

---

# CONTENTS

1	Introduction .....	3
2	Items in QB-FPGA200EP2C Starter Kit.....	4
3	Component Description of QB-FPGA200EP2C Starter Kit .....	6
	Cyclone II EP2C5Q in the -8 speed grade FPGA Device .....	6
	QB-EP2C208-BK Main Board .....	8
	QB-ALTERA-AT ByteBlasterII.....	8
4	QB-EP2C208-BK Main Board.....	9
	Board Installation .....	10
	Kyes.....	11
	DIP switch.....	12
	FND(4 digit numeric display) .....	12
	LEDs .....	14
	Power Block .....	14
	Extension Port (EXTA, EXTB, EXTC) .....	14
	Serial Communication Port (USART).....	17
	Character LCD.....	18
	LCD Module initialization and Display Sequence .....	27

---

5	Port MAP .....	29
6	Tutorial.....	32
6.1	Design Flow on Quartus II .....	33
6.2	A simple schematic design example .....	35
6.2.1	Creating Your Own Project .....	36
6.2.2	Setting Unused pins .....	41
6.2.3	Create a New Schematic Design File .....	43
6.2.4	Pin Assignment.....	49
6.2.5	Simulation .....	54
6.2.6	Programming FPGA.....	60
6.3	A VHDL design example .....	65
6.3.1	Creating Your own Project.....	66
6.3.2	Setting Unused pins .....	71
6.3.3	Create a New VHDL Design File .....	73
6.3.4	Pin Assignment.....	77
6.3.5	Simulation .....	82
6.3.6	Programming EPCS1 Configuration ROM.....	83
7	Useful Links .....	91
8	Appendix .....	92

---

# 1 Introduction

Thank you for purchasing QuantumBase' Level-up series QB-FPGA200EP2C Starter Kit. The FPGA kit has every thing you need to start designing and learning with the powerful features of Altera® Cyclone II™ FPGA devices. The QB-FPGA200EP2C Starter Kit allows you to implement your own designed digital circuit easily with minimal design overhead. Whenever you want to learn about FPGA design, or have a specific design implementation to complete, the starter kit will jump-start of your own Cyclone II™ efforts. The kit provides an Altera's Byte Blaster II compatible QB-Altera-AT programmer, which supports Altera's active serial devices and JTAG devices.

Altera's Quartus II is a state of the art software tool which allows coding, compilation and simulation of complex digital circuit designs in a programmable logic environment, and the recent free version of Quartus II web edition can be downloaded from Altera's website <http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html>.

The SignalTap II Logic Analyzer, available with TALKBACK enabled, a software module in Quartus II, is a powerful system-level debugging tool that captures and displays real-time signal behavior, allowing you to observe interactions between hardware and software in system designs. The Quartus II software allows you to select the signals to capture, when signal capture starts, and how many data samples to capture. You can use a QB-Altera-AT programmer we provide with to employ full features of the SignalTap II Logic Analyzer. *The Quartus II Web Edition is free for everyone regardless of its use. All the user has to do is enjoy it.*

With Level-up QB-FPGA200EP2C Starter Kit, you can easily learn and enjoy designing, implementing, and system level debugging of your own digital circuit

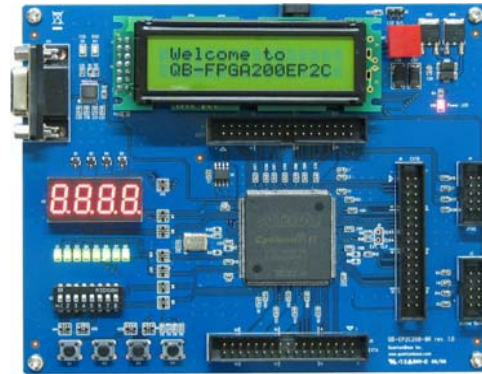
---

## 2 Items in QB-FPGA200EP2C Starter Kit

- ✓ QB-EP2C208BK main board
- ✓ QB-Altera-AT ByteBlaster II compatible board
- ✓ Male to male 25 pin parallel cable
- ✓ Male to female 9 pin serial cable
- ✓ DVD (Datasheets, Manual, Tutorial, Quartus II 8.1 Web edition )

(Note: Please keep the QB-EP2C208-BK and QB-Altera-AT board away from conducting materials while they are powered.)

(Note: External 9VDC, 2.1mm coaxial plug, **center positive** and >0.45A output current adaptor should be used for this kit.)



(a)



(b)

Fig 2.1 (a)QB-EP2C208-BK Main board (b)QB-ALTERA-AT ByteBlasterII

---

(Warning! Please reserve **all unused pins as inputs, tri-stated** by select this option on the **Unused Pins** tab of the **Device & Pin Options** dialogue box, in order to prevent the circuit board from being damaged by unwanted short current between the DIP switch and unused FPGA pins)

(WARNING! When the Extension Ports are **not** used in your design, please **DO NOT touch them with any conducting materials**. Reserving **all unused pins as inputs, tri-stated** can be the good choice to avoid damages by unwanted short current. )

Please refer to the tutorial chapter for reserving unused pins as inputs

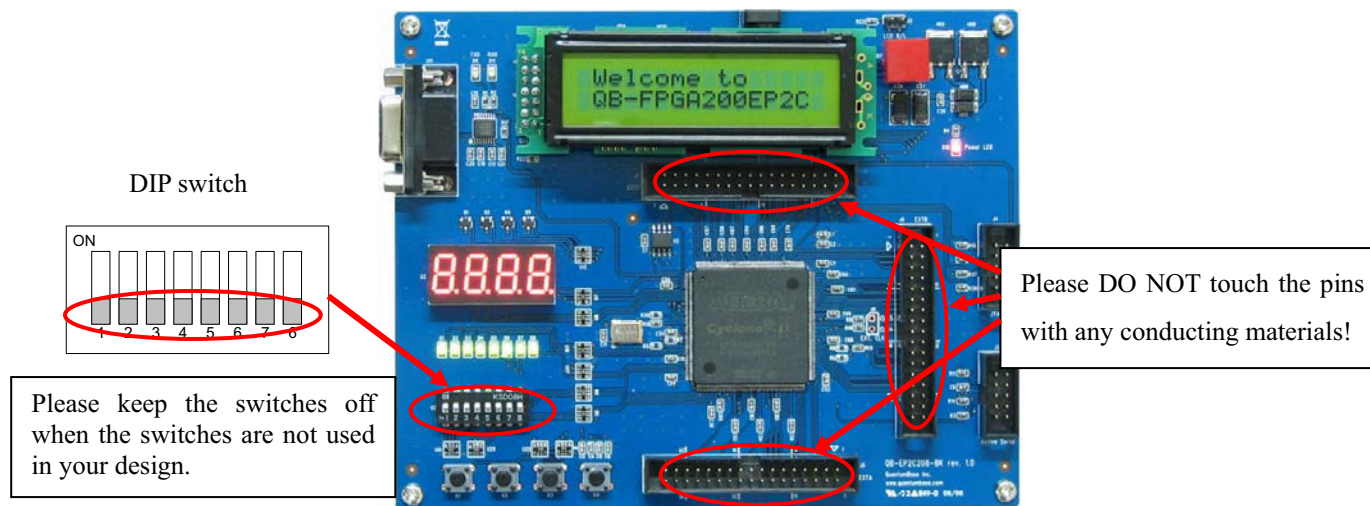


Fig 2.2 Location of the extension port and the DIP switch.

---

### 3 Component Description of QB-FPGA200EP2C Starter Kit

#### Cyclone II EP2C5Q in the -8 speed grade FPGA Device

Cyclone EP2C3Q FPGA device contain a two-dimensional row- and column-based architecture to implement custom logic. Column and row interconnects of varying speeds provide signal interconnects between LABs and embedded memory blocks. The logic array consists of LABs, with 16 LEs in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device. EP2C5Q FPGA device has 4,608 LEs. M4K RAM blocks are true dual-port memory blocks with 4K bits of memory plus parity (512 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 36-bits wide. These blocks are grouped into columns across the device in between certain LABs. The EP2C5Q device provides a global clock network and two PLLs. The global clock network consists of eight global clock lines that drive throughout the entire device. The global clock network can provide clocks for all resources within the device, such as IOEs, LEs, and memory blocks. The global clock lines can also be used for control signals. The Cyclone II PLL provides general-purpose clocking with clock multiplication and phase shifting as well as external outputs for high-speed differential I/O support.

Table 3.1 Cyclone II EP2C5Q Device Features

<b>Logic Elements ( LEs )</b>	<b>4,608</b>	<b>Phase Lock Loop Count ( PLL )</b>	<b>2</b>
<b>M4K RAM Blocks ( 4608 bits/block )</b>	<b>26</b>	<b>Maximum user I/O</b>	<b>158</b>
<b>Total RAM Bits</b>	<b>119,808</b>		



---

### **QB-EP2C208-BK Main Board**

	Specification	Description
FPGA Device	EP2C5Q208C8 (default)	Altera Cyclone II FPGA Device ( upgradable to EP2C8Q208C8)
Configuration	EPCS1 (default)	Altera Configuration ROM(Active serial mode) (upgradable to EPCS4)
Clock	50MHz Main Clock	50MHz external clock is supplied to the CLK2 port of the FPGA device
LCD	16 x 2 Character LCD	Detachable character LCD
LED	8 LEDs	8 LEDs
FND	4 Digits	4 Digit Numeric Displays
Keys	4 Tactile switches	User defined Keys
DIP Switch	8 bit DIP Switches	User defined switches
Extension I/O	34pin Extension Port	90 user IO, 6 of GNDs, 6 of 3.3V outputs
Power	5V,3.3V,1.5V DC Power	---

### **QB-ALTERA-AT ByteBlasterII**

Altera's Byte Blaster II compatible QB-Altera-AT programmer supports Altera's active serial devices and JTAG devices such as Cyclone, Cyclone II, Stratix, Stratix II, EPSCX, and EPCX devices.

## 4 QB-EP2C208-BK Main Board

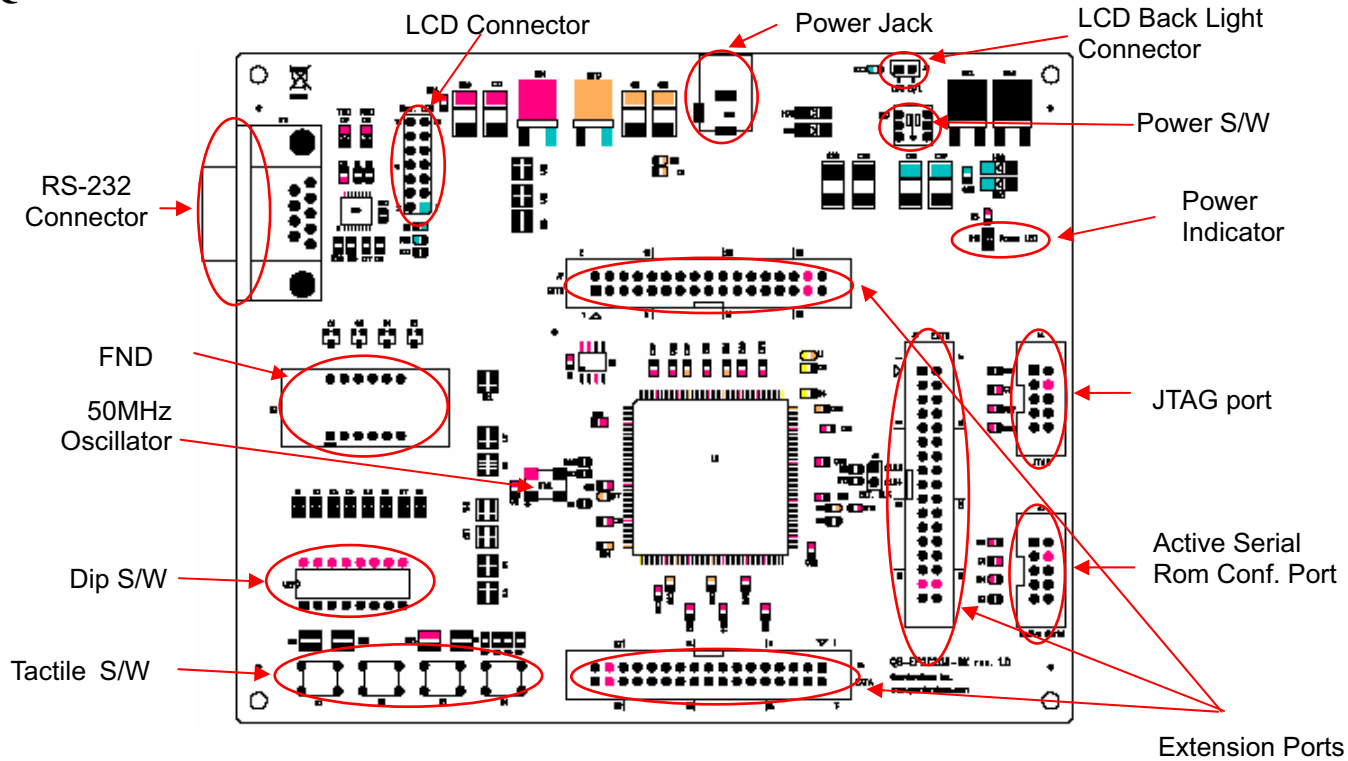


Fig 4.1 QB- EP2C208-BK Baord Layout

## Board Installation

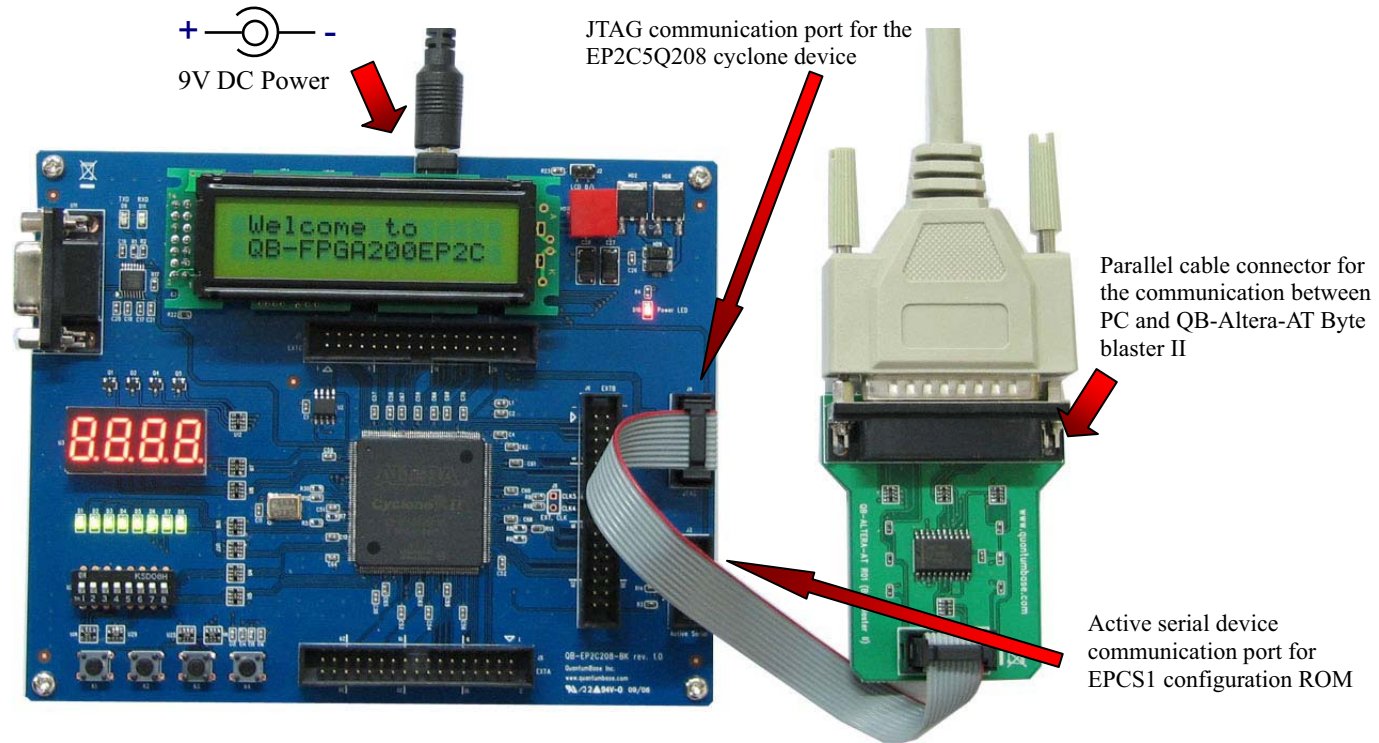


Fig 4.2 Board installation for programming the cyclone device directly through JTAG port

---

## Keys

4 tactile switches for keying in are installed with a 10 kohm pull up resistor as shown Fig 4.3. Thus the default values at the FPGA pins connected to the keys are “Logic High.” Pressing a key down changes the value into “Logic Low.”

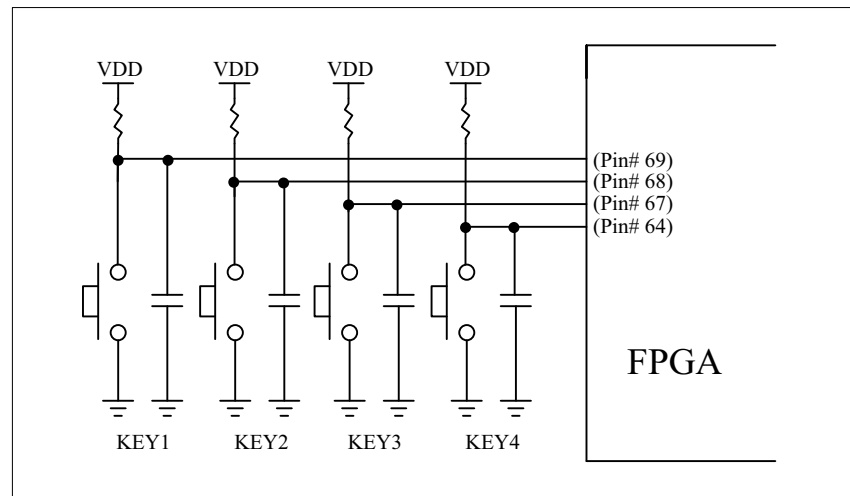


Fig 4.3 Tactile Switches

---

## DIP switch

Another keying-in device installed is an 8 pole DIP switch, which would be helpful to set fixed parameter of a designed digital circuit. When the switch is on, the FPGA pins connected shows it value as “Logic high”.

(Warning! When the DIP switch are **not** used in your design, please **keep all the 8 switches off**, or reserve **all unused pins as inputs, tri-stated** by select this option on the **Unused Pins** tab of the **Device & Pin Options** dialogue box, in order to prevent the circuit board from being damaged by unwanted short current between the DIP switch and unused FPGA pins)

## FND(4 digit numeric display)

Fig. 4.4 shows internal component connection of the installed FND of 4 digit numeric display.

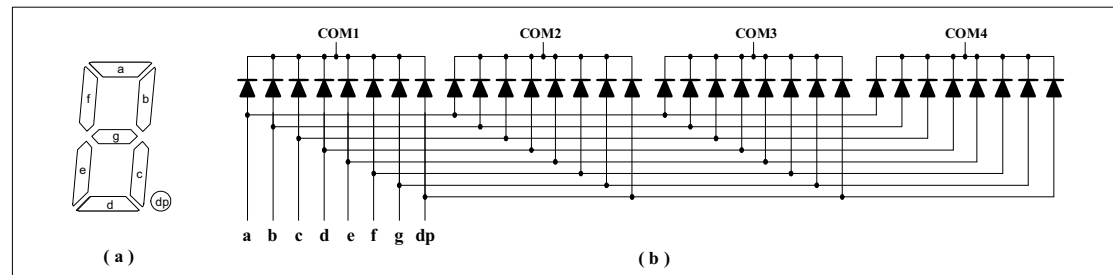


Fig 4.4 Internal component connection of the FND

As shown in Fig. 4.4, COM1, COM2, COM3, and COM4 are the digit selection ports connected to the cathodes of LEDs for each 4 digits. In addition, a, b, c, d, e, f, g, and dp ports are the segment selection port of a digit, which are connected to the anodes of the internal LEDs. By scanning COM1~COM4 ports with proper duration and frequency, 4 different digits can be simultaneously displayed.

