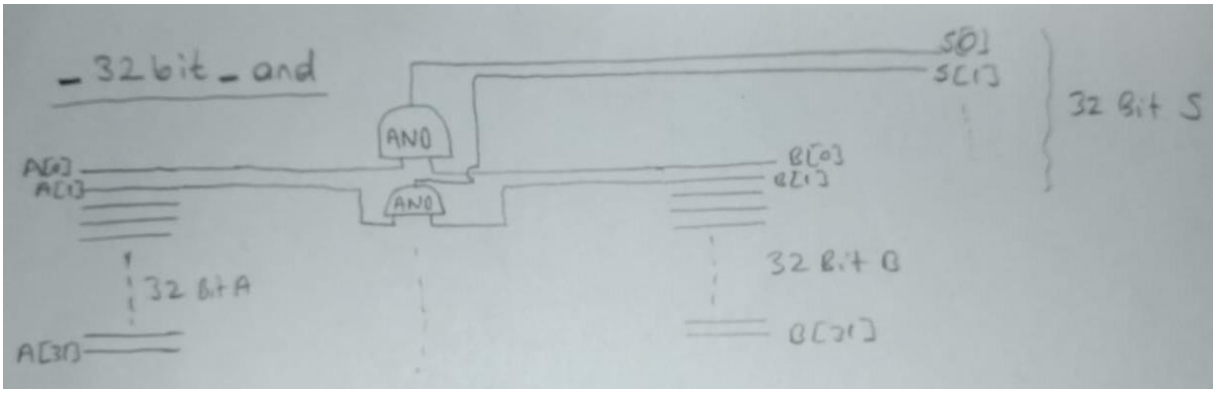
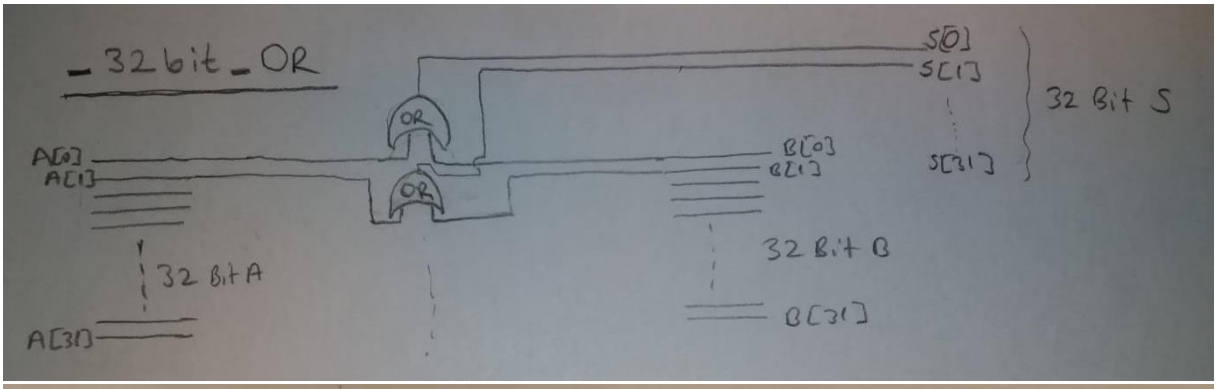
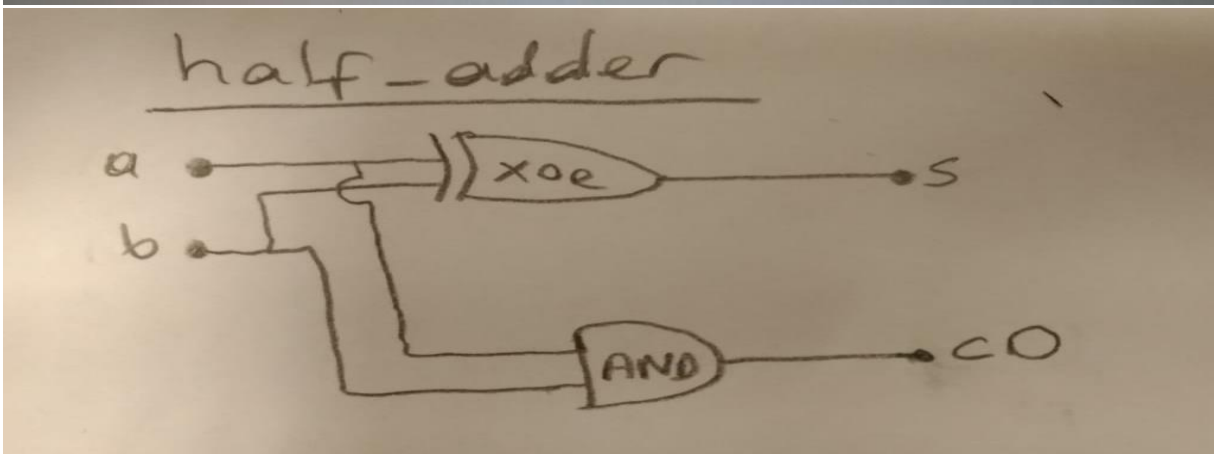
