

Lab. 05

Logic Design Lab.

Spring 2023

Prof. ChangGun Lee

(cglee@snu.ac.kr)

TA. Seonghyeon Park

TA. Jihwan Kim

TA. Hoyong Lee

(rubis.ld.ta@gmail.com)

Contents

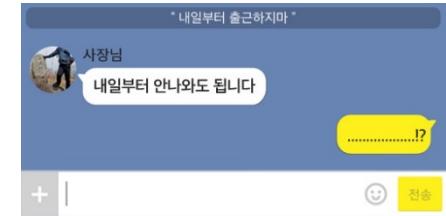
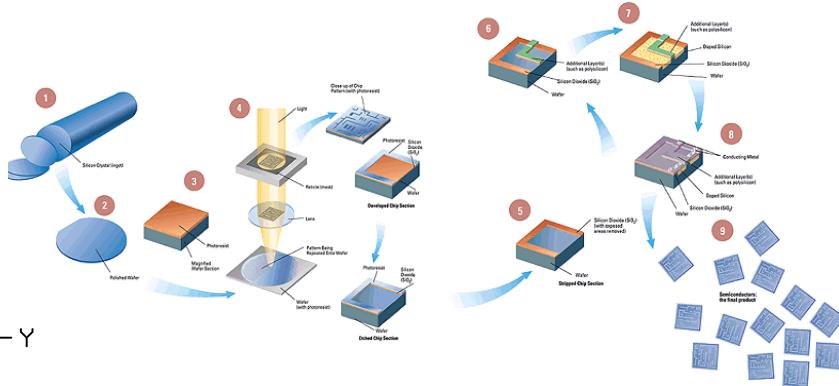
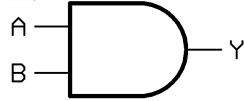
- **FPGA (Field Programmable Gate Array)**
- **SNU Logic Design Board**
- **Sample Implementation**
- **Lab**

FPGA

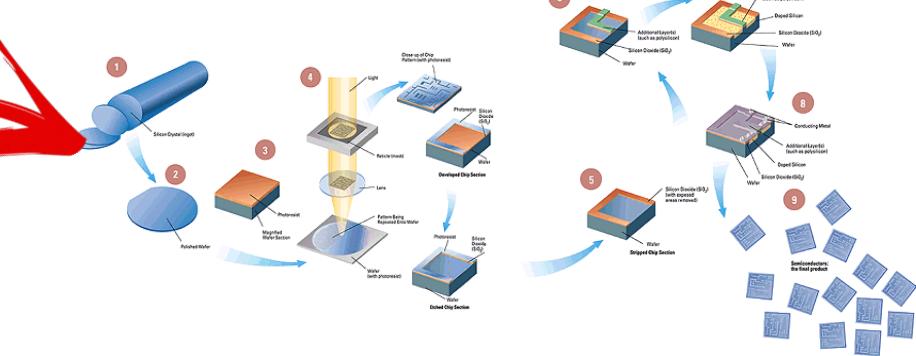
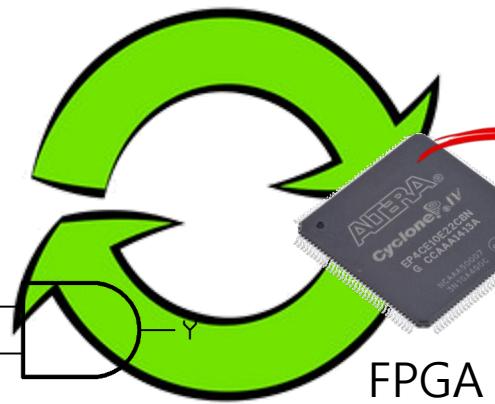
(Field Programmable Gate Array)

What is FPGA?

without FPGA

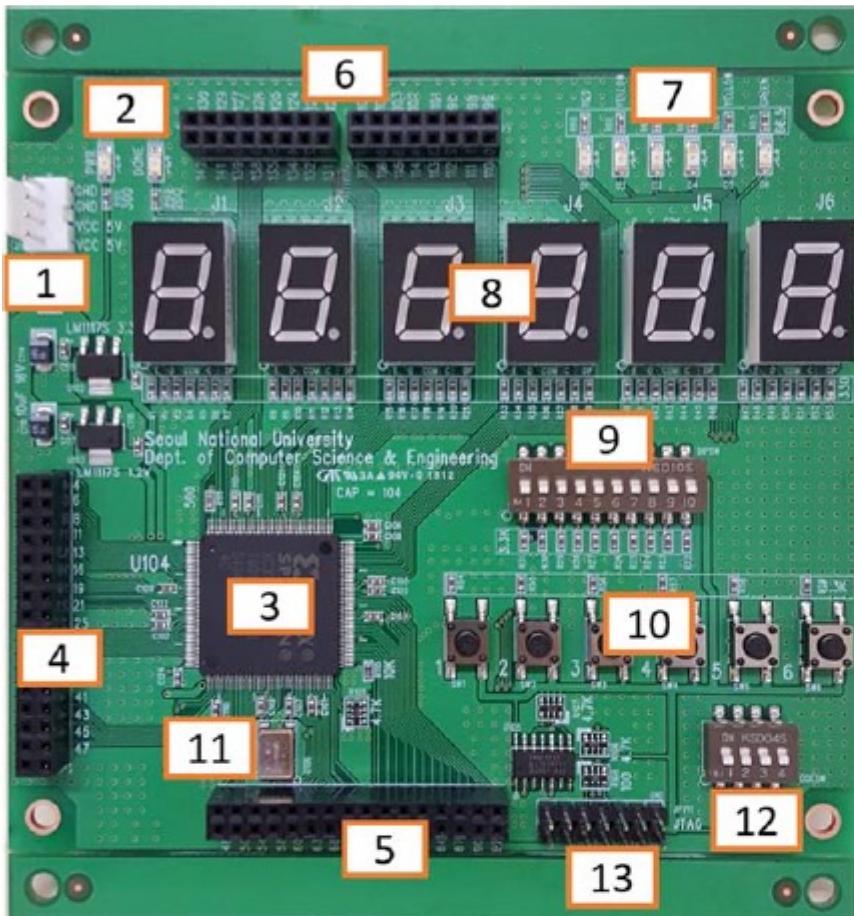


with FPGA



SNU Logic Design Board

SNU Logic Design Board



- ① Power Input : Molex Connector (DC 5V Only)
- ② Power, DONE LED
- ③ FPGA : XC3S50AN-4TQG144C
- ④ User I/O Ports (P1) : 16X2 2.54mm Pitch
- ⑤ User I/O Ports (P2) : 16X2 2.54mm Pitch
- ⑥ User I/O Ports (P3, P4) : 2 8X2 2.54mm Pitch
- ⑦ User Output LEDs : 6 output LEDs
- ⑧ User Output LEDs : 6 7-segment LEDs
- ⑨ User Input Switches : 10pin Dip Switch
- ⑩ User Input Switches : 6 Tactile Switches
- ⑪ Oscillator : 50MHz
- ⑫ Mode Select Switch
- ⑬ JTAG Header

