



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





Terasic DE0 Field Programmable Gate Array (FPGA) Development Board





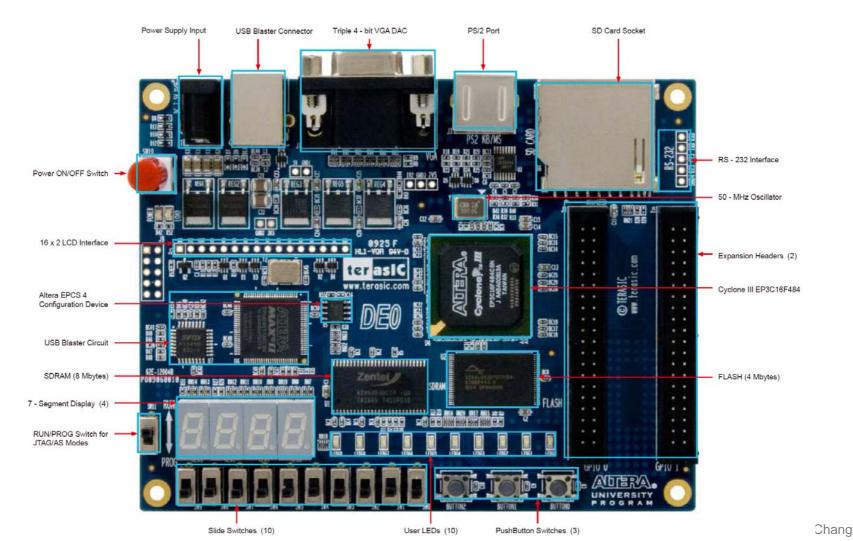


March 29, 2010





Layout and Components of DE0



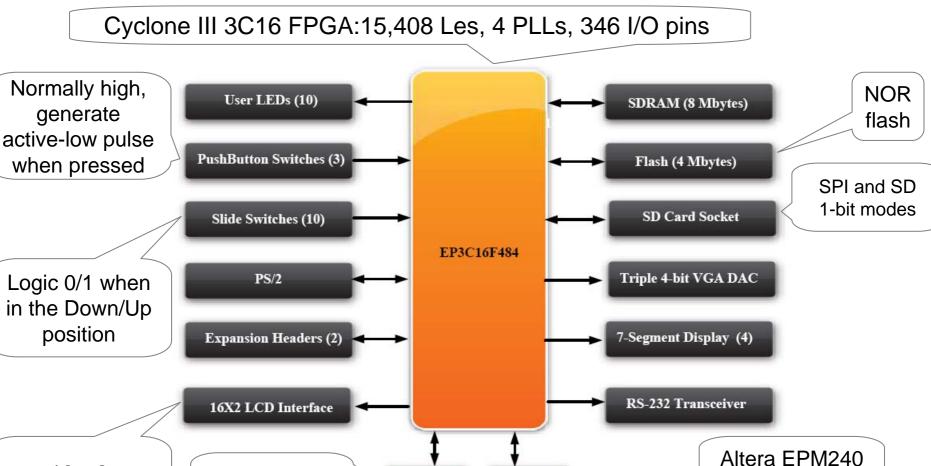


16 x 2

Character LCD



Block Diagram of the DE0 Board



EPCS4

Config

Device

USB

Blaster

EEPROM

/ Yuan-Hao Chang

CPLD





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:





March 29, 2010

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)

尋找新増硬體積靈	
	歡 迎使用尋找新增硬體精 霊
	Windows 將會搜尋您的電腦、硬體完裝 CD 或 Windows Update 網站 (您尤許的話) 來尋找目前的以及已更新的軟 但仍是我們的認識為 聖明
	Windows 是否可以連線到 Windows Update 尋找軟體?
	 ○是,只有現在(Y) ○是,現在以及每次我連接了一個裝置時(E) ○不,現在不要(I)
	諸技 [下一步] 繼續。
	<上一步(B) 下一步(D) 取消 取消 (1) (下一步(C) (1) (1) (1) (1) (1) (1) (1) (1) (1) (1

