

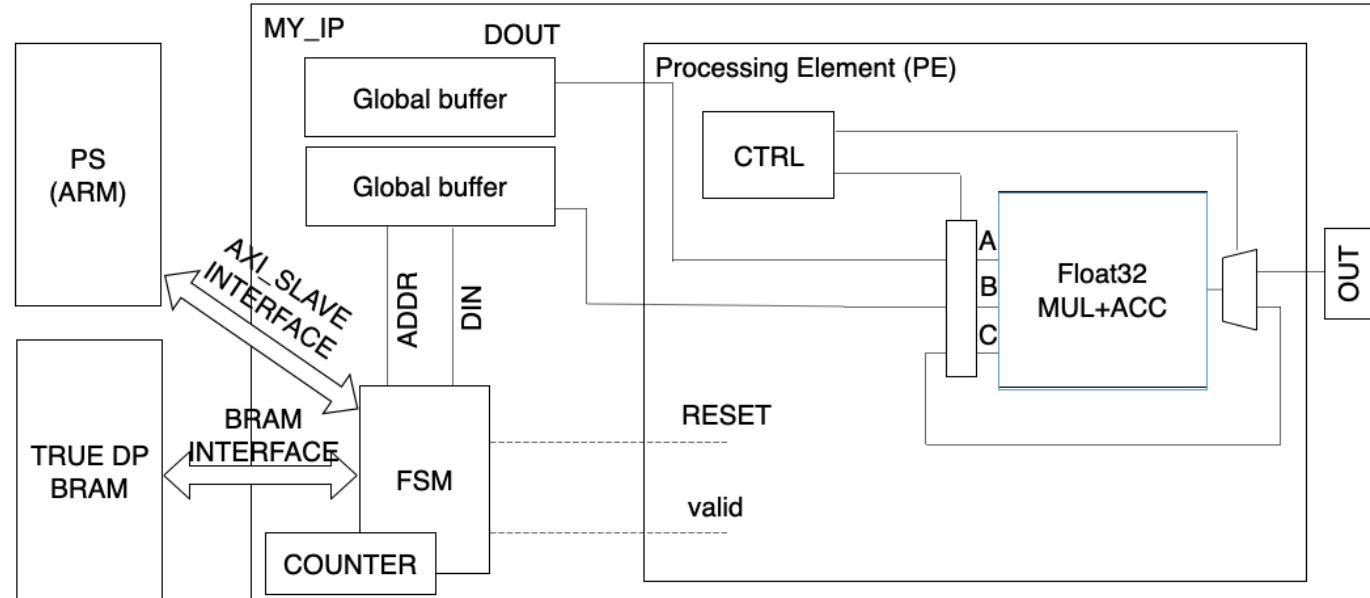
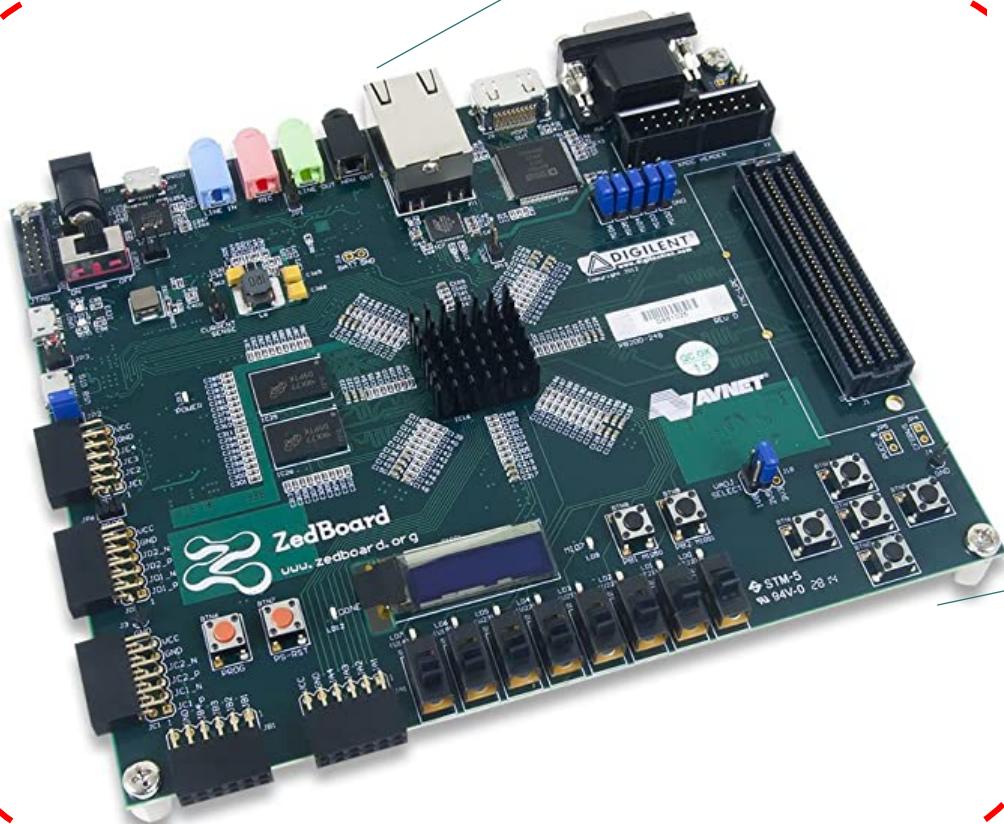
Practice 7

- How to use FPGA board

Computing Memory Architecture
Lab.

Final Project Overview: Matrix Multiplication IP

FPGA Board



Overview

■ ZED Board Tutorial

- Setup the board
- SW2LED module
 - A combinational logic that blinks [7:0] LED in response to [7:0] SWITCH

