

컴퓨터 공학 기초 실험2 보고서

실험제목: Ripple Carry Adder

실험일자: 2023년 09월 15일 (금)

제출일자: 2023년 09월 28일 (목)

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 금요일 0, 1, 2

학 번: 2020202037

성 명: 엄정호

1. 제목 및 목적

A. 제목

Ripple Carry Adder

B. 목적

Half-adder와 Full-adder 모듈을 설계하고, full-adder모듈을 이용해 4bit rca모듈을 구성해 본다. 설계한 4bit rca모듈을 이용해 어떻게 32bit rca를 구상할 수 있을지 생각해 본다.

2. 원리(배경지식)

Half adder : 2개의 1-bit입력을 받아 sum과 carry out을 출력하는 가산기이다. A, B 두 개의 입력을 받아 sum을 의미하는 S와 carry out을 의미하는 co를 출력한다.

Full adder : 2 개의 1-bit 입력과 1-bit carry in 을 입력으로 받아 sum 과 carry out 을 출력하는 가산기이다. a, b, ci 3 개의 입력을 받아 sum 을 의미하는 S 와 carry out 을 의미하는 co 을 출력한다.

전가산기는 여러가지 방법으로 구성 될 수 있는데 반가산기 두 개를 이용해서 구성하거나, $S = a \text{ XOR } b \text{ XOR } ci$ 와 $Cout = (a*b)+(ci+(a+b))$ 로 표현 가능하다.

Ripple carry adder : N-bit ripple carry adder 는 n-bit 를 가지는 두 개의 수를 더하기 위한 가장 간단한 형태의 가산기이다. Ripple carry adder 는 더하기 하려는 수의 bit 개수만큼 full adder 를 연결하여 구현한다.

N-bit ripple carry adder 의 worst-case delay 는 각각 1-bit full adder 의 carry delay 를 모두 더한 시간이 지난 후에 결과 값이 나오기 때문에 연산 속도가 느리다.

2의 보수 : 어떤 수의 커다란 2의 제곱수에서 빼서 얻은 이진수이다. 대부분의 산술 연산에서 원래 숫자의 음수로 취급된다. 주어진 이진수 보다 한 자리 높고 MSB가 1인 나머지가 0인 수를 빼서 얻은 수가 2의 보수이다. 또는 주어진 이진수의 모든 자릿 0을 1로 바꾼 후 1을 더하면 2의 보수를 구할 수 있다.

Ex) 14 : 0000 1110 -> 1111 0001 -> 1111 0010 : - 14

0000 1110 과 1111 0010을 더해보면 0이 나옴을 확인할 수 있다.

3. 설계 세부사항

Half-adder truth table

Input	Output
-------	--------

a	b	co	Sum s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Carry out - Karnaugh Map and Boolean Equation

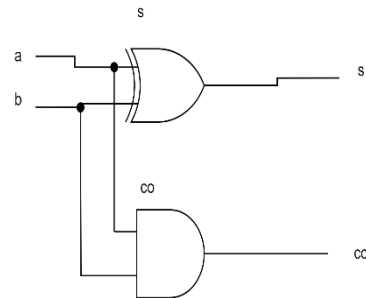
a \ b	0	1
0	0	0
1	0	1

Carry out = $a \& b$

Sum s - Karnaugh Map and Boolean Equation

a \ b	0	1
0	0	1
1	1	0

Sums $s = a'b + ab'$



Full adder truth table

Input			Output	
ci	a	b	co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Sum-s Karnaugh Map and Boolean Equation

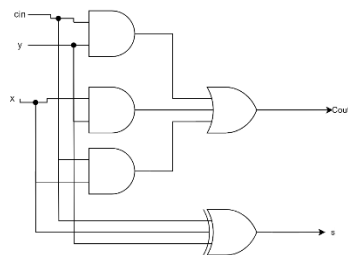
ci \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$\begin{aligned}
\text{Sums} &= a'b'ci + a'bci' + ab'ci' + abci \\
&= a'(b'ci + bci') + a(b'ci' + abci) \\
&= a'(b \text{ XOR } ci) + a((b \text{ XOR } ci)') \\
&= a \text{ XOR } (b \text{ XOR } ci) = (a \text{ XOR } b) \text{ XOR } ci
\end{aligned}$$