找新增硬售	
諸選擇您	的搜尋和安裝運項。
②在這	些位置中搜尋最好的驅動程式(3)
使用 裝找	于列核取方塊來限制或擴充包括本機路徑和可卸除式媒體的預設搜尋,將安 到的最佳驅動程式。
] 搜尋可卸除式媒體 (軟碟,CD-ROM)(M)
5	2.搜尋時包括這個位置(0):
	C.'altera'91'quartus'drivers'usb-blaster 🛛 😧 瀏覽 🕲
〇不要	·搜尋,我將選擇要安裝的驅動程式(D)
選擇 式最	這個選項來從著單中選取裝置驅動程式。Windows不保證您所選取的驅動程 符合您的硬體。
	<上一步 田) 下一步 田) 下一步 田) 下



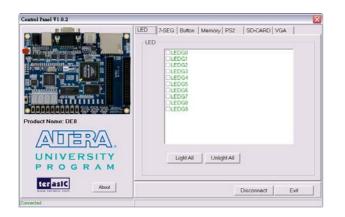
Copyright © All Rights Reserved by Yuan-Hao Chang





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.



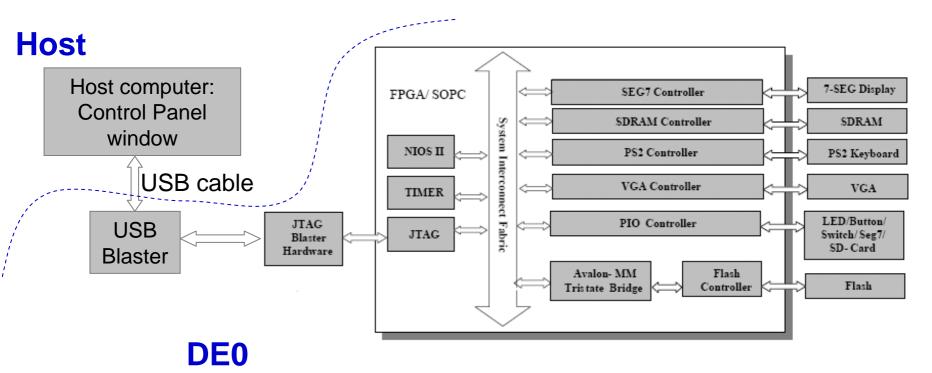




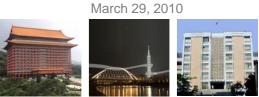


DE0 Control Panel (Cont.)

- 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.







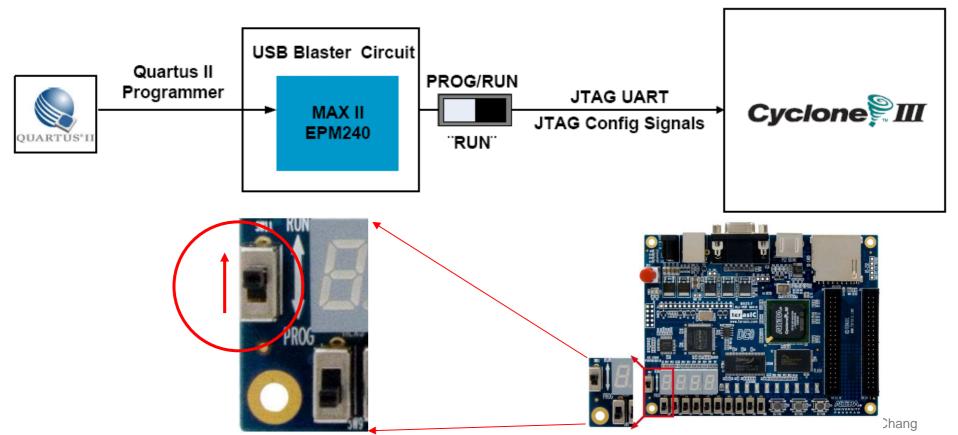
Configuring the Cyclone III FPGA

- 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.

