Lab. 05

Logic Design Lab.

Spring 2022

Prof. ChangGun Lee

(cglee@snu.ac.kr)

TA. Hayeon Park

TA. Seohwan Yoo

TA. Haejoo Jeon

(rubis.ld.ta@gmail.com)

Anouncements

- **Windows Laptop**
  - If no teammates have Windows laptop, we plan to borrows laptop for team
  - Related announcements will be uploaded to ETL until this Wednesday
- **Team Change**
  - If a majority wants, the team will change from next week
  - If team is changed, all All borrowed laptops must be returned and a new laptop must be rented if the new team wants it.
  - Please complete the team change survey at the following link
    - ✓ Class #001: https://forms.gle/Ee5AUy2t8JKeWZhSA
    - ✓ Class #002: https://forms.gle/8GokVoQkH9EmNMXa8
    - ✓ Class #003: https://forms.gle/RD4DDyJFkzVczFd17

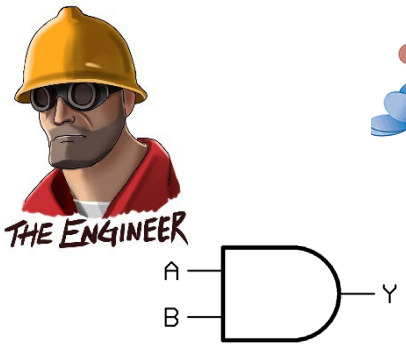
Contents

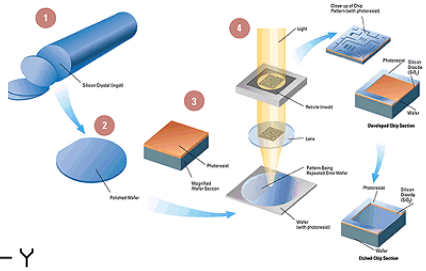
- FPGA **(Field Programmable Gate Array)**
- SNU Logic Design Board
- Sample Implementation
- Lab

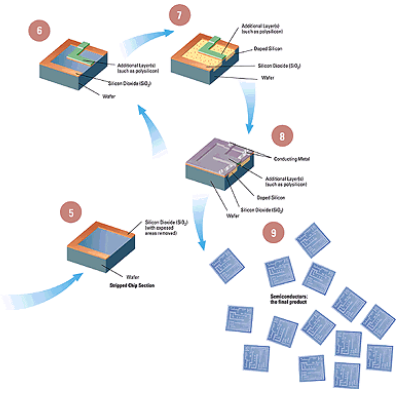
FPGA
(Field Programmable Gate Array)

What is FPGA?