As shown Fig. 4.5, SEGa~g and SEGdp ports of the FNDs are connected through resistor arrays to limit the current flowing through the segments.

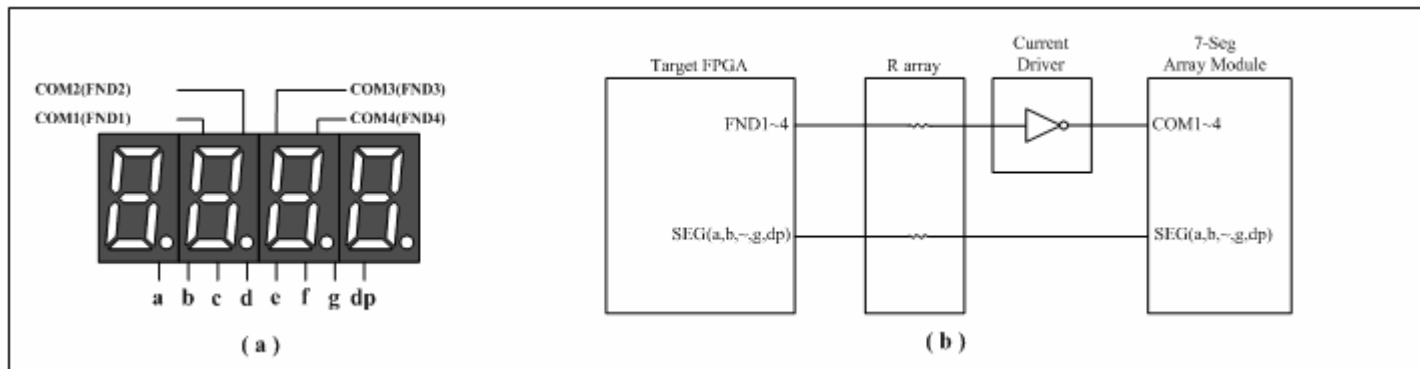


Fig 4.5 Simplified Block diagram between FPGA and the 4 digit numeric display

---

## **LEDs**

The QB-EP2C208-BK main board comes populated with 8 discrete indicator LEDs, which are connected through 330 ohm resistor array to the FPGA ports. Asserting port will turn the connected LED on.

## **Power Block**

The QB-EP2C208-BK has 3 regulator group which supply 1.5V, 3.3V, and 5V from a wall mount type 9VDC/500mA power supply respectively.(US or Asian/European model, depending on which was ordered) When the power are supplied to the QB-EP2C208-BK board properly, the power LED at upper right of the main board stays on.

## **Extension Port (EXTA, EXTB, EXTC)**

Three 34-pin extension port groups, named EXTA, EXTB, EXTC are placed on top, right, and bottom sides of the QB-EP2C208-BK main board. All pin headers of the extension port are placed on 0.1 inch spacing to allow the use of standard prototyping module with the QB-EP2C208-BK. As shown in Fig. 4.6, among 34 pins of each extension port group, 2 pins are assigned to GND, and the additional 2pins 3.3VDC power. A user should note that total power dissipation of the prototyping module must not exceed 150mA

(WARNING! When the Extension Ports are **not** used in your design, please **DO NOT touch them with any conducting materials**. Reserving **all unused pins as inputs, tri-stated** can be the good choice to avoid damages by unwanted short current.)

(WARNING! Do not directly connect 5V devices to the QB-EP2C208-BK extension pins. See the Cyclone device handbook, which can be downloaded at [http://www.altera.com/literature/hb/cyc2/cyc2\\_cii5v1.pdf](http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf) , for proper connection method.)

(WARNING! Improper connection to the extension ports may cause serious problem to the main FPGA device.)

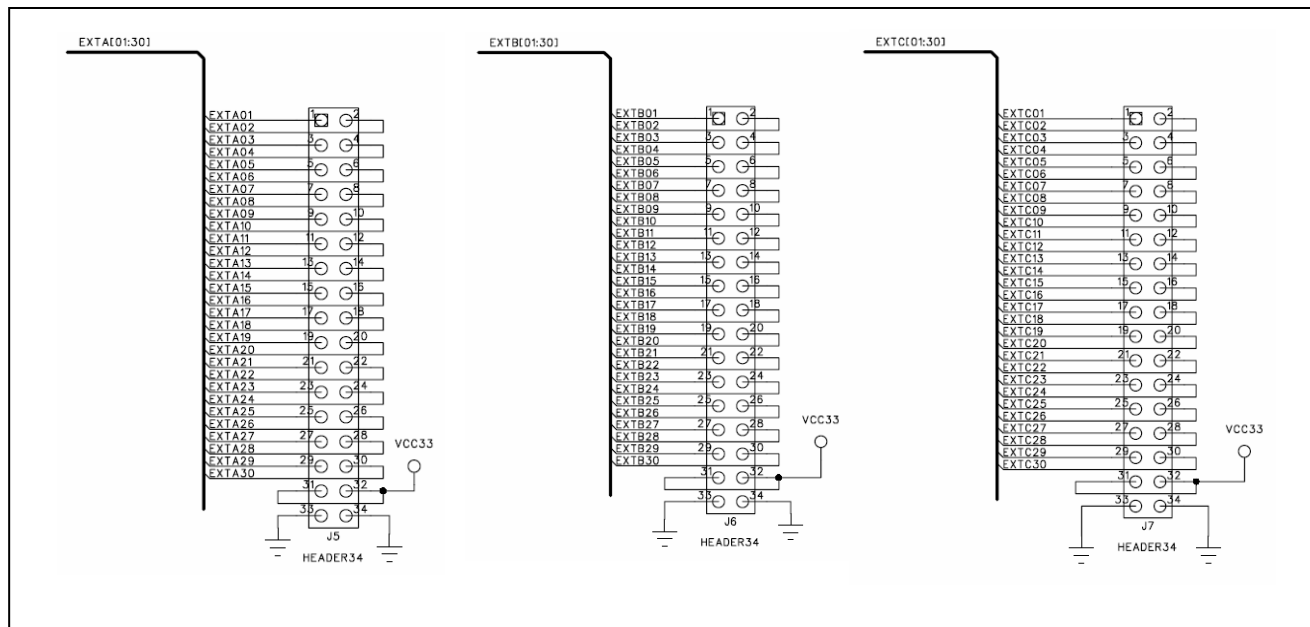


Fig 4.6 Extension port configuration



---

## Serial Communication Port (USART)

9pin D-sub connector is installed for RS-232 communication between the Cyclone II FPGA device, as shown in Fig. 4.7.

(**Note:** You can see blinking light of the TXD and RXD diodes during communication. They are installed on the upper right side of the QB-EP2C208-BK main board.)

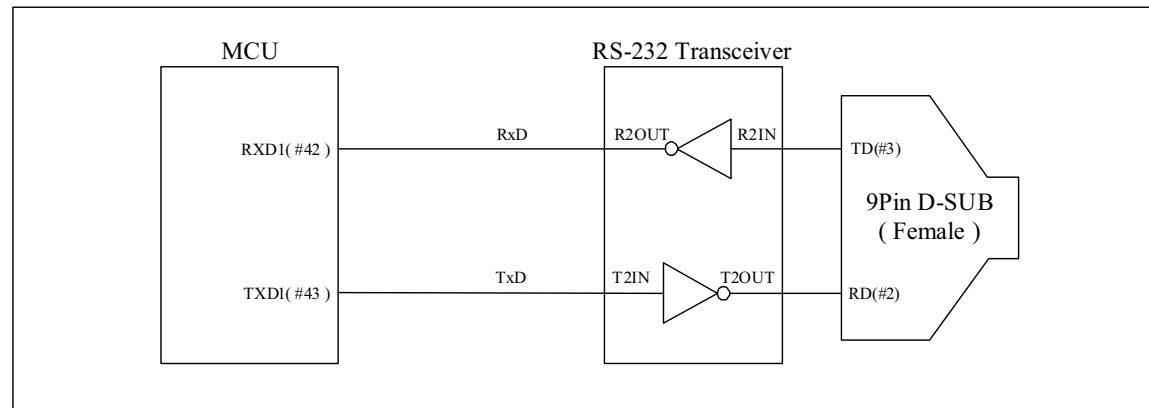


Fig 4.7 Block diagram of the USART block

---

## Character LCD

Supplied 16 X 2 character LCD with QB-EP2C208-BK board is designed to be detachable from the board. It can be controlled through the three control line and 8-bit data bus. The LCD's main controller is widely used LCD controller IC of S6A0069 (Samsung semiconductor) and is easy to use.

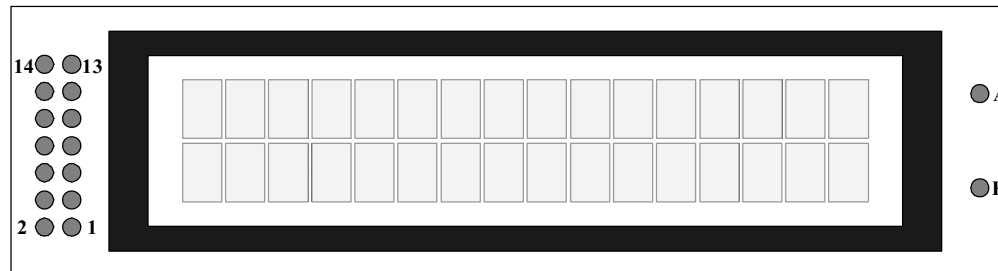


Fig 4.8 16x2 Character LCD Module

Fig 4.9 shows how to attach the LCD module to the QB-EP2C208-BK Cyclone II main board. As shown in Fig. 4.9 it can be easily installed by just connecting the pin header at the left side of the LCD module to the pin header socket on the FPGA main board. In addition, LCD back lighting is possible by just connecting a jumper cable between A & K marked pads and the pin header at the upper right of the FPGA main board ( The jumper cable for the back light is not supplied in the kit)

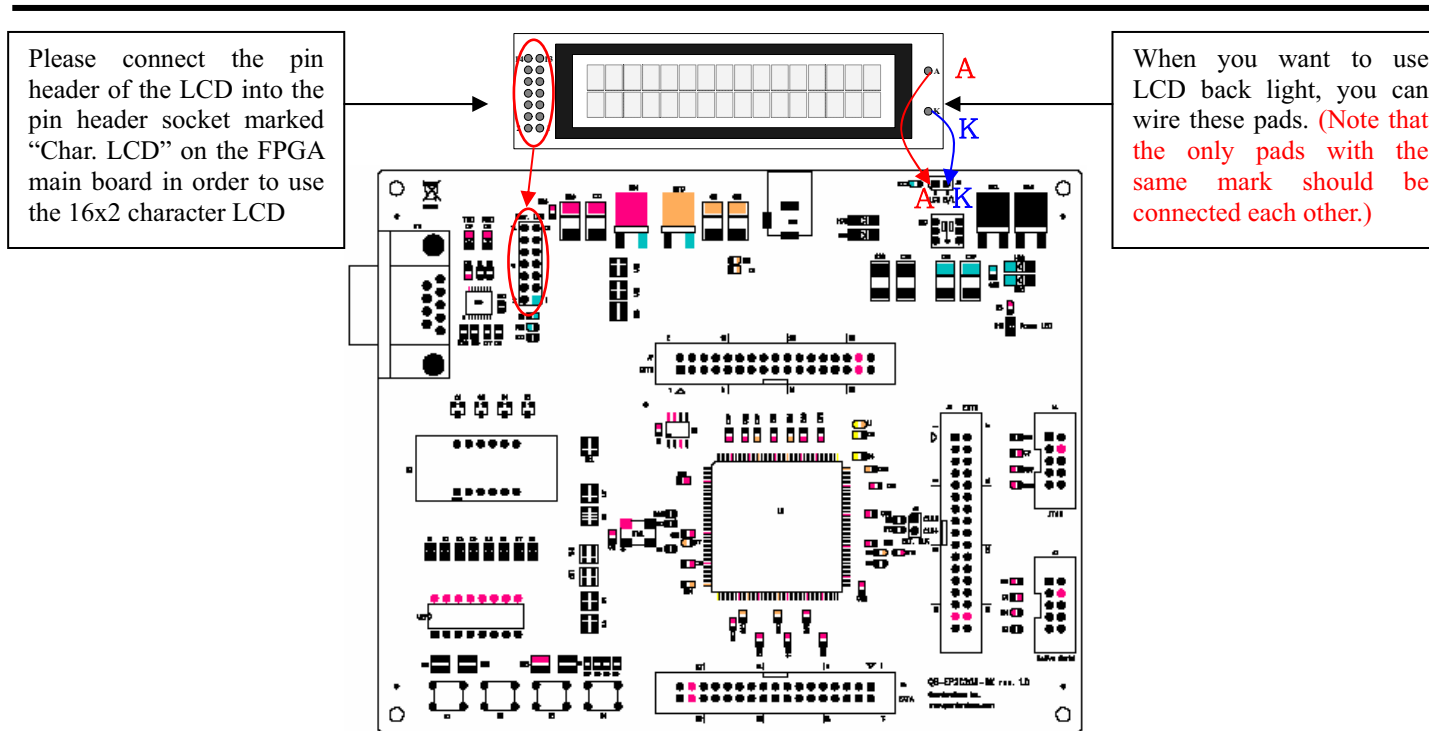


Fig 4.9 16x2 Character LCD Module Installation

Fig. 4.10 shows the connection scheme among the target FPGA, EXTB ports and the LCD. As shown in Fig. 4. 10, the control lines and the data bus are shared by some of the extension ports of EXTB06 ~ EXTB16. Thus you should detach the LCD module from

---

the board when you need to use the extension ports freely.

(Pleaser refer to “Port MAP” in Chapter 5 for more connection details.)

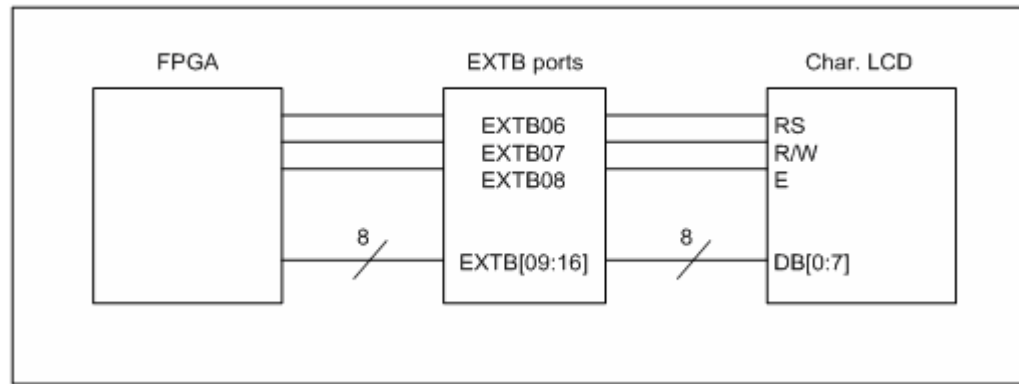


Fig 4.10 Block diagram of the LCD connection among the target FPGA, EXTB ports and the LCD

Table 4.1 Character LCD's timing

Mode	Characteristics	Symbol	Min	Typ	Max	Unit
Write Mode (refer to Figure-6)	E Cycle Time	$t_c$	500	-	-	ns
	E Rise / Fall Time	$t_R, t_F$	-	-	20	
	E Pulse Width (High, Low)	$t_w$	230	-	-	
	R/W and RS Setup Time	$t_{su1}$	40	-	-	
	R/W and RS Hold Time	$t_{H1}$	10	-	-	
	Data Setup Time	$t_{su2}$	80	-	-	
	Data Hold Time	$t_{H2}$	10	-	-	
Read Mode (refer to Figure-7)	E Cycle Time	$t_c$	500	-	-	ns
	E Rise / Fall Time	$t_R, t_F$	-	-	20	
	E Pulse Width (High, Low)	$t_w$	230	-	-	
	R/W and RS Setup Time	$t_{su}$	40	-	-	
	R/W and RS Hold Time	$t_H$	10	-	-	
	Data Output Delay Time	$t_D$	-	-	120	
	Data Hold Time	$t_{DH}$	5	-	-	

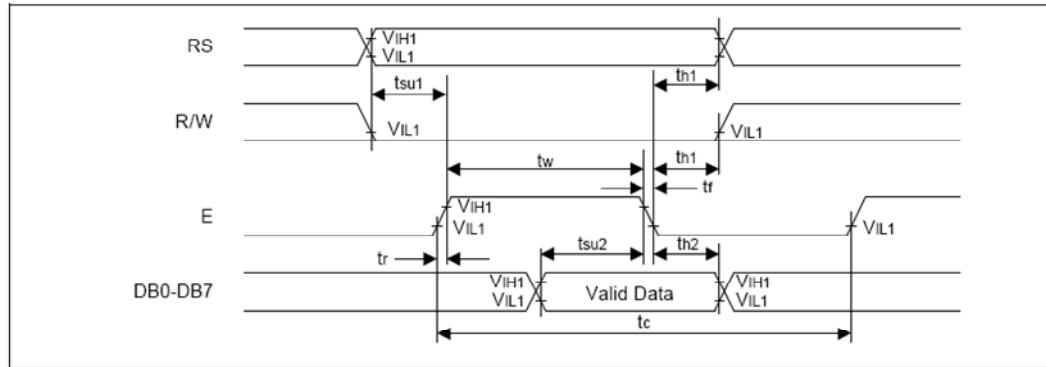


Fig 4.11 Write Mode Timing Diagram

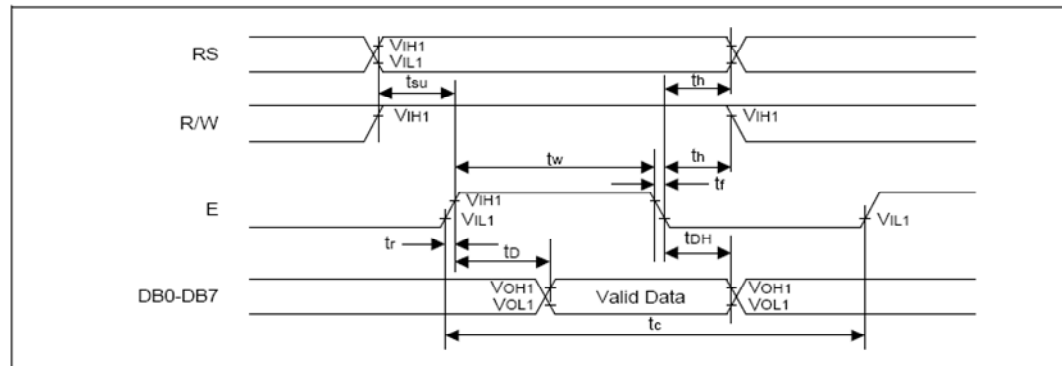


Fig 4.12 Read Mode Timing Diagram

---

There are some important registers, flags, and memory in the LCD module as shown below, and a user need to understand them.

- ✓ Busy Flag (BF) : When BF = “High”, it indicates that the internal operation is being processed. So during this time the next instruction cannot be accepted. BF can be read, when RS = Low and R/W = High (Read Instruction Operation), through DB7 port. Before executing the next instruction, be sure that BF is Low
- ✓ Address Counter (AC) : AC stores DDRAM/CGRAM address, transferred from IR. After writing into (reading from) DDRAM/CGRAM, AC is automatically increased (decreased) by 1. When RS = “Low” and R/W = “High”, AC can be read through DB0 ~ DB6 ports.
- ✓ Display Data RAM (DDRAM) : DDRAM stores display data of maximum 80 X 8 bits(80 characters), but the LCD module installed can display 16 characters by 2 lines. Fig. 4.13 shows the DDRAM address with respect to the character location.

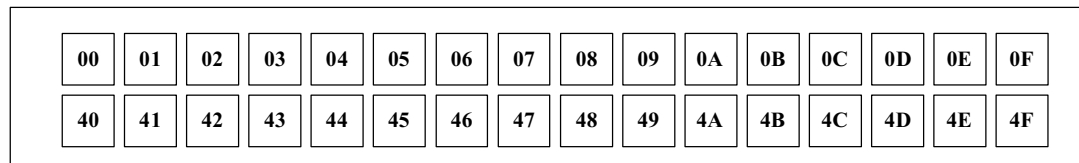


Fig 4.13 The character location v.s DDRAM address(default)

- ✓ Character Generator ROM (CGROM) : CGROM has 204 character patterns of 5 x 8 dots, and 32 character patterns of 5 x 11 dots as shown in Table 4.2.