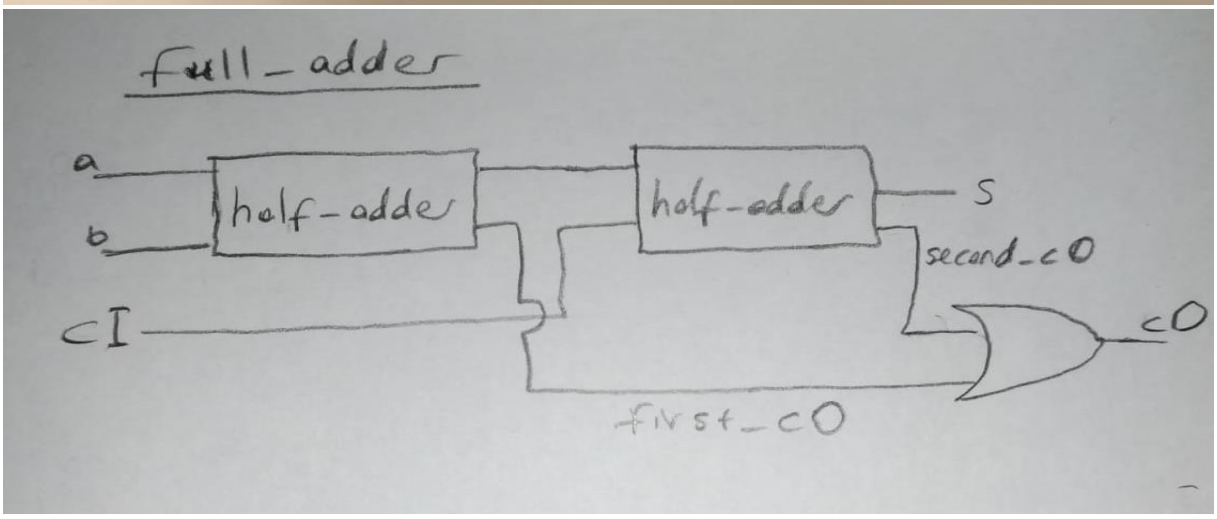
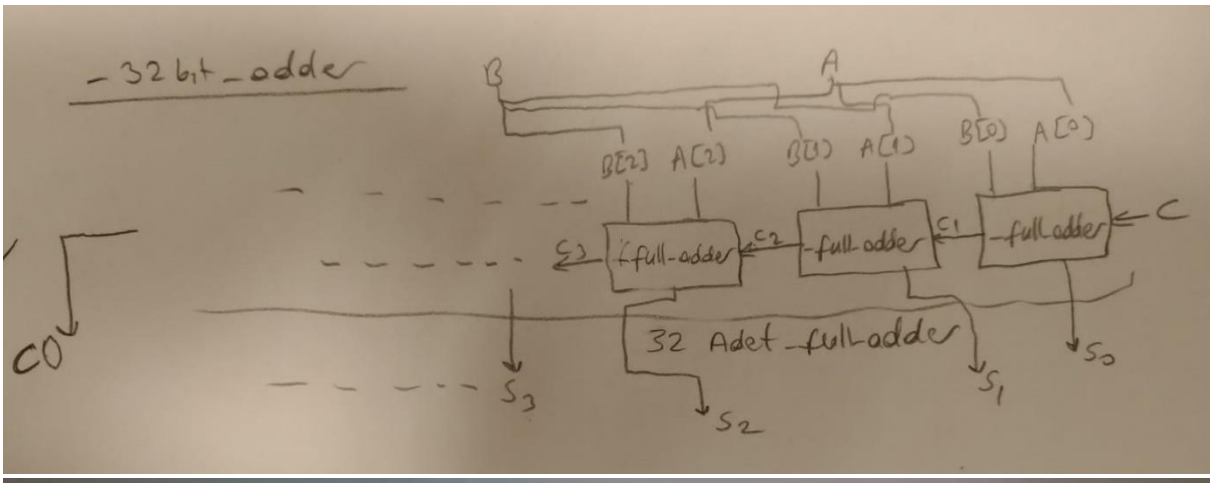
1.Schematic Designs For All Modules.

