



TFT-LCD (FPGA Programming)

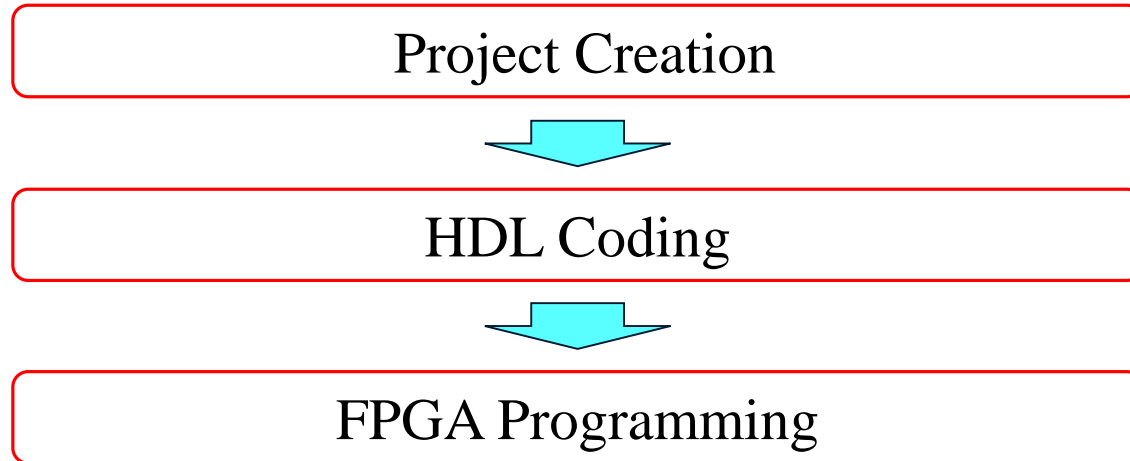
문 병 인

bihmoon@knu.ac.kr

경북대학교 전자공학부



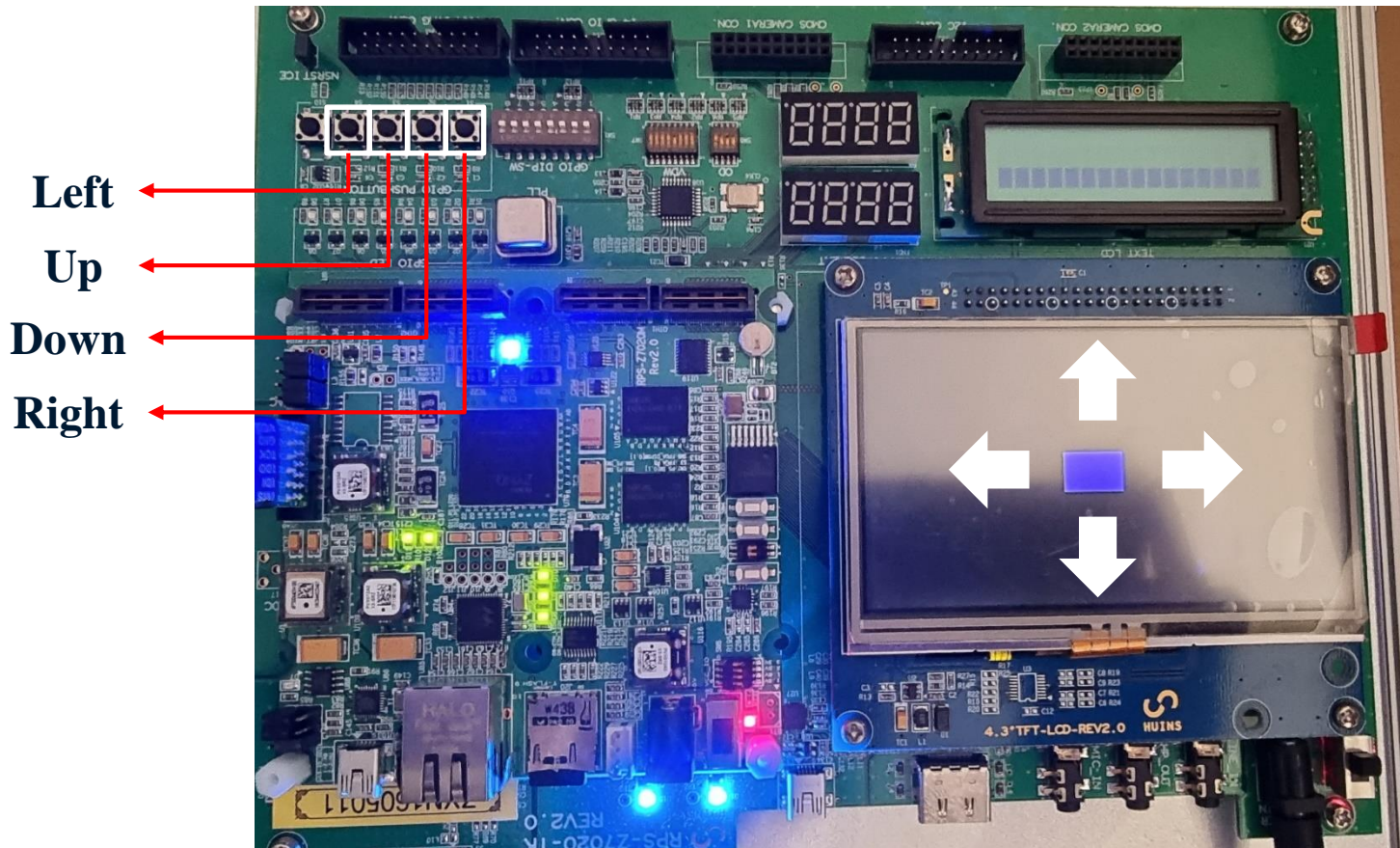
- 배경 지식
- 실습 과제 개요
- 실습 환경
- 실습 과정



TFT-LCD Controller Design

3

□HDL을 이용하여 Pushbutton 입력에 따라 TFT-LCD에 표시되고 있는 물체의 위치를 변경시켜주는 회로 설계

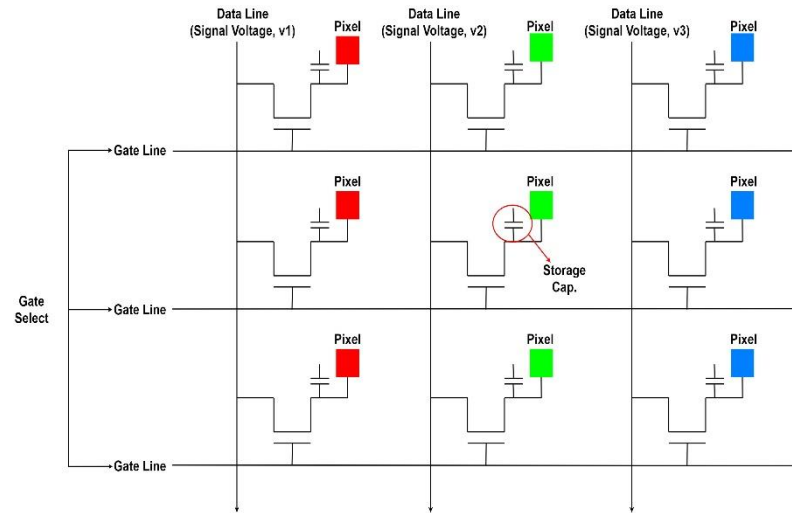


□ TFT-LCD 구동 원리

- ❖ Gate Line에 전압(신호)을 인가
- ❖ TFT가 Turn-on됨
- ❖ Pixel에 액정을 구동할 수 있는 신호 인가

□ Example (2번째 Gate Line 기준)

- ❖ Gate 전극에 전압이 인가되고 있으며, 또한 모든 Source Line에 신호 전압이 인가됨
- ❖ 신호 전압 V_2 에 액정을 구동 시킬 수 있는 신호가 인가되고, 나머지 V_1 과 V_2 에 구동 전압보다 낮은 신호가 인가되면 V_2 Pixel 화소만 동작 가능함
- ❖ Gate 전압 신호가 인가되지 않은 라인의 TFT들은 모두 Turn-off 상태를 유지하며, Storage Cap에 인가된 신호를 유지시킴

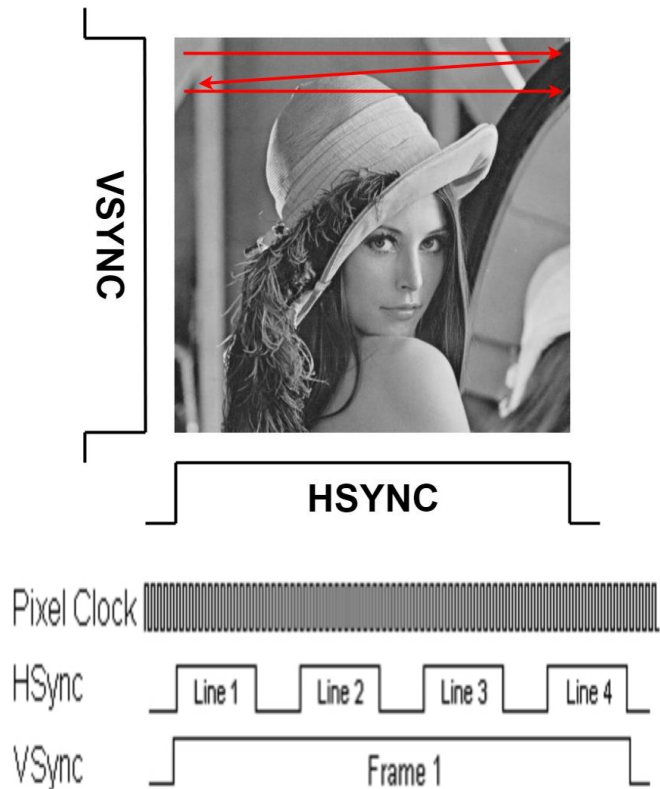


□ VSync(Vertical Synchronize)

