گزارش تكليف سوم كامپيوترى مدار منطقى البرز محموديان 810101514

سوال 1: ابتدا کدی که برای مدار behavorial ALU زدیم را به yosys می دهیم تا ان را برای ما synthesise کند

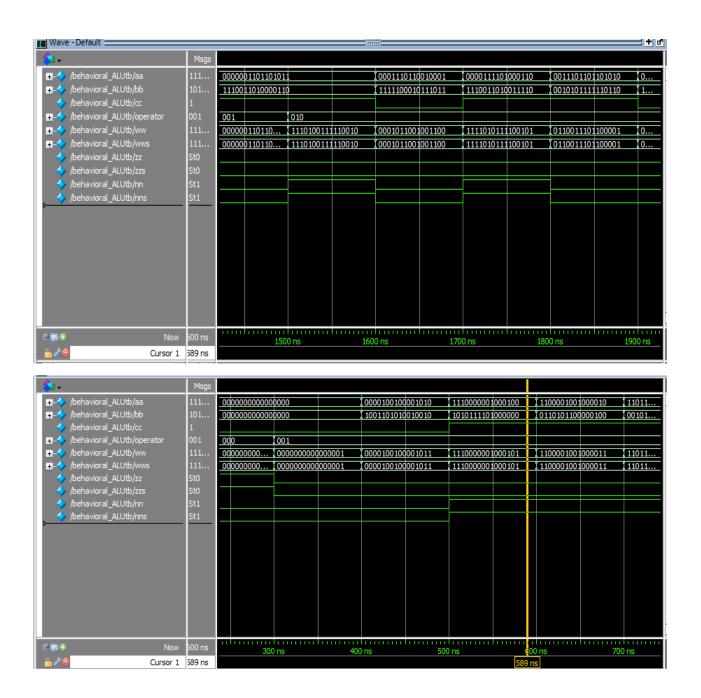
در netlist زیر مشاهده می کنیم که کد ما را توسط gate 480 طراحی کرده است.

```
C:\Users\Mahmodiyan-PC\Desktop\assignment3\yosys.exe
                                                                                                                                                                                           X
Top module: \behavioral_ALU
Removed 0 unused modules.
2.23. Printing statistics.
  == behavioral ALU ===
    Number of wires:
Number of wire bits:
Number of public wires:
Number of public wire bits:
     Number of memories:
                                                          0
0
480
59
62
10
1
40
78
64
37
     Number of memory bits:
     Number of processes:
Number of cells:
        $_AND_
$_AOI3_
        $_A0I4_
        $_MUX_
$_NAND
        $_NOR_
$_NOT_
        $_0AI3
$_0AI4
                                                           29
82
        $ OR
        $_XNOR_
        $ XOR

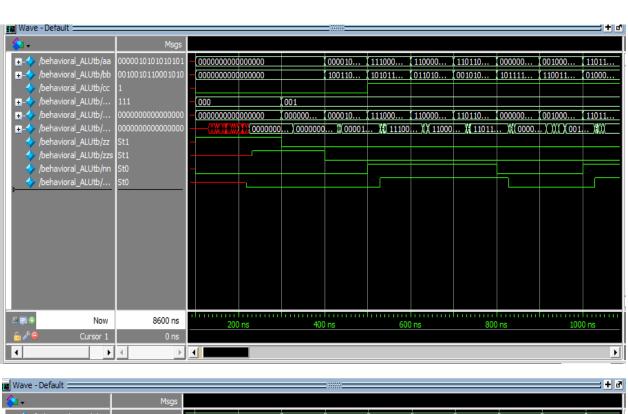
    2.24. Executing CHECK pass (checking for obvious problems).
    checking module behavioral ALU..
```

```
ABC: + write_blif <abc-temp-dir>/output.blif
4.1.2. Re-integrating ABC results.
ABC RESULTS:
                          NAND cells:
                                           208
ABC RESULTS:
                          NOR cells:
                                           336
ABC RESULTS:
                           NOT cells:
                                           112
ABC RESULTS:
                    internal signals:
                                           463
ABC RESULTS:
                       input signals:
                                            36
ABC RESULTS:
                      output signals:
                                            17
Removing temp directory.
yosys>
```