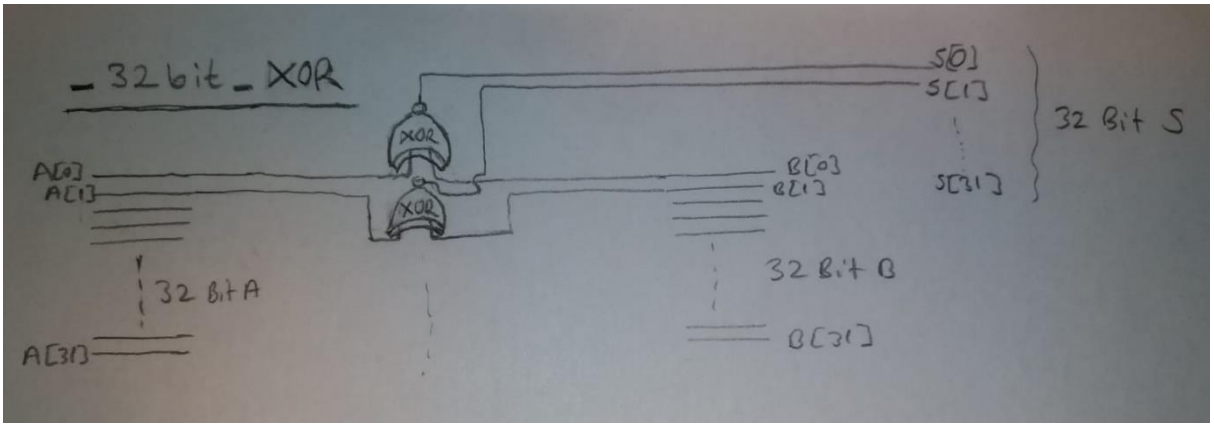


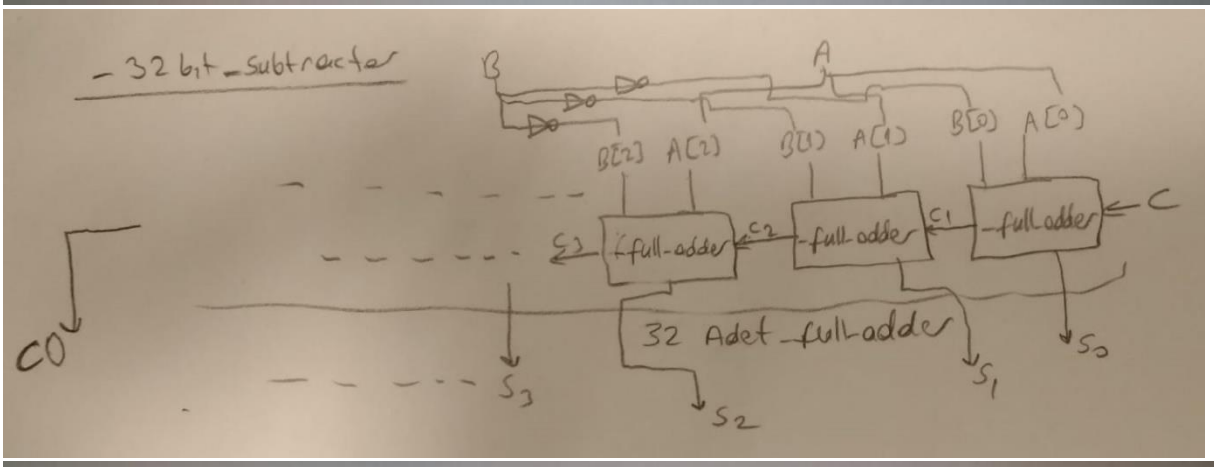


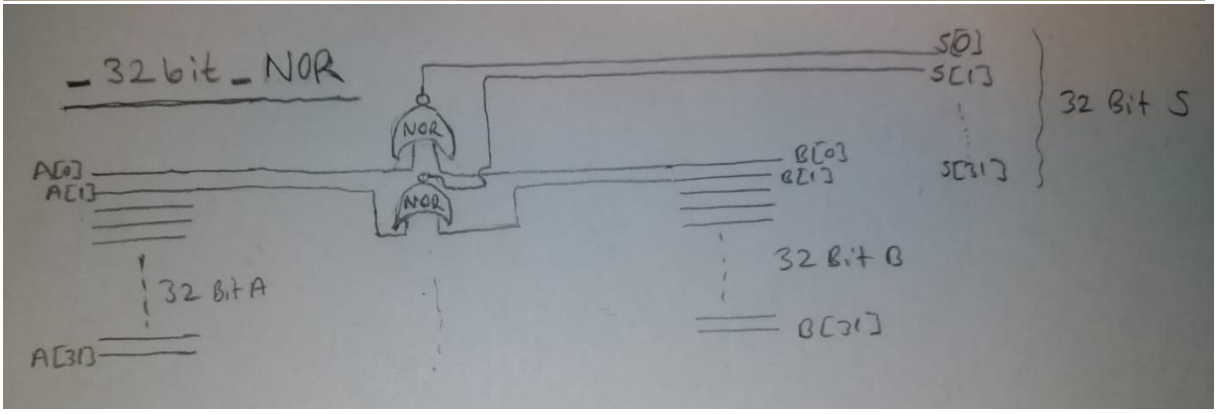




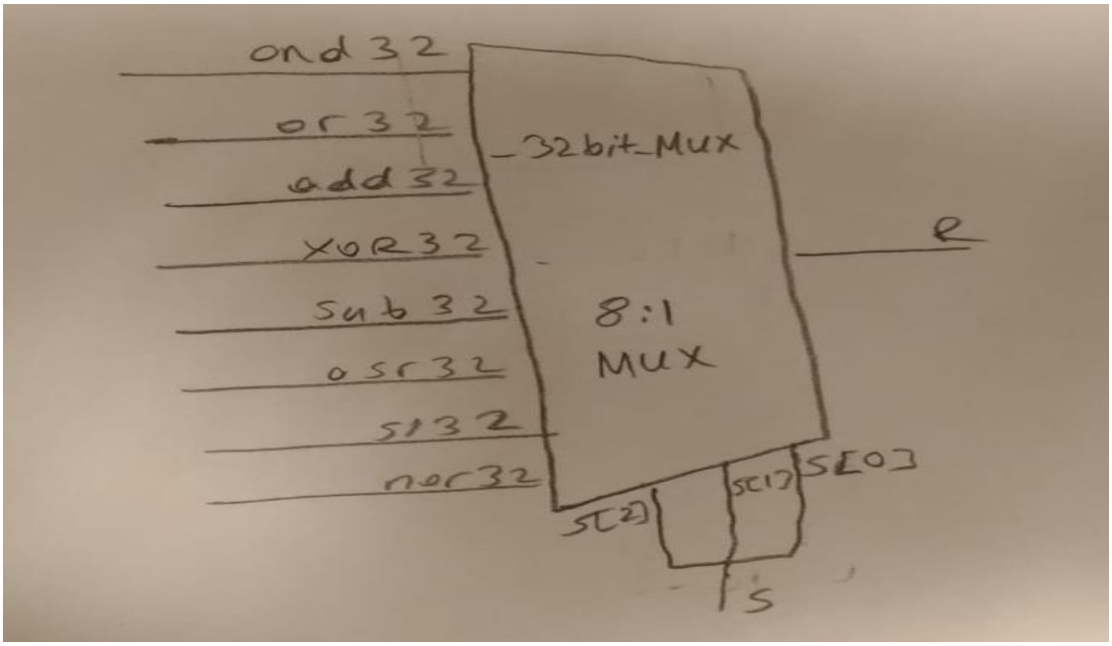

- 32 bit _ adder
B
A
B[2] A[2] B[1] A[1] B[0] A[0]
full-adder
full-adder
full-adder
C
C1
C2
C3
C0
32 Adet _ full-adder
S3
S2
S1
S0

