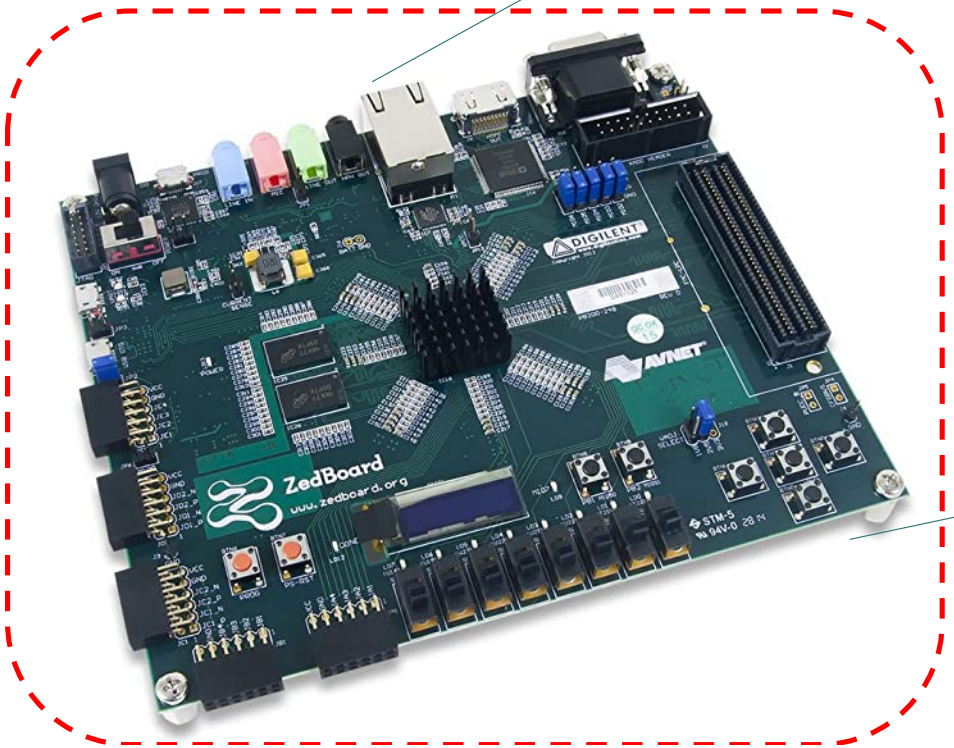
Practice 8

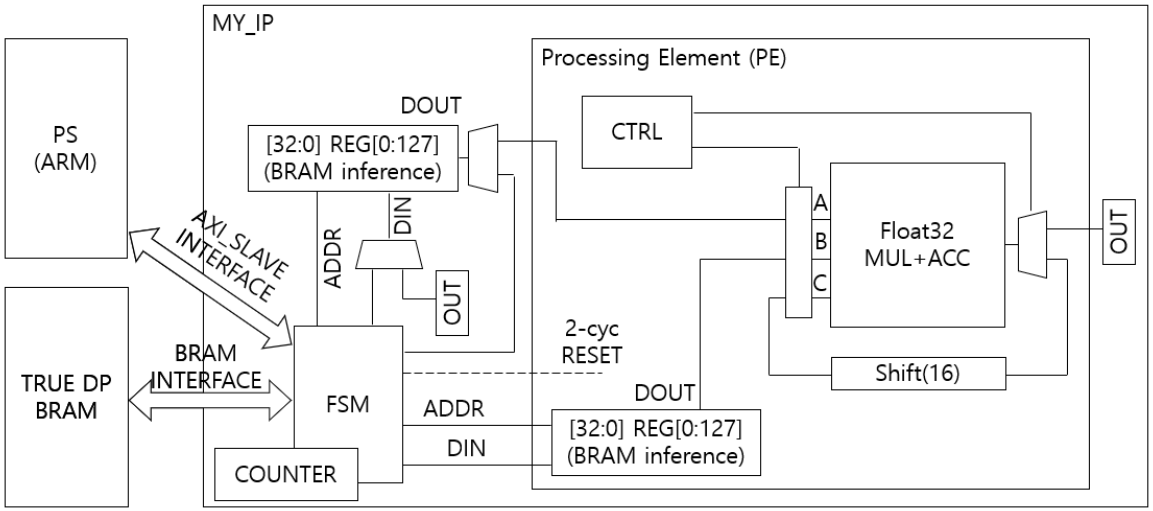
- How to use FPGA board

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

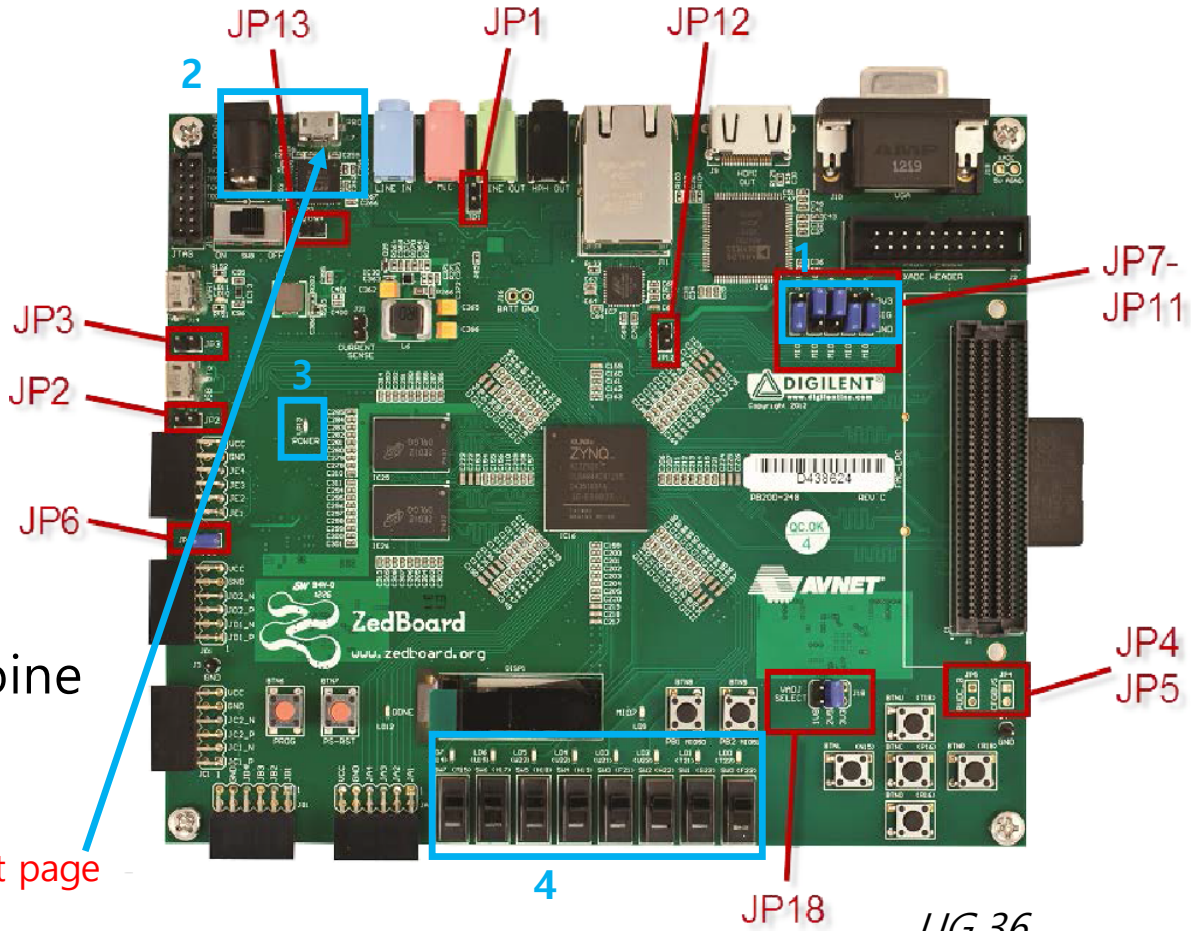
Notations

- 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
- DC.LLL (Design Constraint)
  - Line LLL of zedboard_master_XDC_RevC_D_v3.xdc
- BD.PPP (Board)
  - Part PPP of the board

[ZEDBoard Hardware User's Guide, 2014]

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

Figure 20 - ZedBoard Jumper Map

UG.36

[ZEDBoard Hardware User's Guide, 2014]

Cautions for the Board

- Micro 5pin slots
  - Well known for breaking down.
    - Weak soldering
    - Stiff
    - Structural deficientcy
  - 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)

