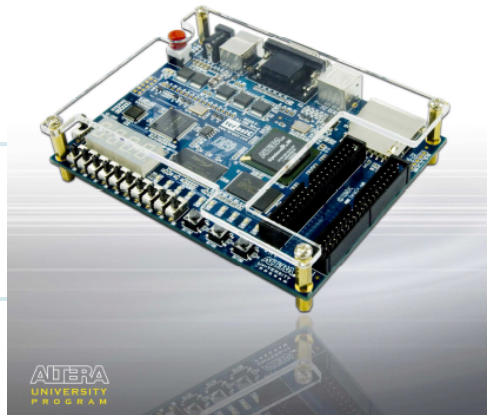
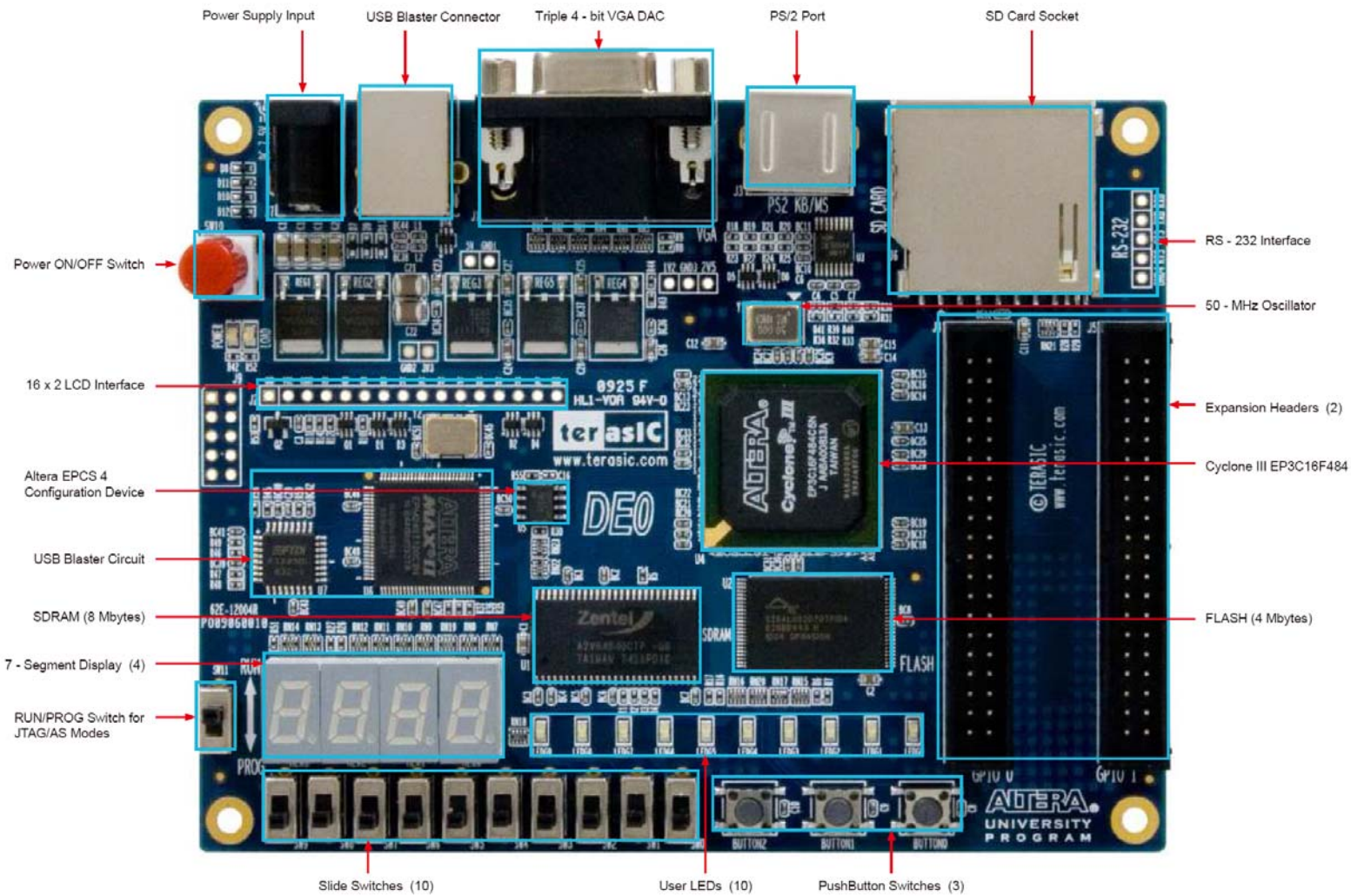


Class 5
DE0 FPGA Development Board and Quartus II FPGA Design Software

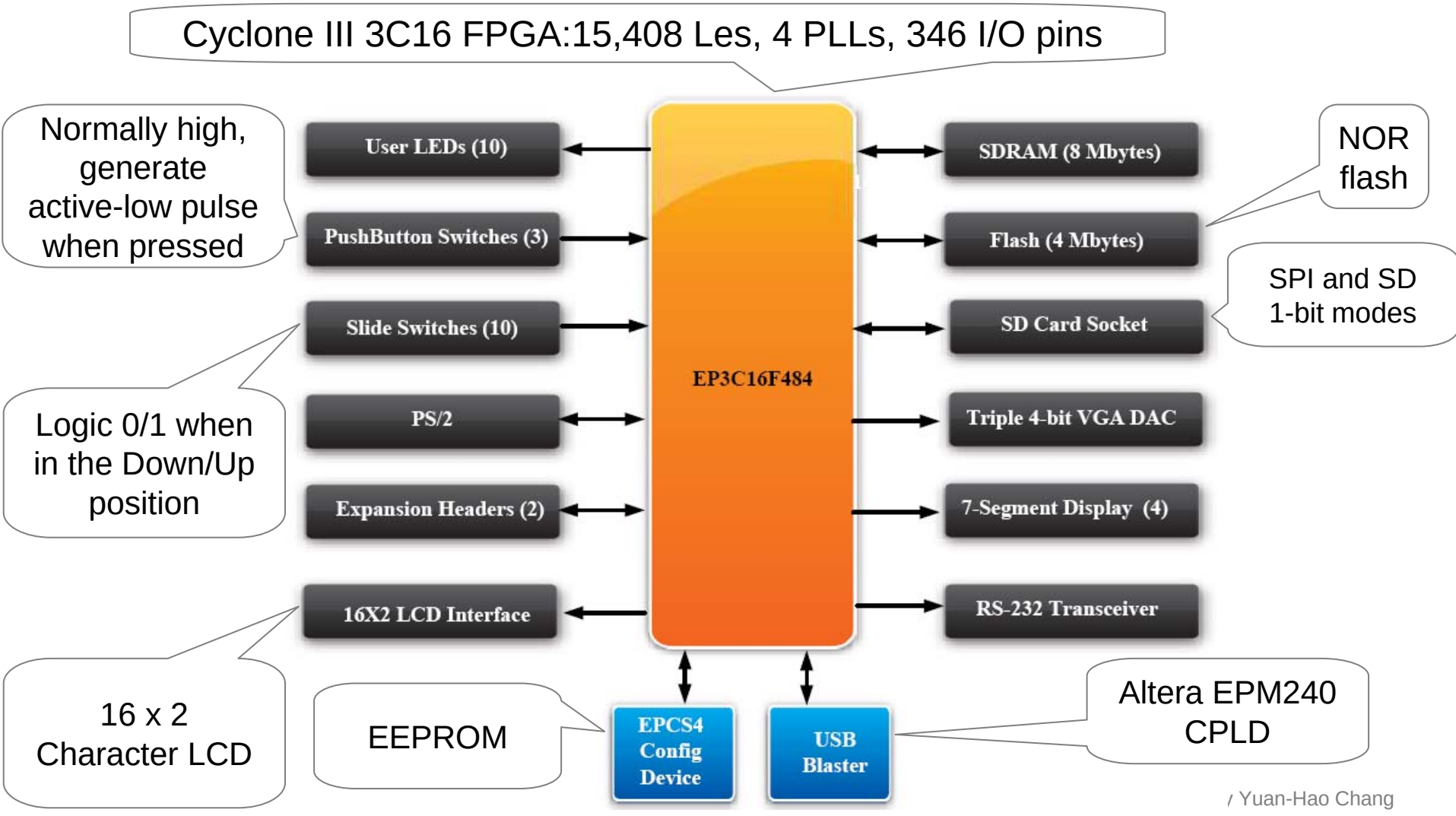
Terasic DE0 Field Programmable Gate Array (FPGA) Development Board



Layout and Components of DE0



Block Diagram of the DE0 Board



Power-Up the DE0 Board

- The DE0 board comes with a preloaded configuration bit stream to demonstrate some feature of the board.
  - All user LEDs are flashing.
  - All 7-segment displays are cycling through 0 to F.
  - The VGA monitor displays the image as shown a the right-hand side:

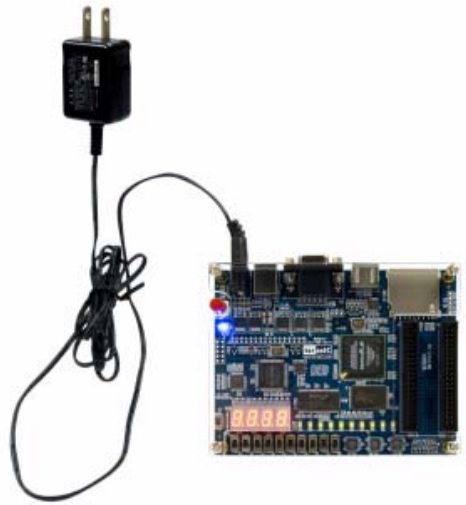


DE0 Installation

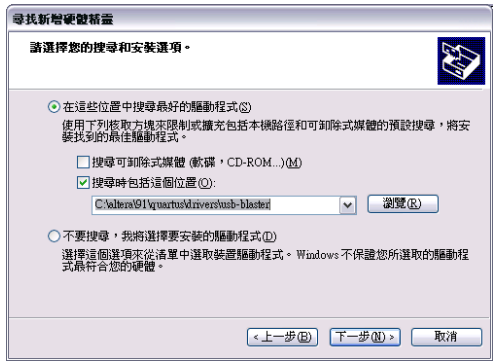
- Step 1: Install the Altera Design Software on the host computer.
  - Download the software: http://www.altera.com/download
    - Quartus II: the primary FPGA development tool
    - Nios II: soft-core embedded processor
    - ModelSim-Altera: Simulation tool

Windows Software Downloads	Download	File Size
Quartus® II Web Edition Software v9.1 Service Pack 1 (Now with the MegaCore® IP Library, which includes the Nios® II Processor) Windows Vista (32 bits) and Windows XP (32 bits)	Download No license required	1.5 GB
Nios II Embedded Design Suite *(1)* Windows Vista (32 bits) and Windows XP (32 bits)	Download Download Service Pack No license required	563 MB 13 MB
ModelSim®-Altera® Starter Edition v6.5b for Quartus II Software v9.1 Windows Vista (32 bits) and Windows XP (32 bits)	Download Download Service Pack No license required	573 MB 574 MB

DE0 Installation (Cont.)

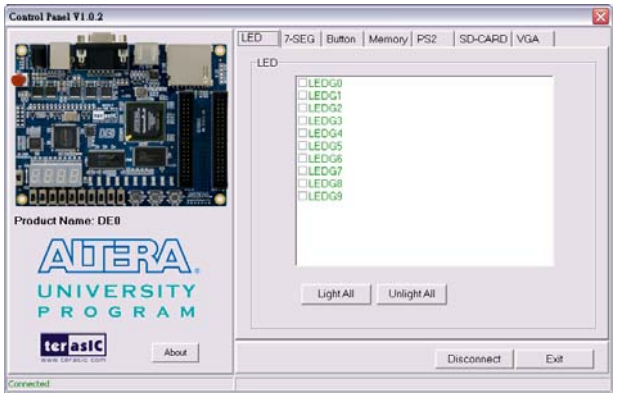


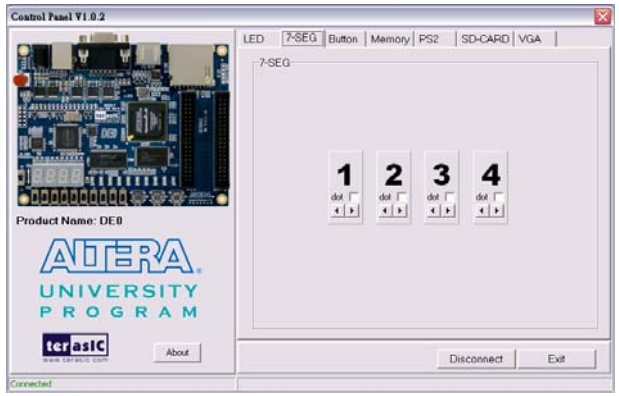
- Step 2: Install the USB Blaster
  - Plug in the power cable.
  - Use the USB cable co connect the USB connector on the DE0 board to a USB port on a computer.
    - 1. Recognize the new hardware connected
    - 2. Specify the path for USB Blaster driver
    - 3. Select appropriate driver (**C:\altera\91\quartus\drivers\usb-blaster**)
    - 4. Install USB Blast driver (**C:\altera\91\quartus\drivers\usb-blaster\x32**)



DE0 Control Panel

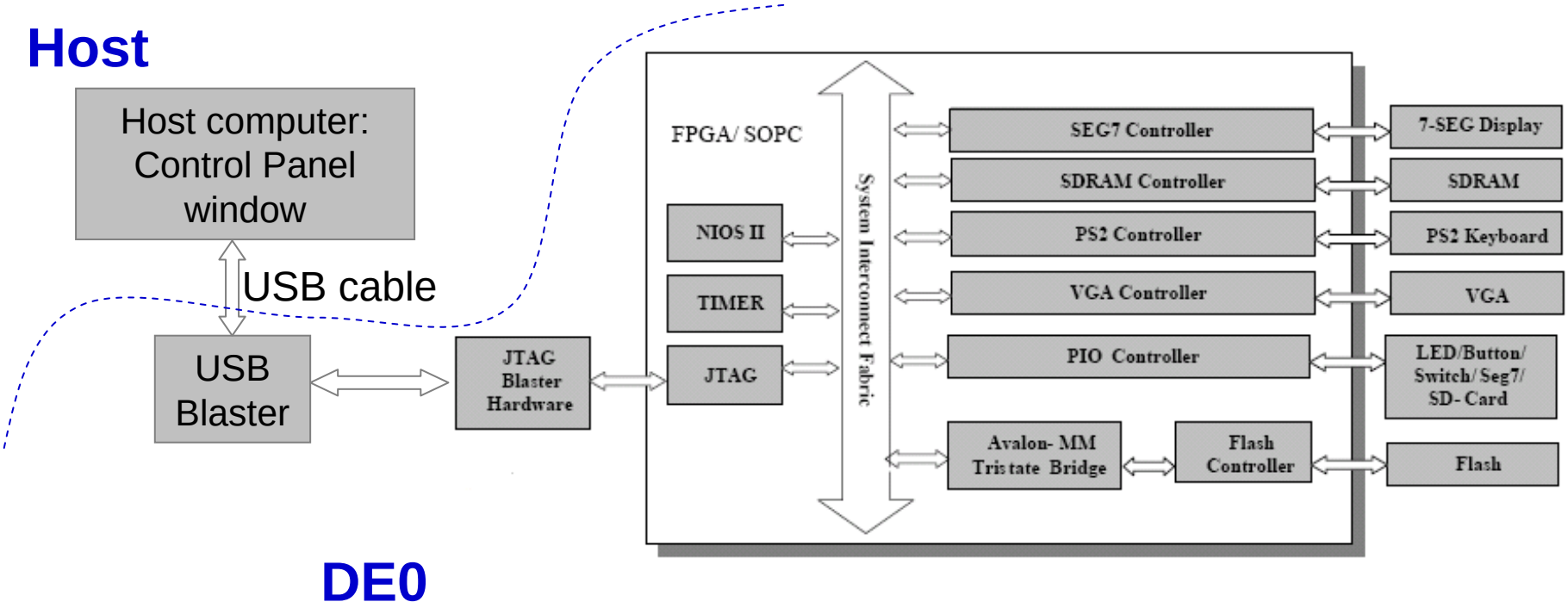
- The DE0 board comes with a Control Panel facility. (Start the executable *DE0_ControlPanel.exe*)
  - Allows users to access various components on the board from a host computer.
  - Connect the host computer with the DE0 board through an USB connection.
  - Verify the functionality of components on the board.





DE0 Control Panel (Cont.)

- The control codes that perform the control functions
  - Is implemented in the FPGA board, and
  - Communicates with the Control Panel window on the host computer.