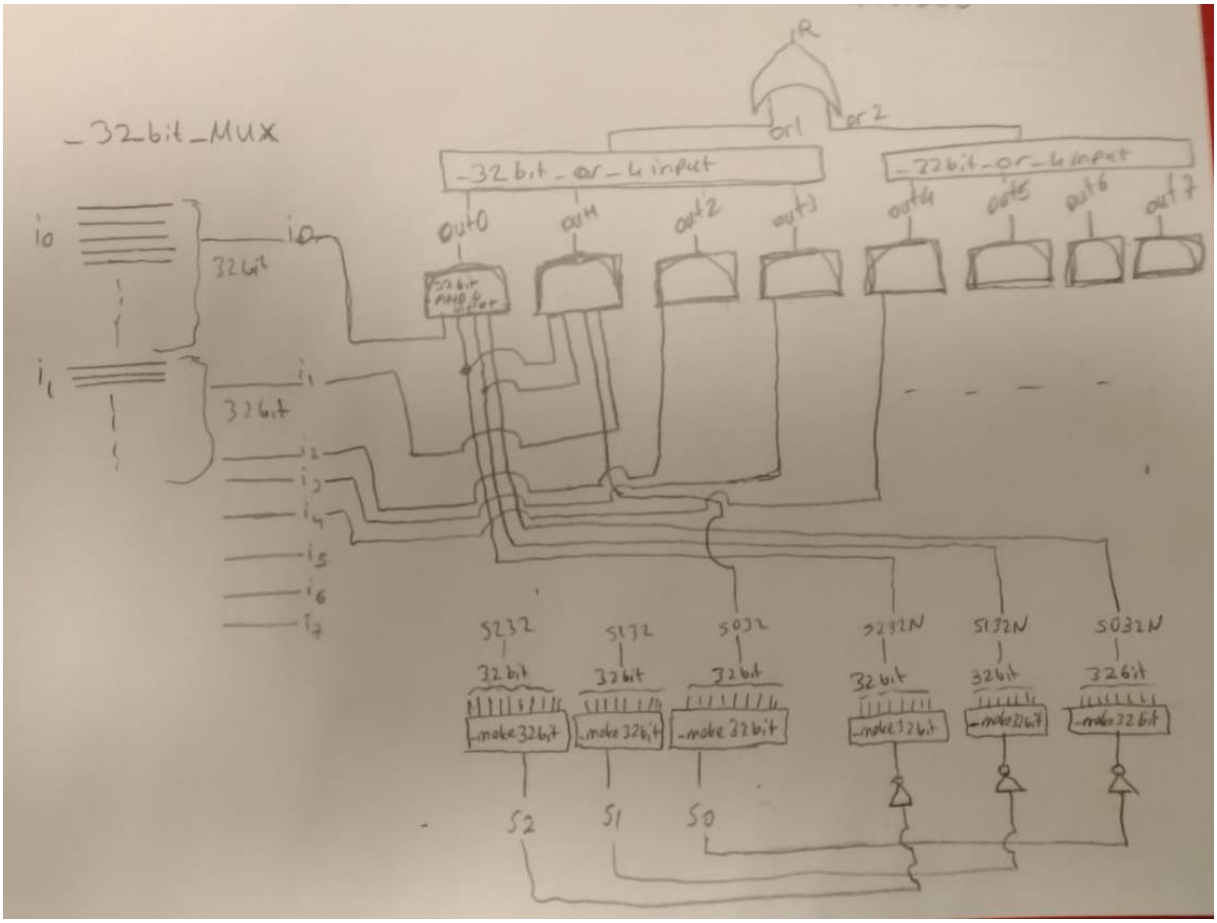

- 32 bit _ XOR
XOR
XOR
A[0]
A[1]
A[31]
32 Bit A
B[0]
B[1]
B[31]
32 Bit B
S[0]
S[1]
S[31]
32 Bit S

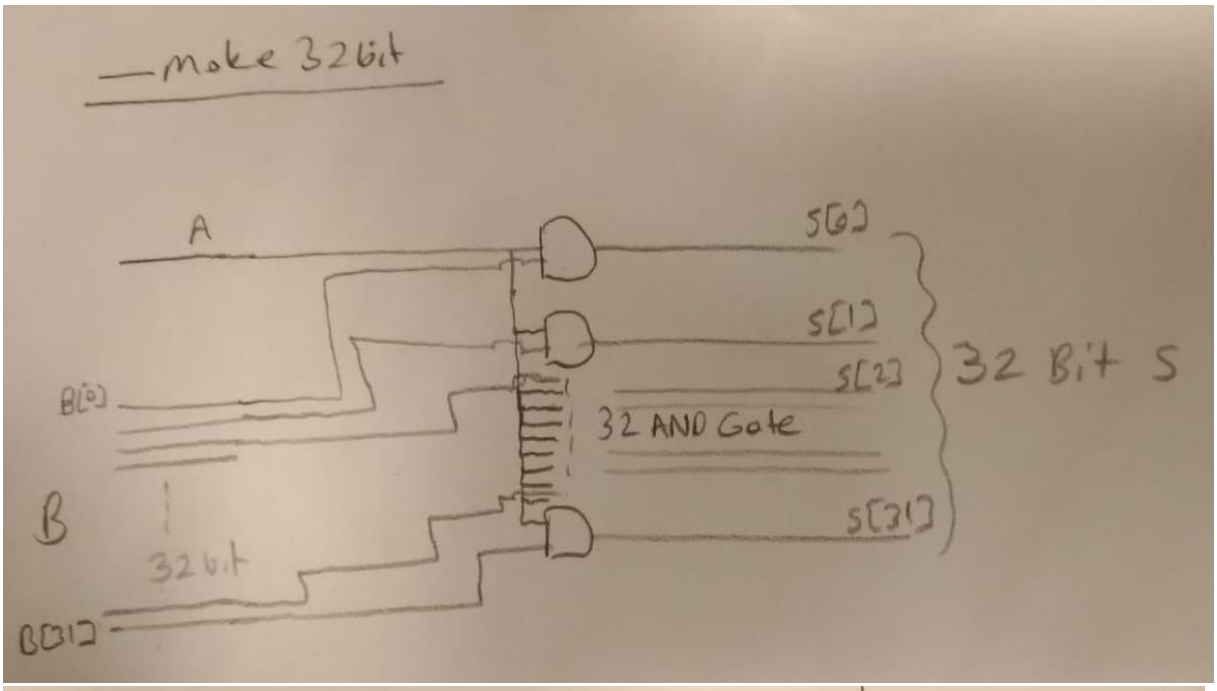

- 32 bit _ Subtractor
B
A
B[2] A[2] B[1] A[1] B[0] A[0]
full-adder
full-adder
full-adder
C
C1
C2
C3
C0
32 Adet _ full-adder
S3
S2
S1
S0