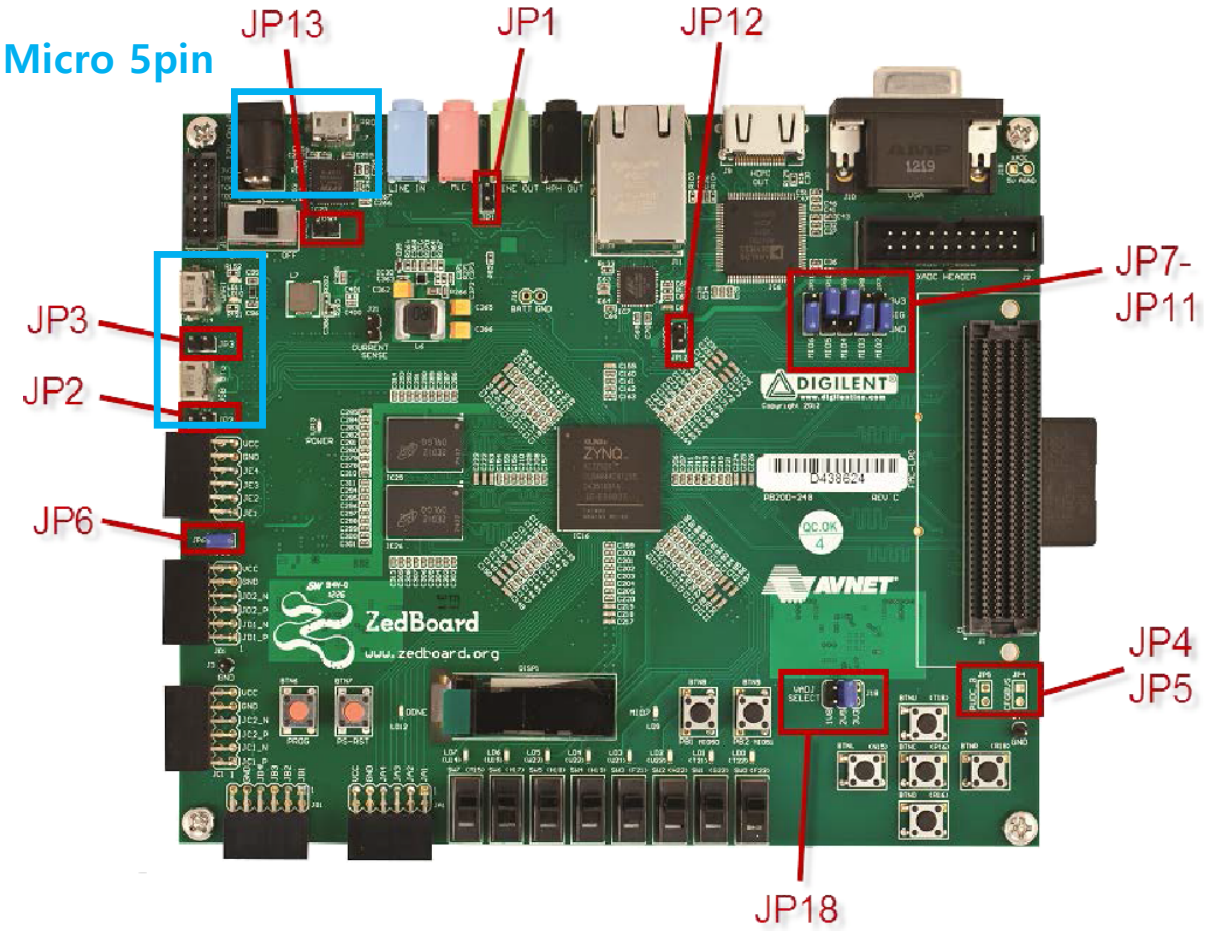
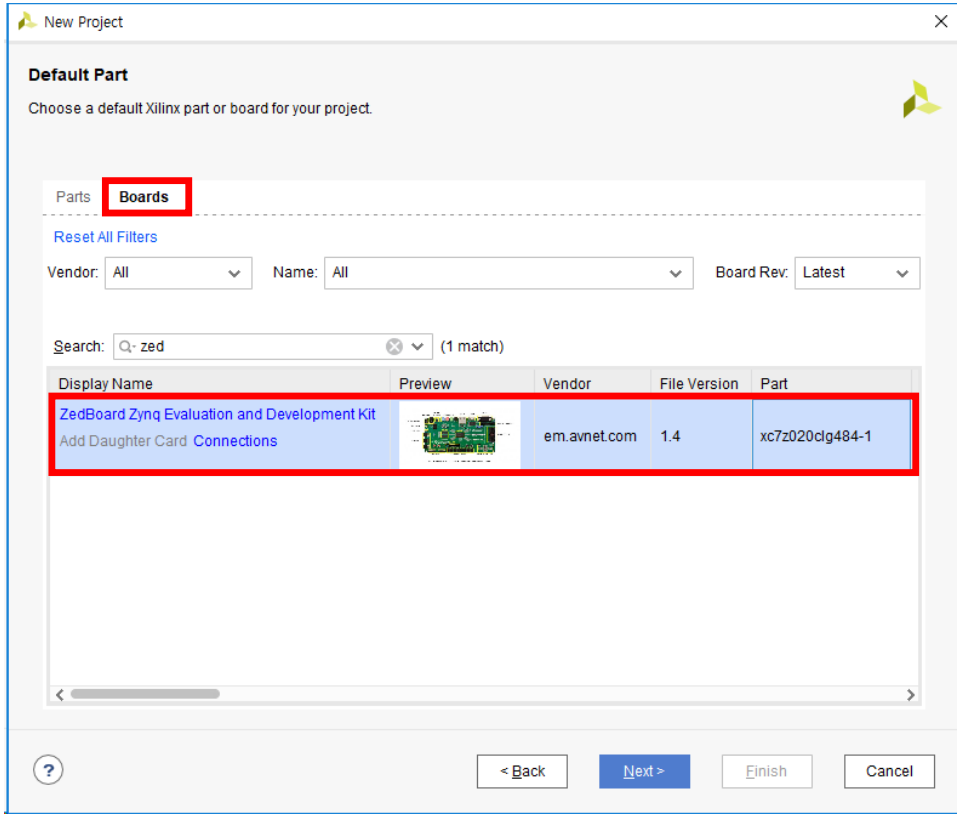