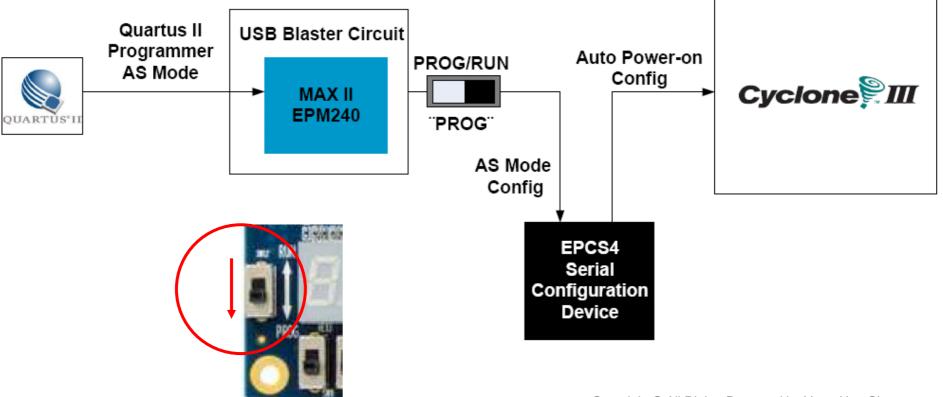
Conclusion Project Naviganenth Processon Totals Totals Audylin Tool Project Navigator Project Navigator Project Navigator Project Navigator Conclusion Research Office Project Navigator Project Navigator Project Navigator Conclusion Research Office Project Navigator Project Navigator Project Navigator Conclusion Research Office Project Navigator Project Navigator Project Navigator Conclusion Research Office Project Navigator Project Navigator Project Navigator Conclusion Research Office Project Navigator Project Navigator Project Navigator Totals Office Project Navigator Project Navigator Project Navigator Project Navigator Design Audyling Navigator Project Navigator Project Navigator Project Navigator Design Audyling Navigator Project Navigator Project Navigator Project Navigator Project Navigator Design Audyling Navigator Design Audyling Navigator Project Navigator Project Navigator Project Navigator Project Navigator Design Audyling Navigator Design Audyling Navigator Project Navigator Project Navigator Proje	Quartus II - [Chain1.cdf] Eva Edit Processing Tools Window Audo Detect Audo Detect Currently selected hardware: USB-Blaster [USB-0]	Quartus II - [Chain1.edf] Eile Edit Processing Tools Window
Teld Compliation Prior Compliation Teld Compliation Image: Compliation Compliation Image: Compliation Image: Compliation Image:	Select Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window. Currently selected hardware: USB-Blaster [USB-0] Velece	₩ Stop





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.)

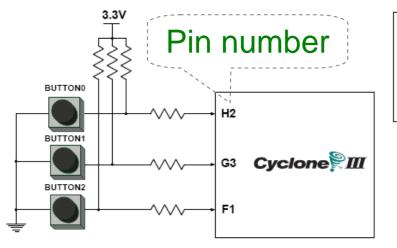
- The steps to program Programmer Object File (.pof) into the EPCS4 device on the DE0 board are as follows:
 - Step 1: Power on DE0 board with SW11 to **PROG** mode and connect it to the host.
 - − Step 2: Open Quartus II, and choose Tools → Programmer
 - Step 3: Click "Hardware Setup" and then Select "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")