Carry-out Karnaugh Map and Boolean Equation

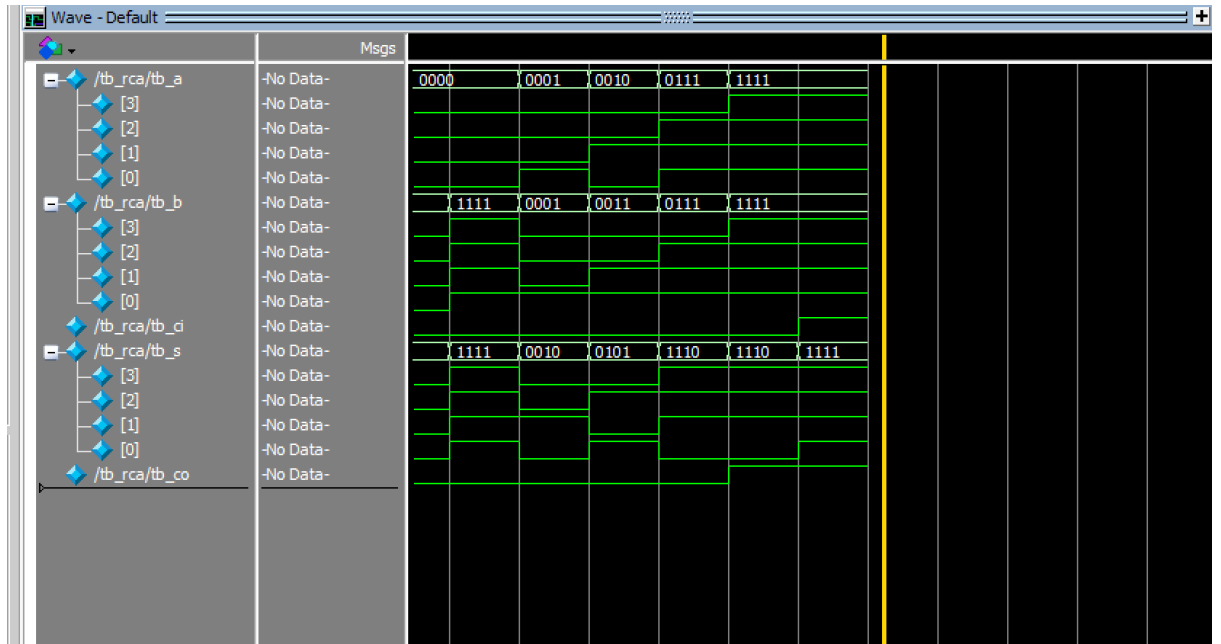
ab \ ci	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$\begin{aligned}
\text{Cout} &= a'bci + ab'ci + abci' + abci \\
&= ci(a'b + ab') + ab(ci' + ci) \\
&= ci(a \text{ XOR } b) + ab
\end{aligned}$$



4. 설계 검증 및 실험 결과

A. 시뮬레이션 결과



a: 0000 b: 1111 ci: 0 모든 RCA에서 S값이 출력 되는지 확인.

