컴퓨터공학 기초 실험2

Lab #3

Ripple Carry Adder

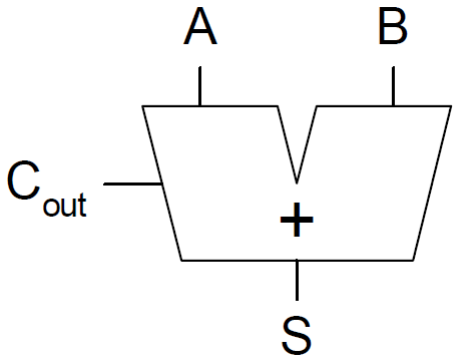
RIPPLE CARRY ADDER

Half Adder

- Half adder는 2개의 1-bit 입력을 받아 sum과 carry out을 출력하는 가산기

input		Output	
a	b	co	Sum s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Truth Table



or

Symbol

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 = ab′ + a′b = a ⊕ b

Karnaugh Map & Boolean Equation

Full Adder

- Full adder는 2개의 1-bit 입력과 1개의 1-bit carry in을 입력으로 받아 sum과 carry out을 출력하는 가산기

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

Truth Table

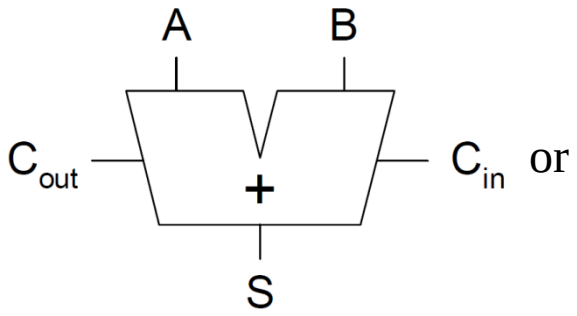
ci \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Boolean Equation

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

Boolean Equation

Karnaugh Map
& Boolean Equation

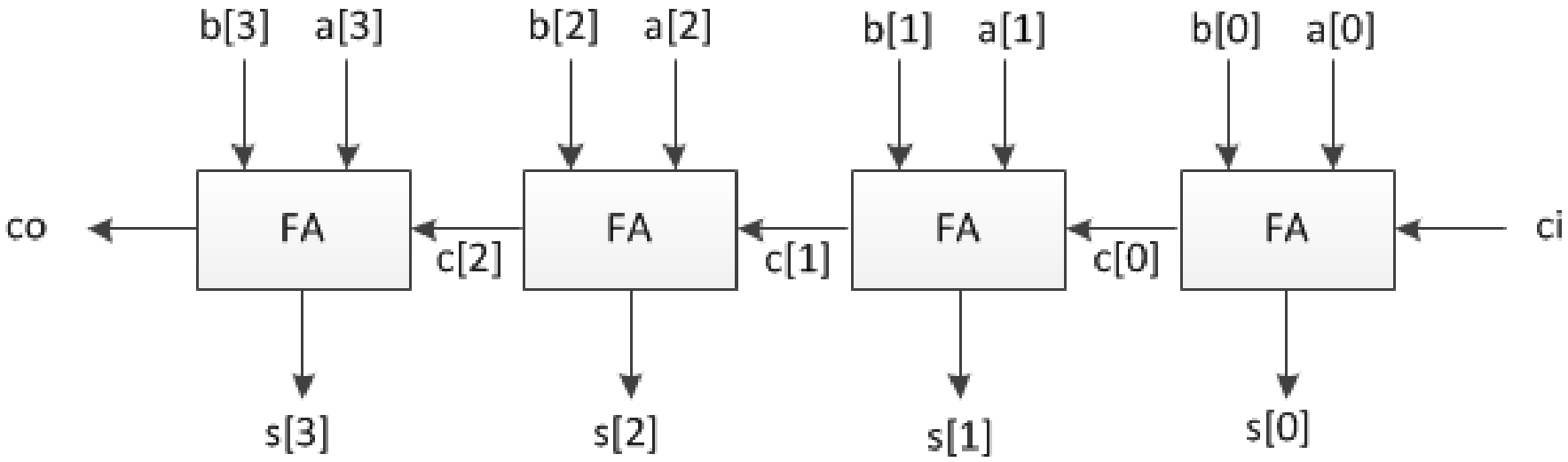


or

Symbol

Ripple Carry Adder

- Ripple Carry Adder는 여러 bit을 가지는 두 개의 수를 더하기 위한 간단한 형태의 가산기
  - ✓ Ripple Carry Adder는 더하고자 하는 수의 bit 개수만큼 full adder를 연결하여 구성



Half Adder & Full Adder

PRACTICE

Basic Logic Gates

- Inverter

```
module _inv(a,y);
input a;
output y;
assign y=~a;
endmodule
```

- 2-to-1 nand gate

```
module _nand2(a,b,y);
input a,b;
output y;
assign y=~(a&b);
endmodule
```

- 2-to-1 and gate

```
module _and2(a,b,y);
input a,b;
output y;
assign y=a&b;
endmodule
```

- 2-to-1 or gate

```
module _or2(a,b,y);
input a,b;
output y;
assign y=a|b;
endmodule
```

