

Vivado Bitstream 파일 생성 방법

Digital System Design and Experiment

Made by MMS Students

Graduate School of Convergence Science and Technology

Seoul National University



Mobile Multimedia Systems Group

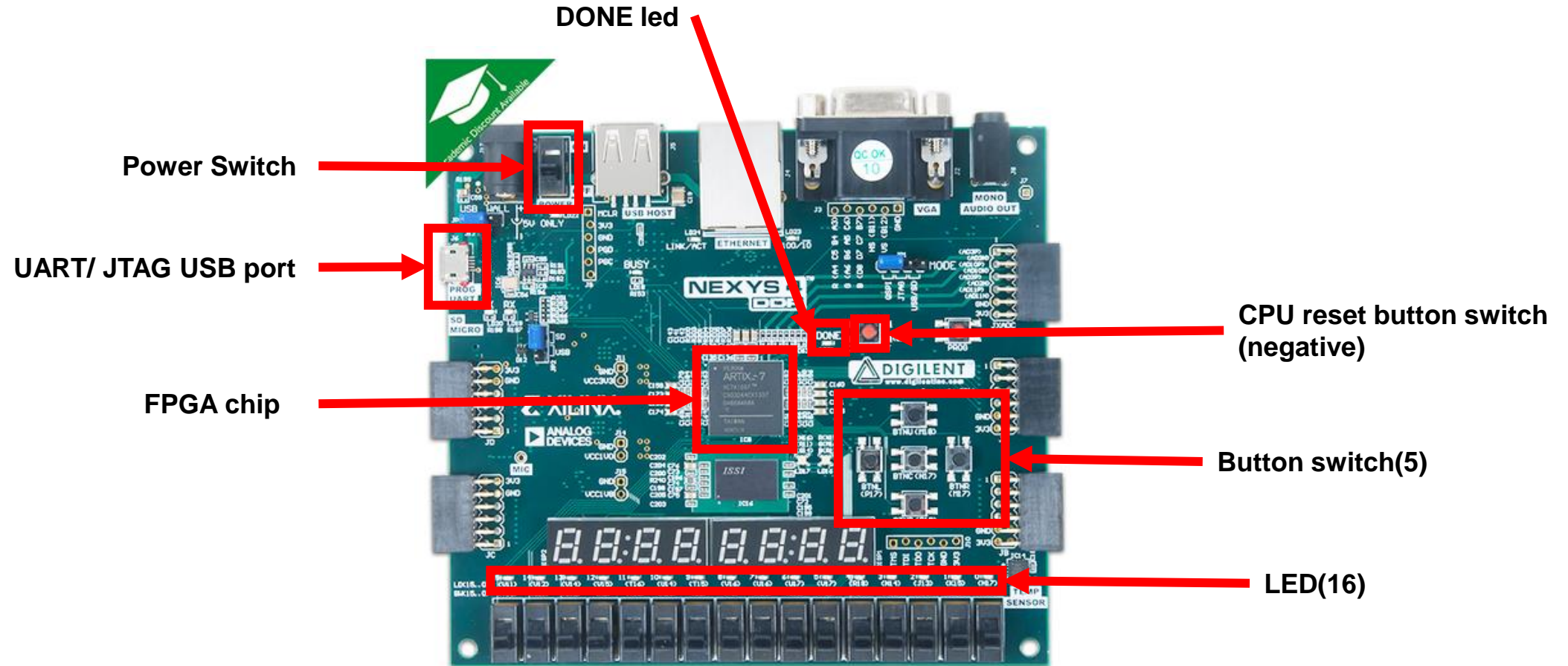
Nexys 4 DDR 보드 개요

- Artix7 계열의 Xilinx FPGA 교육용 보드임
- 15,850 logic slice (1 logic slice = 4 x 6-input LUT + 8 x Flip Flop)
- 4,860Kbit BRAM
- 128MiB DDR2 RAM
- No power cable
 - Shared UART/ JTAG USB port를 통해 전원 공급
- 자세한 내용은 아래 링크 참조