🗘 Qeartes II	🖺 Quartus II - [Chain1.cdf]								
Eile Edit Yinv Eroject Arriguments Processing Tools Window Help	File Edit Processing Tools Window		🕒 Quartus II - [Chai	in1 cdfl					
Run EDA Tanua Andrea Toda	Ardware Setup USB-Blaster [USB-0]		File Edit Processing					كالك	
Project Navigator	Mode: 1. JTAG	•		USB-Blaster [USB-0]					
Compilation Hierarchy O TaneQuest Taning Analyter	Progress: Hardware Setup			JTAG					-
Advisors	Enable real-time Hardware Settings JTAG Settings		Progress:			0%			
Hierarchy 🗈 Files 🖉 Design Units 🔹 Design Partition Planner Free Online	M Start	-1 i	Enable real-time ISP	to allow background prog	ramming (for MAX II de	vices)			
Tasks Netterit jiewent	Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.		🔊 Start 🛛 🖡	ile	Device	Checksum	Usercode	Program/ Configure	Veri
Task (2) Control Editor	Auto Detect Currently selected hardware: USB-Blaster [USB-0]		🖬 Stop						
Compile Design Design			Auto Detect						
Cumentation	Add Hardware Server Port Add Hardware		X Delete						
Type Hessage You Magging and Physical Manager	Add File USB-Blaster Local USB-0 Remove Hardware		Add File						
elli 20PC genker Tel Segge	😰 Change File		Change File						
	📑 Save File		Save File						
System / Processing / Exits Info / Info / Continues Continues Continues Continues Continues Continues Locate Locate Locate	Add Device		Add Device						
Open a Programmer window									
		ī							
	Down		Down	•					>
	For Help, press F1	///p	⊖ For Help, press F1						11.



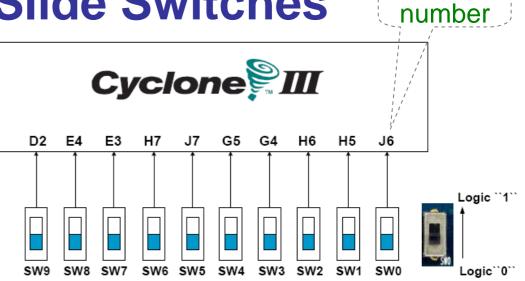


Pushbutton and Slide Switches



3 Pushbutton switches: Not pressed \rightarrow Logic High Pressed \rightarrow Logic Low

Signal Name	FPGA Pin No.
BUTTON [0]	PIN_H2
BUTTON [1]	PIN_G3
BUTTON [2]	PIN_F1



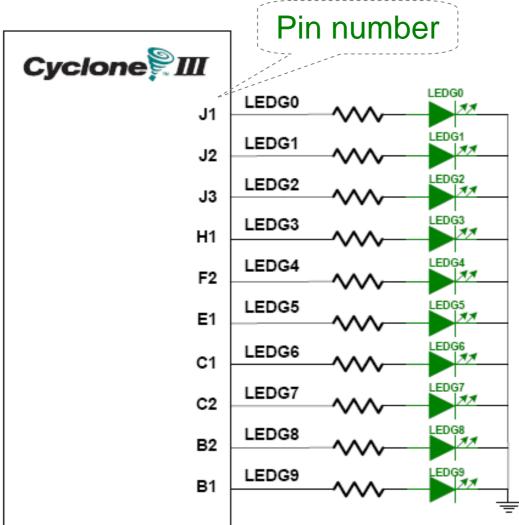
10 Slide switches (Sliders): Up \rightarrow Logic High Down \rightarrow Logic