SNU Logic Design Board

- I/O Connectors

Pin Num	Component	
3	7-Segment Display [J1]	A
4		B
5		C
6		D
7		E
8		F
10		G
11	7-Segment Display [J2]	A
12		B
13		C
15		D
16		E
18		F
19		G
20		A
21	7-Segment Display [J3]	B
24		C
25		D
27		E
28		F
29		G
30		1
31	DIP Switch [DipSW1]	2
32		3
33		4
41		5
42		6
43		7
44		8
45		9
46		10
47	Tactile Switch [SW1]	

Pin Num	Component	
48	Tactile Switch [SW2]	
49	Tactile Switch [SW3]	
50	Tactile Switch [SW4]	
51	Tactile Switch [SW5]	
54	Tactile Switch [SW6]	
55	7-Segment Display [J4]	A
58		B
59		C
60		D
62		E
63		F
64		G
68	7-Segment Display [J5]	A
69		B
70		C
71		D
72		E
75		F
76		G
77	7-Segment Display [J6]	A
78		B
79		C
82		D
83		E
84		F
85		G
87	LED [D1]	Red
88	LED [D2]	Yellow
90	LED [D3]	Green
91	LED [D4]	Red
92	LED [D5]	Yellow
93	LED [D6]	Green

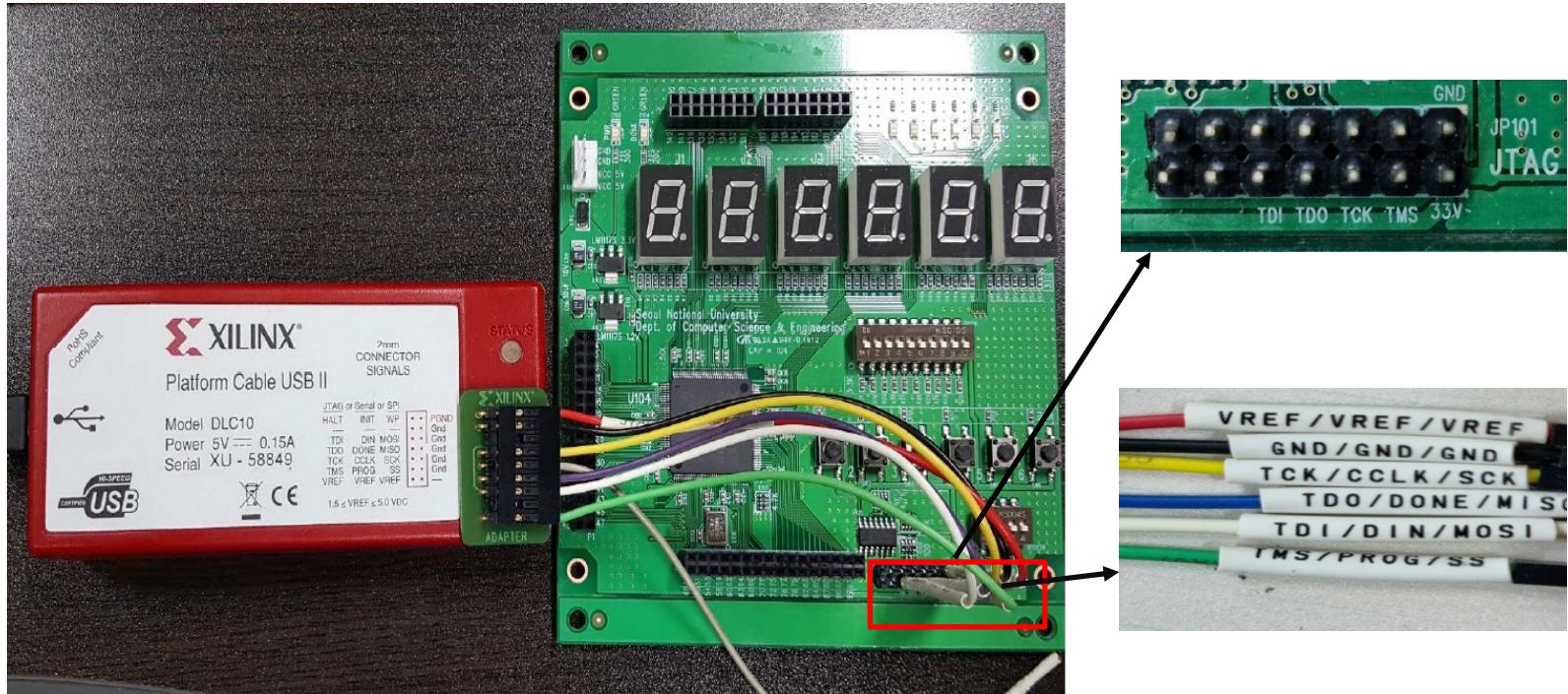
Pin Num	Component	
96	P3	120
98		121
99		124
101		125
102		126
103		127
104		129
105		130
110		131
111		132
112	P4	134
113		135
114		138
115		139
116		141
117		142

Pin	Component	
1	JP101	TMS
2		TDI
107		TDO
109		TCK
37	Mode SW	M1
38		M0
39		M2
144		PROG
67	FPGA	INIT
73		DONE
74		SUSPEND
35		
53	FPGA	Not Connected (Input Only/VREF)
80		
97		
123		
140		

Pin	Component	
57	VCCO	Clock 50MHz
14		
23		
40		
61		
86		
95		
119		
136		
36		
66	VCCAUX	
108		
133		
22		
52	VCCINT	
94		
122		
9		
17	GND	
26		
34		
56		
65		
81		
89		
100		
106		
118		
128		
137		

