

컴퓨터 공학 기초 실험2 보고서

실험제목: Assignment_02 RCA

실험일자: 2020년 09월 14일 (월)

제출일자: 2020년 09월 21일 (월)

학 과: 컴퓨터정보공학부

담당교수: 공진흥 교수님

실습분반: 월요일 0, 1, 2

학 번: 2019202052

성 명: 김 호 성

1. 제목 및 목적

A. 제목

Half Adder, Full Adder, Ripple Carry adder

B. 목적

- Half Adder에 대해 이해하고, 프로그래밍한다.
- Full Adder에 대해 이해하고, 프로그래밍한다.
- 4-bit Ripple Carry Adder를 만들어 본다.

2. 원리(배경지식)

Two's complement

2's complement란, 1의 보수에 1을 더한 것이라고 말할 수 있다. 또한 1의 보수는 2의 보수보다 1이 작다고 말할 수 있다. 2's complement를 사용하는 방법으로는, 감수에 2의 보수를 취해주고, 피감수와 더해주면 된다. 이때, 덧셈을 하면서 발생하는 마지막 자리 올림수는 무시해준다

RCA Boolean equation 와 카르노맵 (1bit RCA와 동일하다. = Full Adder)

4 bit RCA

input			output	
a	b	C _i	S	C _o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

K-map (S)

C _i \ a b	00		01		10		11	
	0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0	1

$\therefore S = A \oplus B \oplus C$

K-map (C_o)

C _i \ a b	00		01		10		11	
	0	1	0	1	0	1	0	1
0	0	0	0	1	1	1	1	0
1	1	1	1	0	0	0	0	1

$\therefore C_{o} = AB + AC_{in} + BC_{in}$

3. 설계 세부사항

Half Adder

- 입출력

Input: a, b

Output: S(sum), co(Carry out)

- 진리표

➤ 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

➤ Karnaugh Map and Boolean Equation

a \ b	0	1
0	0	0
1	0	1

Carry out co = AB

a \ b	0	1
0	0	1
1	1	0

Sum s = $A'B + AB'$ (XOR)

Full Adder

- 입출력

Input: a, b, ci (Carry in)

Output: s(Sum), co (Carry out)

- 진리표

➤ Truth Table & Karnaugh Map

✓ 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

➤ Karnaugh Map and Boolean Equation

✓ About Sum - s

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

$$\text{Sum } s = \underline{A'B'C(\text{in}) + A'BC(\text{in})' + AB'C(\text{in})' + ABC(\text{in}) = (\text{XOR } ABC(\text{in}))}$$

(직접 boolean equation 을 작성하여 본다. - 레포트에 정리)

➤ About Carry Out - co

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

$$\text{Carry out co} = \underline{AB + AC(\text{in}) + BC(\text{in})}$$

(직접 boolean equation 을 작성하여 본다. - 레포트에 정리)

Ripple Carry Adder

- 입출력

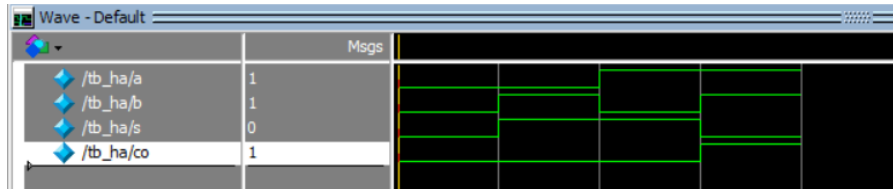
Input: a(4bits), b(4bits), ci (carry in)

Output: s(4bits), co (carry out)