■ Practice

- Implement a simple sequential logic with external clock

ZED Board Tutorial

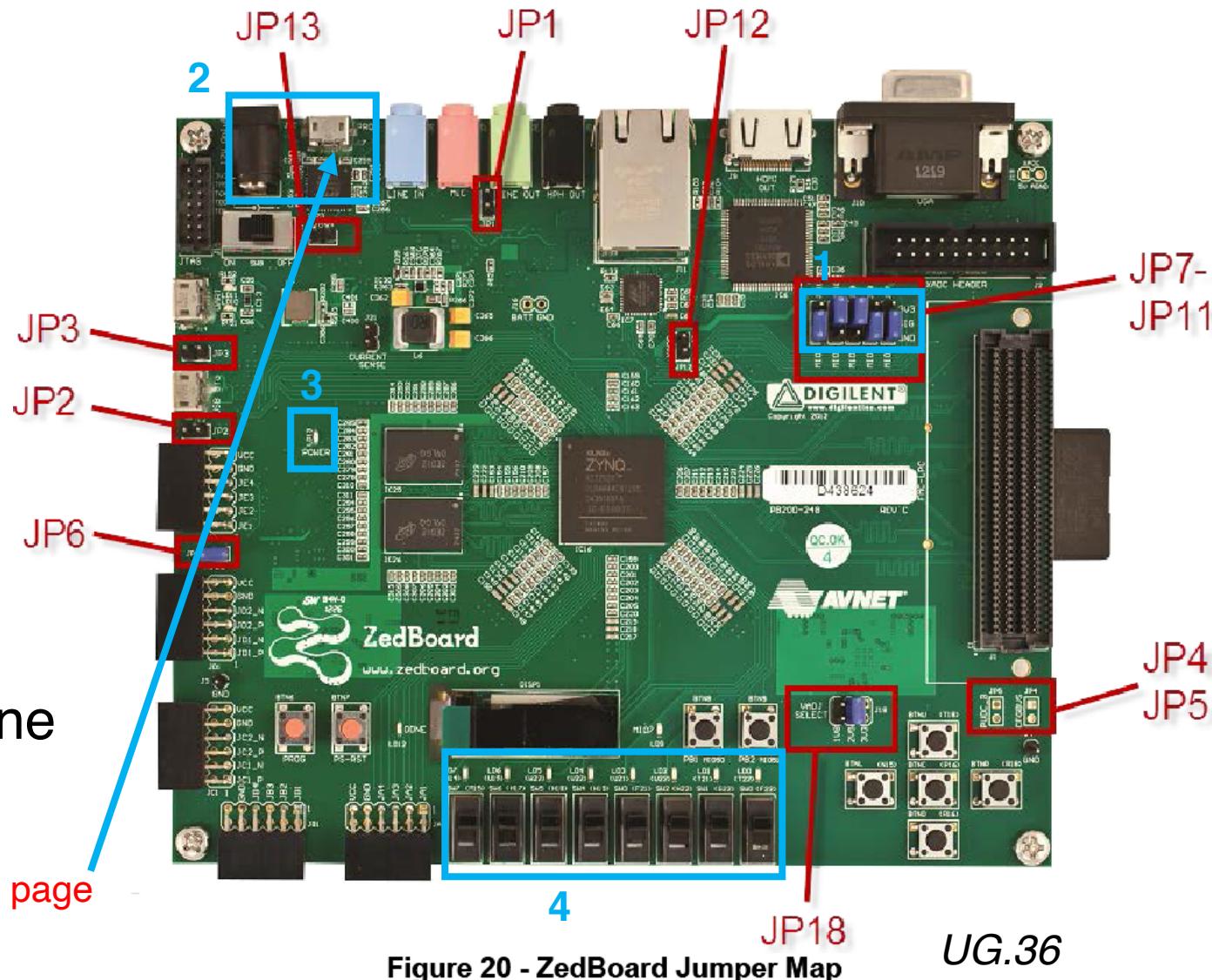
Notations

- **UG.XXX** (User Guide)
 - Page XXX of ZedBoard_HW_UG_v2_2.pdf
 - With respect to *the PDF page number* not footer
- **SM.YYY** (Schematic)
 - Page YYY of ZedBoard_RevD.2_Schematic_130516.pdf
- **DCLLL** (Design Constraint)
 - Line LLL of zedboard_master_XDC_RevC_D_v3.xdc
- **BD.PPP** (Board)
 - Part PPP of the board

Preparing the Board

- Set JTAG boot mode (1)
 - BD.JP7 ~ BD.JP11
 - 5'b01100 -> 5'b00000
- Insert USB-JTAG cable (2)
 - BD.J17
- Insert power cable (2)
 - BD.J20, BD.SW8
 - Power ON -> see BD.LD13 (3)
- SW/LED (4)
 - Check locations of today's heroine
 - BD.SW0~7, BD.LD0~7

Cautions on next page



Cautions for the Board

■ Micro 5pin slots

- Well known for breaking down.
 - Weak soldering
 - Stiff
 - Structural deficiency
- If the cable connection is stiff, press the hook with a knife tip

■ No extra board available

- No further labs (unable to do without board)

