ECE 385
2021 FA

LAB 3
Introduction
to SystemVerilog, FPGA, CAD, and 16-bit Adders

Zhou Qishen, Xie Mu

Introduction

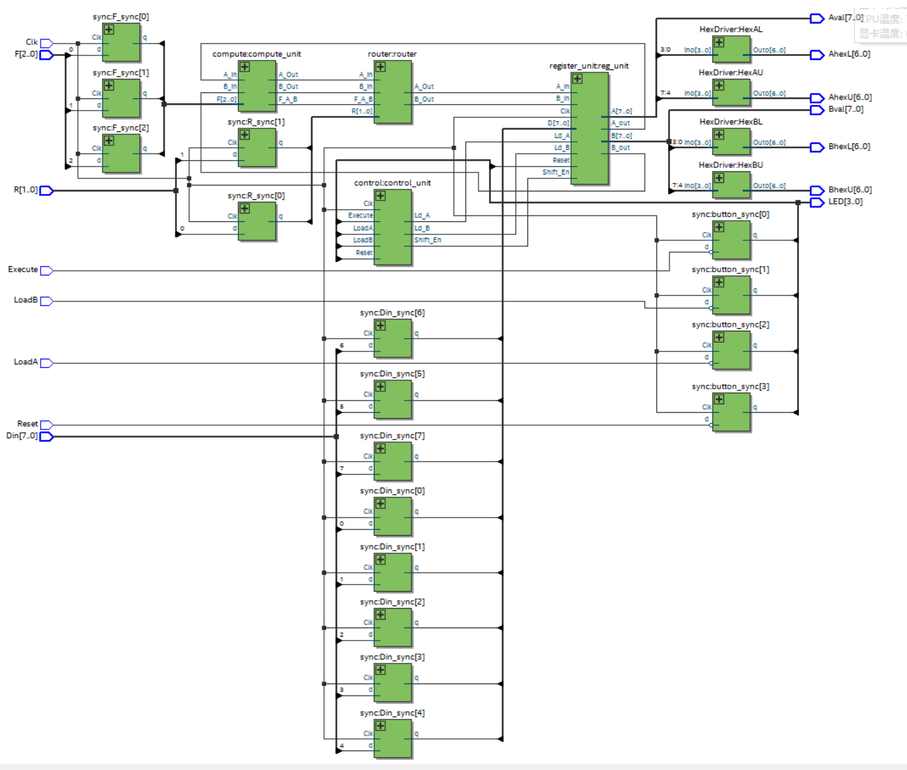
In this lab, we designed three different adder: ripple carry adder, carry lookahead adder, and carry select adder , and extended 4-bit processor to 8-bit.

Processor Part

Description

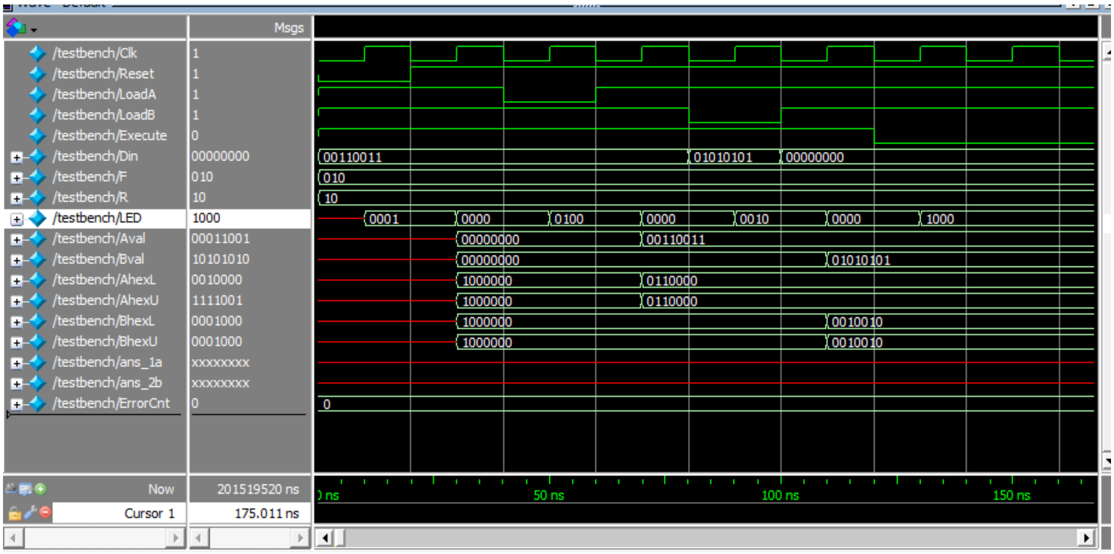
In this lab, we made some changes on the original code to extend the 4-bit processor to 8 bit. The first part modified is all the input, output and register, all changing from 4 bit to 8 bit. The rest thing we need to do is modifying the state machine, to match the iteration number of the register.