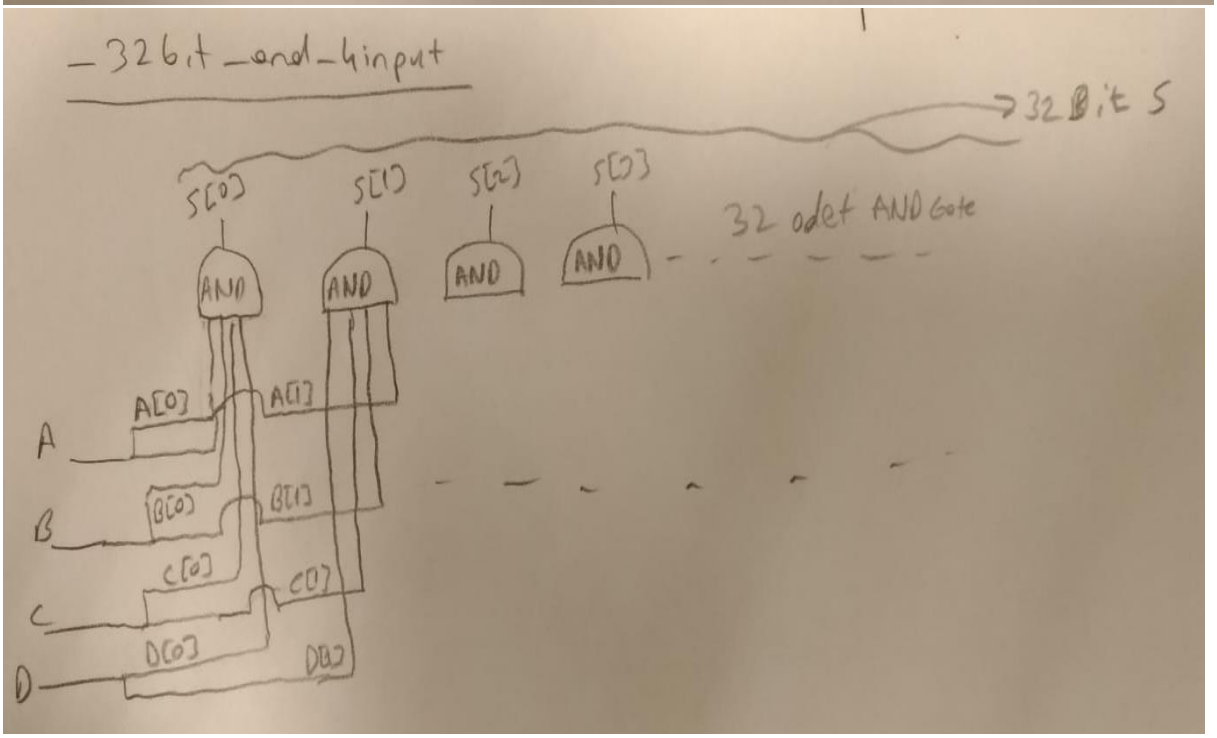

- 32 bit _ NOR
NOR
NOR
A[0]
A[1]
A[31]
32 Bit A
B[0]
B[1]
B[31]
32 Bit B
S[0]
S[1]
S[31]
32 Bit S

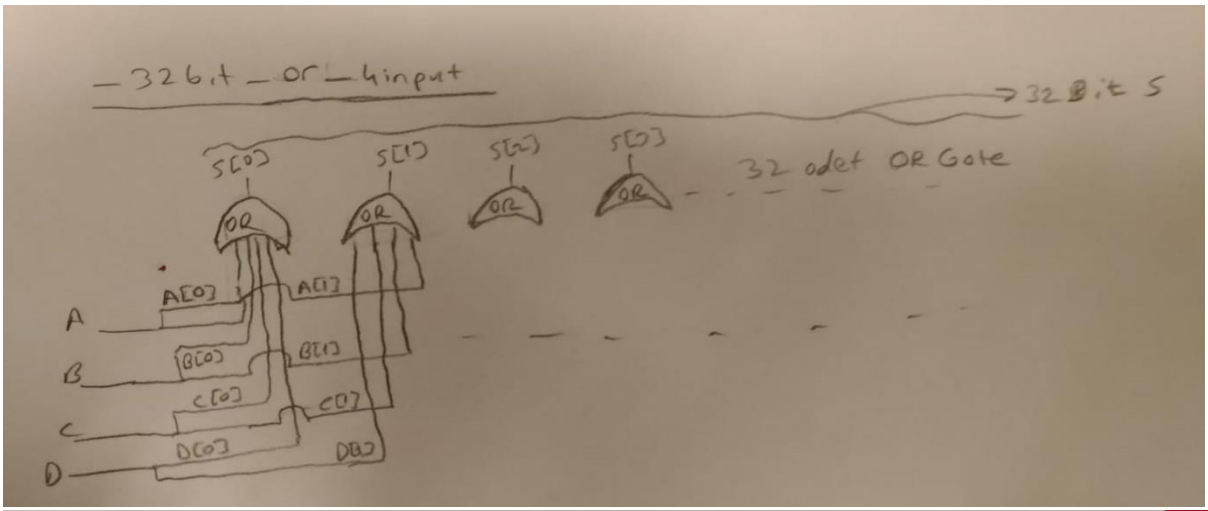
ALU32:

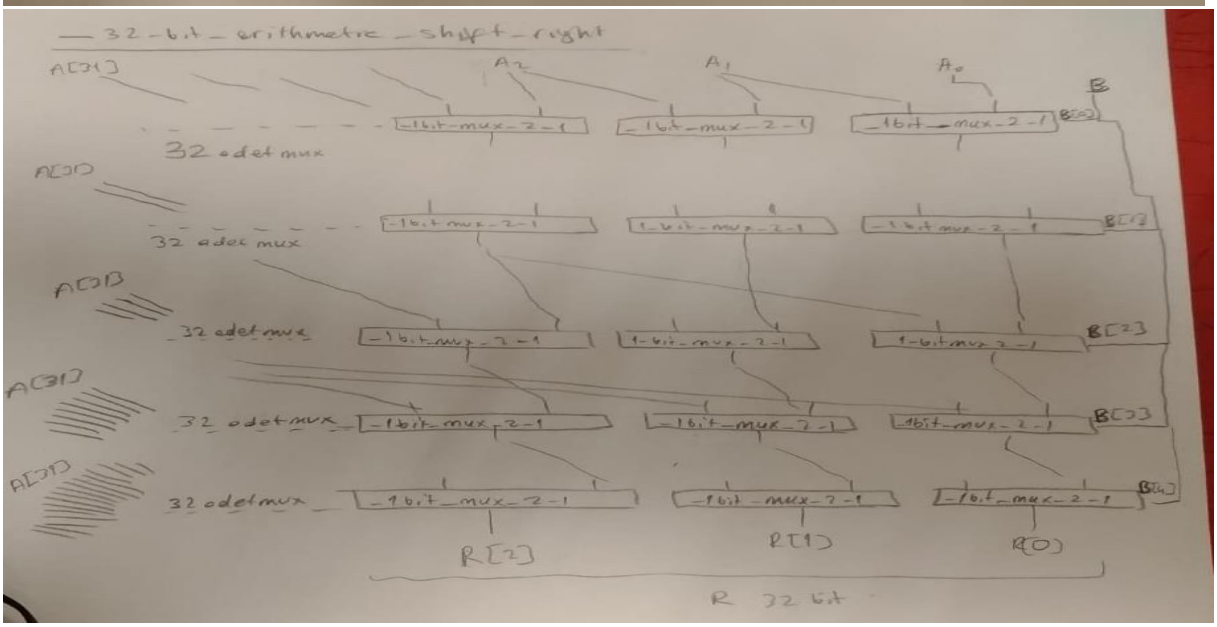

and32
or32
add32
XOR32
sub32
osr32
sl32
nor32
_32bit_MUX
8:1
MUX
R
S[2]
S[1]
S[0]
S

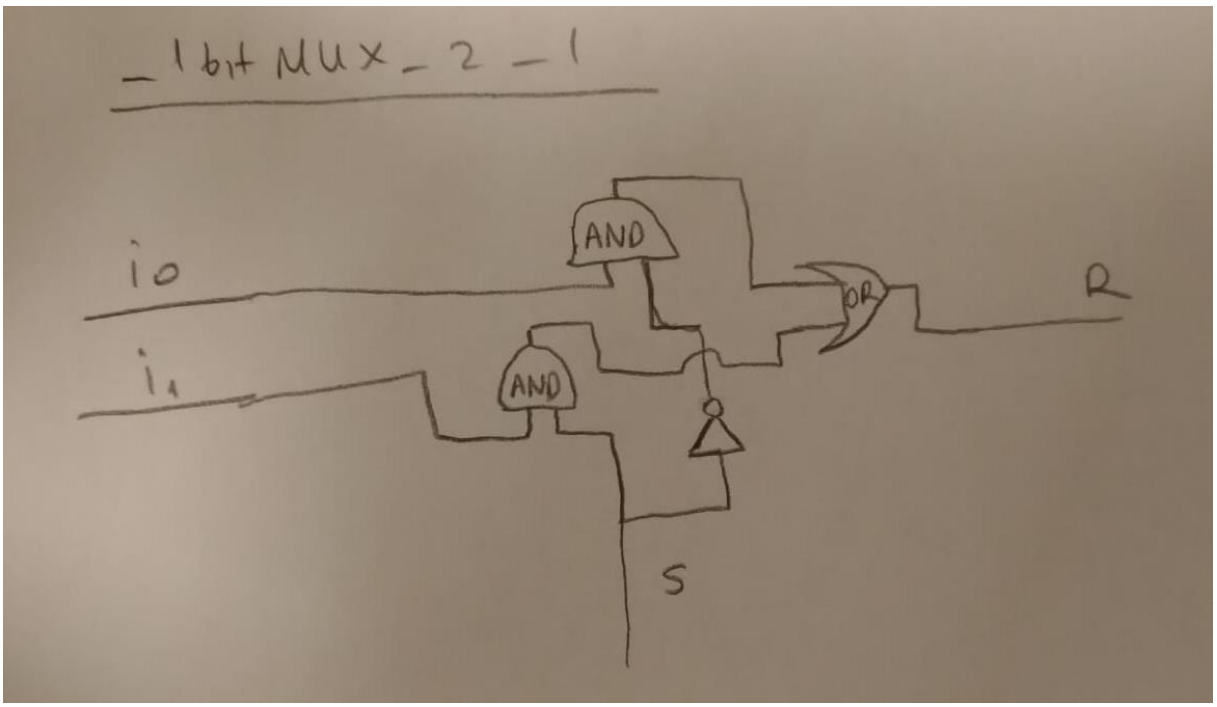

_32bit_MUX
R
or1
or2
_32bit_or_4input
_32bit_or_4input
out0
out1
out2
out3
out4
out5
out6
out7
i0
i1
32bit
i2
i3
i4
i5
i6
i7
S232
S132
S032
S232N
S132N
S032N
32bit
_make32bit
S2
S1
S0

