

NAME : MUAZ KURT

NO: 151044062

ALU modülü base of the Project olarak kabul edilmiştir.

Girdi olarak 32 bitlik iki register ve 3 bitlik ALU_select alır.

Proje için Full adder tanımlanmıştır. Full adder, çıkarma işlemi, and, xor ve or değerlerinin bulunmasına olanak sağlar. Bunun yanında eğer overflow olursa bunun tespiti de sağlanabilir.

Adder, left ve right olarak 32 bitlik girdiler alır, 1 bitlik select seçeneği, $((ALU_2) * !(ALU_1) * !(ALU_0))$ işleminin sonucuna (çıkarma seçeneği) bağlıdır. Select, Left'in 32 biti ile XOR'lanarak Left_2 değerini oluşturur. Aynı zamanda select 0. bit için yapılacak toplama işlemine girecek olan Carriage_In'e de bağlanacaktır. Bu sayede çıkarma işlemi için yapılması gereken 2's compliment alma operasyonu sağlanacaktır.

Adder için yazılan testbench içinde overflow tespiti denenmiş ve başarılmıştır. Ancak istenen sistem içinde bunun çıktısını göstermek için bir select olmadığı için gereken bit ALU seviyesine çıkarılmamıştır.

Adder'den alınan OR değeri, 32 bitlik not modülüne sokularak, NOR elde edilmiştir.

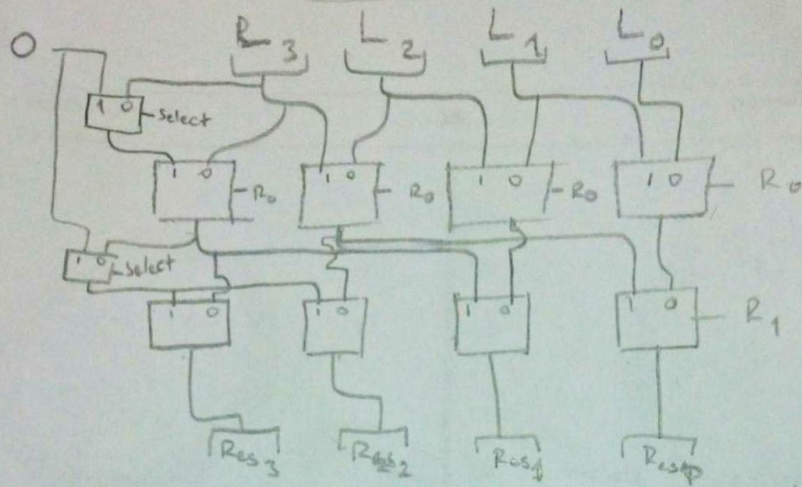
Right Shift için, 5 seviye 2x1 lik muxlar kullanılmıştır. MSB için, aritmetic/logic shift seçeneği bir bit olarak verilmiş ve logic shift olduğu zaman MSB 0 olacak, aritmetic shift için MSB aynı kalmaya devam edecek, kaydığı yerler de MSB olacaktır. Right shift için, yapılacak shift miktarını belirten, ikinci girdinin [31:5] bitleri içinde en az 1 tane 1 olması durumunda, sonuç 2 kere 16 bit sağa kaydırılmış olarak sonulur. Logic veya Aritmetic olmasına göre, tüm bit dizisi 0 veya 1 ler ile doldurulur.

Görselde taslak olarak 4 bit için uyarlanmış hali mevcuttur.

Left Shift için, Input değeri, 32 bitlik buffer yardımı ile [31:0] tipinden [0:31] e çevrilmiştir. Ters çevrilen değer, Right Shift modülüne, logic shift biti ile verilir. Çıktısı alınan değer, tekrar 32 bitlik buffer yardımı ile [31:0] şekline çevrilir. Böylelikle Left Shift gerçekleşmiş olur.

Son olarak elde edilen 8 adet 32 bitlik sonuç, 32 adet 8x1 mux'a ALU bitleri ile sokularak, 32 bitlik result oluşturulur ve işlem tamamlanır.