- ✓ Character Generator RAM (CGRAM) : CGRAM has up to 5 x 8 dots, 8 characters. By writing font data to CGRAM, user defined characters can be used.

Table 4.2 Stored character sets

LOWER + BITS	UPPER + BITS	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)		0	@	P	\	P		-	夕	ミ	α	p	
0001	(2)	!	1	A	Q	a	q	。	ア	チ	ㄥ	ä	q	
0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	β	θ	
0011	(4)	#	3	C	S	c	s	」	ウ	テ	ε	ε	ω	
0100	(5)	\$	4	D	T	d	t	、	エ	ト	†	μ	Ω	
0101	(6)	%	5	E	U	e	u	・	オ	ナ	1	ε	Ü	
0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	p	Σ	
0111	(8)	'	7	G	W	g	w	ヲ	キ	ヌ	ラ	g	π	
1000	(1)	<	8	H	X	h	x	イ	ク	ネ	リ	」	×	
1001	(2)	>	9	I	Y	i	y	ウ	ケ	ル	」	」	y	
1010	(3)	*	:	J	Z	j	z	エ	コ	ハ	レ	j	≠	
1011	(4)	+	;	K	[	k	[	オ	サ	エ	ロ	°	π	
1100	(5)	,	<	L	¥	l	l	ヤ	シ	フ	ワ	φ	π	
1101	(6)	-	=	M	J	m	〉	ユ	ズ	ヘ	ン	も	÷	
1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ッ	ñ		
1111	(8)	/	?	O	_	o	←	ッ	ツ	マ	°	ö	■	

Table 4.3 shows the instruction sets of the LCD Module. When you make a LCD control logic, please special care must be taken to grantee the execution time requirements of each instructions.

Table 4.3 Instruction Sets of the LCD Module

Instruction	Instruction Code										Description Instruction Code	Execution time (fsoc=270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC.	1.53ms
Return Home	0	0	0	0	0	0	0	0	1	X	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39μs

Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39 $\mu$ s
Function Set	0	0	0	0	1	DL	N	F	X	X	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39 $\mu$ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 $\mu$ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 $\mu$ s
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 $\mu$ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 $\mu$ s
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 $\mu$ s

(When an FPGA logic with checking the Busy Flag (DB7) is made, it must be necessary 1/2 fosc is necessary for executing the next instruction by the falling edge of the 'E' signal after the Busy Flag (DB7) goes to "LOW".)

- ✓ Clear Display : clear all the display data by writing "0x20"(space code) to all DDRAM address, and set DDRAM address

---

to 0x00 into address counter. Return cursor to the original status,

- ✓ Return Home : set DDRAM address to “0x00” into the address counter. Return cursor to its original site and return display to its original status, if shifted. Contents of DDRAM does not changed
- ✓ Entry Mode Set : set the moving direction of cursor and display. When I/D = “High”, cursor/blink moves to right and DDRAM address is increased by 1. When I/D = “Low”, cursor/blink moves to left and DDRAM address is decreased by 1.
- ✓ Display On/Off Control : control Display(D), cursor(C), and Blink(B) ON/OFF 1bit register. When the bit is set, the function is enable.
- ✓ Cursor or Display Shift : Shifting of right/left cursor position or display without writing or reading of display data.
- ✓ Function Set : 8bit data interface(DL=1), 2 lines(N=1), and 5x8 font(F=0) is recommended to be selected in using QB-EP2C208-BK board. Please refer to S6A0069 data sheet on the document CDROM for more details.
- ✓ Set CGRAM address : Set CGRAM address to AC.
- ✓ Set DDRAM address : Set DDRAM address to AC.
- ✓ Read BF & address : Read Busy Flag(BF) and address counter.

### **LCD Module initialization and Display Sequence**

- ✓ Wait for the initialization of LCD Module(> 30ms) after system reset
- ✓ Perform Display On/Off Control command

- 
- ✓ Perform Entry mode set command
  - ✓ Set DDRAM address(1st line)
  - ✓ Send character data to display in series (1st Line)
  - ✓ Set DDRAM address(2nd Line)
  - ✓ Send character data to display in series (2nd Line)

## 5 Port MAP

FPGA Pin#	Board Connection	FPGA Pin#	Board Connection	FPGA Pin#	Board Connection
<b>Clock</b>		<b>LED</b>		68	KEY2
23	CLK0	35	LED1 (D1)	69	KEY1
27	CLK2	37	LED2 (D2)	<b>EXTA</b>	
131	CLK5	<b>39</b>	<b>LED3 (D3)</b>	70	EXTA30
132	CLK4	40	LED4 (D4)	72	EXTA29
<b>UART</b>		41	LED5 (D5)	74	EXTA28
3	TXD	43	LED6 (D6)	75	EXTA27
4	RXD	44	LED7 (D7)	<b>76</b>	<b>EXTA26</b>
<b>FND</b>		45	LED8 (D8)	77	EXTA25
5	SEGE	<b>DIP S/W</b>		80	EXTA24
6	SEGA	46	DIP1	81	EXTA23
8	SEGF	56	DIP2	82	EXTA22
10	SEGB	57	DIP3	<b>84</b>	<b>EXTA21</b>
<b>11</b>	<b>SEGG</b>	58	DIP4	86	EXTA20
12	SEGC	<b>59</b>	<b>DIP5</b>	87	EXTA19
13	SEGDP	60	DIP6	88	EXTA18
14	SEGD	61	DIP7	89	EXTA17
30	FND4	63	DIP8	<b>90</b>	<b>EXTA16</b>
<b>31</b>	<b>FND3</b>	<b>KEY</b>		92	EXTA15
33	FND2	64	KEY4	94	EXTA14
34	FND1	67	KEY3	95	EXTA13

“FPGA pin #” column shows the FPGA pin number itself, and “Board Connection” column peripheral information which connected to the FPGA pins at the same row. (Note: Installed 50MHz clock is initially connected to pin 27, CLK2.)

(Continued)

FPGA Pin#	Board Connection	FPGA Pin#	Board Connection	FPGA Pin#	Board Connection
96	EXTA12	133	EXTB22	164	EXTB01
97	EXTA11	134	EXTB21	EXTC	
99	EXTA10	135	EXTB20	165	EXTC30
101	EXTA09	137	EXTB19	168	EXTC29
<b>102</b>	<b>EXTA08</b>	<b>138</b>	<b>EXTB18</b>	169	EXTC28
103	EXTA07	139	EXTB17	170	EXTC27
104	EXTA06	141	EXTB16 (DB7)	<b>171</b>	<b>EXTC26</b>
105	EXTA05	142	EXTB15 (DB6)	173	EXTC25
106	EXTA04	143	EXTB14 (DB5)	175	EXTC24
<b>107</b>	<b>EXTA03</b>	<b>144</b>	<b>EXTB13 (DB4)</b>	176	EXTC23
110	EXTA02	145	EXTB12 (DB3)	179	EXTC22
112	EXTA01	146	EXTB11 (DB2)	<b>180</b>	<b>EXTC21</b>
<b>EXTB</b>		147	EXTB10 (DB1)	181	EXTC20
113	EXTB30	149	EXTB09 (DB0)	182	EXTC19
114	EXTB29	<b>150</b>	<b>EXTB08 (E)</b>	185	EXTC18
115	EXTB28	151	EXTB07 (RW_N)	187	EXTC17
116	EXTB27	152	EXTB06 (RS)	<b>188</b>	<b>EXTC16</b>
<b>117</b>	<b>EXTB26</b>	160	EXTB05	189	EXTC15
118	EXTB25	161	EXTB04	191	EXTC14
127	EXTB24	162	EXTB03	192	EXTC13
128	EXTB23	163	EXTB02	193	EXTC12

.(Note: The extension ports from EXTB06 to EXTB16 are shared by the character LCD control and data ports.)

.(Note: When you use EXTB06 ~ EXTB16 ports for the LCD, please set the I/O standards of the pin as 3.3V-PCI so that it can safely communicate with the 5V operated LCD module)

---

(Continued)

FPGA Pin#	Board Connection
195	EXTC11
197	EXTC10
198	EXTC09
199	EXTC08
<b>200</b>	<b>EXTC07</b>
201	EXTC06
203	EXTC05
205	EXTC04
206	EXTC03
207	EXTC02
208	EXTC01

\* Power output ports for external circuits

Power Output Port	Extension port
DC 3.3V	EXTA31, EXTA32, EXTB31, EXTB32 EXTC31, EXTC32
GND	EXTA33, EXTA34, EXTB33, EXTB34 EXTC33, EXTC34

.(Warning: The total current of external circuits using the power from the extension ports should not be exceeded 200mA, so that the FPGA board can be protected from unwanted electrical damage)

---

## 6 Tutorial

Learning by example seems to be the best way to learn design and implementation procedures with Altera's Quartus II and QB-FPGA200EP2C starter kit. We are sure that you will learn the procedures easily just by following this tutorial.

First of all, please make sure if your QB-FPGA200EP2C starter kit have been installed properly as shown in Fig. 4.2. Secondly, please check if Quartus II Web edition has been installed on your PC. If not, please download up-to-date version of Quartus II Web edition with free of charge from Altera's web site as shown below;

<http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html>.

and then install it on your PC.

The 8.1 or later version of the Quartus II is supplied completely free and doesn't need any license to work.

Now, you are ready to do it yourself with this tutorial.

---

## 6.1 Design Flow on Quartus II

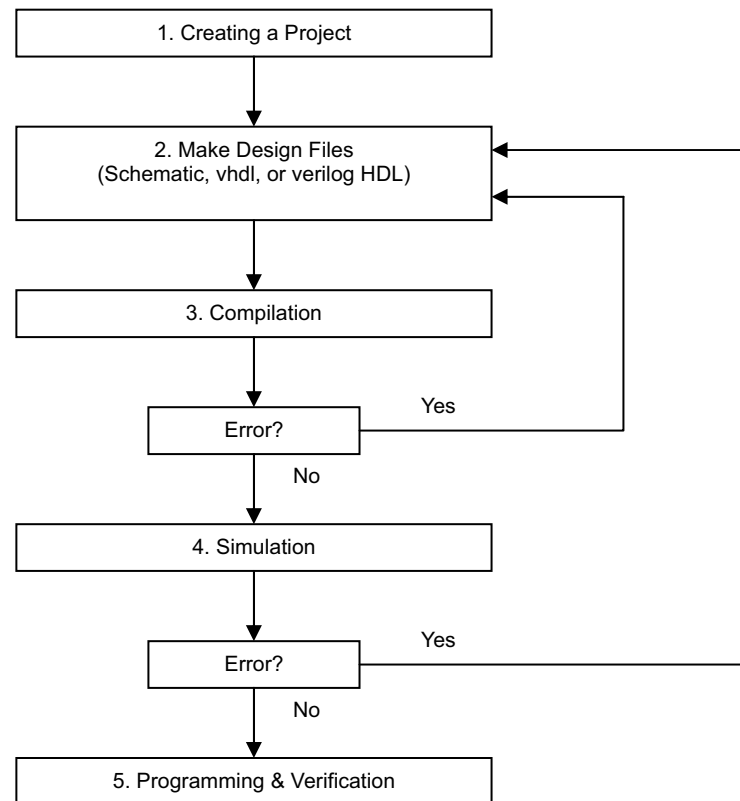


Fig 6.1 Simplified design flow on Quartus II Environment

---

Fig. 6.1 shows the design flow on Quartus II environment. In order to design new circuits on Quartus II, user should create a new Project managing all the design files easily. The file extension of the newly created Project file is **.qpf** (Quartus II Project File). After creating the new project, user can start to design his or her own circuit by schematic, VHDL or Verilog HDL. The file extensions of the schematic, VHDL, and Verilog HDL design files are **.bdf**, **.vhd**, and **.v**, respectively. Whatever you may choose your design method among schematic, VHDL and Verilog, the rest procedures such as compilation, simulation and programming procedure are exactly the same.

QuartusII compilation includes synthesis, place and route (fitting), and timing analysis procedures. During synthesis, QuartusII analyzes the design files, optimizes the designed logic. After the synthesis, optimized logic is placed and routed on a user specified FPGA and timing analysis follows. After all the compilation process completes, device programming images, in the form of Programmer Object Files (**.pof**), and SRAM Object Files (**.sof**) are generated. Where the **.pof** is the file for programming a configuration ROM and the **.sof** is the file for configuring a FPGA. If any error messages are found during the compilation process, a user should go back to the first design stage, correct errors in his or her design file and recompile it.

After the successful compilation, a user can see how it works virtually by the simulation process. A user can make input vectors for simulation easily by creating a Vector Waveform File (**.vwf**). The simulation results show how the newly designed hardware would respond to the input vectors defined on the **.vwf** file. If the simulation results are exactly the same as a user expects, he or she can verify the designed hardware by programming FPGA or configuration ROM on the QB-EP2C208-BK board. Please refer to [http://www.altera.com/literature/manual/intro\\_to\\_quartus2.pdf](http://www.altera.com/literature/manual/intro_to_quartus2.pdf) for more details of the Quartus II software.

---

## 6.2 A simple schematic design example

The tutorial in the section 6.2 shows a design procedure of a simple 2-input AND gate. The input and output device used in this design are designated by red arrows in Fig. 6.3.1.

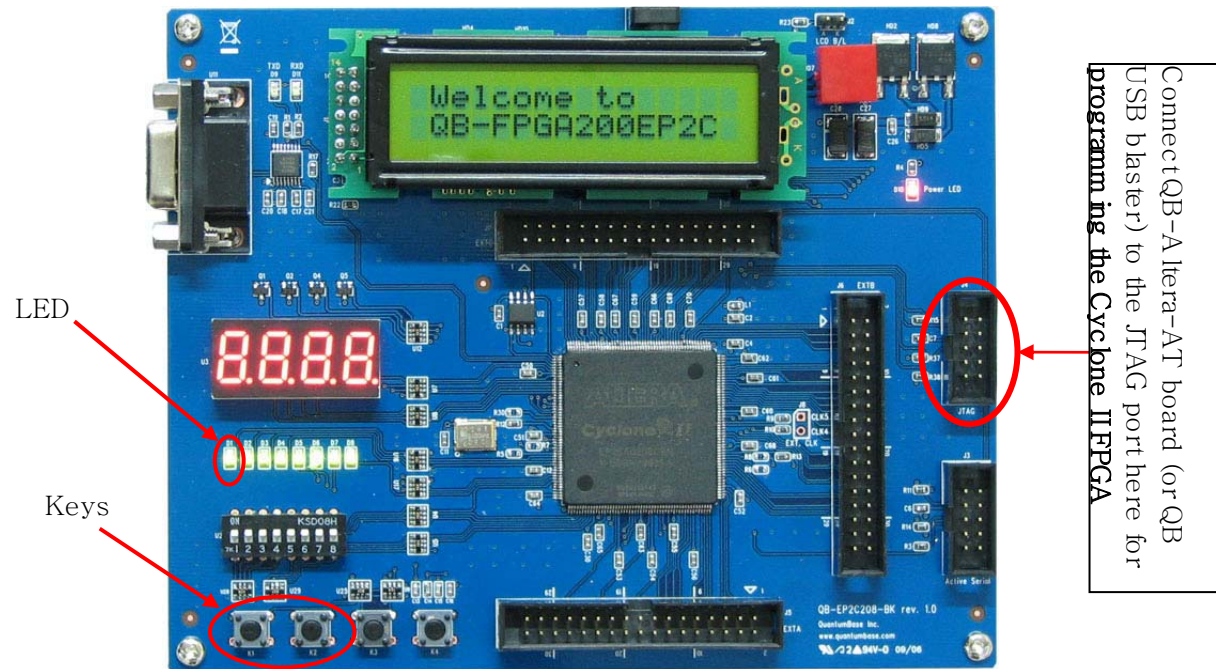


Fig. 6.2.1 The Key and LEDs used in this example

---

## 6.2.1 Creating Your Own Project

Double clicking Quartus II web edition icon on your desktop starts the software tool. Fig. 6.2.1 shows the Quartus II graphical user interface as it appears when you first start the software.

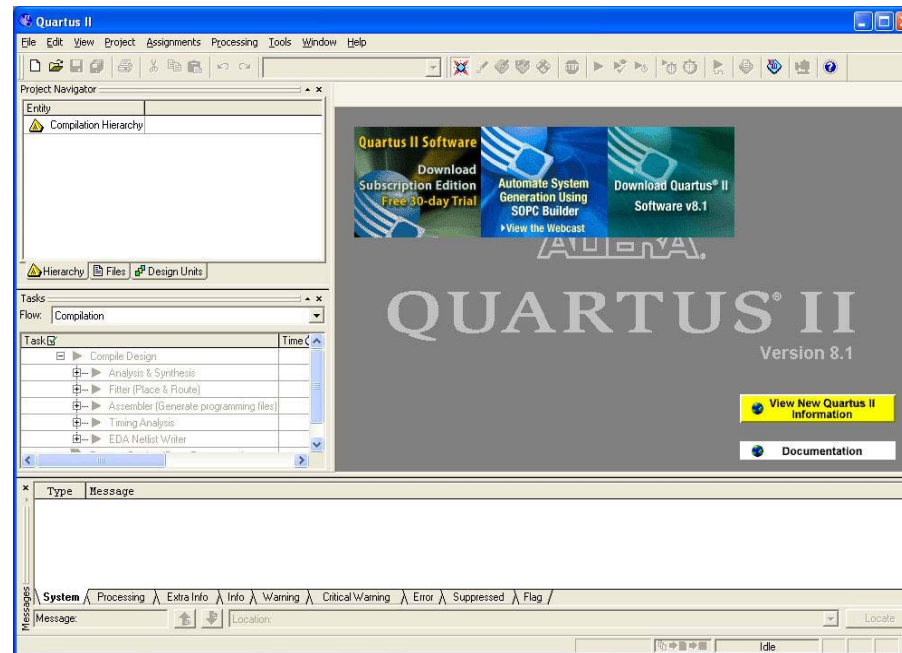


Fig. 6.2.1 Quartus II graphical user interface

In order to create your own project, choose **File/New Project Wizard** as shown in Fig. 6.2.2

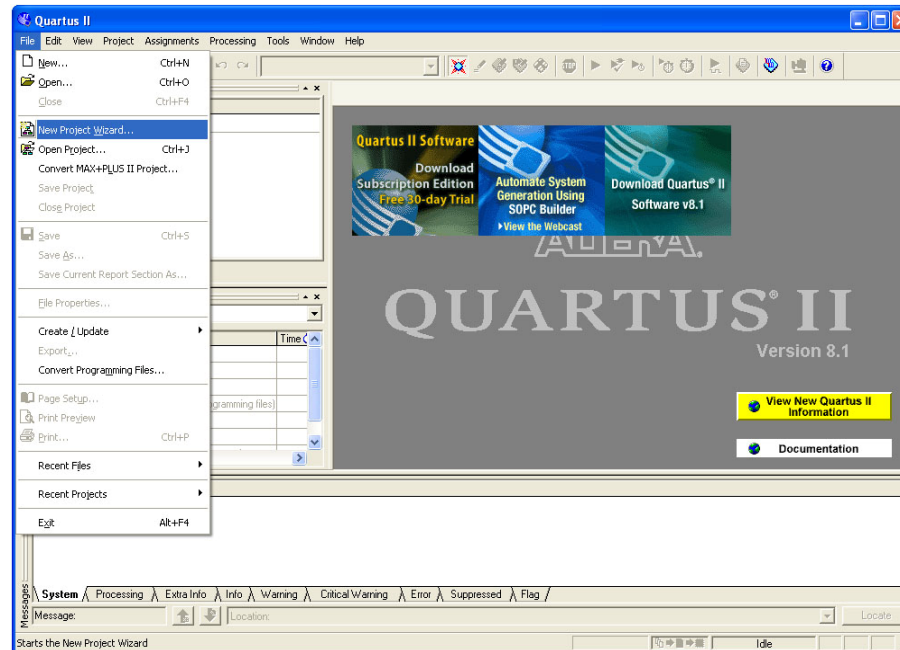


Fig. 6.2.2 Choosing File/New Project Wizard

---

This invokes the “New Project Wizard:Directory, Name, Top-Level Entity” dialog box as shown Fig 6.2.3

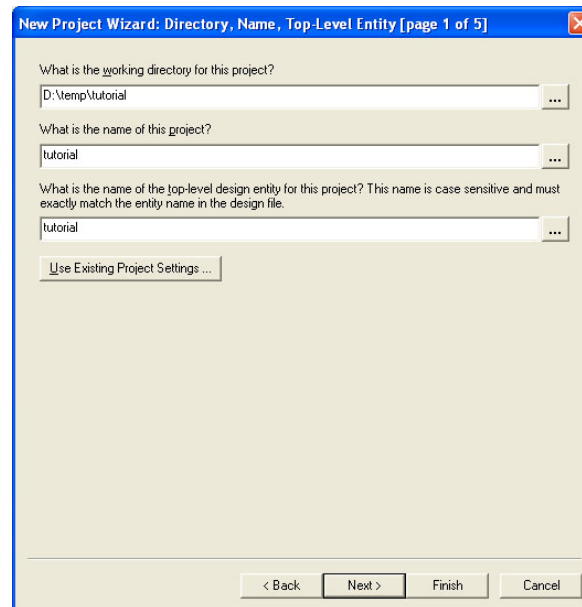


Fig. 6.2.3 The first dialog box regarding “New Project Wizard”

Type the directory name in the working directory box or select the directory with **Browse(...)**. Type a project name you are to create in the project name box. In the top-level design entity box, top level entity name should be shown and it should be exactly the same was the entity name in the design file. Let’s fill in the 3 boxes as shown in Fig. 6.2.3 in this tutorial course and click **Next..**

The **Add Files** page of the **New Project Wizard** appears. Since **tutorial** is a new project, there are no files to add to this project yet, so just click **Next**.