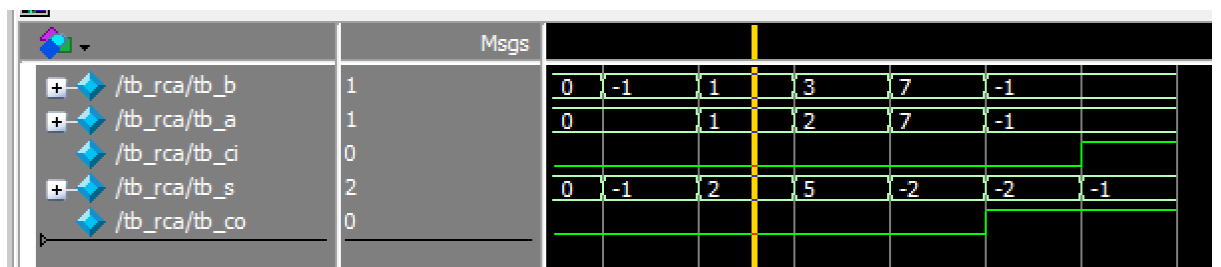
a: 0001 b: 0001 ci: 0 rca0 의 cout rca1 cin 값 전달 확인.

a: 0010 b: 0011 ci: 0 rca1 의 cout rca2 cin 값 전달 확인.

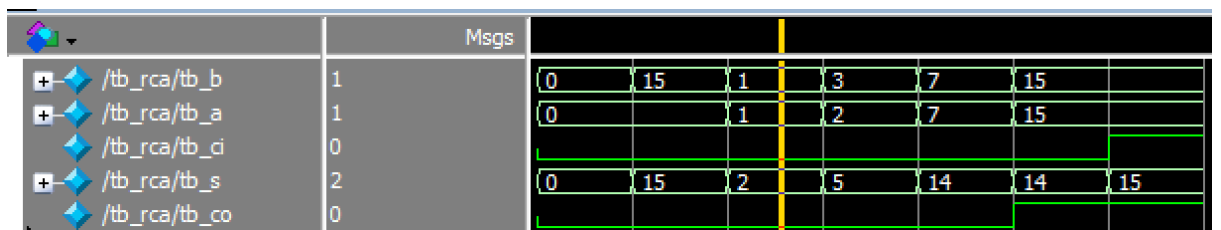
a: 0010 b: 0011 ci: 0 rca1 의 cout rca2 cin 값 전달 확인.

a: 0111 b: 0111 ci: 0 rca2 의 cout rca3 cin 값 전달 확인.

a: 1111 b: 1111 ci 1 모든 rca의 ci co s값 변화 확인 및 rca4 cout 값 출력 확인.



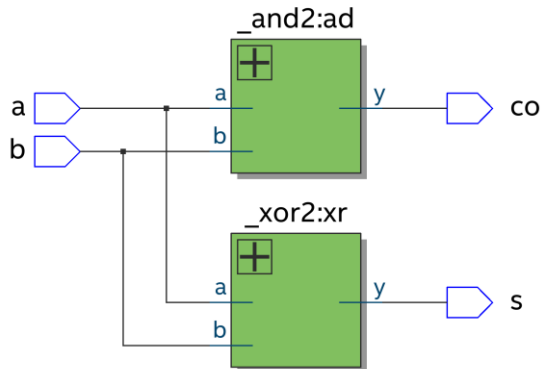
decimal



unsigned

demical로 했을 때 1111과 같이 최상위 비트가 1인 경우 음수로 출력 되는 것을 확인 할 수 있고, unsigned의 경우 4bit 음수 없이 0~15까지의 수로 표현 됨을 확인 했다.

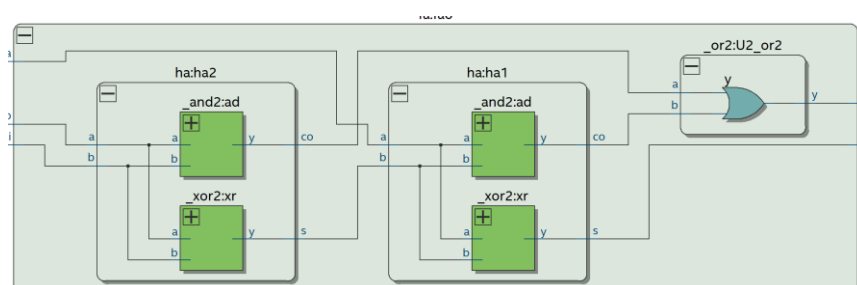
합성(synthesis) 결과



Half-adder

Flow Summary	
Flow Status	Successful - Wed Sep 27 09:15:29 2023
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	rca4
Top-level Entity Name	ha
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	2 / 41,910 (< 1 %)
Total registers	0
Total pins	4 / 499 (< 1 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