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

Pin



LEDs





10 LEDs Opuput high \rightarrow LED on Output low \rightarrow 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

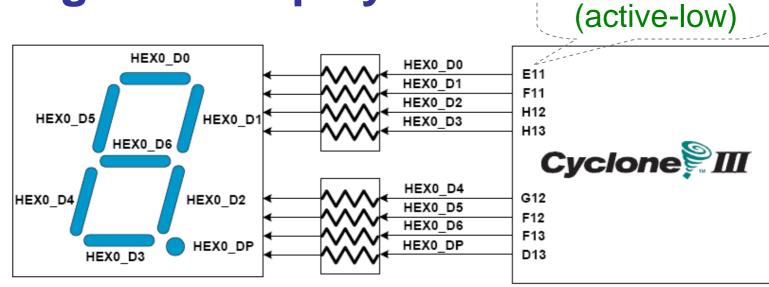
March 29, 2010





Pin number

7-Segment Displays



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

	L		L
HEX2_D[0]	PIN_D15	HEX3_D[0]	PIN_B18
HEX2_D[1]	PIN_A16	HEX3_D[1]	PIN_F15
HEX2_D[2]	PIN_B16	HEX3_D[2]	PIN_A19
HEX2_D[3]	PIN_E15	HEX3_D[3]	PIN_B19
HEX2_D[4]	PIN_A17	HEX3_D[4]	PIN_C19
HEX2_D[5]	PIN_B17	HEX3_D[5]	PIN_D19
HEX2_D[6]	PIN_F14	HEX3_D[6]	PIN_G15
HEX2_DP	PIN_A18 Rights	HEX3_DP	PIN_G16