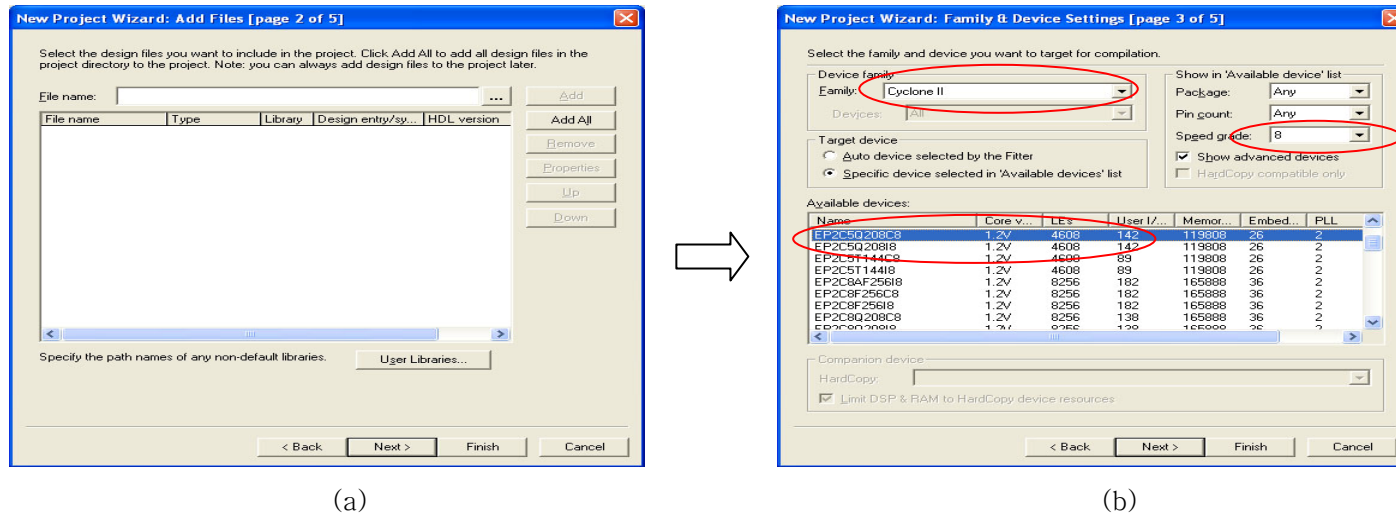


Fig. 6.2.4 (a)Add Files and (b)Family & Devices Settings pages of the New Project Wizard

Now the **Family & Device Settings** page of the **New Project Wizard** appears then. In the **Family** list, select **Cyclone**. In the **Speed grade** list of the **Filters** group, select **8**, and then select **EP2C5Q208C8** in the **Available devices** box

To accept the default setting for the remaining wizard prompts and create the project, click **Next** and **Finish** in turn. The project is now created. The top-level design entity name appears in the **Hierarchy** tab of the Project Navigator window.

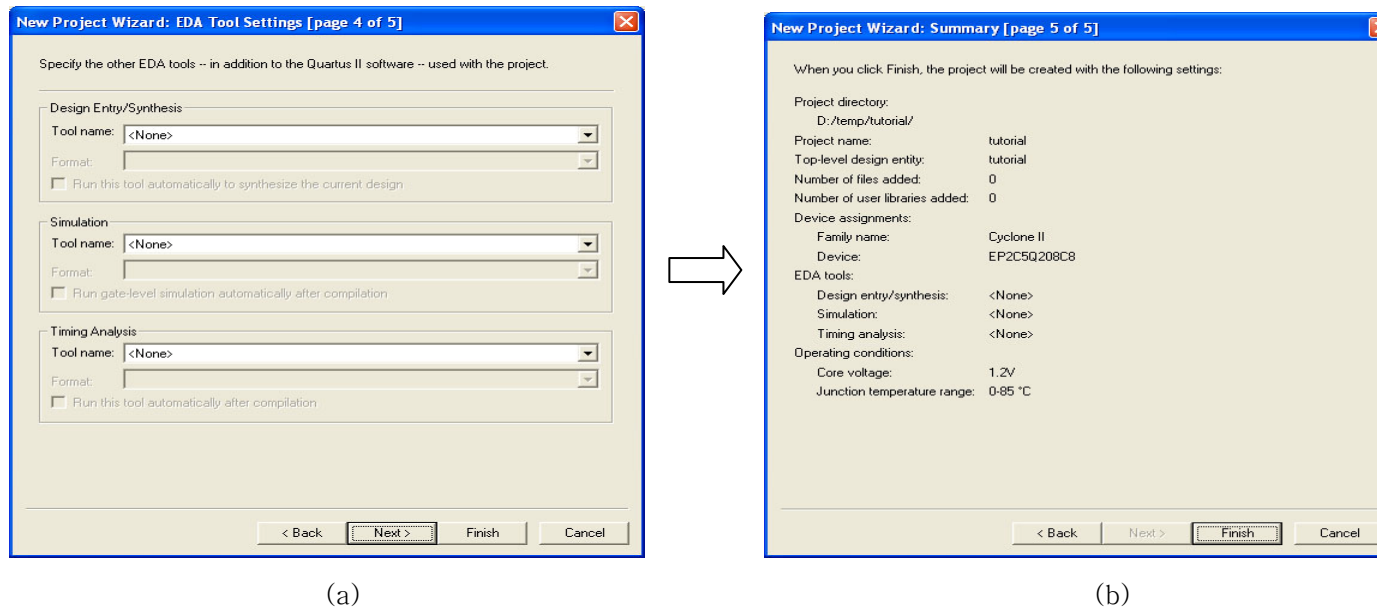


Fig. 6.2.5 (a)EDA Tool Settings and (b)Summary pages of the New Project Wizard

---

## 6.2.2 Setting Unused pins

As mentioned in Chapter 2, users need to give care to unused pin status because the FPGA can be damaged seriously by unintentional over current if both of the pin and a connected device are set as output and their output status are different from each other. Reserving the unused pins as input and making it tri-stated is the very common method to address the problem. Thus users can make the unused pins tri-stated by setting a option in Quartus II as shown below.

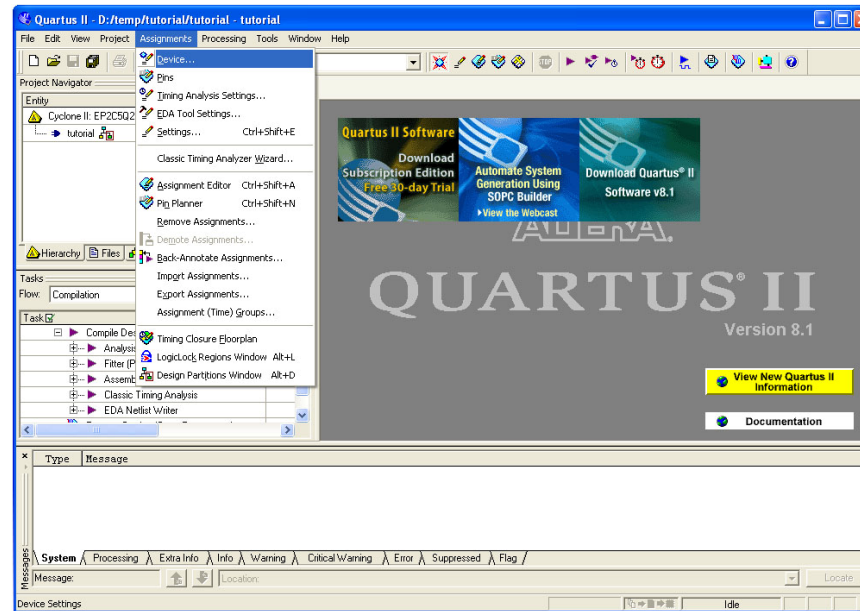
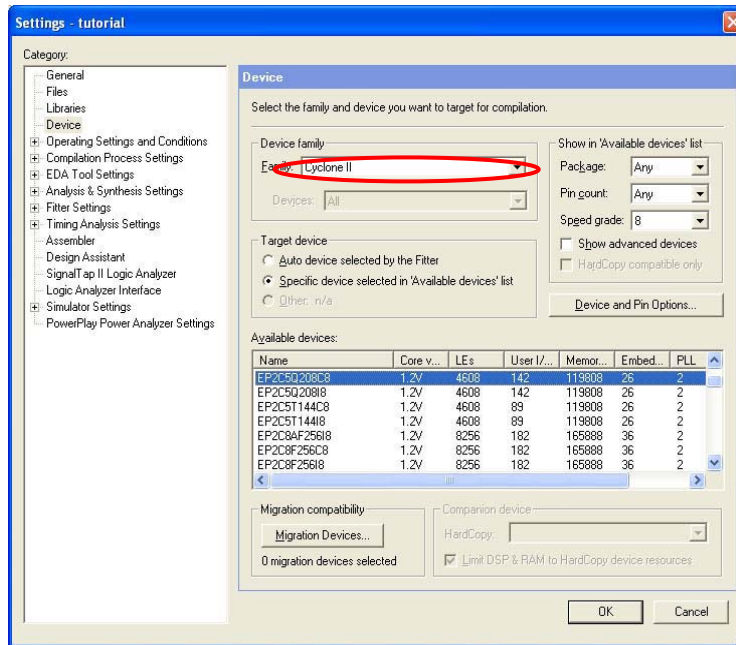
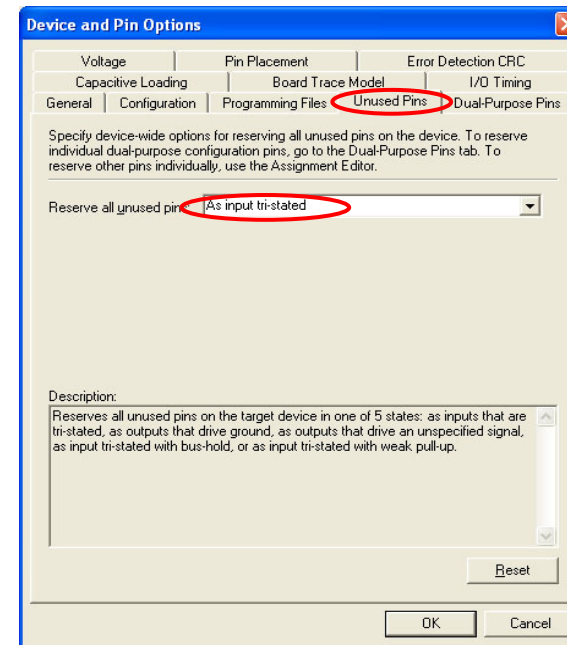


Fig. 6.2.6 Selecting Assignment menu

As shown in Fig. 6.2.6, choose **Device...** (**Assignment menu**). By clicking Device & Pin Options button as shown in Fig 6.2.7(a), user can invoke **Device & Pin Options** dialogue box of Fig. 6.2.7(b). As shown in Fig. 6.2.7(b), select **As inputs tri-stated** in Unused Pins Tab.



(a)



(b)

Fig. 6.2.7 (a) Device Setting Window (b) Device & Pin Options dialogue box

---

### 6.2.3 Create a New Schematic Design File

In this step you create a new schematic file called **tutorial.gdf**. this file is the top-level design entity of the **tutorial** project. Choose **File/NEW**. When the **Device Design Files** tab of the **NEW** dialog box appears, select **Block Diagram/Schematic File** and click **OK**.

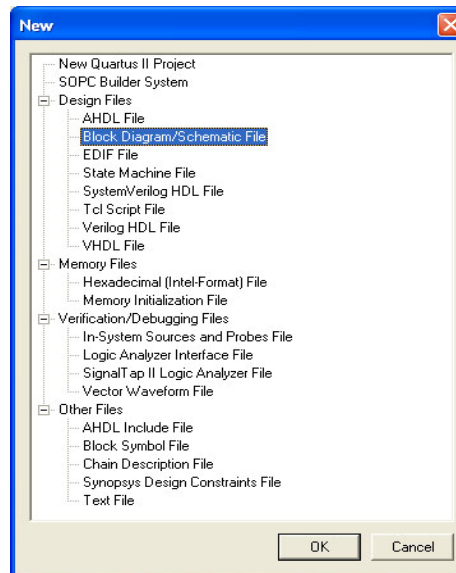


Fig. 6.2.8 New dialogue box for creating new design file

In the invoked schematic editor as shown in Fig. 6.2.9, add 2-input AND gate, 2 input and 1 output pins by using the Symbol tool as shown in Fig. 6.2.10(a) and Fig. 6.2.10(b).

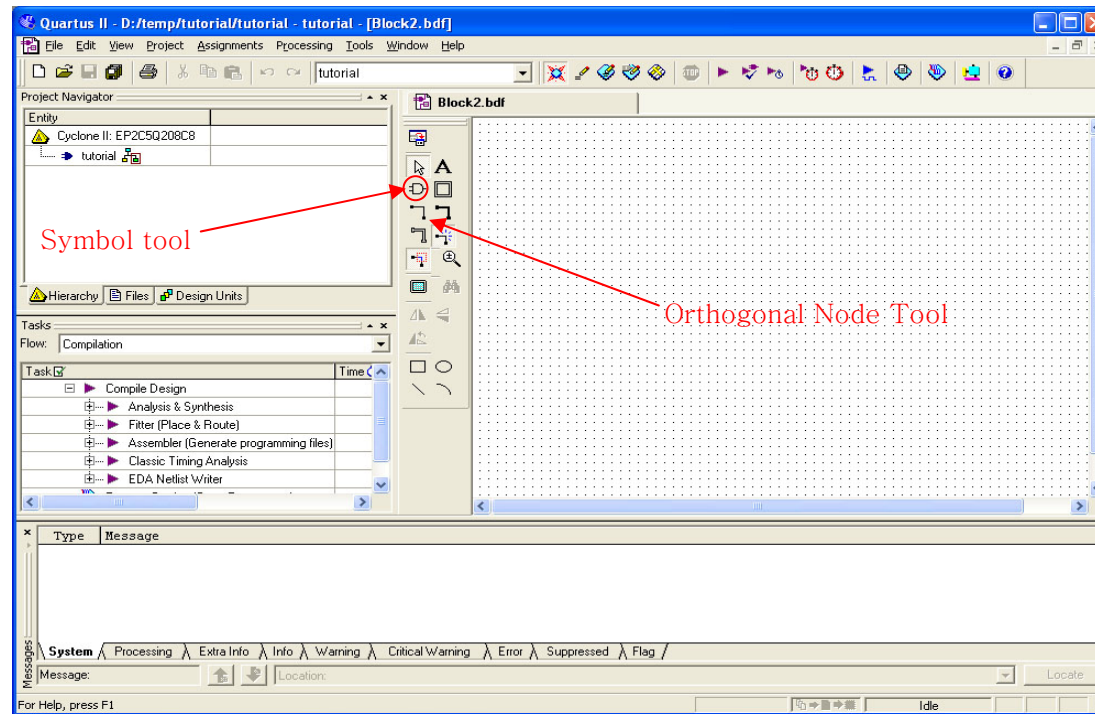
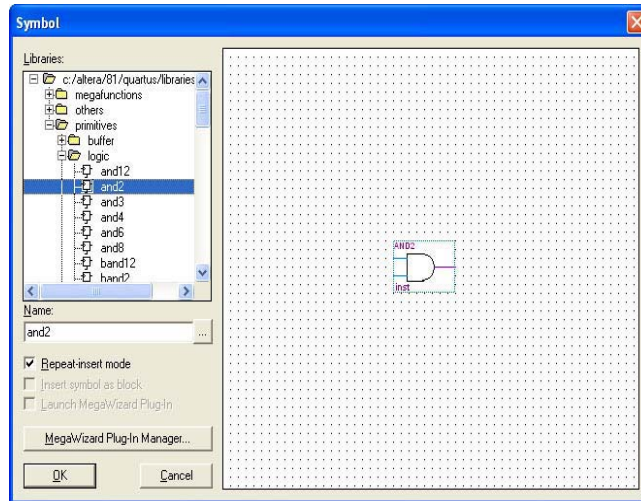
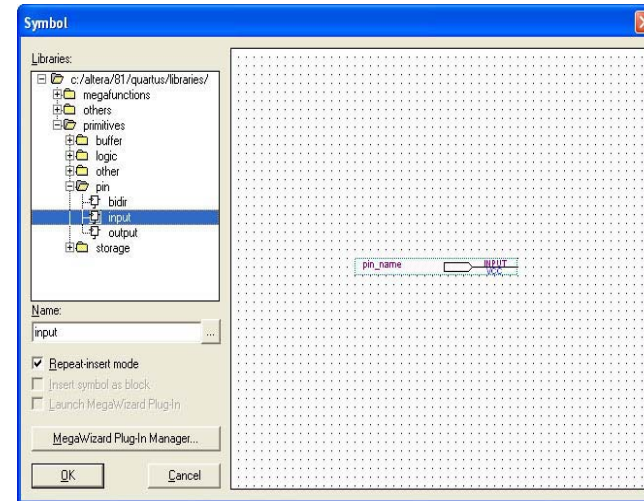


Fig. 6.2.9 Schematic Editor



(a)



(b)

Fig. 6.2.10 Symbol tool when (a) a 2-input AND gate, named **and2**, is selected and (b) a input pin, named **input**, is selected.

After adding the components, make connection between the **AND** gate and **Pins** by using the **Orthogonal Node Tools** shown in Fig. 6.2.9. Please refer to Quaruts II hand book for more details about the schematic editor at the Altera's Website below;

[http://www.altera.com/literature/hb/qts/quartusii\\_handbook.pdf](http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf)

Now 2-input AND gate design on the schematic editor finished as shown in Fig. 6.2.11. Save the file as **tutorial.bdf**.