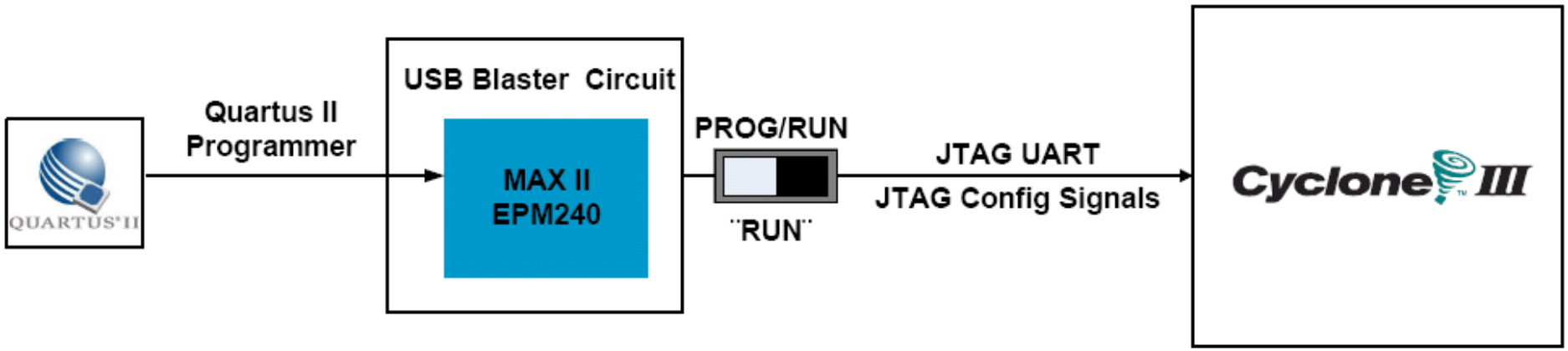


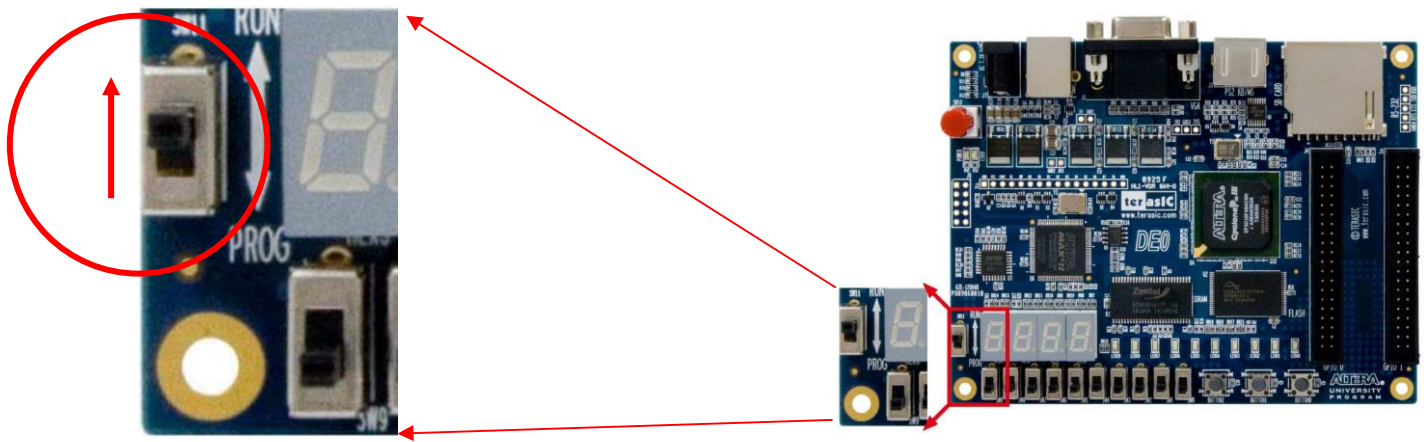
Configuring the Cyclone III FPGA

- The DE0 board contains a serial EEPROM chip (i.e., the EPCS4 device) that stores configuration data for the Cyclone III FPGA.
  - The configuration data is automatically loaded from the EEPROM chip into the FPGA once the power is applied to the board.
  - With Quartus II, it is possible to reprogram the FPGA and to change the non-volatile data in the EEPROM chip.
    - JTAG (Joint Test Action Group) programming: Download the configuration to FPGA directly, but the configuration is lost when the power is off.
    - AS (Active Serial) programming: Download the configuration into the EEPROM chip, and the configuration is retained when the power is off. When the power is on, data is loaded from the EEPROM.

Configuring the FPGA in JTAG Mode

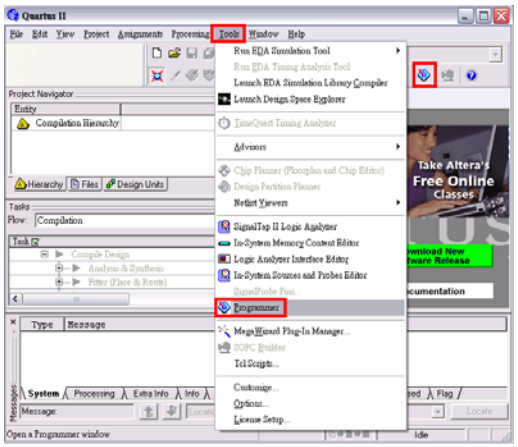
- Download the .sof (SRAM Object File) file by the programmer of Quartus II

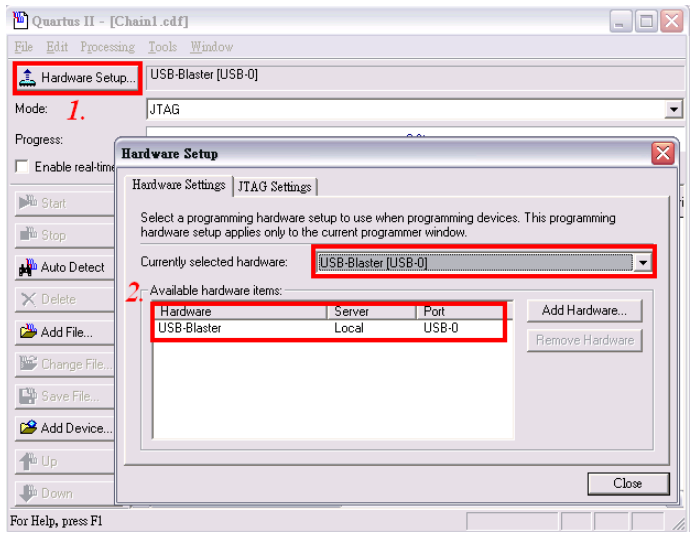


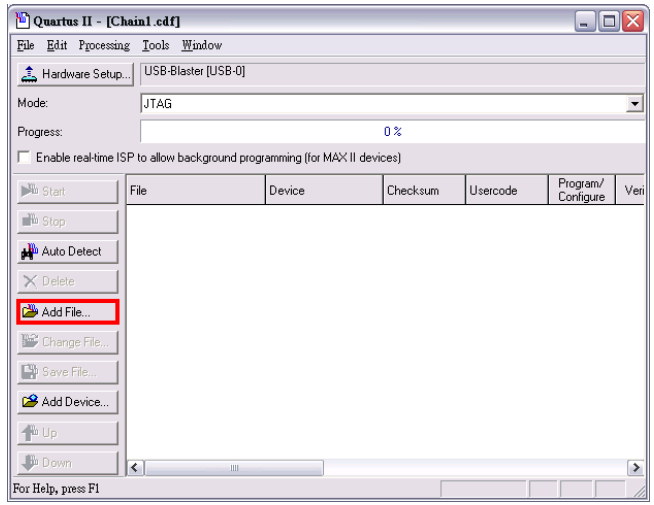


Configuring the FPGA in JTAG Mode (Cont.)

- The steps to program SRAM Object File (.sof) into the FPGA device on the DE0 board are as follows:
  - Step 1: Power on DE0 board with SW11 to **RUN** mode and connect it to the host.
    Step 2: Open Quartus II, and choose **Tools → Programmer**
  - Step 3: Click "**Hardware Setup**" and then Select "**USB Blaster**"
  - Step 4: Click "**Add File**" to select the .sof file in JTAG mode, and then click "**Start**" to program it.