4. 설계 검증 및 실험 결과

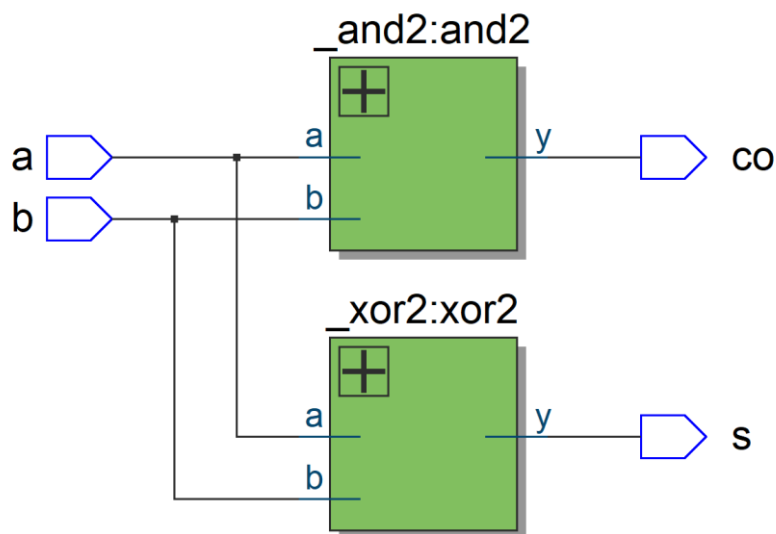
Half Adder

A. 시뮬레이션 결과

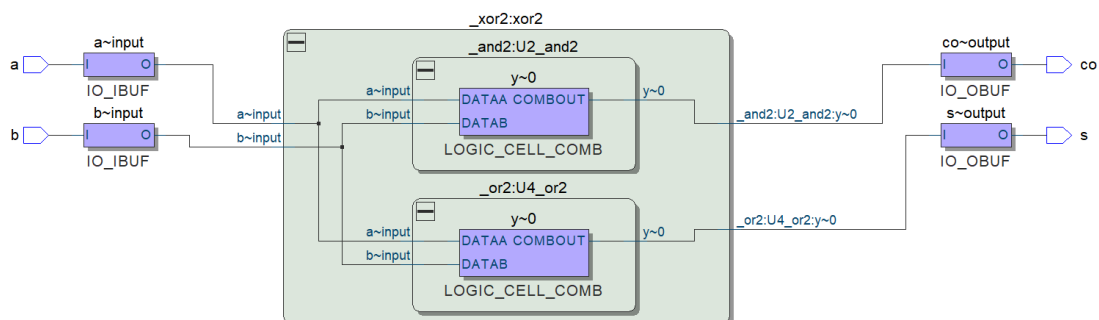


pdf파일과 진리표에 일치하는 것을 확인할 수 있다.

B. 합성(synthesis) 결과



Co(carry out)은 and에 s(Sum)이 XOR에 들어가는 것을 확인했다.

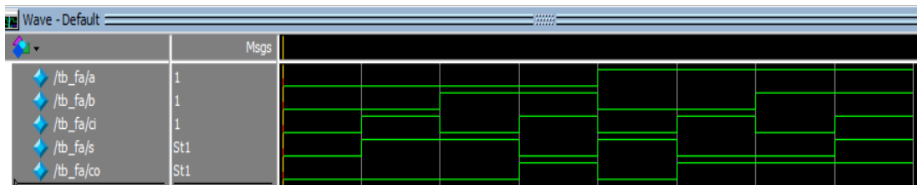


Flow Summary

Flow Status	Successful - Mon Sep 21 13:05:16 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	KimHoseong
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 %)