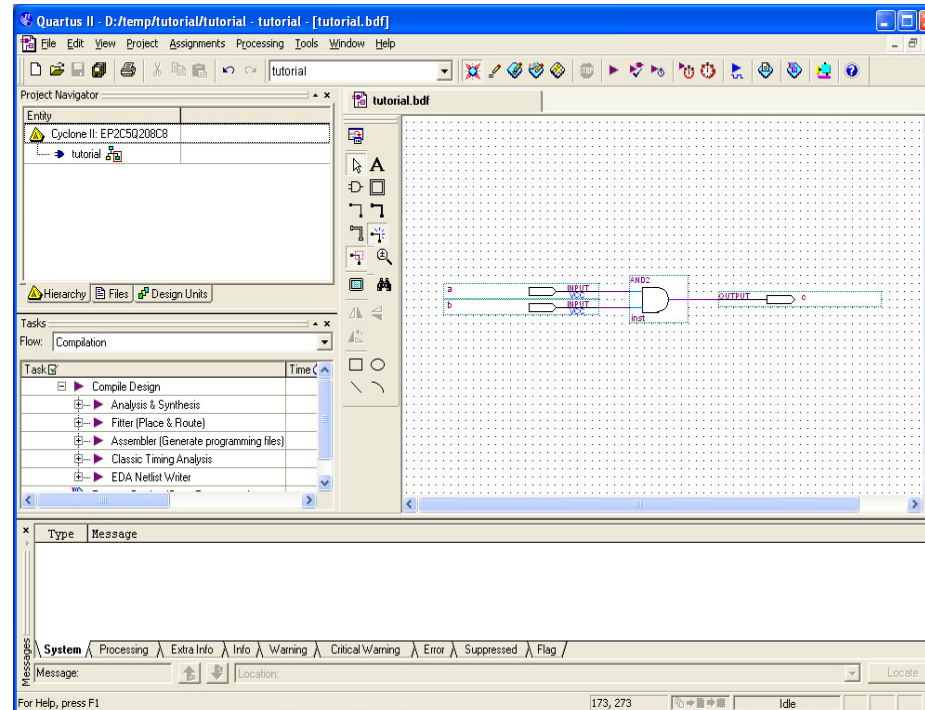


Fig. 6.2.11 Designed 2-input AND gate on the schematic editor

Now **choose Processing/Start Compilation** so that Quartus II makes full compilation of the **.bdf** file. If there are any errors, correct errors and recompile it.

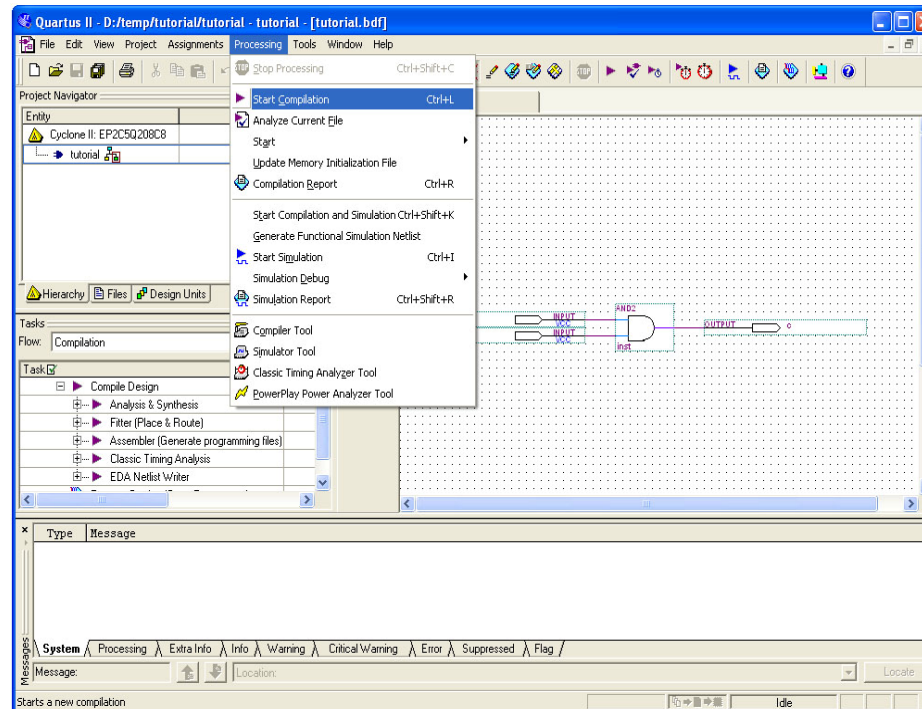


Fig. 6.2.12 Make full compilation of the Processing menu

“Full compilation was successful” message window appears when there are no errors in your schematic design.

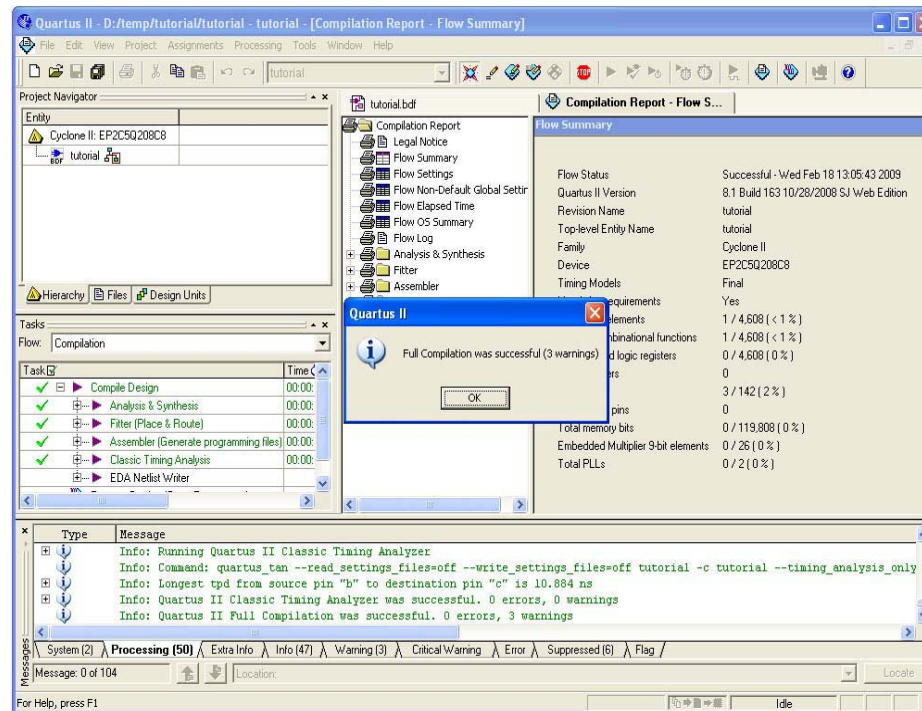


Fig. 6.2.13 Captured image when the full compilation was successful

---

## 6.2.4 Pin Assignment

Since there are no errors in your design now, you should assign FPGA pins to physically connected peripheral resources. In **tutorial.bdf**, 2 input pins of **a** and **b**, and 1 output pin **c** should be assigned to Key1, Key2, and LED1, respectively.

Referring to PORT MAP of Chapter 5, we can assign the input and output pins as shown in Table 6.2.1.

Table 6.2.1 Selected port map for this tutorial

FPGA Pin#	Board Connection	Pin Name in the Design
69	KEY1	a
68	KEY2	b
35	LED1	c

Next page shows the way to assign pins in the Quartus II software tool.

Choose **Assignment Editor** (Assignment menu).

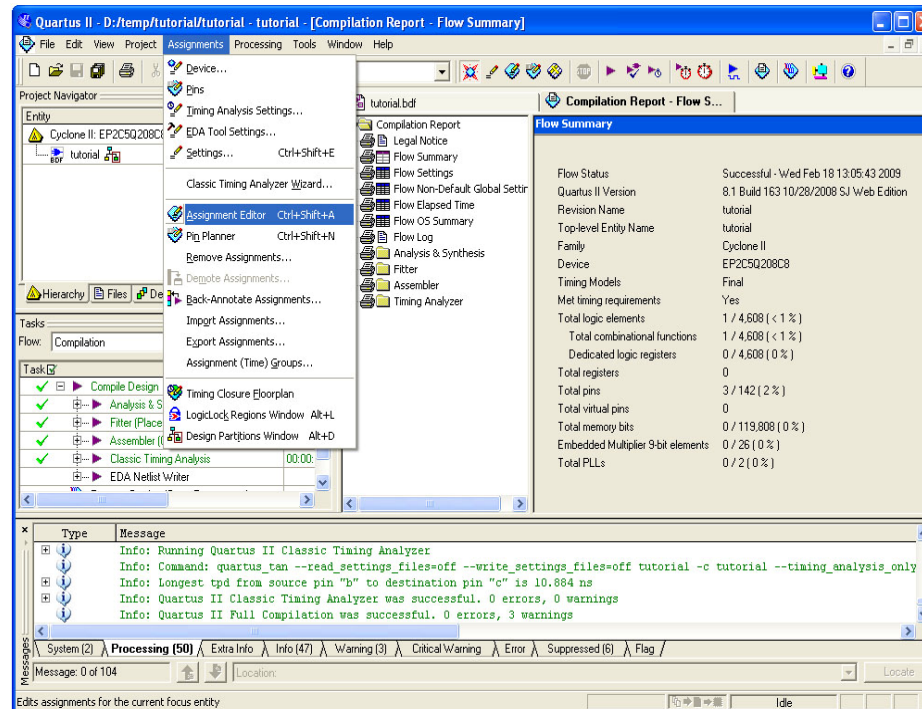


Fig. 6.2.14 Choosing **Assignments/Assignment Editor** for the pin assignment

Then the Assignment Editor appears with the Pin assignment category selected. In the Assignment Editor, double-click the **To** cell and select the **a** input node name.

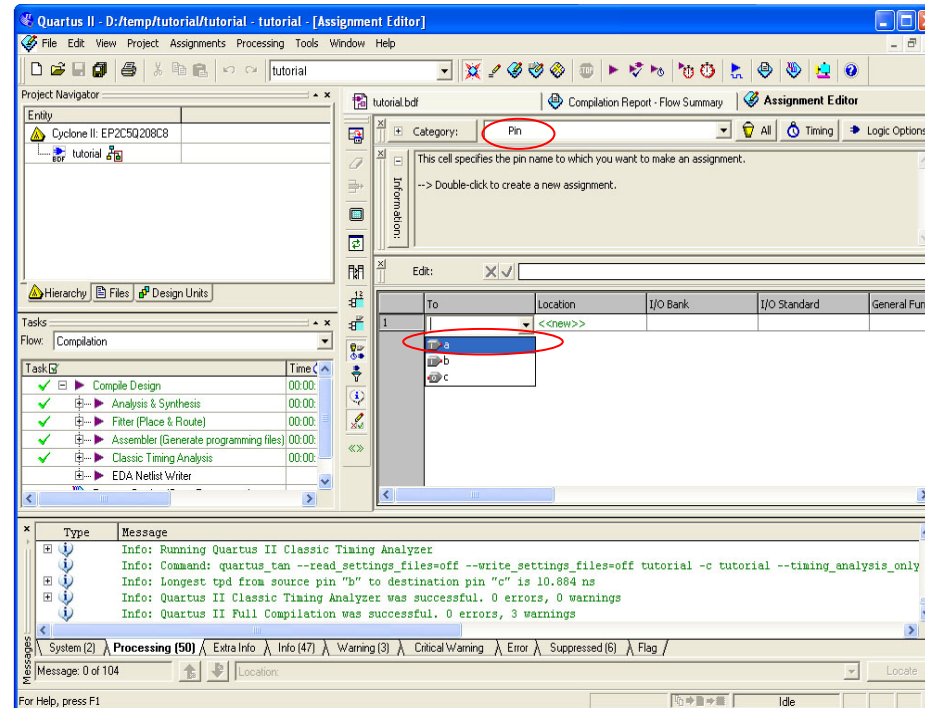


Fig. 6.2.15 Select port in the Assignment Editor

In the assignment Editor, double-click the Location cell and scroll down to select pin 69, an I/O pin.

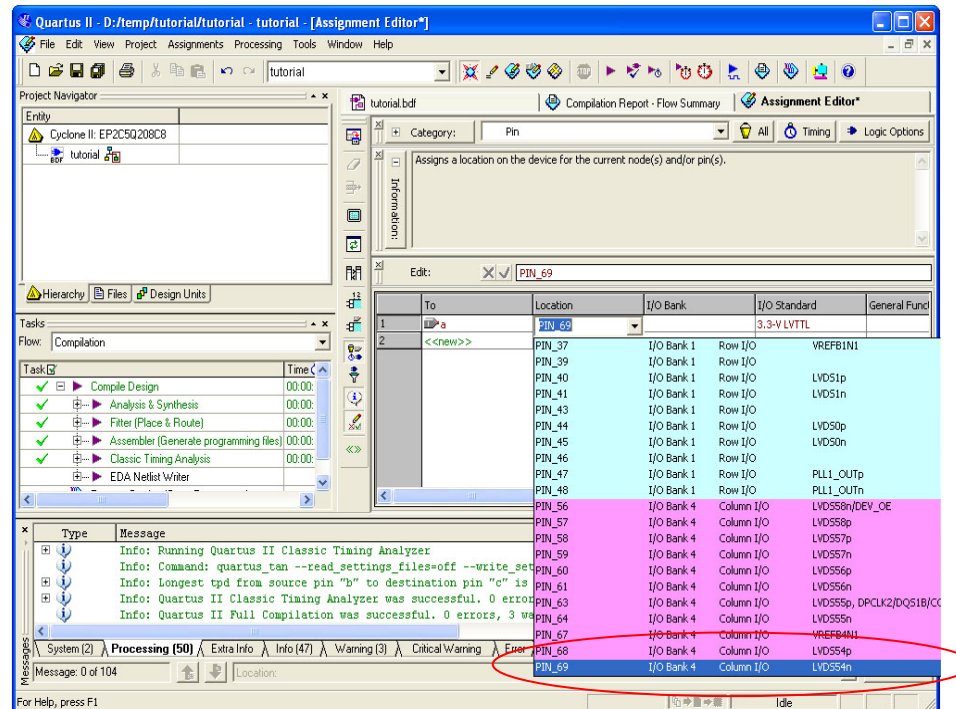


Fig. 6.2.16 Select a pin in the Assignment Editor

Keep assigning remaining pins referring to Table 6.2.1

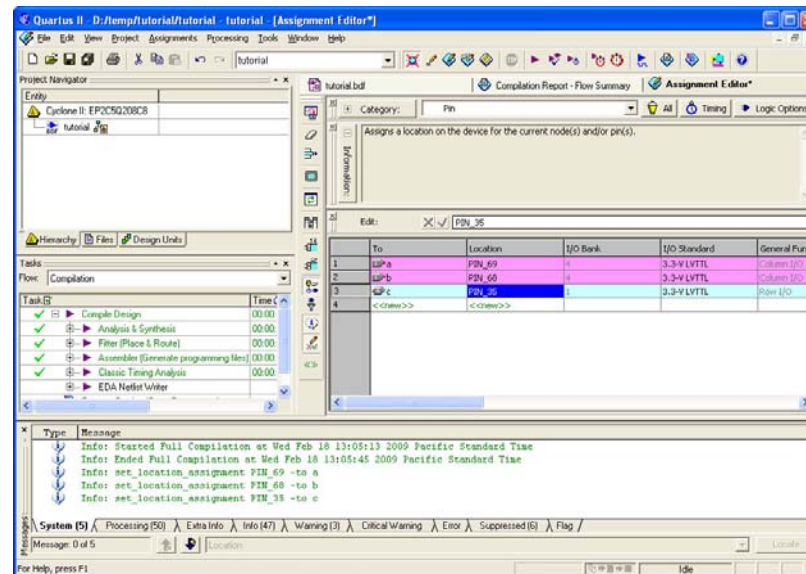


Fig. 6.2.17 Assignment Editor when all the pins are completely assigned.

After the pin assignment is completed, close the Assignment Editor, saving the assignment changes. Now **choose Processing/Start Compilation** again, and you are ready to implementing your design into EP2C5Q208C8 FPGA with “Full compilation was successful” message window.

---

## 6.2.5 Simulation

In order to verify the designed circuit by simulation, a **Vector Waveform File** is needed defining input stimulus of each input ports. Choose **File/NEW**. When the **NEW** dialog box appears, select **Vector Waveform File** on the **Other Files** tab, then click **OK**.

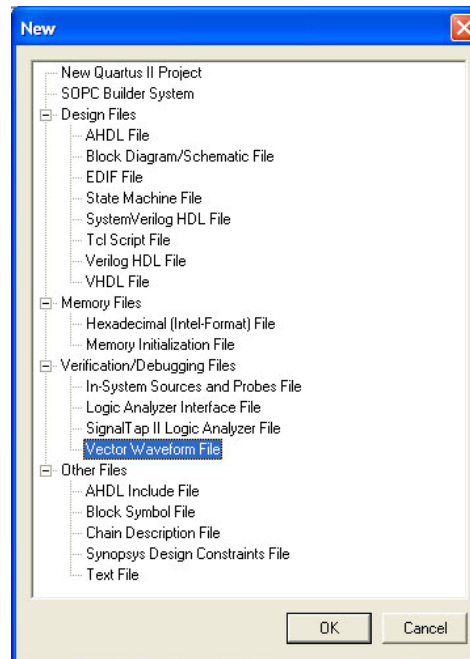


Fig. 6.2.18 **Other Files** Tab in **New** dialogue box for creating new **Vector Waveform File**.

On the created waveform editor of the Vector Waveform File, choose **Insert Node or Bus...** on the menu invoked by clicking the right button of your Mouse at the arrowed region.

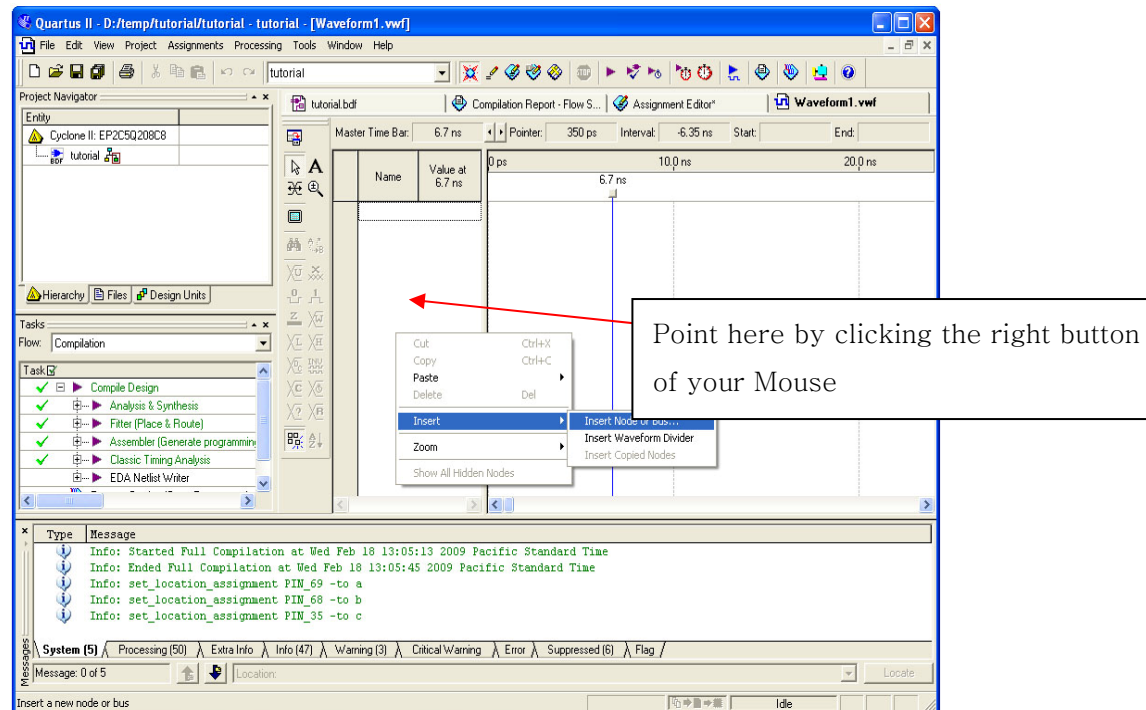


Fig. 6.2.19 A waveform editor of the Vector Waveform File..

---

Now you can see the **Insert Node or Bus** dialogue box as shown in Fig. 6.2.20. Click **Node Finder...**, and the Node Finder dialogue box opens as shown in Fig. 6.2.21.