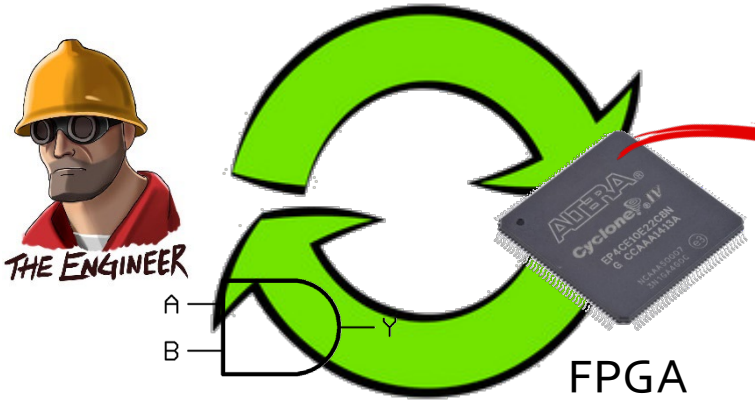
without FPGA

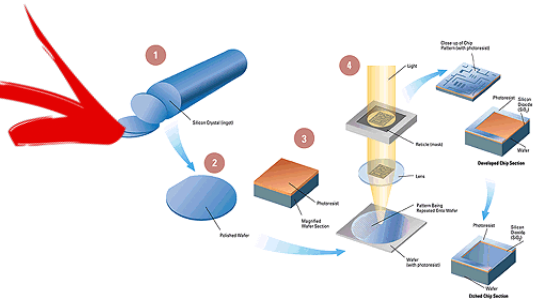


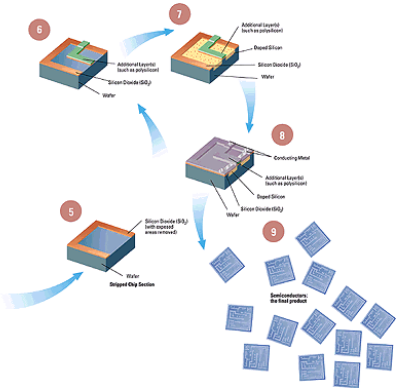




with FPGA

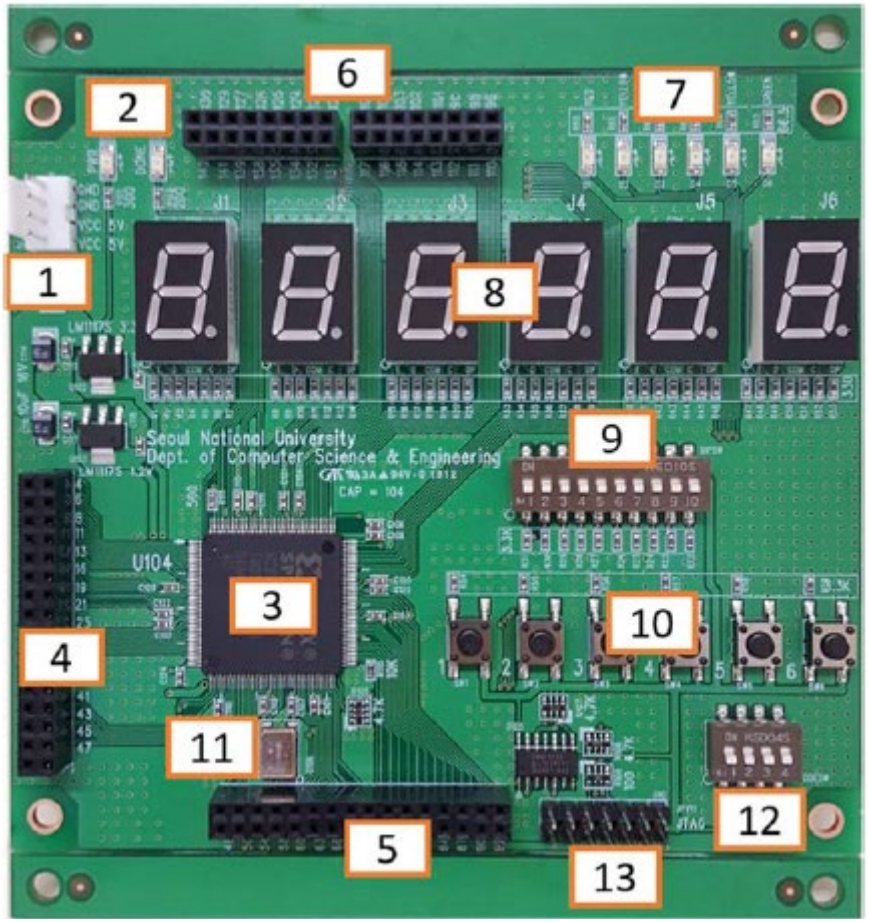






SNU Logic Design Board

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	
P1	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	7-Segment Display [J3]	A
	21		B
	24		C
	25		D
	27		E
	28		F
	29		G
	30	DIP Switch [DipSW1]	1
	31		2
	32		3
	33		4
	41		5
	42		6
	43		7
	44		8
	45		9
	46		10
	47	Tactile Switch [SW1]	

Pin Num		Component	
P2	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	
P3	96
	98
	99
	101
	102
	103
	104
	105
	110
	111
	112
	113
	114
	115
	116
	117

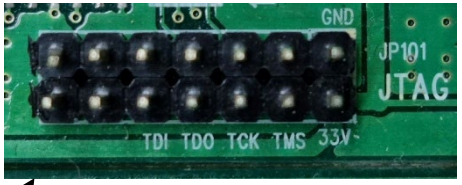
Pin Num	
P4	120
	121
	124
	125
	126
	127
	129
	130
	131
	132
	134
	135
	138
	139
	141
	142

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

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

SNU Logic Design Board

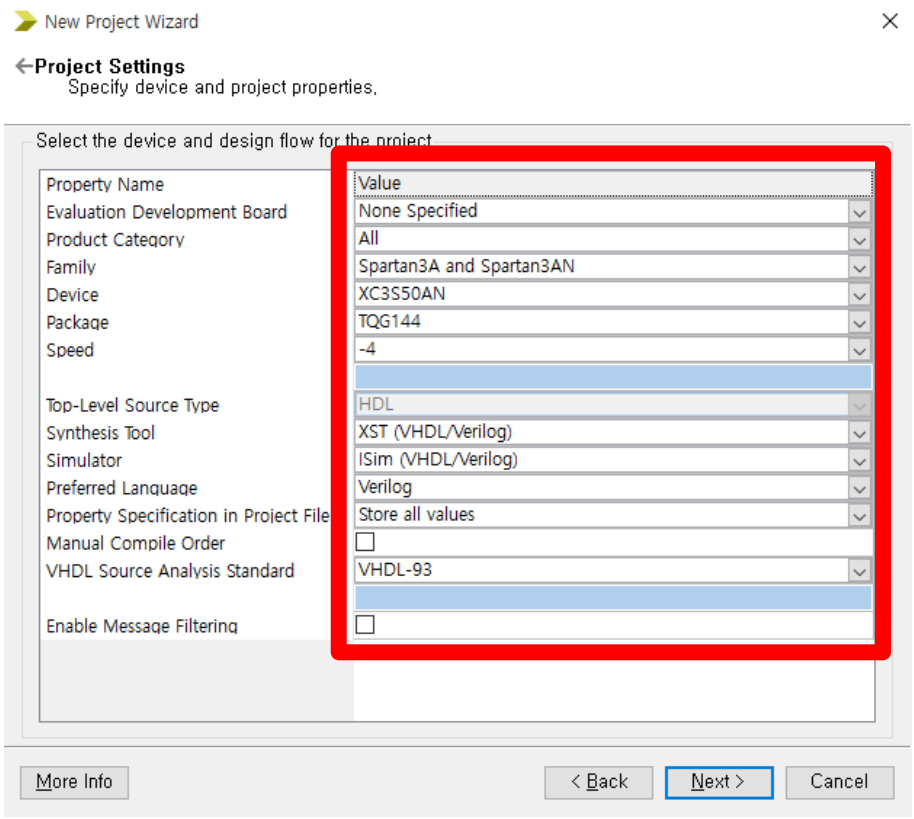
- **Connect to PC with Xilinx Platform Cable USB**
  - HALT/RST Pin(Gray pin) not used.
  - Make sure <u>both</u> board and USB are powered on.

