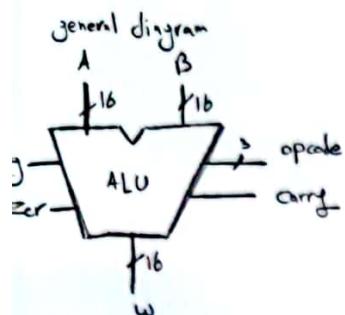


۱) مایلیک AN علیق نماید ۲ در دویس ۱۶-bit باشد (A و B) در هنر این را در دویس کی درست *carry* نامید و توسط ۴-bit *opcode* اعلام کنند خواهیم گفت در خودی مایلیک ۱۶-bit (S) دایم در دار آن ۲ گفالت *zer* و *neg* و دو مم دایم *neg* سه مردمت نیزی ممیزی متفق بود این تعداد (۱۵) عدد برای اسناد در غیر این خصوصیات است *zer* سه مردمت دارد و هر یکی با صفتی در آنها اینگاهی نه اینجا نه اینجا



opcode → this is a multiplexer to choose output

$$000 \quad w = -A + 1 \quad \longrightarrow \quad A \xrightarrow{16} \boxed{\text{Adder}} \xrightarrow{16} w$$

$$001 \quad w = A + 1 \quad \rightarrow \quad A \xrightarrow{16} \boxed{\text{Abber}} \xrightarrow{16} w$$

$$w = A + B + C \quad \rightarrow \quad \begin{array}{c} C \\ | \\ \text{Adder} \\ | \\ A \quad B \\ | \\ w \end{array}$$

$$\textcircled{11} \quad \omega = A + (\beta x, \alpha) \xrightarrow{\downarrow \beta \gg 1} \begin{array}{c} \beta \\ \hline 16 \end{array} \xrightarrow{\gg} \begin{array}{c} \beta \\ \hline 16 \end{array} \rightarrow \begin{array}{c} \beta \\ \hline 16 \end{array} \xrightarrow{\gg} \begin{array}{c} \beta \\ \hline 16 \end{array} \rightarrow \begin{array}{c} \text{Adder Full} \\ \hline 16 \end{array}$$

$$100 \quad w = A \& B \text{ (Bitwise)} \longrightarrow \begin{array}{c} A \\ \frac{1}{\mid} \\ \frac{+}{t} \\ \frac{B}{\mid} \\ \frac{16}{\mid} \end{array} \boxed{\text{Bitwise and}} \quad \begin{array}{c} w \\ \mid \\ 16 \end{array}$$

$$|0\rangle \quad \omega = A|B \rangle \text{ (Bitwise) } \rightarrow \begin{array}{c} +16 \\ \text{Bitwise} \\ \text{CR} \\ -16 \end{array} \quad \omega$$

$$110 \quad w = \{A[7:0], B[7:0]\} \rightarrow \begin{array}{c} A \\ \downarrow \\ B \\ \uparrow \\ \end{array} \xrightarrow{\text{merge}} \boxed{\text{merge}} \xrightarrow{\text{w}} \begin{array}{c} 8 \\ \downarrow \\ 8 \\ \uparrow \\ 16 \\ \downarrow \\ w \end{array}$$

۱-۲) پنجهای مت random را بین A و B و فرو انتهای زیر را درین میان بینایت و رسم داشته باشیم داریم