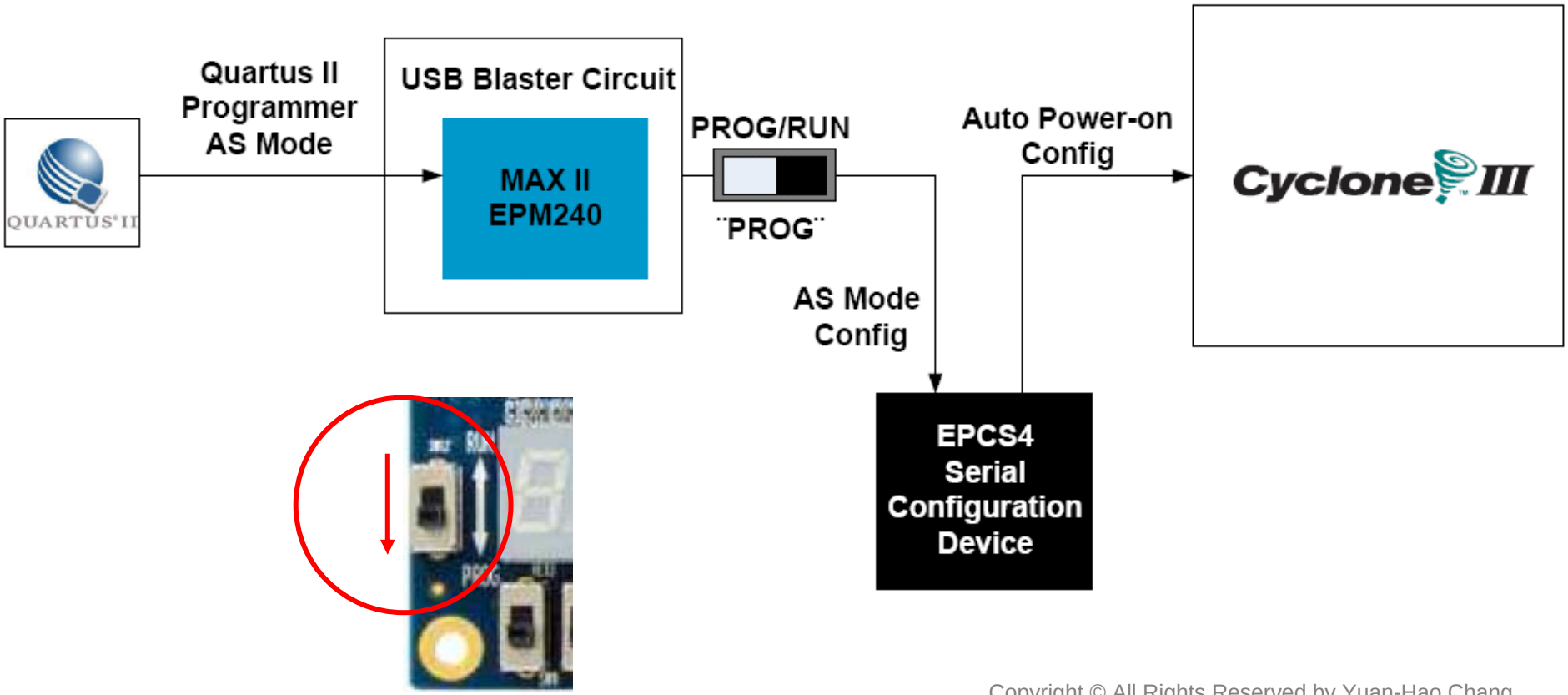






Configuring the EPCS in AS Mode

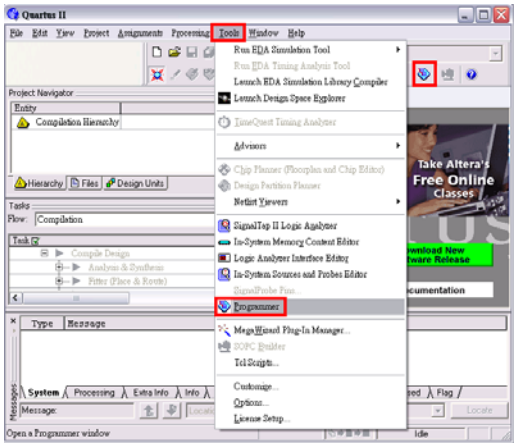
- Download the .pof (Programmer Object File) file by the programmer of Quartus II

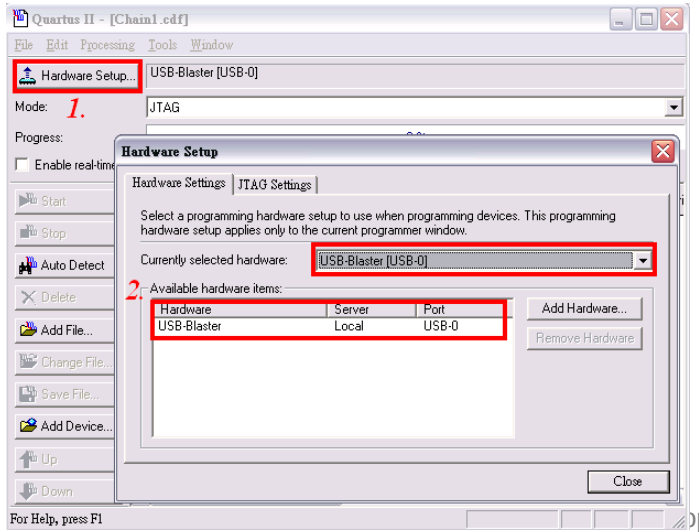


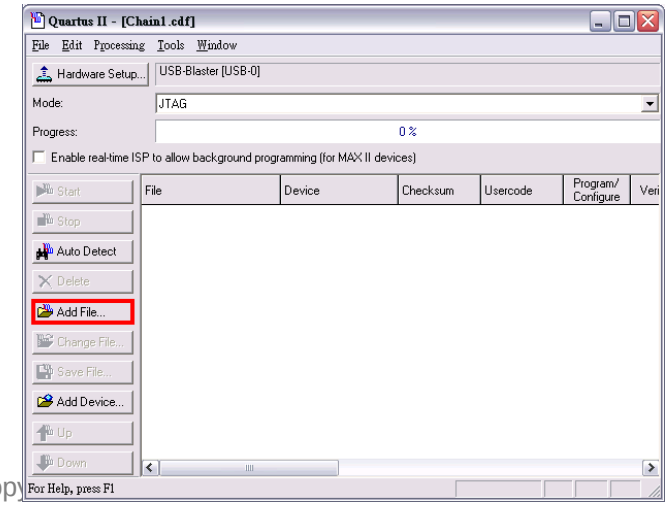


Configuring the EPCS4 in AS Mode (Cont.)

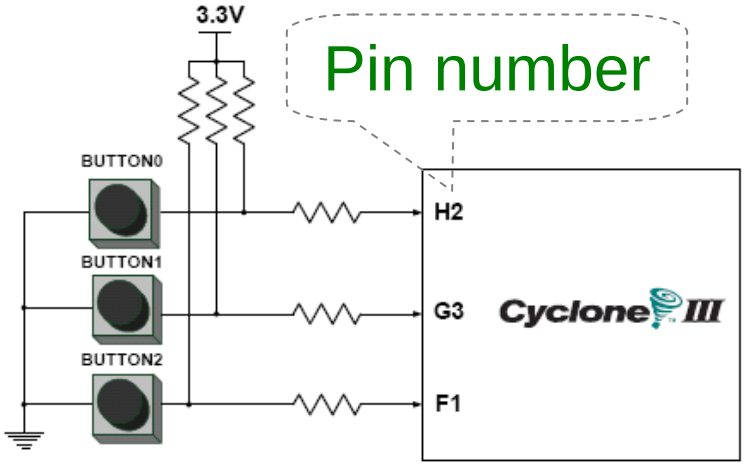
- The steps to program Programmer Object File (.pof) into the EPCS4 device on the DE0 board are as follows:
  - Step 1: Power on DE0 board with SW11 to **PROG** mode and connect it to the host.
  - Step 2: Open Quartus II, and choose **Tools → Programmer**
  - Step 3: Click "**Hardware Setup**" and then Select "**USB Blaster**"
  - Step 4: Click "**Add File**" to select the .pof file in Active Serial Programming mode, and then click "**Start**" to program it. (Remember to select "Add Device →"EPCS4")