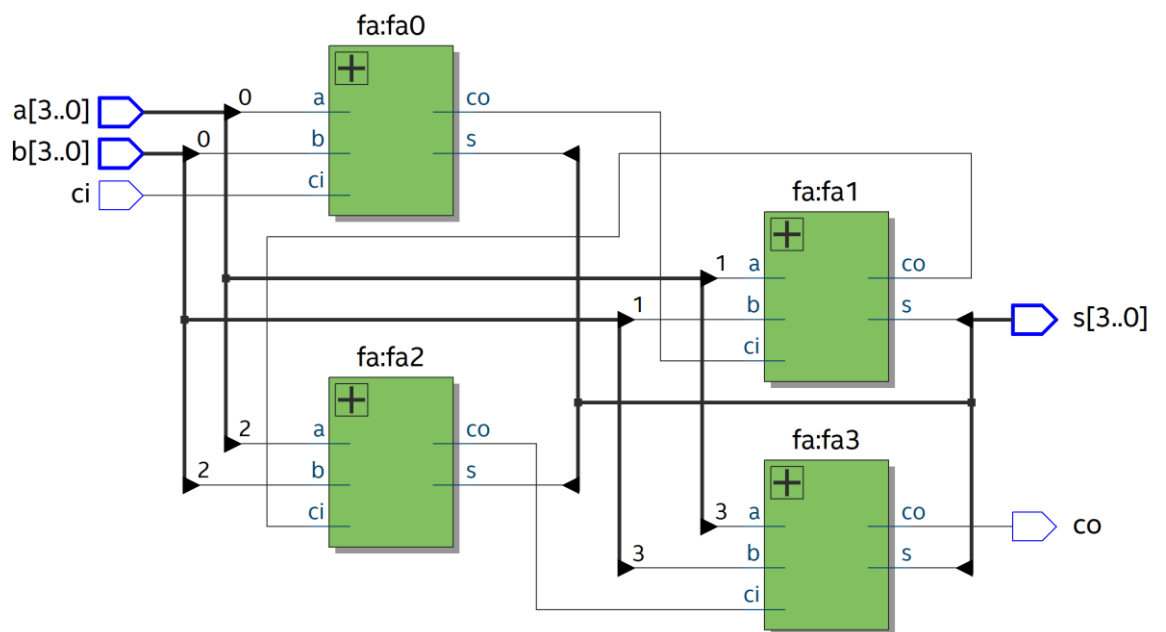
Total pins : input(a, b) + output(co, s)



Full adder

rca4.v		
fa.v		
Compilation Report - rca4		
Table of Contents		
Flow Summary	<<Filter>>	
Flow Settings	Flow Status	
Flow Non-Default C	Successful - Wed Sep 27 09:09:31 2023	
Flow Elapsed Time	Quartus Prime Version	
Flow OS Summary	18.1.0 Build 625 09/12/2018 SJ Lite Edition	
Flow Log	Revision Name	
> Analysis & Synthesi	rca4	
> Fitter	Top-level Entity Name	
> Assembler	fa	
> Timing Analyzer	Family	
> EDA Netlist Writer	Cyclone V	
Flow Messages	Device	
Flow Suppressed M	5CSXFC6D6F31C6	
	Timing Models	
	Final	
	Logic utilization (in ALMs)	
	2 / 41,910 (< 1 %)	
	Total registers	
	0	
	Total pins	
	5 / 499 (1 %)	
	Total virtual pins	
	0	
	Total block memory bits	
	0 / 5,662,720 (0 %)	
	Total DSP Blocks	
	0 / 112 (0 %)	
	Total HSSI RX PCSs	
	0 / 9 (0 %)	
	Total HSSI PMA RX Deserializers	
	0 / 9 (0 %)	
	Total HSSI TX PCSs	
	0 / 9 (0 %)	
	Total HSSI PMA TX Serializers	
	0 / 9 (0 %)	
	Total PLLs	
	0 / 15 (0 %)	
	Total DLLs	
	0 / 4 (0 %)	