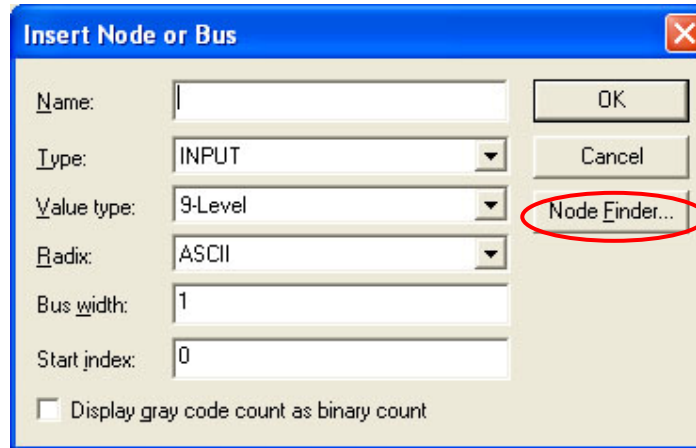


Fig. 6.2.20 **Insert Node or Bus** dialogue box

---

On the **Node Finder** dialogue box, click **List** button first and all the input and output ports will be shown on **Node Found** region. Clicking >> button selects all the nodes shown in **Node Found** Region. Click OK on the **Node Finder** then click OK on **Insert Node or Bus** dialogue box.

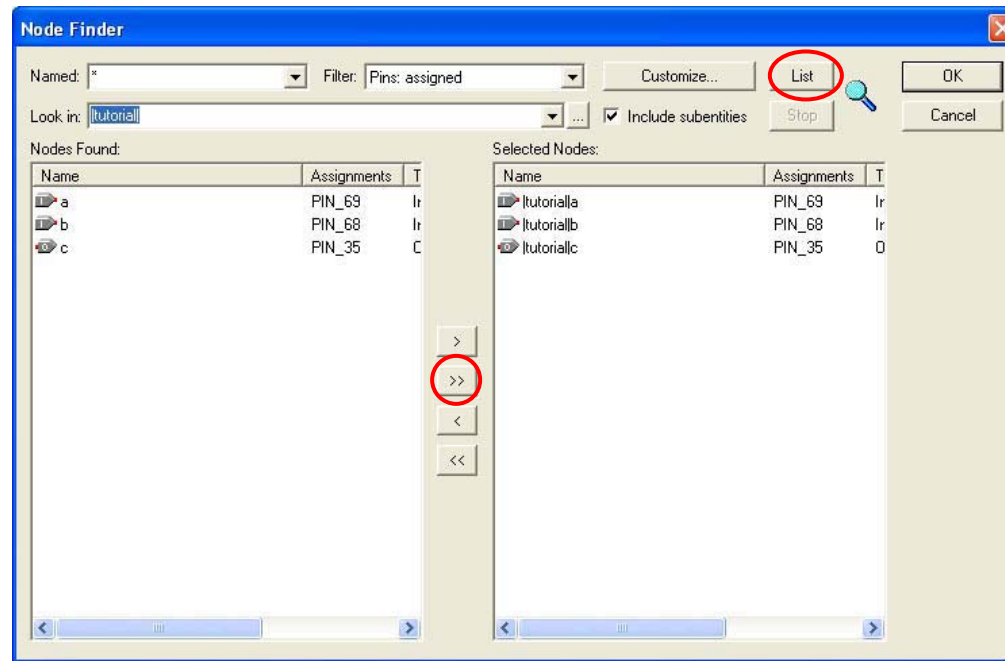


Fig. 6.2.21 Node Finder dialogue box

If all the added nodes are shown on the waveform editor, you can define input stimulus easily by dragging the time region where you want to change status of, and clicking control buttons for editing the wave form as shown in Fig. 6.2.22. Save this file as **tutorial.vwf**.

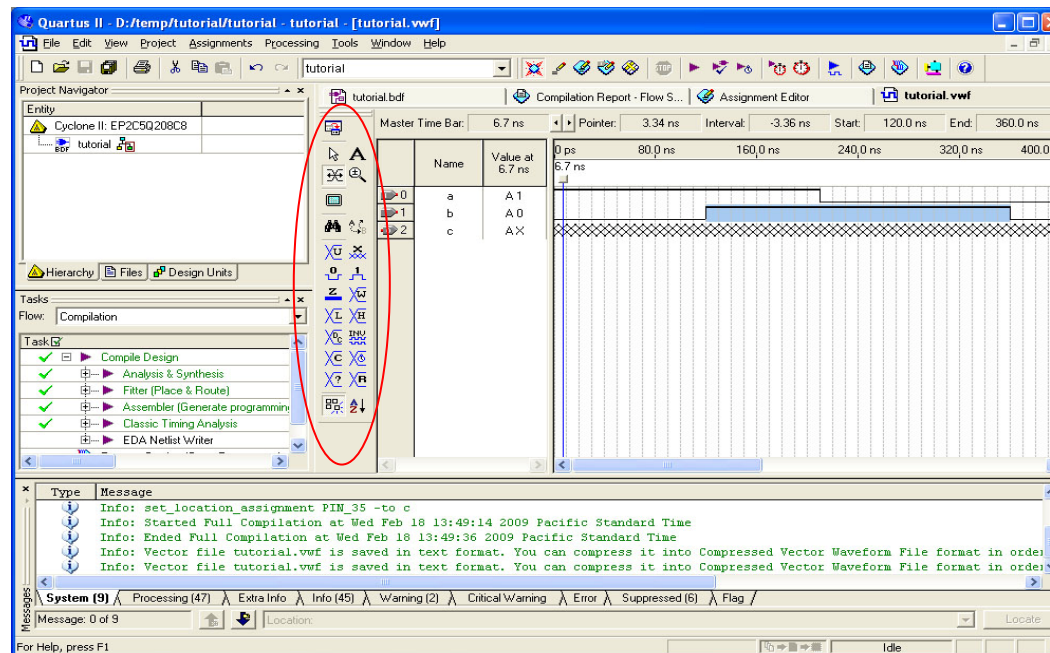


Fig. 6.2.22 Vector Waveform file after defining input stimulus.

Now choose **Processing/Start Simulation**, and you can find simulated output waveform on the **Simulation Waveform** window as shown in Fig. 6.2.23.

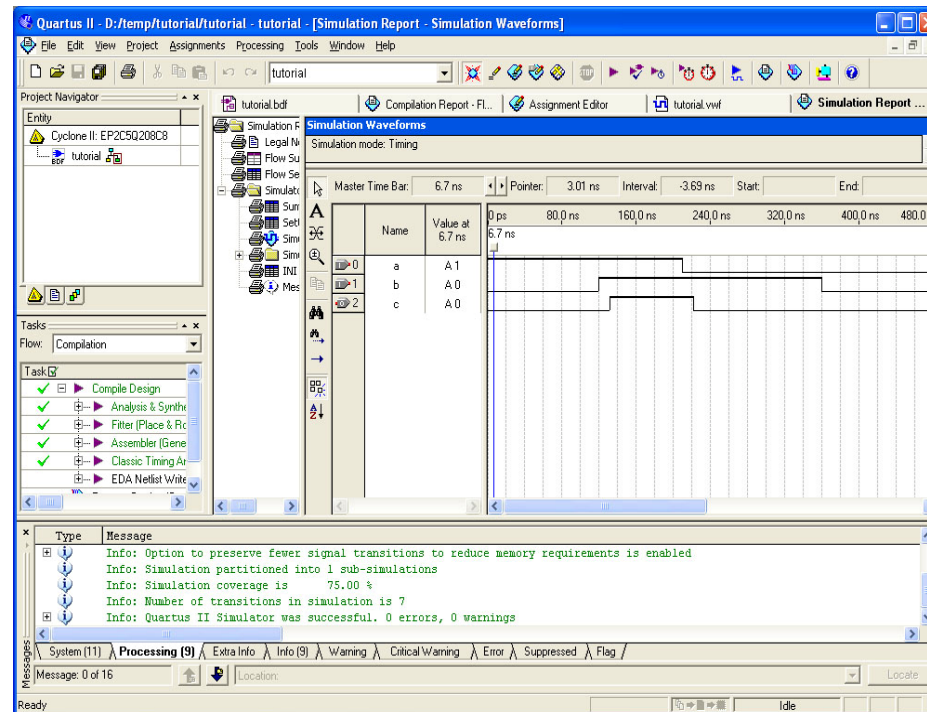


Fig. 6.2.23 Simulation result on the **Simulation Waveform** window.

## 6.2.6 Programming FPGA

The programmer allows you to use files generated by the Compiler to program and/or configure all Altera® devices supported by the Quartus® II software. You can program or configure the EP2C5Q208C8 device in JTAG mode.

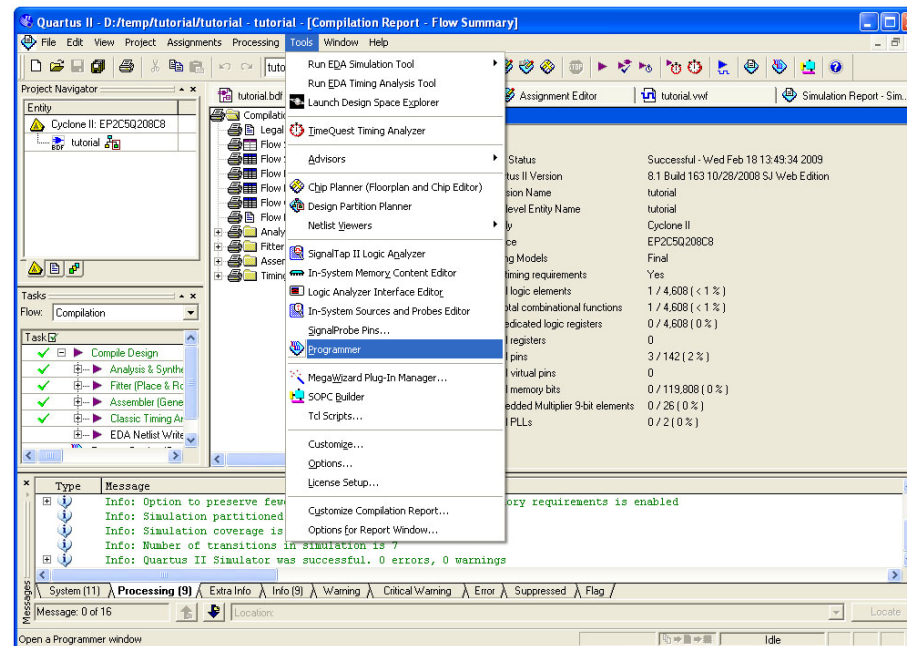


Fig. 6.2.24 Choosing **Programmer** in the **Tools** menu

Choose **Programmer** (Tools menu), A new CDF(Chain Device File) opens in the **Programmer** window automatically listing the **tutorial.sof** file as the current programming file

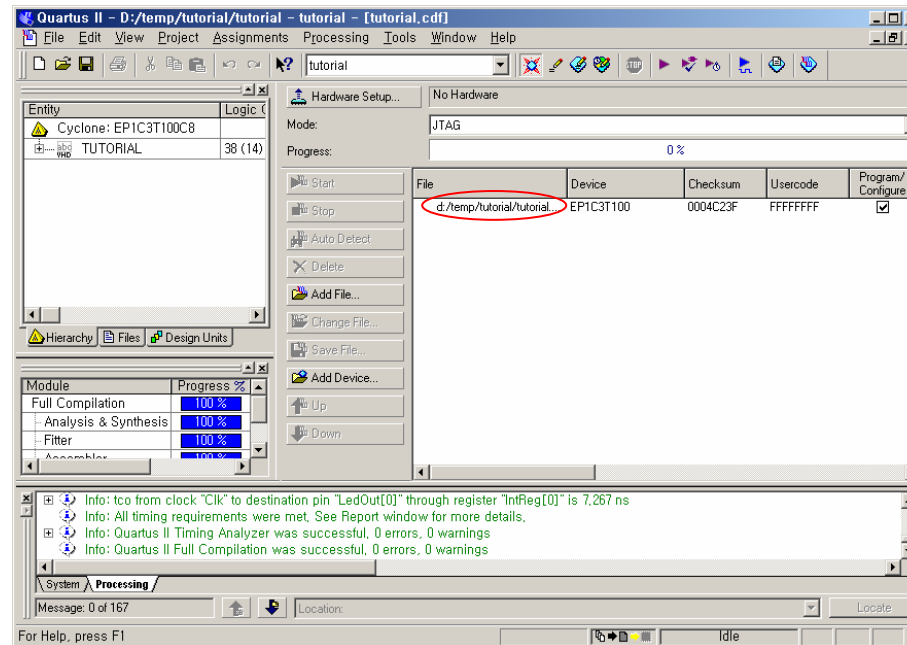
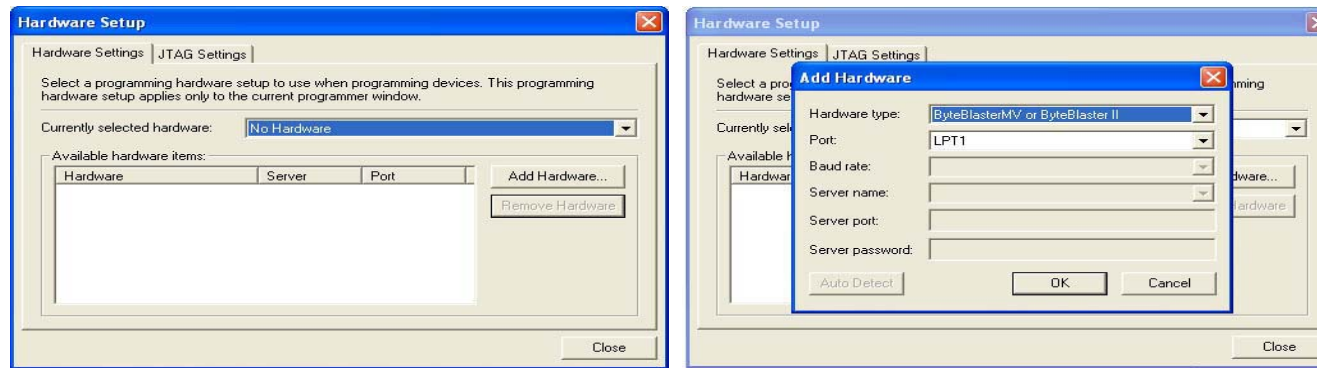


Fig. 6.2.25 Programmer window in the Tools menu

---

In the Mode list of the **Programmer** window, select **JTAG**. When the Programmer window is first invoked, “No Hardware” message would be shown at the right of the **Hardware Setup...** button. Then you need to set up programming hardware first. Click **Hardware Setup** and the **Hardware Setup** dialog box appears as shown Fig. 6.2.26(a)



(a)

(b)

Fig. 6.2.26 (a) **Hardware Setup** and (b) **Add Hardware** dialog box.

Now, click **Add Hardware** and the **Add Hardware** dialog Box appears as shown Fig. 6.2.26(b). In the **Hardware type** list, select **ByteBlasterMV or ByteBlasterII** and select **LPT1** in the Port list. Click **OK** in the **Add Hardware** dialog Box and click **Close** in the **Hardware Setup** dialog Box. Then save the current hardware setup information by choosing **Save** (File menu).

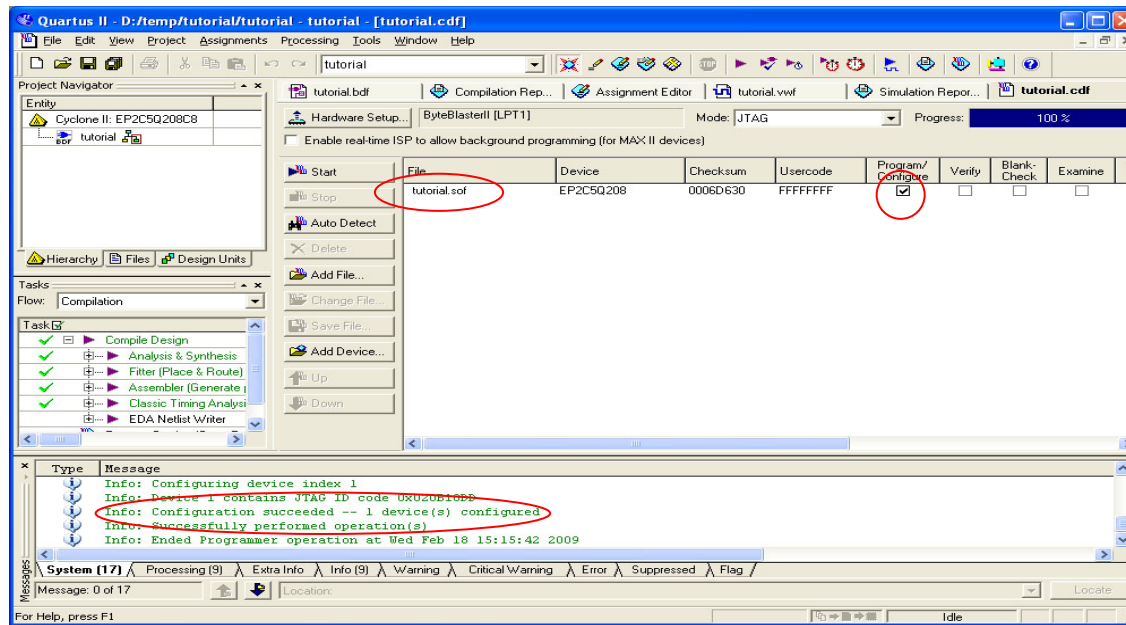


Fig. 6.2.27 Programmer window when the programming procedure was completed

Select **Program/Configure** check box of the **tutorial.sof** file, and click **Start**. “Configuration succeeded -- 1 device(s) configure” message in the **System** tab, shows the success in the programming the FPGA device. On finishing programming, the EP2C5Q208C8 FPGA device on the QB-EP2C208-BK board start to operate as you designed. LED1 will be on and off depending

---

on the status of Key1 and Key2, based on the 2-input AND gate logic table. The design procedure in this tutorial is not optimal but simple for novices. In order to learn the optimal design procedure, please refer to Quartus II handbook at the Altera's Website below;

[http://www.altera.com/literature/hb/qts/quartusii\\_handbook.pdf](http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf)

### 6.3 A VHDL design example

The tutorial in the section 6.3 shows a design procedure of the digital circuit to turn 8 LEDs on and off in regular sequence from left to right. The input and output device used in this design are designated by red arrows in Fig. 6.3.1.

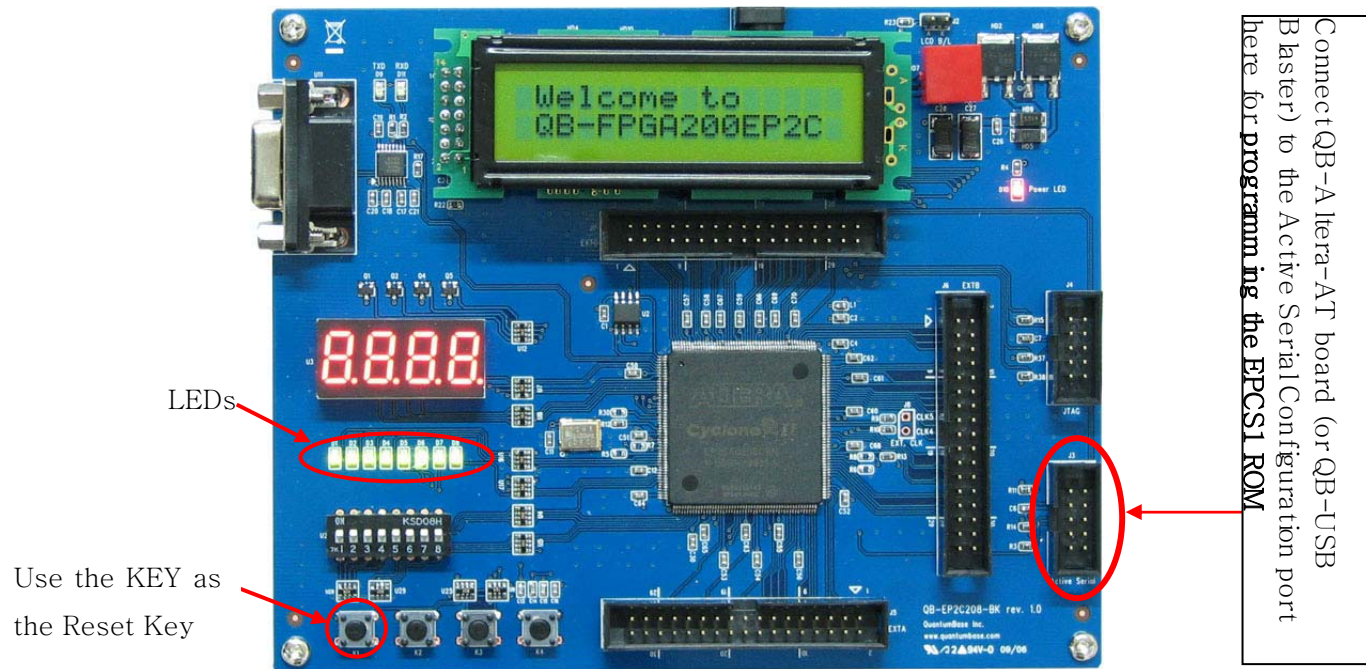


Fig. 6.3.1 The Key and LEDs used in this example

---

### 6.3.1 Creating Your own Project

Double clicking Quartus II web edition icon on your desktop starts the software tool. Fig. 6.3.2 shows the Quartus II graphical user interface as it appears when you first start the software.