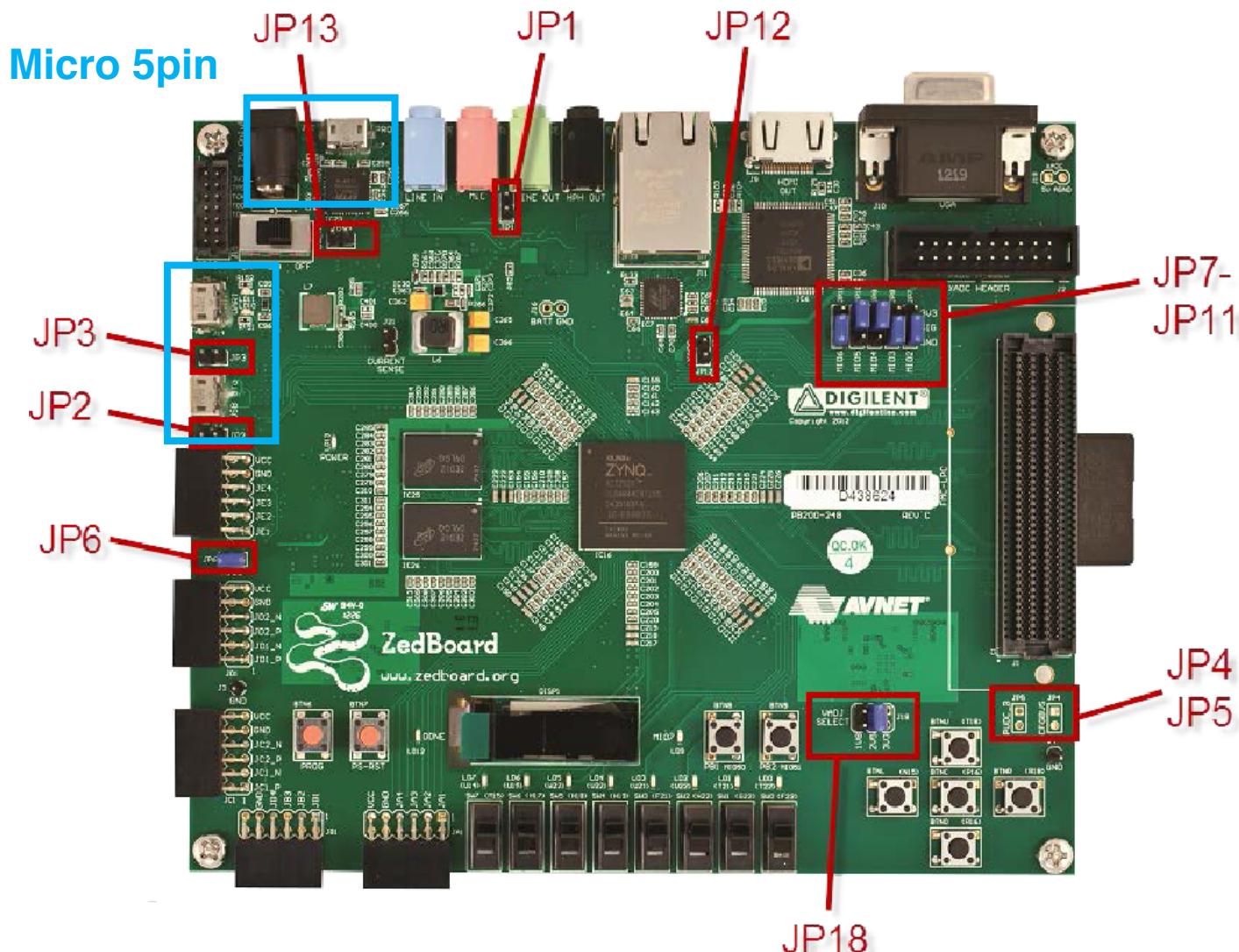
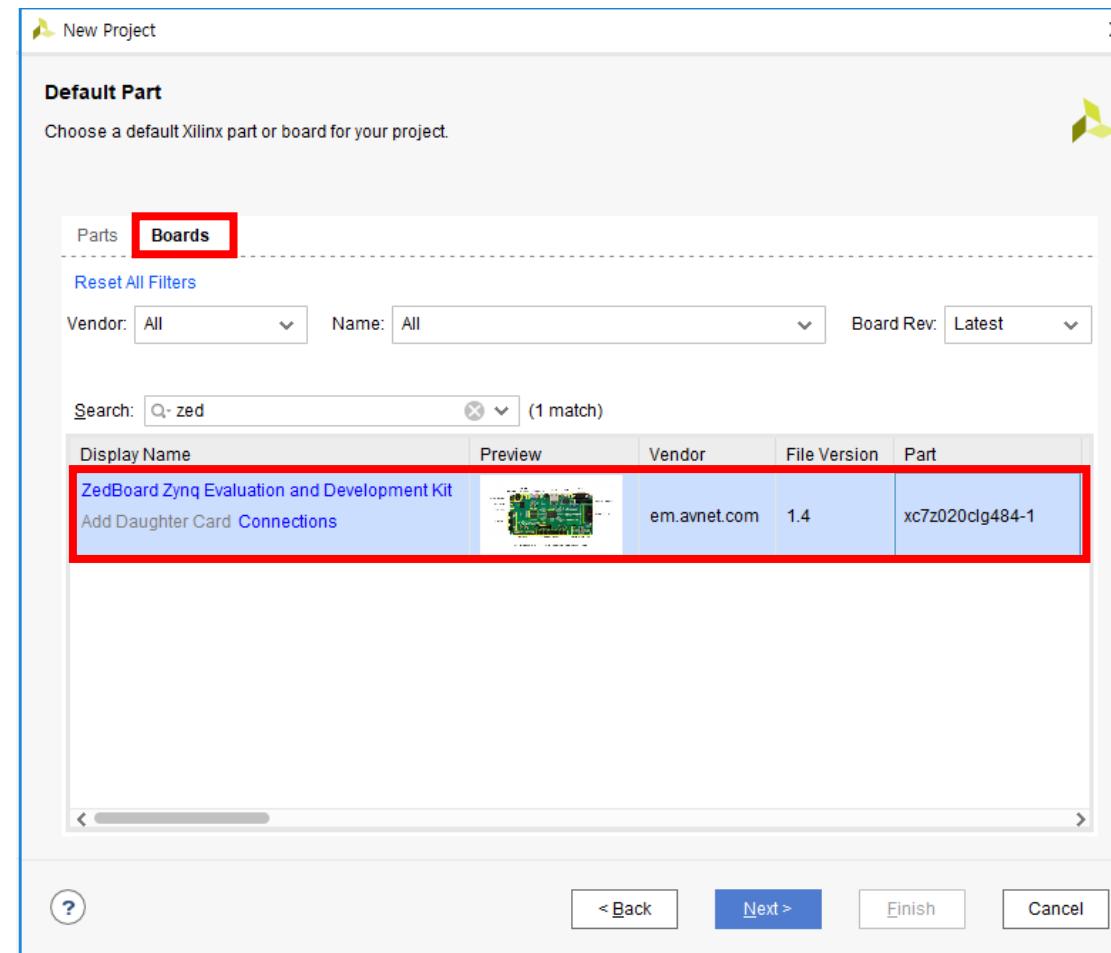


Figure 20 - ZedBoard Jumper Map

Vivado Project Creation

- Create Project - Choose part or board
 - ZedBoard



SW2LED Module

■ Edit your design

- A Verilog file (1)
 - Add or create **design sources**
 - Simple SW+LED module
 - [7:0] LD = [7:0] SW;
- An XDC (constraint) file (2)
 - Add or create **constraints**
 - 8-switches
 - PACKAGE_PIN: UG.20, DC.237, SM.9
 - IOSTANDARD: UG.5, DC.371, BD.J18
 - 8-LEDs
 - PACKAGE_PIN: UG.21, DC.175, SM.9
 - IOSTANDARD: UG.5, DC.362/367

The screenshot shows the Vivado IDE interface with the 'Project Summary' tab selected. Below it, the 'sw2led.v' file is open in the editor. The code is a simple Verilog module definition:

```
// Dependencies:  
//  
// Revision:  
// Revision 0.01 - File Created  
// Additional Comments:  
//  
module sw2led(  
    input [7:0] SW,  
    output [7:0] LD  
);  
    assign LD = SW;  
endmodule
```

The code is numbered from 14 to 31. Lines 20 through 22 are highlighted with a yellow background.