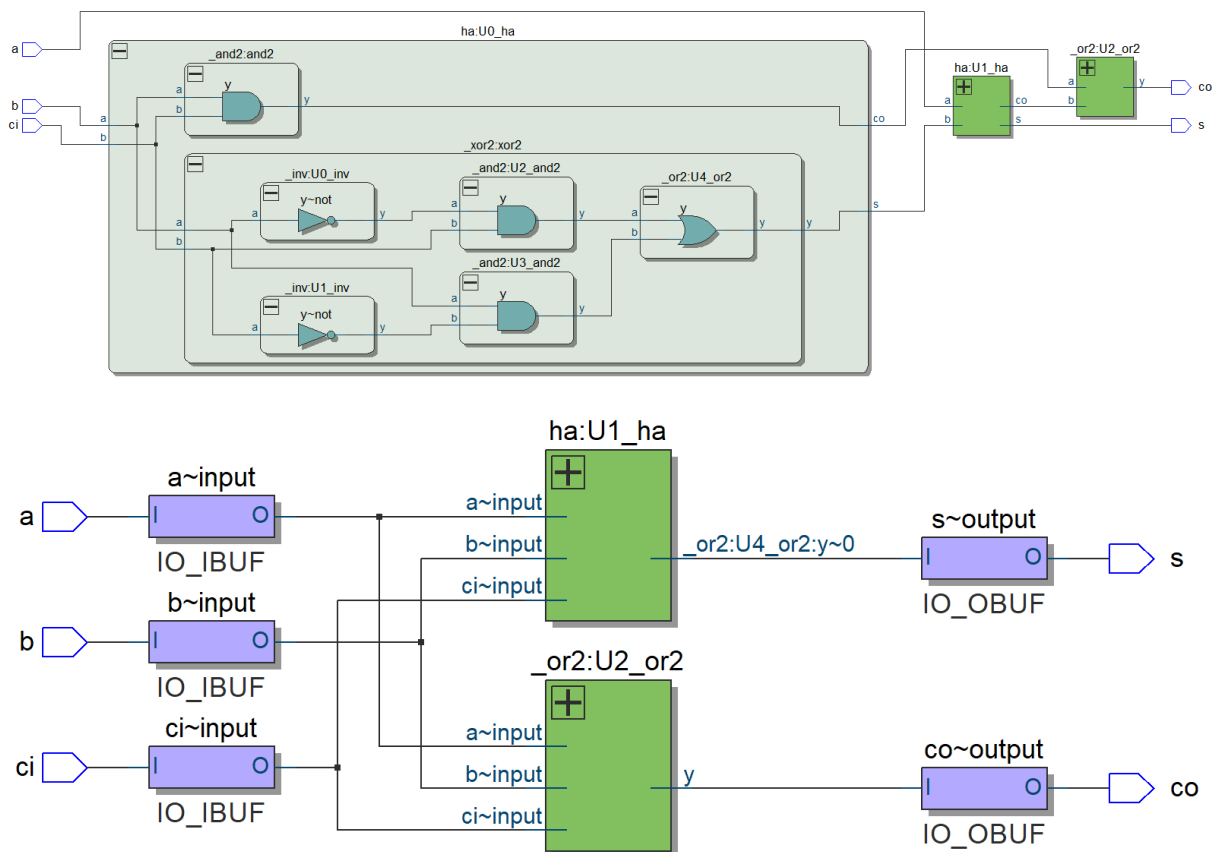
Full Adder

A. 시뮬레이션 결과



Pdf와 동일하게 나왔으며, 진리표와 순서만 다를 뿐 값은 전부 같다.

B. 합성(synthesis) 결과



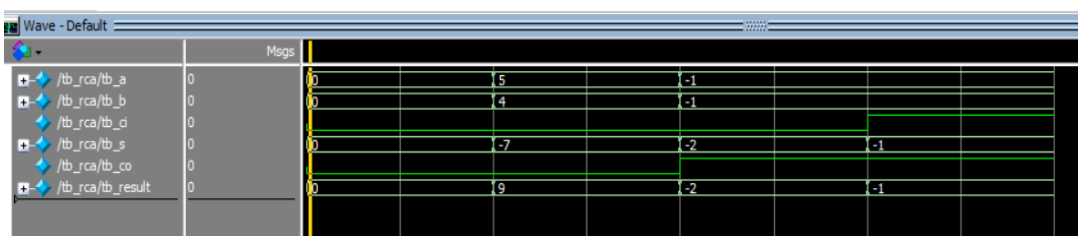
주어진 pdf와 동일하게 나온 것을 확인할 수 있다.

Flow Summary	
Flow Status	Successful - Mon Sep 21 13:05:16 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	KimHoseong
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 %)