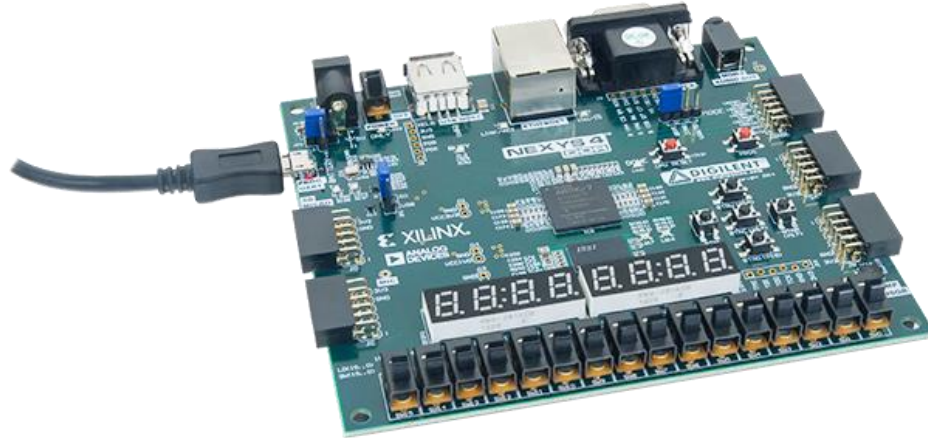
<https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/reference-manual>



Nexys 4 DDR 보드 개요



Nexys 4 DDR 보드 개요



Micro-usb pin과 FPGA 보드를 연결할 때 주의해주세요!

| Bitstream 파일 만드는 절차

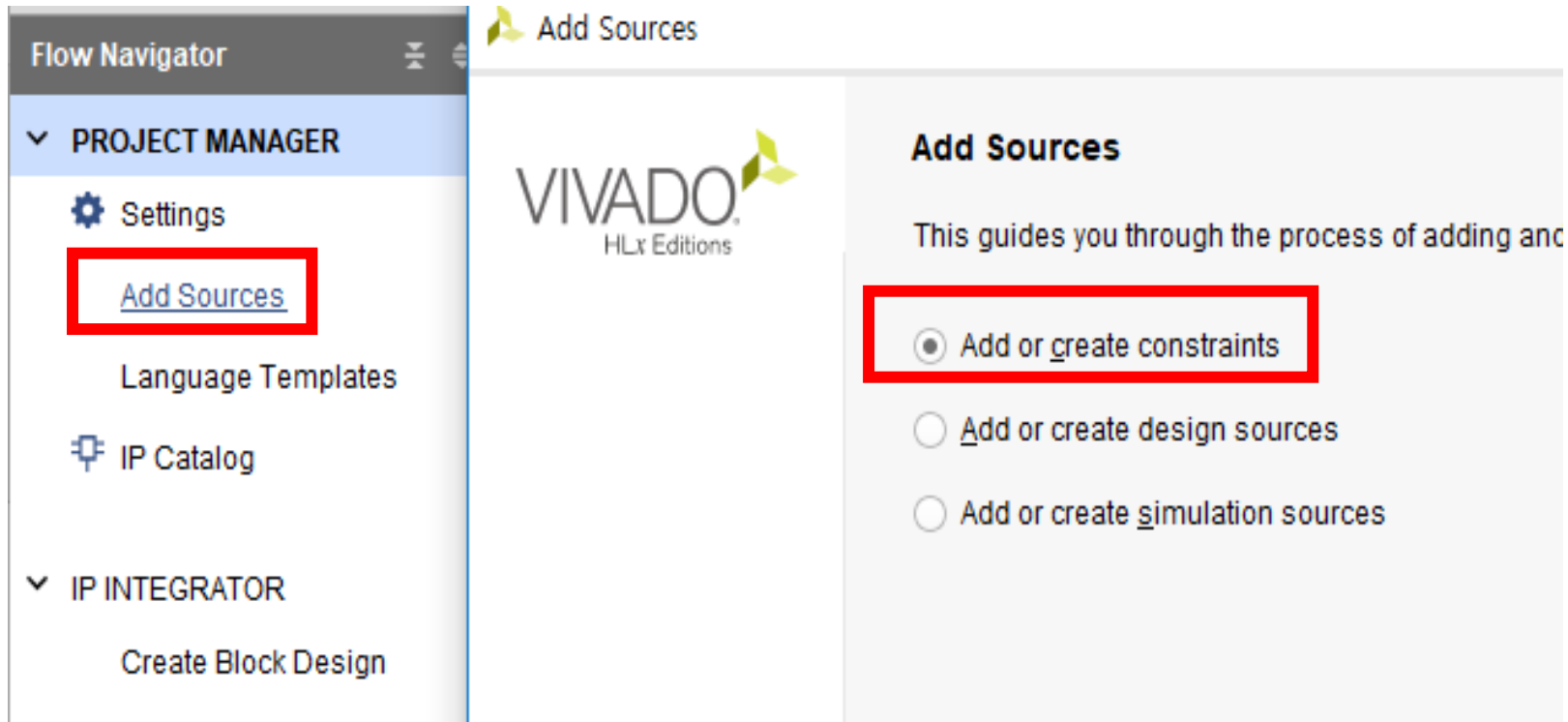
- Step 1: RTL 설계 (Verilog/VHDL)
- Step 2: Simulation
- Step 3: Synthesis
- Step 4: Implementation
 - Constraint file 추가
- Step 5: Generate Bitstream