SNU Logic Design Board

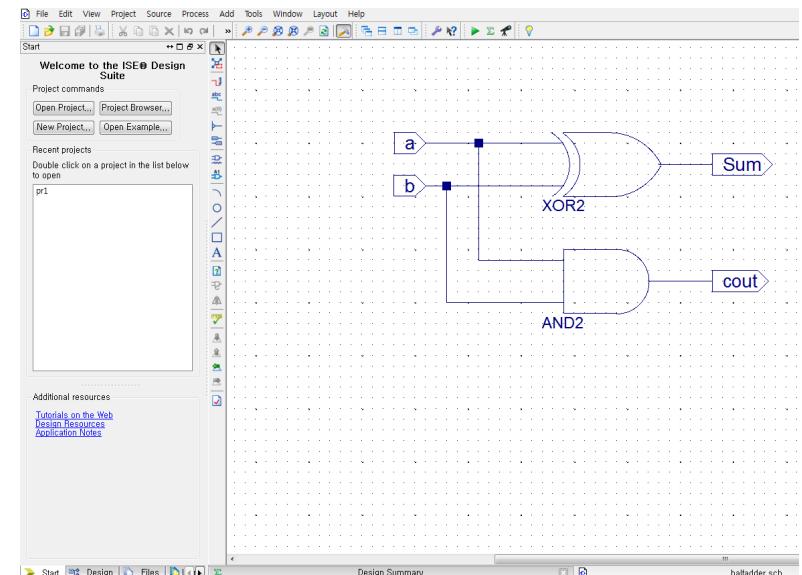
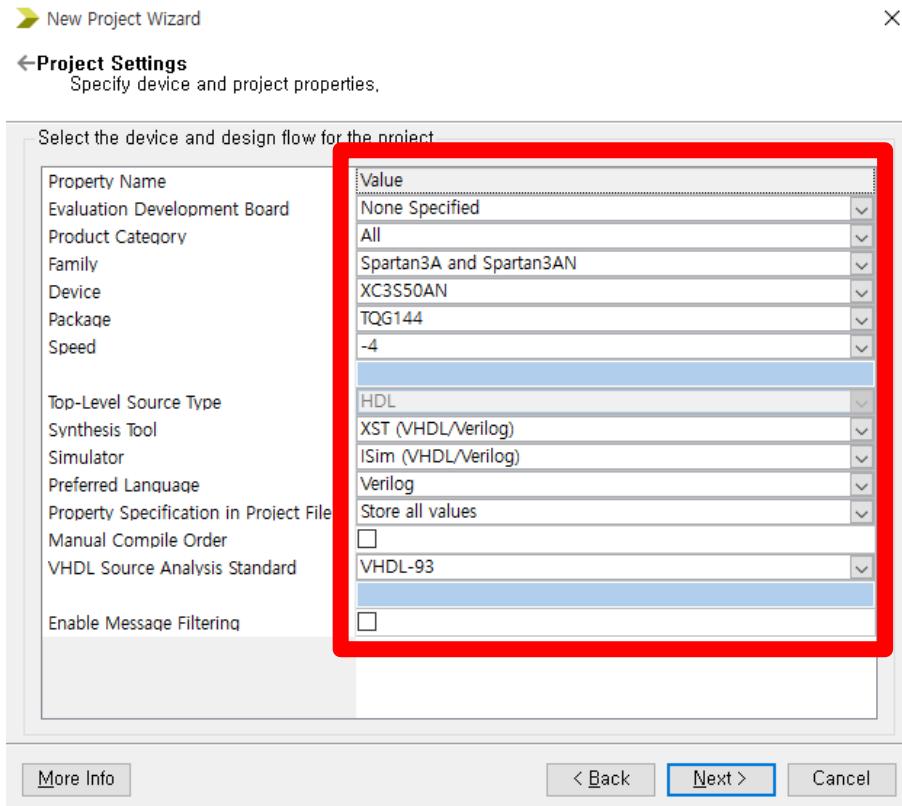
- **Connect to PC with Xilinx Platform Cable USB**
 - HALT/RST Pin(Gray pin) not used.
 - Make sure both board and USB are powered on.