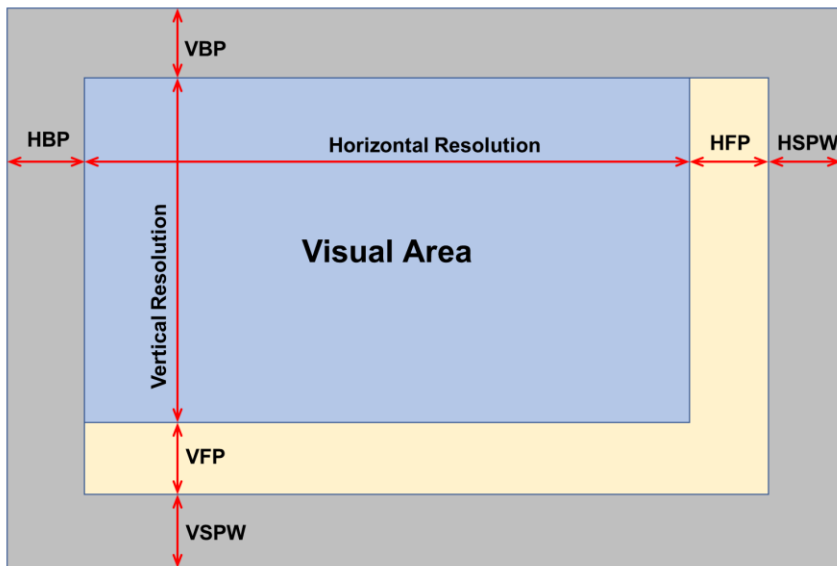
- ❖ 1장의 영상을 프레임(frame)이라 함
- ❖ Vsync는 한 프레임이 시작될 때 시작 지점을 알려주는 역할
- ❖ 영상을 읽기(read) 시작하는 좌표 지점부터 영상의 마지막 픽셀 지점까지 Vsync는 high값을 유지함

□ HSync(Horizontal Synchronize)

- ❖ 영상은 여러 개의 줄(line)으로 구성됨
- ❖ 영상에서 한 줄이 시작될 때 시작 지점을 알려주는 역할
- ❖ 영상을 읽은(read) 뒤 LCD에 쓰기(write) 위해서는 왼쪽의 Lenna 영상과 같이 빨간색 라인을 따라 각 줄을 읽고 쓰는 과정을 거침



□ TFT-LCD Display Signal



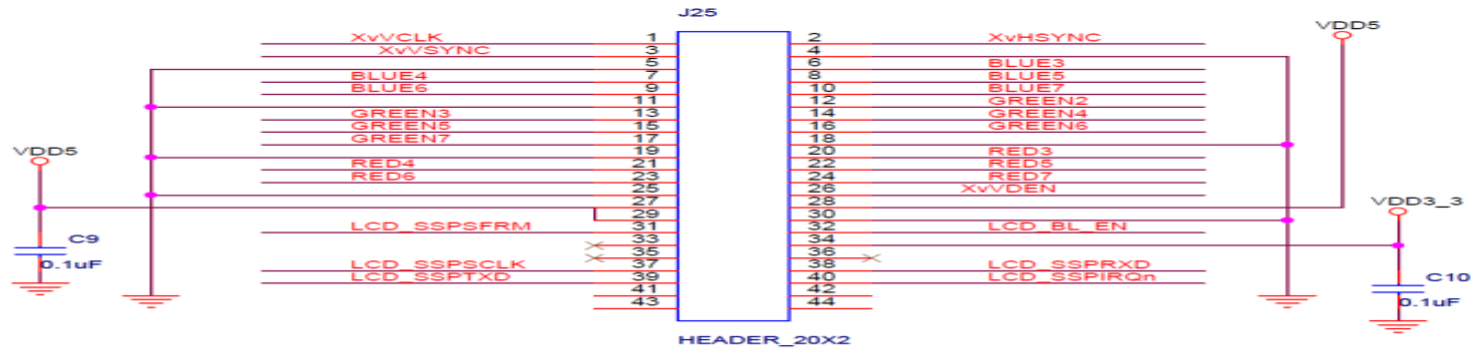
- ❖ **VBP: Vertical Back Porch**
- ❖ **VFP: Vertical Front Porch**
- ❖ **VSPW: Vertical Sync Pulse Width**
- ❖ **HBP: Horizontal Back Porch**
- ❖ **HFP: Horizontal Front Porch**
- ❖ **HSPW: Horizontal Sync Pulse Width**
- ❖ **TFT-LCD에 화면을 띄우기 위해서는 porch 구간과 pulse width를 고려해야 함**

배경 지식 (4/5)

7

□ TFT-LCD I/O ports

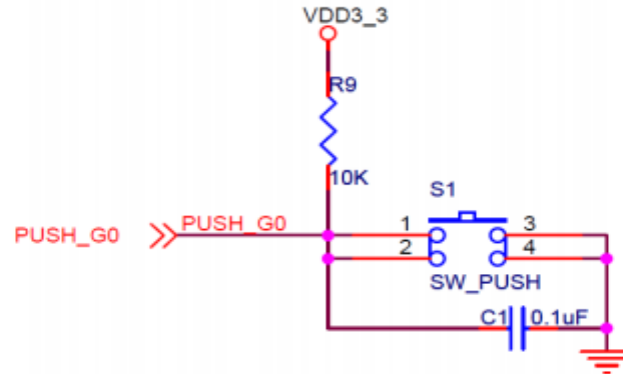
❖ 사용해야하는 I/O ports는 다음과 같음



NET	PL(FPGA) Pin	CONNECTOR	Device	Signal Name	BANK
FPGA_CL54	L17	QTH2-61	J25-1	XvVCLK	34
FPGA_CL55	M17	QTH2-63	J25-3	XvVSYNC	34
FPGA_CL56	N17	QTH2-65	J25-2	XvHSYNC	34
FPGA_CL57	N18	QTH2-67	J25-6	BLUE3	34
FPGA_CL58	M15	QTH2-69	J25-7	BLUE4	34
FPGA_CL59	M16	QTH2-71	J25-8	BLUE5	34
FPGA_CL60	J18	QTH2-73	J25-9	BLUE6	34
FPGA_CL61	K18	QTH2-75	J25-10	BLUE7	34
FPGA_CL62	J21	QTH2-77	J25-12	GREEN2	34
FPGA_CL63	J22	QTH2-79	J25-13	GREEN3	34
FPGA_CL64	J20	QTH2-81	J25-14	GREEN4	34
FPGA_CL65	K21	QTH2-83	J25-15	GREEN5	34
FPGA_CL66	L21	QTH2-85	J25-16	GREEN6	34
FPGA_CL67	L22	QTH2-87	J25-17	GREEN7	34
FPGA_CL68	K19	QTH2-89	J25-20	RED3	34
FPGA_CL69	K20	QTH2-91	J25-21	RED4	34
FPGA_CL96	T21	QTH2-97	J25-22	RED5	33
FPGA_CL97	U21	QTH2-99	J25-23	RED6	33
FPGA_CL98	T22	QTH2-101	J25-24	RED7	33
FPGA_CL99	U22	QTH2-103	J25-26	XvVDEN	33
FPGA_CL100	V22	QTH2-105	J25-31	LCD_SSPTSFRM	33
FPGA_CL101	W22	QTH2-107	J25-32	LCD_BL_EN	33
FPGA_CL102	W20	QTH2-109	J25-37	LCD_SSPTSCLK	33
FPGA_CL103	W21	QTH2-111	J25-39	LCD_SSPTXD	33
FPGA_CL104	U20	QTH2-113	J25-38	LCD_SSPTXD	33
FPGA_CL105	V20	QTH2-115	J25-40	LCD_SSPIRQn	33

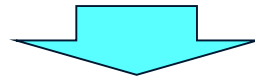
□ Pushbutton I/O ports

- ❖ 본 예제에서 사용하는 Pushbutton은 눌려졌을 때 Low가 되고, 초기상태에서는 High를 유지함
- ❖ 사용하는 I/O ports는 다음과 같음

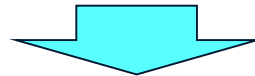


NET	PL(FPGA) Pin	CONNECTOR	Device	Signal Name	BANK
FPGA_CL108	Y18	QTH2-98	S1_1/2	PUSH_G0	33
FPGA_CL109	AA18	QTH2-100	S2_1/2	PUSH_G1	33
FPGA_CL110	Y19	QTH2-102	S3_1/2	PUSH_G2	33
FPGA_CL111	AA19	QTH2-104	S4_1/2	PUSH_G3	33

Project Creation



HDL Coding



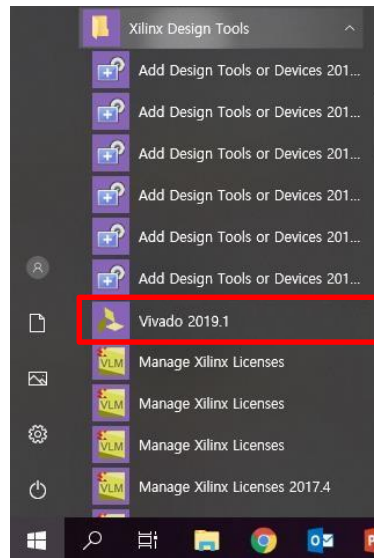
FPGA Programming

□ Vivado 19.1

- ❖ Xilinx의 7 series FPGA 및 ZYNQ를 활용하기 위한 IDE(Integrated Development Environment) tool

□ Vivado 19.1 설치 확인

- ❖ 시작 → 모든 프로그램 → Xilinx Design Tools → Vivado 2019.1 → Vivado 2019.1 또는 바탕화면에 있는 Vivado 2019.1 아이콘을 click



<바탕화면 아이콘>

실습 환경 (2/3)

