EHB425E – VLSI Circuit Design II   Due Date: 12/03/2024

Homework 3

In this homework you will design a multiplier for 16-bit positive integers Multiplier, *X*, and Multiplicand, *Y*. The architecture of the circuit is given in Fig. 1. You can find reference web links in classroom source files and some papers in the homework source files.

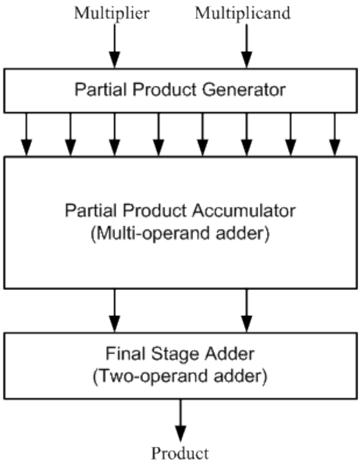

Figure 1: Architecture of a multiplier [1]

1) Design a circuit, PPG, for production of 16 32-bit partial products (PPs). For example

PP_0	=	0	0	…	0	0	Y(0) AND X(15)	Y(0) AND X(14)	…	Y(0) AND X(1)	Y(0) AND X(0)
PP_1	=	0	0	…	0	Y(1) AND X(15)	Y(1) AND X(14)	…	Y(1) AND X(1)	Y(1) AND X(0)	0
…											
PP_{14}	=	0	0	Y(14) AND X(15)	Y(14) AND X(14)	…	Y(14) AND X(1)	Y(14) AND X(0)	0	...	0
PP_{15}	=	0	Y(15) AND X(15)	Y(15) AND X(14)	…	Y(15) AND X(1)	Y(15) AND X(0)	0	…	0	0

a) Write a Verilog code for PPG. Save this file by giving name as “PPG.v”. Synthesize this module in OpenLane and add the resulting dot schematics to your report. Examine what type of elements are used to model the written HDL code.

b) Write a test bench file with name “PPG_tb.v” to test your design. Perform a behavioral simulation for your design in Icarus Verilog (iverilog), covering 10 inputs calculated using Eq. (1). Add the result to your report to prove that the circuit is working as expected (console outputs or gtkwave waveform).

$$\text{For } 0 \leq i \leq 10,\ \{X, Y\} = ((\text{Student ID number}+i) \bmod 2^{32})_2 \qquad (1)$$

2) You will use Wallace tree shown in Fig. 2 for Partial Product Accumulator.

Design a circuit, Compressor_42, for adding five 32-bit positive integers, x_1, x_2, x_3, x_4, c_{in} and producing three outputs as 32-bit positive integers, *Carry, Sum and* c_{out}. The schematic for 4:2 compressor circuit is shown in Fig. 3.

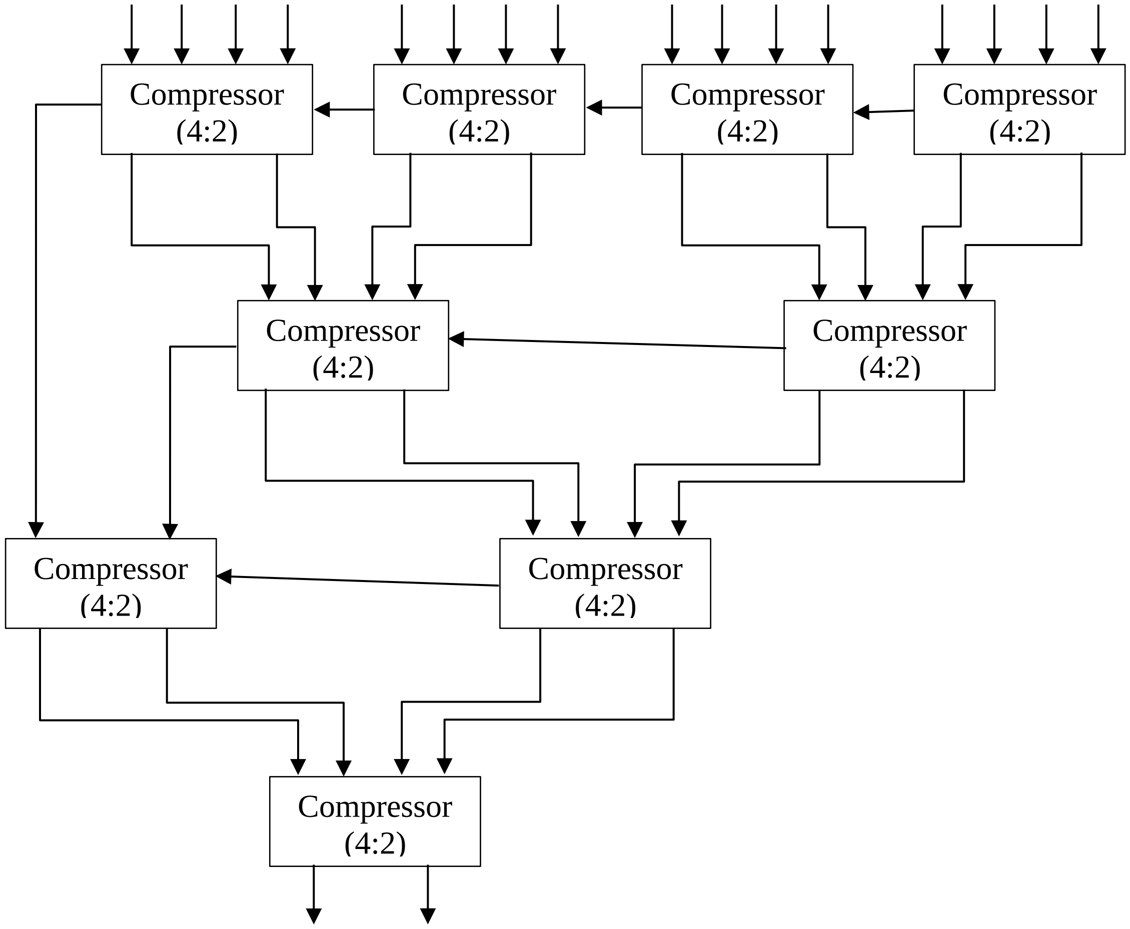

Figure 2: Wallace Tree Structure [2]