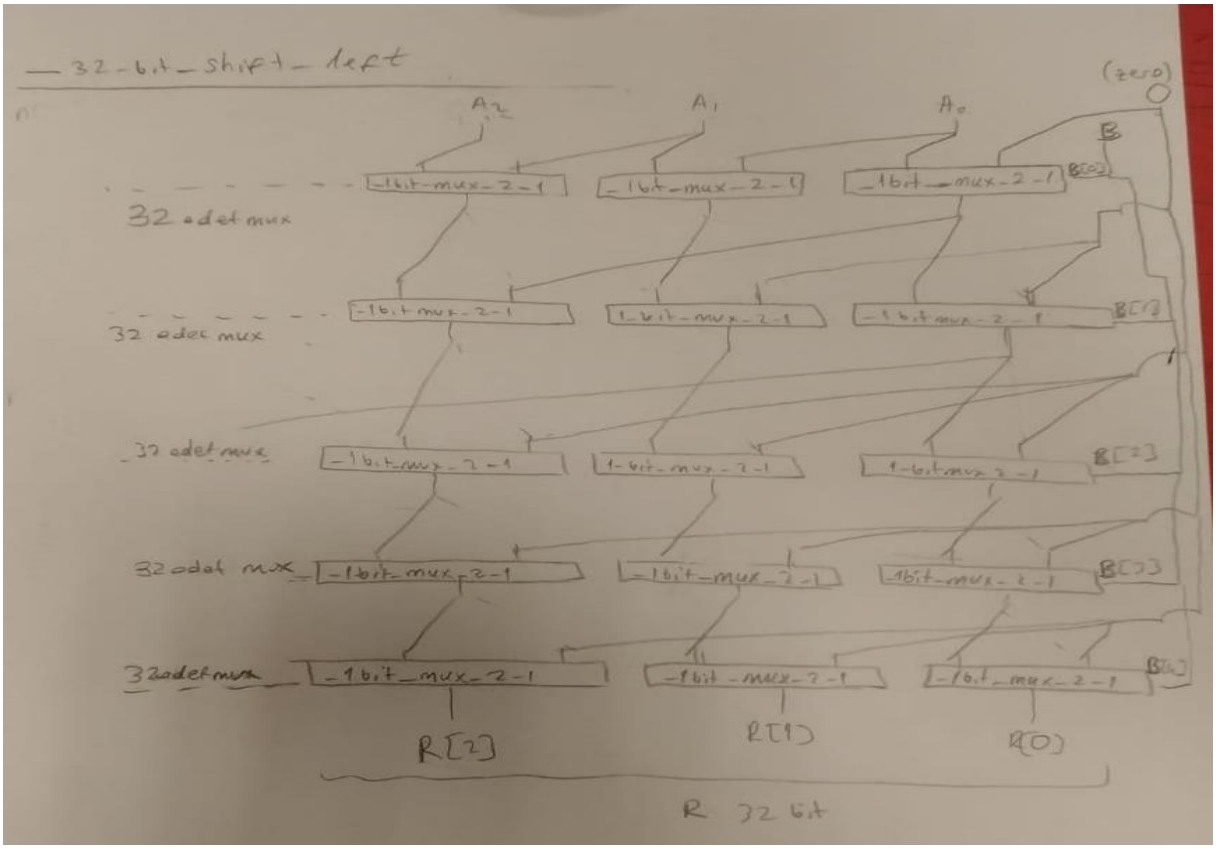

_make 32bit
A
B[0]
B
32bit
B[31]
S[0]
S[1]
S[2]
S[31]
32 AND Gate
32 Bit S


_32bit_and_4input
32 Bit S
S[0]
S[1]
S[2]
S[3]
32 odet AND Gate
AND
AND
AND
AND
A[0]
A[1]
A
B[0]
B[1]
B
C[0]
C[1]
C
D[0]
D[1]
D


32 bit OR 4input
32 bit S
S[0]
S[1]
S[2]
S[3]
OR
32 odet OR Gate
A
B
C
D
A[0]
A[1]
B[0]
B[1]
C[0]
C[1]
D[0]
D[1]


32 bit arithmetic shift right
32 odet mux
1 bit mux 2 1
R[2]
R[1]
R[0]
R 32 bit


1 bit MUX 2 1
AND
AND
OR
S
R



2. Verilog Modules and Their Description

1. **_32bit_and+**
   Bu modül ALU nun 000 select biti için kullanılmıştır.32 bitlik iki A ve B inputu alır ve bu inputların aynı digitteki değerlerini birbiriyle AND kapısına sokar ve çıkan sonucu yine 32 bitlik S outpunun yine aynı indexine yazarak S outputunu doldurur.
2. **_32bit_OR+**
   Bu modül ALU nun 001 select biti için kullanılmıştır.32 bitlik iki A ve B inputu alır ve bu inputların aynı digitteki değerlerini birbiriyle OR kapısına sokar ve çıkan sonucu yine 32 bitlik S outpunun yine aynı indexine yazarak S outputunu doldurur.
3. **half_adder++**
   Bu modül half add işlemi yapar. Full adderda kullanılmıştır. Carry in almaz. Gelen a ve b 1 bitlik inputlarını toplar ve carryOut ve işlem sonucunu output olarak çıkartır.
4. **full_adder++**
   Bu modül full add işlemi yapar. _32bit_adder modulünde kullanılmıştır .Carry in alır. Gelen a ve b 1 bitlik inputlarını toplar.Ardından carryOut ve işlem sonucunu output olarak çıkartır.

5. **_32bit_adder++**
   Bu modül ALU nun 010 select biti için kullanılmıştır.İki adet A ve B 32 bitlik lik biribiriyle toplanacak sayı ve birde carry in biti alır. Full adder 32 defa kullanılarak 32 bitlık A ve B sayıları toplanarak S outputuna atılır ve carry out olarak C outpuna atılır.
6. **_32bit_XOR++**
   Bu modül ALU nun 011 select biti için kullanılmıştır.32 bitlik iki A ve B inputu alır ve bu inputların aynı digitteki değerlerini birbiriyle XOR kapısına sokar ve çıkan sonucu yine 32 bitlik S outpunun yine aynı indexine yazarak S outputunu doldurur.
7. **_32bit_subtractor++**
   Bu modül ALU nun 100 select biti için kullanılmıştır.İki adet A ve B 32 bitlik lik A-B işlemini yapacak sayı ve birde carry in biti alır. Full adder 32 defa kullanılarak 32 bitlık (A) ve (B nin değili)sayıları toplanarak S outputuna atılır ve carry out olarak C outpuna atılır.
8. **_32bit_NOR++**
   Bu modül ALU nun 111 select biti için kullanılmıştır.32 bitlik iki A ve B inputu alır ve bu inputların aynı digitteki değerlerini birbiriyle NOR kapısına sokar ve çıkan sonucu yine 32 bitlik S outpunun yine aynı indexine yazarak S outputunu doldurur.
9. **alu32++**
   Bu modül ALU görevi görür.Top level entitydir. 8 farklı ALU işlemini de yapar ve 8 farklı wire a atar. Daha sonra bu wire ları 8:1 lik 32 bitlik mux a select biti ile gönderir ve istenen çıktının R outputuna yazılmasını sağlar.
10. **_32bit_MUX++**
    Bu modül 8:1 lik Mux görevi görür.alu32 modülünde kullanılmıştır.8 adet 32 bitlik input, 1 adet 3 bitlik select inputu ve bir adette 32 bitlik output vardır. Gelen select bite göre 8 adet inputtan birini output olarak dışarı verir.