11

□ 실습 환경을 위한 directory 생성

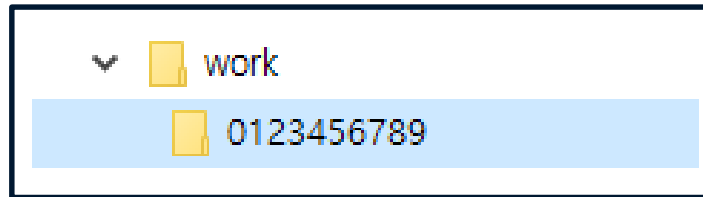
❖ *Directory 생성 시 주의점*

➤ 폴더 경로에 한글이 있는 경우 Vivado 에서 error 발생

❖ C 폴더 루트(c:\) 에 work 폴더를 생성

➤ work 폴더 내부에 *학번을 이용한 폴더(ex. 0123456789)* 를 생성

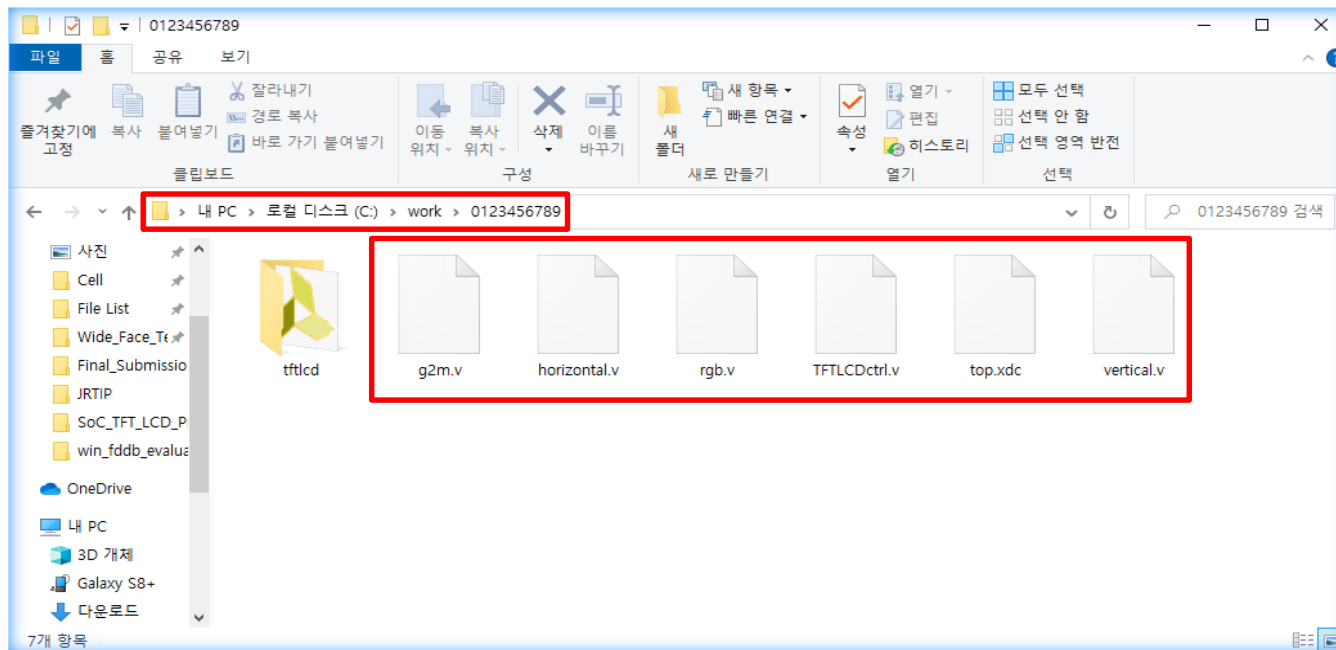
➤ 금일 실습을 수행할 폴더



□ 실습에서 사용할 file download

- ❖ 수업 게시판에서 verilog source file(**g2m.v**, **horizontal.v**, **rgb.v**, **TFTLCDCtrl.v**, **vertical.v**) constraint file(**top.xdc**)을 download 받은 후, 다음 경로에 위치

➤ “C:/work/“**학번**”/” 에 위치



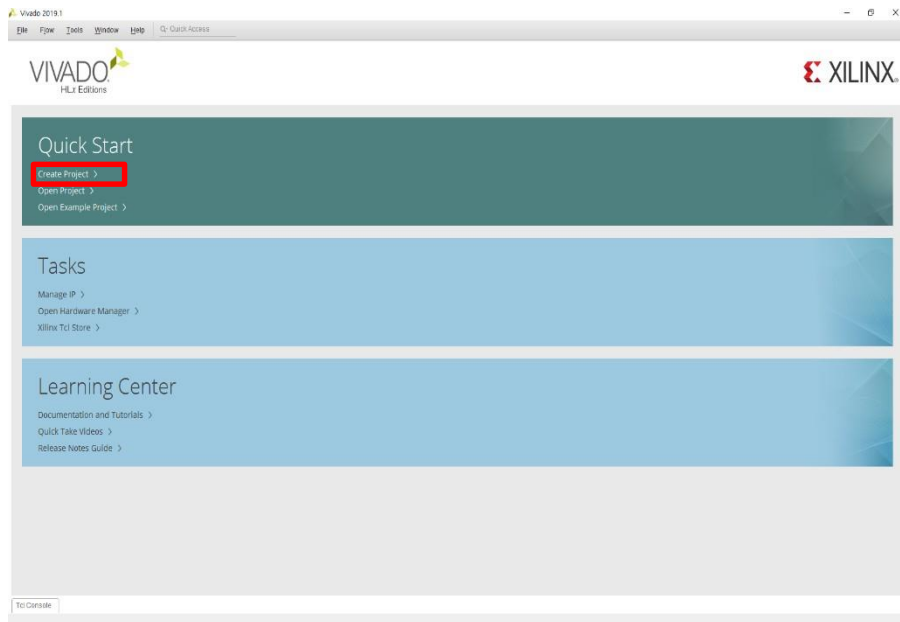
Project Creation (1/8)

13

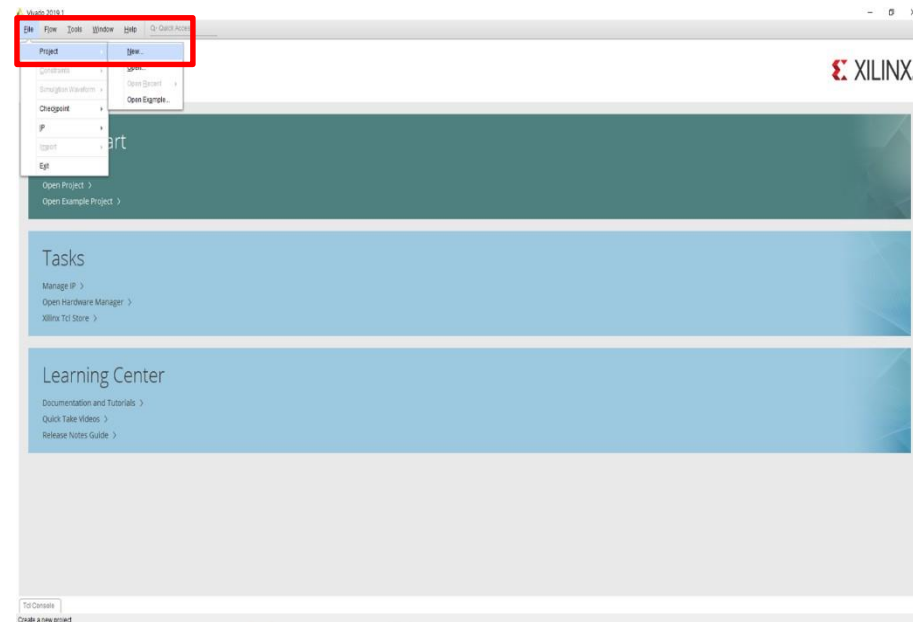
□ 새 project 생성

❖ *Vivado* 는 *project* 단위의 개발 환경을 제공

- Quick Start → Create Project 또는
- File → Project → New



or



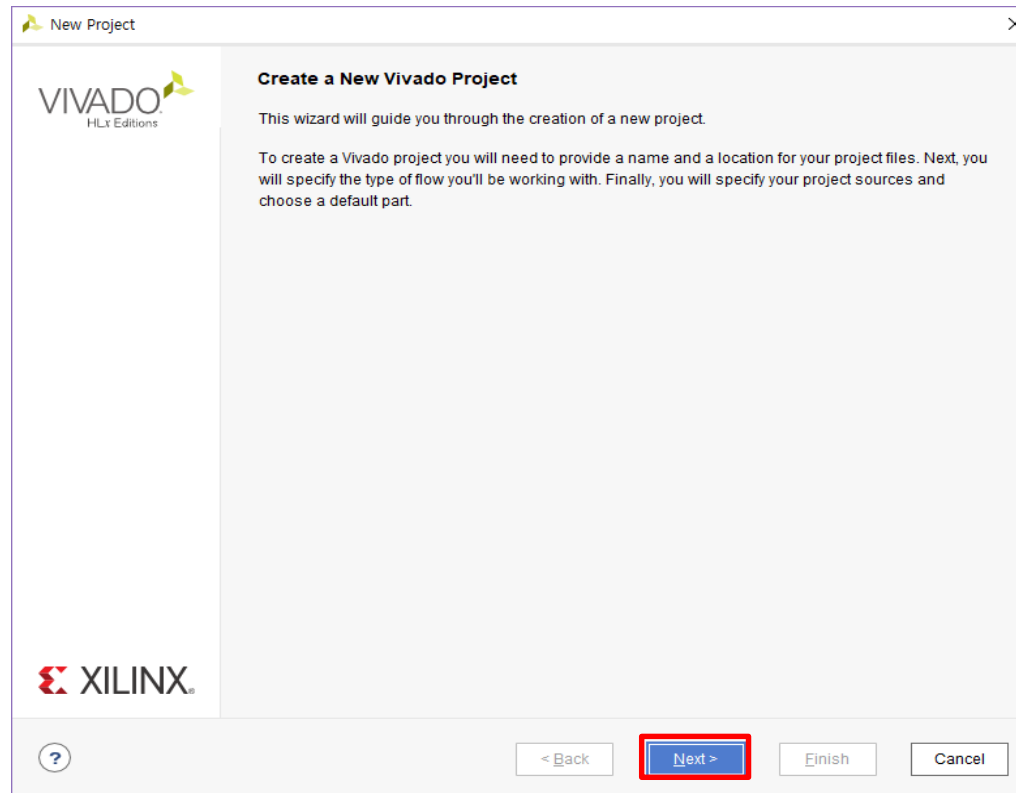
Project Creation (2/8)

14

□ New Project wizard

❖ *Project 생성에 관한 안내 문구와 함께 New Project wizard 실행*

➤  버튼을 클릭하여 New Project wizard 시작



Project Creation (3/8)