Block Diagram



Processor Block Diagram

Simulation



Simulation Result for 8 bit Processor

Adder Part

Carry Ripple Adder

Carry Ripple Adder is an adder with n full adder implemented in series.

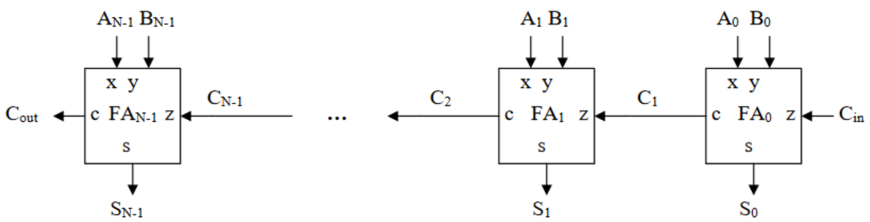
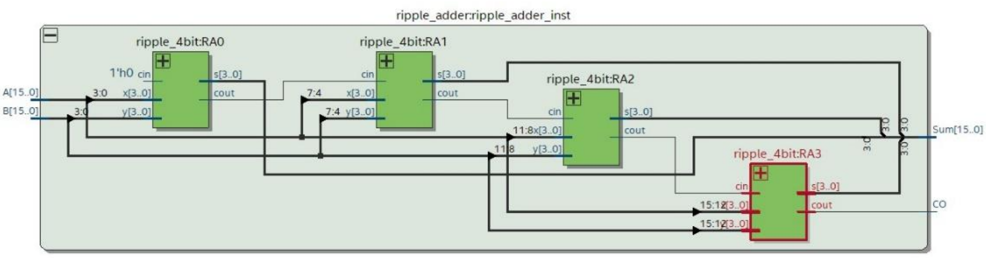


Figure 3: *N*-bit Carry-Ripple Adder Block Diagram



Carry Lookahead Adder

The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder.

P and G is calculated by formula:

$$P = A \oplus B; \;\; G = A \cdot B$$

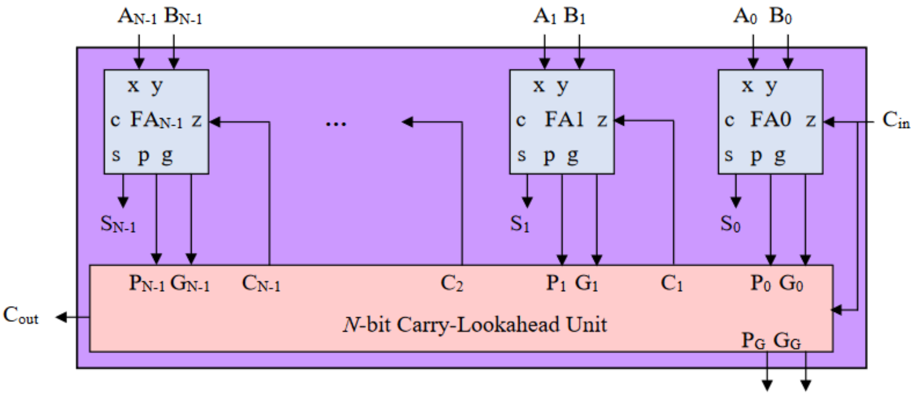
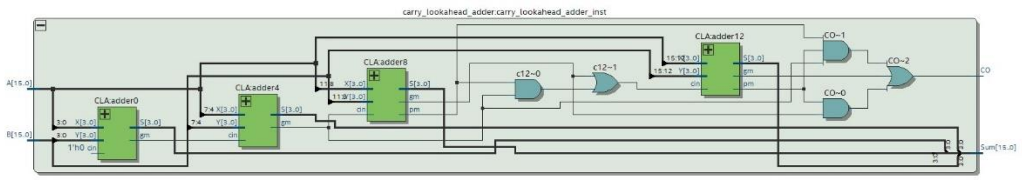