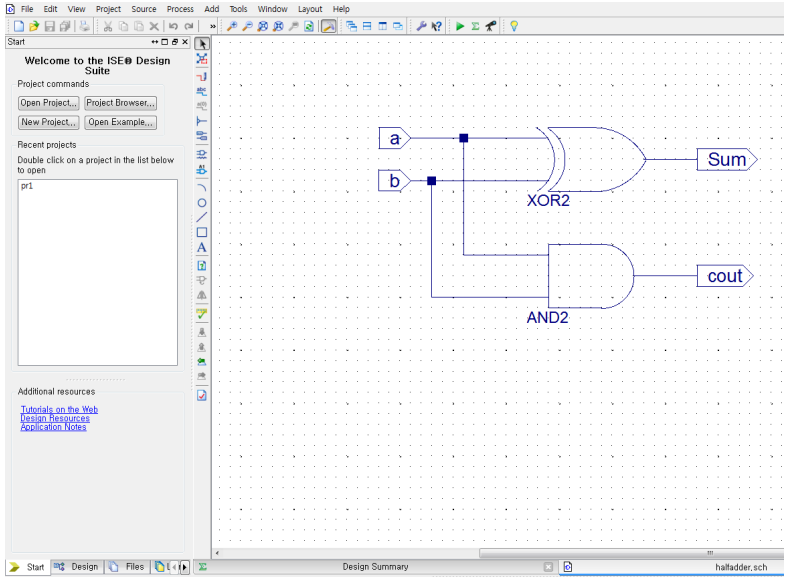


Sample Implementation

Sample Implementation

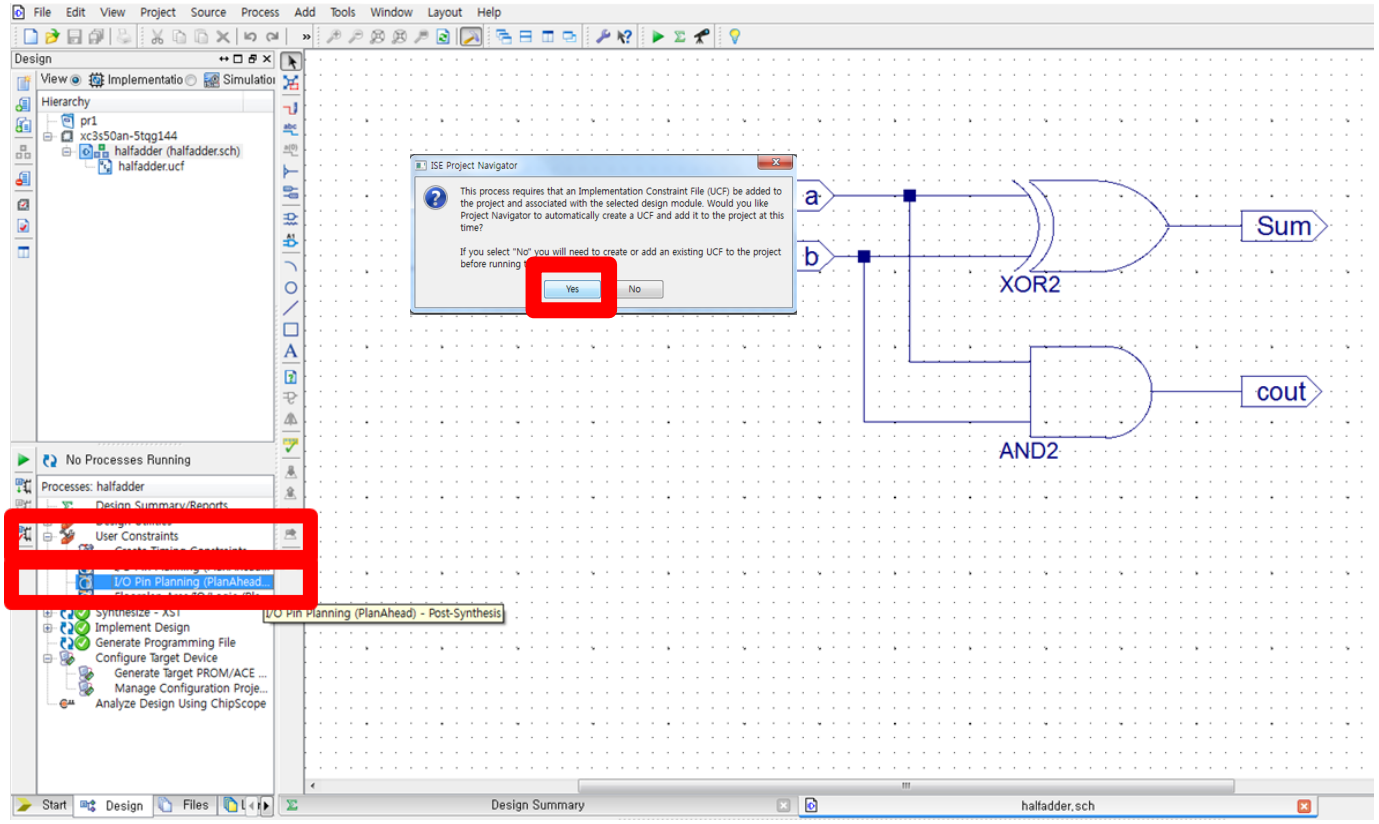
- **Create new project, build a half adder schematic.**
- **For detailed settings, refer to guidelines in Lab03.**