- 2-to-1 xor gate

```
module _xor2(a,b,y);
input a, b;
output y;
wire inv_a, inv_b;
wire w0, w1;
```

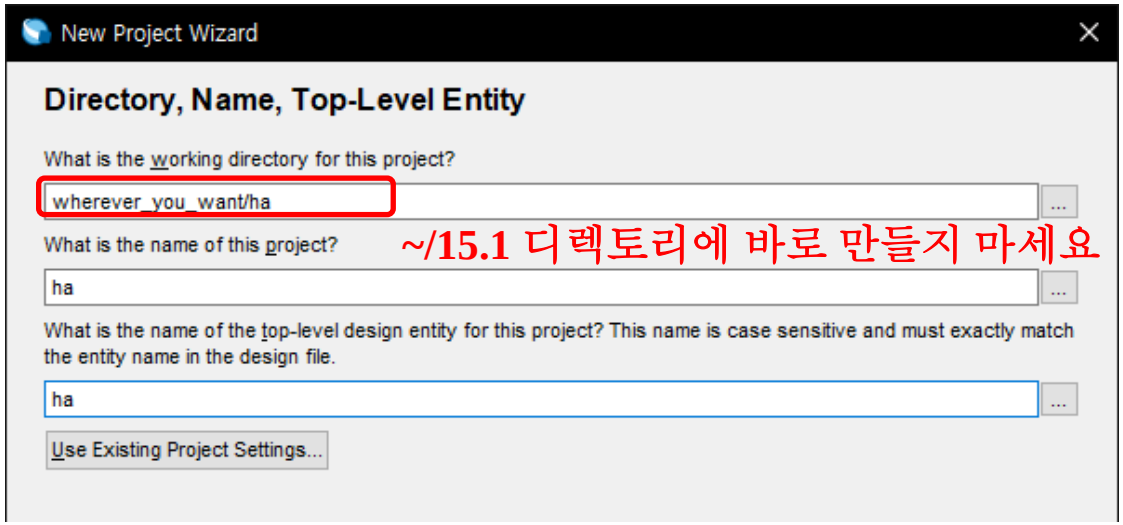
Instance

```
endmodule
```

- **gates.v로 저장한다.**

Half Adder Project Summary

- New Project Wizard
  - Project name : ha
  - Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)
- Verilog file
  - Add file : gates.v
  - New file : ha.v



Half Adder (1/3)

```
module ha(a, b, s, co);
  input a, b;
  output s, co ;
```

Instance

```
endmodule
```

ha.v로 저장

```
`timescale 1ns/100ps

module tb_ha;
reg a;
reg b;
wire s;
wire co;
```

