

# VLSI Circuit Design II– EHB 425E HOMEWORK III Yiğit Bektaş GÜRSOY 040180063

Class Lecturer: Sıddıka Berna Örs Yalçın

Class Assistant: Yasin Fırat Kula

# INTRODUCTION

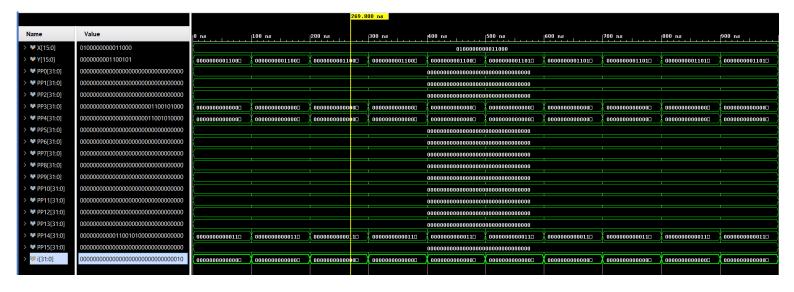
The author of this report has designed a 16-bit multiplier which has 3 main blocks: Partial Product Generator, Partial Product Accumulator, and Final Stage Adder. The HDL codes and dot files will be uploaded to Ninova in a zip file, but they are not included in this report because the dot files are large and some of them did not produce .pdf or .svg files. The author will include all the dot files in the zip file

#### 1- PPG

• Behavioral Simulation Results

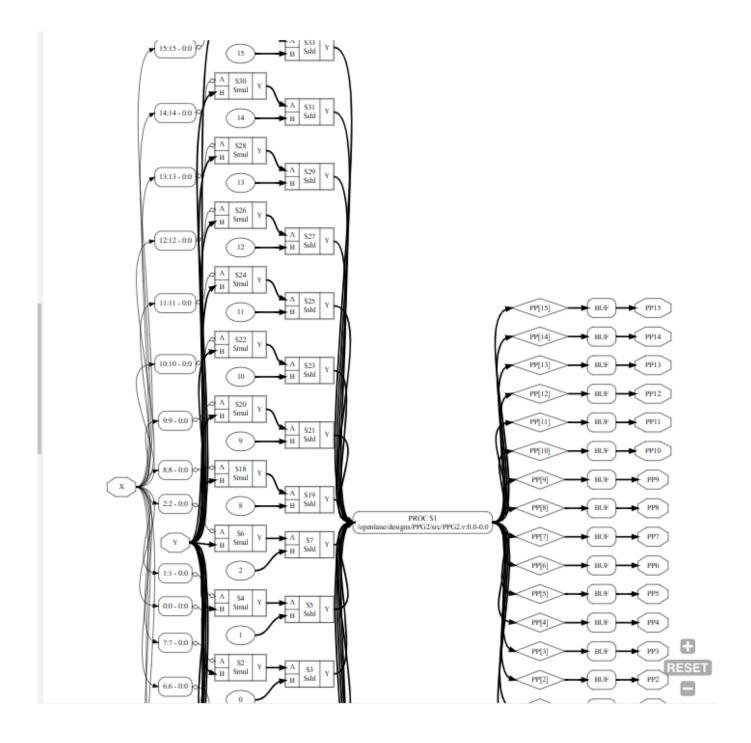


Partial product results are given as seen above. The first bit of X is 1 and accordingly the first PP value must be Y itself. Then the 2nd bit of X is 0, so the 2nd value of PP should be 0. The 3rd bit of X is 1, so its bit must be shifted to the left by 1 minus. In other words, if the 3rd bit is 1, the value of the 3rd element Y of the PP should be written 2 shifted to the left. The pattern continues in this way. The simulation is working correctly.

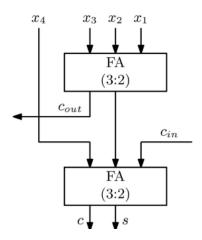


#### WHOLE SIMULATION

The hierarchy dot file belonging to PPG is given below. Here, it is seen that the generated wire signals and the generated signals are connected to the output. In the left digit, it is seen that a number is shifted to the left side by increasing gradually.