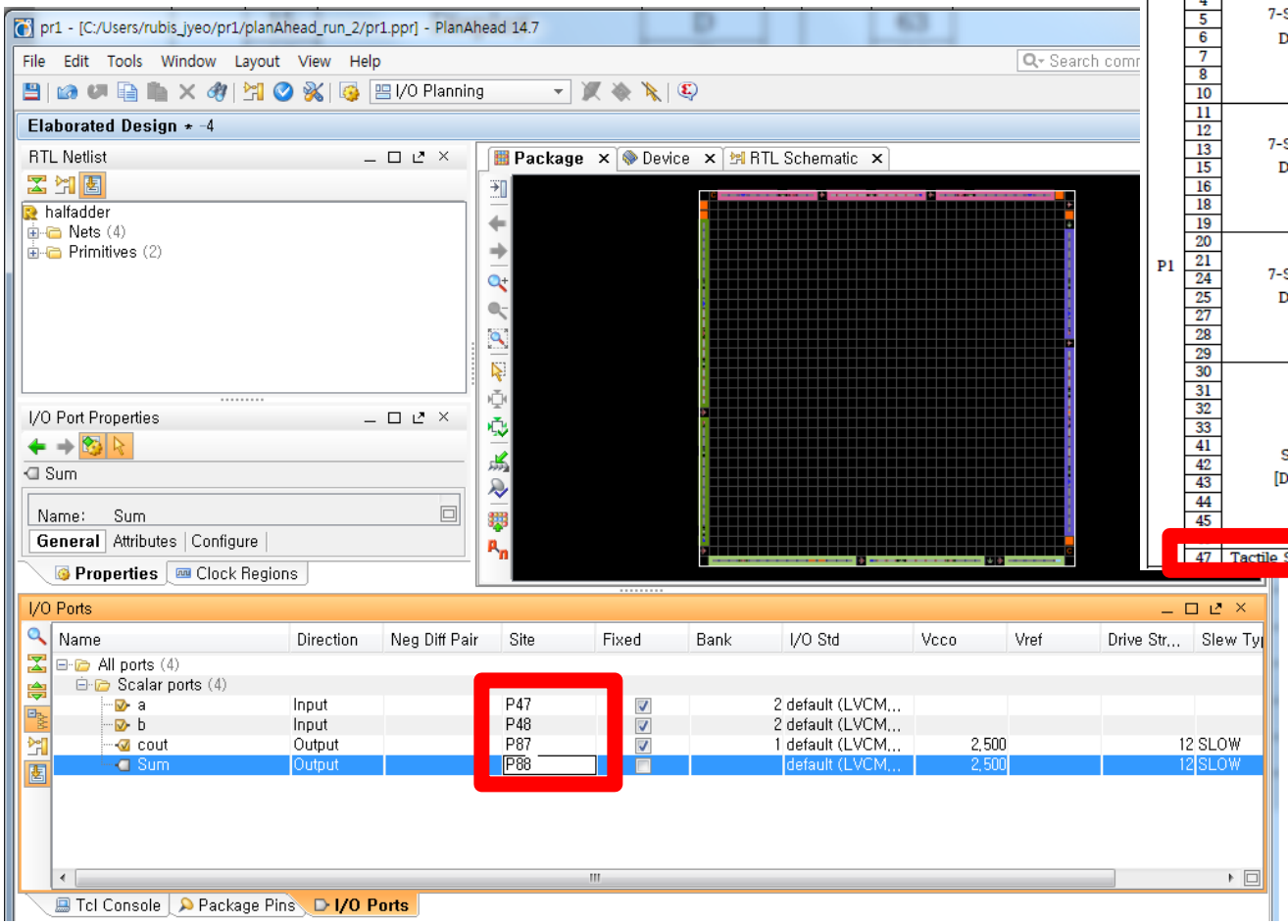
Sample Implementation

- 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

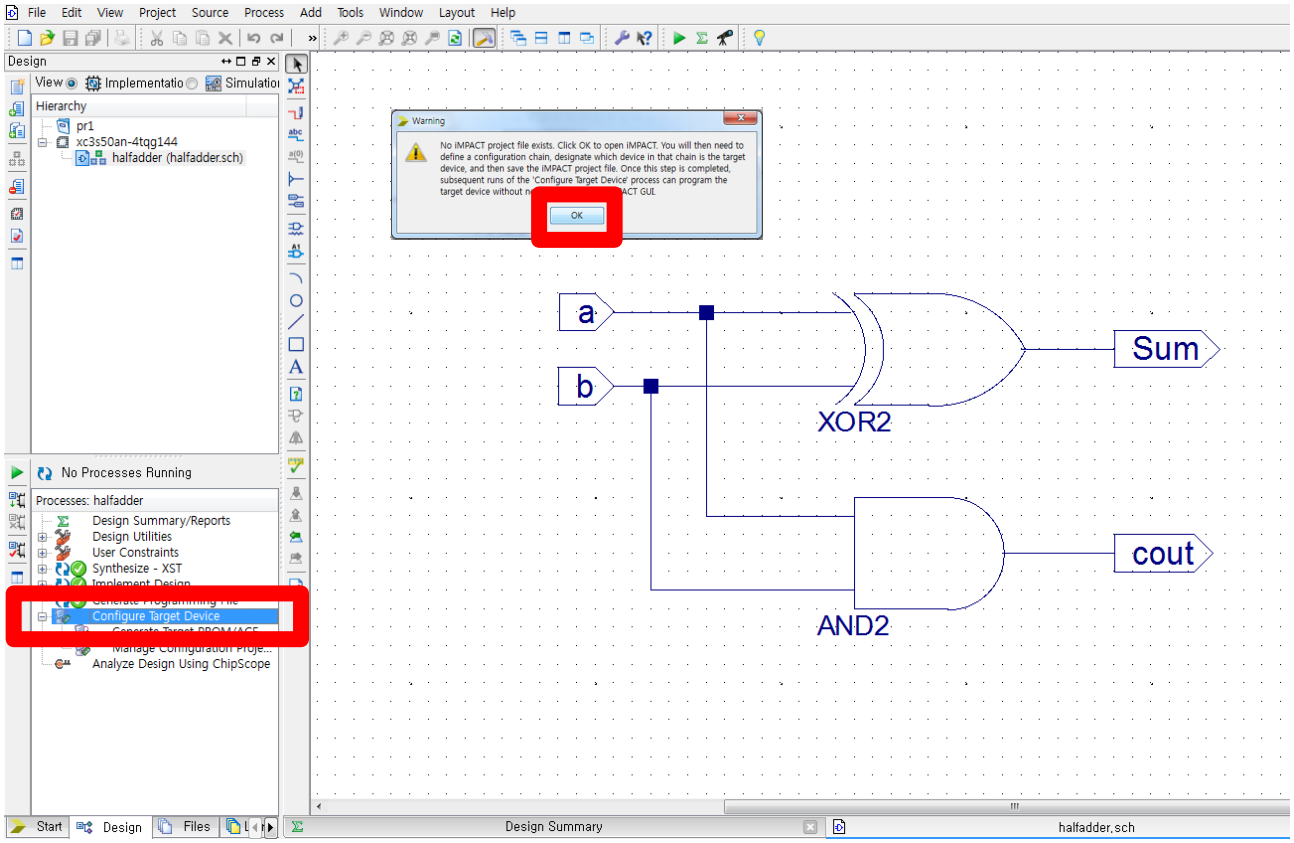


Pin Num		Component	
P1	3	7-Segment Display [J1]	A
	4		B
	5		C
	6		D
	7		E
	8		F
	10		G
	11	7-Segment Display [J2]	A
	12		B
	13		C
	15		D
	16		E
	18		F
	19		G
	20	7-Segment Display [J3]	A
	21		B
	24		C
	25		D
	27		E
	28		F
	29		G
	30	DIP Switch [DipSW1]	1
	31		2
	32		3
	33		4
	41		5
	42		6
	43		7
	44		8
	45		9
	47	Tactile Switch [SW1]	

Pin Num		Component	
P2	48	Tactile Switch [SW2]	
	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
	87	LED [D1]	Red
	88	LED [D2]	Yellow
	91	LED [D4]	Red
	92	LED [D5]	Yellow
	93	LED [D6]	Green