Instance

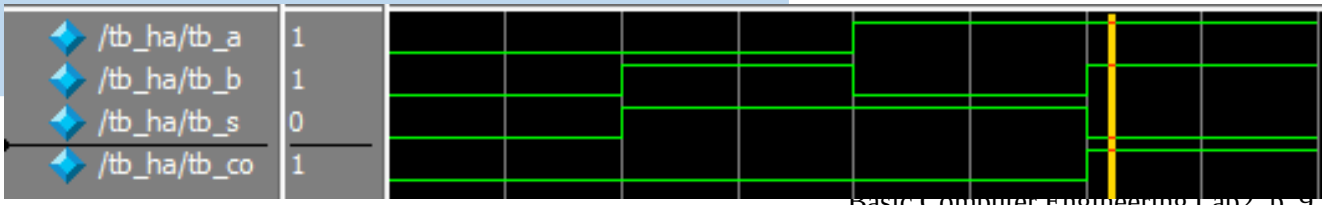
```
initial
begin
```

Testbench

```
end
endmodule
```

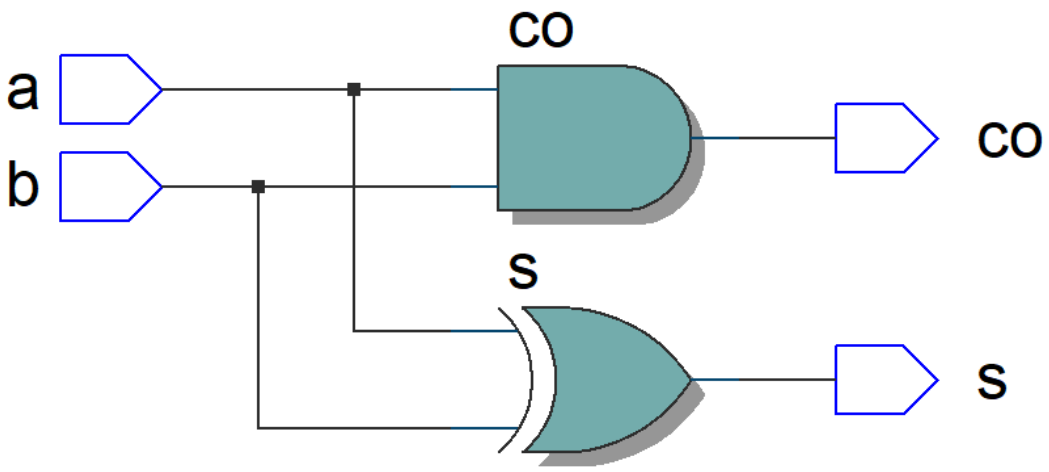
tb_ha.v로 저장

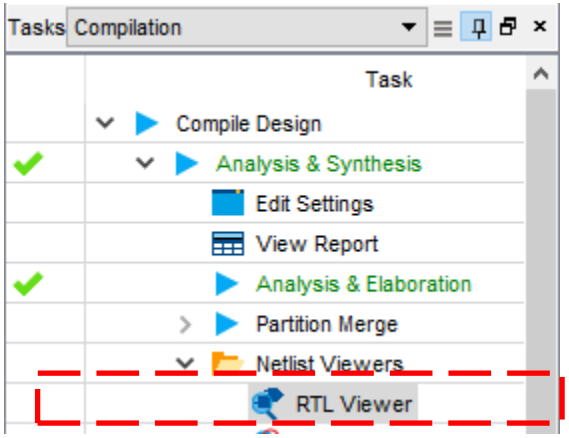
Waveform



Half Adder (2/3)

- RTL Viewer





Half Adder (3/3)

- Flow Summary

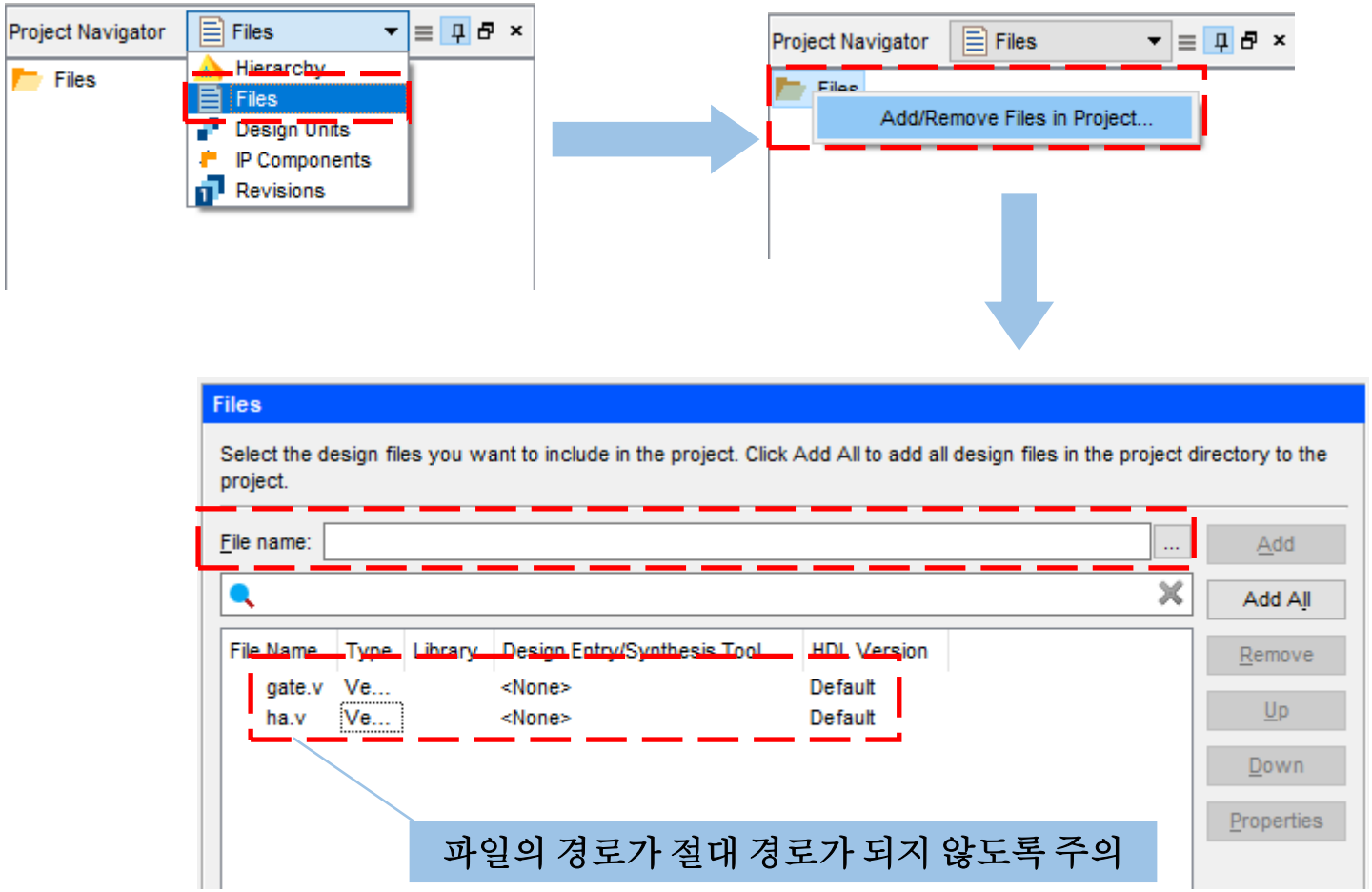
Analysis & Synthesis Summary	
Analysis & Synthesis Status	Successful - [redacted]
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	[redacted]
Top-level Entity Name	ha
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	4
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Full Adder Project Summary