15

□ Project 경로 설정

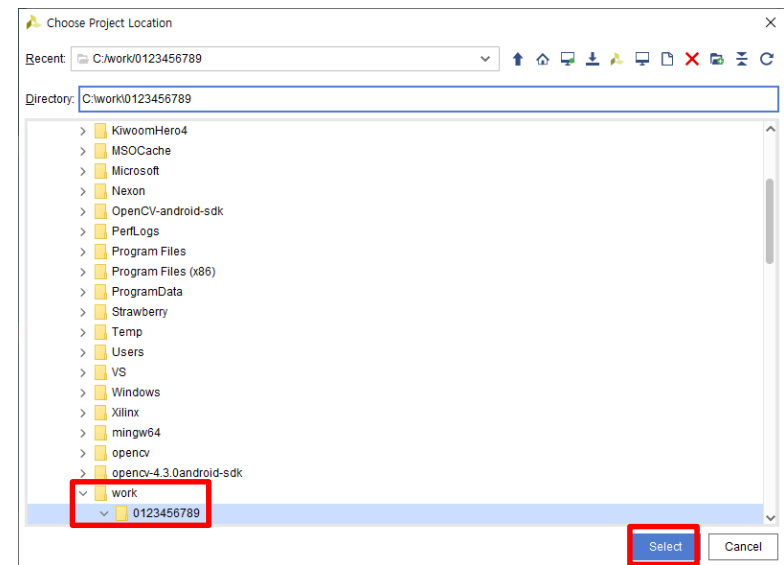
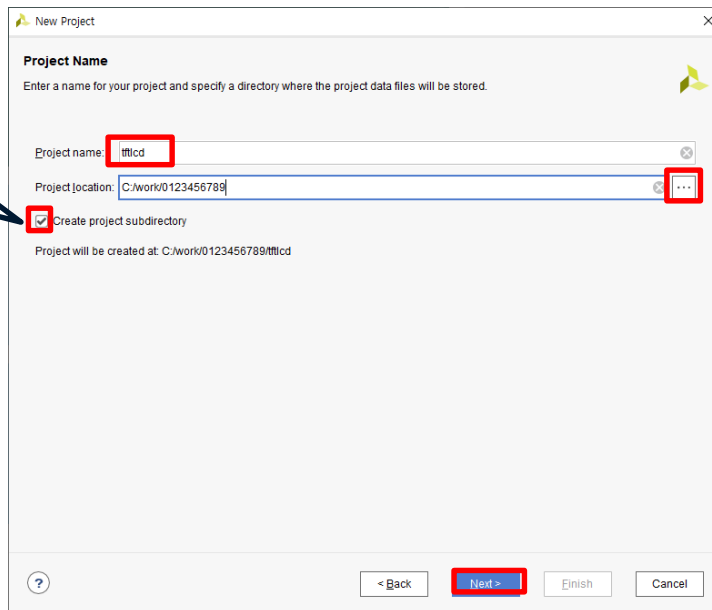
❖ 해당 project의 이름과 경로를 설정

➤ Project name : tftlcd

➤ Project location : ... 버튼 click

▪ 컴퓨터 → 로컬 디스크 (C:) → work → “학번 폴더” 클릭 후, Select 클릭

➤ Create project subdirectory : project name으로 하위 폴더를 생성(check)



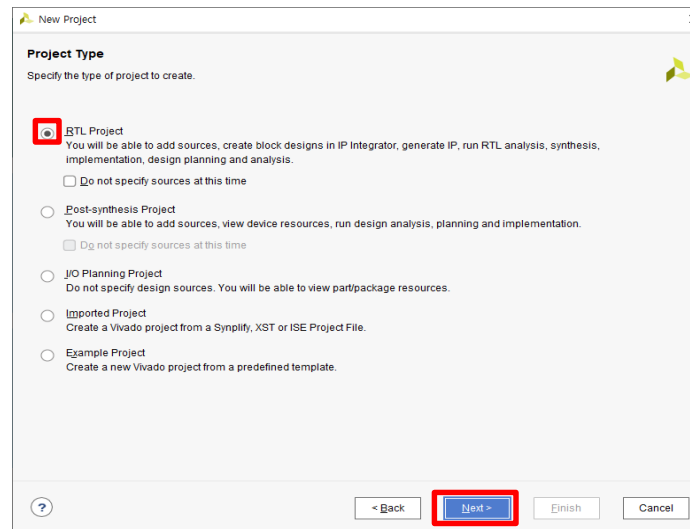
Project Creation (4/8)

16

□ Project Type 설정

❖ Project Type 을 설정 (Project Type 에 따라 개발 flow 가 달라짐)

- RTL Project는 HDL 코딩, synthesis, place and route, 등 FPGA와 ZYNQ 개발의 A to Z를 제공
- RTL Project 외의 Project Type은 다른 project의 결과를 불러와 활용
- Do not specify sources at this time : project 생성 시 소스 파일을 추가하지 않음(추후에 필요 시 추가 가능)



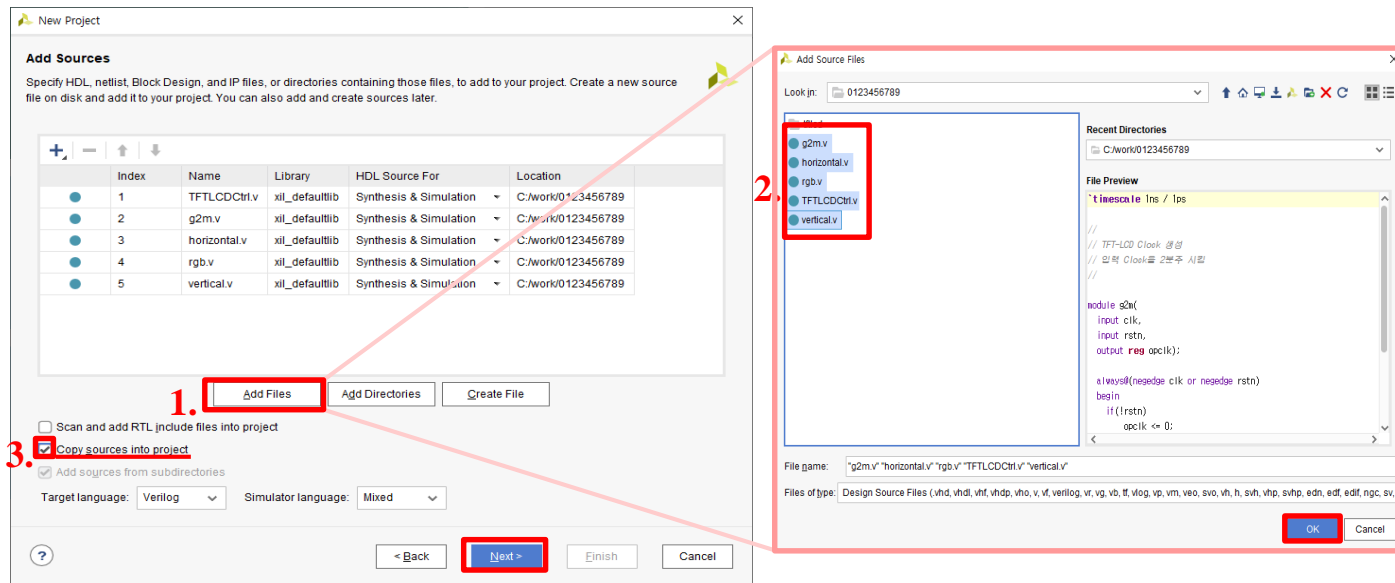
Project Creation (5/8)

17

□ Source file 추가

❖ HDL source file 생성 또는 추가

- 1. Add Files 클릭
- 2. 앞서 download한 verilog source file(**g2m.v**, **horizontal.v**, **rgb.v**, **TFTLCDCtrl.v**, **vertical.v**) 선택 후 **OK** 클릭 **Next >**
- 3. Copy sources into project check 후 Next 클릭



Project Creation (6/8)

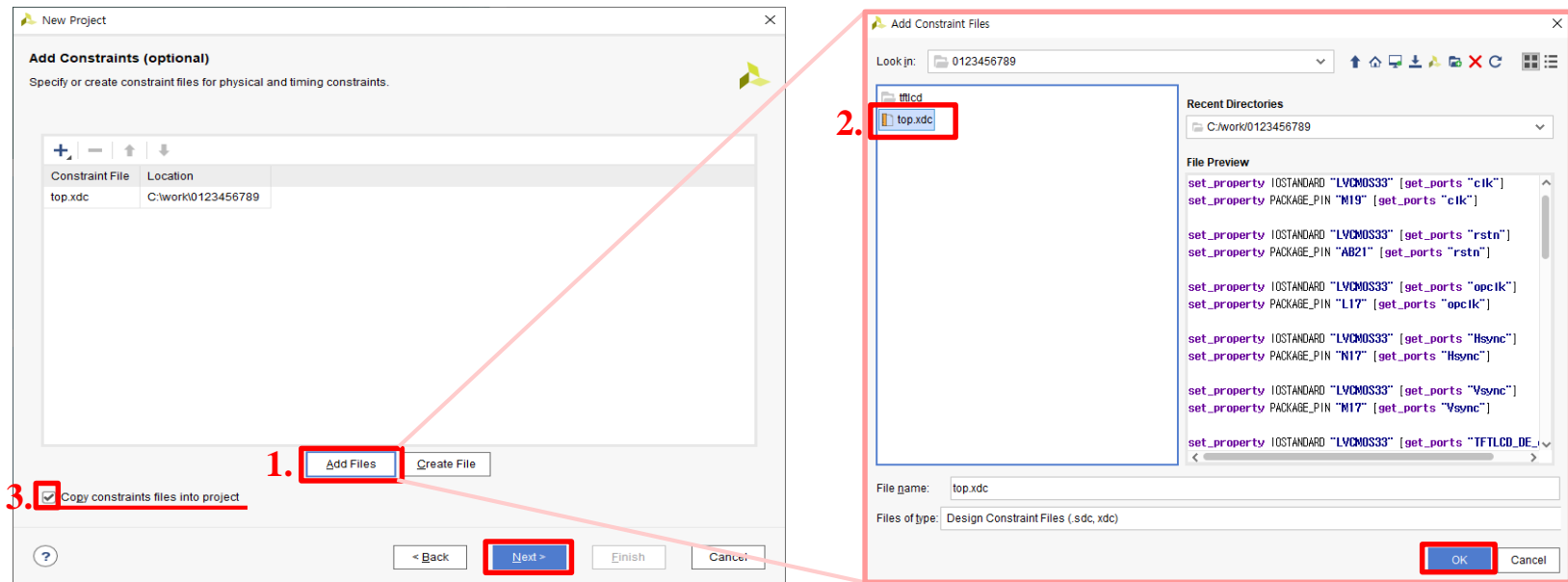
18

□ Constraint file 추가

❖ *Constraint file 생성 또는 추가*

(constraint file 은 pin assign, timing 에 관한 정보를 가짐)

- 1. Add Files 클릭
- 2. 앞서 download한 constraint file(**top.xdc**) 선택 후 **OK** 클릭
- 3. Copy constraints files into project check 후 **Next >** 클릭



Project Creation (7/8)

19

□ Target device 설정

❖ 사용하는 *FPGA / ZYNQ chip(part)* 또는 *board*를 선택 (*board*의 경우 주변장치에 관한 *preset* 제공)

➤ Filter → Search에서 “xc7z020clg484-1”를 입력하여 해당 chip 선택

New Project

Default Part

Choose a default Xilinx part or board for your project.