| Constraint File 추가

- Constraint File 이란?
 - HDL로 설계한 회로를 작동시키기 위하여, 실제 보드에서 사용할 물리적인 Pin에 대한 정보 및 Clock Frequency 등을 Vivado에게 알려주기 위한 File
 - Code에서 사용하는 모듈 혹은 Pin을 실제 FPGA 보드에 mapping하는 것
- 자세한 내용은 아래 링크를 통해 확인 가능
 - <https://reference.digilentinc.com/learn/software/tutorials/vivado-xdc-file>

Constraint File 추가

- Project Manager > Add Sources > Add or create constraints > "Nexys4DDR_Master.xdc" > Finish
 - 꼭 **"Copy constraints files into project"**를 check 하길



Constraint File 수정

- 보드 제작사에서 제공하는 “____.xdc” 파일은 주석처리 되어 있음.
- 수정 방법
 - 필요한 pin을 찾아 주석 제거
 - 해당 Pin과 상호작용할 Port로 변경
 - **get_ports { **** 바꿀 곳 **** }**

LEDs

```
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { in_data[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { in_data[1] }]; #IO_L24P_T3_R31_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { in_data[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14 IOSTANDARD LVCMOS33 } [get_ports { in_data[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 } [get_ports { in_data[4] }]; #IO_L12P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMOS33 } [get_ports { in_data[5] }]; #IO_L18N_T2_A14_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 } [get_ports { in_data[6] }]; #IO_L17P_T2_A14_D08_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 } [get_ports { in_data[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
#set_property -dict { PACKAGE_PIN V16 IOSTANDARD LVCMOS33 } [get_ports { LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
#set_property -dict { PACKAGE_PIN T15 IOSTANDARD LVCMOS33 } [get_ports { LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
#set_property -dict { PACKAGE_PIN U14 IOSTANDARD LVCMOS33 } [get_ports { LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
#set_property -dict { PACKAGE_PIN T16 IOSTANDARD LVCMOS33 } [get_ports { LED[11] }]; #IO_L15N_T2_DQS_DOUT_CS0_B_14 Sch=led[11]
#set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMOS33 } [get_ports { LED[12] }]; #IO_L16P_T2_CS1_B_14 Sch=led[12]
#set_property -dict { PACKAGE_PIN V14 IOSTANDARD LVCMOS33 } [get_ports { LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
#set_property -dict { PACKAGE_PIN V12 IOSTANDARD LVCMOS33 } [get_ports { LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
#set_property -dict { PACKAGE_PIN V11 IOSTANDARD LVCMOS33 } [get_ports { LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]
```