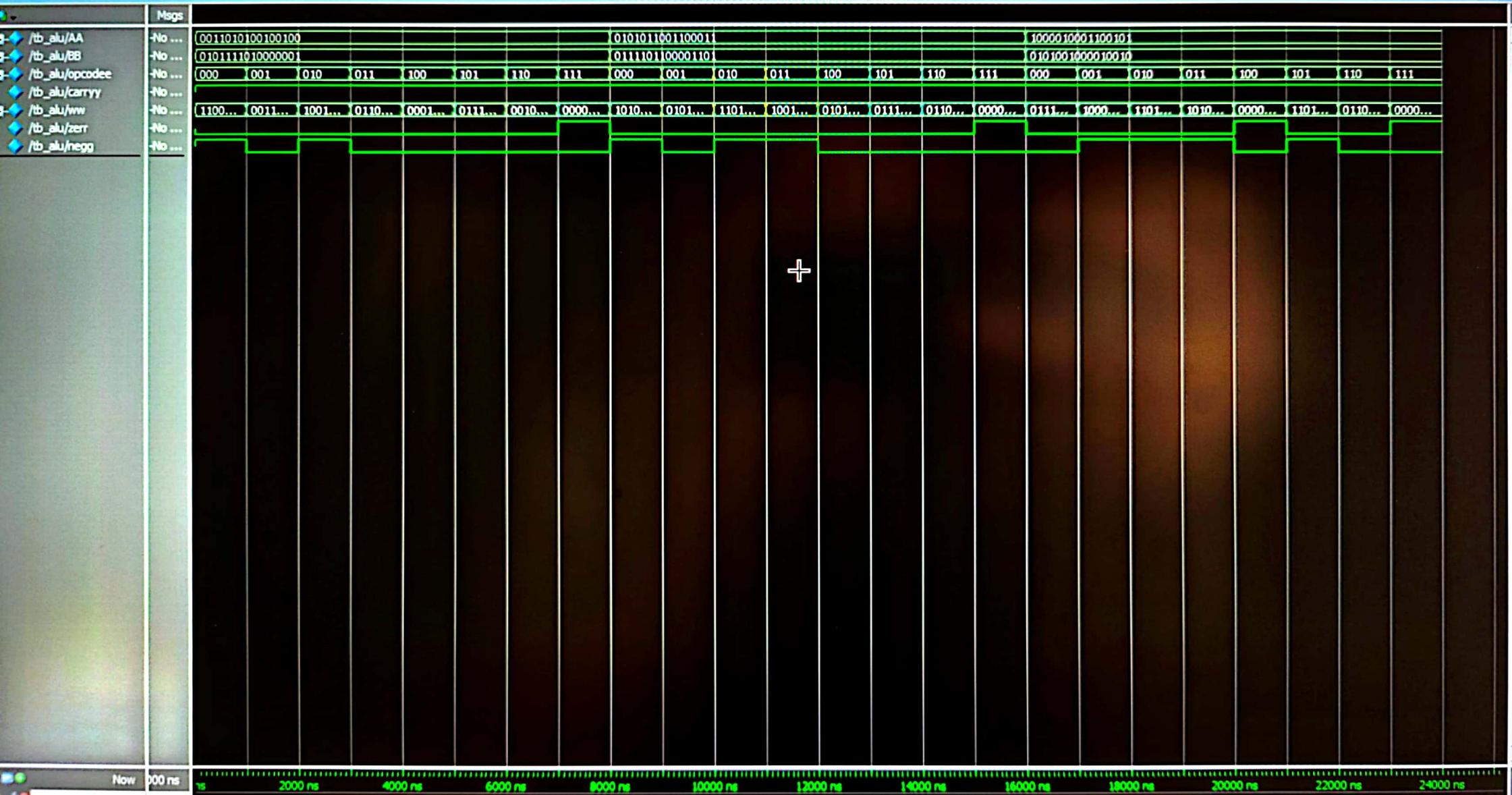
b) می توانیم آن را به دلخواه اینستیتیوی *synthesize* کنیم، با اینکه آن را در اینجا آن نیز نمایم و مارک آن را *synthesize* کنیم بداریم
مانند *library* هدایت کنید. بعد از آن بعزم تابا انتخاب از آن ALU را تسطیح NOR، NAND، Not بل و در آنرا مزوب کنیم و سپس
test bench ساخته کنیم امتحان فرود

020.1

Wave Tools Layout Bookmarks Window Help



Wave - Default



Project × alu_tb.v × alu.v × sim × Wave

Now: 24 us Delta: 1

sim/tb.ski

2.21.5. Finished fast OPT passes.

2.22. Executing HIERARCHY pass (managing design hierarchy).

2.22.1. Analyzing design hierarchy..

Top module: \ALU

2.22.2. Analyzing design hierarchy..

Top module: \ALU

Removed 0 unused modules.

2.23. Printing statistics.

==== ALU ===

Number of wires:	473
Number of wire bits:	520
Number of public wires:	7
Number of public wire bits:	54
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	483
\$_AND_	59
\$_AOI3_	59
\$_AOI4_	11
\$_MUX_	1
\$_NAND_	34
\$_NOR_	82
\$_NOT_	65
\$_OAI3_	39
\$_OAI4_	8
\$_OR_	34
\$_XNOR_	85
\$_XOR_	6

2.24. Executing CHECK pass (checking for obvious problems).

checking module ALU..