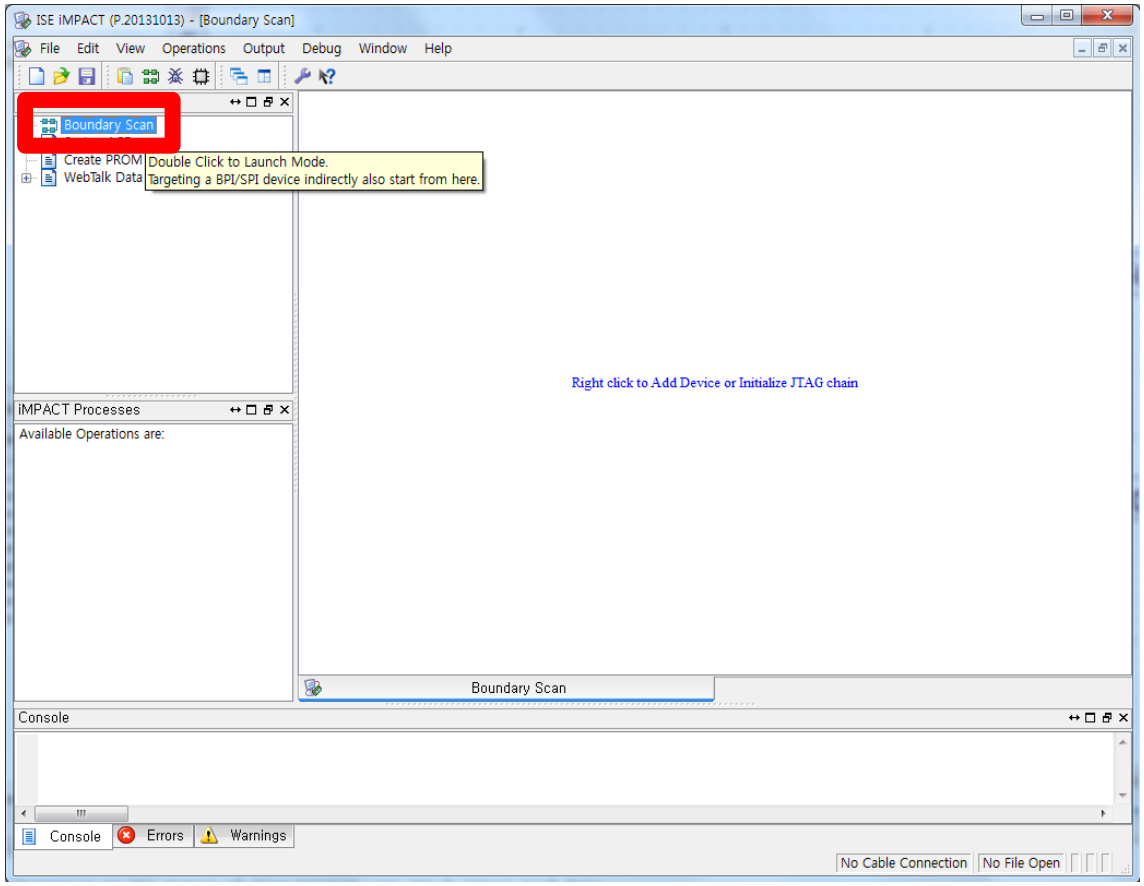
Sample Implementation

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



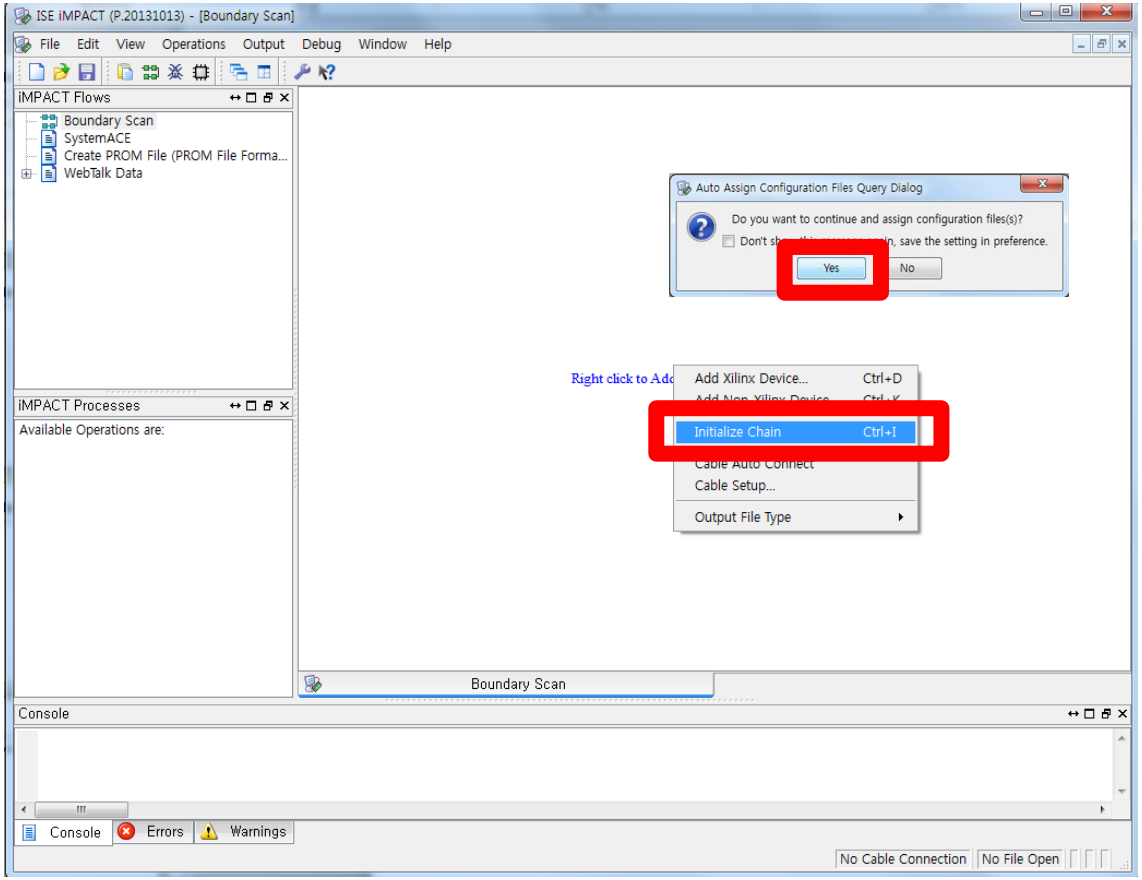
Sample Implementation

- Double click “Boundary Scan”.