Sample Implementation

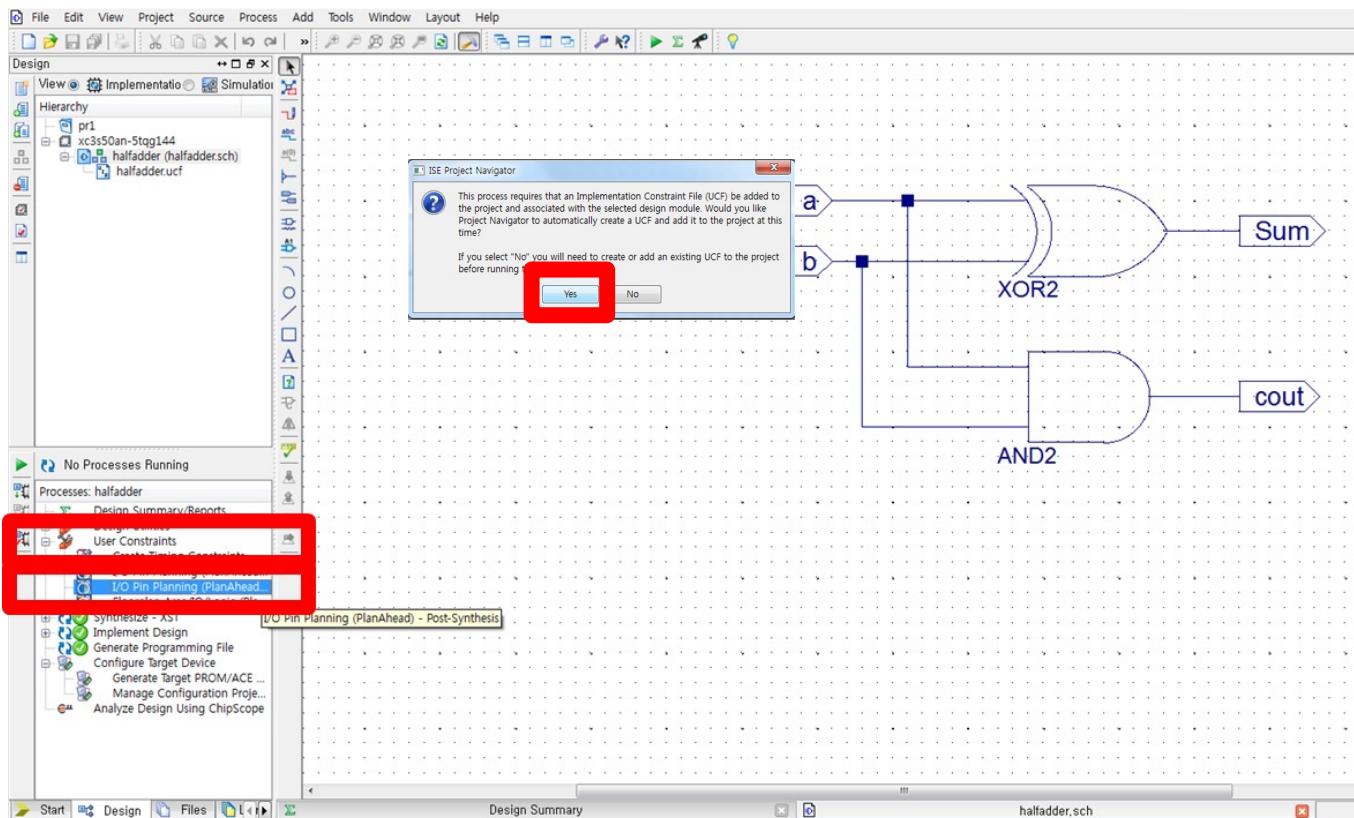
Sample Implementation

- Create new project, build a half adder schematic.
- For detailed settings, refer to guidelines in Lab03.



Sample Implementation

- Design tab > User Constraints > I/O Pin Planning(PlanAhead) – Post-Synthesis.
- Click 'Yes' to start PlanAhead, and to create an UCF file.



Sample Implementation

- Assign I/O pins for the half adder

The screenshot shows the Xilinx PlanAhead 14.7 interface with the following components:

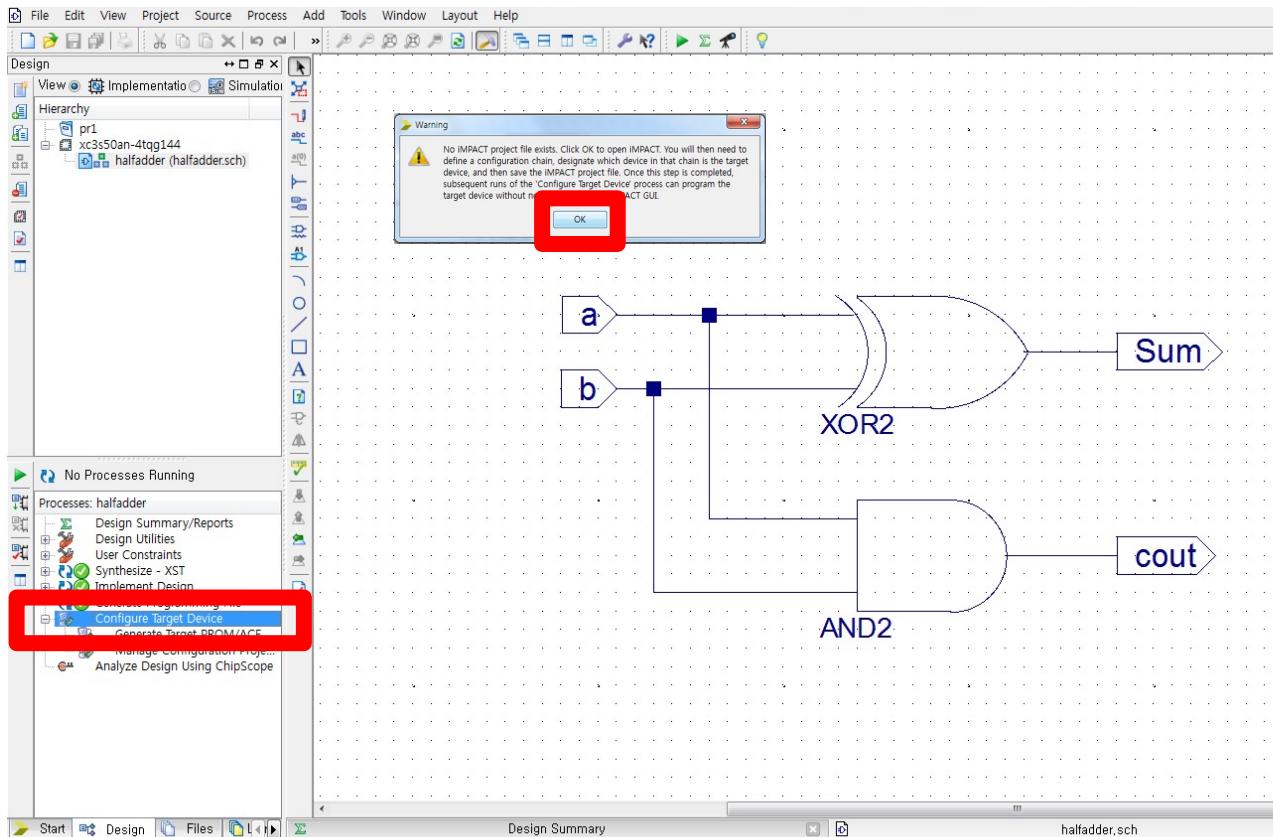
- Elaborated Design**: Shows a hierarchical tree with a **halfadder** component containing **Nets (4)** and **Primitives (2)**.
- I/O Port Properties**: Shows the **Sum** port with a **Name: Sum** field.
- Properties**: Shows the **Clock Regions** tab selected.
- RTL Schematic**: Displays a schematic diagram of the half adder circuit.
- Package Pins**: A table showing package pin assignments:

Pin Num	Component	
3		A
4		B
5	7-Segment Display [J1]	C
6		D
7		E
8		F
10		G
11	7-Segment Display [J2]	A
12		B
13		C
15	7-Segment Display [J3]	D
16		E
18		F
19		G
20		A
21	7-Segment Display [J5]	B
24		C
25		D
27	7-Segment Display [J6]	E
28		F
29		G
30		1
31		2
32		3
33	DIP Switch [DipSW1]	4
41		5
42		6
43		7
44		8
45		9
47	Tactile Switch [SW1]	
48	Tactile Switch [SW2]	
50	Tactile Switch [SW4]	A
51	Tactile Switch [SW5]	B
54	Tactile Switch [SW6]	C
55		D
58		E
59	7-Segment Display [J4]	F
60		G
62		A
63		B
64		C
68	7-Segment Display [J5]	D
69		E
70		F
71	7-Segment Display [J6]	G
72		A
75		B
76		C
77		D
78		E
79	7-Segment Display [J1]	F
82		G
83		A
84		B
87	LED [D1]	C
88	LED [D2]	D
91	LED [D4]	E
92	LED [D5]	F
93	LED [D6]	G
- I/O Ports**: A table showing I/O port properties:

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Str...	Slew Typ...
a	Input			P47		2 default (LVCM...				
b	Input			P48		2 default (LVCM...				
cout	Output			P87		1 default (LVCM...	2,500		12 SLOW	
Sum	Output			P88		default (LVCM...	2,500		12 SLOW	
- Tcl Console**, **Package Pins**, and **I/O Ports** tabs at the bottom.