Figure 20 - ZedBoard Jumper Map

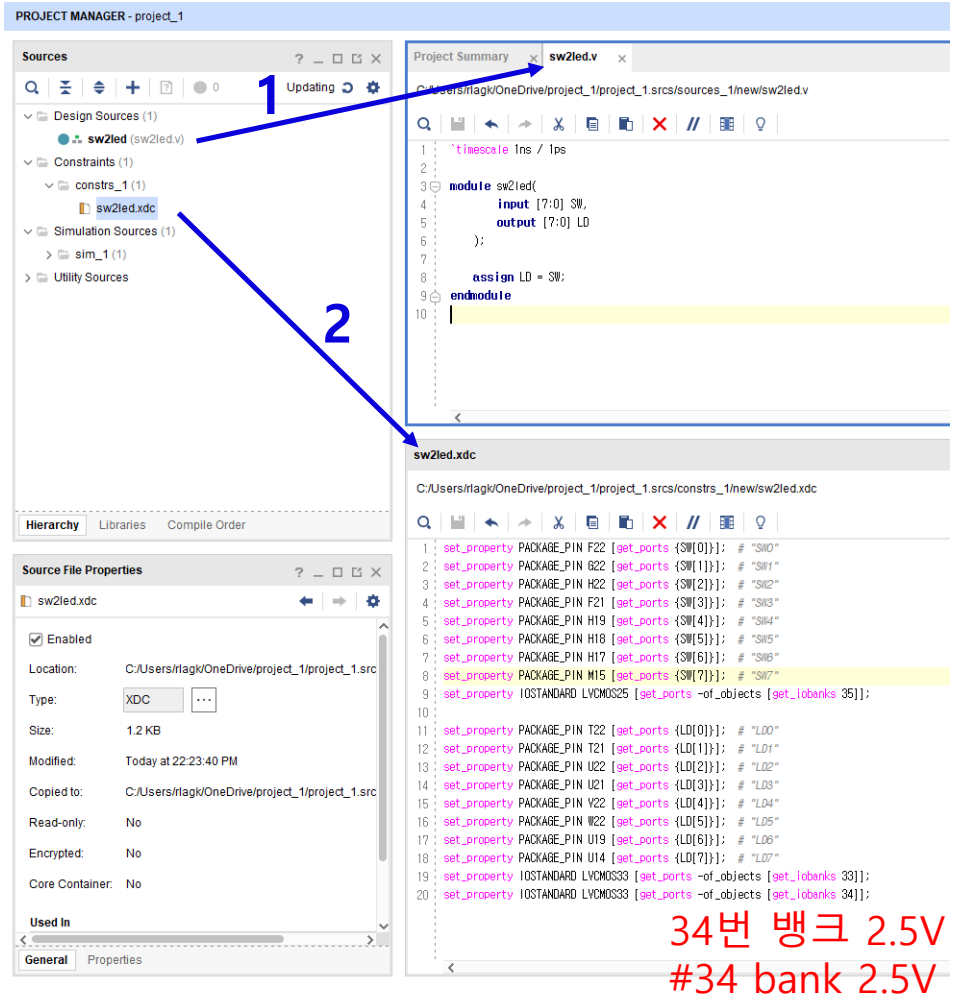
Vivado Project Creation

- Create Project - Choose part or board
  - ZedBoard



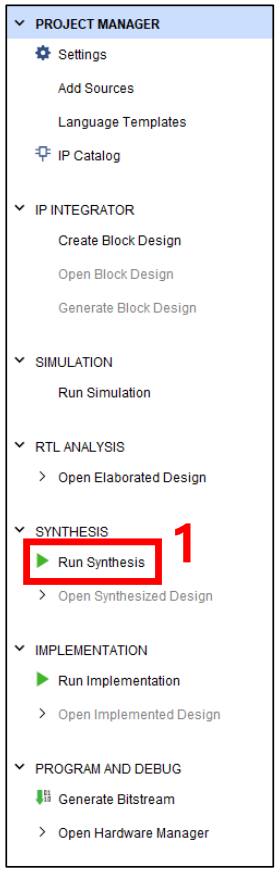
SW2LED Module

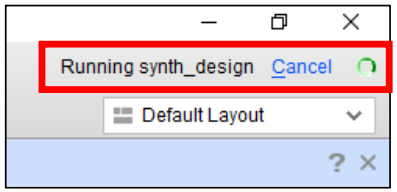
- Edit your design
  - A Verilog file (1)
    - Add or create **design sources**
    - Simple SW+LED module
      - [7:0] LD = [7:0] SW;
  - An XDC (constraint) file (2)
    - Add or create **constraints**
      - 8-switches
        - PACKAGE_PIN: UG.20, DC.237, SM.9
        - IOSTANDARD: UG.5, DC.371,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