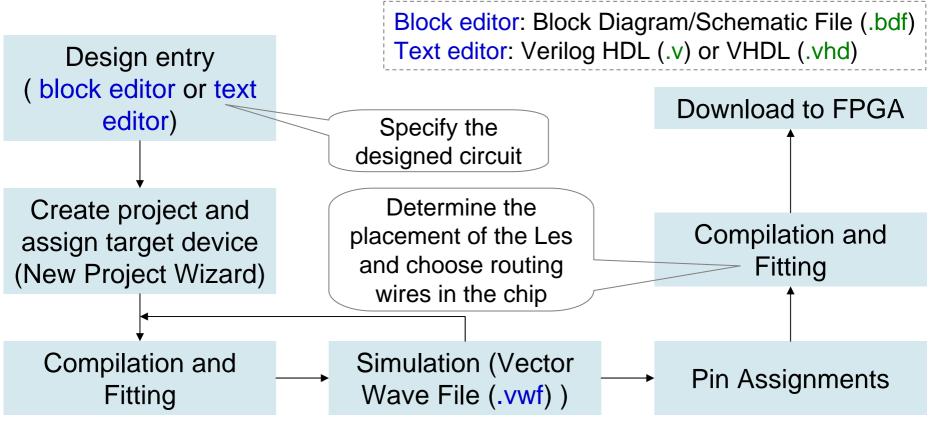
Quartus II FPGA Design Software







Simplified Design Flow of Quartus II



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



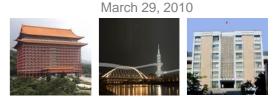


Schematic Design with Quartus II

- 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

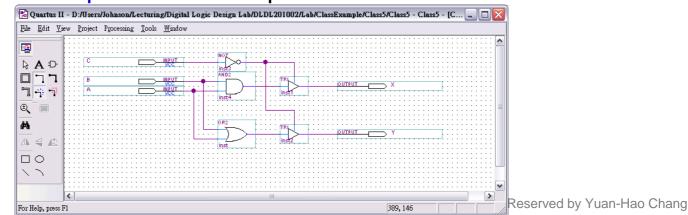
-Device family				- Show in 'A'	vailable de [.]	vice' list—
Eamily: Cyclone II	1		-	Package:	Any	
,						
Devices: All			<u> </u>	Pin <u>c</u> ount:	Any	
Target device				Speed grad	de: Any	•
C Auto device selec	ted by the Fitter			Show a	advanced o	devices
 Specific device se 	•		list	HardCo		
∖ <u>v</u> ailable devices:						
Name	Core v	LEs	User I/	. Memor	Embed	PLL
EP3C16F256A7	1.2V	15408	169	516096	112	4
EP3C16F256C6	1.2V	15408	169	516096	112	4
	4 767	15408	169	516096	112	4
EP3C16F256C7	1.2V	10400				
EP3C16F256C7 EP3C16F256C8	1.2V 1.2V	15408	169	516096	112	4
				516096 516096		4 4
EP3C16F256C8	1.2V	15408	169		112	4 4 4
EP3C16F256C8 EP3C16F256I7 EP3C16F484A7 EP3C16F484C6	1.2V 1.2V 1.2V 1.2V	15408 15408 15408 15408	169 169 347 347	516096 516096 516096	112 112 112 112 112	4
EP3C16F256C8 EP3C16F256I7 EP3C16F484A7 EP3C16F484C6 EP3C16F484C7	1.2V 1.2V 1.2V	15408 15408 15408 15408 15408	169 169 347	516096 516096	112 112 112	4 4
EP3C16F256C8 EP3C16F256I7 EP3C16F484A7 EP3C16F484C6	1.2V 1.2V 1.2V 1.2V	15408 15408 15408 15408	169 169 347 347	516096 516096 516096	112 112 112 112 112	4
EP3C16F256C8 EP3C16F256I7 EP3C16F484A7 EP3C16F484C6 EP3C16F484C6 EP3C16F484C7 <	1.2V 1.2V 1.2V 1.2V	15408 15408 15408 15408 15408	169 169 347 347	516096 516096 516096	112 112 112 112 112	4 4
EP3C16F256C8 EP3C16F256I7 EP3C16F484A7 EP3C16F484C6 EP3C16F484C7	1.2V 1.2V 1.2V 1.2V	15408 15408 15408 15408 15408	169 169 347 347	516096 516096 516096	112 112 112 112 112	4 4





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

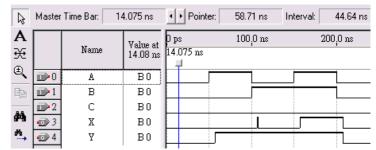




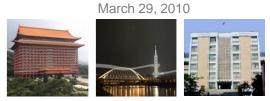


Schematic Design with Quartus II (Cont.)

- 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. xc
 - 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 \rightarrow 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 \rightarrow 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	Ble Efst Processi	/Owrn/Johanna/Lectur ng Ioch <u>Window</u> 5. USB Blanter (USB 0)		iiga Lab/DLDL2	tode JTAG	lan5/Clan5 -	Propess			
1	Ê	A	Input	PIN_J6		SP to allow background p				Verily Blank Check	-	Security Erace	1	
2	Ê	В	Input	PIN_H5	Auto Delect	window and								
3	Ê	С	Input	PIN_H2	Add File.									
4	þ	Х	Output	PIN_J1	Add Device									
5	þ	Y	Output	PIN_J2	For Help, press Pl									



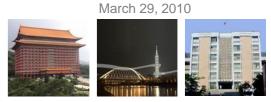
Lab 5

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

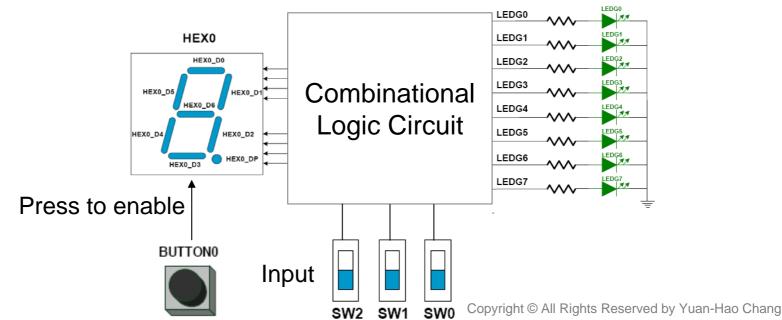
• Part 2 - Transferring a Design to a Target FPGA