## 2- 4:2 COMPRESSOR



A basic 4:2 compressor circuit

#### • Behavioral Simulation Results

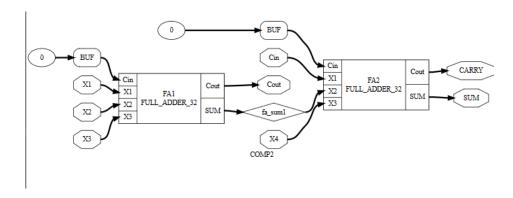
											1,000.000 ns
Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns
> W X1[31:0]	40180074	40180064	40180065	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073
> W X2[31:0]	40180075	40180065	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074
> <b>W</b> X3[31:0]	40180076	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075
> <b>W</b> X4[31:0]	40180077	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076
> W Cin[31:0]	40180078	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077
> W Cout[31:0]	0						D				
> ₩ CARRY[31:0]	0						D				
> ™ SUM[31:0]	200900380	200900330	200900335	200900340	200900345	200900350	200900355	200900360	200900365	200900370	200900375
> <b>W</b> i[31:0]	10	0	1	2	3	4	5	6	7	8	9

$$\begin{vmatrix} x_1+x_2+x_3+x_4+c_{in}=s+2(c+c_{out})\\ \textit{4:2 Compressor Equation} \end{vmatrix}$$

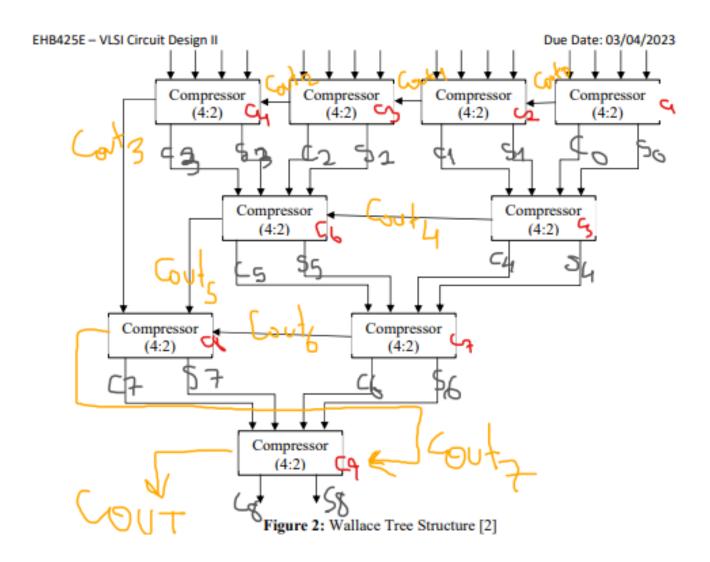
$\frac{(40180064 + 40180068)}{2} \cdot 5$	= 200 900 330	$\frac{(40180069 + 40180073)}{2} \cdot 5$	= 200 900 355
$\frac{(40180065 + 40180069)}{2} \cdot 5$	= 200 900 335	$\frac{(40180070 + 40180074)}{2} \cdot 5$	= 200 900 360
$\frac{(40180066 + 40180070)}{2} \cdot 5$	= 200 900 340	$\frac{(40180071 + 40180075)}{2} \cdot 5$	= 200 900 365
$\frac{(40180067 + 40180071)}{2} \cdot 5$	= 200 900 345	$\frac{(40180072 + 40180076)}{2} \cdot 5$	= 200 900 370
$\frac{(40180068 + 40180072)}{2} \cdot 5$	= 200 900 350	$\frac{\left(40180073 + 40180077\right)}{2} \cdot 5$	= 200 900 375

The equation for 4:2 compressor is as stated above. The numbers indicated by the sequential number addition formula were added. When the additions were made, the correctness of the result was verified with the calculator. The calculator results are also shown above.

Below is the output of the .dot file of 4:2 Compressore. Compressore's cin, cout, sum, carry, input signals and also some empty inputs are assigned 0 by giving buf.