SW2LED Module

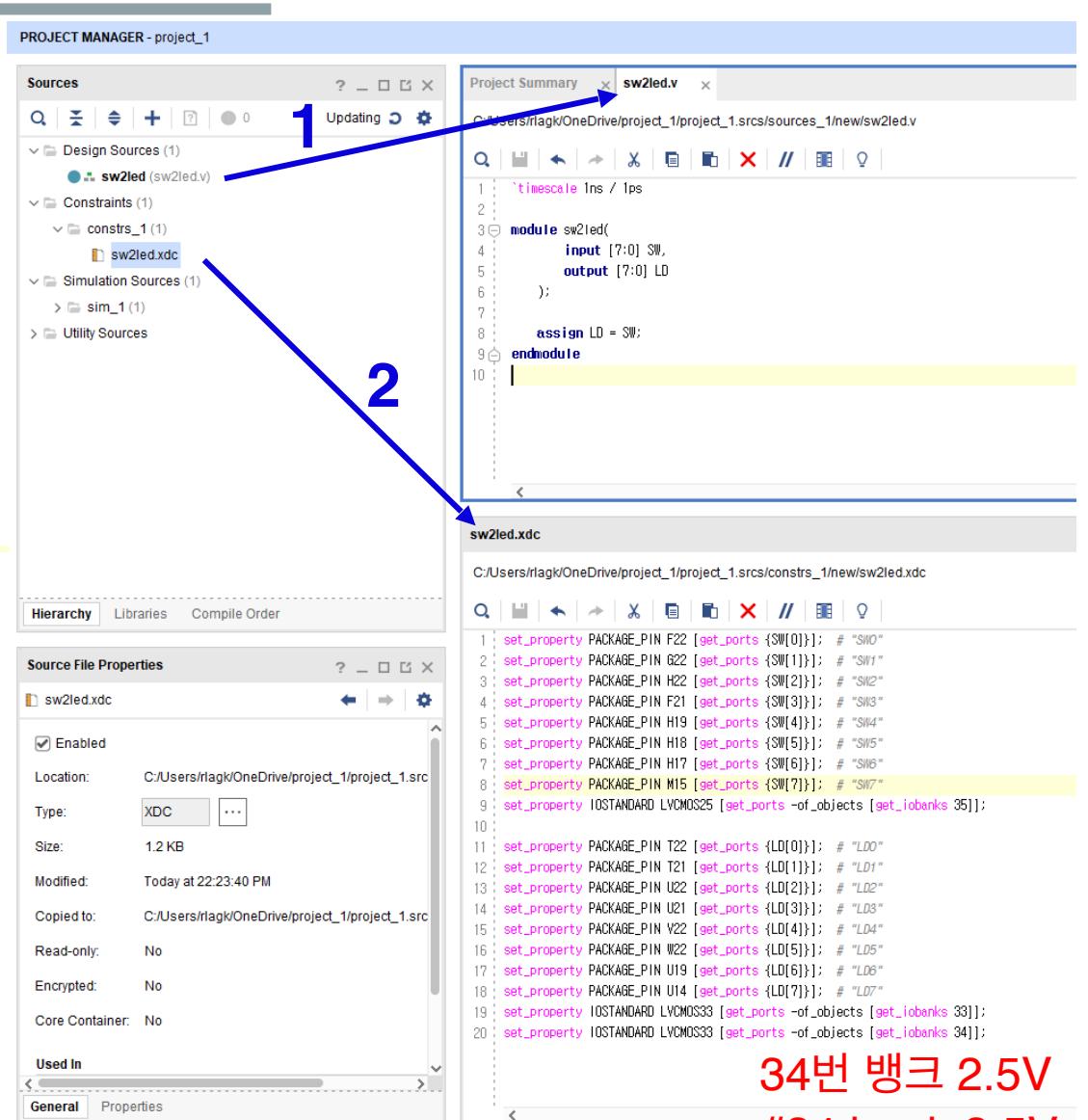
Edit your design

- A Verilog file (1)
 - Add or create **design sources**
 - Simple SW+LED module
 - [7:0] LD = [7:0] SW;
- An XDC (constraint) file (2)
 - Add or create **constraints**
 - 8-switches
 - PACKAGE_PIN: UG.20, DC.237, SM.9
 - IOSTANDARD: UG.5, DC.371, BD.J18
 - 8-LEDs
 - PACKAGE_PIN: UG.21, DC.175, SM.9
 - IOSTANDARD: UG.5, DC.362/367

```
module sw2led(
    input [7:0] SW,
    output [7:0] LD
);

    assign LD = SW;

endmodule
```



34번 뱅크 2.5V
#34 bank 2.5V

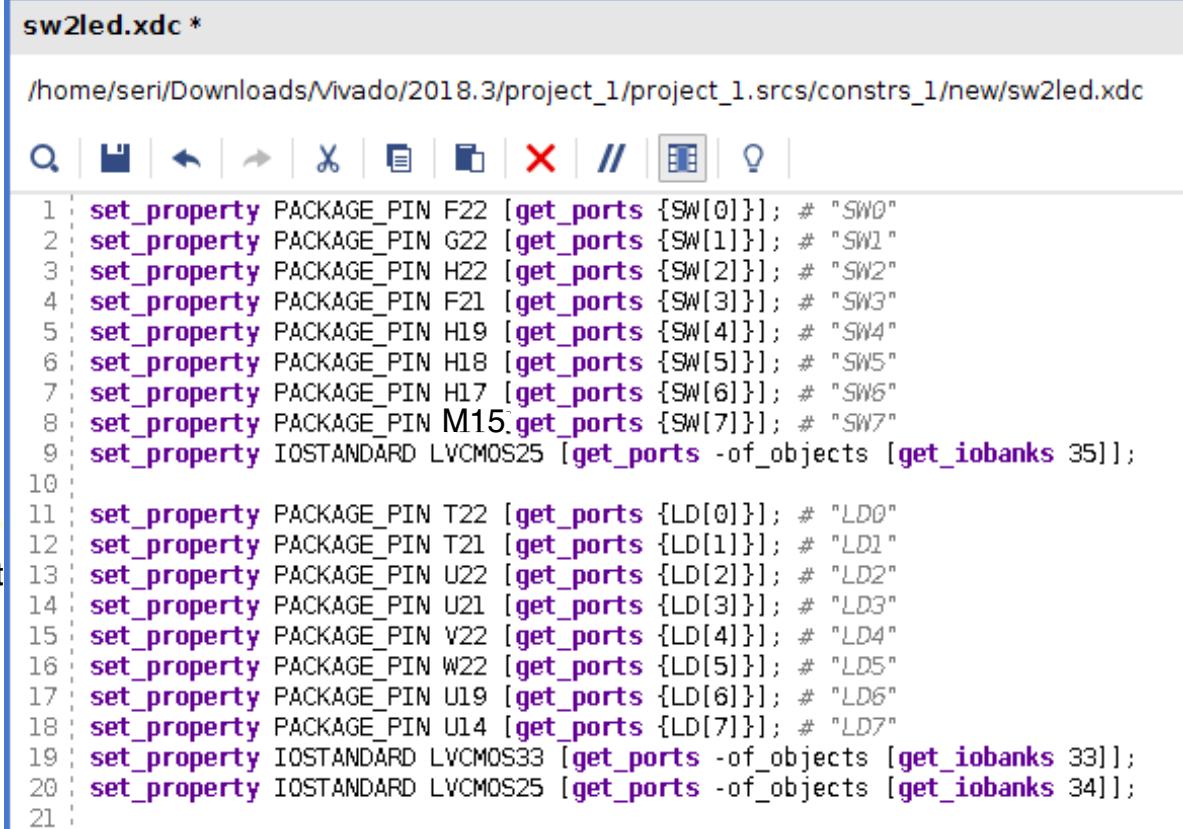
SW2LED Module

Edit your design

- A Verilog file (1)
 - Add or create **design sources**
 - Simple SW+LED module
 - $[7:0]$ LD = $[7:0]$ SW;
- An XDC (constraint) file (2)
 - Add or create **constraints**
 - 8-switches
 - PACKAGE_PIN: UG.20, DC.237, SM.9
 - IOSTANDARD: UG.5, DC.371, BD.J18
 - 8-LEDs
 - PACKAGE_PIN: UG.21, DC.175, SM.9
 - IOSTANDARD: UG.5, DC.362/367

```
module sw2led(
    input [7:0] SW,
    output [7:0] LD
);
    assign LD = SW;
endmodule
```