حالا برای مقایسه ی wave های ALU ای که خودمان طراحی کردیم و کد synthesise شده ی yosys در فایل mycells تاخیر های گیت های not, or, and را بر میداریم تا بتوانیم دو موج را با هم مقایسه کنیم که در عکس های زیر مشاهده می کنیم w ها کاملا با هم یکسان هستند و همچنین zero flag, negative flag نیز کاملا با هم در دو مدار هماهنگ هستند:



با در نظر گرفتن delayها در mycells می بینیم که waveها کامل روی هم نمی افتند و تغییرات delay ها روی wave دوم قابل مشاهده است که همین سبب می شود چون گاهی بیت ها با تاخیر تغییر می کنند خروجی نیزاشتباه شود که در دو عکس زیر این مسعله را مشاهده می کنیم:





سوال 2: ابتدا کدی که برای مدار structural ALU زدیم را به yosys می دهیم تا ان را برای ما synthesise کند که در انتهای گزارش توضیح داده شده که چگونه از behavorial ALU به behavorial ALU رسیدیم.

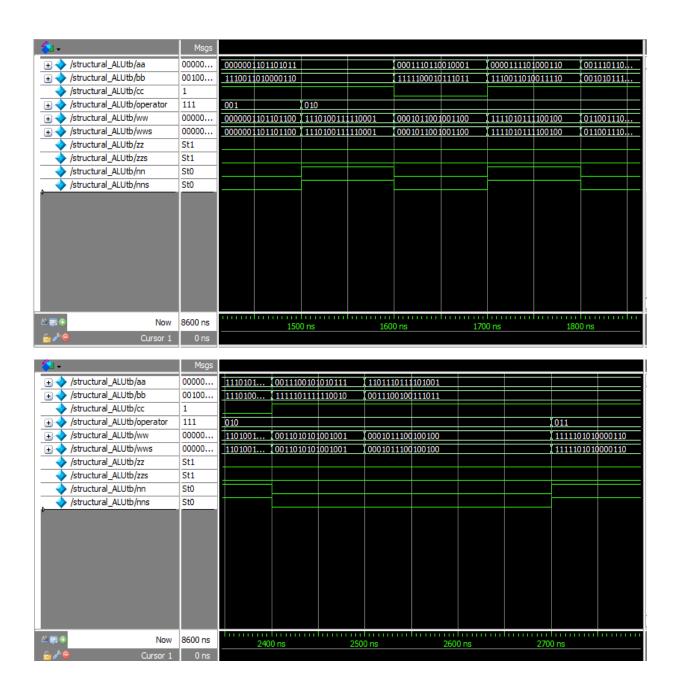
در netlist زیر مشاهده می کنیم که کد ما را توسط 274 gate طراحی کرده است. که بسیار بهتر از حالت قبل (سوال 1) است که این کار را با 480 گیت انجام دادیم.

و توانستین با یک طراحی بهینه شده از behavorial ALU به structural ALU برسیم و در نهایت باعث شد که تعداد گیت های استفاده شده در مدار را کاهش دهد که این مسعله را می توانیم در دو عکس زیر که کتابخانه های yosys و mycells به ما داده اند مشاهده کنیم:

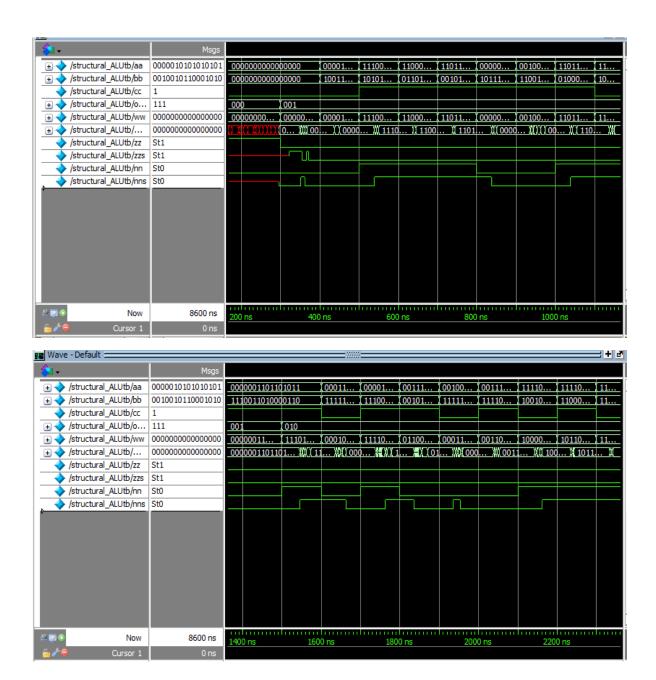
```
C:\Users\Mahmodiyan-PC\Desktop\assignment3\yosys.exe
                                                                                                                                                     2.23. Printing statistics.
 == structural_ALU ===
   Number of wires:
Number of wire bits:
Number of public wires:
                                               264
    Number of public wire bits:
    Number of memories:
    Number of memory bits:
    Number of processes:
Number of cells:
                                               274
                                                38
       $_AND_
       $_A0I3_
       $_MUX_
                                                1
20
32
27
36
       $ NAND
       $ NOR
       $ NOT
       $ OAI3
       $_OAI4
        XNOR
2.24. Executing CHECK pass (checking for obvious problems).
checking module structural_ALU...
     ning: found logic loop in module structural_ALU:
cell $abc$698$auto$blifp<u>arse.cc:286:parse_blif$722 (</u>$_NOT_)
```

```
ABC: + strash
ABC: + dch -f
ABC: + map
ABC: + write_blif <abc-temp-dir>/output.blif
4.1.2. Re-integrating ABC results.
ABC RESULTS:
                          NAND cells:
ABC RESULTS:
                          NOR cells:
                                           209
ABC RESULTS:
                          NOT cells:
                                            82
ABC RESULTS:
                                           256
                    internal signals:
ABC RESULTS:
                      input signals:
                                            48
ABC RESULTS:
                      output signals:
                                            18
Removing temp directory.
yosys>
```

حالا برای مقایسه ی wave های structural ALU ای که خودمان طراحی کردیم و کد yosys شده ی yosys در فایل mycells تاخیر های گیت های not, or, and را بر میداریم تا بتوانیم دو موج را با هم مقایسه کنیم که در عکس های زیر مشاهده می کنیم w ها کاملا با هم یکسان هستند و همچنین zero flag, negative flag نیز کاملا با هم در دو مدار هماهنگ هستند:



با در نظر گرفتن delayها در mycells می بینیم که waveها کامل روی هم نمی افتند و تغییرات delay ها روی wave دوم قابل مشاهده است که همین سبب می شود چون گاهی بیت ها با تاخیر تغییر می کنند خروجی نیزاشتباه شود که در دو عکس زیر این مسعله را مشاهده می کنیم:



برای بخش آخر می خواهیم که دو مدار behavorial و structural را با هم مقایسه کنیم که ایا عملکرد یکسانی با وجود ساختار متفاوت دارند یا خیر که برای اینکار موج های این دو ساختار را با هم مقایسه می کنیم که در عکس های زیر مشاهده می کنیم که این دو ساختار متفاوت عملا عملکرد یکسانی را دارند با این وجود که مدار دوم بسیار بهینه تر شده است :