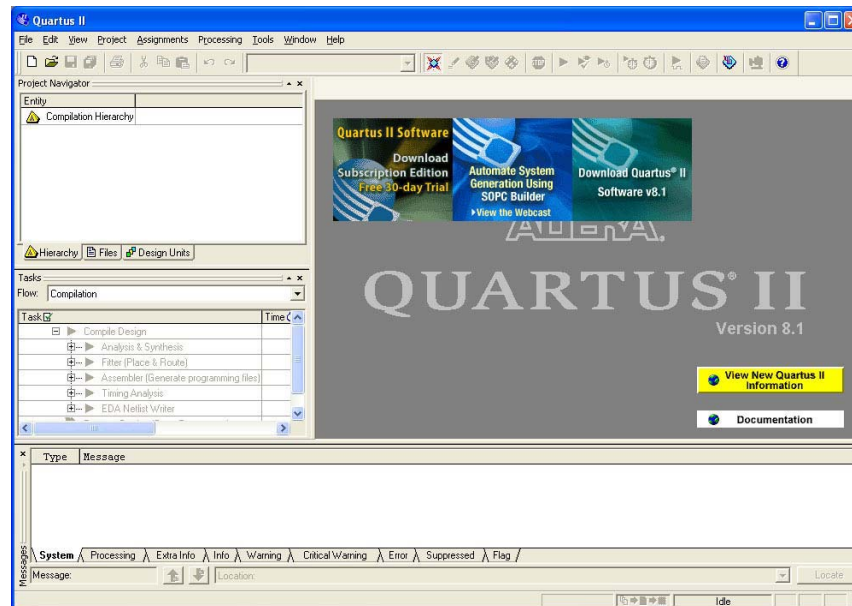


Fig. 6.3.2 Quartus II graphical user interface

In order to create your own project, choose File/New Project Wizard as shown in Fig. 6.3.3

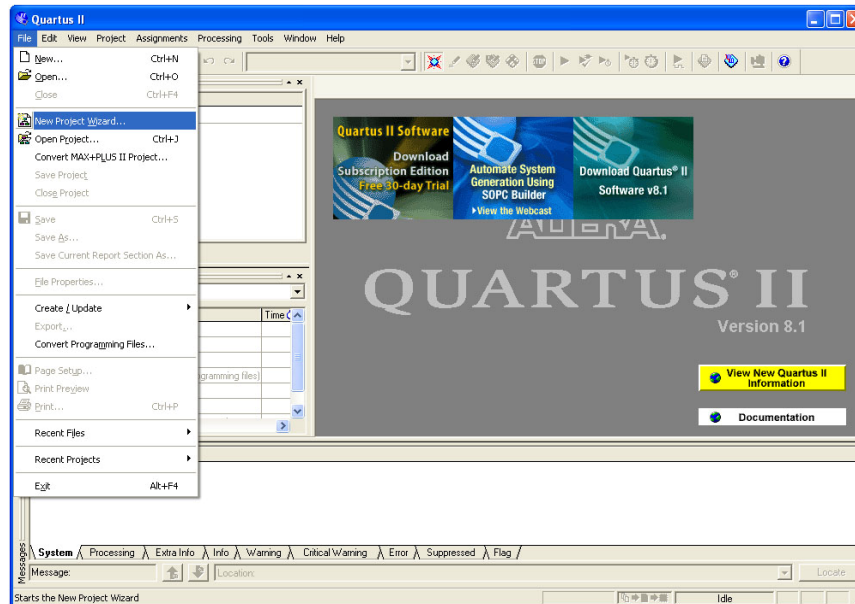


Fig. 6.3.3 Choosing File/New Project Wizard

---

This invokes the “New Project Wizard:Directory, Name, Top-Level Entity” dialog box as shown Fig 6.3.4

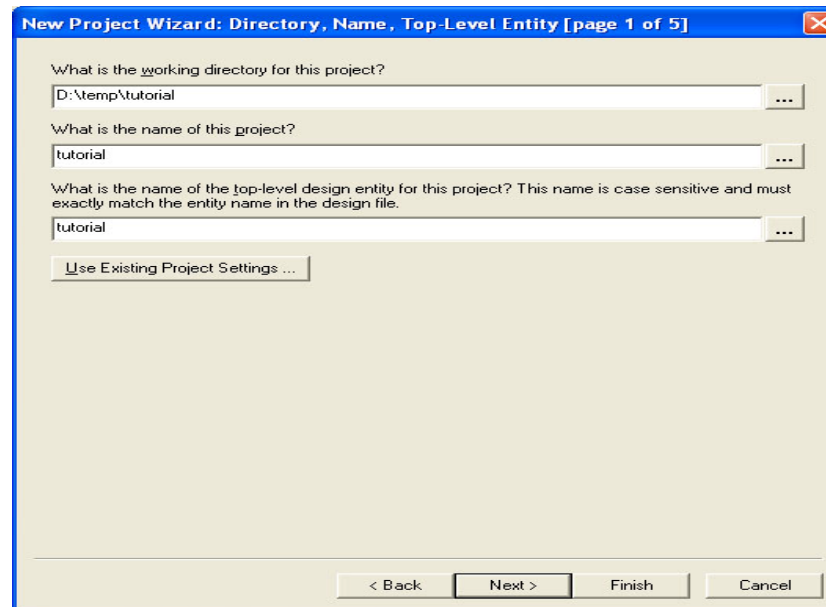


Fig. 6.3.4 The first dialog box regarding “New Project Wizard”

Type the directory name in the working directory box or select the directory with **Browse(...)**. Type a project name you are to create in the project name box. In the top-level design entity box, top level entity name should be shown and it should be exactly the same was the entity name in the design file. Let’s fill in the 3 boxes as shown in Fig. 6.3.4 in this tutorial course and click **Next..**

The **Add Files** page of the **New Project Wizard** appears. Since **tutorial** is a new project, there are no files to add to this project yet, so just click **Next**.

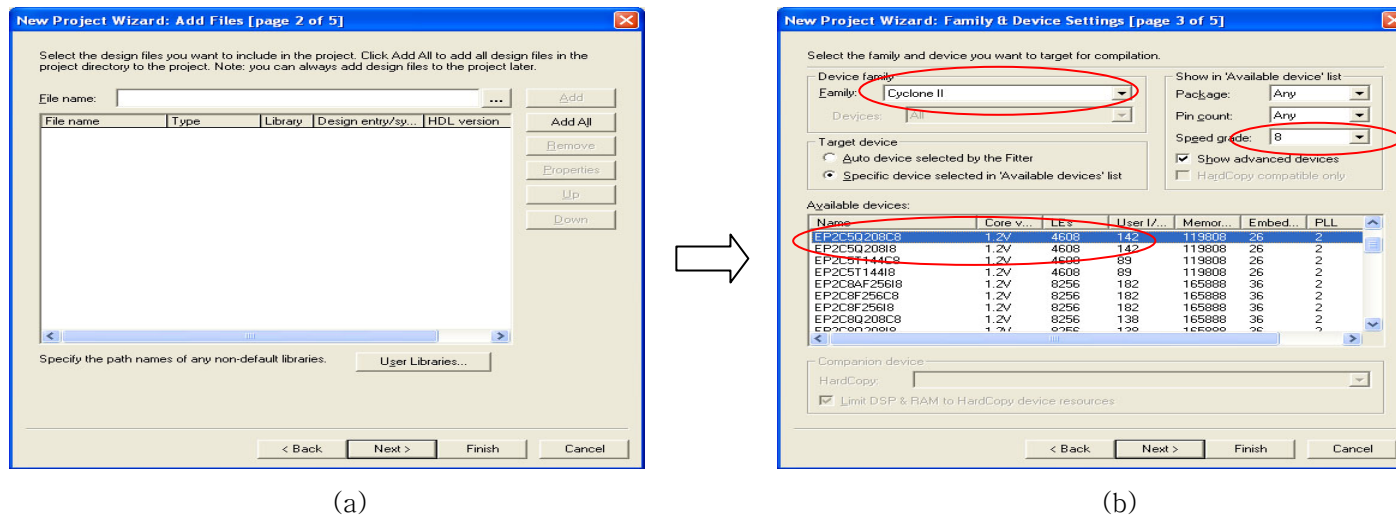


Fig. 6.3.5 (a)Add Files and (b)Family & Devices Settings pages of the New Project Wizard

Now the **Family & Device Settings** page of the **New Project Wizard** appears then. In the **Family** list, select **Cyclone**. In the **Speed grade** list of the **Filters** group, select **8**, and then select **EP2C5Q208C8** in the **Available devices** box

To accept the default setting for the remaining wizard prompts and create the project, click **Next** and **Finish** in turn. The project is now created. The top-level design entity name appears in the **Hierarchy** tab of the Project Navigator window.

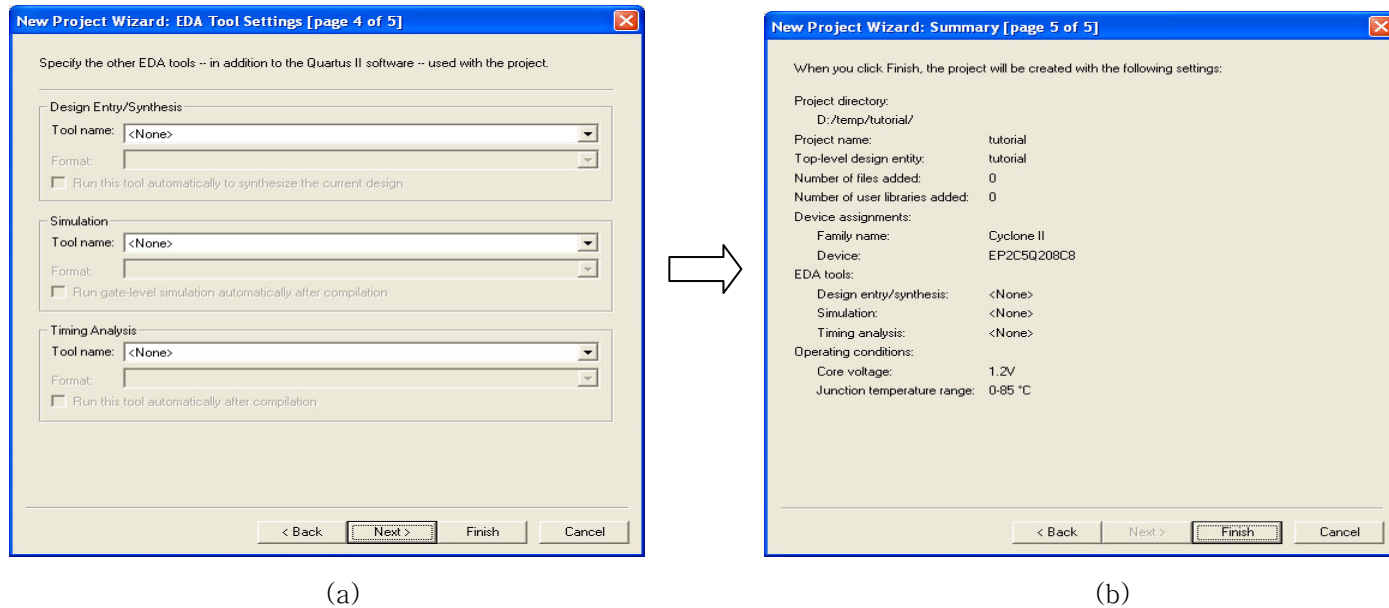


Fig. 6.3.6 (a)EDA Tool Settings and (b)Summary pages of the New Project Wizard

---

### 6.3.2 Setting Unused pins

As mentioned in Chapter 2, users need to give care to unused pin status because the FPGA can be damaged seriously by unintentional over current if both of the pin and a connected device are set as output and their output status are different from each other. Reserving the unused pins as input and making it tri-stated is the very common method to address the problem. Thus users can make the unused pins tri-stated by setting a option in Quartus II as shown below.

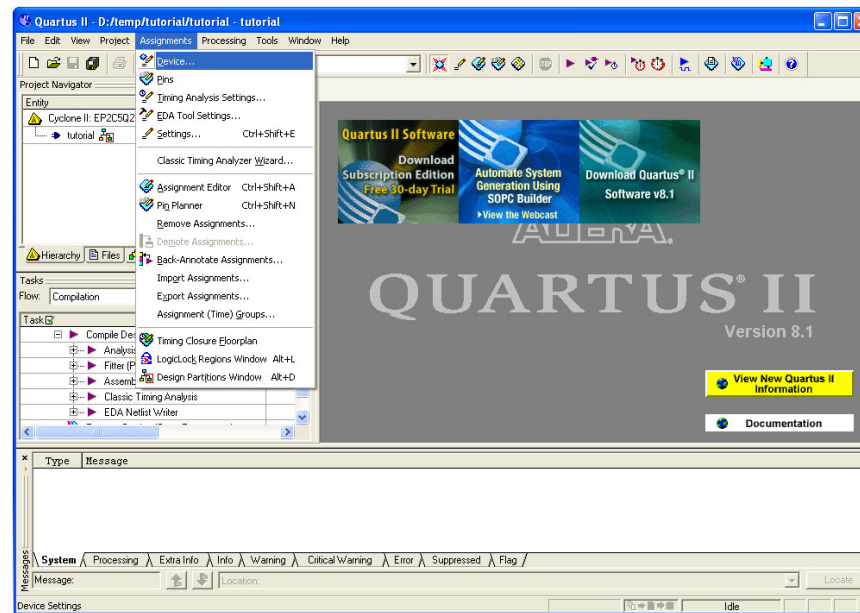
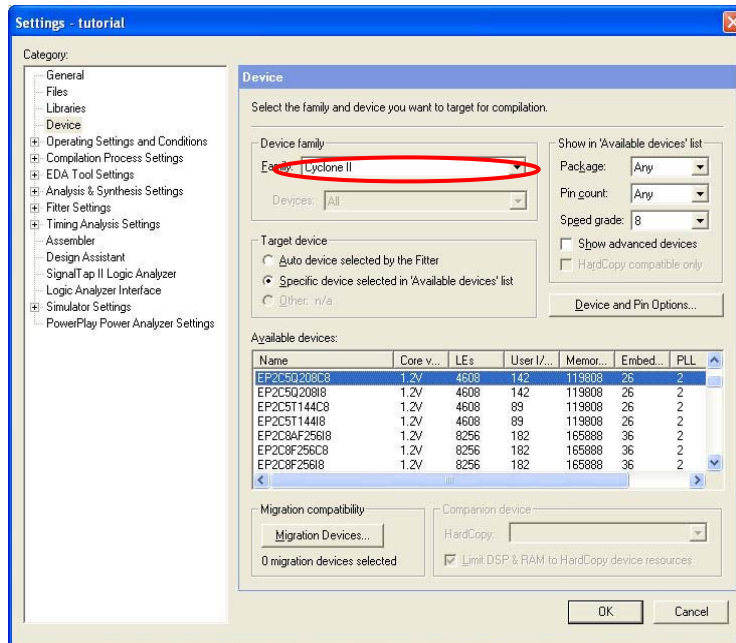
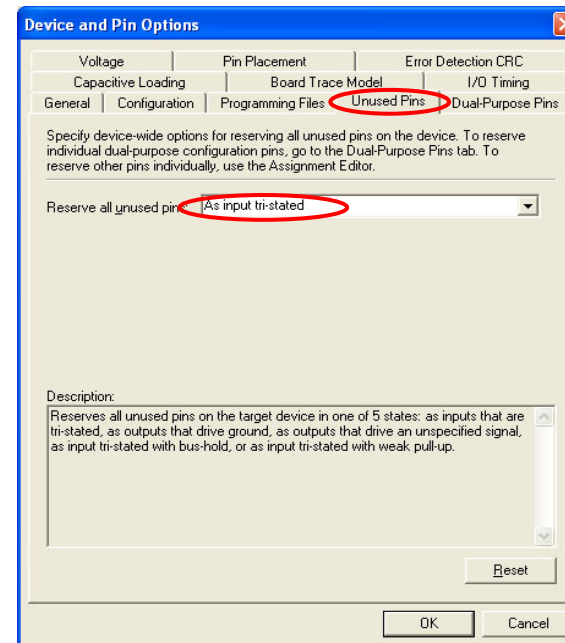


Fig. 6.3.7 Selecting Assignment menu

As shown in Fig. 6.3.7, choose **Device...** (**Assignment menu**). By clicking Device & Pin Options button as shown in Fig 6.3.8(a), user can invoke **Device & Pin Options** dialogue box of Fig. 6.3.8(b). As shown in Fig. 6.3.8(b), select **As inputs tri-stated** in Unused Pins Tab.



(a)



(b)

Fig. 6.3.8 (a) Device Setting Window (b) Device & Pin Options dialogue box

---

### 6.3.3 Create a New VHDL Design File

In this step you create a new VHDL file called **tutorial.vhd**. this file is the top-level design entity of the **tutorial** project. Choose **File/NEW**. When the **Device Design Files** tab of the **NEW** dialog box appears, select VHDL File and click **OK**.

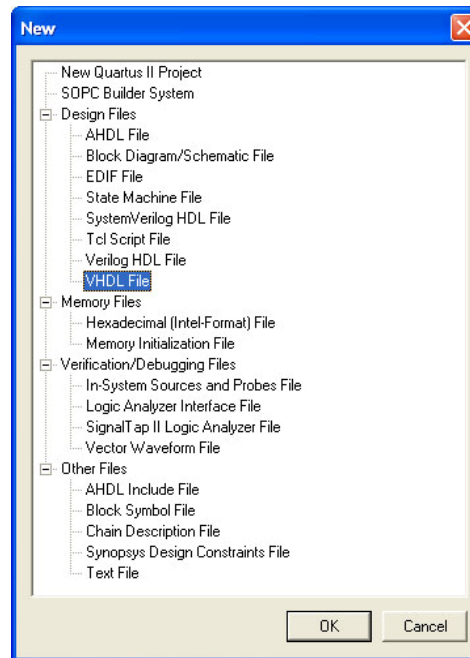
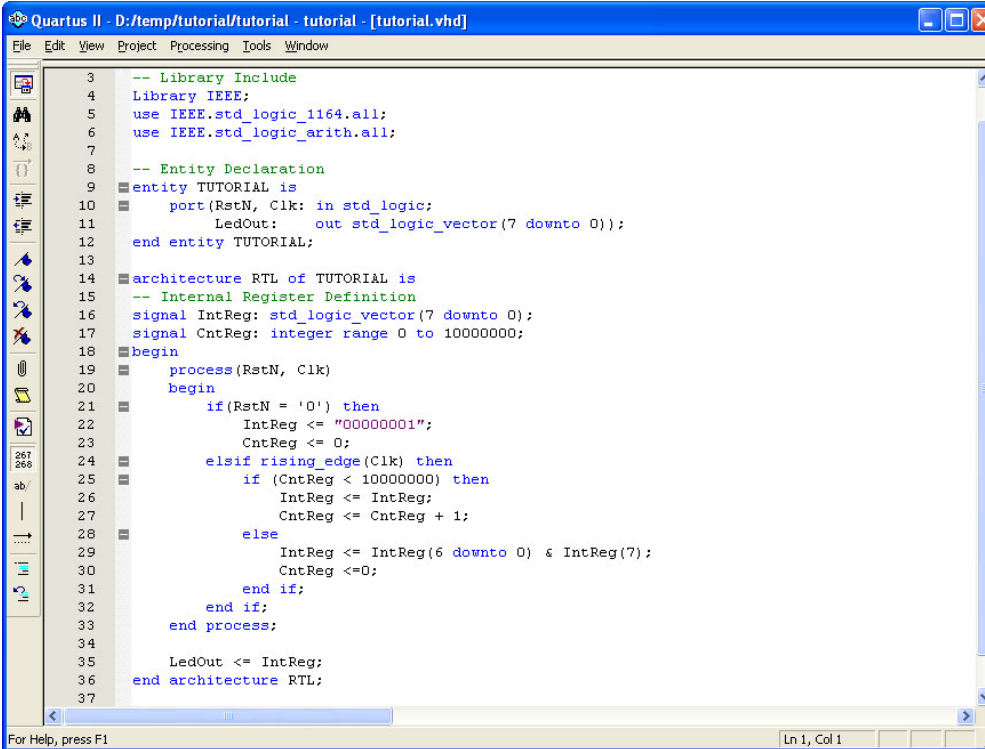


Fig. 6.3.9 Device Design Files tab of the New dialogue box

In the invoked editor, type the VHDL code as shown Fig 6.3.10, and save it as the file named **tutorial.vhd** by choosing **Save** (File menu)



```
3  -- Library Include
4  Library IEEE;
5  use IEEE.std_logic_1164.all;
6  use IEEE.std_logic_arith.all;
7
8  -- Entity Declaration
9  entity TUTORIAL is
10     port (RstN, Clk: in std_logic;
11           LedOut: out std_logic_vector(7 downto 0));
12 end entity TUTORIAL;
13
14 architecture RTL of TUTORIAL is
15     -- Internal Register Definition
16     signal IntReg: std_logic_vector(7 downto 0);
17     signal CntReg: integer range 0 to 10000000;
18 begin
19     process (RstN, Clk)
20     begin
21         if (RstN = '0') then
22             IntReg <= "00000001";
23             CntReg <= 0;
24         elsif rising_edge(Clk) then
25             if (CntReg < 10000000) then
26                 IntReg <= IntReg;
27                 CntReg <= CntReg + 1;
28             else
29                 IntReg <= IntReg(6 downto 0) & IntReg(7);
30                 CntReg <= 0;
31             end if;
32         end if;
33     end process;
34
35     LedOut <= IntReg;
36 end architecture RTL;
37
```

Fig. 6.3.10 VHDL code example for the tutorial

Now **choose Processing/Start Compilation** so that Quartus II makes full compilation of the VHDL code. If there are any errors, correct errors and recompile it.