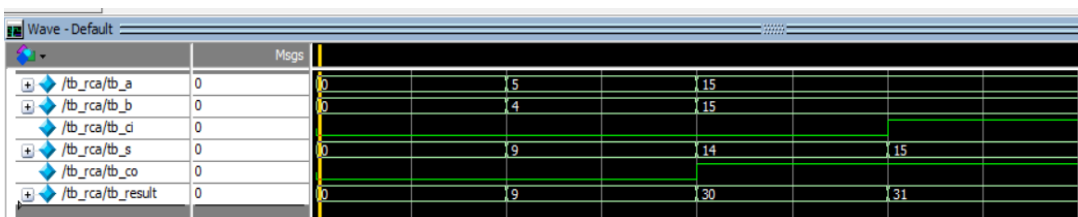
Ripple Carry Adder

A. 시뮬레이션 결과

Decimal

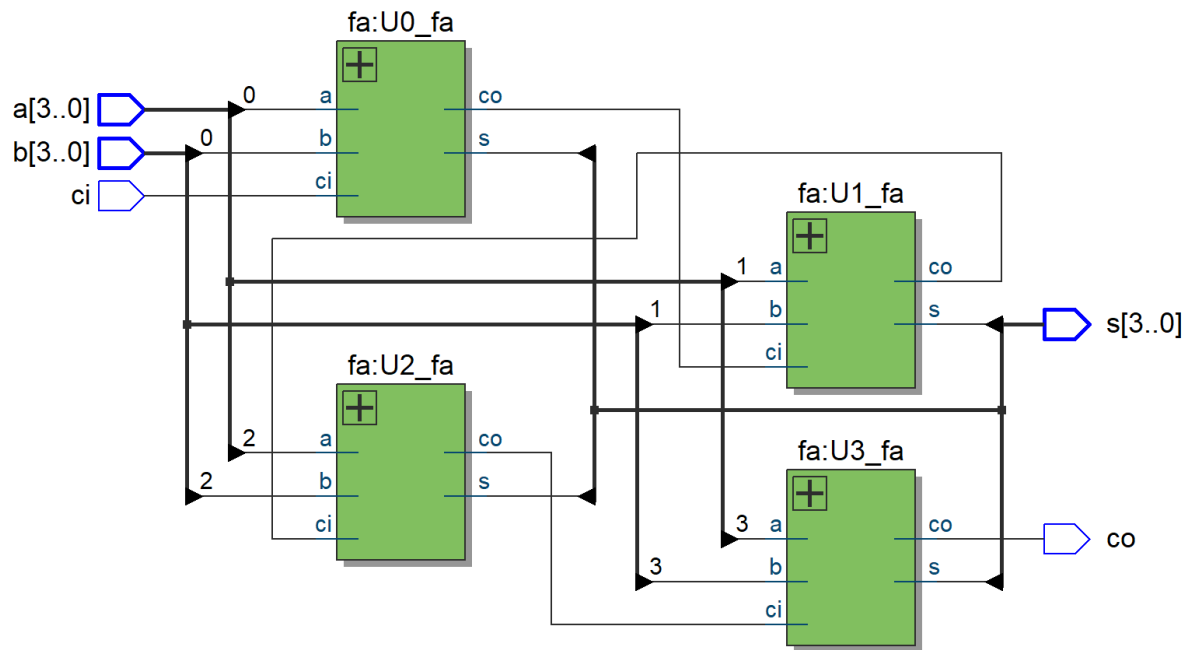


Unsigned



Unsigned로 했을 때 올바른 값이 나온다.

B. 합성(synthesis) 결과



Flow Summary	
Flow Status	In progress - Mon Sep 21 13:03:19 2020
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	rca
Top-level Entity Name	rca
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	4 / 41,910 (< 1 %)
Total registers	0
Total pins	14 / 499 (3 %)
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 %)

5. 고찰 및 결론

A. 고찰

Test bench를 작성할 때 입력 순서의 차이일 뿐인데, 값이 정확하게 나오지 않는 것 같은 불안함이 있었다. 하나하나 대조해보고 순서가 다른 것이라고 알았을 때 그제서야 안도할 수 있었으며, 생각보다 겁먹지만 않는다면 쉽게 해결할 수 있는 문제였다.

RCA의 경우 계산이 느리다는 단점이 있어, 후에 이를 보완한 CLA를 배운다는 것을 알았다. CLA의 경우 RCA보다 부품이 많이 든다는 단점이 있지만, 계산속도가 많은 bit를 다룰 때 압도적으로 빠르다는 것을 배웠다. RCA와 CLA를 적절히 사용한다면 좋은 효율을 가지는 하드웨어를 만들 수 있다고 느꼈다.

B. 결론

32-bit RCA설계하는 방법

RCA의 경우 Full Adder를 직선방향으로 이은 꼴이다. 그러므로 4-bit와 유사한 코드구조를 가지고, 비트 수를 [31:0] 으로 조정한다면 쉽게 만들 수 있다.

6. 참고문헌

공영호교수님/2주차강의와 3주차강의/광운대학교(온라인수업)/2020

<https://m.blog.naver.com/PostView.nhn?blogId=asd7979&logNo=30108134761&proxyReferer=https:%2F%2Fwww.google.com%2F>

(Half adder 진리표)