Right / Left Shift



$L_3 \ L_2 \ L_1 \ L_0$

actually 5 bits
(this is for explain)

$R_1 \ R_0$ Select

Arithmetic
or logic
Shift

Actually 32 bits.
for left shift, it is
reverse order.
for an

$L_0 \ L_1 \ L_2 \ L_3$

and the select
is going to 1.
So that left shift
will be done by
reverse right = logic shift
and another reverse operation.

Right[31:0] Left[31:0] ALU (A)

$R_{(i)}$	$L_{(i)}$	$C_{in(i)}$	$R_{res(i)}$	$C_{in(i+1)}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A_2	A_1	A_0	
1	1	1	
1	1	1	
1	1	1	
0	1	0	Add
1	1	1	
1	0	0	Sub

for this

Left's 32 bit
should be reversed
(xor with 1)
and for example: to it.

010100
- 110000

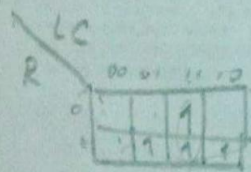
010100
+ 001100

100000
110000
011111
001111

100100
↑
came from
Carriage in

$$R_{(i)} = \bar{R} \bar{L} C + \bar{R} L \bar{C} + R \bar{L} \bar{C} + R L C$$

On add

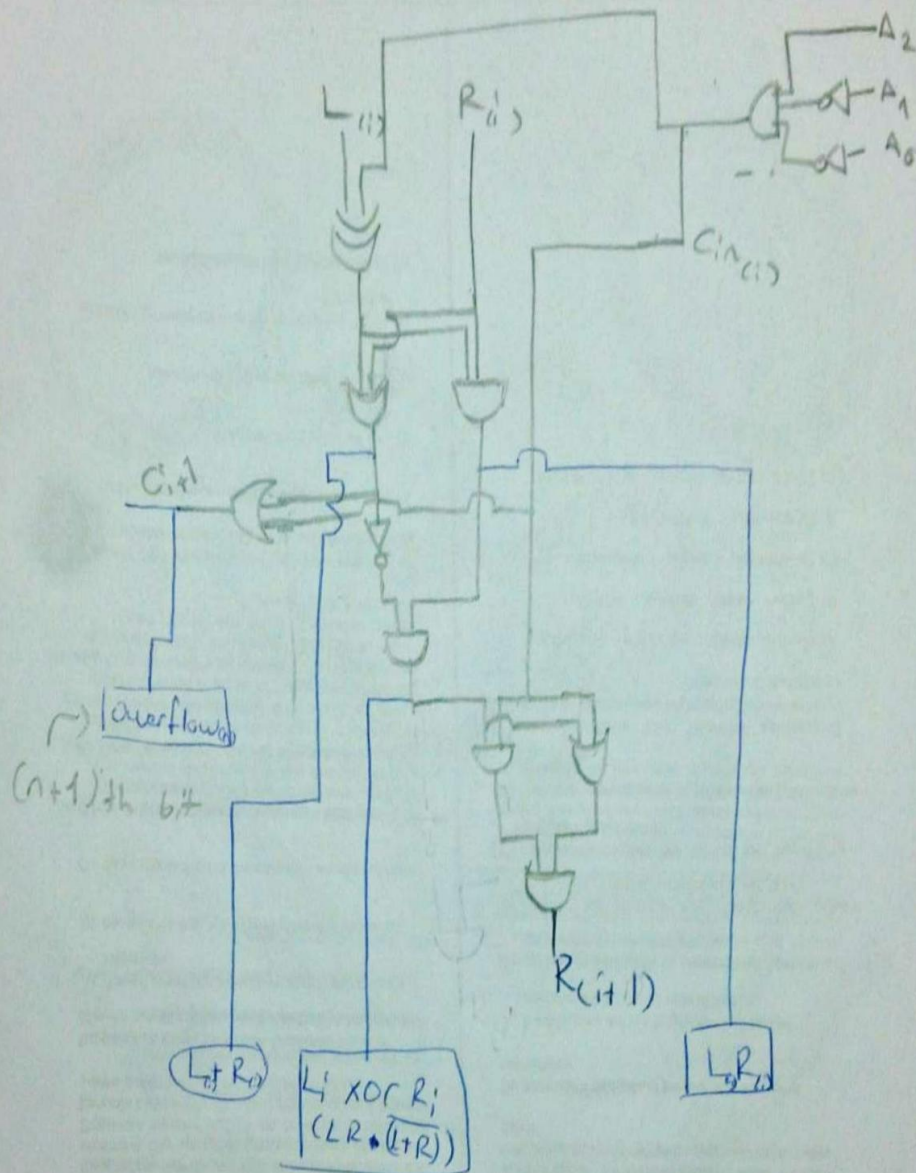


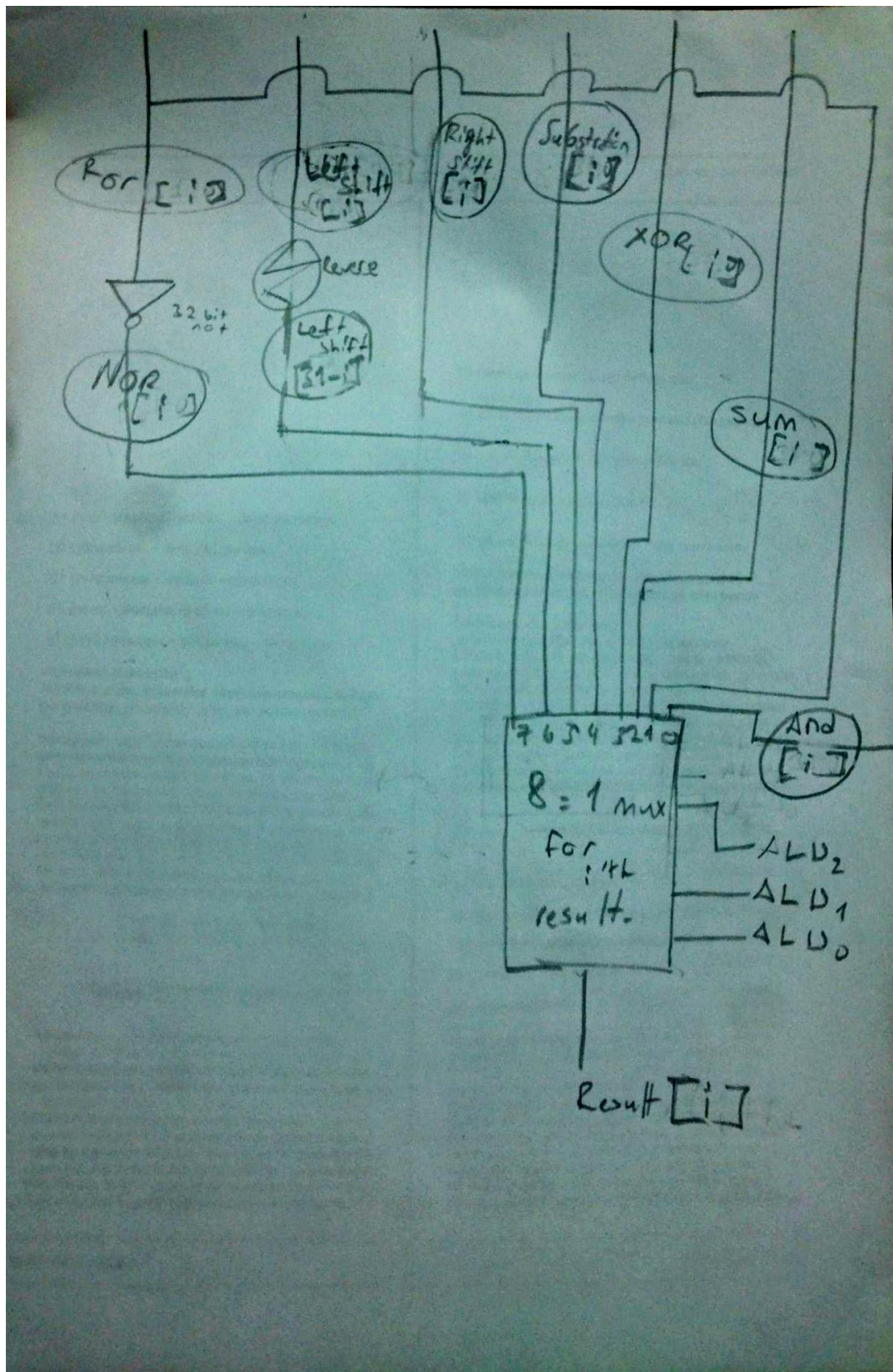
$$C_{(i+1)} = R C + R L + L C$$

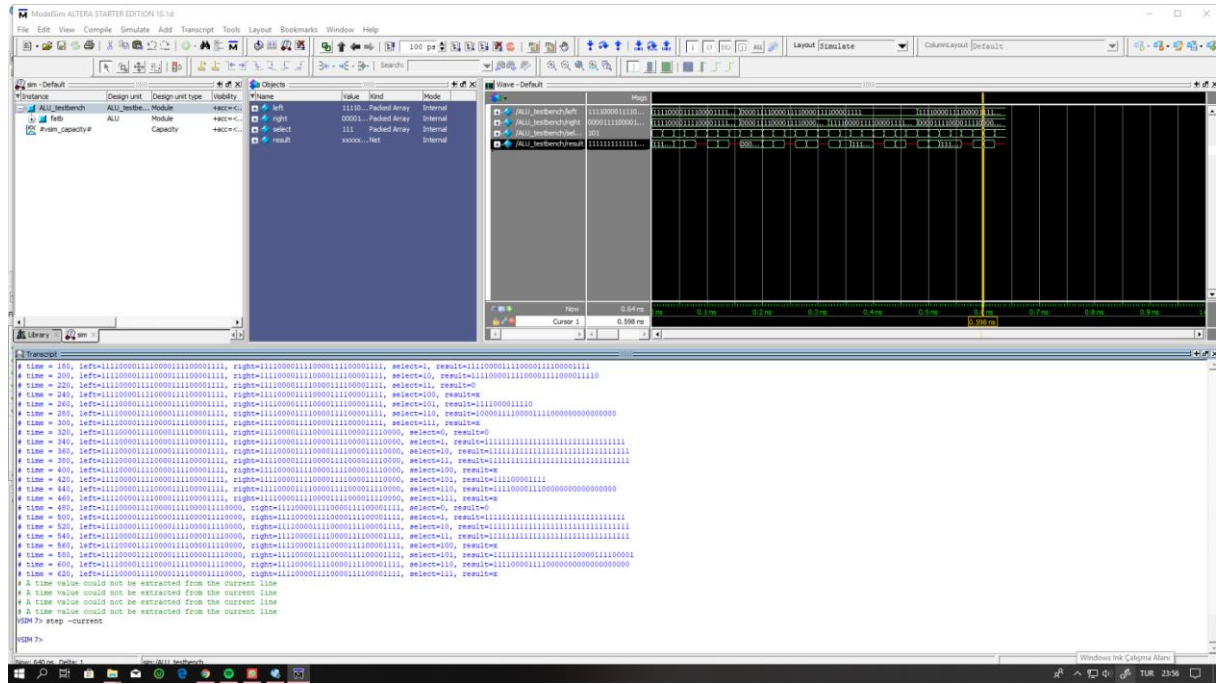
$$C_{(i+1)} = C (R + L) + R L$$

ALU's "100" input declares

that, operation is going to be subtraction







NOT: ALU test sırasında, result birkaç durumda hata oluşturuyor, çözemedim neyden kaynaklandığını.