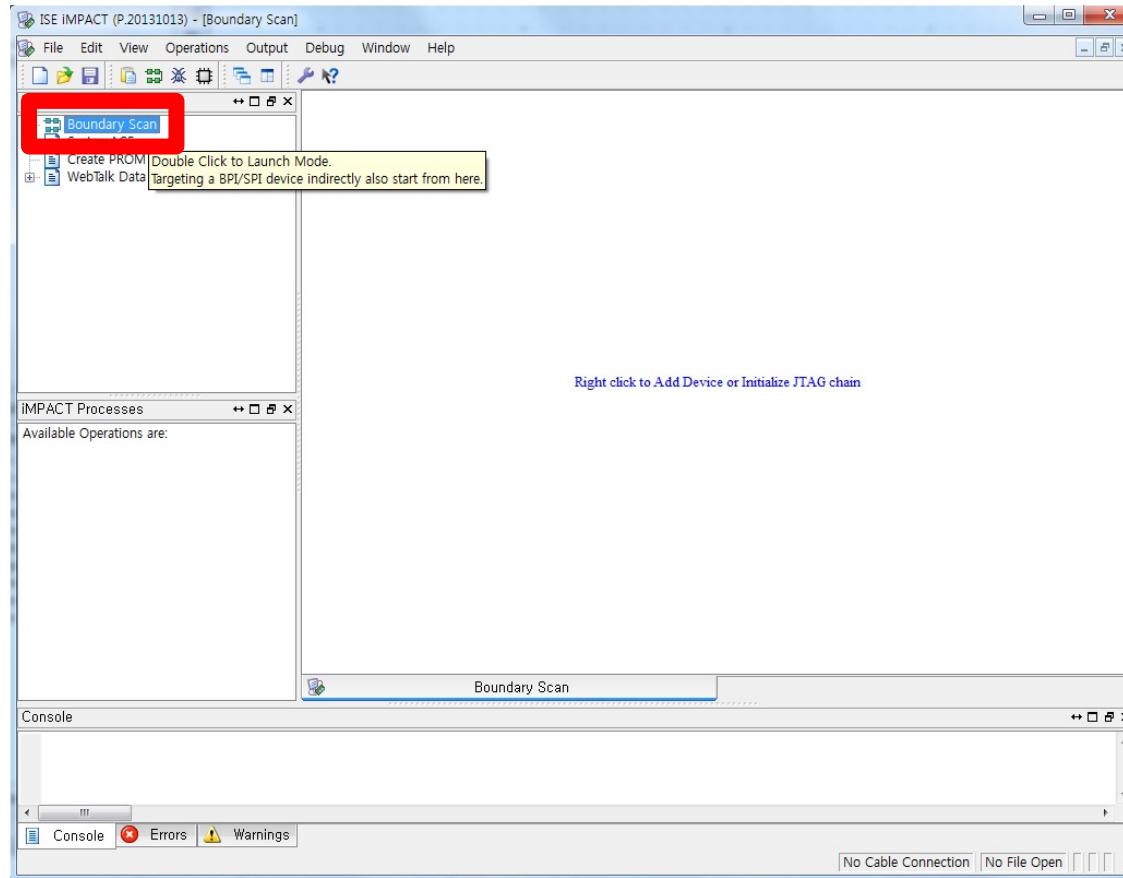
Sample Implementation

- Design tab > double click Configure Target Device
- Click OK to create an iMPACT file.