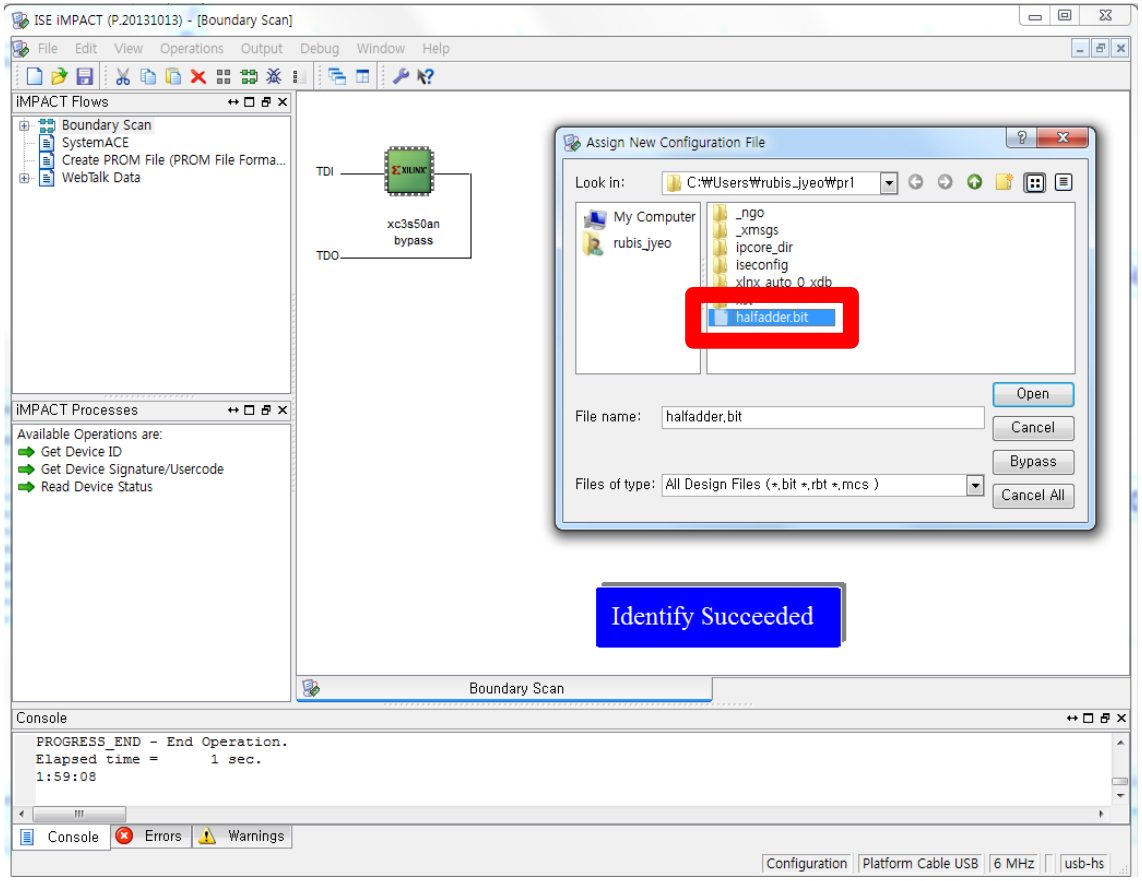
Sample Implementation

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



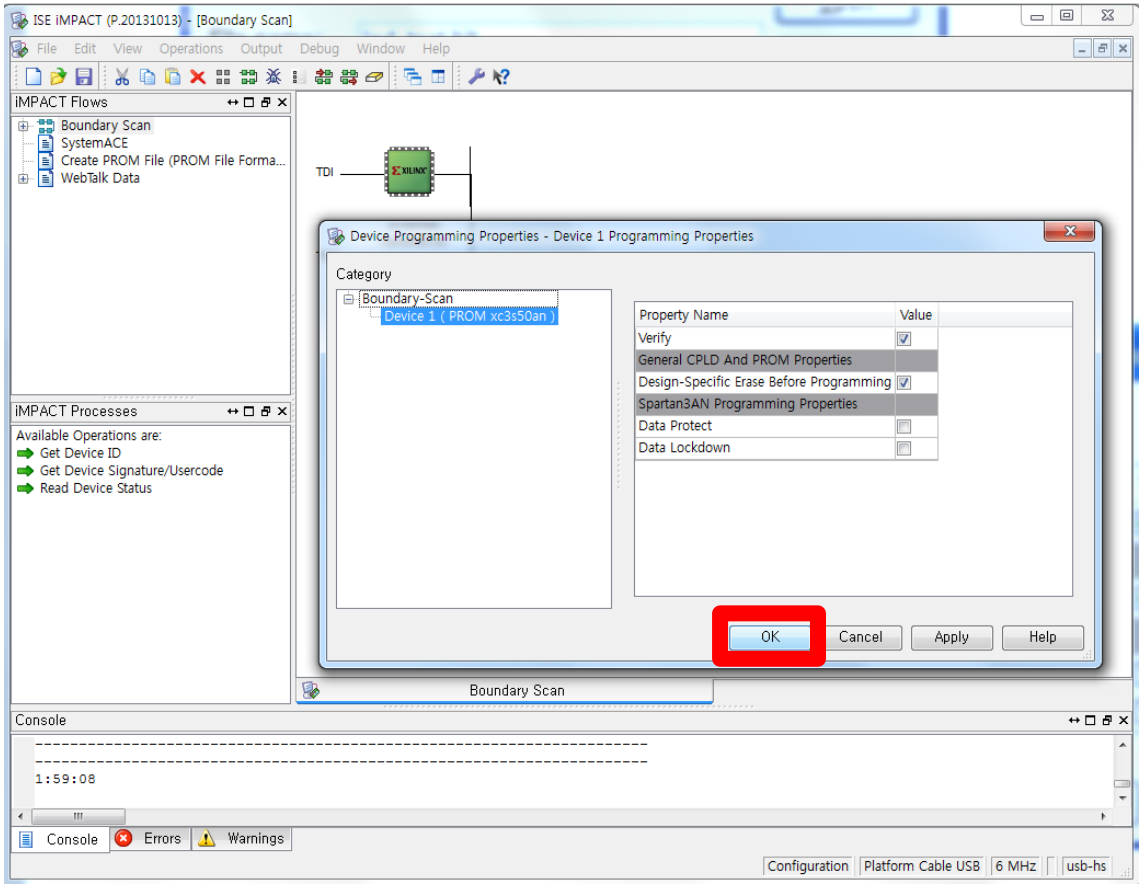
Sample Implementation

- **Choose the schematic source you've written.**