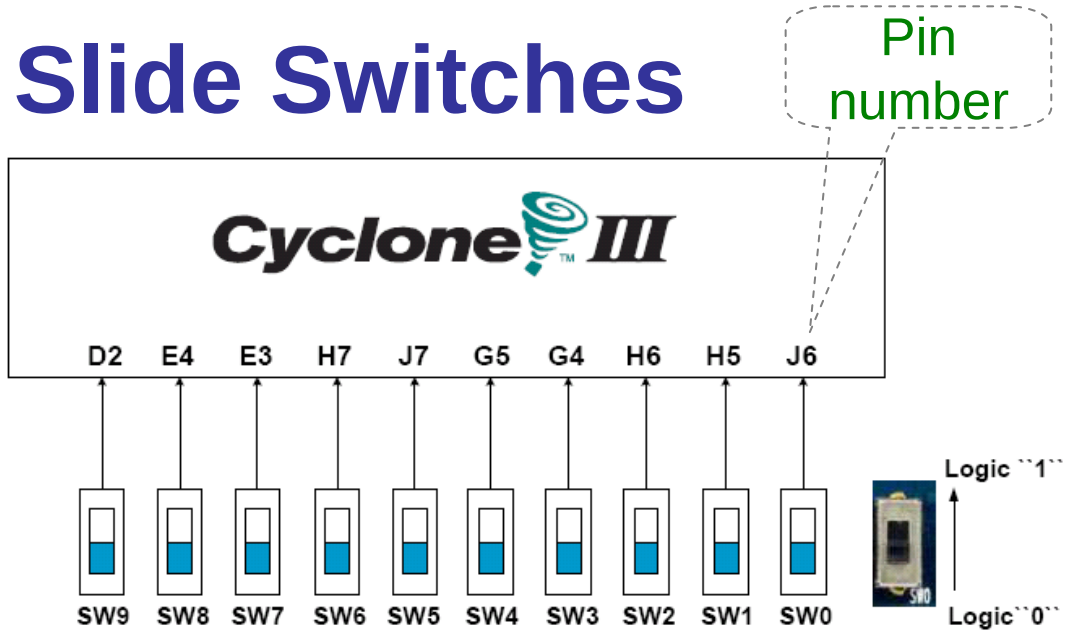






Pushbutton and Slide Switches





3 Pushbutton switches:
Not pressed → Logic High
Pressed → Logic Low

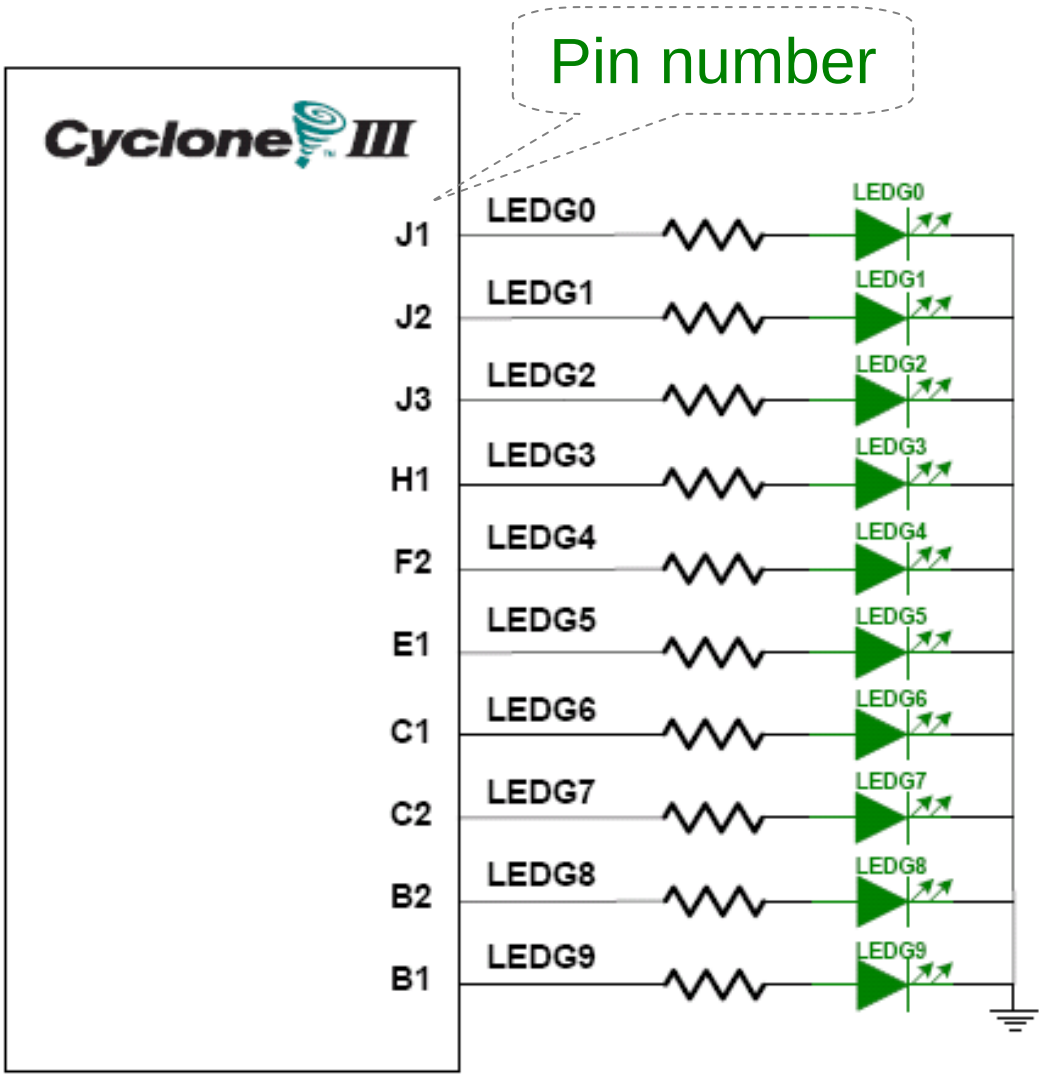
Signal Name	FPGA Pin No.
BUTTON [0]	PIN_ H2
BUTTON [1]	PIN_ G3
BUTTON [2]	PIN_ F1

10 Slide switches (Sliders):
Up → Logic High
Down → Logic

SW[0]	PIN_J6	SW[5]	PIN_J7
SW[1]	PIN_H5	SW[6]	PIN_H7
SW[2]	PIN_H6	SW[7]	PIN_E3
SW[3]	PIN_G4	SW[8]	PIN_E4
SW[4]	PIN_G5	SW[9]	PIN_D2



LEDs

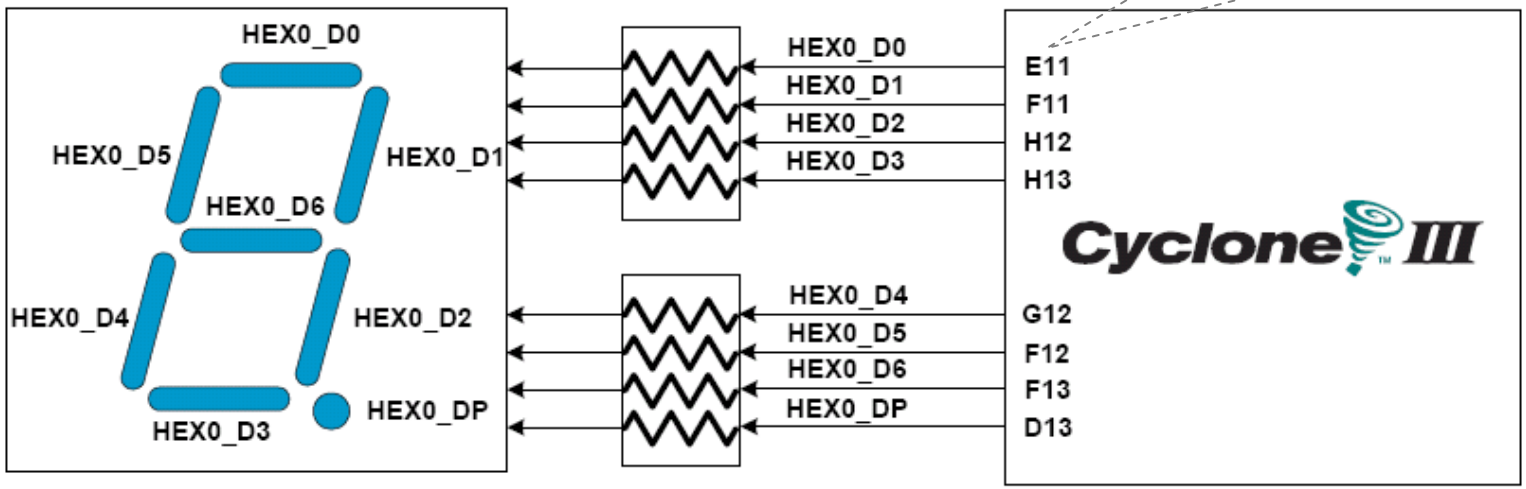


10 LEDs
Opuput high → LED on
Output low → LED off

Signal Name	FPGA Pin No.
LEDG[0]	PIN_J1
LEDG[1]	PIN_J2
LEDG[2]	PIN_J3
LEDG[3]	PIN_H1
LEDG[4]	PIN_F2
LEDG[5]	PIN_E1
LEDG[6]	PIN_C1
LEDG[7]	PIN_C2
LEDG[8]	PIN_B2
LEDG[9]	PIN_B1

7-Segment Displays

Pin number (active-low)



Signal Name	FPGA Pin No.
HEX0_D[0]	PIN_E11
HEX0_D[1]	PIN_F11
HEX0_D[2]	PIN_H12
HEX0_D[3]	PIN_H13
HEX0_D[4]	PIN_G12
HEX0_D[5]	PIN_F12
HEX0_D[6]	PIN_F13
HEX0_DP	PIN_D13

HEX1_D[0]	PIN_A13
HEX1_D[1]	PIN_B13
HEX1_D[2]	PIN_C13
HEX1_D[3]	PIN_A14
HEX1_D[4]	PIN_B14
HEX1_D[5]	PIN_E14
HEX1_D[6]	PIN_A15
HEX1_DP	PIN_B15

HEX2_D[0]	PIN_D15
HEX2_D[1]	PIN_A16
HEX2_D[2]	PIN_B16
HEX2_D[3]	PIN_E15
HEX2_D[4]	PIN_A17
HEX2_D[5]	PIN_B17
HEX2_D[6]	PIN_F14
HEX2_DP	PIN_A18