Figure 4: *N*-bit Carry-Lookahead Adder Block Diagram



Carry Select Adder

The carry-select adder generally consists of ripple-carry adders and a multiplexer, where adding two n-bit numbers is done with two adders in order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one.

One adder computes the sum and carry-out based on the assumption that the carry-in is 0, and the other assumes that the carry-in is 1. In this way, both possible outcomes are pre-computed. Once the real carry-in arrives, the corresponding sum and carry-out is selected to be delivered to the next stage.

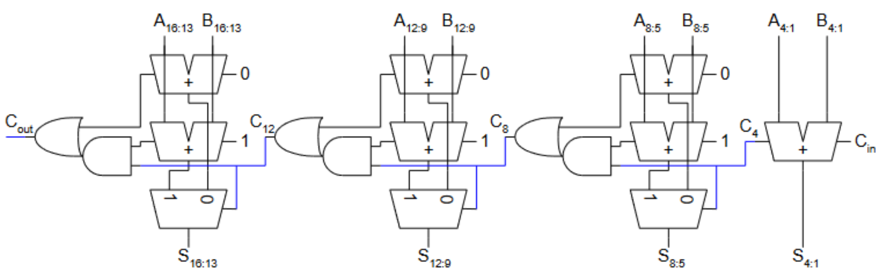
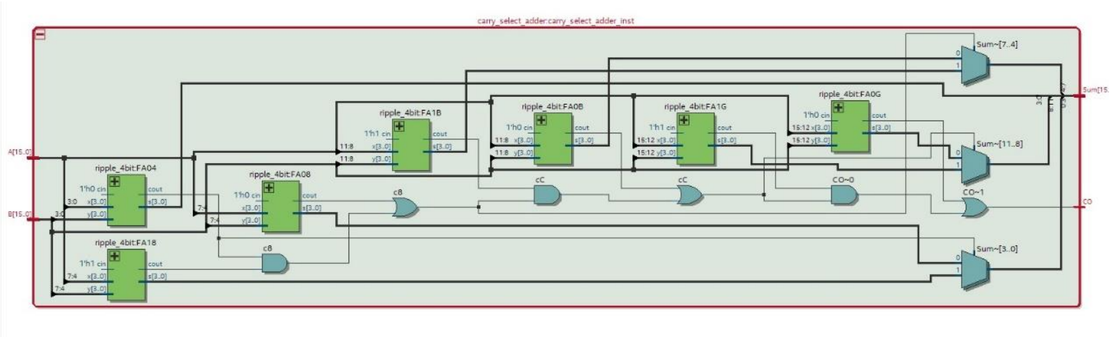
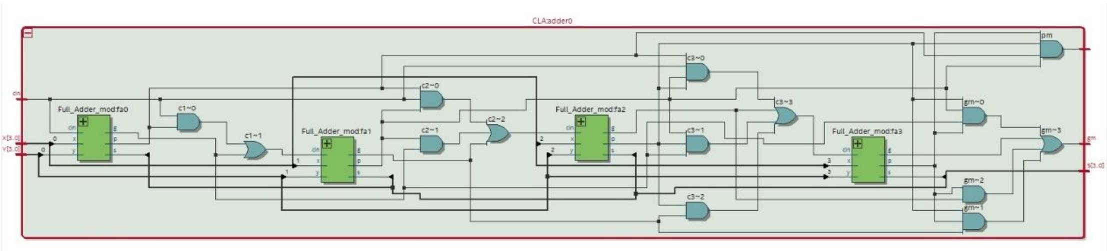