Text



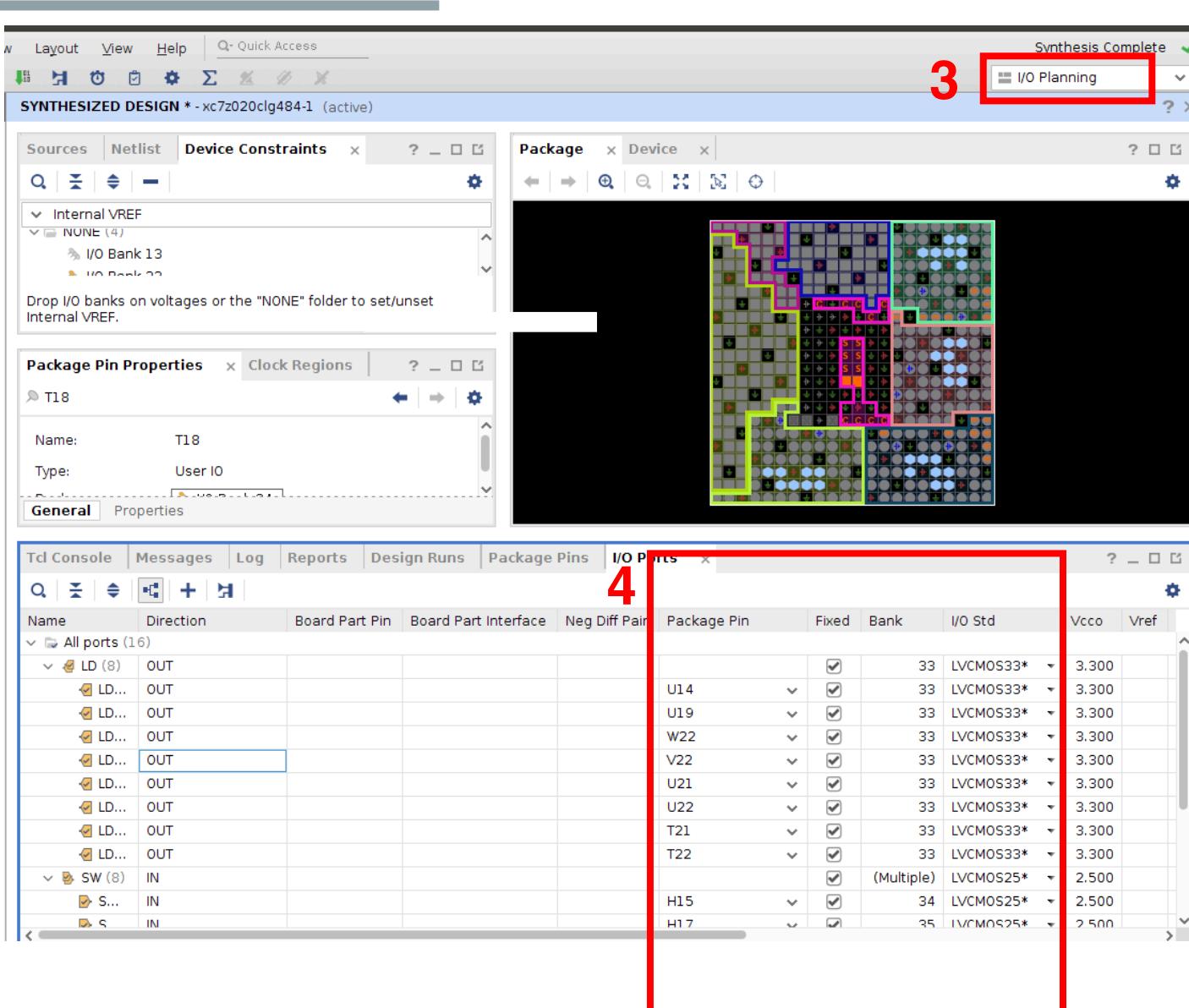
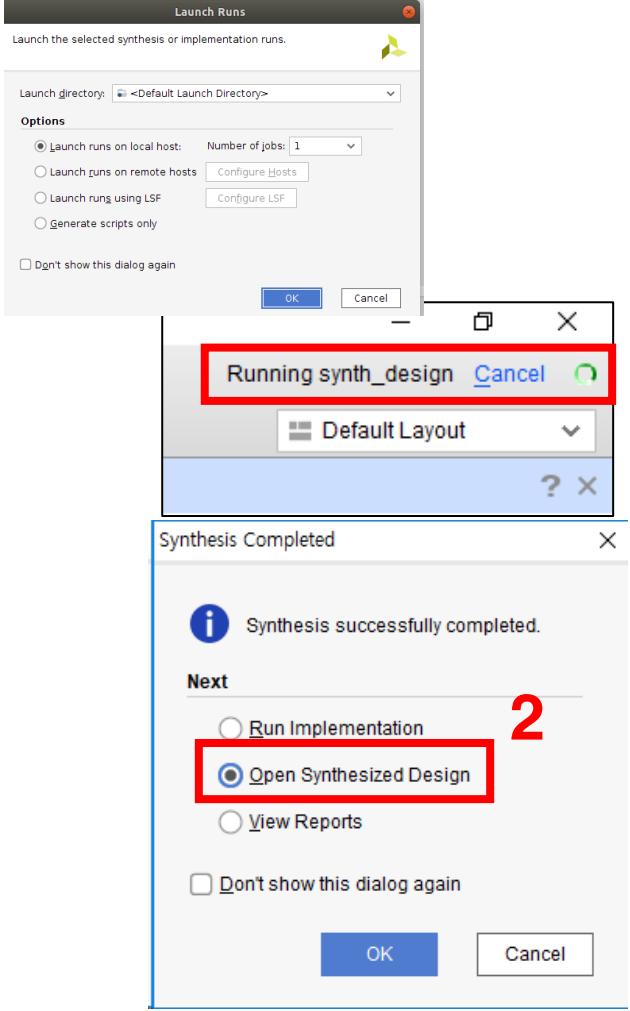
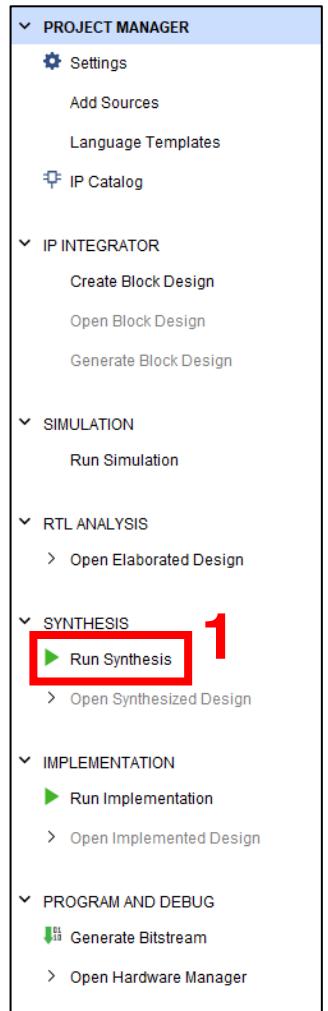
The screenshot shows the Vivado XDC editor interface with the file "sw2led.xdc" open. The code lists various package pin assignments and standard settings:

```
set_property PACKAGE_PIN F22 [get_ports {SW[0]}]; # "SW0"
set_property PACKAGE_PIN G22 [get_ports {SW[1]}]; # "SW1"
set_property PACKAGE_PIN H22 [get_ports {SW[2]}]; # "SW2"
set_property PACKAGE_PIN F21 [get_ports {SW[3]}]; # "SW3"
set_property PACKAGE_PIN H19 [get_ports {SW[4]}]; # "SW4"
set_property PACKAGE_PIN H18 [get_ports {SW[5]}]; # "SW5"
set_property PACKAGE_PIN H17 [get_ports {SW[6]}]; # "SW6"
set_property PACKAGE_PIN M15 [get_ports {SW[7]}]; # "SW7"
set_property IOSTANDARD LVCMOS25 [get_ports -of_objects [get_iobanks 35]];

set_property PACKAGE_PIN T22 [get_ports {LD[0]}]; # "LD0"
set_property PACKAGE_PIN T21 [get_ports {LD[1]}]; # "LD1"
set_property PACKAGE_PIN U22 [get_ports {LD[2]}]; # "LD2"
set_property PACKAGE_PIN U21 [get_ports {LD[3]}]; # "LD3"
set_property PACKAGE_PIN V22 [get_ports {LD[4]}]; # "LD4"
set_property PACKAGE_PIN W22 [get_ports {LD[5]}]; # "LD5"
set_property PACKAGE_PIN U19 [get_ports {LD[6]}]; # "LD6"
set_property PACKAGE_PIN U14 [get_ports {LD[7]}]; # "LD7"
set_property IOSTANDARD LVCMOS33 [get_ports -of_objects [get_iobanks 33]];
set_property IOSTANDARD LVCMOS25 [get_ports -of_objects [get_iobanks 34]];
```