- New Project Wizard
  - Project name : fa
  - Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)
- Verilog file
  - Add file : gates.v, ha.v
  - New file : fa.v

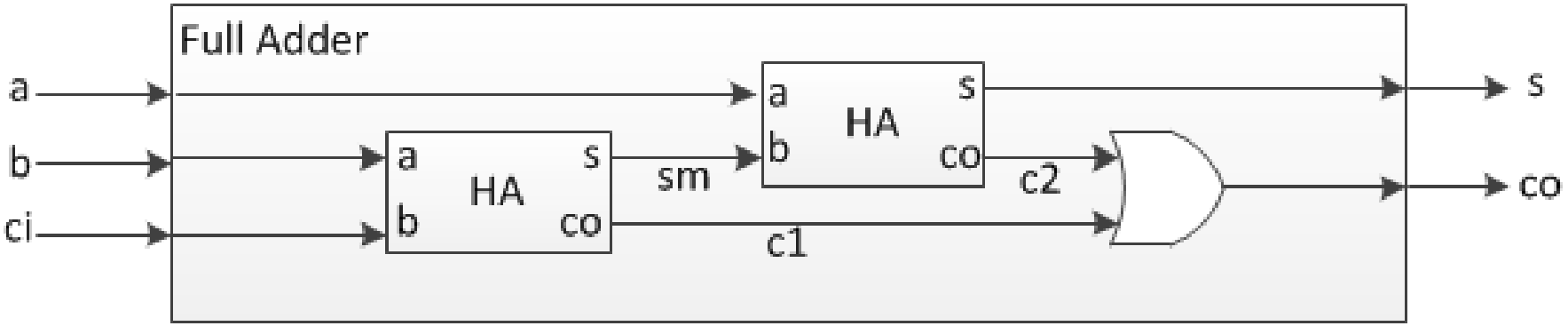
Add Files for Full Adder

- 앞서 만들었던 ha.v와 gates.v를 복사 후 추가하여 준다



Full Adder (1/4)

- Structural Description




```
module fa(a,b,cis,co);
input a,b,ci;
output s, co;
wire c1,c2,sm;
```

Instance

```
Endmodule
```

- ✓ fa.v로 저장

Full Adder (2/4)

- tb_fa.v로 저장

```
`timescale 1ns/100ps

module tb_fa;
reg a, b, ci;
wire s, co;
```