found and reported 0 problems.

```
ABC: ABC command line: "source <abc-temp-dir>/abc.script"
ABC:
ABC: + read_blif <abc-temp-dir>/input.blif
ABC: + read_lib -w C:\Users\ROG\Desktop\Yosys/mycells.lib
ABC: Parsing finished successfully. Parsing time = 0.
ABC: Warning: Templates are not defined.
ABC: Liberty parser cannot read "time_unit". Assuming ti
ABC: Liberty parser cannot read "capacitive_load_unit". Ass
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF"
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_
ABC: Library "demo" from "C:\Users\ROG\Desktop\Yosys/mycel
te; 0 no func). Time = 0.00 sec
ABC: Memory = 0.00 MB. Time = 0.00 sec
ABC: + strash
ABC: + dc2
ABC: + scorr
ABC: Warning: The network is combinational (run "fraig" or
ABC: + ifraig
ABC: + retime -o
ABC: + strash
ABC: + dch -f
ABC: + map
ABC: + write_blif <abc-temp-dir>/output.blif
```

4.1.2. Re-integrating ABC results.

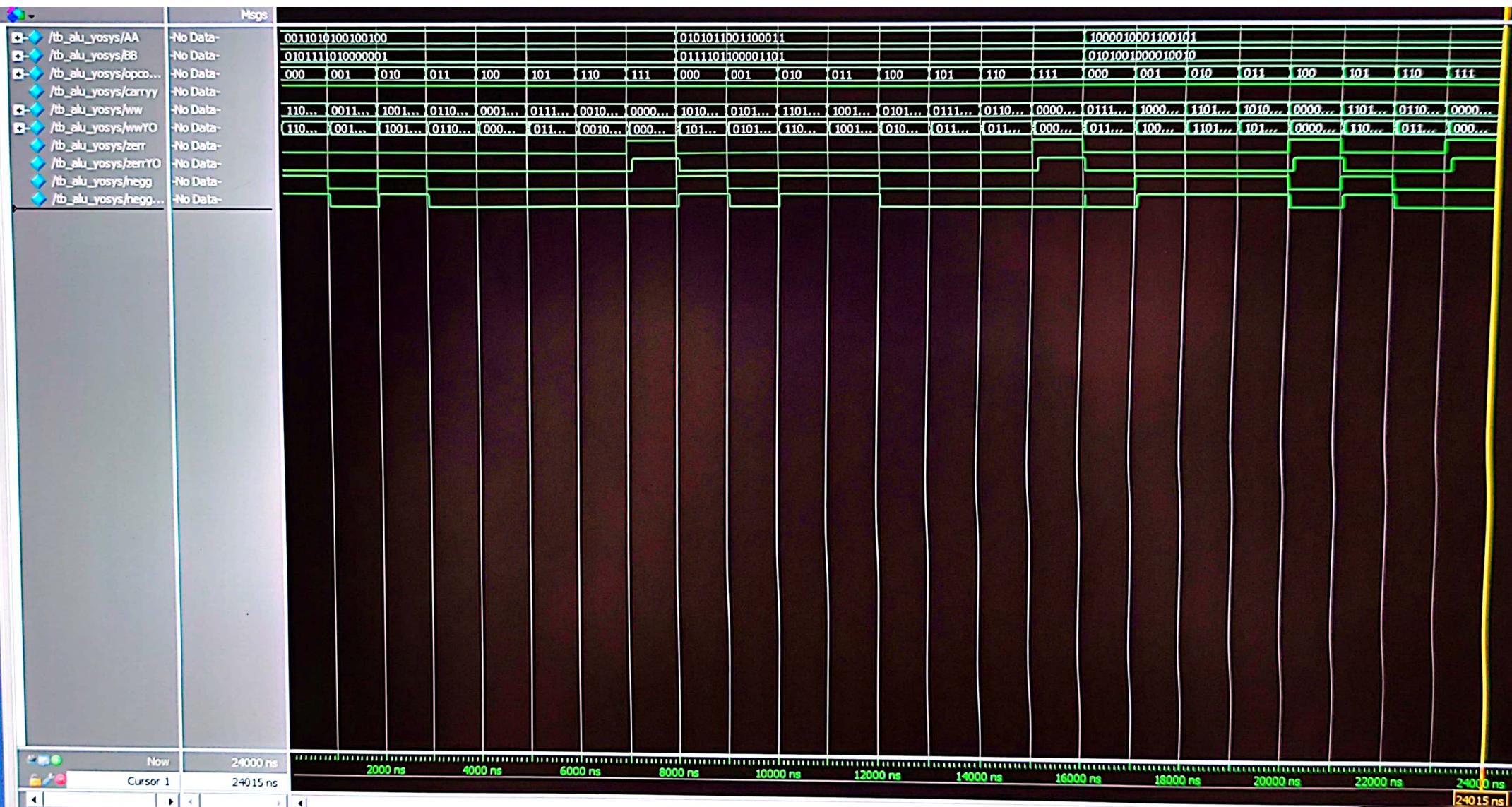
ABC RESULTS:	NAND cells:	212
ABC RESULTS:	NOR cells:	361
ABC RESULTS:	NOT cells:	128
ABC RESULTS:	internal signals:	466
ABC RESULTS:	input signals:	36
ABC RESULTS:	output signals:	17