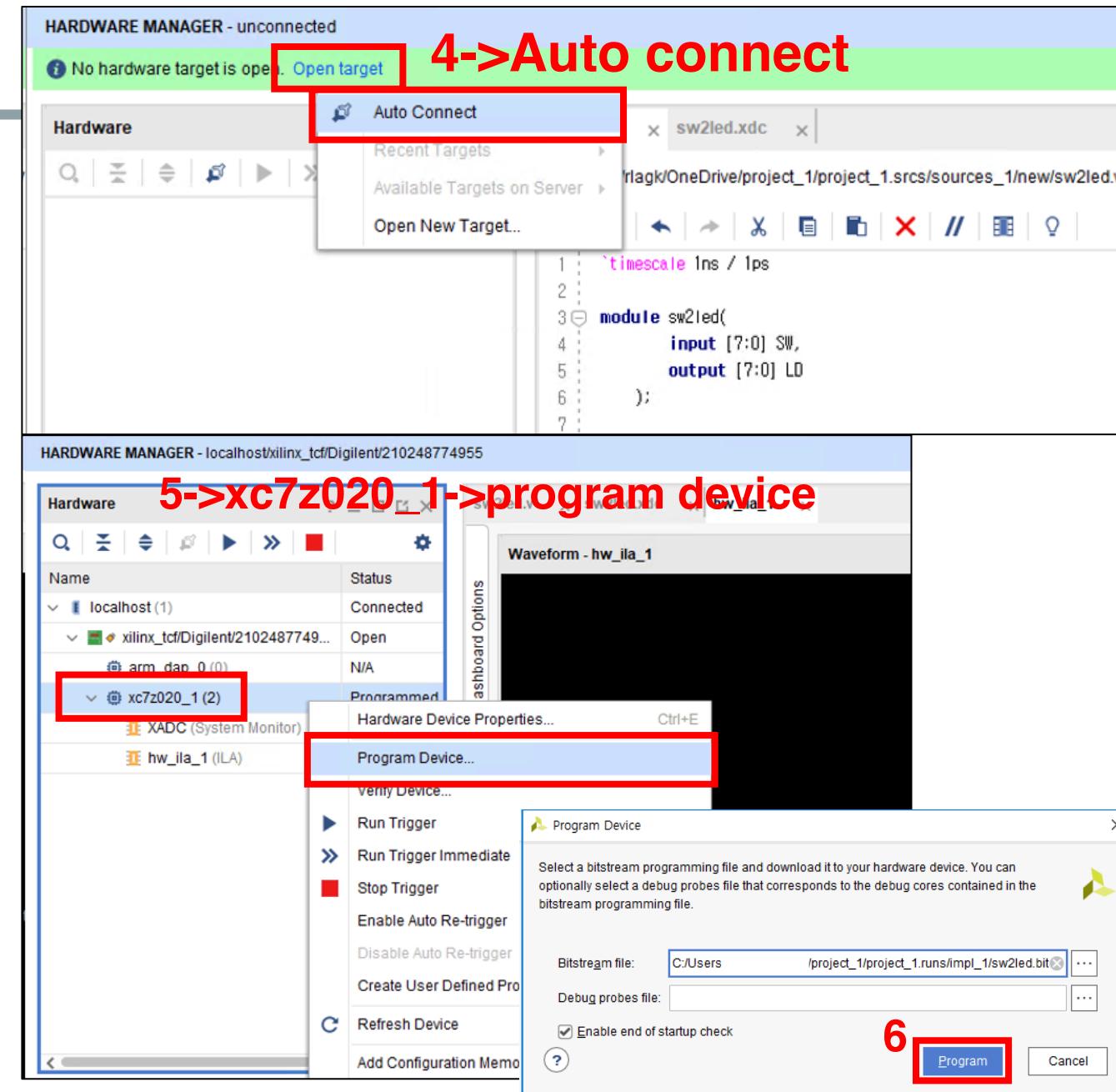
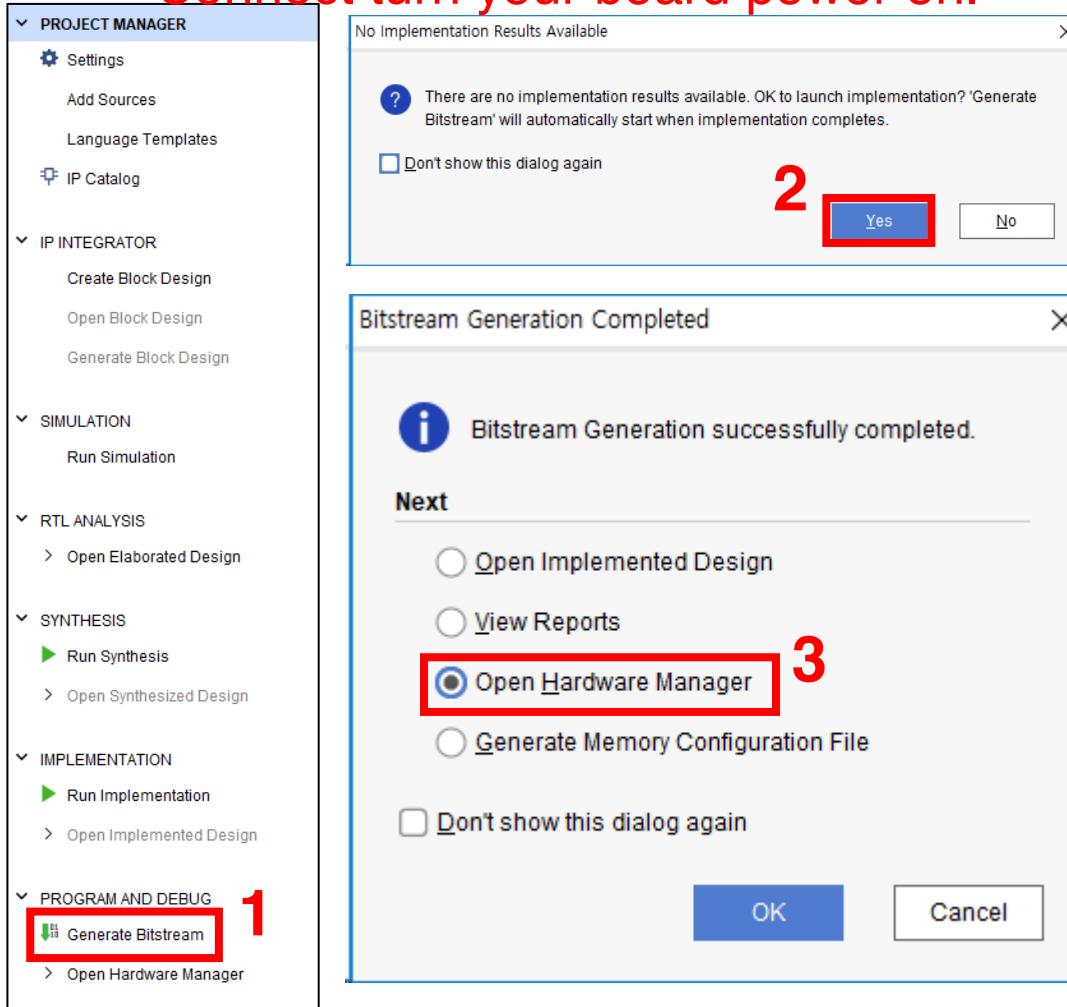
Synthesis