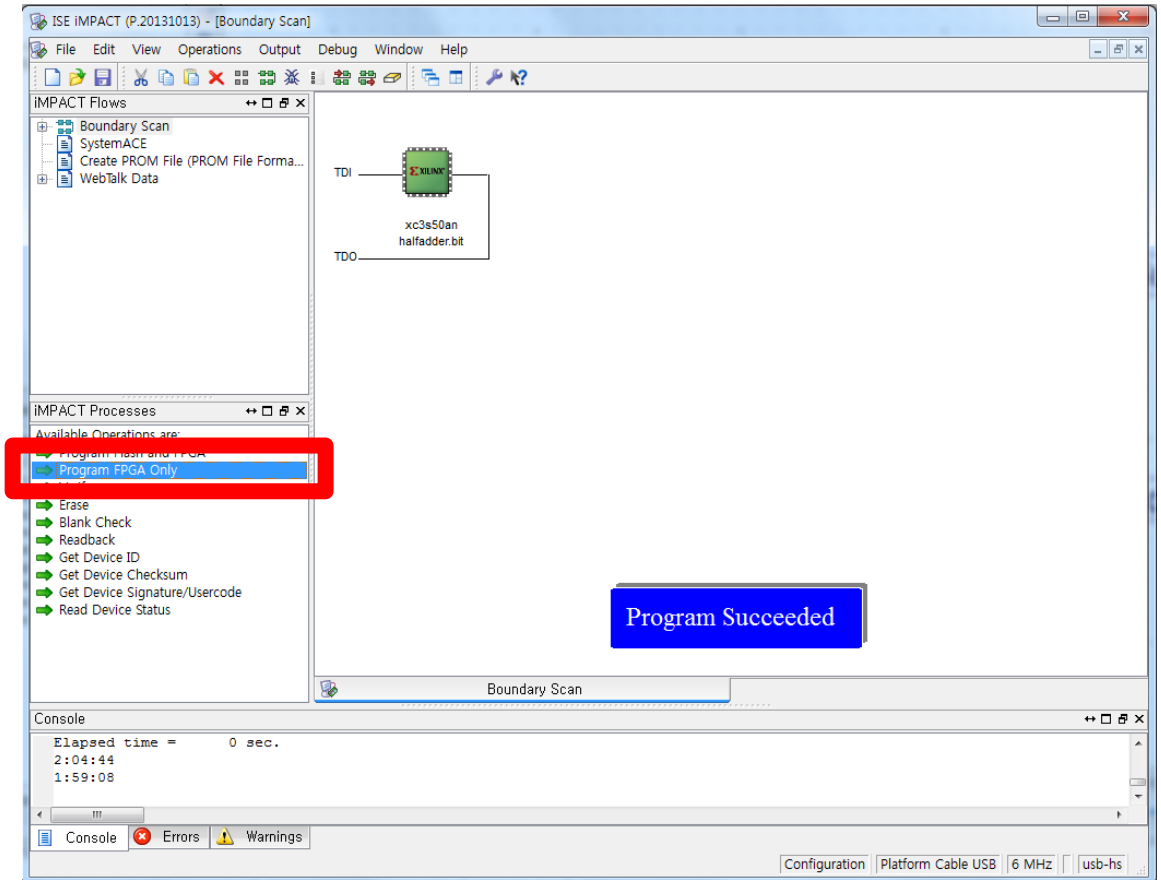
Sample Implementation

- **Check settings > 'OK'.**



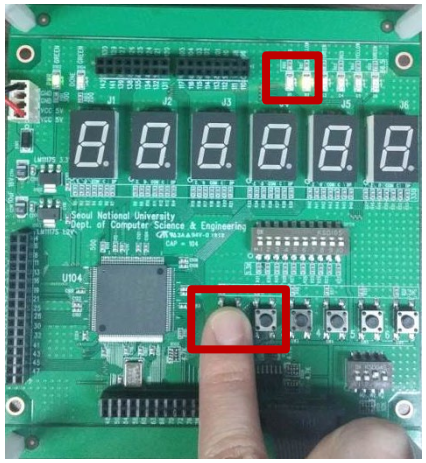
Sample Implementation

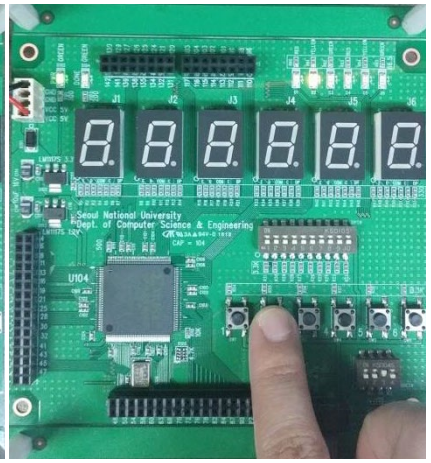
- **Program FPGA Only 〉Program Succeeded**

