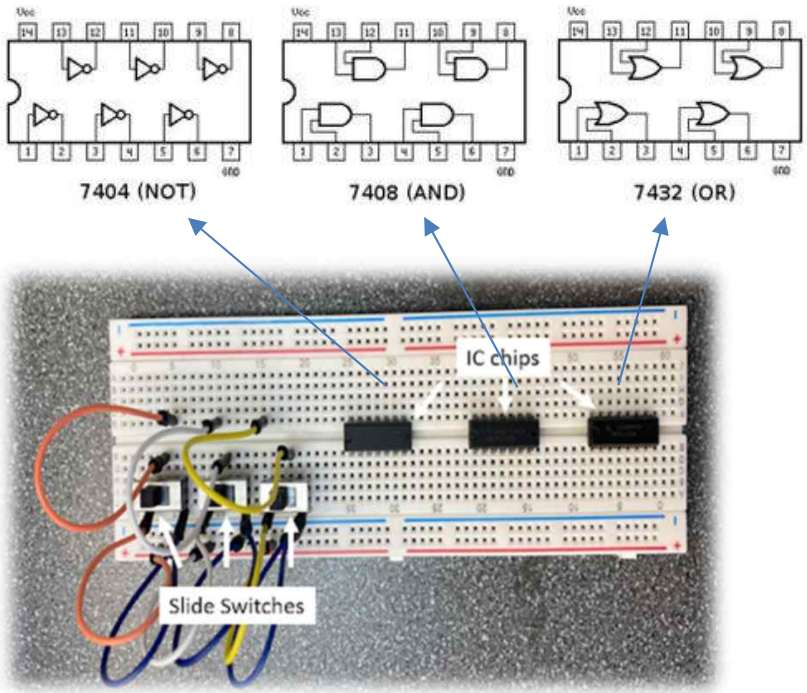
4bit binary Adder 구현 (FPGA 프로그래밍)

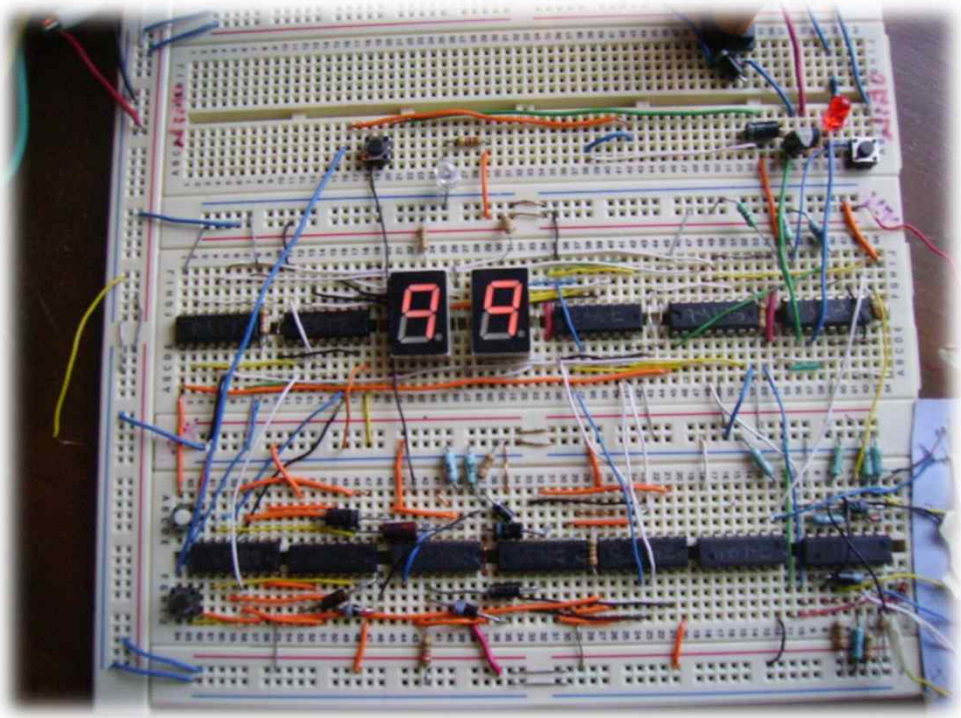
Shinwoong Kim

Integrated Circuit (IC)