Removing temp directory.

yosys> clean

Removed 0 unused cells and 519 unused wires.

yosys>



در این مامن $f_{\text{op}} = \text{add}$ می‌باشد. سار آنرا با opode می‌optimize کنید. این از adder استفاده نمایم و نتیجه را با f_{op} نشان دهید.

ماشین را قبل testbench می‌سازیم و میل فرآوری را با میل می‌قبل شاید که نتا از درس آن مطمن شویم. در اینجا بعد آن را به Yosys راه نماییم و آن را با synthesis می‌سازیم. این را در خود مداری کنید و بعد با یک library راه آن بسیم که ماوراء library را داریم برای testbench . بعد از آن خوبی کریم میل داده و را در testbench نصب کنیم و نتیجه آن را مشاهده کنیم.

* وقتی ب اقسام (تصویرها) در این امثله تفاوت بارزیابی آن را بین $\text{number cells before mapping to library}$ و $\text{after mapping to library}$ بیندازیم.

حال ۱ \rightarrow 483 cell

$212_{\text{NAND}} + 361_{\text{NOR}} + 128_{\text{Not}} = 701 \text{ cell}$

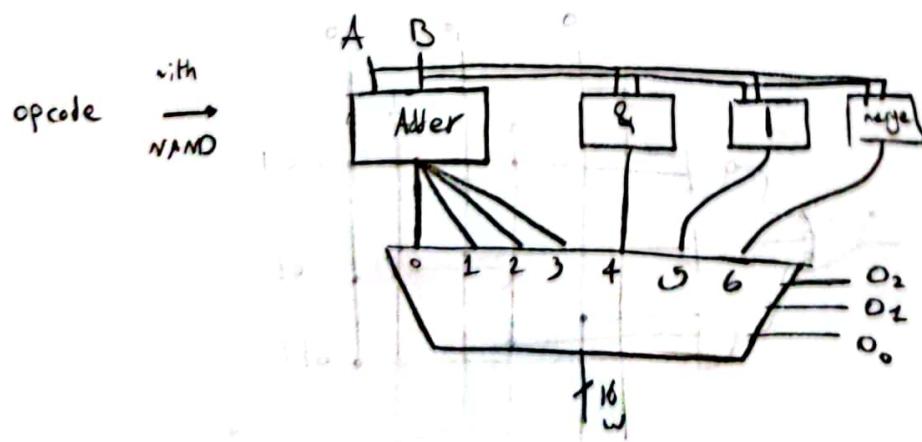
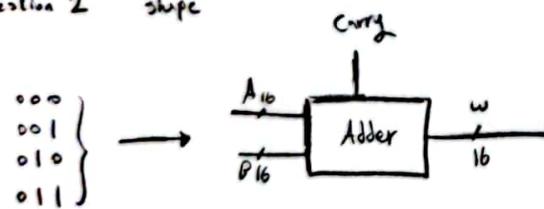
حال ۲ $\xrightarrow{\text{minimize}}$ 270 cell

