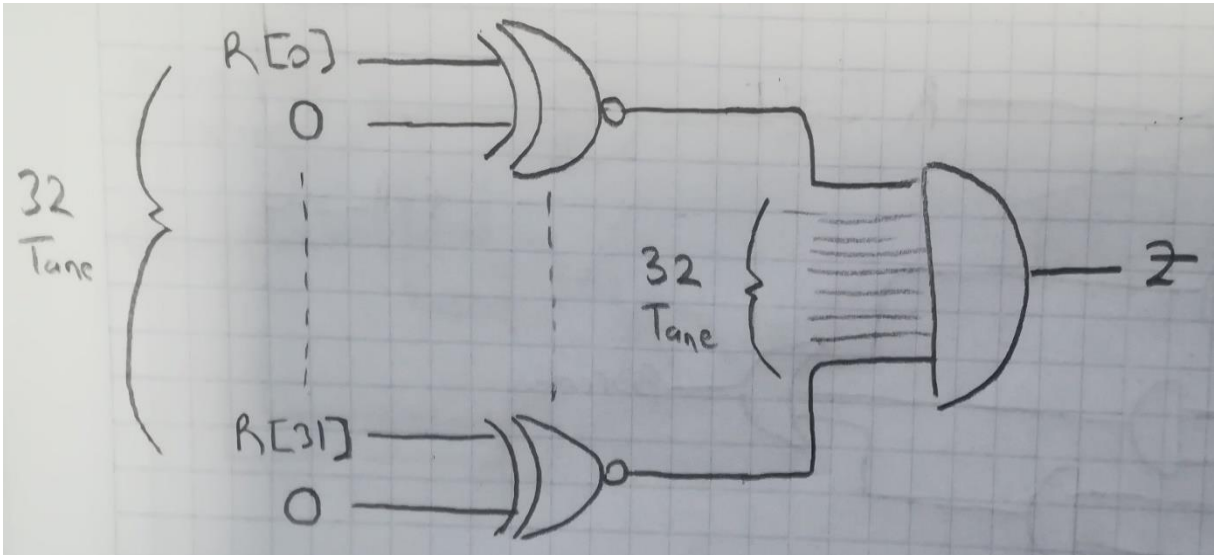




**Module mux8x1\_32:** 32 bit 8 sayı ve 3 seçici biti input olarak alarak bu 32 bitlik 8 sayıdan birini (Hesaplanmış 8 operasyon sonucundan biri) verir.



**Module zero:** Input olarak verilen 32 bit sayının tüm bitlerinin 0 olup olmadığını kontrol eder. Tüm bitleri sıfır ise Zero bitini output olarak 1 verir.



```
# time = 0, A = 111000011000000110101110000101, B = 1110000000011100000001110000101, Operation = 000, Result = 111000000000000000000001110000000, Zero=0, Overflow=0
# time = 10, A = 11010101010101001011010100101010, B = 01011110001100000001110011110110, Operation = 000, Result = 010101000010000000101000100010, Zero=0, Overflow=0
# time = 20, A = 10101010101010101010101010101010, B = 01010101010101010101010101010101, Operation = 000, Result = 00000000000000000000000000000000, Zero=1, Overflow=0
# time = 30, A = 11100000000000000111111111111111, B = 0000001110000000111111000001111, Operation = 001, Result = 11100011100000001111111111111111, Zero=0, Overflow=0
# time = 40, A = 110101010101100101101010010101010, B = 01011110001100000001110011110110, Operation = 001, Result = 110111110111100101110111111110, Zero=0, Overflow=0
# time = 50, A = 10101010101010101010101010101010, B = 01010101010101010101010101010101, Operation = 001, Result = 11111111111111111111111111111111, Zero=0, Overflow=0
# time = 60, A = 01000000000000000000000000000000, B = 01000000000000000000000000000000, Operation = 010, Result = 10000000000000000000000000000000, Zero=0, Overflow=1
# time = 70, A = 00000000000000000000000000001100, B = 00000000000000000000000000010001, Operation = 010, Result = 00000000000000000000000000001101, Zero=0, Overflow=0
# time = 80, A = 00000000000000000000000001011110, B = 00000000000000000000000000001101, Operation = 010, Result = 00000000000000000000000001011101, Zero=0, Overflow=0
# time = 90, A = 11100000000000001111111111111111, B = 0000001110000000111111000001111, Operation = 011, Result = 11100011100000000000000011111000, Zero=0, Overflow=0
# time = 100, A = 110101010101010010101010010101010, B = 01011110001100000001110011110110, Operation = 011, Result = 10001011100111001010100111011100, Zero=0, Overflow=0
# time = 110, A = 10101010101010101010101010101010, B = 01010101010101010101010101010101, Operation = 011, Result = 11111111111111111111111111111111, Zero=0, Overflow=0
# time = 120, A = 0000000000000000000000000100000, B = 1000000000000000000000000000011001, Operation = 100, Result = 10000000000000000000000000000111, Zero=0, Overflow=1
# time = 130, A = 00000000000000000000000000000001, B = 0000000000000000000000000000001100, Operation = 100, Result = 000000000000000000000000000000101, Zero=0, Overflow=0
# time = 140, A = 0000000000000000000000000000101110, B = 00000000000000000000000000000001101, Operation = 100, Result = 000000000000000000000000000010100001, Zero=0, Overflow=0
# time = 150, A = 111110000000000000000000000011111, B = 00000000000000000000000000000101, Operation = 101, Result = 11111111100000000000000000000000, Zero=0, Overflow=0
# time = 160, A = 00011100011100000000001111000111, B = 000000000000000000000000000000111, Operation = 101, Result = 000000000011100011100000000000111, Zero=0, Overflow=0
# time = 170, A = 10101010101010101010101010101010, B = 0000000000000000000000000000001111, Operation = 101, Result = 11111111111111111111111111111111, Zero=0, Overflow=0
# time = 180, A = 11111000000000000000000000000001, B = 00000000000000000000000000000101, Operation = 110, Result = 0000000000000000000000000001001100000, Zero=0, Overflow=0
# time = 190, A = 00011100011100000000001111000111, B = 000000000000000000000000000000111, Operation = 110, Result = 00111000000000011110001110000000, Zero=0, Overflow=0
# time = 200, A = 01010100101010101010101010101010, B = 0000000000000000000000000000001111, Operation = 110, Result = 00000000000000000000000000000000, Zero=1, Overflow=0
# time = 210, A = 1110000000000000001111111111111111, B = 00000011100000000111111000001111, Operation = 111, Result = 00011000111111100000000000000000, Zero=0, Overflow=0
# time = 220, A = 110101010101010010101010010101010, B = 0101111000110000000111011111101101, Operation = 111, Result = 00100000010000110100001000000001, Zero=0, Overflow=0
# time = 230, A = 10101010101010101010101010101010, B = 01010101010101010101010101010101, Operation = 111, Result = 00000000000000000000000000000000, Zero=1, Overflow=0
```