- 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

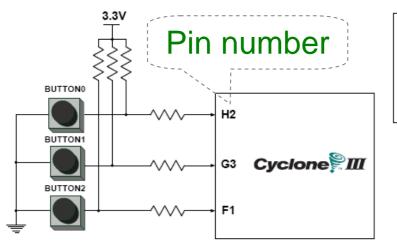
- Part 1 Simulation
 - Explain the simulation result. (說明實驗結果的原因)
 - Write down what you learned from this experiment (實驗心得)
- Part 2 Transferring a Design to a Target FPGA
 - Explain the process of the circuit design (說明電路設計的過程)
 - Write down what you learned from this experiment (實驗心得)





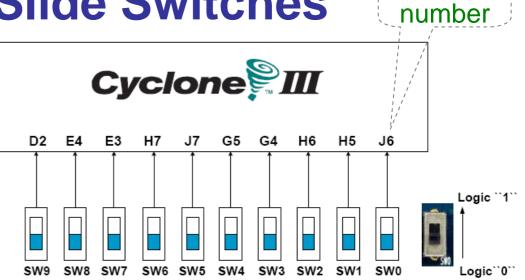


Pushbutton and Slide Switches



3 Pushbutton switches: Not pressed \rightarrow Logic High Pressed \rightarrow Logic Low

Signal Name	FPGA Pin No.				
BUTTON [0]	PIN_H2				
BUTTON [1]	PIN_G3				
BUTTON [2]	PIN_F1				



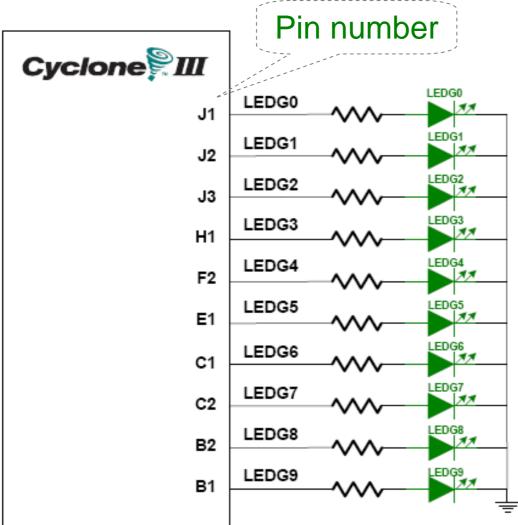
10 Slide switches (Sliders): Up \rightarrow Logic High Down \rightarrow Logic

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

Pin



LEDs





10 LEDs Opuput high \rightarrow LED on Output low \rightarrow 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

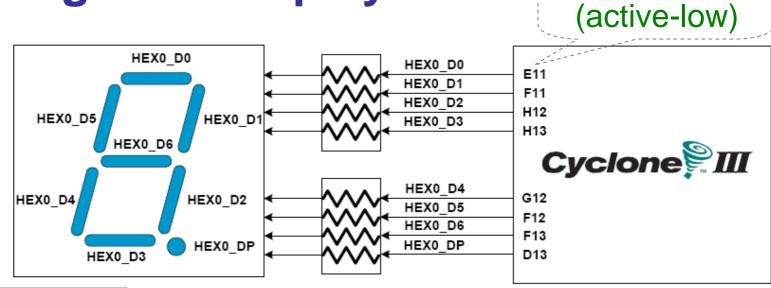
March 29, 2010





Pin number

7-Segment Displays



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

			I
HEX2_D[0]	PIN_D15	HEX3_D[0]	PIN_B18
HEX2_D[1]	PIN_A16	HEX3_D[1]	PIN_F15
HEX2_D[2]	PIN_B16	HEX3_D[2]	PIN_A19
HEX2_D[3]	PIN_E15	HEX3_D[3]	PIN_B19
HEX2_D[4]	PIN_A17	HEX3_D[4]	PIN_C19
HEX2_D[5]	PIN_B17	HEX3_D[5]	PIN_D19
HEX2_D[6]	PIN_F14	HEX3_D[6]	PIN_G15
HEX2_DP	PIN_A18 Rights	HEX3_DP	PIN_G16