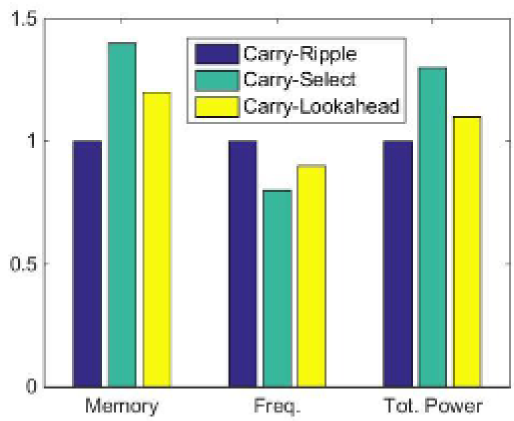
Figure 5: 16-bit Carry-Select Adder Block Diagram





Designing Comparison

For CRA, it is the simplest adder among three adders above. However, due to the nature of in series, it is quite slow. For SLA, the adder is more complex and more faster by predicting the future state, which means consuming much more energy. Finally, the CSA, it gained limited time improvement, while using twice of the space.



Module Description:

Module: carry_lookahead_adder.sv
Inputs: logic[15:0] A, logic[15:0] B,
Outputs: logic[15:0] Sum, logic CO
Description: carry lookahead adder.
Purpose: carry lookahead adder.

Module: carry_select_adder.sv
Inputs: logic[15:0] A, logic[15:0] B,
Outputs: logic[15:0] Sum, logic CO
Description: carry selected adder.
Purpose: carry selected adder.

Module: ripple_adder.sv
Inputs: logic[15:0] A, logic[15:0] B,
Outputs: logic[15:0] Sum, logic CO
Description: ripple adder.
Purpose: ripple adder

Post Lab

Generally speaking, LUT and DFLIP-FLOP match for small scale while memory is suitable for larger scale. LUT is the most simple but will cost most space, since we have to save all its possibility for future calculate. And for memory, it is well organized. Thus, it also will consume significant amount of resources.

	Resource	Usage
1	Estimated Total logic elements	130
2		
3	Total combinational functions	114
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	74
2	-- 3 input functions	8
3	-- <=2 input functions	32
5		
6	Logic elements by mode	
1	-- normal mode	114
2	-- arithmetic mode	0
7		
8	Total registers	105
1	-- Dedicated logic registers	105
2	-- I/O registers	0
9		
10	I/O pins	93
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	Clk~input
15	Maximum fan-out	105
16	Total fan-out	826
17	Average fan-out	2.04

	Resource	Usage
1	Estimated Total logic elements	139
2		
3	Total combinational functions	123
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	79
2	-- 3 input functions	14
3	-- <=2 input functions	30
5		
6	Logic elements by mode	
1	-- normal mode	123
2	-- arithmetic mode	0
7		
8	Total registers	105
1	-- Dedicated logic registers	105
2	-- I/O registers	0
9		
10	I/O pins	93
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	Clk~input
15	Maximum fan-out	105
16	Total fan-out	860
17	Average fan-out	2.08

	Resource	Usage
1	Estimated Total logic elements	137
2		
3	Total combinational functions	121
4	Logic element usage by number of LUT inputs	
1	-- 4 input functions	77
2	-- 3 input functions	19
3	-- <=2 input functions	25
5		
6	Logic elements by mode	
1	-- normal mode	121
2	-- arithmetic mode	0
7		
8	Total registers	105
1	-- Dedicated logic registers	105
2	-- I/O registers	0
9		
10	I/O pins	93
11		
12	Embedded Multiplier 9-bit elements	0
13		
14	Maximum fan-out node	Clk~input
15	Maximum fan-out	105
16	Total fan-out	857
17	Average fan-out	2.08

The carry selected adder consumes the most energy, since it do the most calculation.

Conclusion

In conclusion, in this lab, we learnt how to compose and test System Verilog by expending 4-bit processor to 8 bit and designing three different adders.