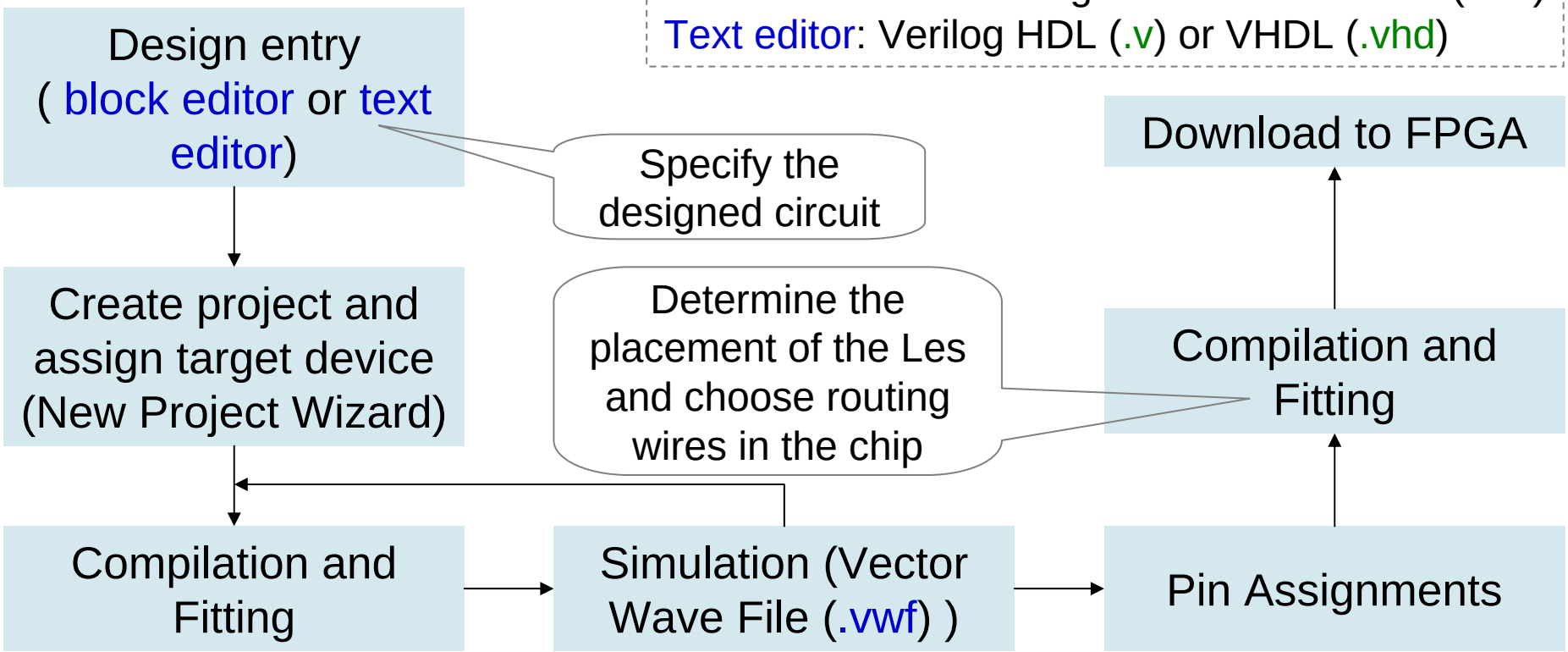
HEX3_D[0]	PIN_B18
HEX3_D[1]	PIN_F15
HEX3_D[2]	PIN_A19
HEX3_D[3]	PIN_B19
HEX3_D[4]	PIN_C19
HEX3_D[5]	PIN_D19
HEX3_D[6]	PIN_G15
HEX3_DP	PIN_G16



Quartus II FPGA Design Software

Simplified Design Flow of Quartus II

Block editor: Block Diagram/Schematic File (.bdf)
Text editor: Verilog HDL (.v) or VHDL (.vhd)

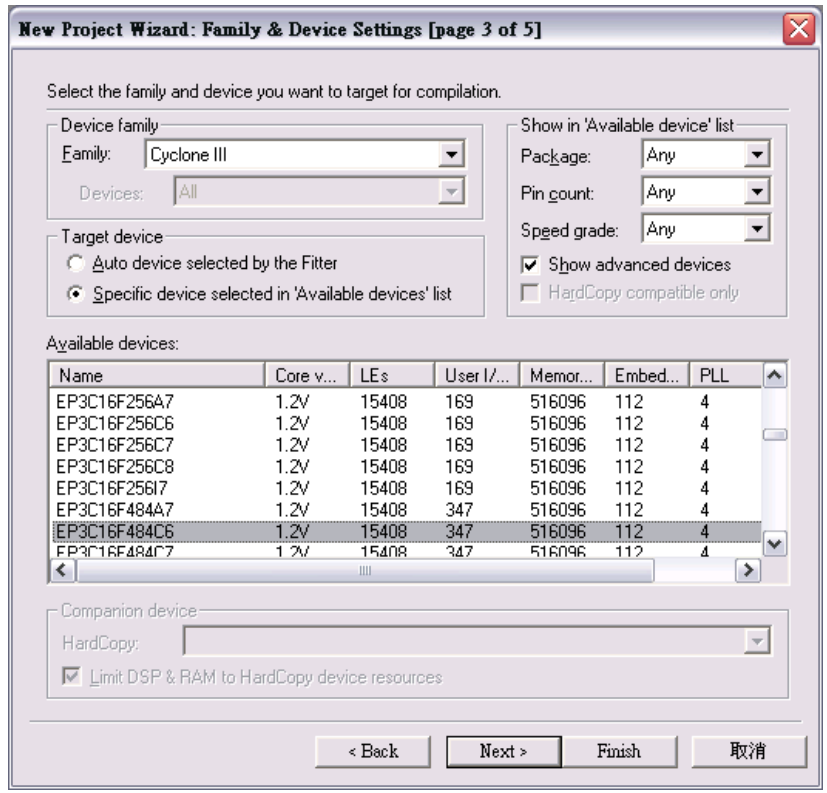


Timing simulation: verify functional correctness and timing issues.
Functional simulation: verify functional correctness without considering timing issues.



Schematic Design with Quartus II

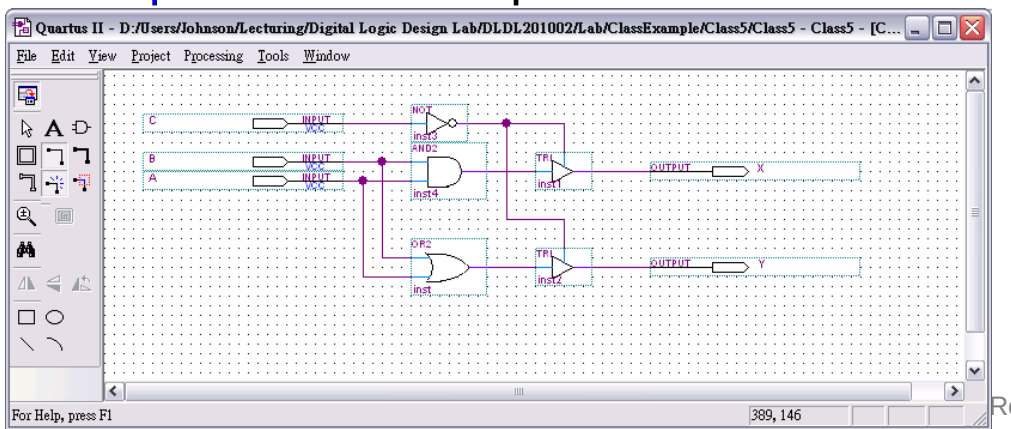
- Example: When the BUTTON0 is pressed, LEDG0 shows the ANDed result of SW0 and SW1 and LEDG1 shows the ORed result of SW0 and SW1.
- Step 1: Start a new project
  - Select File → New Project Wizard
    - Working directory: Class5
    - Project name: Class5
    - Top-level design entry: Class5
  - Family & Device Settings
    - Device family: Cyclone III
    - Available device: EP3C16F484C6
  - EDA Tool Settings
    - Leave it alone at the moment





Schematic Design with Quartus II (Cont.)

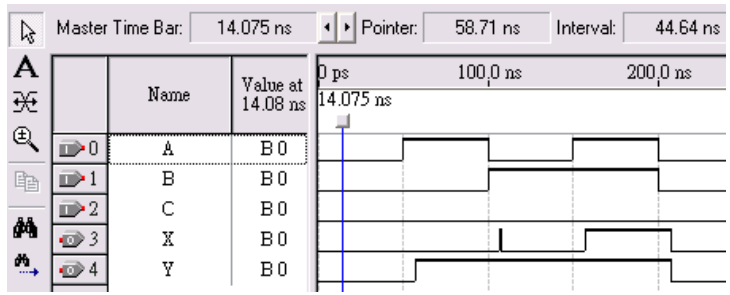
- Step 2: Design entry using the graphic editor
  - Select File → New → Block Diagram/Schematic File (.bdf)
  - Save as "Class5.bsf" (check "Add file to current project")
  - Select "primitives" of "Symbol Tool" to add 
    - Three input pins A, B, and C, two output pins X and Y
    - One AND gate, one OR gate, two tri-state buffers, and one NOT gate.
  - Select "Orthogonal Node Tool" to connect the nodes. 
  - Select "Start Compilation" to compile the circuit





Schematic Design with Quartus II (Cont.)