$42_{\text{NAND}} + 105_{\text{NOR}} + 56_{\text{Not}} = 206 \text{ cell}$

$117_{\text{NAND}} + 155_{\text{NOR}} + 66_{\text{Not}} = 338 \text{ cell}$

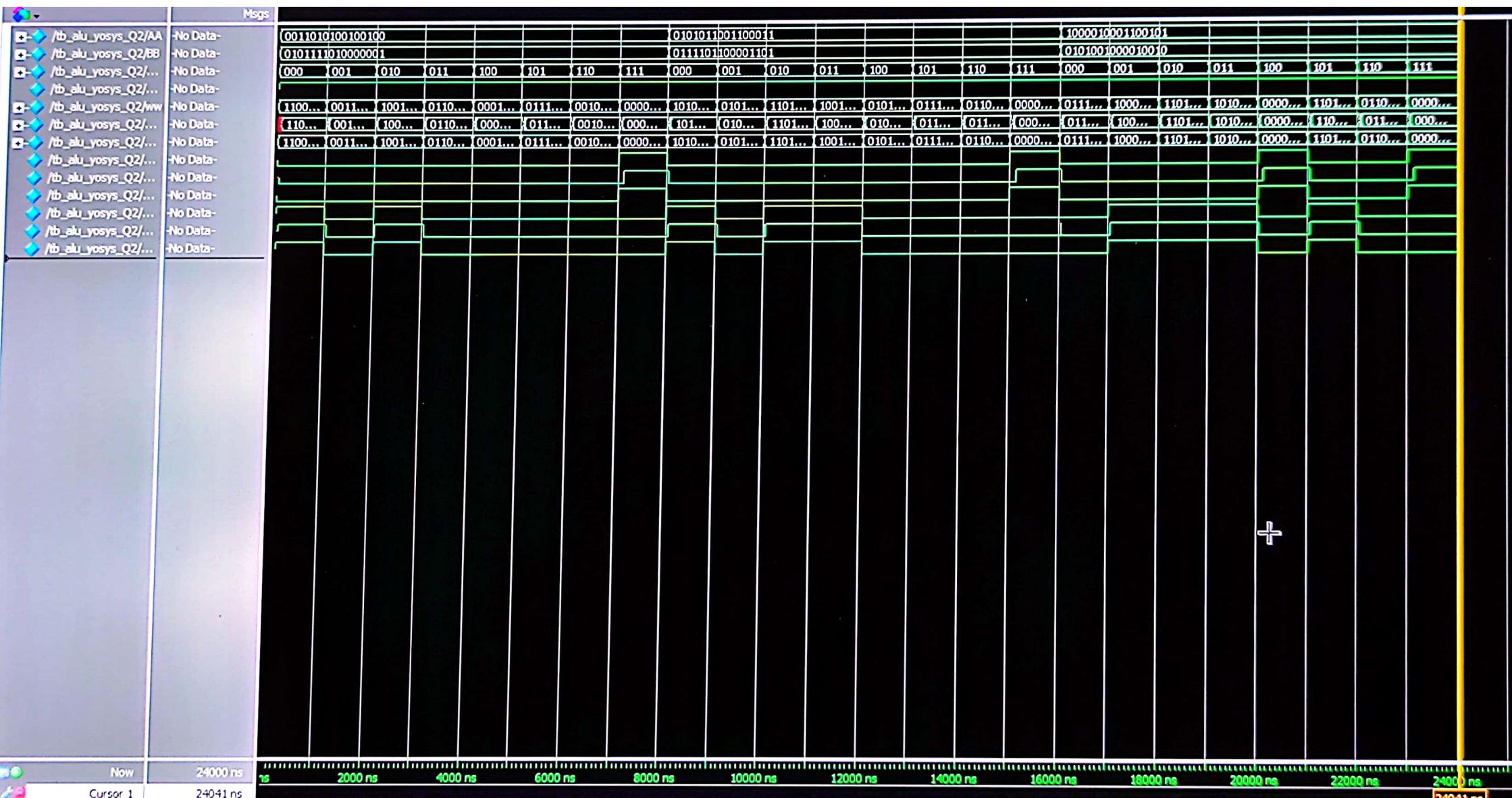
$\overline{544 \text{ cell}}$

Question 2 shape



log w_16 Q1 Yosys $000 \rightarrow 68 \text{ ns}$

Q2 Yosys $000 \rightarrow 100 \text{ ns}$



== ALU_Q2 ==

Number of wires:	140
Number of wire bits:	232
Number of public wires:	11
Number of public wire bits:	103
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	180
\$_AND_	16
\$_AOI3_	25
\$_AOI4_	16
\$_MUX_	15
\$_NAND_	25
\$_NOR_	26
\$_NOT_	4
\$_OAI3_	18
\$_OR_	18
\$_XNOR_	16
adder	1

== adder ==

Number of wires:	79
Number of wire bits:	124
Number of public wires:	4
Number of public wire bits:	49
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	91
\$_AND_	14
\$_AOI3_	11
\$_NAND_	14
\$_NOR_	2
\$_NOT_	5
\$_OAI3_	8
\$_OR_	4
\$_XNOR_	16
\$_XOR_	17

