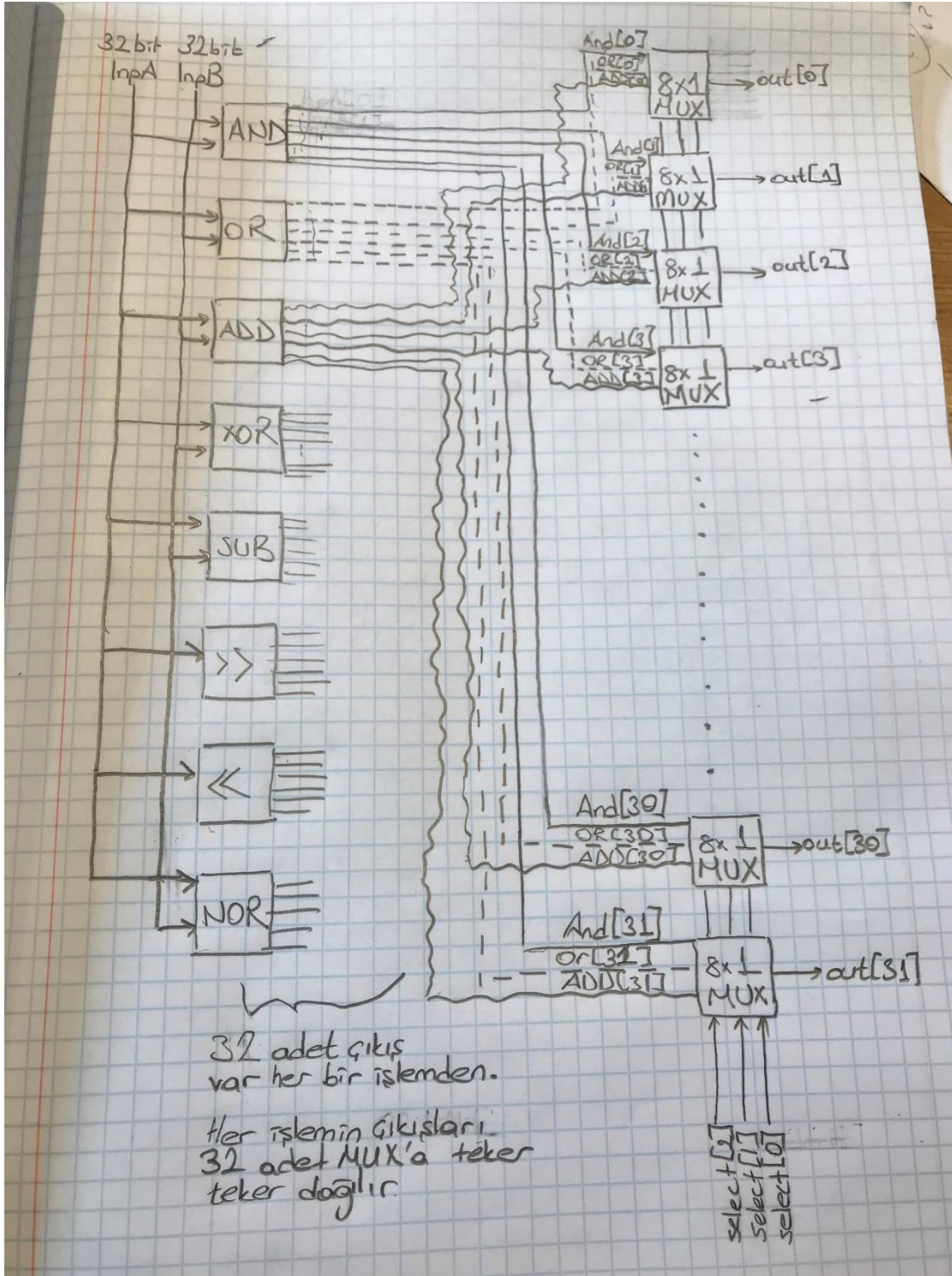


CSE 331 Computer Organization

Project 2 – ALU with Structural Verilog

Proje Ana Yapısı



Projede kullandığım yapı yukarıdaki şekilde gibidir.

And komutu için	<code>and_32bit(S,A,B);</code>	modülü
Or komutu için	<code>or_32bit(S, A, B);</code>	modülü
Add komutu için	<code>fulladder_32bit(sum, carry_out, input1, input2,carry_in);</code>	modülü
Xor komutu için	<code>xor_32bit(S,A,B);</code>	modülü
Sub komutu için	<code>fulladder_32bit(sum, carry_out, input1, input2,carry_in);</code>	modülü
>> komutu için	<code>sra_module(outfifth, inp, select1,select2,select3,select4,select5);</code>	modülü
<< komutu için	<code>sl_module(outfifth, inp, select1,select2,select3,select4,select5);</code>	modülü
Nor komutu için	<code>nor_32bit(S, A, B);</code>	modülünü tasarladım.