- Step 3: Simulation with Vector Waveform File (.vwf)
  - Select File → New → Vector Waveform File (.vwf)
  - Save as "Class5.vwf" (check "Add file to current project")
  - Select "Edit → Insert → Insert Node or Bus → Node Finder" to add input/output pins into the simulation.
  - Select "Edit → End Time" and select "Edit → Grid Size" to config the simulation period and count period.
    - A: count value, binary, count every 50ns, multiplied by 1.
    - B: count value, binary, count every 50ns, multiplied by 2.
    - C: forcing high or forcing low.
  - Select "Start Simulation" to simulate the circuit.
  - Functional simulation
    - Select "Assignments → Settings → Simulator Settings" to set "Simulation mode" as Functional.
    - Select "Processing → Generate Functional Simulation Netlist"
    - Select "Start Simulation" to simulate the circuit.





Schematic Design with Quartus II (Cont.)

- Step 3: Simulation with Vector Waveform File (.vwf)
  - Select "Assignments → Device" to configure the board settings.
    - Set Family as Cyclone III and Device as EP316F484C6
    - Select "Device and Pin Options"
      - Select and set "Unsigned Pings" as "As input tri-stated" and
      - Select "Configuration" to set configuration scheme as "Active Serial" and configuration device as "EPCS4"
  - Select "Assignments → Pins" to activate the "Pin Planner".
  - Select "Start Compilation" to compile the circuit with circuit assignment.
  - Select "Tools → Programmer" to download the .soft file to the FPGA board for testing.

	Node Name	Direction	Location
1	A	Input	PIN_J6
2	B	Input	PIN_H5
3	C	Input	PIN_H2
4	X	Output	PIN_J1
5	Y	Output	PIN_J2



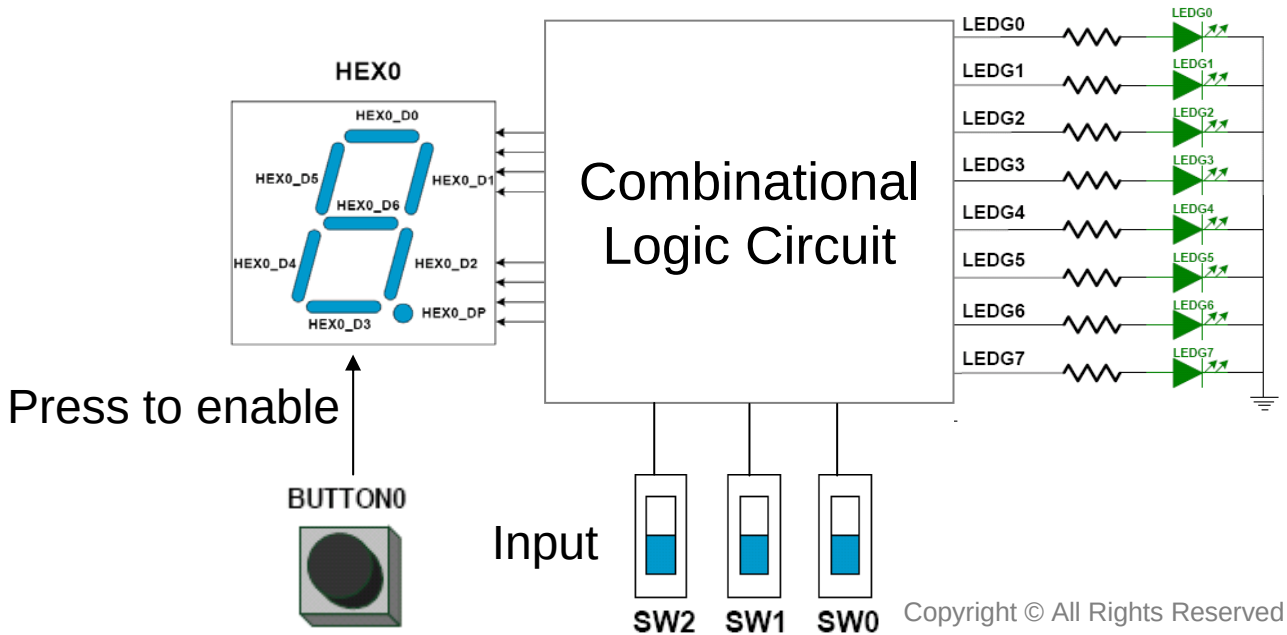
Lab 5

- Part 1 - Simulation
  - Use the block editor (Block Diagram/Schematic File: .bdf) to design a NAND gate with one output pin F and two input ping A and B. Then use Vector Waveform File (.vwf) to simulate the results.
    - A: count value, binary, simulation period=4us, advanced by 1 every 100ns
    - B: count value, binary, simulation period=4us, advanced by 1 every 200ns
- Part 2 - Transferring a Design to a Target FPGA
  - Use three slides (SW2-SW0) as the binary input value.
    - The corresponding LED (LEDG0-7) is on when selected by the binary input. Other LEDs are off. E.g., 100 (SW2-SW0) lights LEDG4.
    - The first 7-segment LED (HEX0) shows the decimal value of the binary input when the first pushbutton (BUTTON0) is pressed. Otherwise, HEX0 is off. E.g., When BUTTON0 is pressed and the binary input is 101 (SW2-SW0), HEX0 shows 5.



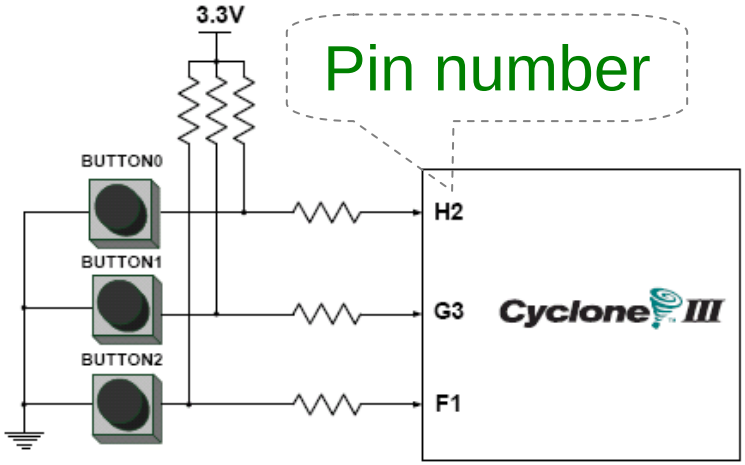
Report

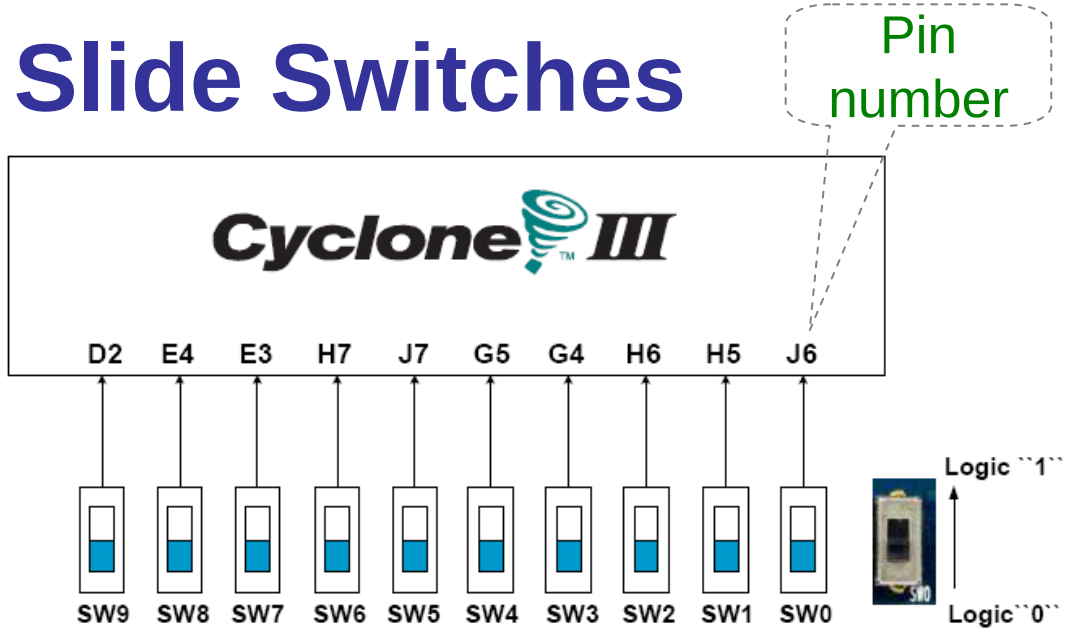
- Part 1 – Simulation
  - Explain the simulation result. (說明實驗結果的原因)
  - Write down what you learned from this experiment (實驗心得)
- Part 2 - Transferring a Design to a Target FPGA
  - Explain the process of the circuit design (說明電路設計的過程)
  - Write down what you learned from this experiment (實驗心得)