- **Small/medium-scale integration (SSI, MSI)**
  - ✓ It contains only a few transistors → a few logic gates
  - ✓ In case of small size system, it can be designed by the discrete logic gate IC
    - State diagram → state table → K-map → state equation → implement system



*https://manual.eg.poly.edu/index.php/Introduction_to_Digital_Logic

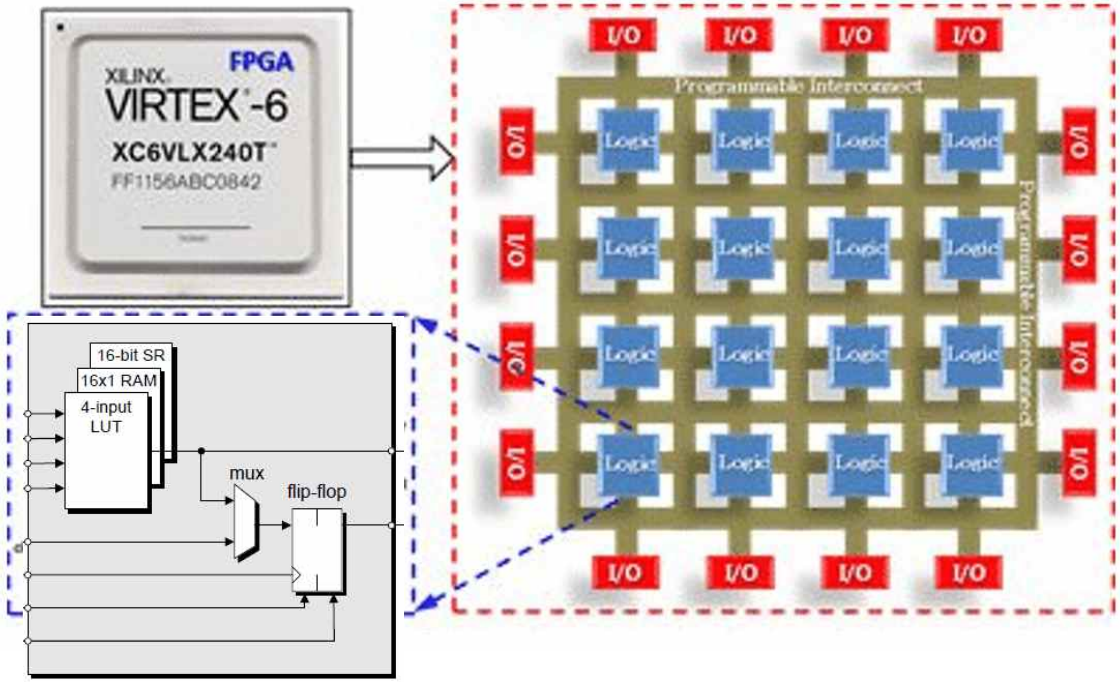


*https://www.instructables.com/Digital-Combination-Lock/

FPGA

- **Field Programmable Gate Array**
  - ✓ Consists of simple programmable logic blocks and massive fabric of interconnection

*https://www.researchgate.net/figure/FPGA-architecture-description_fig3_308883347