Parts | Boards

Reset All Filters

Category: All Package: All Temperature: All

Family: All Speed: All Static power: All

Search: xc7z020clg484-1 (1 match)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	D:
xc7z020clg484-1	484	200	53200	106400	140	0	2

< Back Next > Finish Cancel

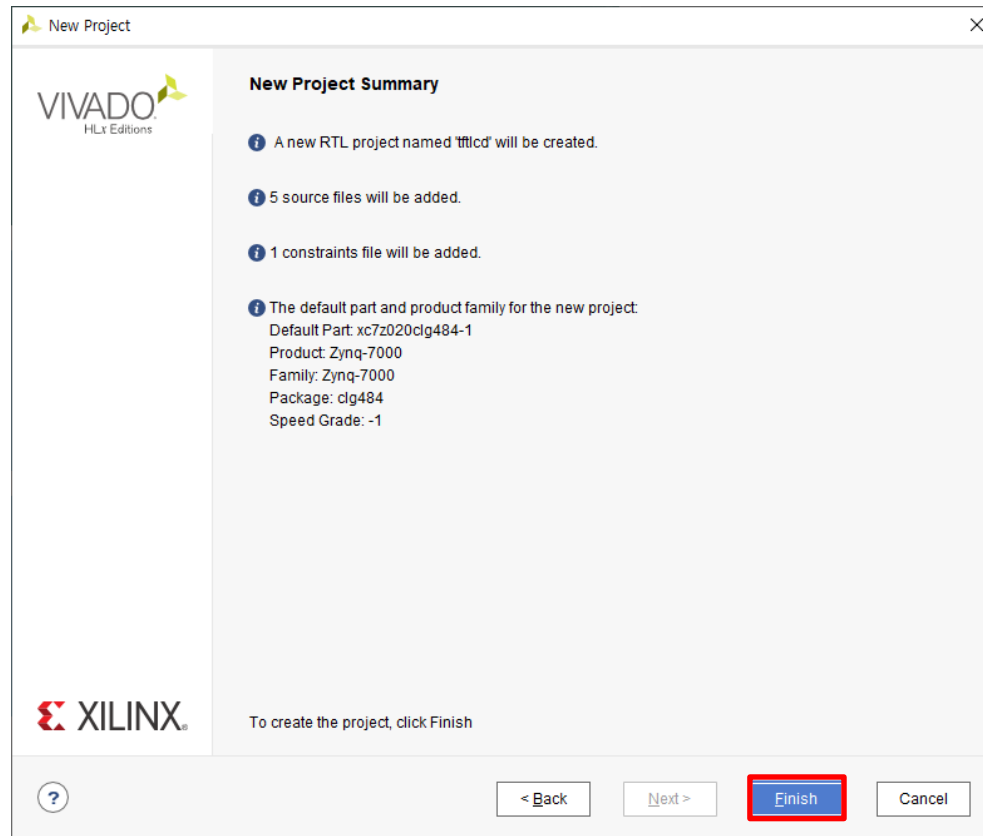
Project Creation (8/8)