■ Synthesize and check



Implementation

- Implement your design into FPGA
 - Connect turn your board power on.



Enjoy Your Design!

- Toggle BD.SW0~7
 - And see BD.LD0~7

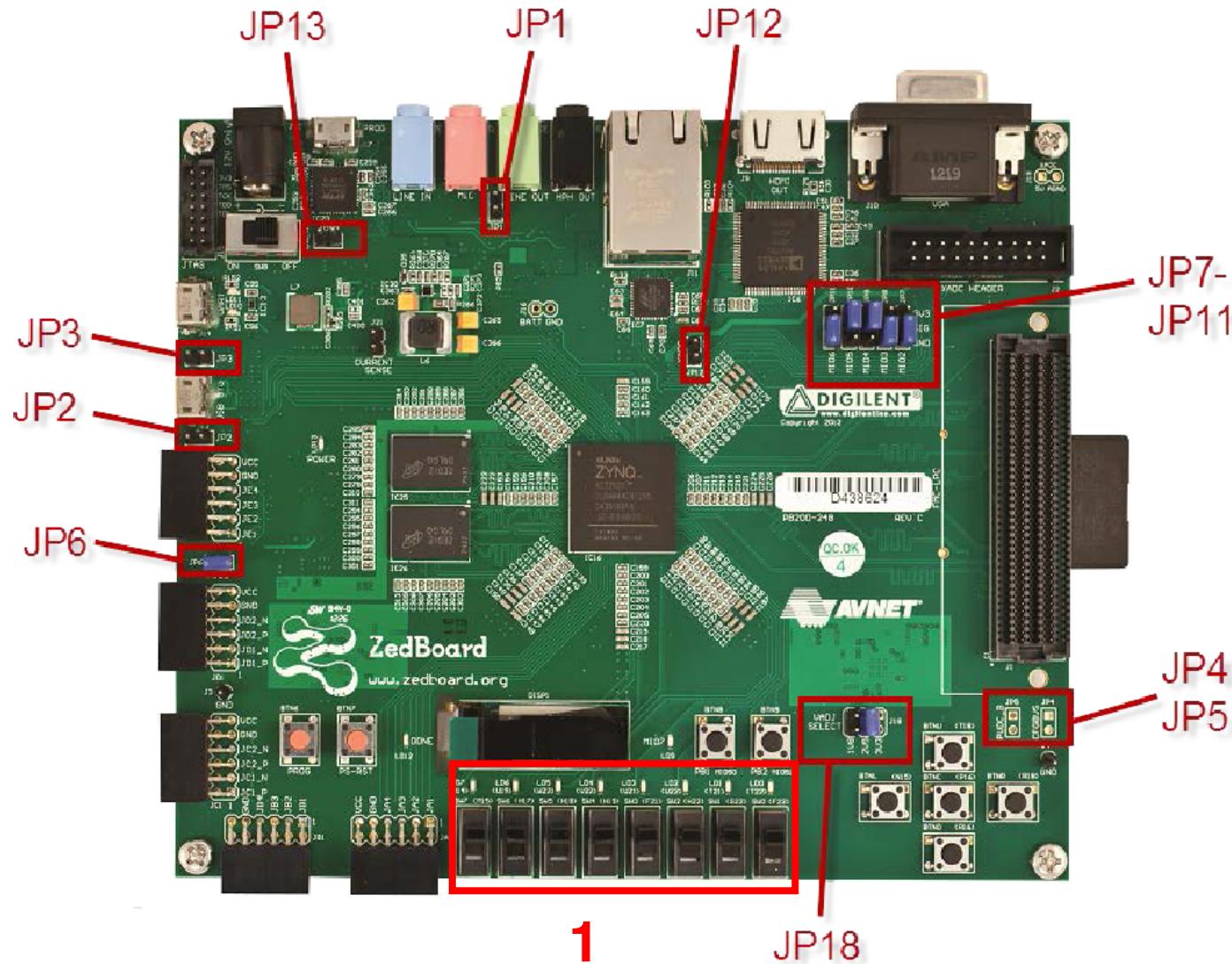
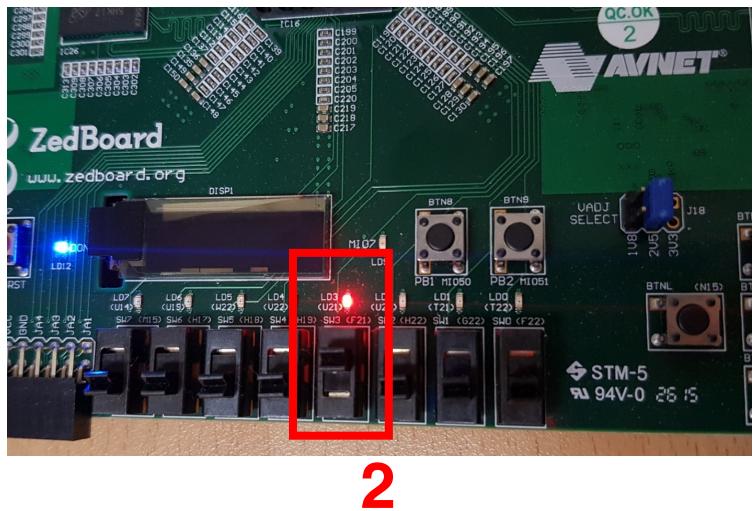