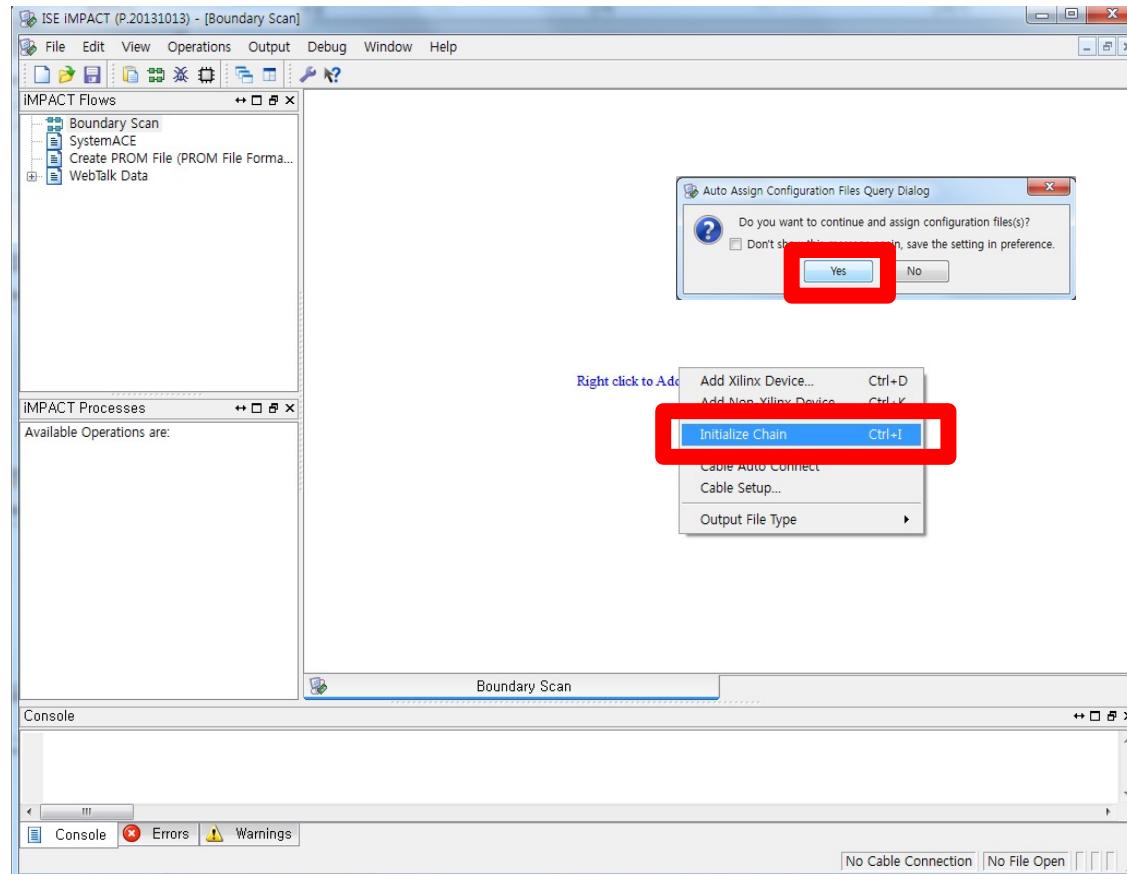
Sample Implementation

- Double click “Boundary Scan”.



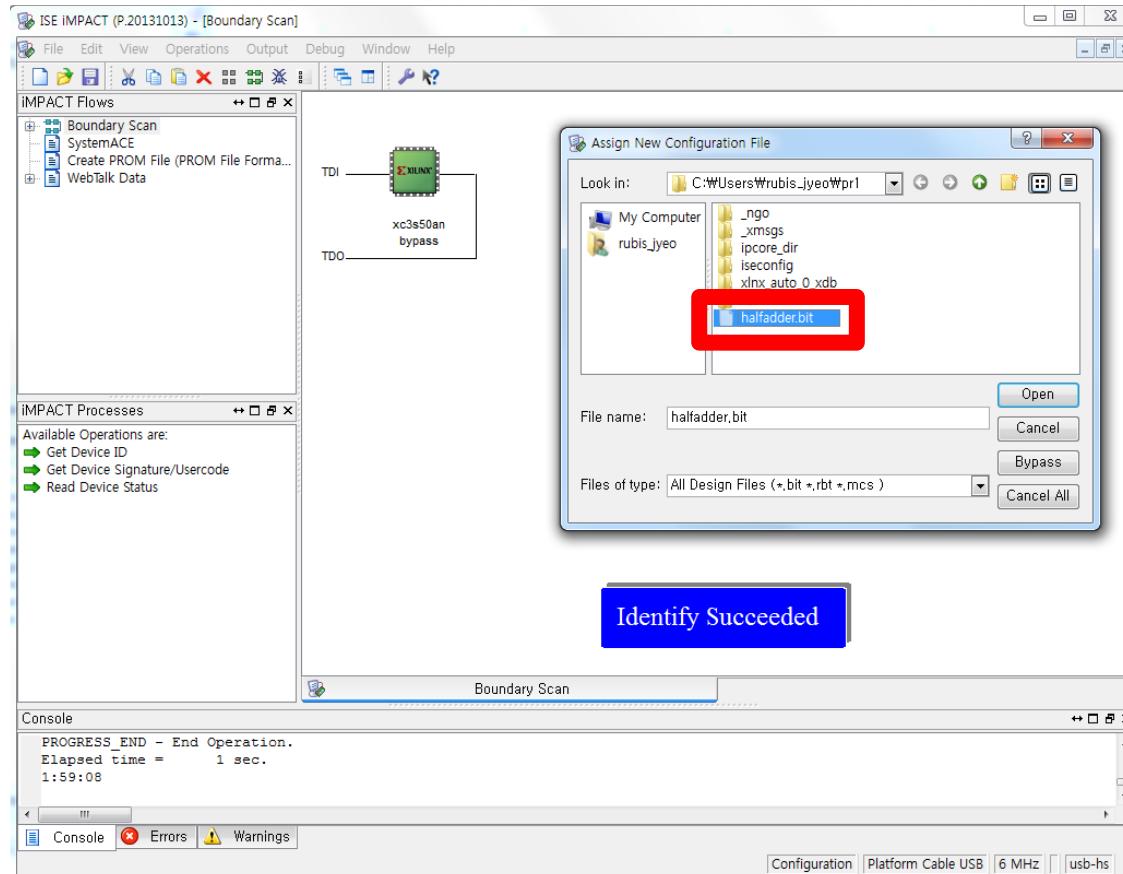
Sample Implementation

- Right click on the right window > click 'Initialize Chain'.