20

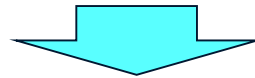
□ Project 생성 마무리

❖ 앞서 설정한 옵션에 관한 *Summary*를 확인

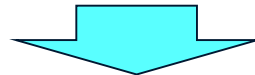
➤  버튼을 click하여 project 생성 마침



Project Creation



HDL Coding

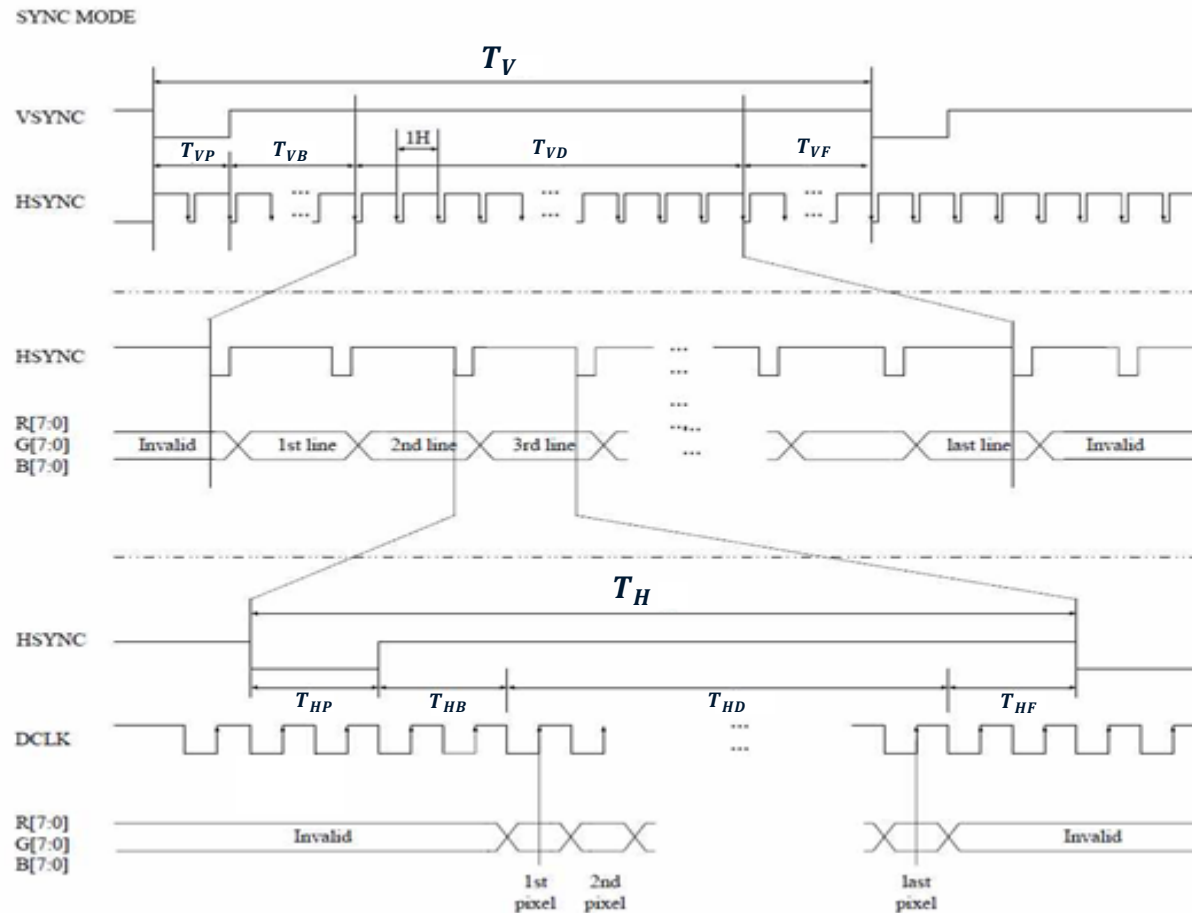


FPGA Programming

TFT-LCD Controller Design

22

□ TFT-LCD timing characteristics



TFT-LCD Controller Design

23

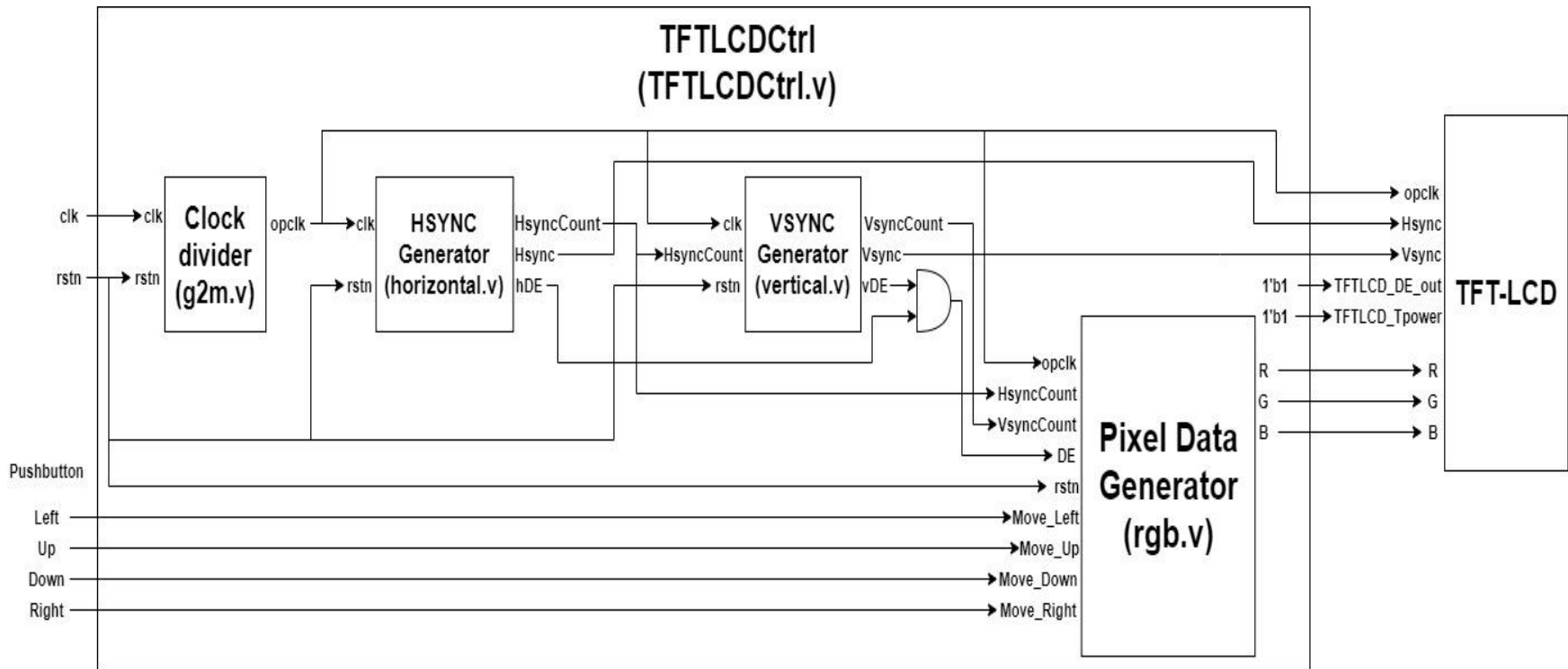
□ TFT-LCD timing characteristics

Parameter	Symbol	Spec.			Unit
		Min.	Typ.	Max.	
Horizontal cycle	T_H	525	525	605	DCLK
Horizontal display period	T_{HD}		480		DCLK
Horizontal pulse width	T_{HP}	2	41	41	DCLK
Horizontal back porch	T_{HB}	2	2	41	DCLK
Horizontal front porch	T_{HF}	2	2	82	DCLK
Vertical cycle	T_V	285	286	399	HSYNC
Vertical display period	T_{VD}		272		HSYNC
Vertical pulse width	T_{VP}	1	10	11	HSYNC
Vertical back porch	T_{VB}	1	2	11	HSYNC
Vertical front porch	T_{VF}	1	2	227	HSYNC

TFT-LCD Controller Design

24

□ TFT-LCD controller Block diagram



□ TFT-LCD Controller Block diagram

❖ Clock divider

- *TFT-LCD 구동에 사용되는 12.5MHz의 클럭을 생성하는 클럭 분주 모듈*

❖ HSYNC Generator

- *Horizontal Synchronize에 필요한 신호들을 생성하는 모듈*

❖ VSYNC Generator

- *Vertical Synchronize에 필요한 신호들을 생성하는 모듈*

❖ RGB Generator

- *HSYNC, VSYNC Generator에서 만들어진 신호를 바탕으로 TFT-LCD에 표시할 Pixel data를 생성하는 모듈*

❖ TFTLCDCtrl

- *위의 모듈들을 연결시켜주고, TFT-LCD 구동에 필요한 값들을 미리 셋팅 해주고 각 모듈에서 생성된 신호들을 최종적으로 TFT-LCD에 출력해주는 모듈*

■ 기존에 제공한 RGB Generator 코드는 정적인 화면을 출력하므로, 동적인 화면을 출력하기 위해서는 수정이 필요함.

■ 나머지 모듈의 경우 기존에 제공된 자료와 동작이 동일하므로 수정이 필요하지 않음

□Pixel Data Generator

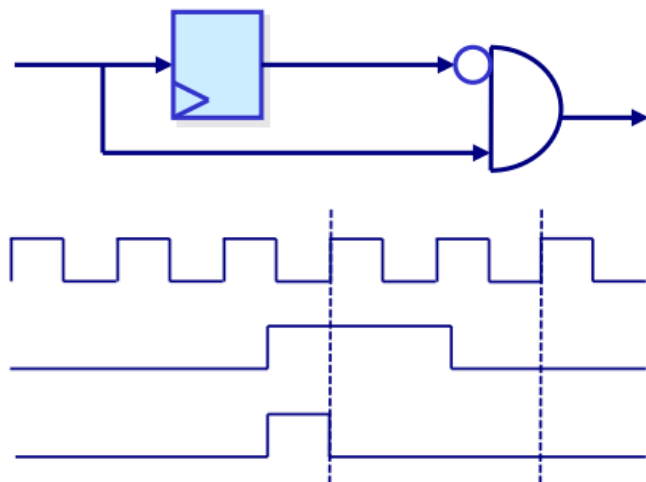
❖ *HSYNC, VSYNC Generator*에서 만들어진 신호를 바탕으로 *TFT-LCD*에 표시할 *Pixel data*를 생성하는 모듈

- 크게 3가지 부분으로 나눌 수 있음
- 버튼을 한 번 누를 때, 버튼을 누르고 있는 시간에 상관없이 한번의 펄스만 발생시켜주는 **Level to Pulse Converter** 회로
- 만들어진 펄스를 바탕으로 카운트를 하는 **Counter** 회로
- 카운트 된 위치 값들을 바탕으로 해당되는 위치에 표시될 **RGB data**를 만들어내는 회로

❑ Level-to-Pulse Converter

❖ 버튼을 누르는 동작은 사람의 입장에서 보면 잠깐이지만 12.5MHz로 동작하는 회로 관점에서는 매우 오랫동안 값이 유지되는 것이기 때문에 이를 펄스 신호로 바꾸어 처리하면 Pushbutton을 사용할 때 편리함

▶ 아래와 같이 단순한 회로를 설계하여 위와 같은 동작을 하는 회로를 구현할 수 있음



```

37 assign Left_oneshot      =  (~Left_sig)&(~Move_Left);
38
39 always@(negedge clk or negedge rstn)
40 begin
41     if(!rstn)
42         Left_sig    <=  1'b0;
43     else begin
44         Left_sig    <=  ~Move_Left;
45     end
46 end

```

□ Counter

- ❖ *Pushbutton 에 의해 pulse 가 만들어질 때 마다 8 pixels 씩 움직이도록 값을 카운트하는 회로 설계 (Left, Up, Down, Right 4 방향의 움직임을 카운트해야 하므로 아래와 같은 카운터가 각 방향별로 필요)*

```
78 ⊖ always@(negedge clk or negedge rstn)
79 ⊖ begin
80 ⊖     if(!rstn)
81 ⊖         Left_cnt    <=  10'd0;
82 ⊖     else begin
83 ⊖         if(Left_oneshot)
84 ⊖             Left_cnt    <=  Left_cnt    +   10'd8;
85 ⊖         else
86 ⊖             Left_cnt    <=  Left_cnt;
87 ⊖     end
88 ⊖ end
89 ⊖
```

□RGB Generator

❖ *Counter* 에서 카운팅 된 값에 따라 데이터를 표시할 위치를 계산하여 해당되는 영역에만 R,G,B 데이터를 전송 시켜 영상을 출력시키는 회로이며, 오직 조합회로로만 설계했기 때문에 의도하지 않은 래치가 생성되는 것을 방지하기 위해 모든 if문에 대응되는 else 문을 추가하여 설계해야 함

```
127 always@(DE,HsyncCount,VsyncCount,rstn,Right_cnt,Left_cnt,Up_cnt,Down_cnt)
128 begin
129     if(!rstn)
130     begin
131         R = 0;
132         G = 0;
133         B = 0;
134     end
135
136     else if(DE == 1)
137     begin
138         if((HsyncCount >= (10'd43 + (Right_cnt) - (Left_cnt))) && (HsyncCount <= (10'd103 + (Right_cnt) - (Left_cnt))))
139         begin
140             if ((VsyncCount > 10'd11 - (Up_cnt) + (Down_cnt)) && (VsyncCount <= 10'd51 - (Up_cnt) + (Down_cnt)))
141             begin
142                 R = 5'b1111;
143                 G = 6'b000000;
144                 B = 5'b1111;
145             end
146
147             else
148             begin
149                 R = 0;
150                 G = 0;
151                 B = 0;
152             end
153         end
154     end
```

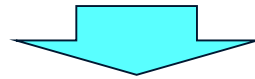
```
154     else
155     begin
156         R = 0;
157         G = 0;
158         B = 0;
159     end
160 end
161
162 else
163 begin
164     R = 0;
165     G = 0;
166     B = 0;
167 end
168 end
```

□TFTLCDCtrl

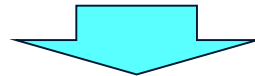
❖ *TFT-LCD 제어에 필요한 신호를 생성하는 모듈들을 서로 연결 시켜 주고, 그 외 필요한 신호들을 미리 셋팅 해주고 각 모듈에서 만들어진 신호를 최종적으로 TFT-LCD 에 출력해주는 모듈*

- TFT-LCD의 Backlight 전원 신호인 TFTLCD_Tpower 신호를 High로 셋팅
- TFT-LCD의 Data enable 신호인 TFTLCD_DE_out 신호를 High로 셋팅
- Block diagram 을 참고하여 관련된 모듈들 끼리 연결