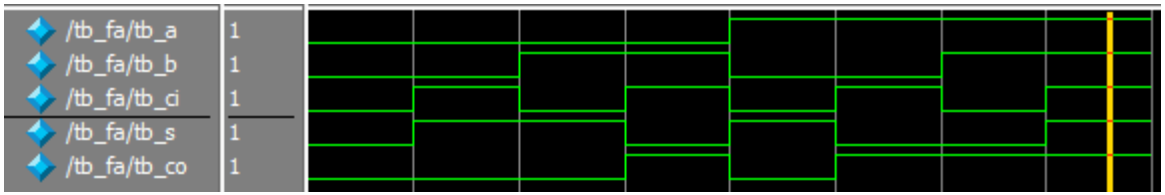
Instance

```
initial
begin
```

Testbench

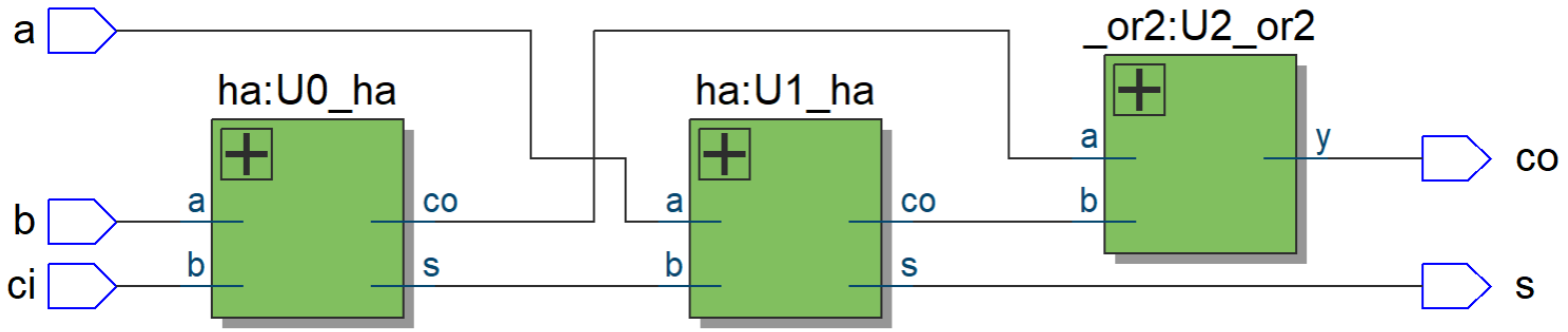
```
end
endmodule
```

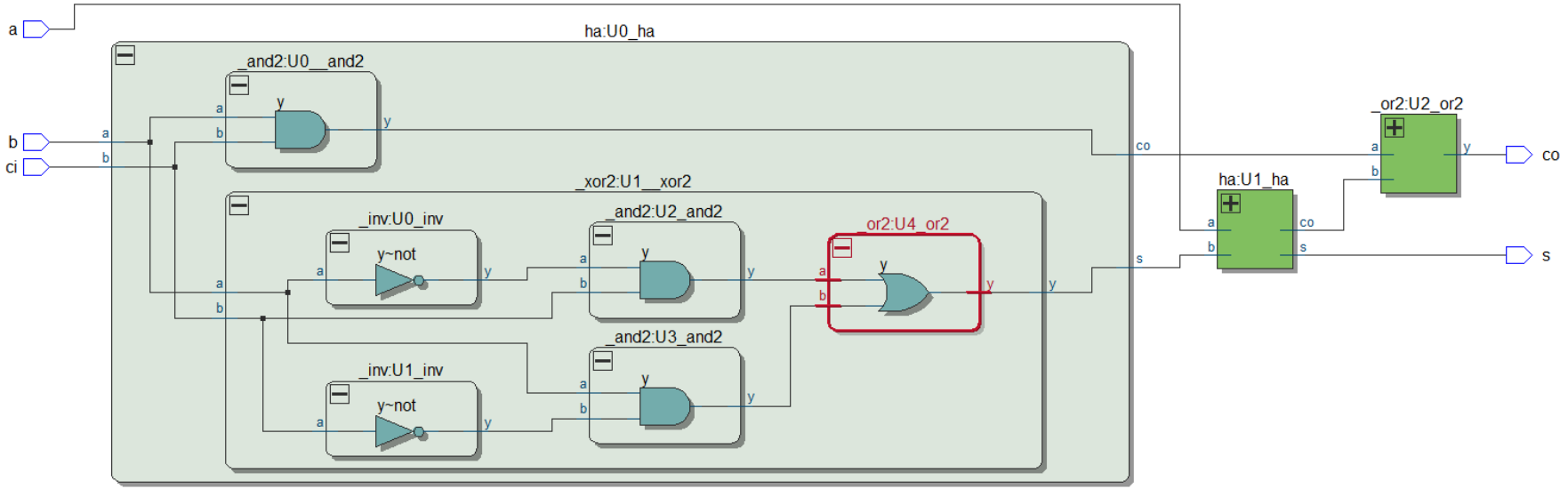
- Waveform



Full Adder (3/4)

- RTL Viewer





Full Adder (4/4)

- Flow Summary

Flow Summary	
Flow Status	Successful - [redacted]
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	[redacted]
Top-level Entity Name	fa
Family	Cyclone V
Device	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 %)