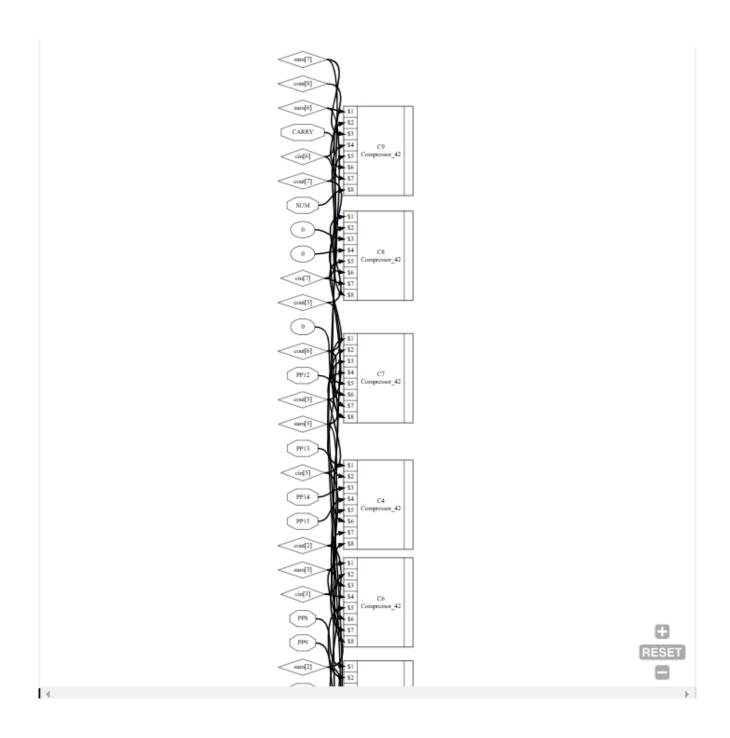


### 3- PPA



Above is the block from which the PPA was derived with the 4:2 Compressor. Verilog code is written and tested by simulation. The simulation results are mentioned below.

In the dot extension hierarchy file, Compressors cin, cout, sum and input signals are displayed as connected to each other. I have outlined part of the circuit below when the image is too large.



## • Behavioral Simulation Results

											1,000.000 ns
Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns
> W PP0[31:0]	40180073	40180063	40180064	40180065	40180066	40180067	40180068	40180069	40180070	40180071	40180072
> W PP1[31:0]	40180074	40180064	40180065	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073
> W PP2[31:0]	40180075	40180065	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074
> W PP3[31:0]	40180076	40180066	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075
> W PP4[31:0]	40180077	40180067	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076
> W PP5[31:0]	40180078	40180068	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077
> W PP6[31:0]	40180079	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077	40180078
> W PP7[31:0]	40180080	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077	40180078	40180079
> W PP8[31:0]	40180081	40180071	40180072	40180073	40180074	40180075	40180076	40180077	40180078	40180079	40180080
> W PP9[31:0]	40180082	40180072	40180073	40180074	40180075	40180076	40180077	40180078	40180079	40180080	40180081
> ™ PP10[31:0]	40180083	40180073	40180074	40180075	40180076	40180077	40180078	40180079	40180080	40180081	40180082
> W PP11[31:0]	40180084	40180074	40180075	40180076	40180077	40180078	40180079	40180080	40180081	40180082	40180083
> W PP12[31:0]	40180085	40180075	40180076	40180077	40180078	40180079	40180080	40180081	40180082	40180083	40180084
> W PP13[31:0]	40180086	40180076	40180077	40180078	40180079	40180080	40180081	40180082	40180083	40180084	40180085
> <b>W</b> PP14[31:0]	40180087	40180077	40180078	40180079	40180080	40180081	40180082	40180083	40180084	40180085	40180086
> W PP15[31:0]	40180088	40180078	40180079	40180080	40180081	40180082	40180083	40180084	40180085	40180086	40180087
> <b>₩</b> COUT[31:0]	0	(					0				
> W CARRY[31:0]	0	(					0				
> <b>W</b> SUM[31:0]	642881288	642881128	642881144	642881160	642881176	642881192	642881208	642881224	642881240	642881256	642881272
> 🕨 i[31:0]	10	0	1	2	3	4	5	6	7	8	9

$\frac{\left(40180063+40180078\right)}{2}\cdot 16$	= 642881128	$\frac{(40180068 + 40180083)}{2} \cdot 16$	= 642 881 208
$\frac{\left(40180064+40180079\right)}{2}\cdot 16$	= 642881144	$\frac{(40180069 + 40180084)}{2} \cdot 16$	= 642 881 224
$\frac{\left(40180065 + 40180080\right)}{2} \cdot 16$	= 642881160	$\frac{(40180070 + 40180085)}{2} \cdot 16$	= 642881240
$\frac{(40180066 + 40180081)}{2} \cdot 16$	= 642881176	$\frac{(40180071 + 40180086)}{2} \cdot 16$	= 642 881 256
$\frac{(40180067 + 40180082)}{2} \cdot 16$	= 642881192	$\frac{(40180072 + 40180087)}{2} \cdot 16$	= 642881272

The PPA circuit shown above is a circuit that shows the sum of the PP values. The results of the simulation are compared with the results of the calculator. It has been verified that the circuit is working correctly.