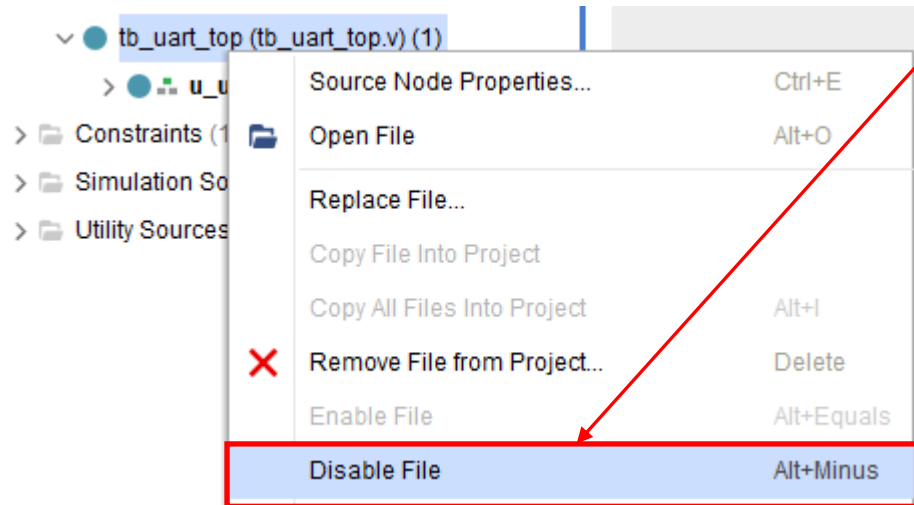
바꾼 port name

Clock signal

```
set_property -dict { PACKAGE_PIN E3 IOSTANDARD LVCMOS33 } [get_ports { clk }]; #IO_L12P_T1_MRCC_35
Sch=clk100mhz
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {clk}];
```


Bistream 파일 생성

- RTL 설계하고 testbench로 검증 후 아래와 같이 수행
 - Step 1 – tb_file disable
 - Step 2 – Run Synthesis
 - Step 3 – Run Implementation
 - Step 4 – Generate Bitstream

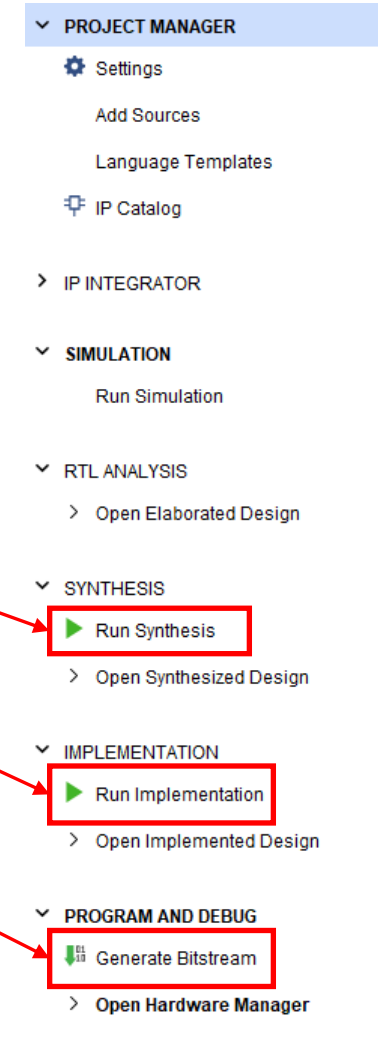


Step 1

Step 2

Step 3

Step 4



보드에 올리기

- Step 0: FPGA power on
- Step 1: Generate Bitstream 하단, Open Target > Auto Connect
- Step 2: Hardware Manage 하단, Program device 클릭
- Step 3: Program 클릭
- Step 4: DONE led 확인

(Note: Step 3에서 Bitstream File이 없을 경우, “[project name]폴더> / *.runs / impl1” 에서 “**** .bit**”파일을 찾으세요.)