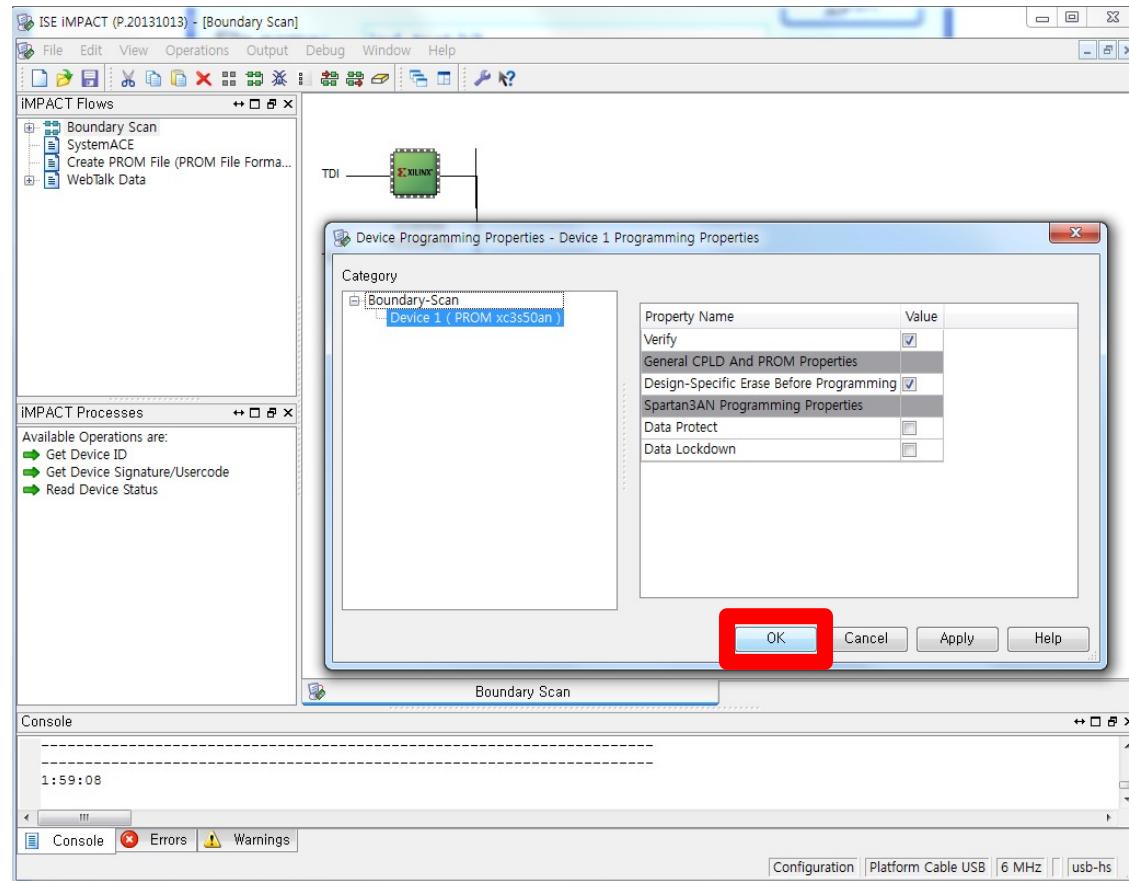
Sample Implementation

- Choose the schematic source you've written.



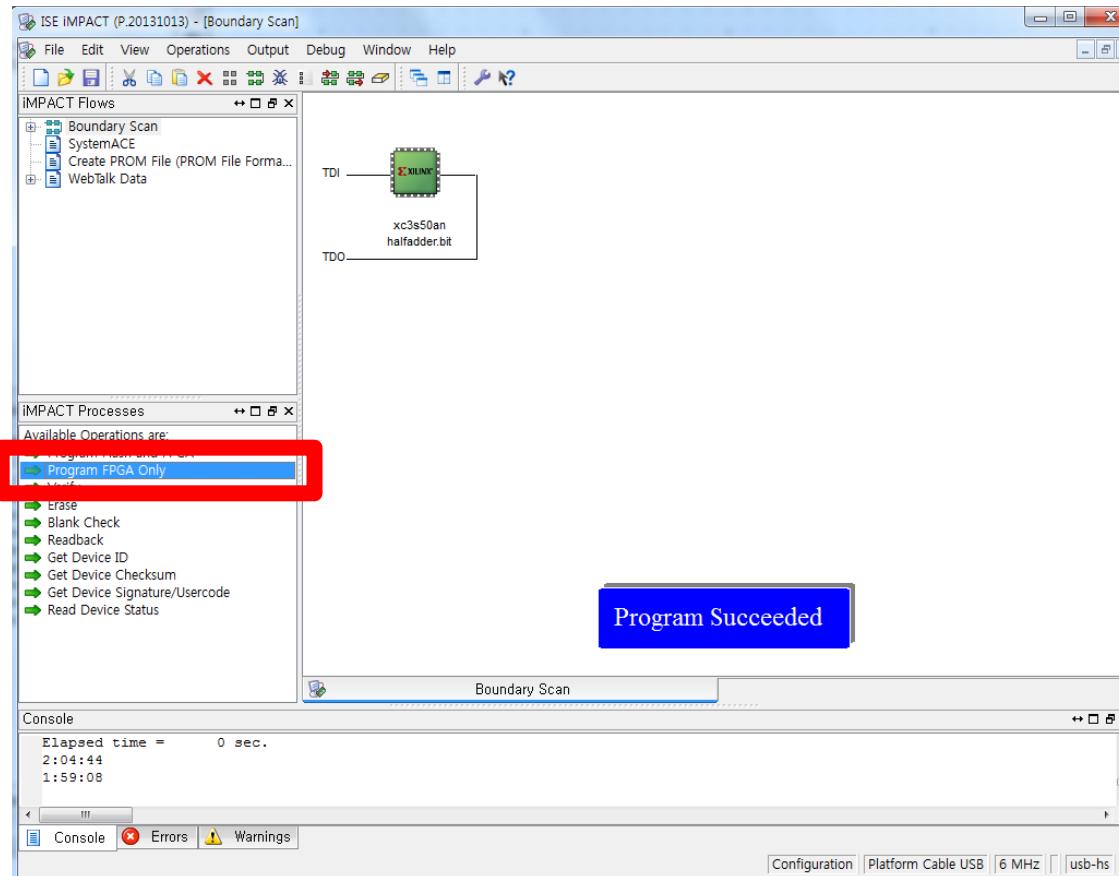
Sample Implementation

- Check settings > 'OK'.