#### 4- RCA

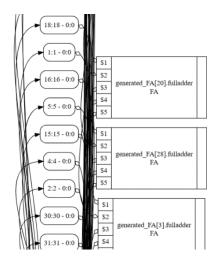
• Behavioral Simulation Results

											1,000.000 ns
Name	Value	0 ns	100 ns	200 ns	300 ns	400 ns	500 ns	600 ns	700 ns	800 ns	900 ns
> <b>W</b> X[31:0]	40180079	40180069	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077	40180078
> 🕨 Y[31:0]	40180080	40180070	40180071	40180072	40180073	40180074	40180075	40180076	40180077	40180078	40180079
₩ CIN	0									j	
™ COUT	0									i	
> <b>W</b> S[31:0]	80360159	80360139	80360141	80360143	80360145	80360147	80360149	80360151	80360153	80360155	80360157
> 🍯 i[31:0]	10	0	1	2	3	4	5	6	7	8	9
> W SIZE[31:0]	32	32									
											ı
401800	40180069 + 40180070 = 80360139				40180060 ± 40180070 ± 12 = 80360					260151	

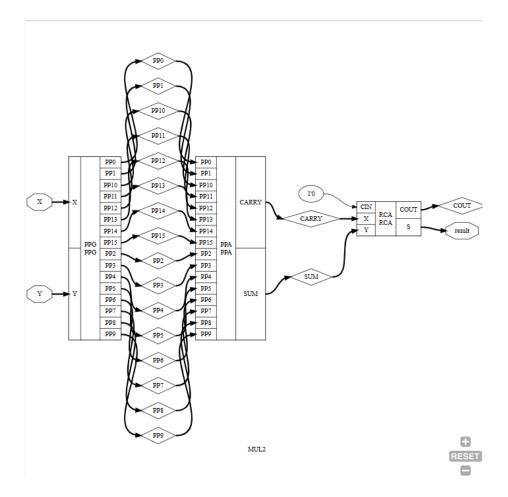
40180069 + 40180070	= 80 360 139	40180069 + 40180070 + 12	= 80 360 151
40180069 + 40180070 + 2	= 80 360 141		
40 180 069 + 40 180 070 + 4	= 80 360 143	40180069 + 40180070 + 14	= 80 360 153
40 180 069 + 40 180 070 + 6	= 80 360 145	40180069 + 40180070 + 16	= 80 360 155
40180069 + 40180070 + 8	= 80 360 147	40.100.000 - 40.100.070 - 10	20.000.157
40180069 + 40180070 + 10	= 80 360 149	40180069 + 40180070 + 18	= 80 360 157

RCA is a summation circuit. Behavioral simulation results of the designed 32-bit RCA circuit are shown above. Both entries were collected from the desmos calculator to verify the results. Collection results are mentioned above. When the summation results and the simulation results of the circuit are compared, it is seen that 1 to 1 correct results are obtained. Thus, it has been verified that the circuit works correctly

In the .dot file the generated FULL adders in the 32bit RCA circuit and the inputs connected to them are seen. For detailed view, you can see the dot file in the folder. I have outlined part of the circuit below when the image is too large.



#### 5- MUL



Behavioral Simulation Results (iVerilog)

```
result
                                       expected result=
           1235, result
                                 2470, expected result=
                                                               2470
          12373, result
                                12373, expected result=
                                                               12373
            151, result =
                                 1510, expected result=
                                                               1510
  10,
1000,
           1235, result =
                              1235000, expected result=
                                                            1235000
          12373, result
                            150171101, expected
                                                result=
                                                          150171101
             18, result
                                  126, expected result=
                                                                126
             19, result
                                   57, expected result=
                                                                 57
              2, result
                                24648, expected result=
                                                              24648
                             16769025, expected result=
           4095, result =
                                                           16769025
```

Post Synthesis Results (iVerilog)

```
0, expected
           1235, result
                                 2470, expected result=
                                12373, expected
          12373, result
                                                 result=
                                                               12373
                                 1510, expected result=
                                                                1510
            151, result
1000,
           1235,
                 result
                              1235000, expected result=
                                                             1235000
                 result
                            150171101, expected
                                                 result=
                                                           150171101
             18, result
                                  126, expected
                                                 result=
                                                                 126
                 result
                                   57, expected
                                                                  57
                                24648, expected result=
                                                               24648
                 result
           4095,
                 result
                             16769025, expected result=
                                                            16769025
```

Both simulations work correctly.

### **Extra Note:**

Since the post\_tech.dot files are too large and do not fit on the page, I archived them and put them in the folders I uploaded in the assignment. When I view it, there are lots of AND OR NORs.

I've imaged the NAND and XOR gates. They were both very small in resolution and outnumbered.