`and_32bit(S,A,B);` —————> 32bit A ve 32bit B inputlarını and işlemine sokar ve sonucu 32bit olan S outputunda gösterir.

`or_32bit(S, A, B);` —————> 32bit A ve 32bit B inputlarını or işlemine sokar ve sonucu 32bit olan S outputunda gösterir.

`fulladder_32bit(sum, carry_out, input1, input2,carry_in);` —————> carry_in inputuna 1b'0 verdiğimizde 32bit A ve 32bit B inputlarını toplar ve sonucu 32bit olan S outputunda gösterir.

`xor_32bit(S,A,B);` —————> 32bit A ve 32bit B inputlarını xor işlemine sokar ve sonucu 32bit olan S outputunda gösterir.

`fulladder_32bit(sum, carry_out, input1, input2,carry_in);` —————> carry_in inputuna 1b'1 verdiğimizde 32bit A ve 32bit B inputlarını çıkarır ve sonucu 32bit olan S outputunda gösterir.

`sra_module(outfifth, inp, select1,select2,select3,select4,select5);` —————> En fazla 32 bite kadar 32bitlik inputu sağa kaydırma işlemi yapar ve outfifth isimli outputta gösterir. Selectlere verdiğimiz değerler doğrultusunda farklı farklı kaydırmalar yapabiliriz. Bu selectlere değerleri alu32.v isimli modülde atanabiliyor ve kaydırma biti sayısına bu modülde karar veriliyor. Selectlere değer atama aşağıdaki şekilde gösterildiği gibidir.

```

module alu32(out,inp_A,inp_B,select);

input  [31:0] inp_A,inp_B;
input  [2:0] select;
output [31:0] out;

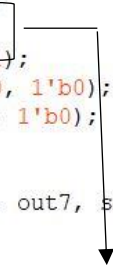
wire[31:0] out0, out1, out2, out3, out4, out5, out6, out7 ;
wire carry_out;

and_32bit      and32bit(out0, inp_A, inp_B);
or_32bit       or32bit (out1, inp_A, inp_B);
fulladder_32bit add32bit(out2, carry_out,inp_A, inp_B,1'b0);
xor_32bit      xor32bit(out3, inp_A, inp_B);
fulladder_32bit addsubfunc(out4, carry_out,inp_A, inp_B,1'b1);
sra_module     srafunc (out5, inp_A, 1'b0, 1'b1, 1'b0, 1'b0, 1'b0);
sl_module      srlfunc (out6, inp_A, 1'b0, 1'b1, 1'b0, 1'b0, 1'b0);
nor_32bit      norfunc (out7, inp_A, inp_B);

mux_32bit MUX1(out, out0, out1, out2, out3 ,out4, out5, out6, out7, select[2],select[1],select[0]);

endmodule

```



Select bitlerine sırası ile select0,select1,select2,select3,select4 değerleri verilir. Yani yukarıdaki gibi 01000 değerleri girildiğinde 2 basamak sağa kayar.

select0=1'b0

select1=1b'1

select2=1'b0

select3=1'b0

select4=1'b0 olduğunda 2 basamak sağa kayar.

sl_module(outfifth, inp, select1,select2,select3,select4,select5); —————> Bu modülde sra_moduldeki sağa kaydırma işleminin tam tersini yani sola kaydırma işleminin yapılması amaçlanmıştır.yani **Select bitlerine sırası ile select0,select1,select2,select3,select4 değerleri verilir. Yani yukarıdaki gibi 01000 değerleri girildiğinde 2 basamak sola kayar.**

select0=1'b0

select1=1b'1

select2=1'b0

select3=1'b0

select4=1'b0 olduğunda 2 basamak SOLA kayması gerekmektedir. Ancak hatalı sonuçlar gelmektedir.

And ,Or ,Add ,Xor ,Sub , >>, <<, Nor işlemlerinden çıkan 32 bitlik sonuçların her bir bitleri 32 tane tasarlanan her muxun bir girişine verilir. (En yukarıda proje tasarımı örneğinde verildiği gibi.)

ALU select (S)	Operation
000	R = A AND B
001	R = A OR B
010	R = A + B
011	R = A XOR B
100	R = A - B
101	R = A >> B (arithmetic shift right)
110	R = A << B (shift left)
111	R = A NOR B

Muxa şekildeki select bitleri verildiğinde o selectin gösterdiği modülün sonucu çıktı olarak output üretir.