11. **_make32bit++**
    Bu modül _32bit_MUX modulünde kullanılmıştır. 32 bit lik değer ile 1 yada 0 bitini AND kapısına sokmak için 1 yada 0 bitini 32 bit haline getirir. Yani bit 0 ise 32 tane 0 biti oluşturur. Bit 1 ise 32 tane 1 biti oluşturur ve S outputuna atar.
12. **_32bit_and_4_input++**
    Bu modül _32bit_MUX modulünde kullanılmıştır.32 Bitlik 4 tane değerin birbiri ile AND kapısına sokulmasını sağlar.Sonucu S outputuna atar.
13. **_32bit_or_4_input++**
    Bu modül _32bit_MUX modulünde kullanılmıştır.32 Bitlik 4 tane değerin birbiri ile OR kapısına sokulmasını sağlar.Sonucu S outputuna atar.
14. **_32bit_arithmetic_shift_right++**
    Bu modül ALU nun 101 select biti için kullanılmıştır.A ve B 32 bitlik input alır ve B nin least 5 significant biti kadar sağa kaydırır her kaydırmada sayının 32 digiti most significant bite yazılır ve sonuc S outputuna atılır.
15. **_1bit_MUX_2_1++**
    Bu modül ALU nun _32bit_arithmetic_shift_right modülü için kullanılmıştır.1 Bitlik iki adet input alır ve gelen bir bitlik select bitine göre bu iki bitten birini R outputuna atar.

16. _32bit_shift_left++

Bu modül ALU nun 110 select biti için kullanılmıştır.A ve B 32 bitlik input alır ve B nin least 5 significant biti kadar sola kaydırır her kaydırmada most significant bite 0 yazılır ve sonuc S outputuna atılır.

3. Modelsim Simulation Results

```
VSIM 4> step -current
# time =   0, A =11101111111111101111111111111111, B=00000000000000000000000000000000,S=000,R=00000000000000000000000000000000
# time =  20, A =11111111111111111111111110111111, B=11111101111111111111111111011111,S=000,R=11111101111111111111111110011111
# time =  40, A =00000000000000000000000000000011, B=00000000000000000000000000000101,S=000,R=00000000000000000000000000000001
# time =  60, A =11111111111111101111111111111110, B=00000000000010000000000000000001,S=001,R=11111111111111101111111111111111
# time =  80, A =11111111111111111111111111111110, B=00000100000000000000000000000011,S=001,R=11111111111111111111111111111111
# time = 100, A =00000000000000000000000000000111, B=00000000000000000000000000000001,S=001,R=00000000000000000000000000000111
# time = 120, A =00000000000000000000000100000011, B=00000000000000000000000000001011,S=010,R=00000000000000000000000100001110
# time = 140, A =00000000000000000000100000001101, B=00000000000000000000000000001111,S=010,R=00000000000000000000100000011100
# time = 160, A =00000000000000000000000000011011, B=00000000000000000000000001000001,S=010,R=00000000000000000000000001011100
# time = 180, A =00000100000000000000010000000111, B=00000000000000000000000010000011,S=011,R=00000100000000000000010010000100
# time = 200, A =00001000000000100000010000000011, B=00000000000010000000000000100001,S=011,R=00001000000010100000010000100010
# time = 220, A =00001000000000100000000000000011, B=00000000000010000000000000100001,S=011,R=00001000000010010000000000100010
# time = 240, A =00000000100000000000000001000011, B=00000000000000000000000010000101,S=100,R=00000000011111111111111110111110
# time = 260, A =00001000000000000000100000000011, B=00000000000000000000010000000001,S=100,R=00001000000000000000010000000010
# time = 280, A =00010000010000000000000000000011, B=00000000000000000000100000000001,S=100,R=00010000001111111111100000000010
# time = 300, A =11000000010011111111111111110000, B=00000000000000000010000000010111,S=101,R=11111111111111111111111110000000
# time = 320, A =11000010000011111111111111110000, B=00000000000000000000000100001101,S=101,R=11111111111111110000100000111111
# time = 340, A =11000000000011111111111111110100, B=00000000000000000000000000100011,S=101,R=11111000000000011111111111111110
# time = 360, A =11110011000011111110111111110000, B=00000000000000000000000000010101,S=110,R=11111110000000000000000000000000
# time = 380, A =11110111000011111111101111110010, B=00000000000000000000000000010111,S=110,R=11111001000000000000000000000000
# time = 400, A =11110011000011110111111111110000, B=00000000000000000000000000001101,S=110,R=11101111111111100000000000000000
# time = 420, A =00010000000100000010000000010011, B=01000000000001000000100000100001,S=111,R=10101111111010111101011111001100
# time = 440, A =00010001000100100010000010010011, B=01000001001000100000100000100011,S=111,R=10101110110011011101011101001100
# time = 460, A =00010000010001000001000000000011, B=00010000100000010000010001000001,S=111,R=11101111001110101110101110111100

VSIM 5>
```