Project Creation



HDL Coding



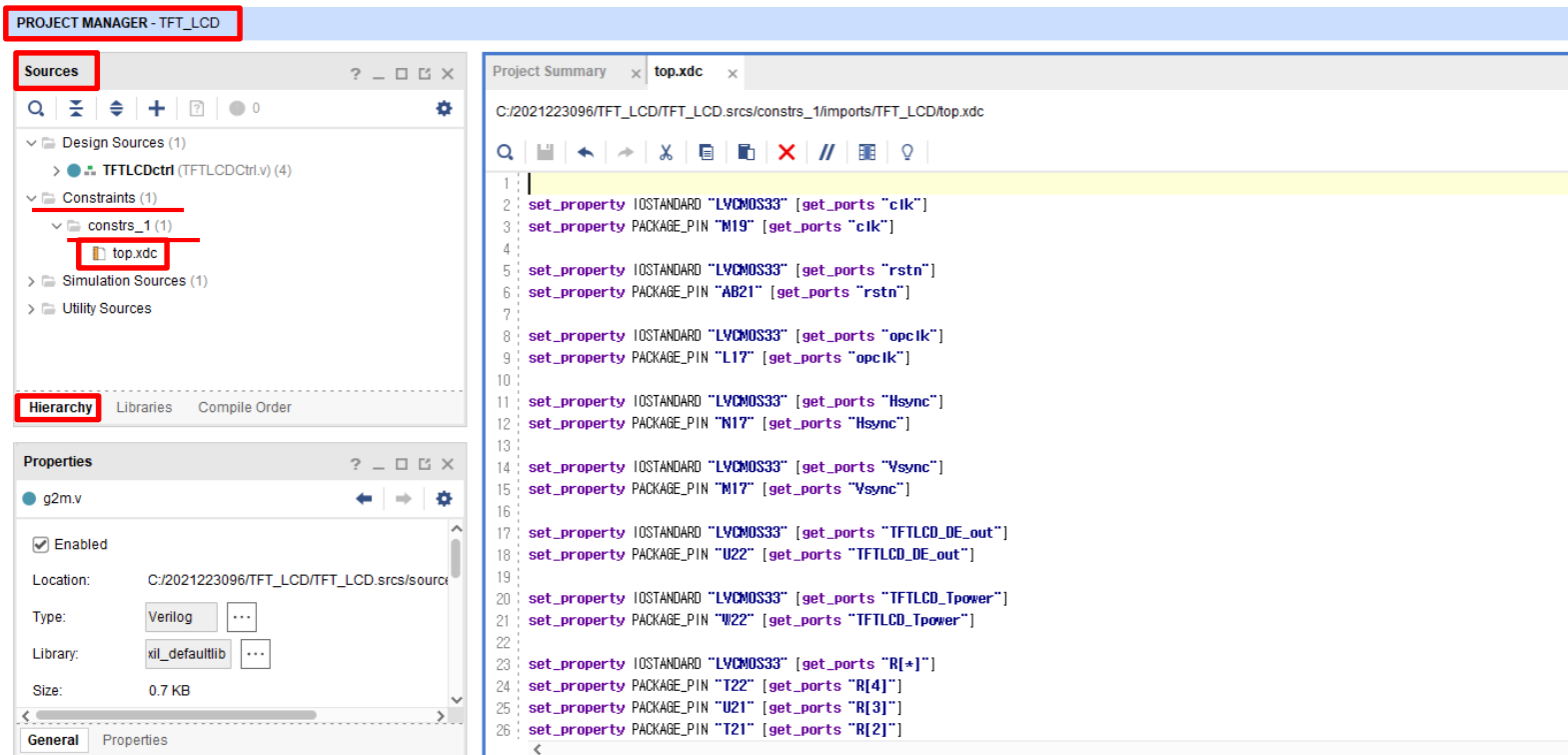
FPGA Programming

Pin Assign (1/2)

32

□ 앞서 추가한 constraint file open

❖ Project Manager → Sources → Hierarchy → Constraints →
constrs_1 → top.xdc (target) double click하여 constraint file open



Pin Assign (2/2)

33

□ Constraint file을 통한 pin assign

❖ 설계한 TFT-LCD Controller(TFTLCDctrl.v)의 I/O ports 를 외부로 연결해야 함

❖ ZYNQ PL part 의 pins 에 top module 의 I/O ports 할당

➤ Constraint file(top.xdc)에 pin assign 정보 확인

```
1 |
2 set_property IOSTANDARD "LVCMOS33" [get_ports "clk"]
3 set_property PACKAGE_PIN "M19" [get_ports "clk"]
4
5 set_property IOSTANDARD "LVCMOS33" [get_ports "rstn"]
6 set_property PACKAGE_PIN "AB21" [get_ports "rstn"]
7
8 set_property IOSTANDARD "LVCMOS33" [get_ports "opclk"]
9 set_property PACKAGE_PIN "L17" [get_ports "opclk"]
10
11 set_property IOSTANDARD "LVCMOS33" [get_ports "Hsync"]
12 set_property PACKAGE_PIN "N17" [get_ports "Hsync"]
13
14 set_property IOSTANDARD "LVCMOS33" [get_ports "Vsync"]
15 set_property PACKAGE_PIN "M17" [get_ports "Vsync"]
16
17 set_property IOSTANDARD "LVCMOS33" [get_ports "TFTLCD_DE_out"]
18 set_property PACKAGE_PIN "U22" [get_ports "TFTLCD_DE_out"]
19
20 set_property IOSTANDARD "LVCMOS33" [get_ports "TFTLCD_Tpower"]
21 set_property PACKAGE_PIN "W22" [get_ports "TFTLCD_Tpower"]
```

```
22
23 set_property IOSTANDARD "LVCMOS33" [get_ports "R[*]"]
24 set_property PACKAGE_PIN "T22" [get_ports "R[4]"]
25 set_property PACKAGE_PIN "U21" [get_ports "R[3]"]
26 set_property PACKAGE_PIN "T21" [get_ports "R[2]"]
27 set_property PACKAGE_PIN "K20" [get_ports "R[1]"]
28 set_property PACKAGE_PIN "K19" [get_ports "R[0]"]
29
30 set_property IOSTANDARD "LVCMOS33" [get_ports "G[*]"]
31 set_property PACKAGE_PIN "L22" [get_ports "G[5]"]
32 set_property PACKAGE_PIN "L21" [get_ports "G[4]"]
33 set_property PACKAGE_PIN "K21" [get_ports "G[3]"]
34 set_property PACKAGE_PIN "J20" [get_ports "G[2]"]
35 set_property PACKAGE_PIN "J22" [get_ports "G[1]"]
36 set_property PACKAGE_PIN "J21" [get_ports "G[0]"]
37
38 set_property IOSTANDARD "LVCMOS33" [get_ports "B[*]"]
39 set_property PACKAGE_PIN "K18" [get_ports "B[4]"]
40 set_property PACKAGE_PIN "J18" [get_ports "B[3]"]
41 set_property PACKAGE_PIN "M16" [get_ports "B[2]"]
42 set_property PACKAGE_PIN "M15" [get_ports "B[1]"]
43 set_property PACKAGE_PIN "N18" [get_ports "B[0]"]
```

```
45 set_property IOSTANDARD "LVCMOS33" [get_ports "Right"]
46 set_property PACKAGE_PIN "Y18" [get_ports "Right"]
47
48 set_property IOSTANDARD "LVCMOS33" [get_ports "Down"]
49 set_property PACKAGE_PIN "AA18" [get_ports "Down"]
50
51 set_property IOSTANDARD "LVCMOS33" [get_ports "Up"]
52 set_property PACKAGE_PIN "Y19" [get_ports "Up"]
53
54 set_property IOSTANDARD "LVCMOS33" [get_ports "Left"]
55 set_property PACKAGE_PIN "AA19" [get_ports "Left"]
56 ;
```

Pushbutton 사용을
위한 I/O ports 할당

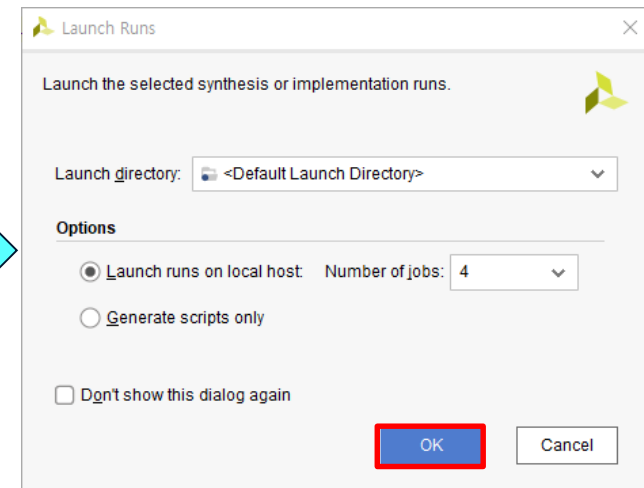
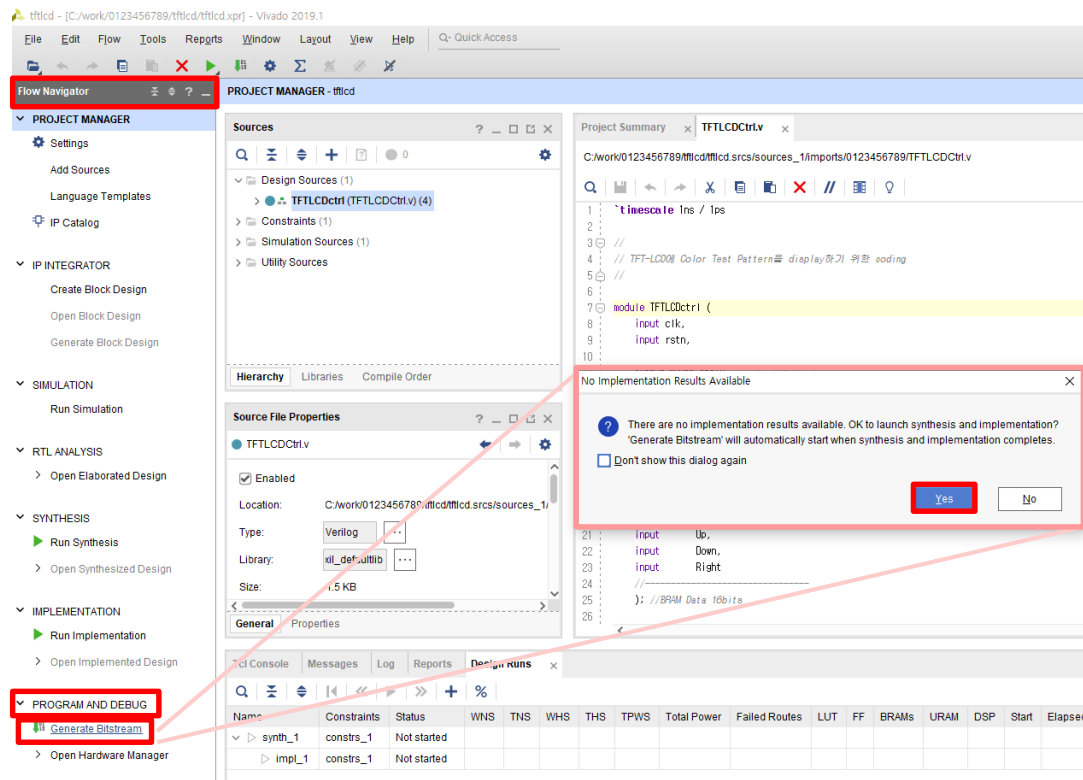
Generate Bitstream (1/2)

34

□ ZYNQ PL programming을 위한 bitstream file 생성

❖ Flow Navigator → Program and Debug → Generate Bitstream 클릭

❖ No Implementation Results Available window → **Yes** 클릭



Generate Bitstream (2/2)