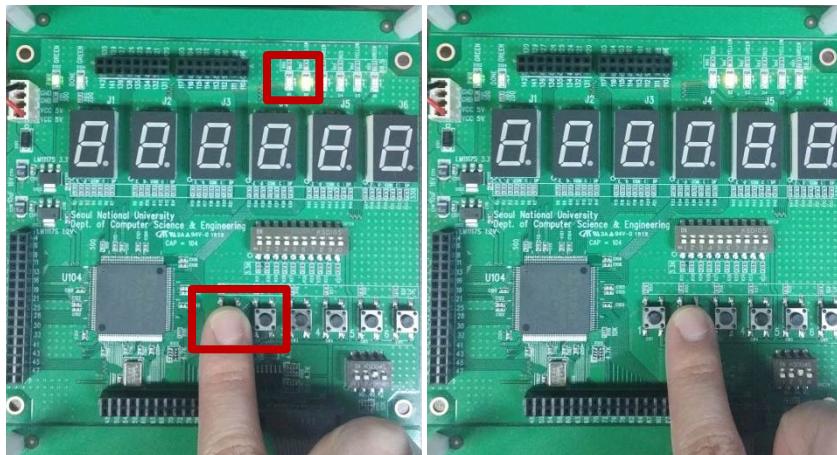
Sample Implementation

- Program FPGA Only > Program Succeeded



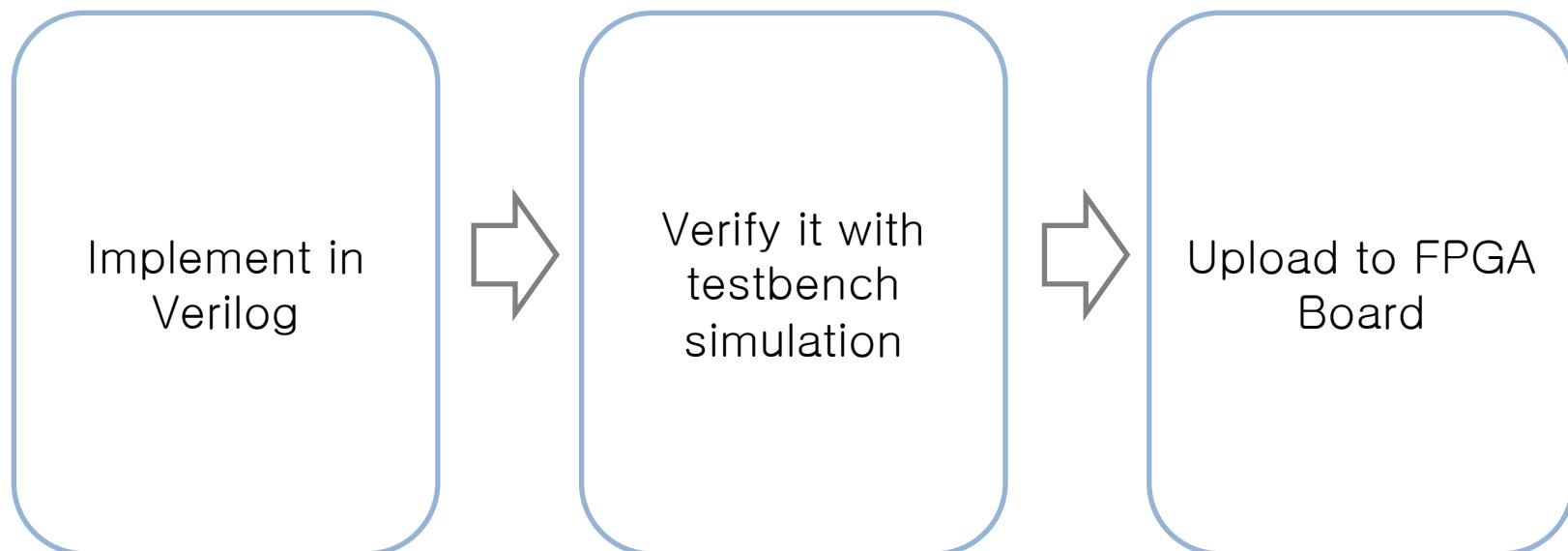
Sample Implementation

- Debug using assigned I/O ports. (This case, LEDs and tactile SWs.)



A	B	Sum	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

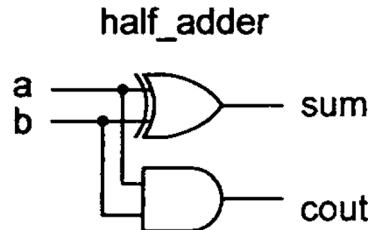
Flow Chart



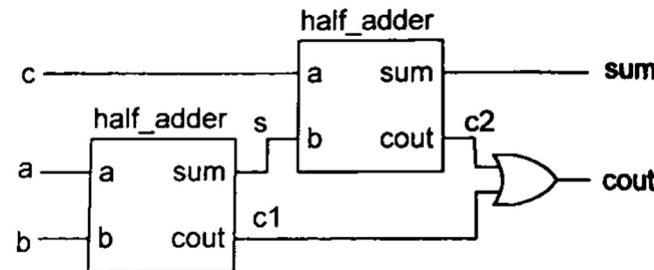
LAB

Today

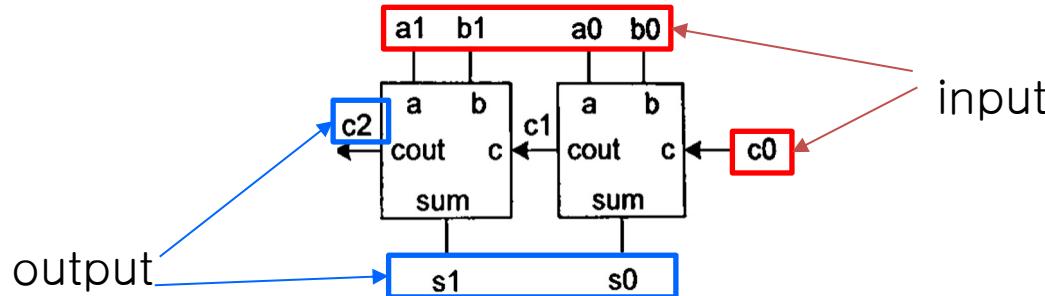
- (1) Implement ripple-carry adder, shown below, in Verilog, (2) and program it on the FPGA board.
- Hint: 5 inputs (tactile SWs), 3 outputs (LEDs)



(a) Half Adder



(b) Full Adder



(c) Ripple Carry Adder

Homework

- **1-bit Arithmetic Logic Unit (ALU).**
 - (1) Follow specification **(a)**. Implement in Verilog.
(M, S1, S0) are control inputs, (Ai, Bi) are data inputs, and (Fi) is outputs.
Use the DIP switch for the inputs.
 - (2) Simulate all possible outputs using Verilog test bench.
 - (3) Program on FPGA board and test all possible outputs.

M	S1	S0	Function	Comment	led
0	0	0	$F_i = A_i$	load	P87
0	0	1	$F_i = \sim A_i$	complement	P88
0	1	0	$F_i = A_i \oplus B_i$	xor	P90
0	1	1	$F_i = \sim(A_i \oplus B_i)$	xnor	P91
1	0	0	$F_i = A_i$	load	P87
1	0	1	$F_i = \sim A_i$	complement	P88
1	1	0	$F_i = A_i + B_i$	Add (ignore carry)	P92
1	1	1	$F_i = \sim A_i + B_i$	Complement and add (ignore carry)	P93

(a) specification

Report

- **Lab part**

1. Xilinx source code
2. Result of simulation
3. Result of test with programmed FPGA

- **Homework part**

1. Xilinx source code
2. Result of simulation
3. Result of test with programmed FPGA

- **Due**

- **5 / 8 (Mon) Class 001**
- **5 / 9 (Tue) Class 002**
- **5 / 11 (Thu) Class 003**