Pushbutton and Slide Switches





3 Pushbutton switches:
Not pressed → Logic High
Pressed → Logic Low

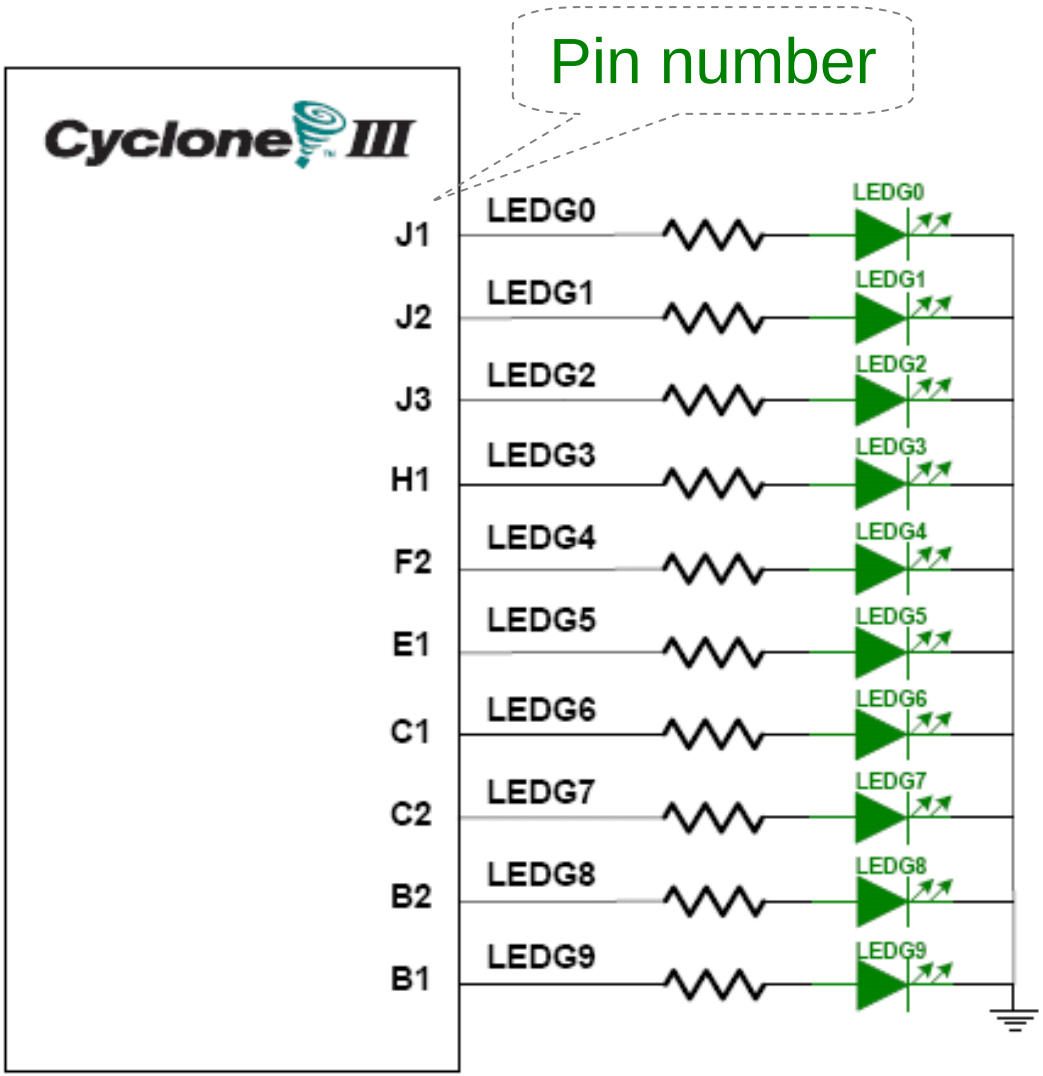
Signal Name	FPGA Pin No.
BUTTON [0]	PIN_ H2
BUTTON [1]	PIN_ G3
BUTTON [2]	PIN_ F1

10 Slide switches (Sliders):
Up → Logic High
Down → Logic

SW[0]	PIN_J6	SW[5]	PIN_J7
SW[1]	PIN_H5	SW[6]	PIN_H7
SW[2]	PIN_H6	SW[7]	PIN_E3
SW[3]	PIN_G4	SW[8]	PIN_E4
SW[4]	PIN_G5	SW[9]	PIN_D2



LEDs

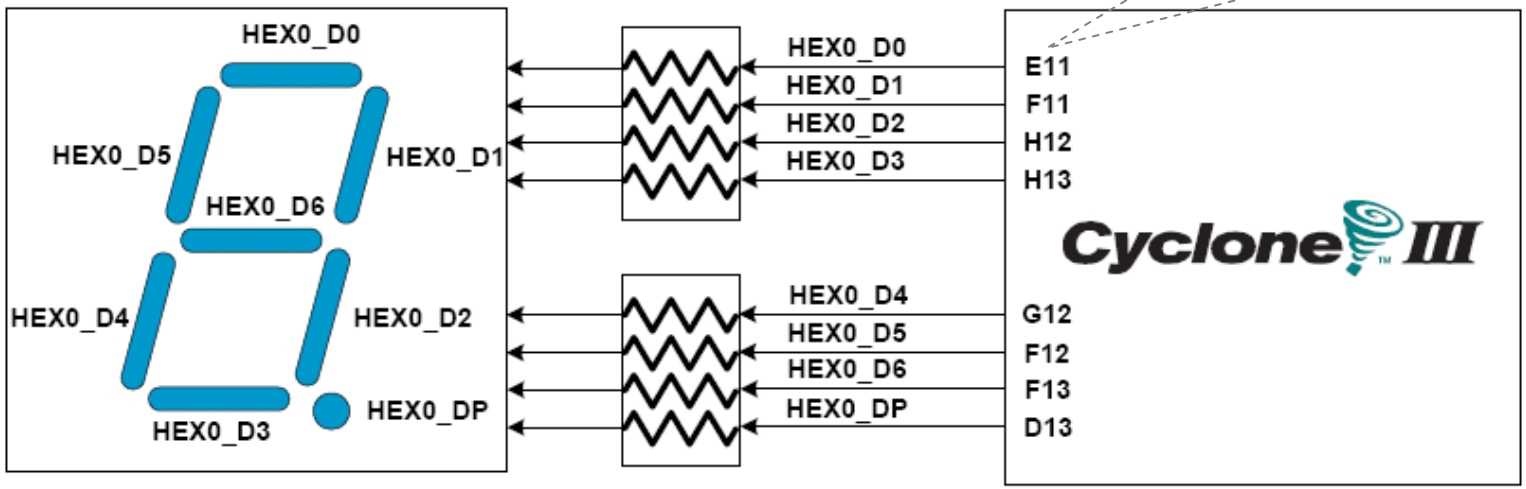


10 LEDs
Opuput high → LED on
Output low → LED off

Signal Name	FPGA Pin No.
LEDG[0]	PIN_J1
LEDG[1]	PIN_J2
LEDG[2]	PIN_J3
LEDG[3]	PIN_H1
LEDG[4]	PIN_F2
LEDG[5]	PIN_E1
LEDG[6]	PIN_C1
LEDG[7]	PIN_C2
LEDG[8]	PIN_B2
LEDG[9]	PIN_B1

7-Segment Displays

Pin number (active-low)



Signal Name	FPGA Pin No.
HEX0_D[0]	PIN_E11
HEX0_D[1]	PIN_F11
HEX0_D[2]	PIN_H12
HEX0_D[3]	PIN_H13
HEX0_D[4]	PIN_G12
HEX0_D[5]	PIN_F12
HEX0_D[6]	PIN_F13
HEX0_DP	PIN_D13

HEX1_D[0]	PIN_A13
HEX1_D[1]	PIN_B13
HEX1_D[2]	PIN_C13
HEX1_D[3]	PIN_A14
HEX1_D[4]	PIN_B14
HEX1_D[5]	PIN_E14
HEX1_D[6]	PIN_A15
HEX1_DP	PIN_B15

HEX2_D[0]	PIN_D15
HEX2_D[1]	PIN_A16
HEX2_D[2]	PIN_B16
HEX2_D[3]	PIN_E15
HEX2_D[4]	PIN_A17
HEX2_D[5]	PIN_B17
HEX2_D[6]	PIN_F14
HEX2_DP	PIN_A18

HEX3_D[0]	PIN_B18
HEX3_D[1]	PIN_F15
HEX3_D[2]	PIN_A19
HEX3_D[3]	PIN_B19
HEX3_D[4]	PIN_C19
HEX3_D[5]	PIN_D19
HEX3_D[6]	PIN_G15
HEX3_DP	PIN_G16