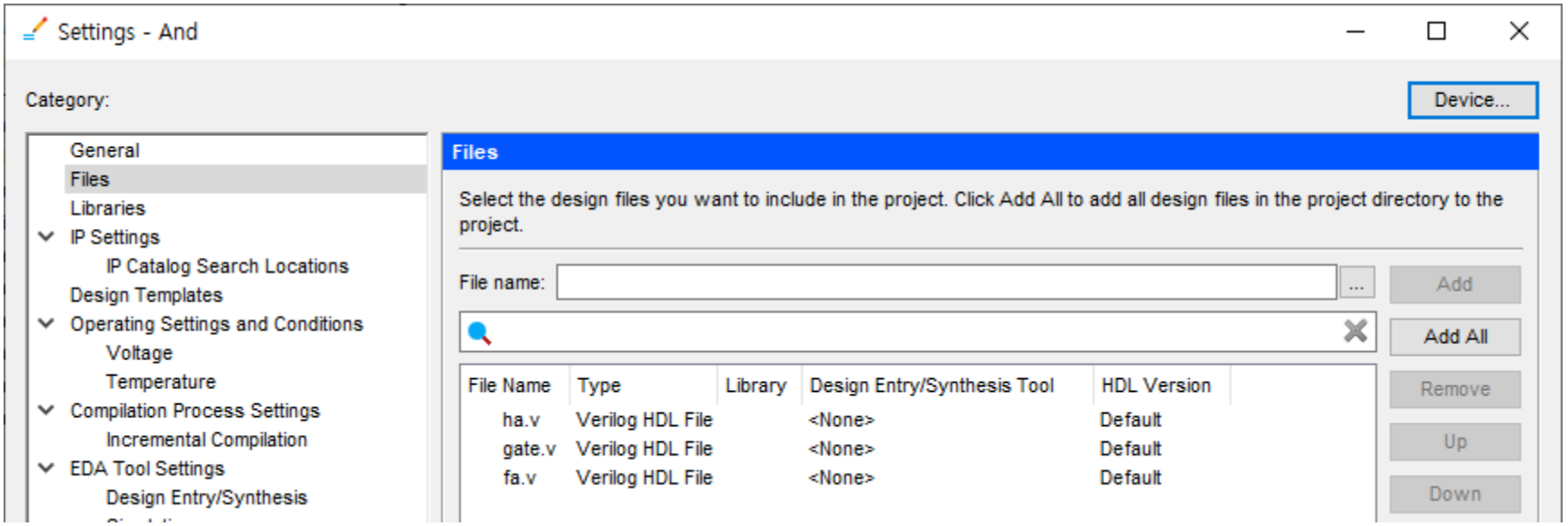
RIPPLE CARRY ADDER

Ripple Carry Adder Project Summary

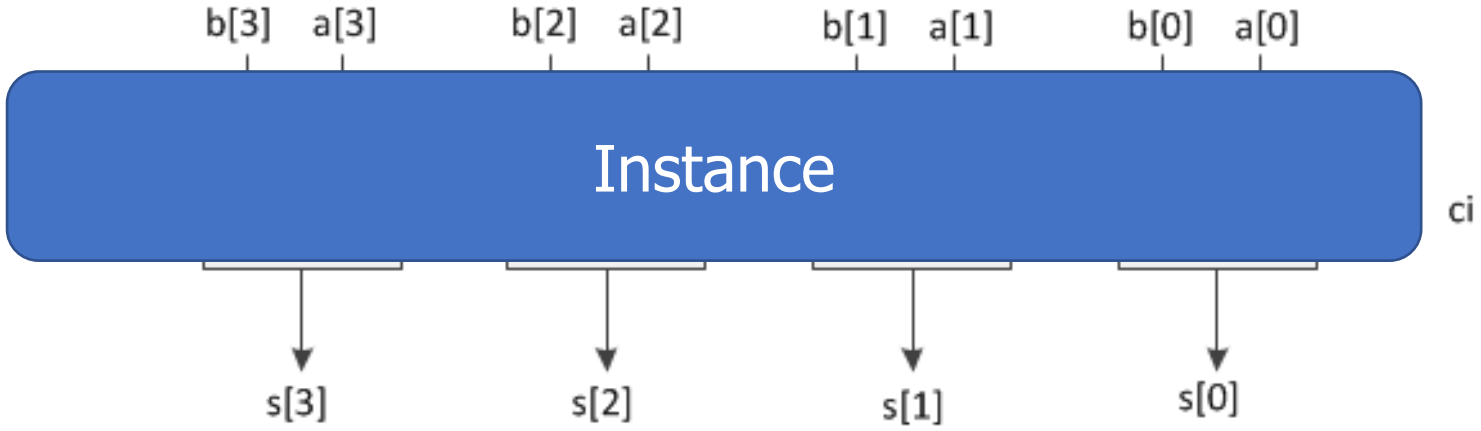
- New Project Wizard
  - Project name : rca
  - Family & Device : Cyclone V 5CSXFC6D6F31C6 (밑에서 6번째)
- Verilog file
  - Add file : gates.v, ha.v, fa.v
  - New file : rca.v

Add Files for RCA

- Files에 fa.v, ha.v, gates.v를 추가



Ripple Carry Adder (1/6)




```
module rca4(a,b,ci,s,co);
input [3:0] a,b;
input ci;
output [3:0] s;
output co;
wire [2:0] c;

  fa U0_fa(.a(a[0]), .b(b[0]), .ci(ci), .s(s[0]),.co(c[0]));
  fa U1_fa(.a(a[1]), .b(b[1]), .ci(c[0]),.s(s[1]), .co(c[1]));
  fa U2_fa(.a(a[2]), .b(b[2]), .ci(c[1]), .s(s[2]), .co(c[2]));
  fa U3_fa(.a(a[3]), .b(b[3]), .ci(c[2]), .s(s[3]), .co(co));

endmodule
```

Ripple Carry Adder (2/6)

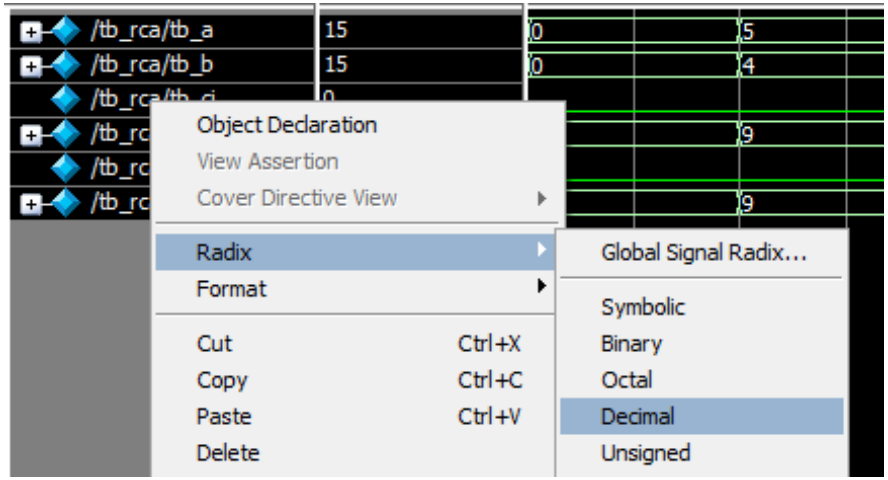
- tb_rca.v로 저장

```
`timescale 1ns/100ps

module tb_rca;
reg[3:0] tb_a, tb_b;
reg tb_ci;
wire[3:0] tb_s;
wire tb_co;
wire [4:0] tb_result;

Instance

initial
begin

Testbench

end

assign tb_result = {tb_co,tb_s};
endmodule
```