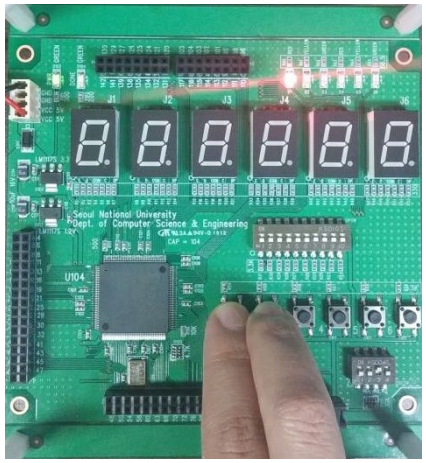


Sample Implementation

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

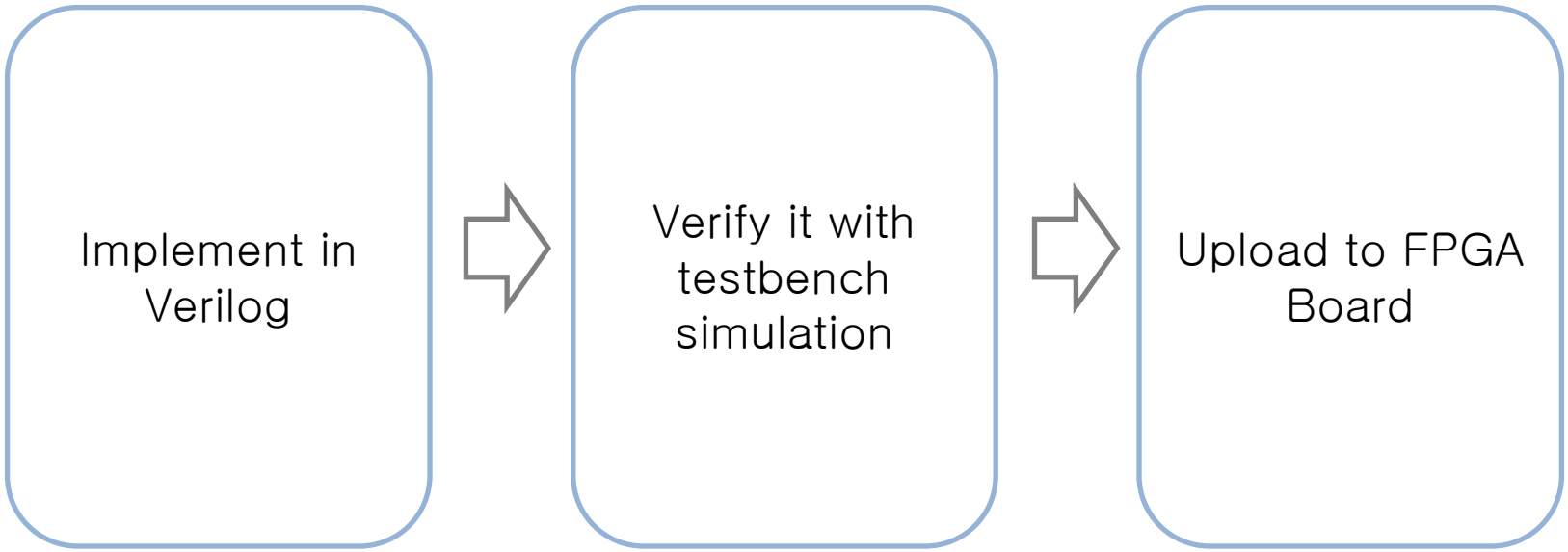






A	B	Sum	Carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

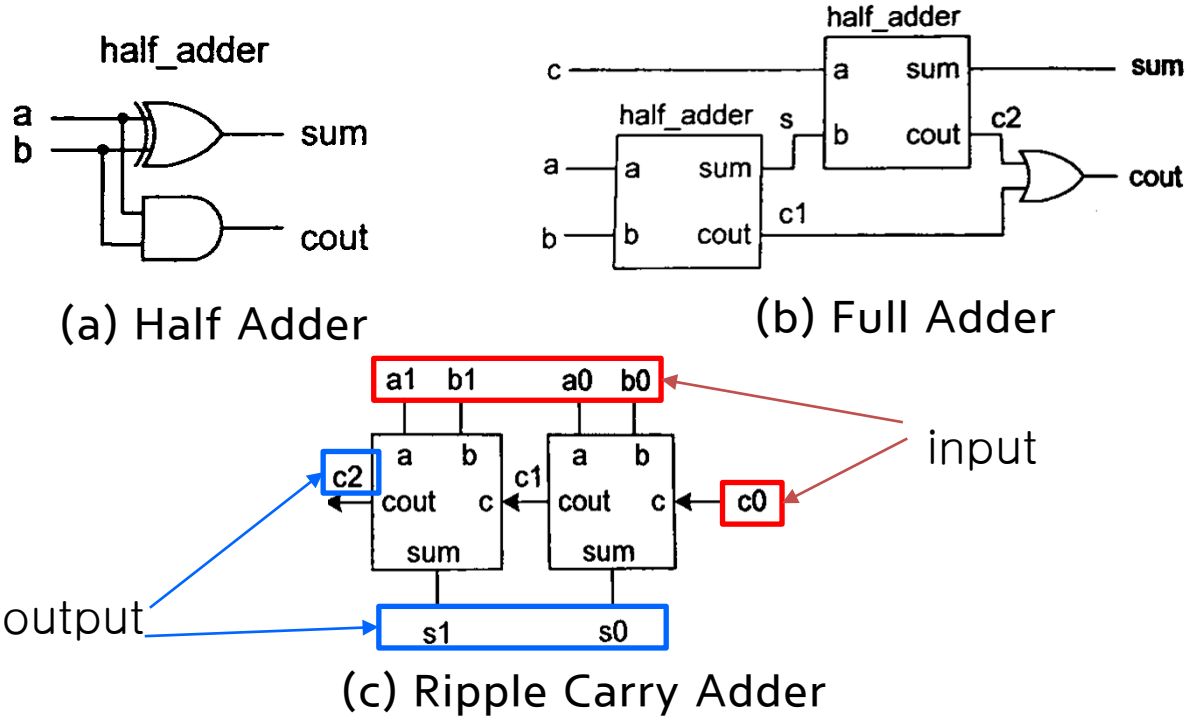
Flow Chart


Implement in Verilog
Verify it with testbench simulation
Upload to FPGA Board

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)



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 / 2 (Mon) Class 001**
  - **5 / 3 (Tue) Class 002**
  - **5 / 4 (Wed) Class 003**