Total pin: input(a, b, ci) + output(co, s)



4bit ripple carry adder

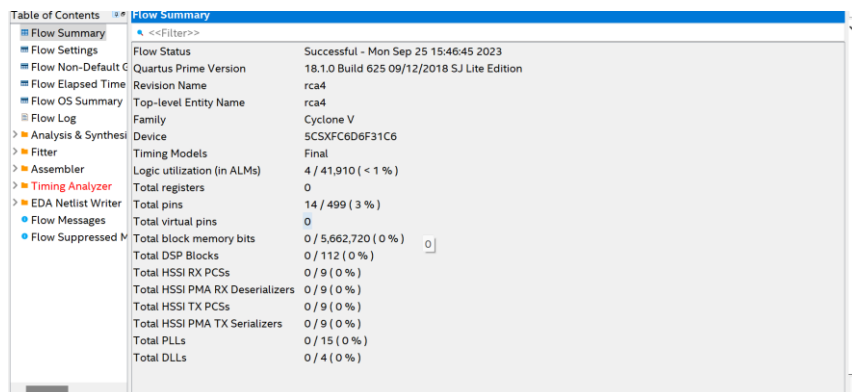


Table of Contents	Flow Summary
Flow Summary	<<Filter>>
Flow Settings	Flow Status
Flow Non-Default C	Quartus Prime Version
Flow Elapsed Time	Revision Name
Flow OS Summary	Top-level Entity Name
Flow Log	Family
Analysis & Synthesi	Device
Fitter	Timing Models
Assembler	Logic utilization (in ALMs)
Timing Analyzer	Total registers
EDA Netlist Writer	Total pins
Flow Messages	Total virtual pins
Flow Suppressed M	Total block memory bits
	Total DSP Blocks
	Total HSSI RX PCSs
	Total HSSI PMA RX Deserializers
	Total HSSI TX PCSs
	Total HSSI PMA TX Serializers
	Total PLLs
	Total DLLs

Total pin = input a 4pin, input b 4pin, output s 4pin

5. 고찰 및 결론

A. 고찰

- 테스트 벤치를 돌릴 때 모듈을 rca4라 설정해놓고 rca U0_rca라고 선언했다.
하지만 에러코드는 발생하지 않고 단지 테스트벤치 화면이 이상하게 나오는 것을 확인했다. 존재하지 않는 모듈의 테스트 벤치를 설정해도 에러가 발생하지 않음을 알게 되었다.
- 이전 실험에서 모듈을 구성할 때 $fa(a,b,ci,s,co)$ 이런 식으로 값을 넣어 명시적으로 모듈의 어떤 변수에 값을 넣었는지 보기 어려웠는데, 이번 실험에서는 $fa1(.a(a[1]),.b(b[1]),.ci(c[0]),.s(s[1]),.co(c[1]))$ 이런 방법으로 코드를 써보니 가시적으로 훨씬 편했다.

B. 결론

32bit rca 구현을 위해선 4bit rca모듈 8개가 필요하다. 각 4bit rca모듈의 cout을 다음 rca 모듈의 cin의 입력 값으로 받으면 쉽게 32bit rca가 구현 가능하다 하지만 32개의 rca 값이 순서대로 전달되어야 값을 출력하기 때문에 시간이 오래 걸린다는 단점이 존재할 것이다.

6. 참고문헌

가산기 / <https://ko.wikipedia.org/wiki/%EA%B0%80%EC%82%B0%EA%B8%B0>

이준환/디지털논리회로2/광운대학교/2023

이형근/컴퓨터공학기초실험2/광운대학교/2023