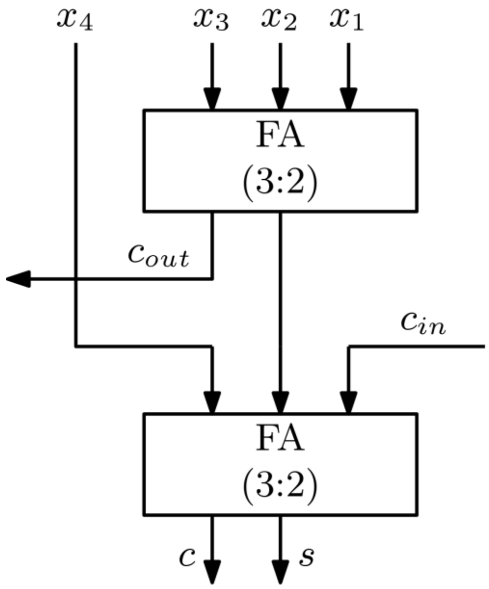

Figure 3: A basic 4:2 compressor circuit [2]

a) Write a Verilog code for the 4:2 compressor block shown in Fig. 3. Save this file by giving name as "Compressor_42.v". Synthesize this module in OpenLane and add the resulting dot schematics to your report. Examine what type of elements are used to model the written HDL code.

b) Write a test bench file with name "Compressor_42_tb.v" to test your design. Perform a behavioral simulation for your design in iverilog, covering 10 inputs calculated using Eq. (2) and (3). Add the result to your report to prove that the circuit is working as expected (console outputs or gtkwave waveform)..

For 0≤i≤10, 1≤j≤4, x_j=((Student ID number+i+j) mod $2^{32}$$)_2$ (2)

For 0≤i≤10, c_{in}=((Student ID number+i+5) mod $2^{32}$$)_2$ (3)

3) Design a circuit for Partial Product Accumulator with the structure shown in Fig. 1.

a) Write a Verilog code for Partial Product Accumulator by using "Compressor_42.v" as building blocks. Save this file by giving name as "PPA.v". Synthesize this module in OpenLane and add the resulting hierarchy dot schematic to your report. Examine what type of elements are used to model the written HDL code.

b) Write a test bench file with name "PPA_tb.v" to test your design. Perform a behavioral simulation for your design in iverilog covering 10 inputs calculated using Eq. (4). Add the result to your report to prove that the circuit is working as expected.

For 0≤i≤10, 0≤j≤15, PP_j=((Student ID number+i+j) mod $2^{32}$$)_2$ (4)

4) Design a circuit for Final Stage Adder shown in Fig. 1. You will use Ripple Carry Adder (RCA) for the final stage adder, the schematic for RCA circuit is shown in Fig. 4.

a) Write a Verilog code for Ripple Carry Adder. Save this file by giving name as "RCA.v". Elaborate this module in OpenLane and add the resulting hierarchy dot schematic to your report. Examine the post_tech schematic to see what type of elements are used to model the written HDL code.

b) Write a test bench file with name "RCA_tb.v" to test your design. Perform a behavioral simulation for your design in iverilog, covering 10 inputs calculated using Eq. (5) and (6). Add the result to your report to prove that the circuit is working as expected.

For 0≤i≤10, x=((Student ID number+i+6) mod $2^{32}$$)_2$ (5)

For 0≤i≤10, y=((Student ID number+i+7) mod $2^{32}$$)_2$ (6)

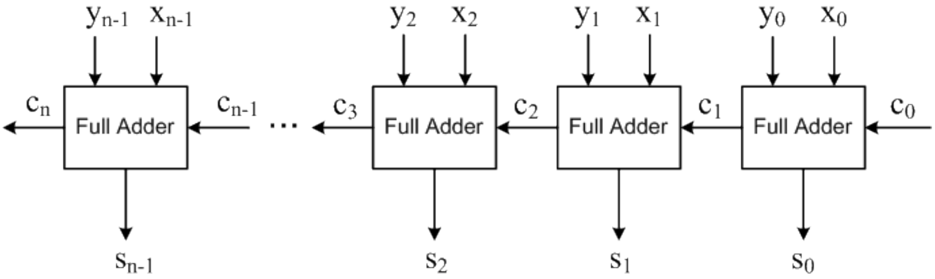

Figure 4: n-bit ripple carry adder [1]

5) Design a circuit for multiplier with the structure shown in Fig. 1.

a) Write a Verilog code for multiplier by using "PPG.v", "PPA.v" and "RCA.v".as building blocks. Save this file by giving name as "MUL.v". Elaborate this module in OpenLane and add the resulting hierarchy schematic to your report. Examine what type of elements are used to model the written HDL code.

b) Write a test bench file with name "MUL_tb.v" to test your design. Perform a behavioral simulation and a post-synthesis functional simulation for your design in iverilog, covering at least 10 possible random inputs. Add the results to your report to prove that the circuit is working as expected.

References

1) Hardware algorithms for arithmetic modules, http://www.aoki.ecei.tohoku.ac.jp/arith/mg/algorithm.html#ppa
2) Shirshendu Roy, "Alternative Techniques for Partial Product Accumulation", https://digitalsystemdesign.in/alternative-techniques-for-partial-product-accumulation/