== design hierarchy ==

ALU_Q2	1
adder	1

Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	91
\$_AND_	14
\$_AOI3_	11
\$_NAND_	14
\$_NOR_	2
\$_NOT_	5
\$_OAI3_	8
\$_OR_	4
\$_XNOR_	16
\$_XOR_	17

==== design hierarchy ===

ALU_Q2	1
adder	1

Number of wires:	219
Number of wire bits:	356
Number of public wires:	15
Number of public wire bits:	152
Number of memories:	0
Number of memory bits:	0
Number of processes:	0
Number of cells:	270
\$_AND_	30
\$_AOI3_	36
\$_AOI4_	16
\$_MUX_	15
\$_NAND_	39
\$_NOR_	28
\$_NOT_	9
\$_OAI3_	26
\$_OR_	22
\$_XNOR_	32
\$_XOR_	17

```
ABC: + read_blif <abc-temp-dir>/input.blif
ABC: + read_lib -w C:\Users\ROG\Desktop\Yosys/mycells.lib
ABC: Parsing finished successfully. Parsing time = 0.00 sec
ABC: Warning: Templates are not defined.
ABC: Liberty parser cannot read "time_unit". Assuming time_un
ABC: Liberty parser cannot read "capacitive_load_unit". Assuming
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_PP0".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_PP1".
ABC: Library "demo" from "C:\Users\ROG\Desktop\Yosys/mycells.lib"
te; 0 no func). Time = 0.00 sec
ABC: Memory = 0.00 MB. Time = 0.00 sec
ABC: + strash
ABC: + dc2
ABC: + scorr
ABC: Warning: The network is combinational (run "fraig" or "frai
ABC: + ifraig
ABC: + retime -o
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:	117
ABC RESULTS:	NOR cells:	155
ABC RESULTS:	NOT cells:	66
ABC RESULTS:	internal signals:	129
ABC RESULTS:	input signals:	52
ABC RESULTS:	output signals:	50

Removing temp directory.

4.2. Extracting gate netlist of module '\adder' to '<abc-temp-dir>/adder.gate.ngc'

Extracted 91 gates and 124 wires to a netlist network with 33 inputs and 33 outputs.

4.2.1. Executing ABC.

Running ABC command: <yosys-exe-dir>/yosys-abc -s -f <abc-temp-dir>/abc.script

```
ABC: ABC command line: "source <abc-temp-dir>/abc.script".
ABC:
ABC: + read_blif <abc-temp-dir>/input.blif
ABC: + read_lib -w C:\Users\ROG\Desktop\Yosys/mycells.lib
ABC: Parsing finished successfully. Parsing time = 0.00 sec
ABC: Warning: Templates are not defined.
```

4.2.1. Executing ABC.

Running ABC command: <yosys-exe-dir>/yosys-abc -s -f <abc-temp-dir>/abc.script
ABC: ABC command line: "source <abc-temp-dir>/abc.script".
ABC:
ABC: + read_blif <abc-temp-dir>/input.blif
ABC: + read_lib -w C:\Users\ROG\Desktop\Yosys/mycells.lib
ABC: Parsing finished successfully. Parsing time = 0.00 sec
ABC: Warning: Templates are not defined.
ABC: Libery parser cannot read "time_unit". Assuming time_unit
ABC: Libery parser cannot read "capacitive_load_unit". Assuming
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_PP0".
ABC: Scl_LibertyReadGenlib() skipped sequential cell "DFF_PP1".
ABC: Library "demo" from "C:\Users\ROG\Desktop\Yosys/mycells.lib"
ABC: Memory = 0.00 MB. Time = 0.00 sec
ABC: + strash
ABC: + dc2
ABC: + scorr
ABC: Warning: The network is combinational (run "fraig" or "frain")
ABC: + ifraig
ABC: + retime -o
ABC: + strash
ABC: + dch -f
ABC: + map
ABC: + write_blif <abc-temp-dir>/output.blif

4.2.2. Re-integrating ABC results.

ABC RESULTS:	NAND cells:	42
ABC RESULTS:	NOR cells:	105
ABC RESULTS:	NOT cells:	56
ABC RESULTS:	internal signals:	75
ABC RESULTS:	input signals:	33
ABC RESULTS:	output signals:	16

Removing temp directory.

yosys> clean

Removed 0 unused cells and 355 unused wires.

yosys> |