Bu yapı alu32.v modülünde kurulmuştur. Aşağıda gösterilmiştir.

```
module alu32(out,inp_A,inp_B,select);

input  [31:0] inp_A,inp_B;
input  [2:0] select;
output [31:0] out;

wire[31:0] out0, out1, out2, out3, out4, out5, out6, out7 ;
wire carry_out;

and_32bit      and32bit(out0, inp_A, inp_B);
or_32bit       or32bit (out1, inp_A, inp_B);
fulladder_32bit add32bit(out2, carry_out,inp_A, inp_B,1'b0);
xor_32bit      xor32bit(out3, inp_A, inp_B);
fulladder_32bit addsubfunc(out4, carry_out,inp_A, inp_B,1'b1);
sra_module     srafunc (out5, inp_A, 1'b0, 1'b1, 1'b0, 1'b0, 1'b0);
sl_module      srlfunc (out6, inp_A, 1'b0, 1'b1, 1'b0, 1'b0, 1'b0);
nor_32bit      norfunc (out7, inp_A, inp_B);

mux_32bit MUX1(out, out0, out1, out2, out3 ,out4, out5, out6, out7, select[2],select[1],select[0]);

endmodule
```

000 için And_32bit çıktısını outa verir.

001 için Or_32bit çıktısını outa verir.

010 için Fulladder_32bit (toplama)çıktısını outa verir.

011 için Xor_32bit çıktısını outa verir.

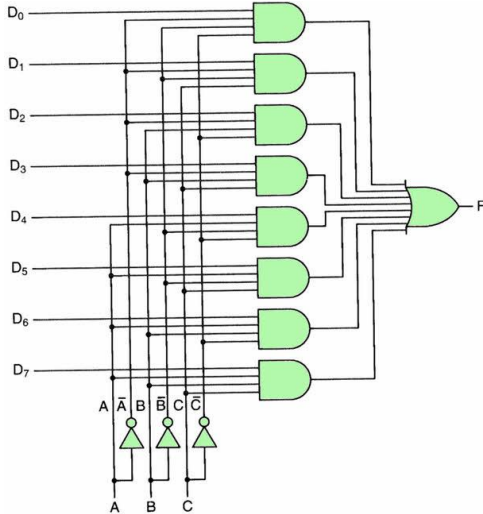
100 için Fulladder_32bit çıktısını outa verir.

101 için Sra_module çıktısını outa verir.

110 için Sl_module çıktısını outa verir.

111 için Nor_32bit çıktısını outa verir.

8*1 mux aşağıdaki mux yapısı ile tasarlanmıştır.



- Projemde her modüle karşılık bir testbench yazılmıştır. Top modülüm alu32 modülüdür. Ve alu32_testbench sonuçları sl(sola kaydırma modülü haricinde) doğru sonuçlar vermektedir ve aşağıdaki gibidir. Her select için 3 farklı sonuç döndürmesi için 3 farklı input verilmiştir.

```
Library  sim
Transcript
add wave -position insertpoint \
sim:/alu32_testbench/inp_A \
sim:/alu32_testbench/inp_B \
sim:/alu32_testbench/select \
sim:/alu32_testbench/out
VSIM 5> step -current
# time = 0, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=000, out=00000000000000000000000000000000
# time = 20, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=000, out=11111111111111111111111111111111
# time = 40, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=000, out=00000000000000000000000000000000
# time = 60, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=001, out=11111111111111111111111111111111
# time = 80, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=001, out=11111111111111111111111111111111
# time = 100, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=001, out=00000000000000000000000000000000
# time = 120, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=010, out=11111111111111111111111111111111
# time = 140, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=010, out=11111111111111111111111111111110
# time = 160, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=010, out=00000000000000000000000000000000
# time = 180, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=011, out=11111111111111111111111111111111
# time = 200, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=011, out=00000000000000000000000000000000
# time = 220, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=011, out=00000000000000000000000000000000
# time = 240, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=100, out=00000000000000000000000000000001
# time = 260, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=100, out=00000000000000000000000000000000
# time = 280, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=100, out=00000000000000000000000000000000
# time = 300, inp_A =11111111111111111111111111111111, inp_B=00000000000000000000000000000000, select=101, out=00000000000000000000000000000000
# time = 320, inp_A =00000000000000000000000000000011, inp_B=00000000000000000000000000000000, select=101, out=00000000000000000000000000000001
# time = 340, inp_A =11111111111111111111111111111110, inp_B=00000000000000000000000000000000, select=101, out=11111111111111111111111111111111
# time = 360, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=110, out=00000000000000000000000000000000
# time = 380, inp_A =11111111111111111111111111111111, inp_B=00000000000000000000000000000000, select=110, out=11111111111111111111111111111111
# time = 400, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=110, out=00000000000000000000000000000000
# time = 420, inp_A =00000000000000000000000000000000, inp_B=11111111111111111111111111111111, select=111, out=00000000000000000000000000000000
# time = 440, inp_A =11111111111111111111111111111111, inp_B=11111111111111111111111111111111, select=111, out=00000000000000000000000000000000
# time = 460, inp_A =00000000000000000000000000000000, inp_B=00000000000000000000000000000000, select=111, out=11111111111111111111111111111111
```