35

□ Generate Bitstream 과정

❖ *Vivado 우측 상단에 status bar로 진행 과정 표시*

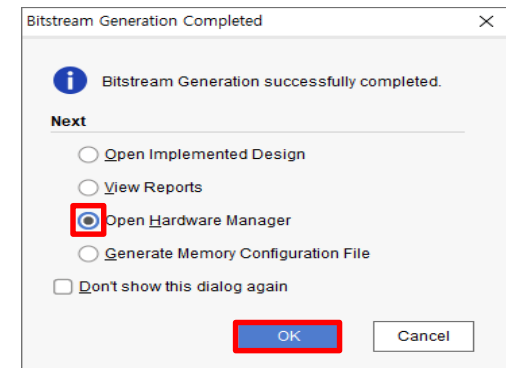
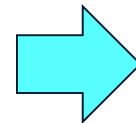
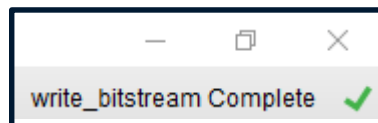
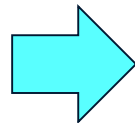
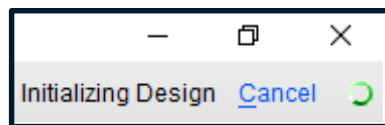
➤ 완료 후 pop-up window에서 Open Hardware Manager 체크 후 **OK** 클릭

❖ *Synthesis, implementation(place and route), generate bitstream 순으로 진행*

➤ Synthesis : HDL을 gate level의 netlist로 translate

➤ Implementation(PnR) : Netlist를 FPGA fabric에 place하고 route

➤ Generate bitstream : 최종 결과를 FPGA fabric에 programming 할 수 있는 bitstream 형태로 저장



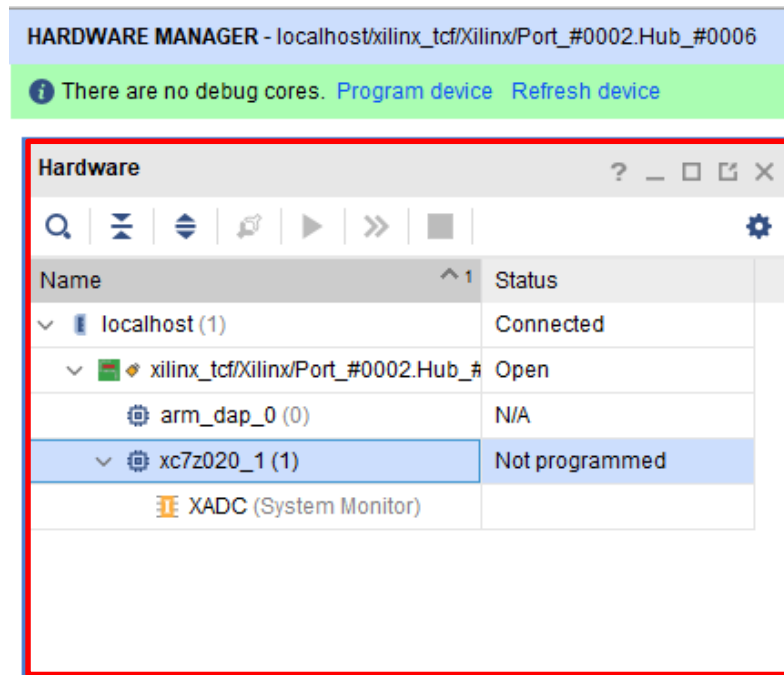
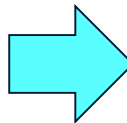
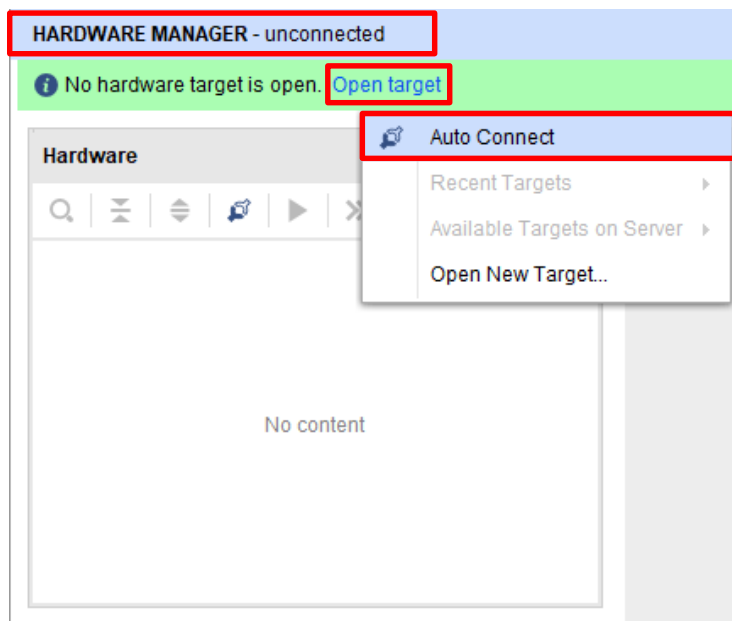
Bitstream Programming (1/3)

36

□ Open target

❖ *Open target 전 RPS-Z7020-TK board 전원 ON 필요*

❖ Hardware Manager window → Open target 클릭 → Auto Connect
클릭

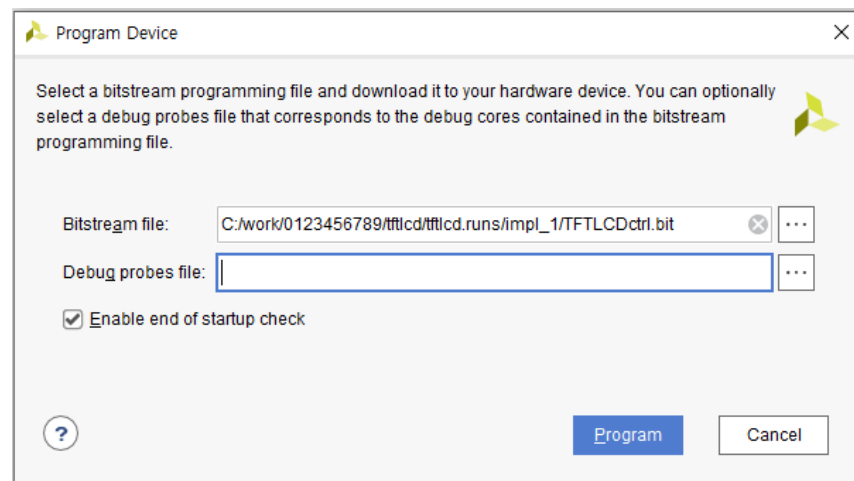
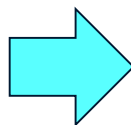
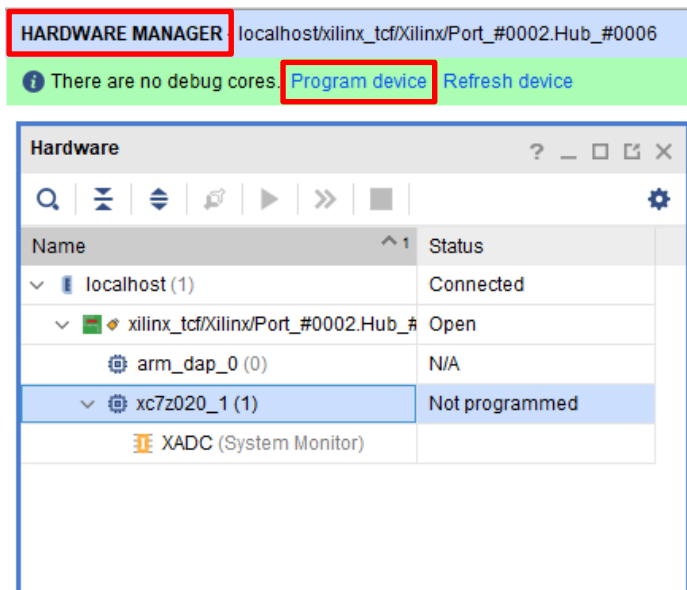


Bitstream Programming (2/3)

37

□ Program device

- ❖ *Bitstream file ≡ ZYNQ PL 에 programming*
- ❖ Hardware Manager window → Program device 클릭
- ❖ Program Device window에서 **Program** 클릭
 - Bitstream file을 자동으로 불러옴

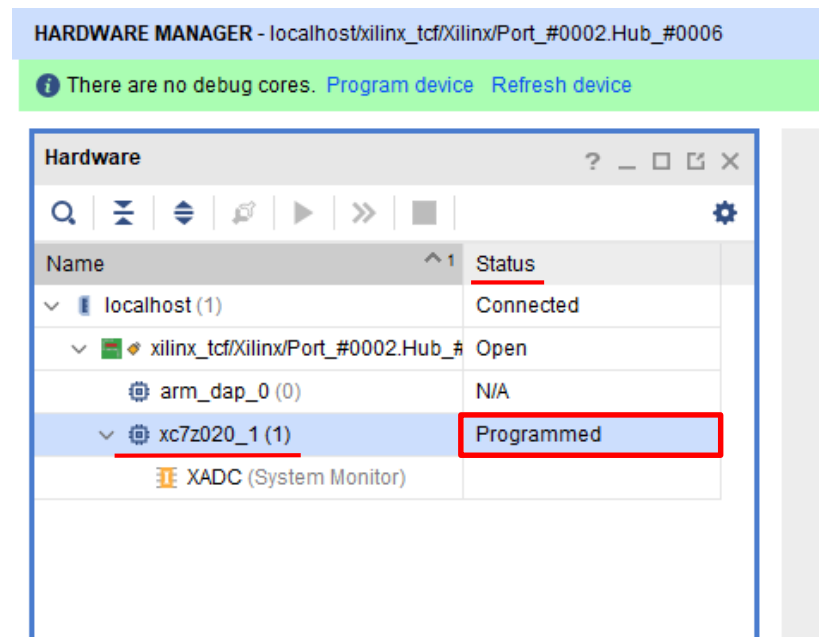
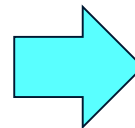


Bitstream Programming (3/3)

38

□ Programming 완료 확인

❖ *Bitstream file programming 완료 후 ZYNQ PL part(xc7z020_1)의 Status가 Programmed로 변경*



□ RPS-Z7020-TK board 확인

- ❖ Pushbutton들을 눌러보면서
TFT-LCD에 표시되는 결과 확인
- ❖ 각 스위치에 해당하는 동작은
다음과 같음
- ❖ S4 = Left,
S3 = Up,
S2 = Down,
S1 = Right