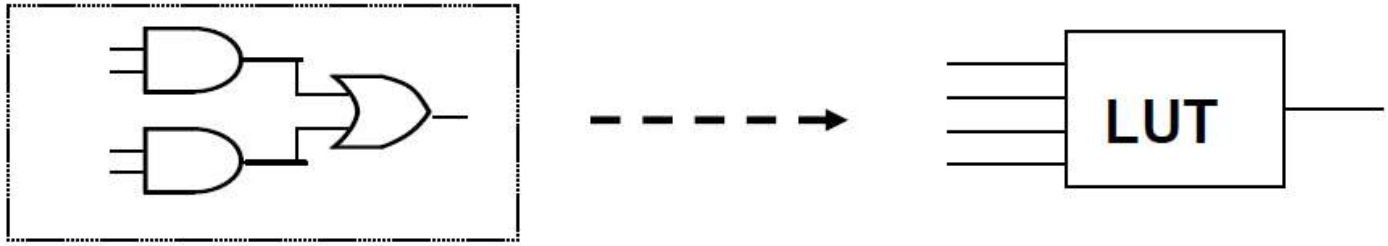


-Large number of Logic bocks (LUT, MUX, FF)
-User programmable interconnection

Circuit Compilation

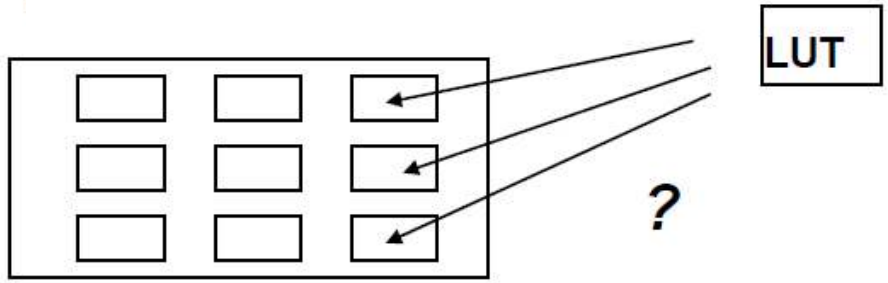
- **1. Technology mapping**

  1) Designed by Verilog-HDL
  2) Logic synthesis (Verilog → logic gates) by SW



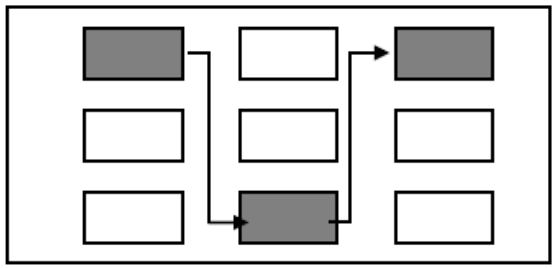
- **2. Placement**

  Assign a logical LUT
  to a physical location



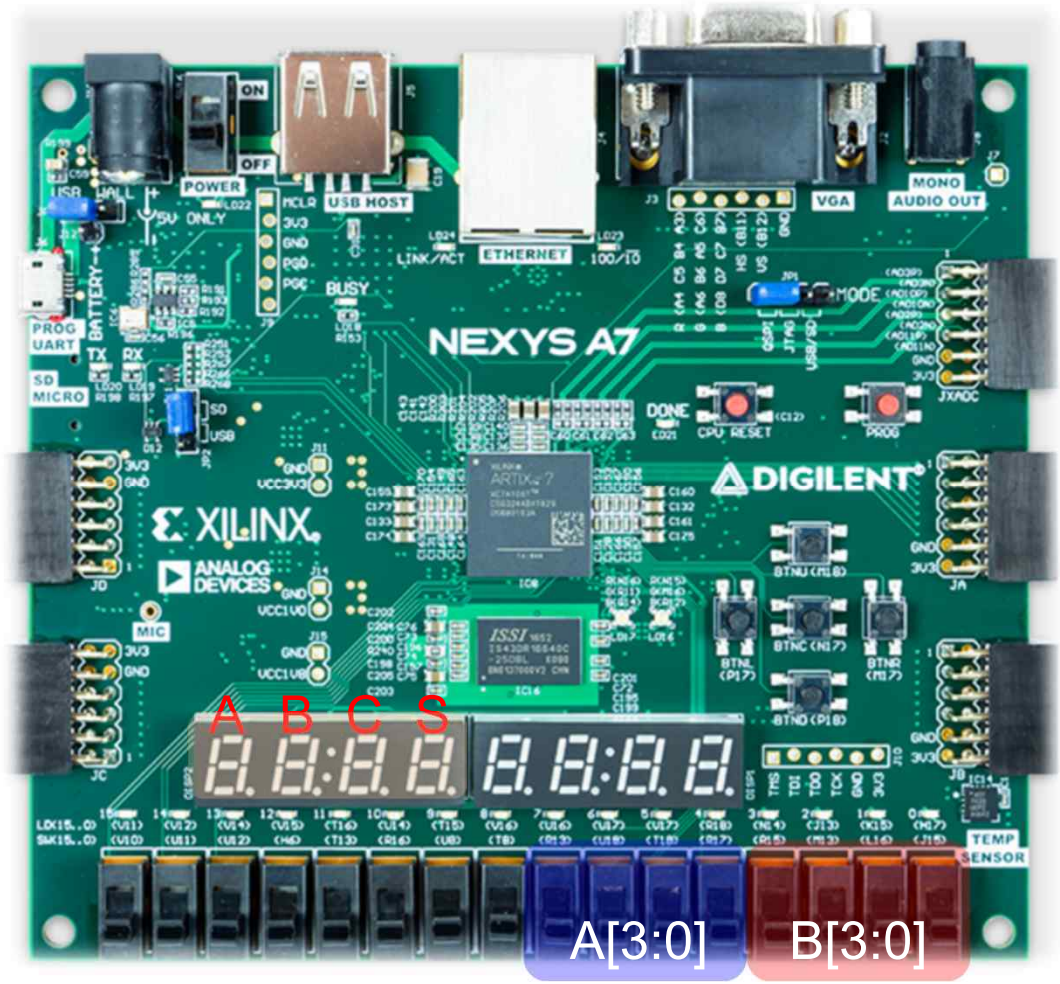
- **3. Routing**

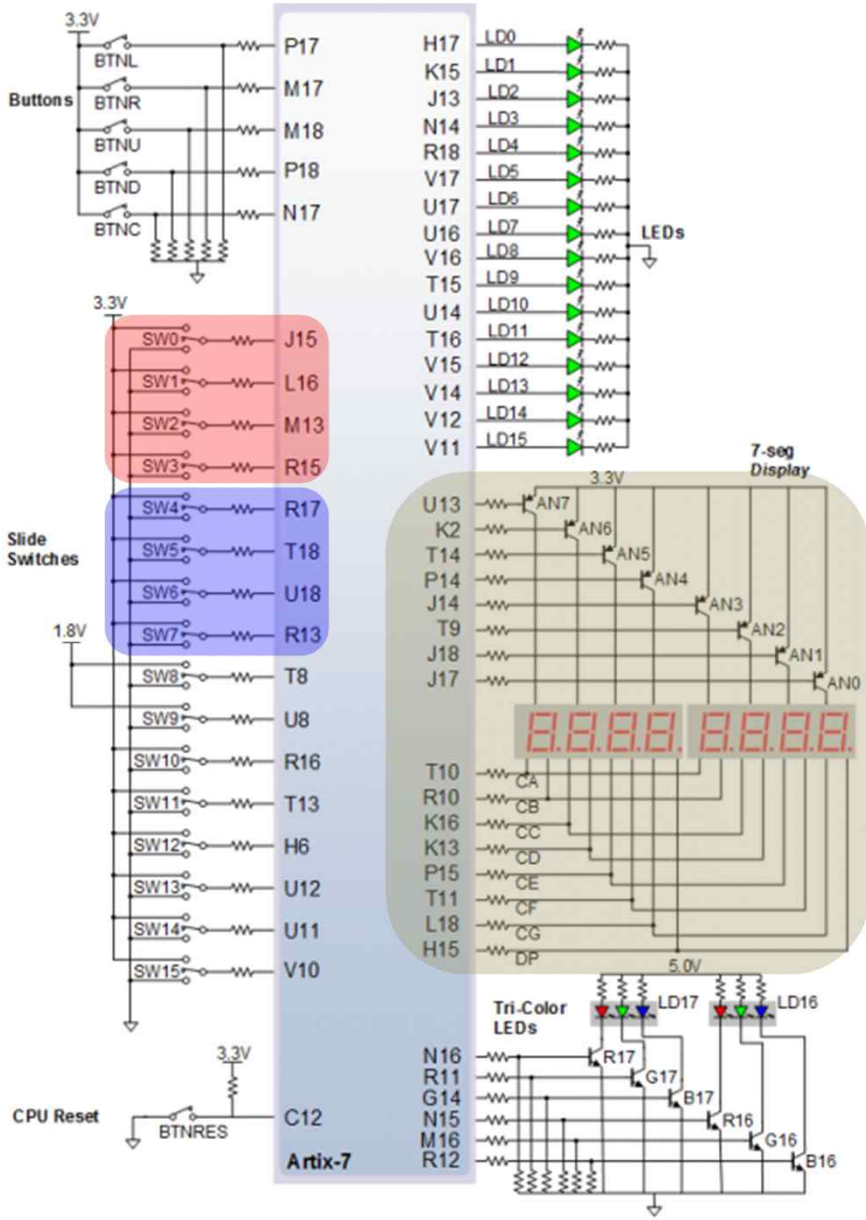
  Select wire segments and
  Switches for interconnection



FPGA Board 소개

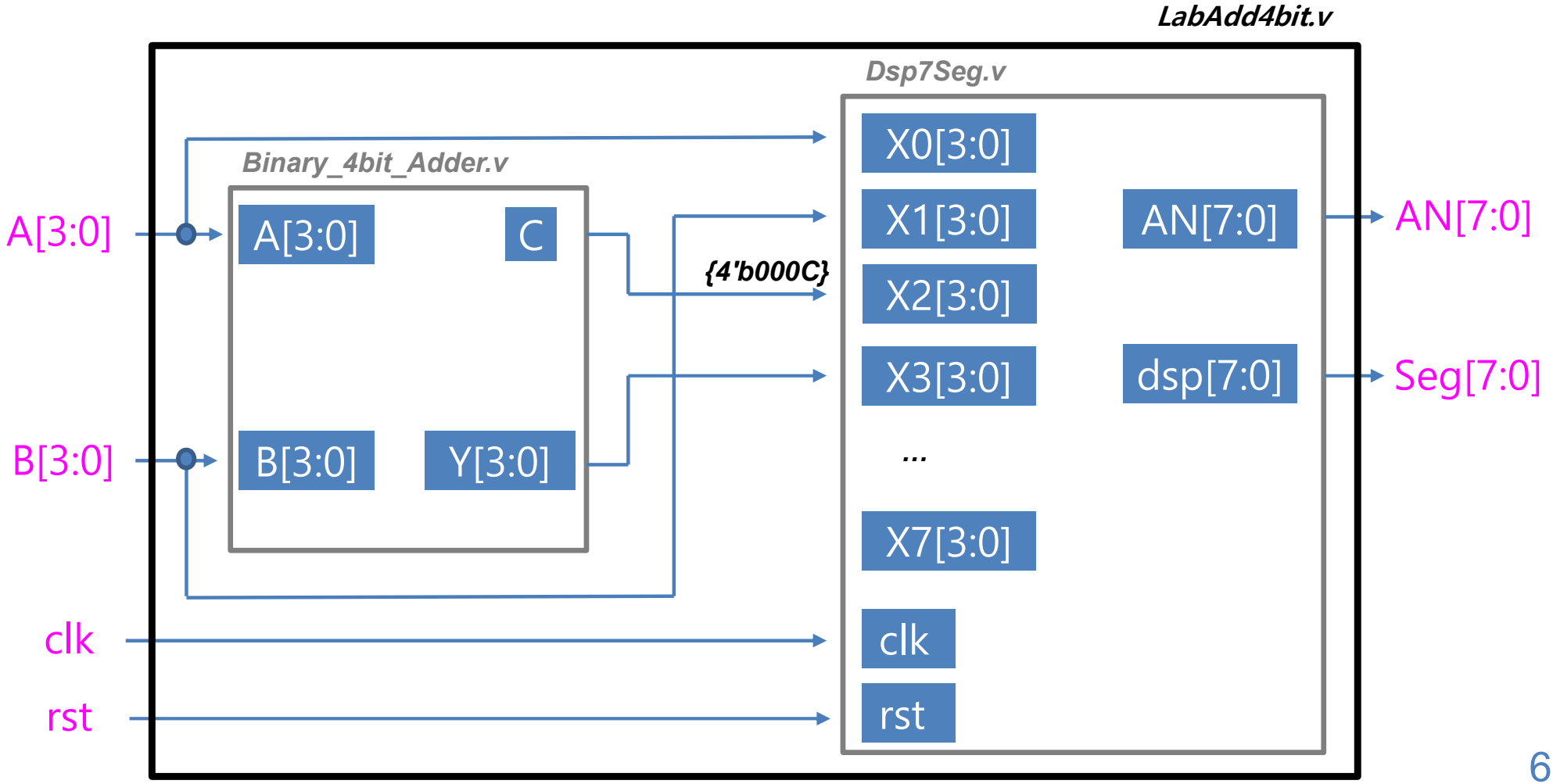
- **Digilent Nexys A7 board**
  - ✓ FPGA chip: Xilinx Artix-7





실험

- **1) Top 모듈 설계 (LabAdd4bit.v)**
  - ✓ 4bit adder: Binary_4bit_Adder.v
  - ✓ 7-segment display decoder: Dsp7Seg.v



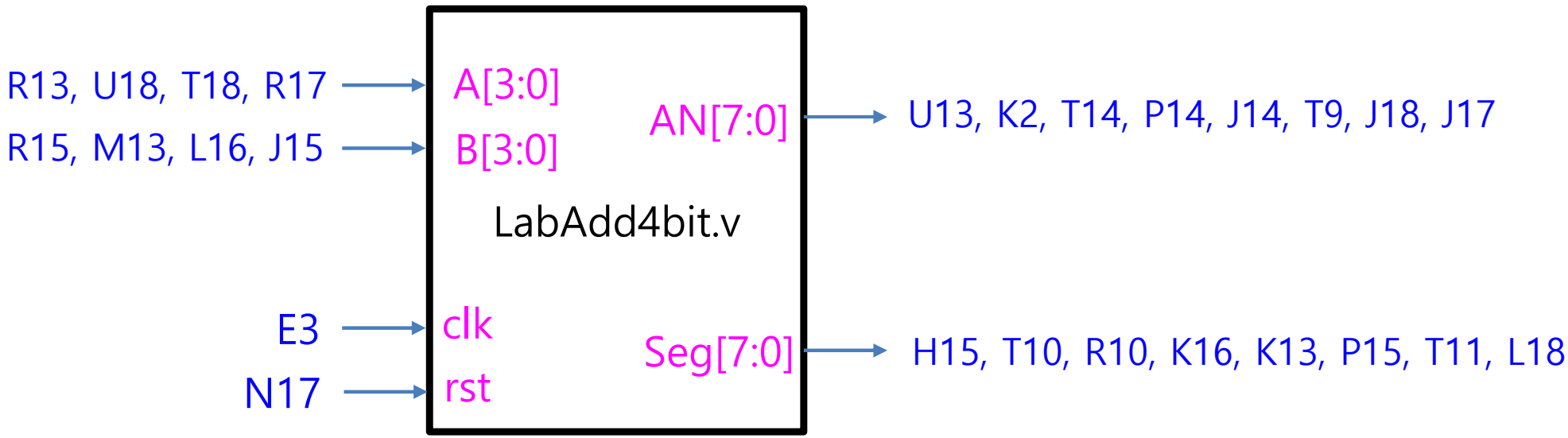
실험

- **1) Top 모듈 설계 (LabAdd4bit.v)**
  - ✓ 4bit adder: Binary_4bit_Adder.v
  - ✓ 7-segment display decoder: Dsp7Seg.v

```
`timescale 1ns / 1ps

module LabAdd4bit( A, B, clk, rst, AN, Seg );

input [3:0] A;
input [3:0] B;
input clk, rst;
output [7:0] AN;
output [7:0] Seg;

wire [3:0] carry, sum;
assign carry[3:1]=3'b000;

Binary_4bit_Adder U1 (.A(A), .B(B), .C(carry[0]), .Y(sum));
Dsp7Seg U2 (.X0(A), .X1(B), .X2(carry), .X3(sum), .clk(clk), .rst(rst), .AN(AN), .dsp(Seg));


endmodule
```

실험

- **2) Pin constraint 작성**
  - ✓ Constraint 선택 → Add sources
  - ✓ Design in/out port를 보드 pin 정보에 맞추어 mapping 필요
    - xdc file 작성
    - https://github.com/Digilent/digilent-xdc/



실험

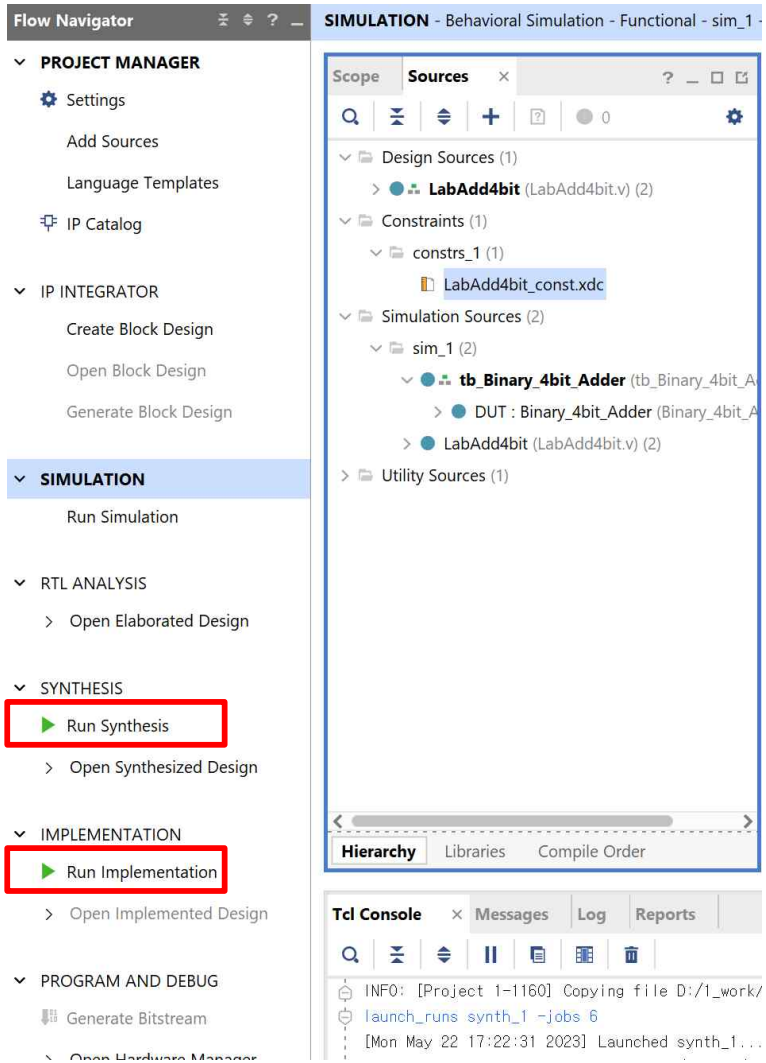
- **2) Pin constraint 작성**
  - ✓ Design in/out port를 보드 pin 정보에 맞추어 mapping 필요

D:/1_work/Xilinx_work/basic_circuit_experiment/basic_circuit_experiment.srcs/constrs_1/new/LabAdd4bit_const.xdc

```
1
2  set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
3  set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports { rst }]; #IO_L9P_T1_DQS_14 Sch=btnc
4
5  # input A
6  set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { A[0] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
7  set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { A[1] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
8  set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { A[2] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
9  set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { A[3] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
10
11 # input B
12 set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { B[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
13 set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { B[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
14 set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { B[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
15 set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { B[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
16
17 # output AN
18 set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }]; #IO_L23P_T3_F0E_B_15 Sch=an[0]
19 set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
20 set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
21 set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
22 set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
23 set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
24 set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
25 set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]
26
27 # output Seg
28 set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { Seg[6] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
29 set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { Seg[5] }]; #IO_25_14 Sch=cb
30 set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { Seg[4] }]; #IO_25_15 Sch=cc
31 set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { Seg[3] }]; #IO_L17P_T2_A26_15 Sch=cd
32 set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { Seg[2] }]; #IO_L13P_T2_MRCC_14 Sch=ce
33 set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { Seg[1] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
34 set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { Seg[0] }]; #IO_L4P_T0_D04_14 Sch=cg
35 set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { Seg[7] }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
```

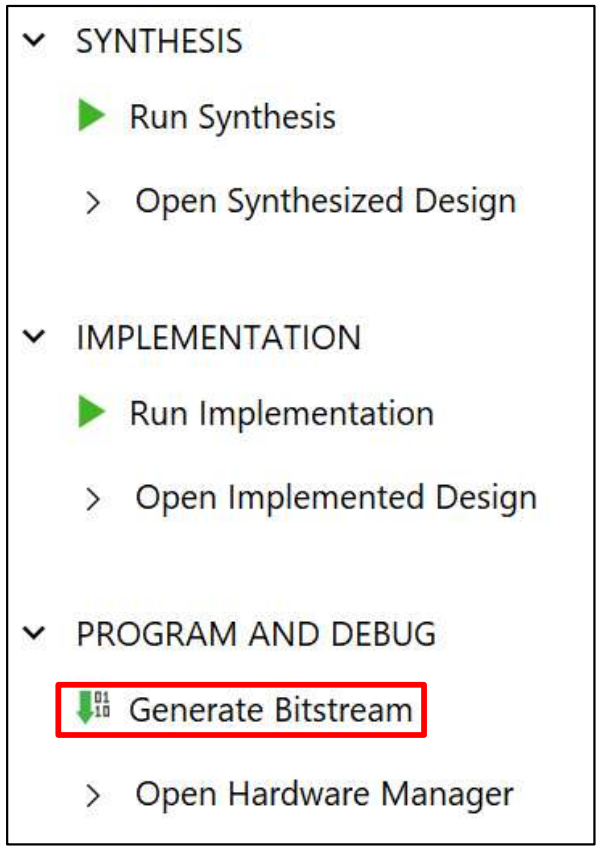
실험

- **3) Implementation**
  - ✓ Synthesis, please & routing



실험

- **4) Generate Bitstream**
  - ✓ .bit file generation



실험

- **5) Program Device**
  - ✓ Open Hardware Manager → Open Target
  - ✓ Program Device