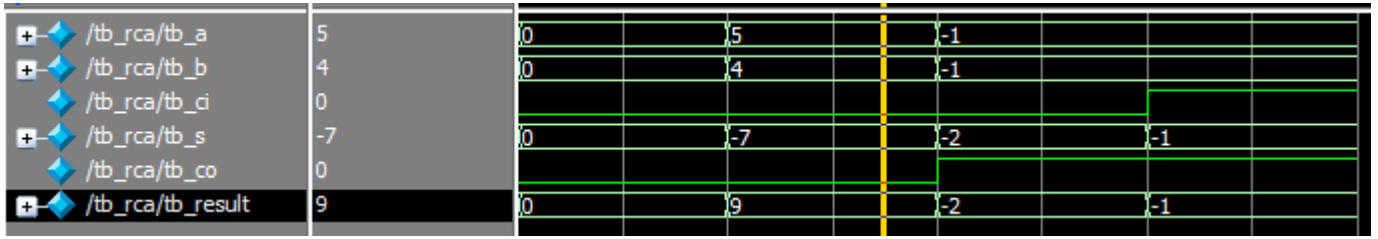
Sum과 carry out을 concatenation해서 결과를 함께 확인

Ripple Carry Adder (3/6)

- Waveform
  - ✓ 신호들을 모두 선택 → 마우스 오른쪽 클릭 → Radix → Decimal

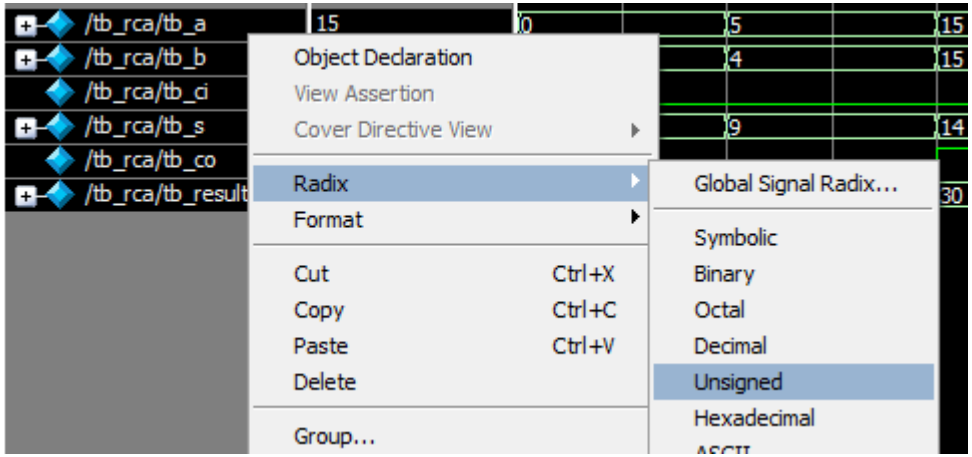


- Decimal results

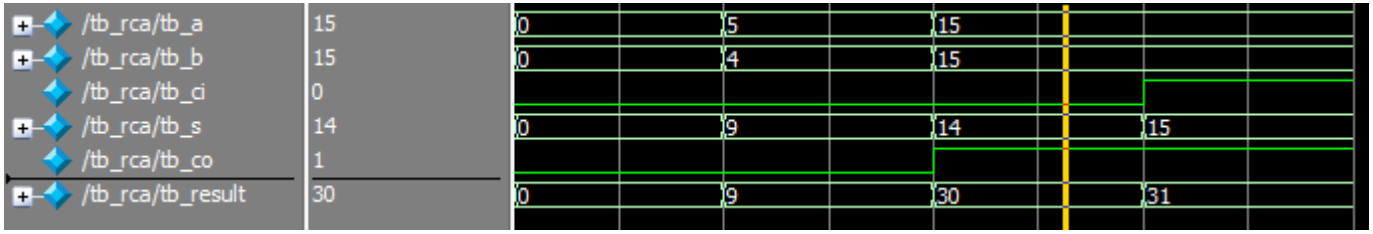


Ripple Carry Adder (4/6)