Structural_behavioral_ALUtb/bb 00100 00100 10100 1010011000011110 01011 00110 01011 0101	wave belaute											
Structural_behavioral_ALUtb/bb	≨ 1 +	Msgs										
// /structural_behavioral_ALUtb/oper 1	-/ /structural_behavioral_ALUtb/aa	00000		00001	00000	11110	11001111	00011000		00111	00100	00010
111		00100		00100	10110	10000	10100110	00011110		01011	00110	01011
		1										
// /structural_behavioral_ALUtb/wws // /structural_behavioral_ALUtb/zz // /structural_behavioral_ALUtb/nn // /structural_		111	101						110			
// /structural_behavioral_ALUtb/zzs	_ ,			00101	10110	11110	11101111	00011110	00011	10110	10010	01100
/structural_behavioral_ALUtb/nn St0 St1 /structural_behavioral_ALUtb/nns St0 St0 Structural_behavioral_ALUtb/nns St0 St0 St0 Structural_behavioral_ALUtb/nns St0 St0 St0 St0 Structural_behavioral_ALUtb/nns St0	_ ·			00101	10110	11110	11101111	00011110	00011	10110	10010	01100
/structural_behavioral_ALUtb/nns St0												
/structural_behavioral_ALUtb/nns St0	· · · · · · · · · · · · · · · · · · ·											
									<u> </u>			
N= 0000 == 1	// /structural_behavioral_ALUtb/nns	St0							<u> </u>			
N= 0.000 ==												
N= 0.000 ==												
N= 0.000 ==												
N= 0000 == 1												
N= 0.000 ==												
N= 0.000 ==												
N= 0.000 ==												
N= (0C00 == 1												
N= 0000 == 1												
	Now	8600 ns			600	Ons	620	Ons	640	Ons	660	Ons
⊕ ∤ ⊖ Cursor 1 0 ns	6 ∕ 6 Cursor 1	0 ns		0.10	000	- 1.0	020	U-110	010	0.110		U-110

^											_
≨1 •	Msgs	L									
	00000		00000000	00000000		00001	11100	11000	11011	00000	00100
≖ - 分 /structural_behavioral_ALUtb/bb	00100	_	00000000	00000000		10011	10101	01101	00101	10111	11001
/structural_behavioral_ALUtb/cc	1	_									
≖ - <pre>/structural_behavioral_ALUtb/oper</pre>	111	_	000		001						
≖ - ∜ /structural_behavioral_ALUtb/ww	00000	_	00000000	00000000	00000	00001	11100	11000	11011	00000	00100
≖ - ∜ /structural_behavioral_ALUtb/wws	00000		00000000	00000000	00000	00001	11100	11000	11011	00000	00100
/structural_behavioral_ALUtb/zz	St1										
/structural_behavioral_ALUtb/zzs	St1										
/structural_behavioral_ALUtb/nn	St0	_									
/structural_behavioral_ALUtb/nns	St0	-									
<u></u>											
△ ■ Now	8600 ns	111									liiiiiiiiiiii
			200) ns	400) ns	600) ns	800) ns	1000 r
6 ✓ Cursor 1	0 ns										

^											
₩ •	Msgs										
≖ - / /structural_behavioral_ALUtb/aa	00000	0000001	101101	00011	00001	00111	00100	00111	11110	11110	111
	00100	1110011	010000	11111	11100	00101	11111	11110	10010	11000	111
/structural_behavioral_ALUtb/cc	1										
∓ - ∜ /structural_behavioral_ALUtb/oper	111	001	010								
∓ - ∜ /structural_behavioral_ALUtb/ww	00000	000	11101	00010	11110	01100	00011	00110	10000	10110	110
→ /structural_behavioral_ALUtb/wws	00000	000	11101	00010	11110	01100	00011	00110	10000	10110	110
/structural_behavioral_ALUtb/zz	St1										
/structural_behavioral_ALUtb/zzs	St1										
/structural_behavioral_ALUtb/nn	St0										
/structural_behavioral_ALUtb/nns	St0										
⊢ '											
Now	8600 ns										1111111
			160	0 ns	180	0 ns	200	0 ns	220	0 ns	
m № Cursor 1	0 ns										

سپس در اخر کار ساختاری را که روی کاغذ برای مدار های behavorial ALU و structural ALU کشیدیم را نشان می دهیم تا بفهمیم که چگونه از ساختار اول به ساختار دوم که بهینه تر بود رسیدیم:

