EasySoC-Z7010 FPGA/SoC 설계 플랫폼

7. FPGA를 이용한 Stepper Motor 제어 실습

개요

- Stepper Moter의 컨트롤러를 HDL로 설계한다.
  - Stepper Moter의 원리를 이해하고 제어할 수 있다.
  - EasySoC-Z7010의 FPGA를 이용하여 외부 디바이스를 제어할 수 있다.

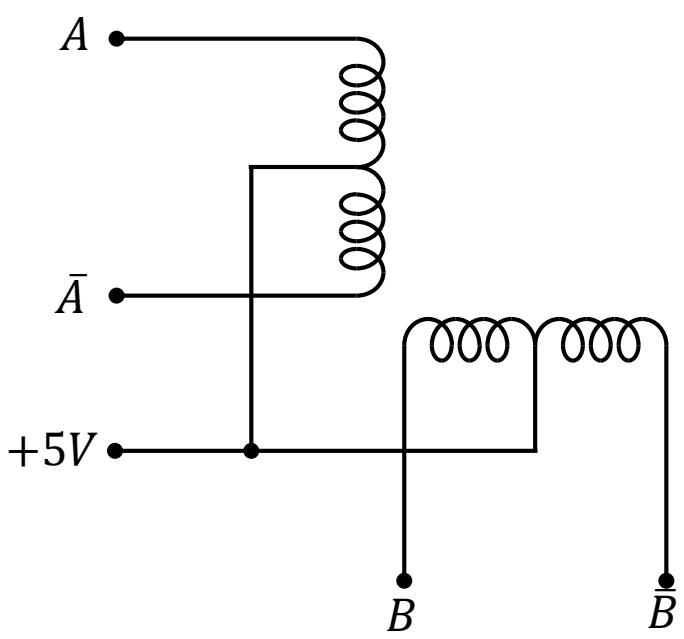
Stepper Motor 구성 및 동작 설명

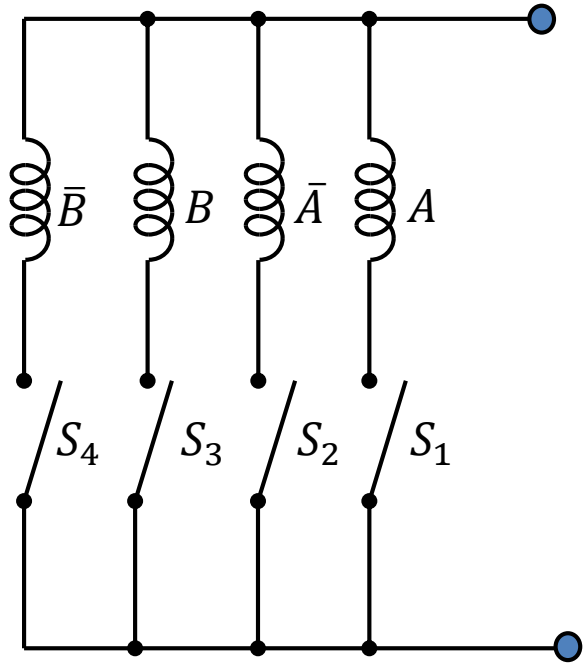
- Stepper Motor : 일정한 각도씩 회전하는 모터
  - 한 바퀴의 회전을 동일한 간격(스텝)으로 나눈 모터.
  - 3D 프린터, 산업용 로봇의 관절 기구 등 정밀한 제어에 사용됨.
  - 회전각 ∝ 입력 펄스 수.
  - 회전 속도 ∝ 입력 펄스 주파수(입력 펄스 수 / sec).
  - Open-Loop System으로 피드백 없이 정해진 각도(스텝)만큼 회전함.
  - 장점
    - ✓ 기동, 정지, 정-역 회전, 변속이 용이하며, 응답 특성도 양호함.
    - ✓ 1 스텝 당 각도 오차가 매우 적다.
    - ✓ DC 모터의 브러쉬에 의한 기계적인 마찰로 인한 파손이 없기 때문에, 신뢰성이 높다.

HUINS

Stepper Motor 구성 및 동작 설명

- Stepper Motor 구조
  - 4상 5선 stepper motor의 내부 결선 및 등가 회로





HUINS

Stepper Motor 구성 및 동작 설명

- 1상 여자방식
  - 하나의 상에만 전류를 흐르게 하는 방식
  - 1주기는 $A \rightarrow B \rightarrow \bar{A} \rightarrow \bar{B}$ 순서로 구성된다.
  - 입력이 1상 뿐이므로, 소비 전력이 낮다.
  - 출력 토크가 높으나, 감쇠 진동이 크고 난조를 일으키기 쉽다.

스텝	1	2	3	4
전류	A′→ A	B′→ B	$\bar{A}$′→ $\bar{A}$	$\bar{B}$′→ $\bar{B}$
회전자 위치				

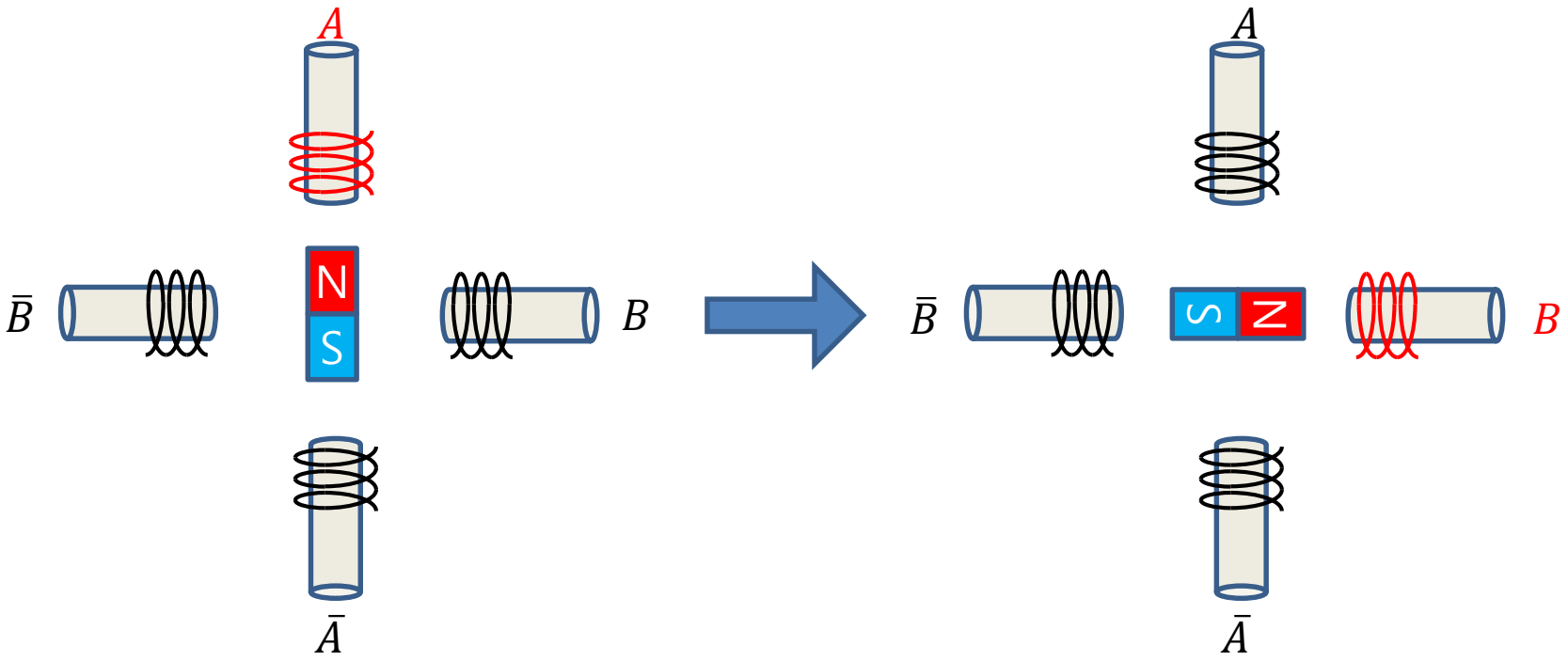
1주기

	1	2	3	4	5	6	7	8
A	1	0	0	0	1	0	0	0
B	0	1	0	0	0	1	0	0
/A	0	0	1	0	0	0	1	0
/B	0	0	0	1	0	0	0	1

HUINS

Stepper Motor 구성 및 동작 설명

- 1상 여자방식



HUINS

Stepper Motor 구성 및 동작 설명

- 2상 여자방식
  - 동시에 두 개의 상에 전류를 흐르게 하는 방식
  - 1주기는 $A, B \rightarrow B, \bar{A} \rightarrow \bar{A}, \bar{B} \rightarrow \bar{B}, A$순서로 구성된다.
  - 소비 전력이 1상 여자방식의 2배이며, 모터의 온도 상승이 있다.
  - 난조가 일어나기 어렵다.

	1	2	3	4
스텝 전류	A'->A B'->B	B'->B $\bar{A}$'->A	$\bar{A}$'->A $\bar{B}$'->B	$\bar{B}$'->B A'->A
회전자 위치				

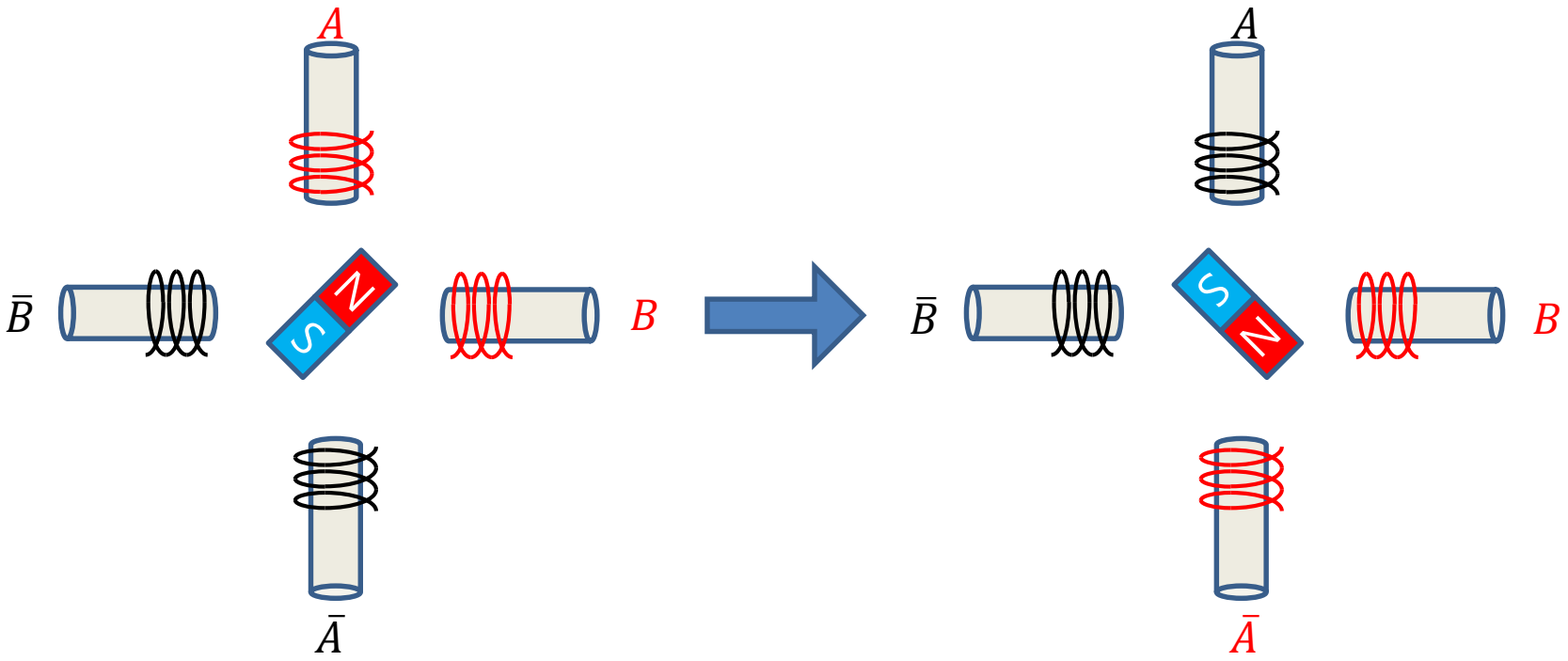
1주기

	1	2	3	4	5	6	7	8
A	1	0	0	1	1	0	0	1
B	1	1	0	0	1	1	0	0
/A	0	1	1	0	0	1	1	0
/B	0	0	1	1	0	0	1	1

HUINS

Stepper Motor 구성 및 동작 설명

- 2상 여자방식



HUINS

Stepper Motor 구성 및 동작 설명

- 1-2상 여자방식
  - 1상 여자 방식과 2상 여자 방식을 교대로 사용하는 방식
  - 45°씩 회전하므로 더욱 정밀한 제어가 가능
  - 한 바퀴 회전에 총 8개의 스텝이 필요

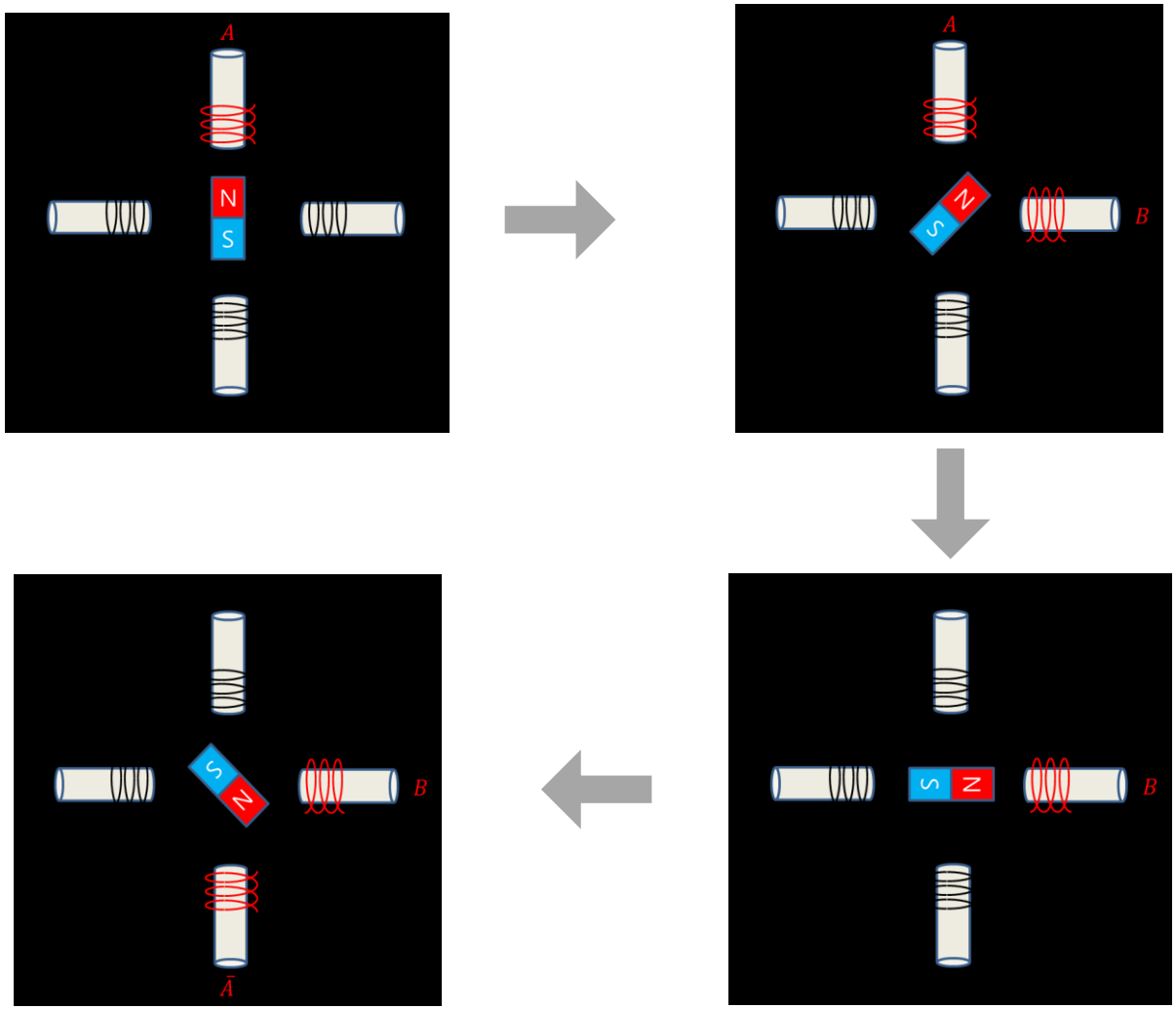
1주기

	1	2	3	4	5	6	7	8
A	1	1	0	0	0	0	0	1
B	0	1	1	1	0	0	0	0
/A	0	0	0	1	1	1	0	0
/B	0	0	0	0	0	1	1	1

HUINS

Stepper Motor 구성 및 동작 설명

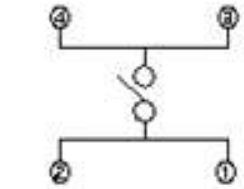
- 1-2상 여자방식

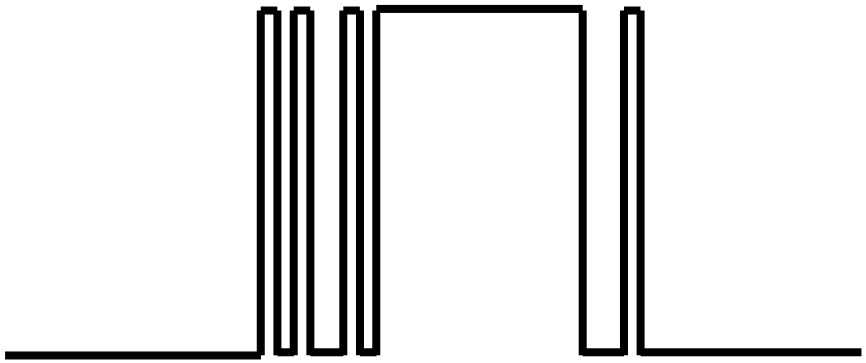


HUINS

Digital Debounce

- 바운싱(Bouncing)
  - 기계적 접점을 가지는 스위치는 물리적 특성으로 인해 누를 때와 뗄 때 무수히 많은 떨림이 발생하며 개/폐(open/close)신호가 발생한다.
  - 이러한 현상을 바운싱(Bouncing) 현상이라고 하며, 원하지 않는 입력이 발생할 수 있다.

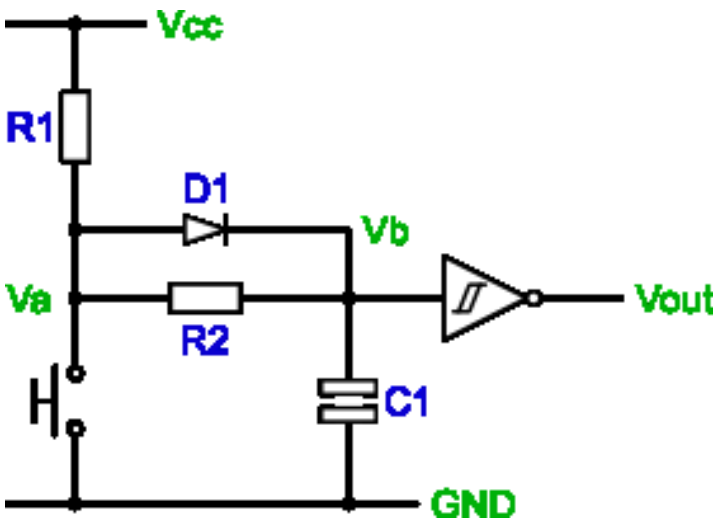


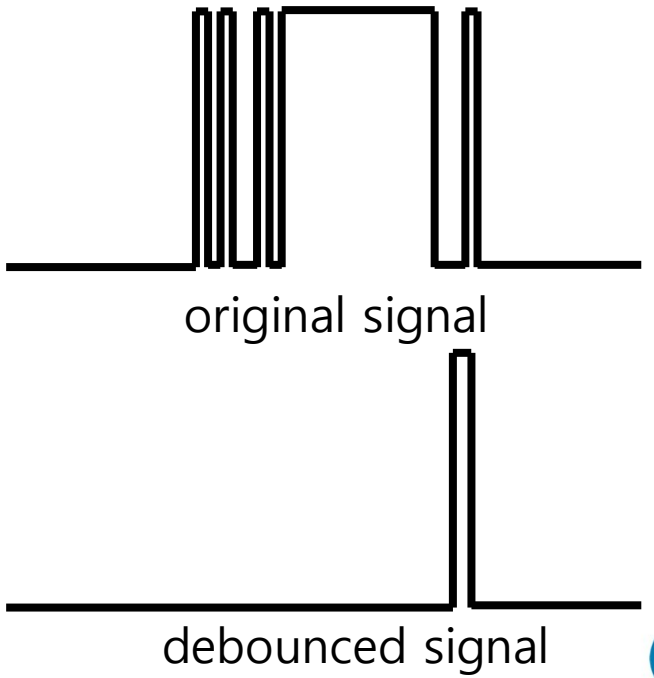


HUINS

Digital Debounce

- 바운싱 현상 제거
  - 바운싱 현상을 해결하는 방식으로는 물리적인 방식과, 소프트웨어적인 방식으로 해결이 가능하다.
    - ✓ 물리적 방식으로는 주로 캐패시터를 사용해 회로를 구성하여 해결한다.
    - ✓ 소프트웨어적인 방식으로는 연속된 값일 때 사용하는 등으로 해결한다.

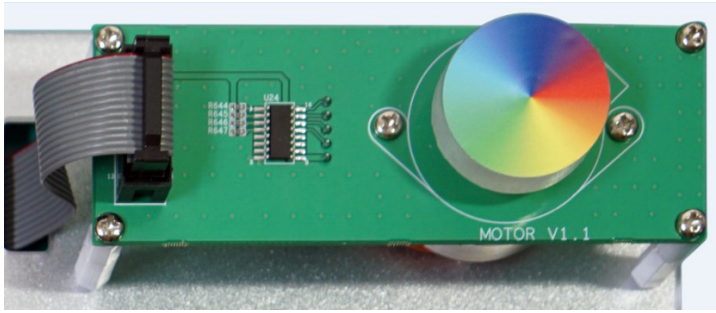




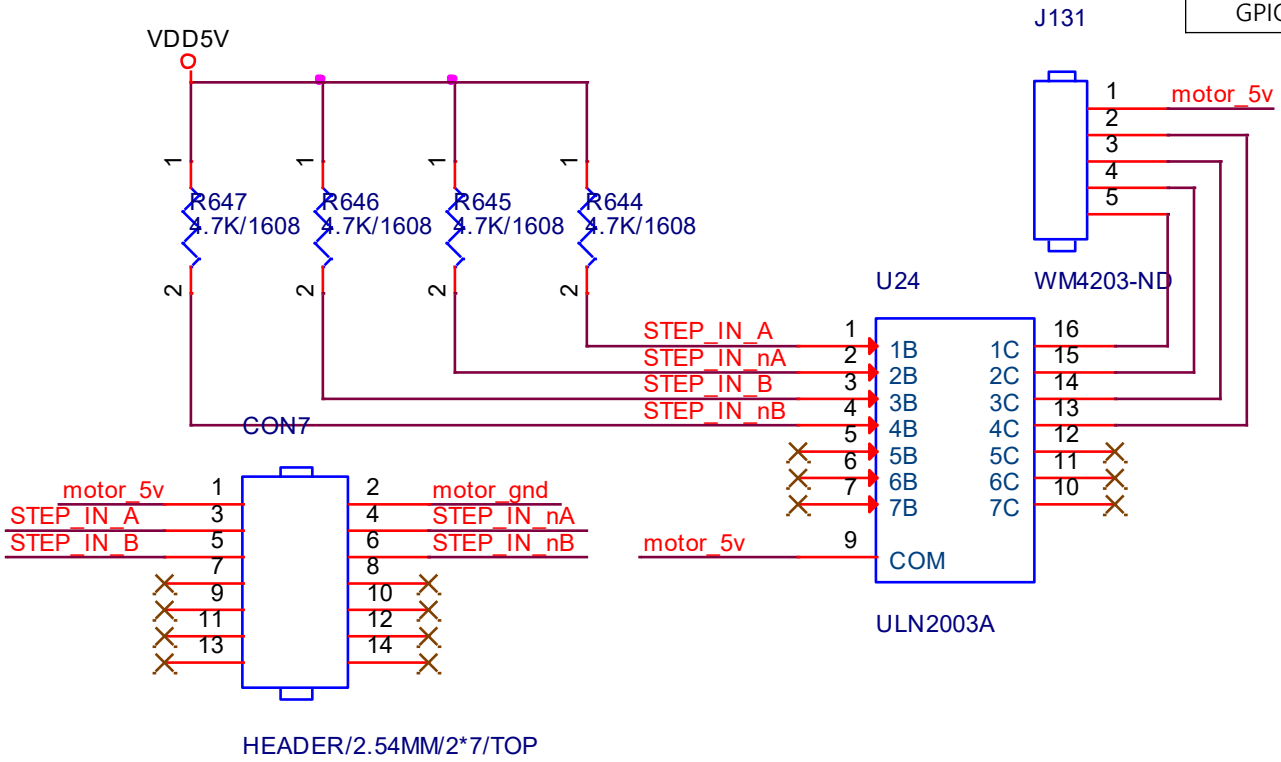
HUINS

Stepper Motor 회로도(Ext. Module)

- Stepper Motor 회로도

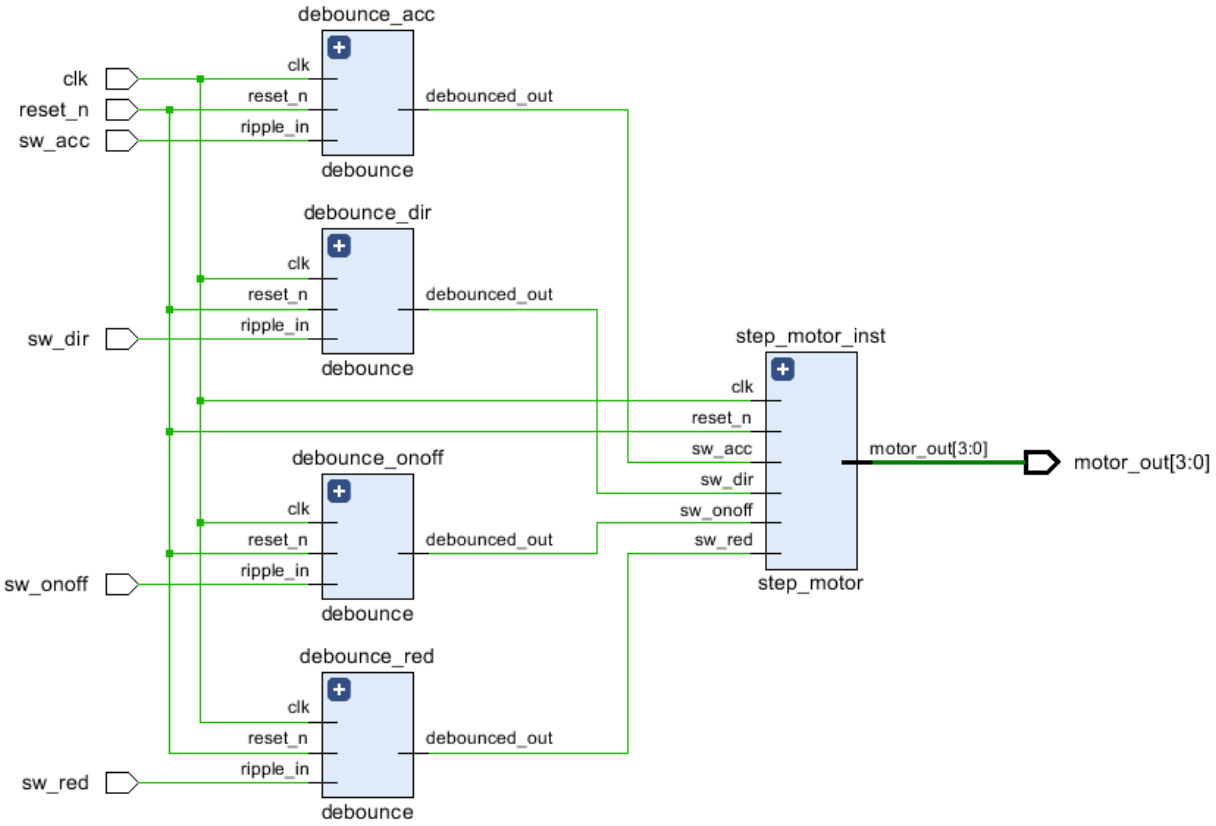


GPIO	Package Pin	External Module
GPIO24	K11	MOTOR[3]
GPIO25	K13	MOTOR[2]
GPIO26	L12	MOTOR[0]
GPIO27	G12	MOTOR[1]



Stepper Motor 제어

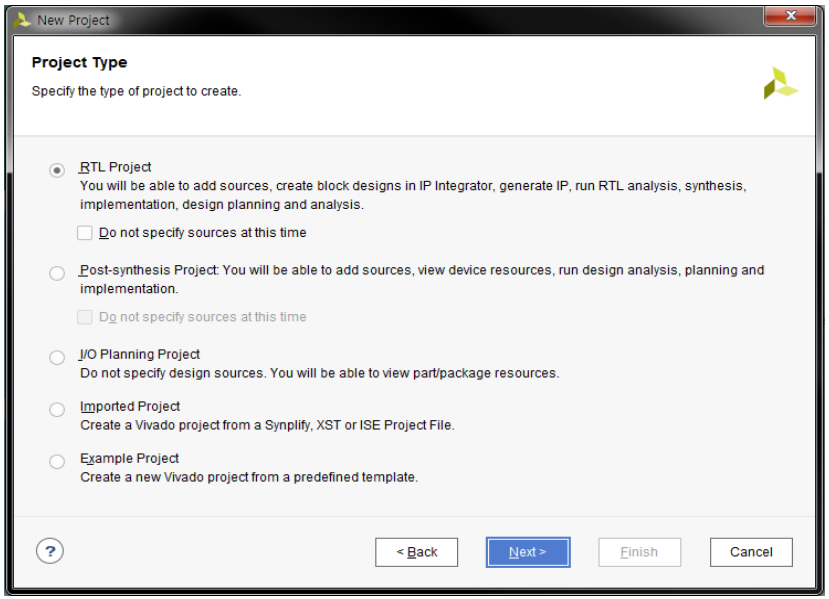
- Stepper Motor 제어 회로 블록 다이어그램
  - 외부로부터 입력된 4개의 푸쉬버튼은 각각 debounce 되어 step_motor 모듈로 입력된다.
  - step_motor 모듈에서는 제어 신호에 맞춰 motor_out (A, $\overline{A}$, B, $\overline{B}$)을 출력한다.

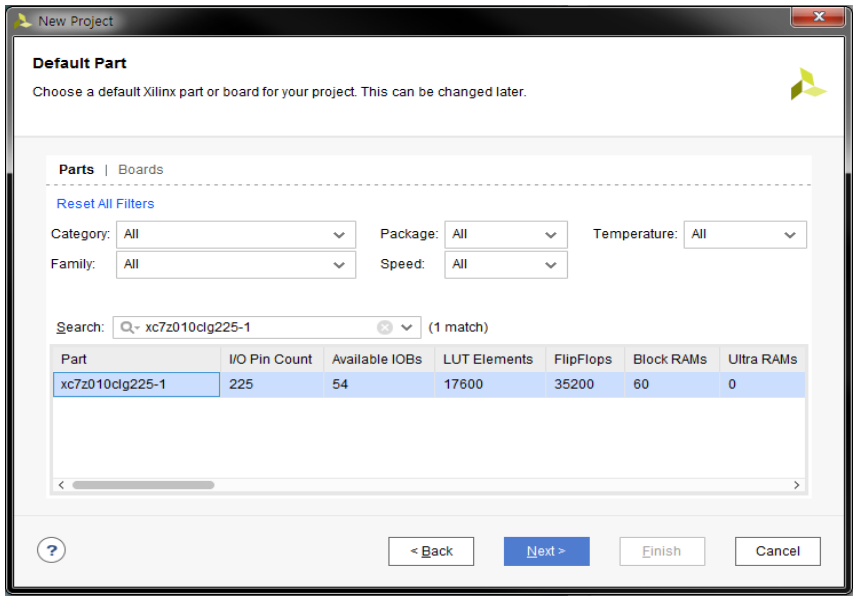


HUINS

Stepper Motor 제어 설계 프로젝트 생성하기

- 프로젝트 생성
  - 이름과 경로를 지정하여 새 프로젝트를 생성한다.
  - 프로젝트의 타입은 RTL Project로, Xilinx part는 xc7z010clg225-1로 설정한다.

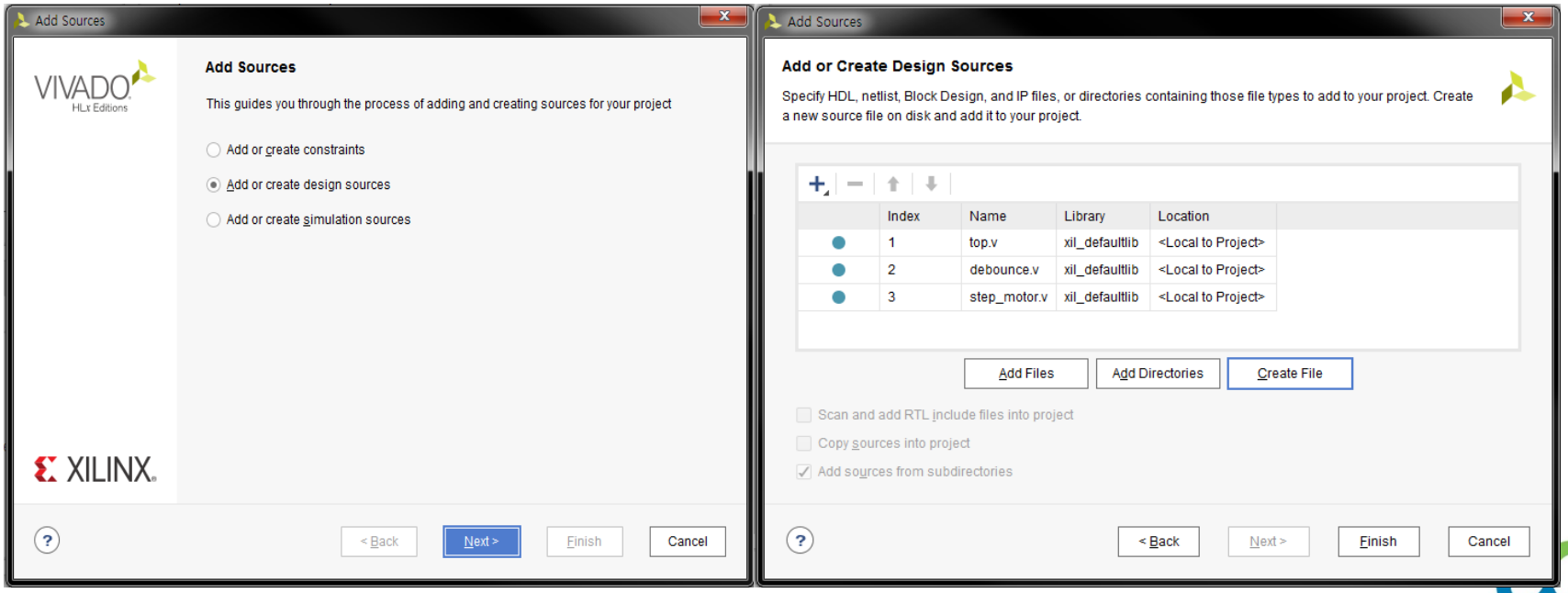




HUINS

Stepper Motor 제어

- Stepper Motor 제어 회로를 설계하기 위한 RTL 코드 파일 생성하기
  - Project Manager – Add sources에서 "Add or Create Design Sources"를 선택한다.
  - Create Files 버튼을 클릭해 RTL 코드를 추가한다.
    - ✓ top.v, debounce.v, step_motor.v



HUINS

소스코드 작성하기

- 소스코드는 총 3개의 파일로 구성되어 있다.
  - ✓ step_motor.v 작성하기(스텝모터 모듈)

```
`timescale 1ns / 1ps
module step_motor(
    input wire clk,
    input wire reset_n,
    input wire sw_onoff,//on_off
    input wire sw_dir,  //direction
    input wire sw_acc,  //acclerator
    input wire sw_red,  //reducer
    output reg [3:0] motor_out  //A,nA,B,nB
);

reg [29:0] motor_count;
reg motor_dir;
reg motor_onoff;
reg [1:0] motor_speed;

always@(posedge clk, negedge reset_n) begin
    if(!reset_n) begin
        motor_dir <= 1'b0;       //0_R, 1_L
        motor_onoff <= 1'b0;     //0_on, 1_off
        motor_speed <= 2'b10;    //1 - 3
    end
    else begin
        if(sw_onoff) motor_onoff <= ~motor_onoff;
        if(sw_dir) motor_dir <= ~motor_dir;
        if(sw_acc) if(motor_speed < 2'b11) motor_speed <= motor_speed + 1;  // 1 - 3
        if(sw_red) if(motor_speed > 2'b01) motor_speed <= motor_speed - 1;  // 3 - 1
    end
end
```

HUINS

소스코드 작성하기

- 소스코드는 총 3개의 파일로 구성되어 있다.
  - ✓ step_motor.v 작성하기(스텝모터 모듈)

```
always@(posedge clk, negedge reset_n) begin
    if(!reset_n) begin
        motor_count <= 30'd0;
        motor_out <= 4'b1001;
    end
    else begin
        if(motor_onoff == 0) begin     // Motor On
            if(motor_count > 30'd960000)
                motor_count <= 30'd0;
            else
                motor_count <= motor_count + motor_speed;   //MoterCount += motor_speed
            case(motor_count)
                0 : if(motor_dir) motor_out <= 4'b1001; else motor_out <= 4'b1001;
                240000 : if(motor_dir) motor_out <= 4'b0101; else motor_out <= 4'b1010;
                480000 : if(motor_dir) motor_out <= 4'b0110; else motor_out <= 4'b0110;
                720000 : if(motor_dir) motor_out <= 4'b1010; else motor_out <= 4'b0101;
                default : motor_out <= motor_out;
            endcase
        end
    end
end


endmodule
```

HUINS

소스코드 작성하기

- debounce.v 코드 작성

```
`timescale 1ns / 1ps

module debounce(
    input wire clk,
    input wire reset_n,
    input wire ripple_in,
    output wire debounced_out
    );

   reg [2:0] cnt = 3'b000;

always @(posedge clk) begin
   if (reset_n == 0)     //negative reset
      cnt <= 3'b000;
   else
      cnt <= {cnt[1:0], ~ripple_in};
end

assign debounced_out = cnt[0] & cnt[1] & !cnt[2];

endmodule
```

소스코드 작성하기

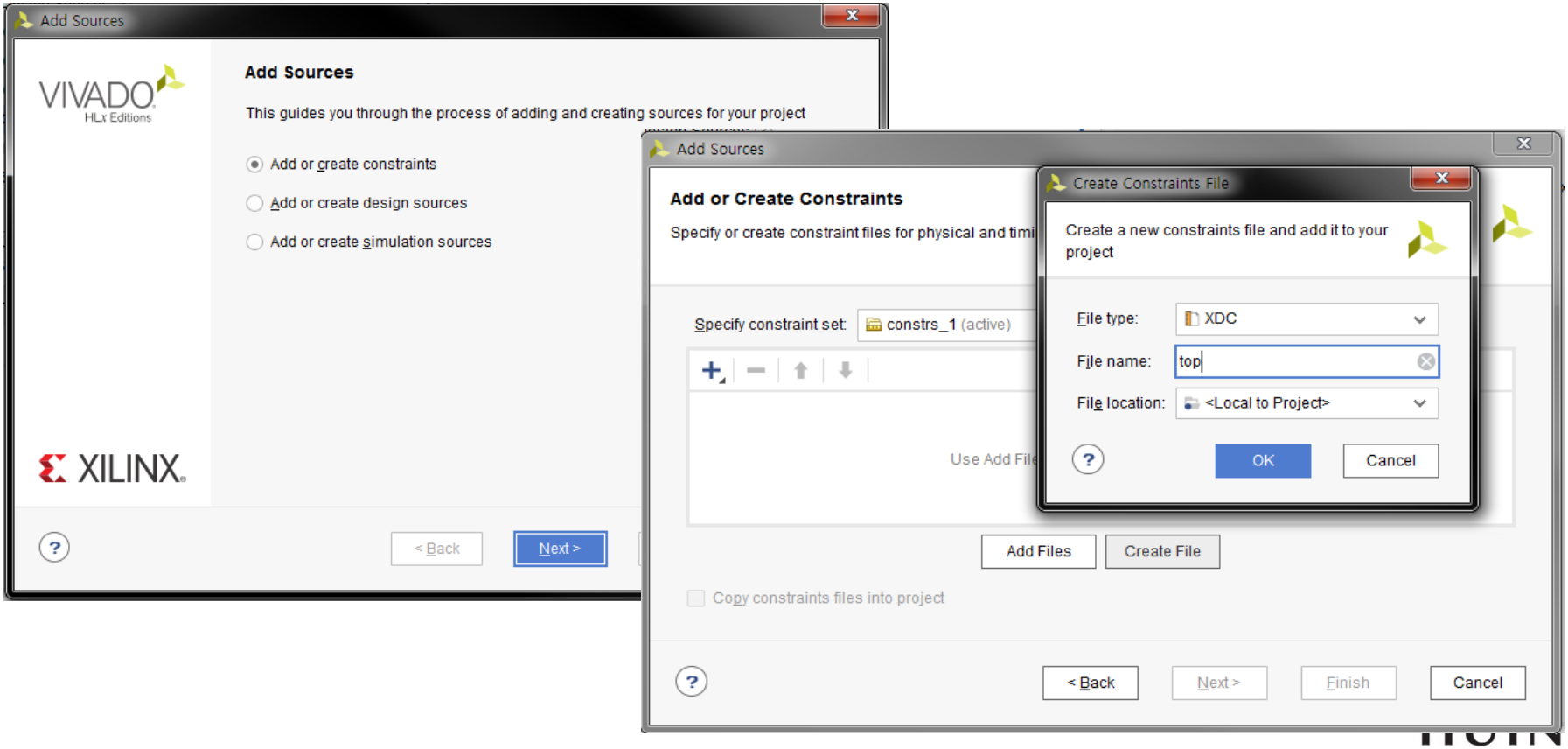
- top.v 코드 작성

```
`timescale 1ns / 1ps

module top(
    input wire clk,
    input wire reset_n,
    input wire sw_onoff,//on_off
    input wire sw_dir,  //direction
    input wire sw_acc,  //acclerator
    input wire sw_red,  //reducer
    output wire [3:0] motor_out  //A,nA,B,nB
);

wire db_onoff;
wire db_dir;
wire db_acc;
wire db_red;

step_motor step_motor_inst(
    .clk(clk),
    .reset_n(reset_n),
    .sw_onoff(db_onoff),//on_off
    .sw_dir(db_dir),  //direction
    .sw_acc(db_acc),  //acclerator
    .sw_red(db_red),  //reducer
    .motor_out(motor_out)
);

debounce debounce_onoff(clk, reset_n, sw_onoff, db_onoff);
debounce debounce_dir(clk, reset_n, sw_dir, db_dir);
debounce debounce_acc(clk, reset_n, sw_acc, db_acc);
debounce debounce_red(clk, reset_n, sw_red, db_red);

endmodule
```

HUINS

프로젝트 컴파일 하기

- Constraints 설정
  - Project Manager - Add Sources에서 "Add or create constrains" 항목을 선택한다.
  - Create File을 선택해 top.xdc 파일을 생성한다.



프로젝트 컴파일 하기

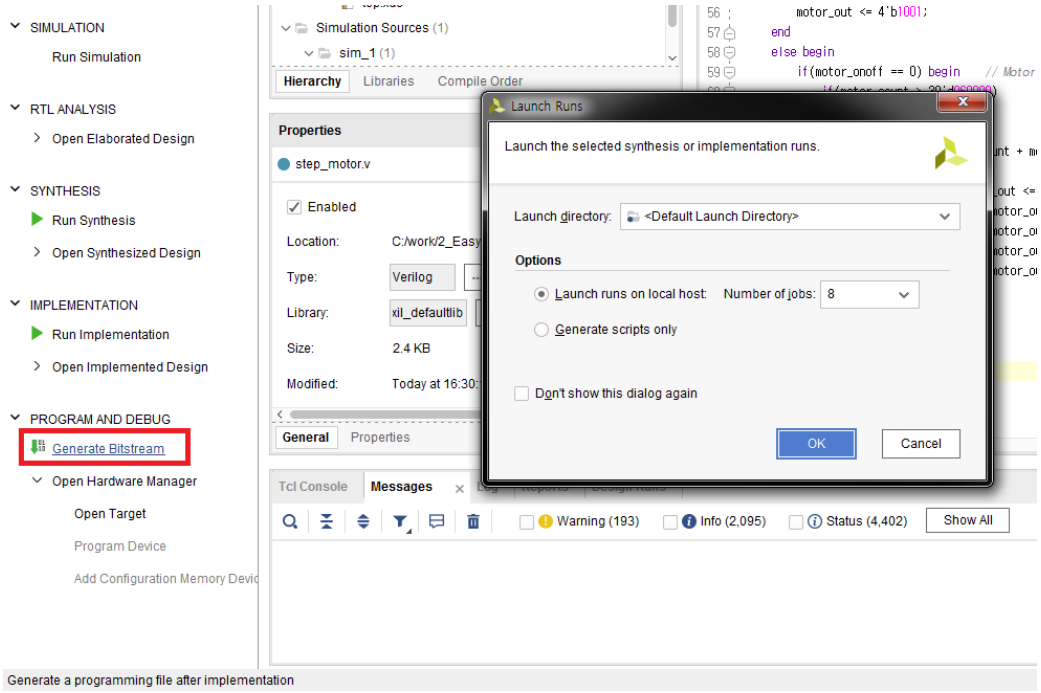
- top.xdc 파일의 내용을 다음과 같이 작성한다.

```
set_property IOSTANDARD "LVCMOS33"      [get_ports "clk"]
set_property PACKAGE_PIN "K15"          [get_ports "clk"]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets "clk"]

set_property IOSTANDARD "LVCMOS33" [get_ports "reset_n"]
set_property PACKAGE_PIN "H14"       [get_ports "reset_n"]

set_property IOSTANDARD "LVCMOS33" [get_ports "sw_onoff"]
set_property PACKAGE_PIN "J14"       [get_ports "sw_onoff"]

set_property IOSTANDARD "LVCMOS33" [get_ports "sw_dir"]
set_property PACKAGE_PIN "H12" [get_ports "sw_dir"]

set_property IOSTANDARD "LVCMOS33" [get_ports "sw_acc"]
set_property PACKAGE_PIN "N14" [get_ports "sw_acc"]

set_property IOSTANDARD "LVCMOS33" [get_ports "sw_red"]
set_property PACKAGE_PIN "R15" [get_ports "sw_red"]

set_property IOSTANDARD "LVCMOS33" [get_ports "motor_out[*]"]
set_property PACKAGE_PIN “K11" [get_ports "motor_out[0]"]
set_property PACKAGE_PIN “L12" [get_ports "motor_out[1]"]
set_property PACKAGE_PIN “K13" [get_ports "motor_out[2]"]
set_property PACKAGE_PIN “G12" [get_ports "motor_out[3]"]
```

HUINS

프로젝트 컴파일 하기

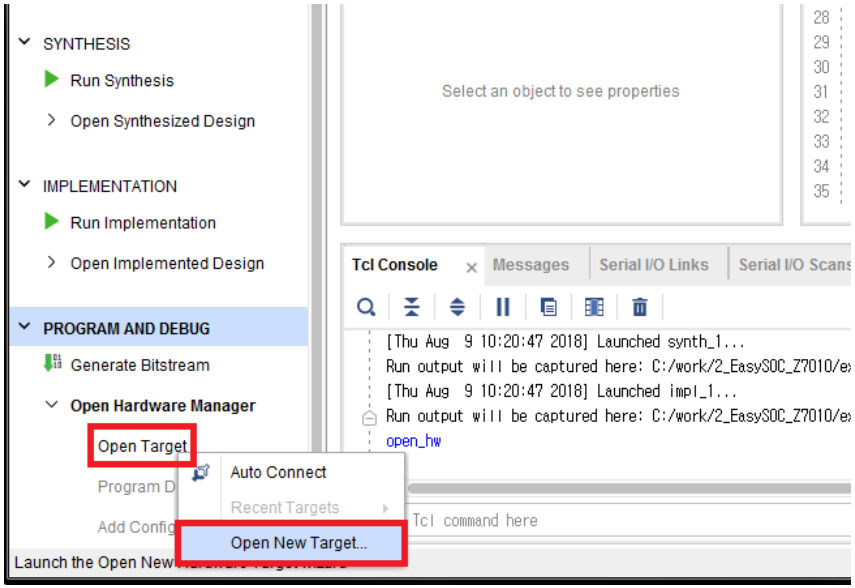
- 프로젝트 컴파일 하기
  - PROGRAM AND DEBUG – Generate Bitstream을 실행하여 PL영역에 프로그램 할 Bitstream파일을 생성한다.
  - Generate Bitstream을 실행할 경우 순서대로 Synthesis, Implementation, Generate Bitstream 순으로 진행이 된다.

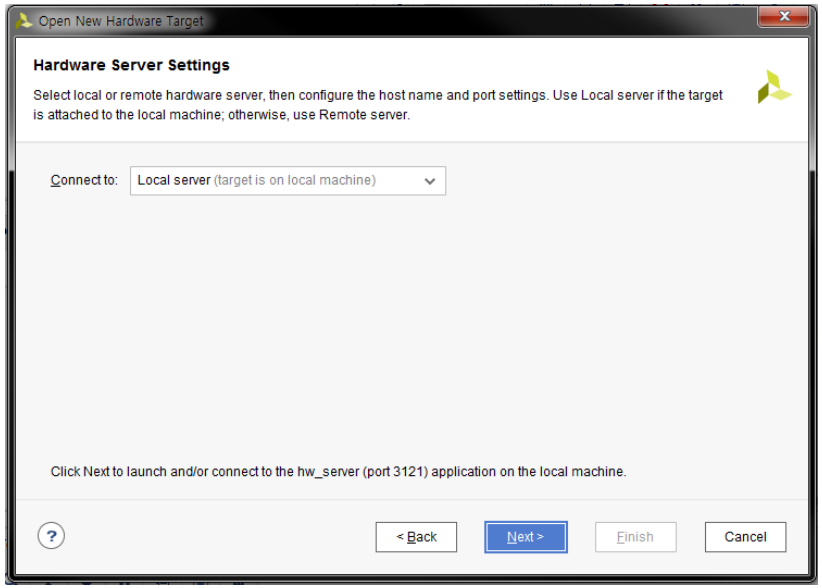


HUINS

EasySoC-Z7010에서 검증하기

- Host에 타겟 연결하기
  - PROGRAM AND DEBUG – Open Hardware Manger – Open Target에서 Open New Target을 클릭한다.
  - PC에 보드를 연결한 뒤 Local server로 연결한다.

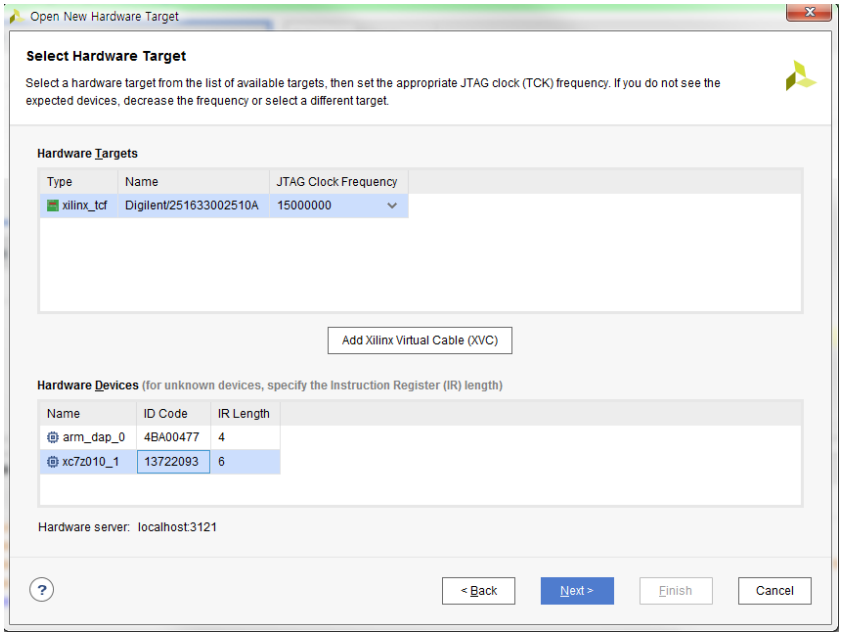


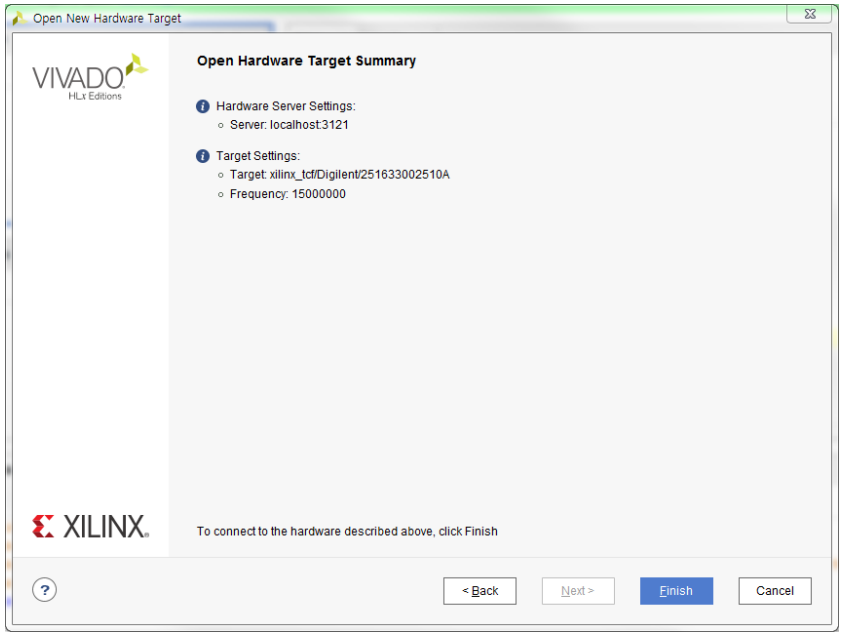


HUINS

EasySoC-Z7010에서 검증하기

- Host에 타겟 연결하기
  - Hardware Devices에 XC7Z010_1을 선택하고 next 버튼을 클릭한다.
  - ZYNQ-7000 PL에 Bitstream 파일을 다운로드 하기 위한 설정 내용을 알려주는 대화상자에서 Finish버튼을 클릭해 연결을 완료한다.

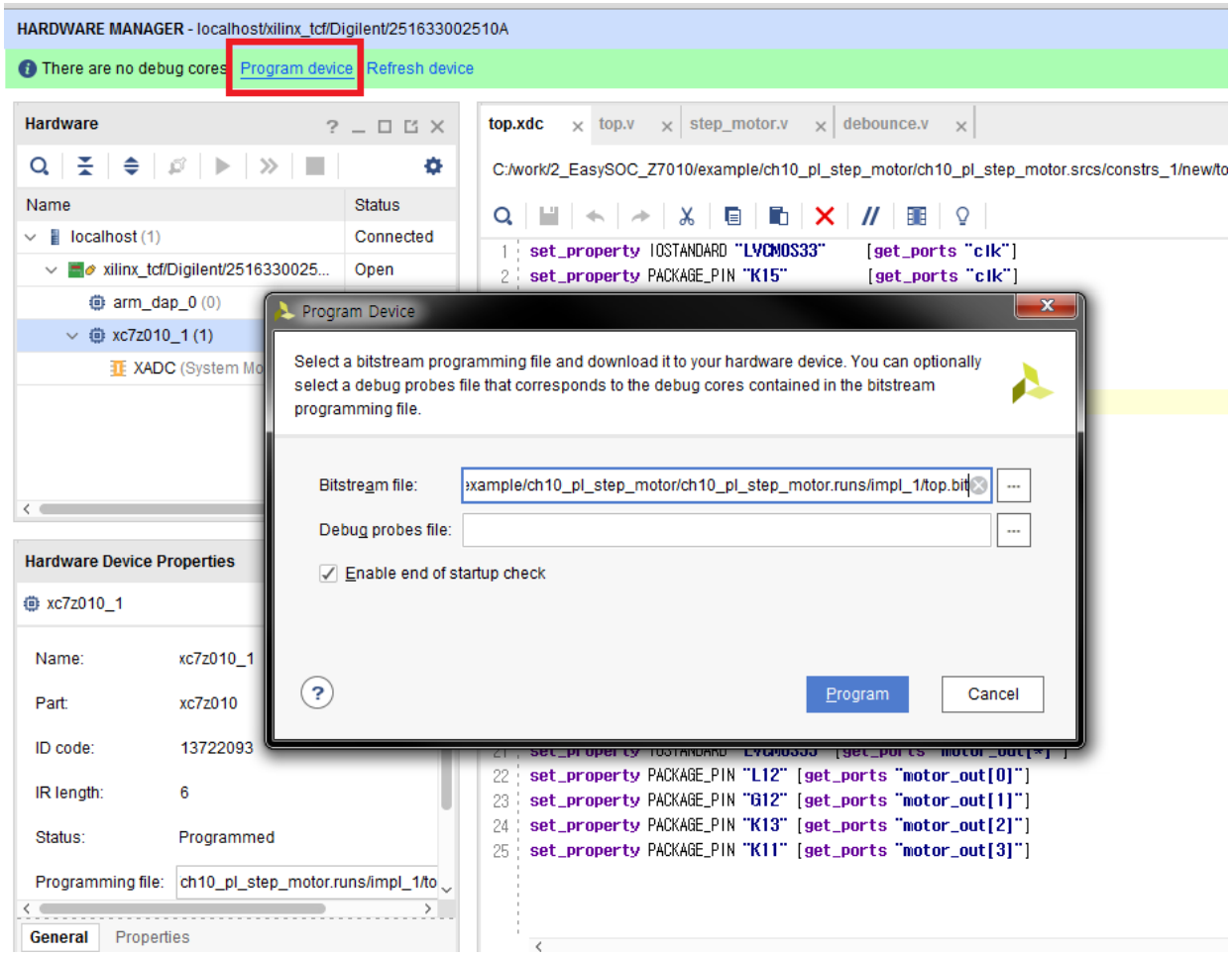




HUINS

EasySoC-Z7010에서 검증하기

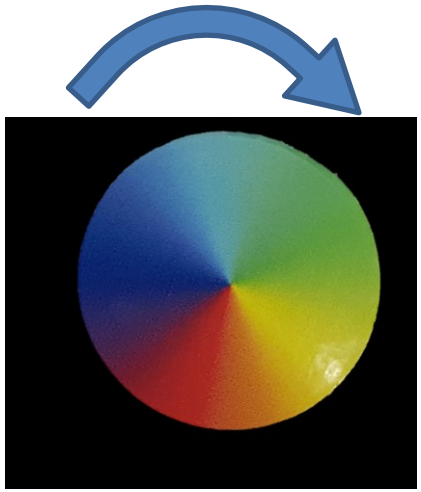
- HARDWARE MANAGER에서 Program Device를 실행한다.
- Z7010의 PL 영역에 다운로드 할 Bitstream file을 선택하고 Program한다.



HUINS

EasySoC-Z7010에서 검증하기

- 다운로드가 완료되면, stepper motor가 오른쪽으로 회전한다.
- 푸쉬버튼으로 모터의 작동을 제어할 수 있다.
  - ✓ SW3 : 동작/일시정지
  - ✓ SW4 : 시계/반시계
  - ✓ SW5 : 가속
  - ✓ SW6 : 감속
  - ✓ SW10 : 리셋버튼



HUINS

감사합니다.

HUINS