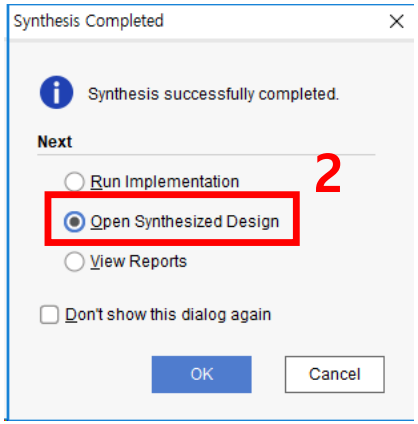


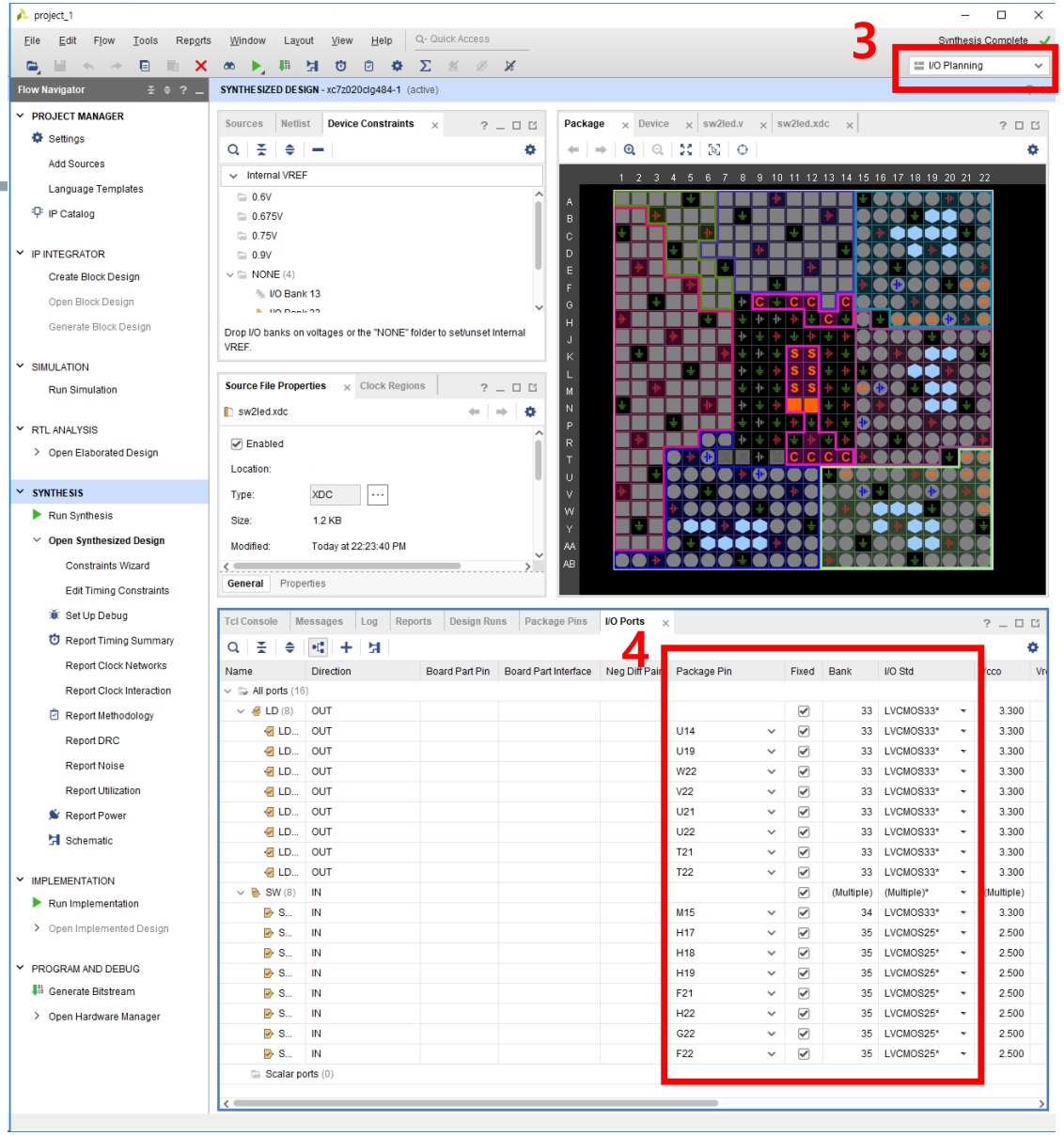
Synthesis

- Synthesize and check



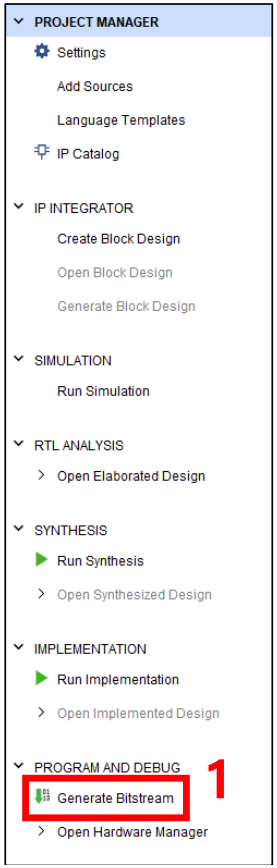


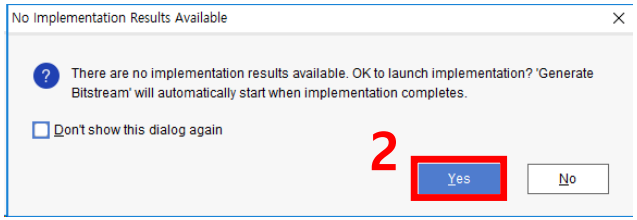


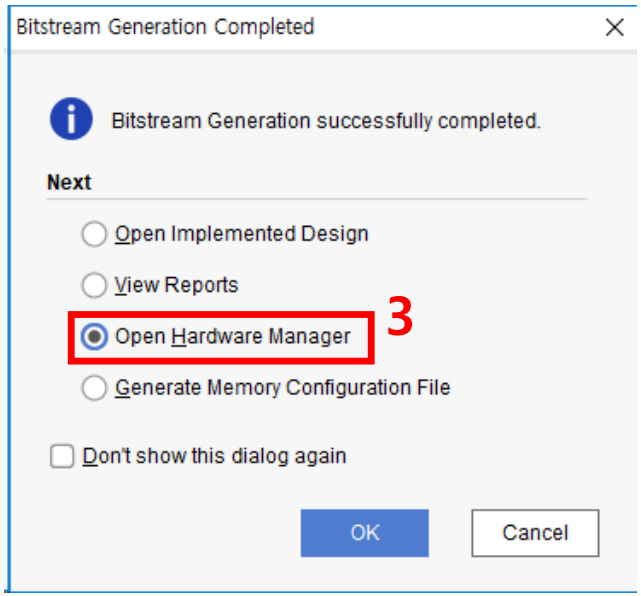


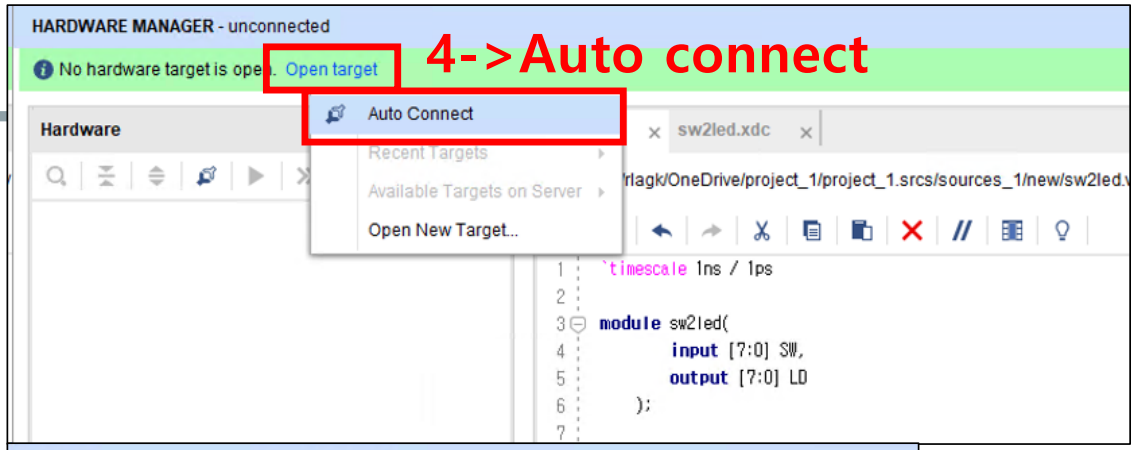
Implementation

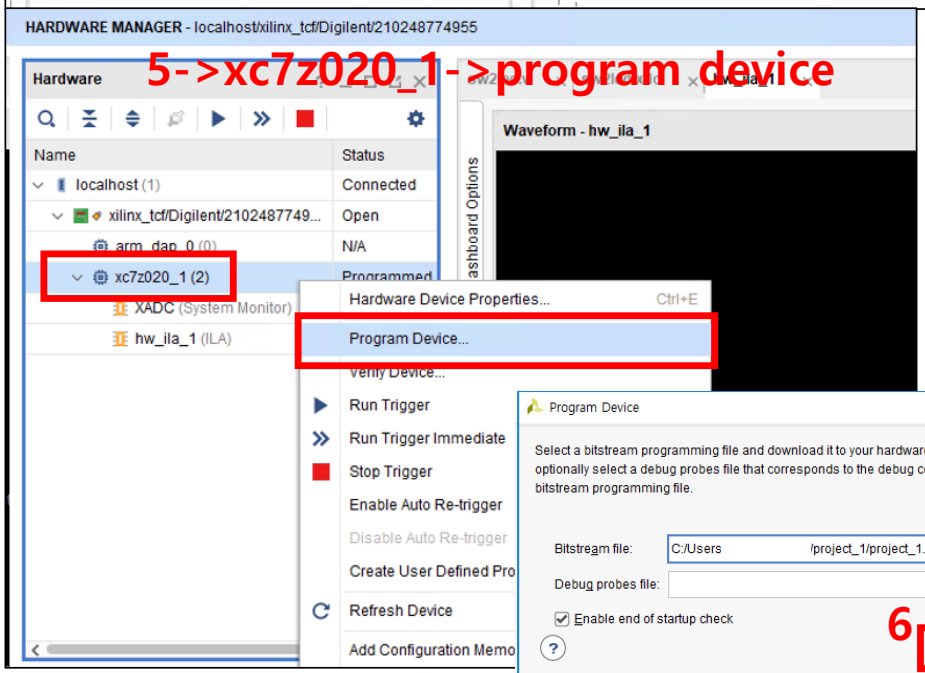
- Implement your design into FPGA
  - Connect turn your board power on.







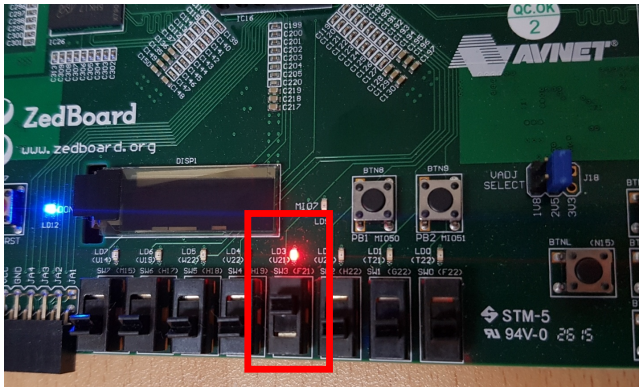




[ZEDBoard Hardware User's Guide, 2014]

Enjoy Your Design!

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



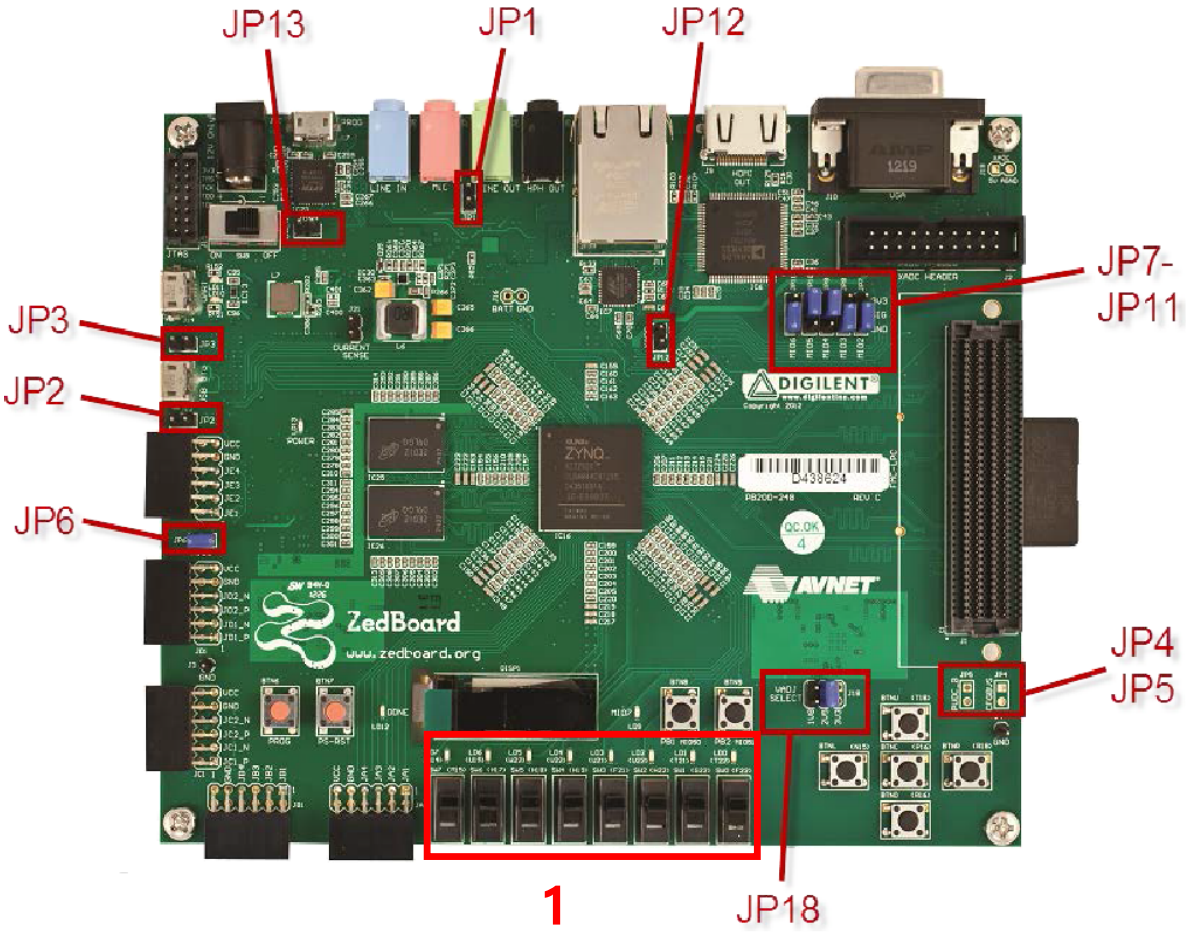
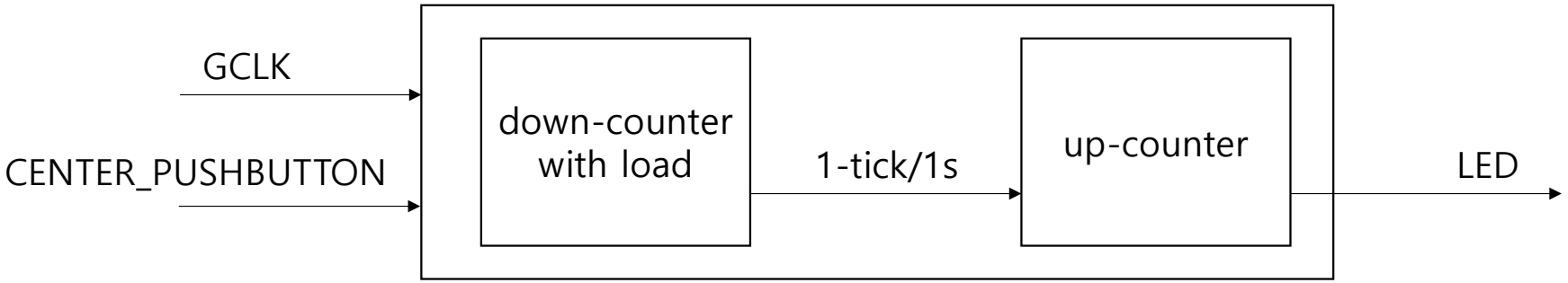


Figure 20 - ZedBoard Jumper Map

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

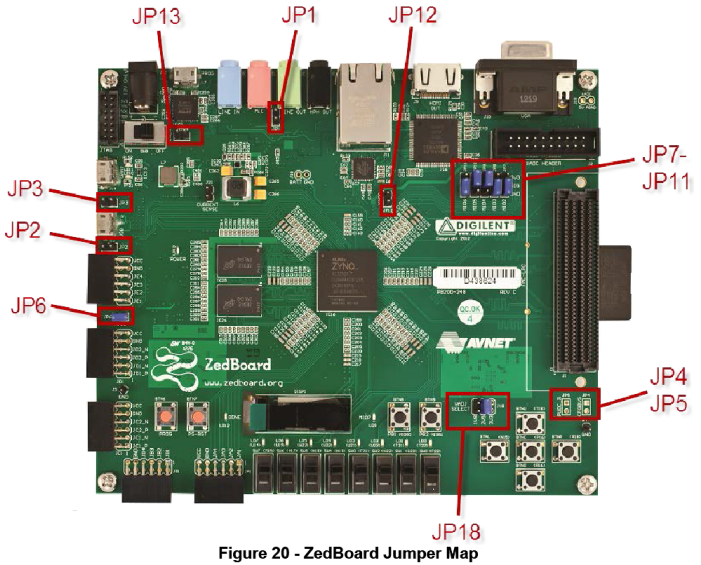


Figure 20 - ZedBoard Jumper Map

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 : [Lab08]name.zip (ex : [Lab08]홍길동.zip)
  - Due: 5/12(TUE) 23:59
    - **No Late Submission**