- Waveform
  - ✓ 신호들을 모두 선택 → 마우스 오른쪽 클릭 → Radix → Unsigned

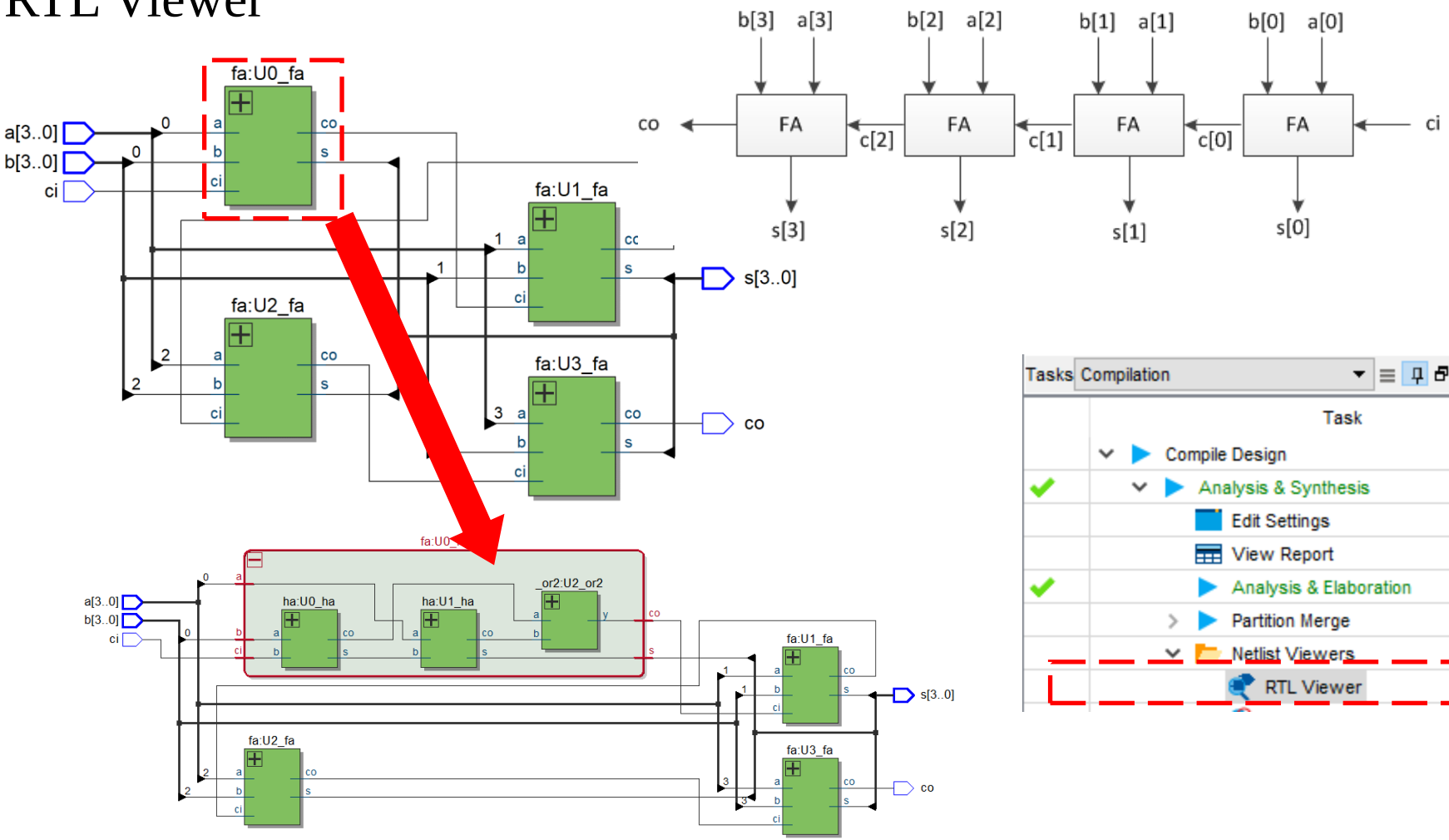


- Unsigned results



Ripple Carry Adder (5/6)

- RTL Viewer



Ripple Carry Adder (6/6)

- Flow Summary

Flow Summary	
Flow Status	Successful - [redacted]
Quartus Prime Version	15.1.0 Build 185 10/21/2015 SJ Lite Edition
Revision Name	[redacted]
Top-level Entity Name	rca
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	N/A until Partition Merge
Total registers	N/A until Partition Merge
Total pins	N/A until Partition Merge
Total virtual pins	N/A until Partition Merge
Total block memory bits	N/A until Partition Merge
Total PLLs	N/A until Partition Merge
Total DLLs	N/A until Partition Merge

Assignment 2

- Report
  - 자세한 사항은 Homework & Practice document 참고
- Submission
  - 과제 기한은 공지 참고
  - 늦은 숙제는 제출 이틀 후 까지만 받음(20% 감점)

채점기준

세부사항		점수	최상	상	중	하	최하
소스코드	Source code가 잘 작성 되었는가? (Structural design으로 작성되었는가?)	10	10	8	5	3	0
	주석을 적절히 달았는가? (반드시 영어로 주석 작성)	20	20	15	10	5	0
설계검증 (보고서)	보고서를 성실히 작성하였는가? (보고서 형식에 맞추어 작성)	30	30	20	10	5	0
	합성결과를 설명하였는가?	10	10	8	5	3	0
	검증을 제대로 수행하였는가? (모든 입력 조합, waveform 설명)	30	30	20	10	5	0
총점		100					

References

- Altera Co., www.altera.com/
- 공영호, 디지털논리회로2 강의자료, 광운대학교, 컴퓨터 공학과, 2022

Q&A

THANK YOU