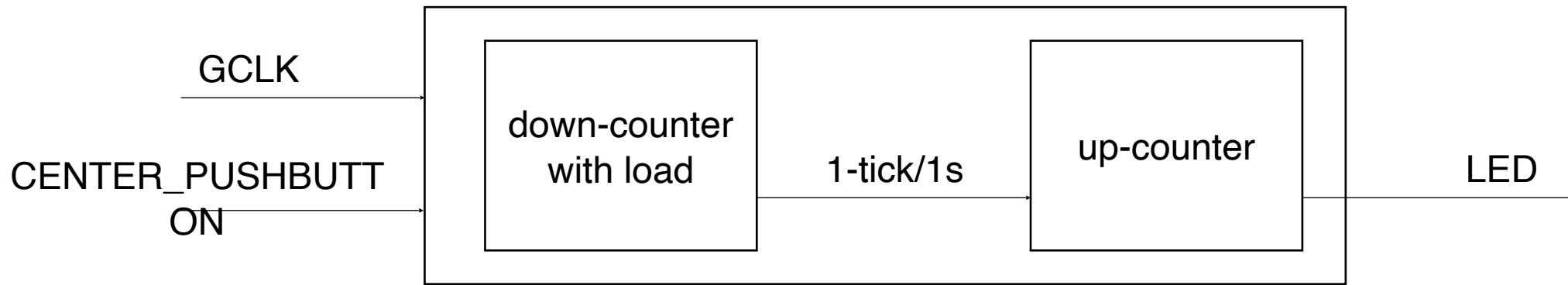
Figure 20 - ZedBoard Jumper Map

Main Practice

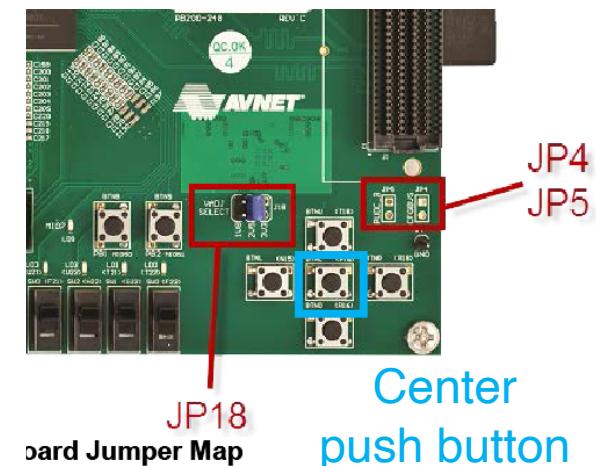
Practice

■ Implementing a simple sequential logic (1-sec checker)

- Implement a “1-sec checker” working with your FPGA board.



- 1-sec checker requirements
 - Use GCLK as an input clock (UG.4) and manipulate it to count 1s
 - Build an additional down-counter
 - Design a simple up-counter ticking every 1-second
 - Use [7:0] LED to visualize the counter value
 - Use CENTER_PUSHBUTTON to assert synchronous reset (UG.4)



Don't Break ZedBoard

- We have no extra ZedBoard to give students who break their ZedBoard
- So please use your ZedBoard carefully

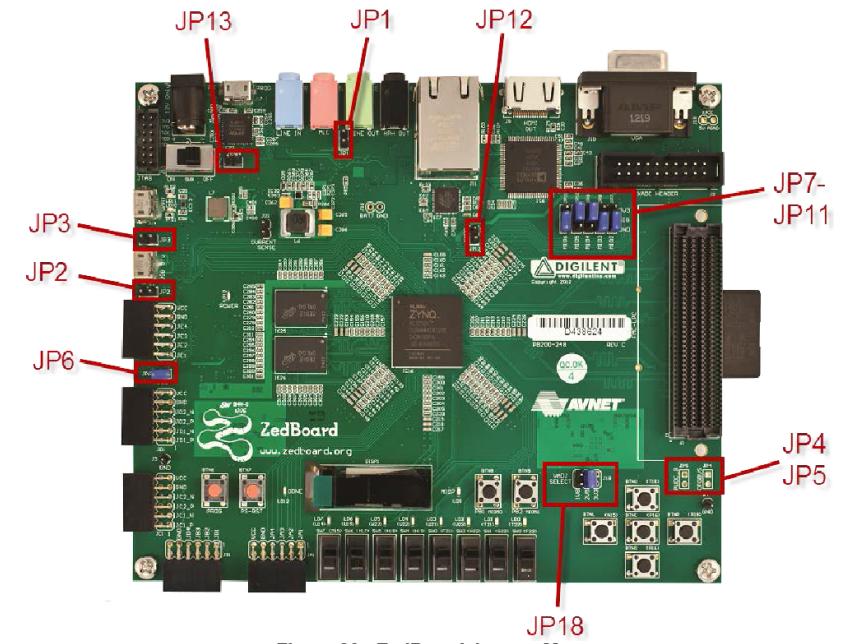


Figure 20 - ZedBoard Jumper Map

Homework

- Requirements

- Result

- Attach your project folder with all your verilog codes (1-sec checker)
 - Attach a video that can show your “1-sec checker” works with LED

- Report

- Explain “1-sec checker” that you implemented
 - In your own words
 - Either in Korean or in English
 - # of pages does not matter
 - **PDF only!!**

- **Result + Report to a .zip**

- Upload (.zip) file on ETL

- Submit one (.zip) file

- zip file name : [Lab07]name.zip (ex : [Lab08]홍길동.zip)

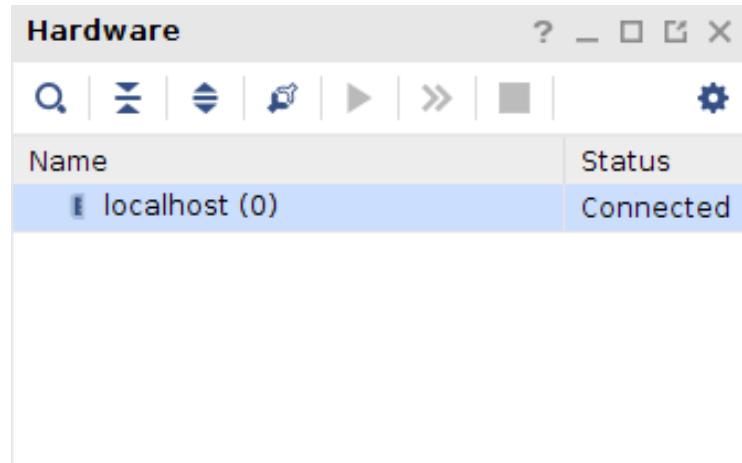
- Due: 5/5(WED) 23:59

- **No Late Submission**

Appendix

Trouble Shooting

Board Driver



다름이 아니라 제가 etl 에도 local host(0)으로 보드가 인식이 안된다고 계속 뜨는데
집에 있는 다른 충전기(?)를 써봐도 연결이 안되는것 같고
제가 집에 다른 윈도우 컴퓨터가 없어서 달리 작동 방법을 확인해볼 수가 없는데

이럴 때는 그냥 랩실에 가서 컴퓨터를 사용해야하는 것인가요ㅠㅠ
혹시 보드에 문제가 있는것은 아니겠죠?

이것저것 검색하다보니

아래처럼 이렇게 `install_drivers_wrapper` 를 수동으로 다시 깔아주고 (또 admin 권한때문에 .bat 파일도 수정했습니다) 해결했습니다!!!

혹여나 다른 학생들도 비슷한 문제가 생긴다면 말씀해주시면 좋을것같습니다 :)

감사합니다.

```
cd <Vivado install path>\data\xicom\cable_drivers\nt64
```

```
install_drivers_wrapper.bat <Vivado install path>\data\xicom\cable_drivers\nt64 C<Vivado install path>\install.log  
<Vivado install path>\
```