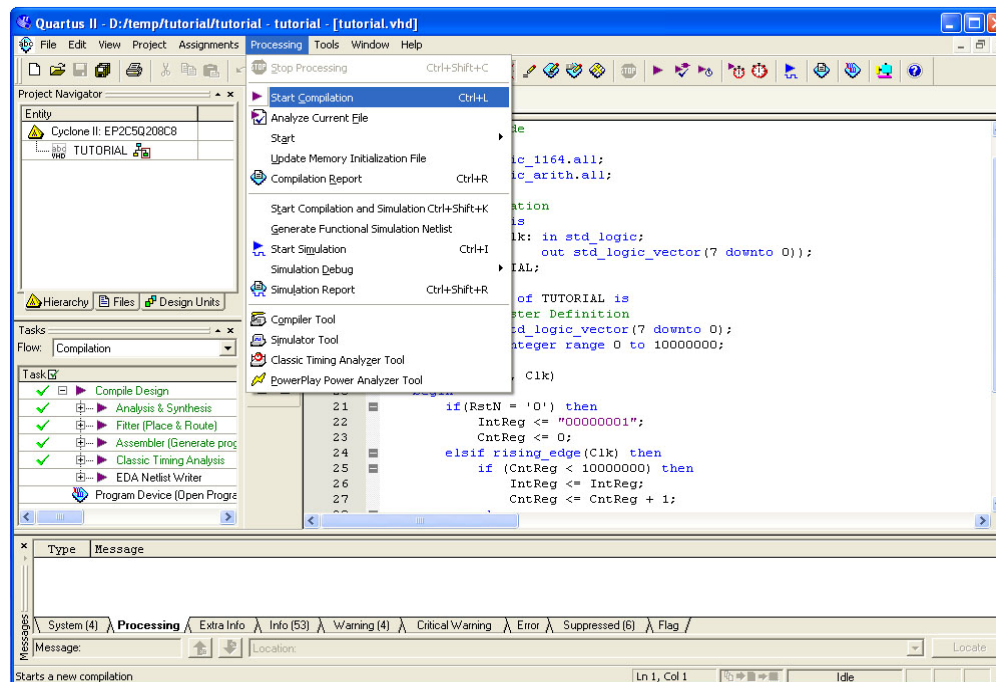


Fig. 6.3.11 Make full compilation of the Processing menu

“Full compilation was successful” message window appears when there are no errors in the VHDL code.

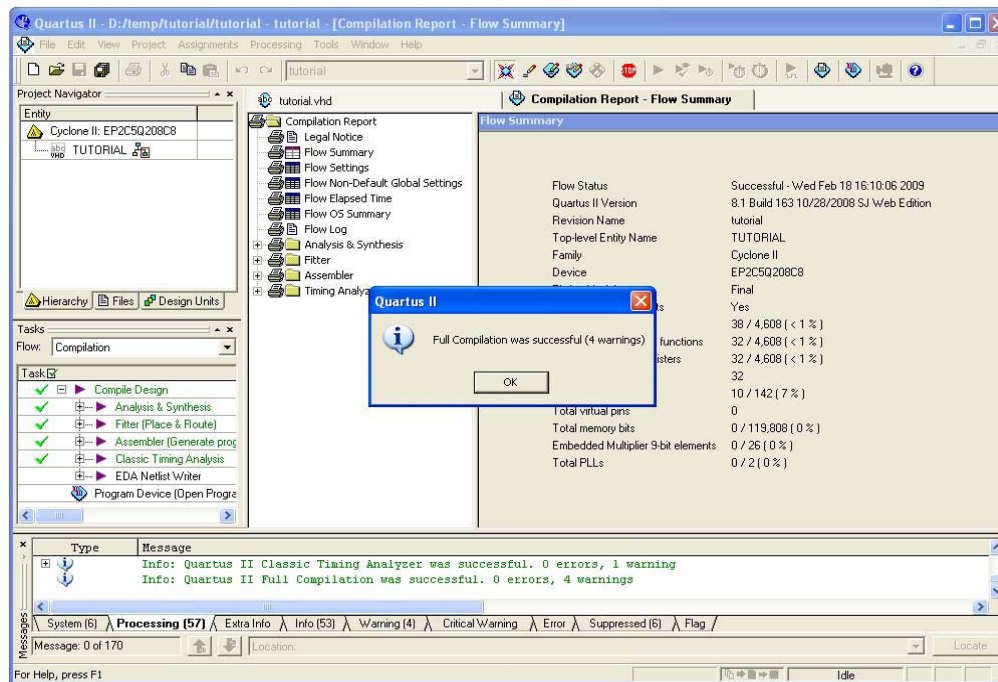


Fig. 6.3.12 Captured image when the full compilation was successful

---

### 6.3.4 Pin Assignment

Since there are no errors in your design now, you should assign FPGA pins to physically connected peripheral resources. Referring to the entity definition in the tutorial.vhd, you can notice that RstN, Clk, and 8bit LedOut port should be assigned.

*Since 50MHz external clock is supplied to CLK2 (Pin number: 27) of the EP2C5Q208C8 FPGA device, you can assign pin 27 to the input port Clk of the entity.* Then assign pin 69 to input port RstN of the entity, pin 45 to LedOut[7], pin 44 to LedOut[6], pin 43 to LedOut[5], pin 41 to LedOut[4], pin 40 to LedOut[3], pin 39 to LedOut[2], pin 37 to LedOut[1], and pin 35 to LedOut[0].

Table 6.3.1 Selected port map for this tutorial

FPGA Pin#	Board Connection	FPGA Pin#	Board Connection	FPGA Pin#	Board Connection
23	CLK0	45	LED8	40	LED4
27	CLK2	44	LED7	39	LED3
69	KEY1	43	LED6	37	LED2
		41	LED5	35	LED1

Next page shows the way to assign pins in the Quartus II software tool.

Choose **Pins** (Assignment menu).

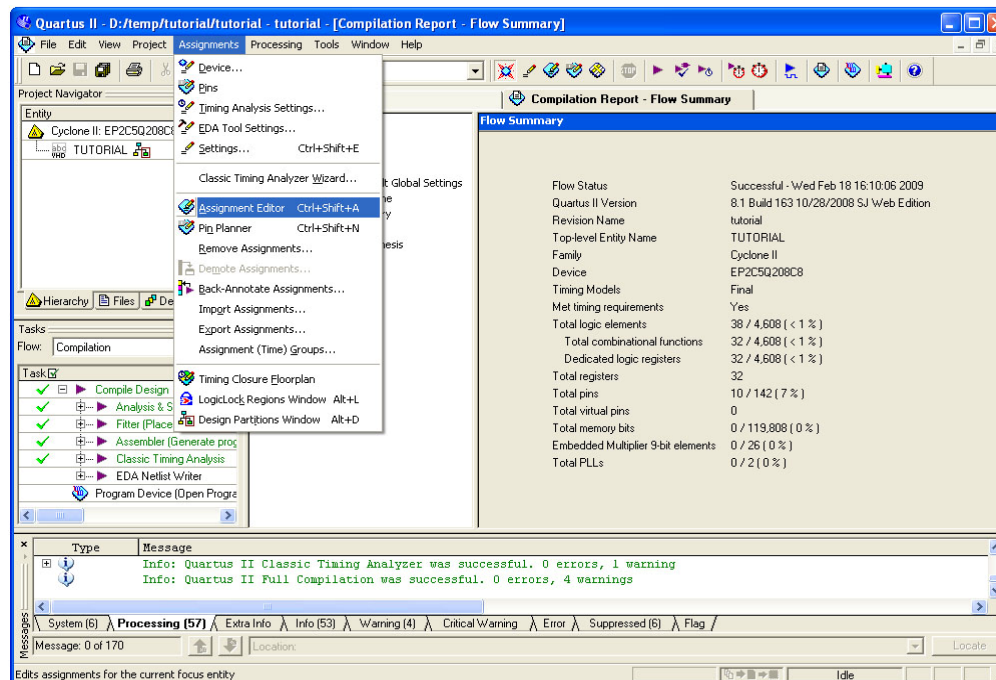


Fig. 6.3.13 Choosing **Assignments/Pin** for the pin assignment

Then the Assignment Editor appears with the Pin assignment category selected. In the Assignment Editor, double-click the **To** cell and select the **Clk** input node name.

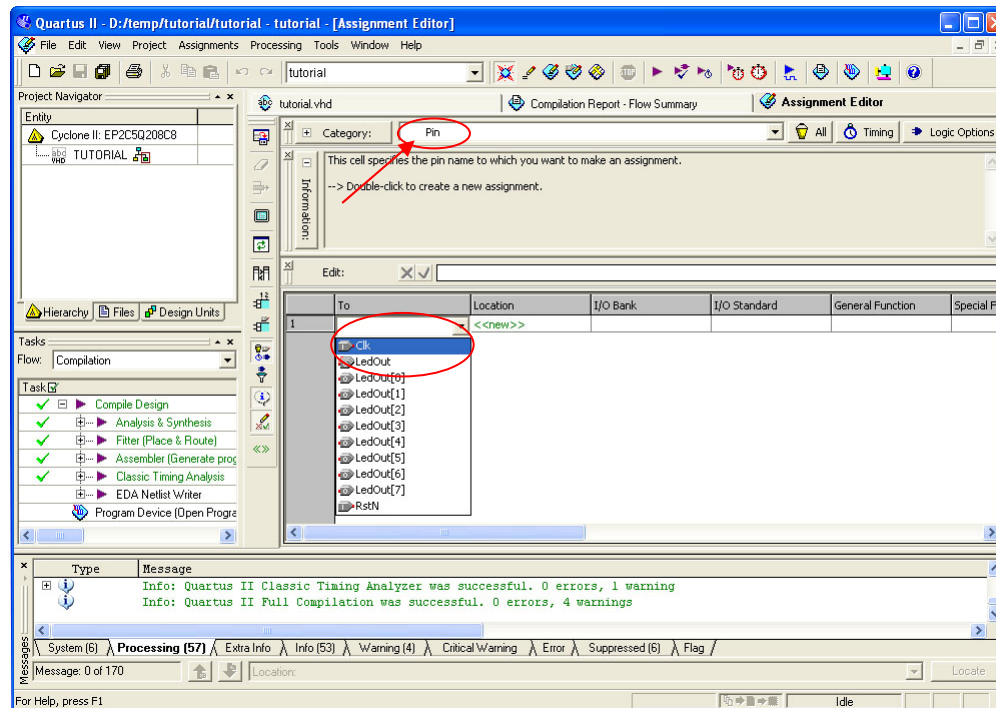


Fig. 6.3.14 Select port in the Assignment Editor

In the assignment Editor, double-click the Location cell and scroll down to select pin 27, dedicated clock I/O pin.

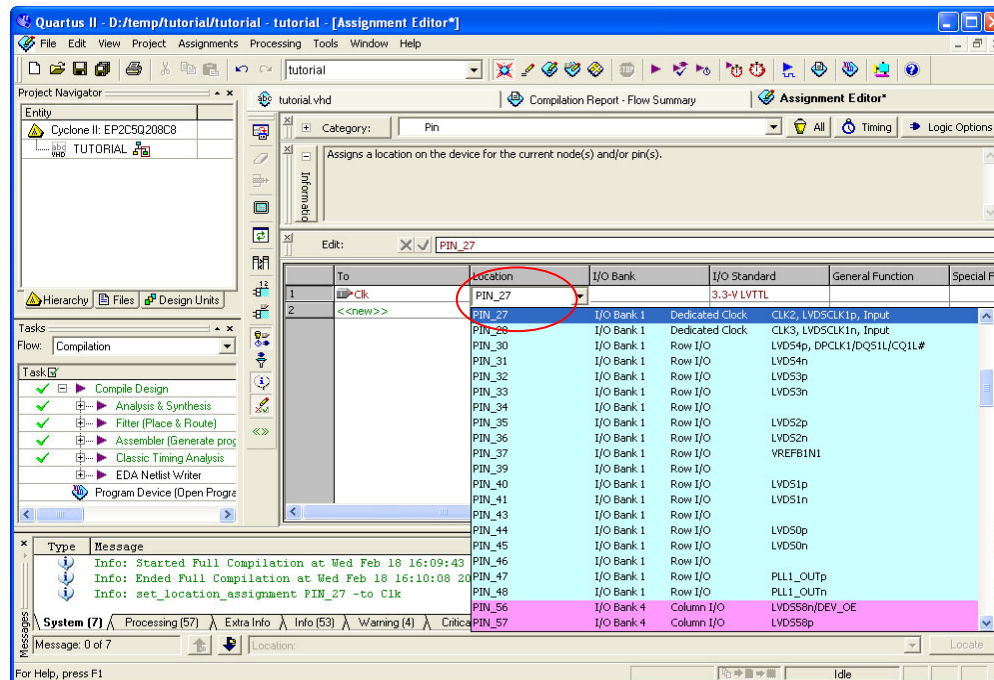


Fig. 6.3.15 Select a pin in the Assignment Editor

Keep assigning remaining pins referring to Table 6.3.1

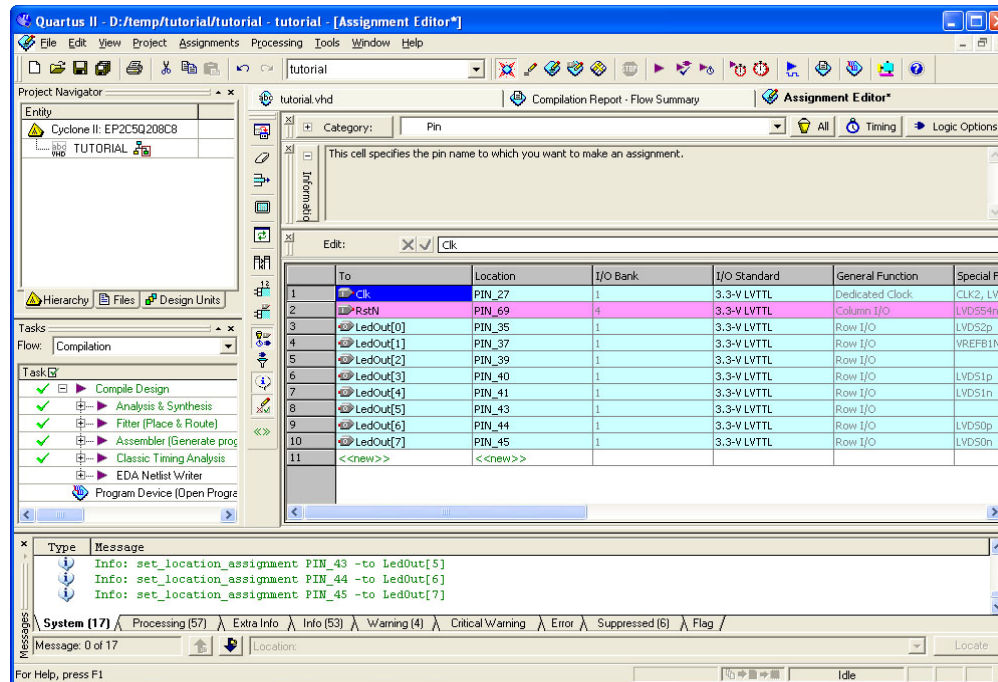


Fig. 6.3.16 Assignment Editor when all the pins are completely assigned.

After the pin assignment is completed, close the Assignment Editor, saving the assignment changes. Now **choose Processing/Start**

---

**Compilation** again, and you are ready to implementing your design into EP2C5Q208C8 FPGA with “Full compilation was successful” message window.

### **6.3.5 Simulation**

Simulation procedure is the same as that of the schematic based design as shown in the section 6.2.5. Thus the simulation procedure is not included intentionally.

---

### 6.3.6 Programming EPCS1 Configuration ROM

In section 6.2, we show the programming procedure which configures FPGA using JTAG. Port. FPGA is the SRAM based device and the configuration data get erased when the power switch is turned off. However, if we program the EPCS1 configuration ROM installed on the QB-EP2C208-BK board, we can make the FPGA configured automatically by the EPCS1 ROM every time the power supplied to the board. In order to do this, connect the flat cable of the QB-Altera-AT board to the **Active Serial Conf.** port marked in Fig. 6.3.17. ( Please refer to Appendix of this document to learn how to set the EPCS1 as a configuration ROM. )

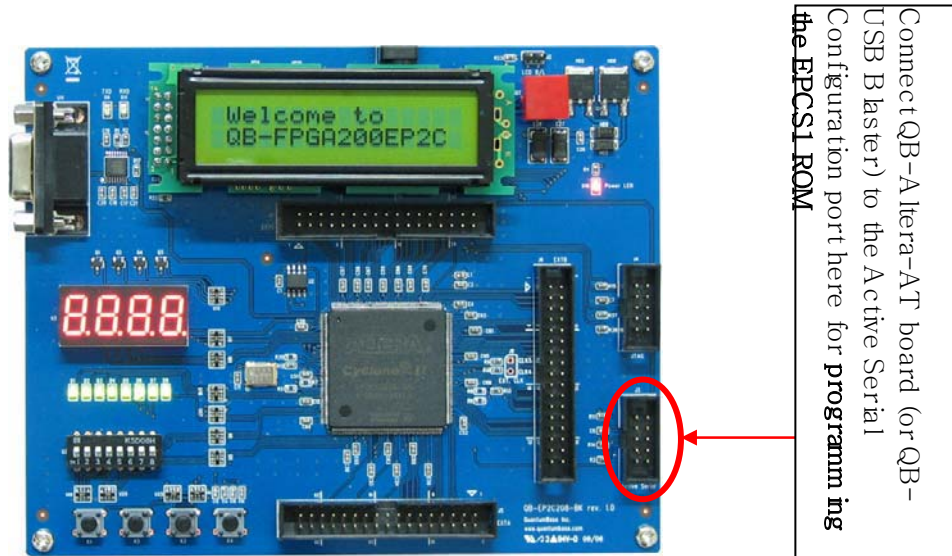


Fig. 6.3.17 Connection of the QB-Alter-AT ByteblasterII board for programming EPCS1 ROM

The programmer allows you to use files generated by the Compiler to program and/or configure all Altera® devices supported by the Quartus® II software. You can program or configure the EPCS1 device in Active Serial mode.

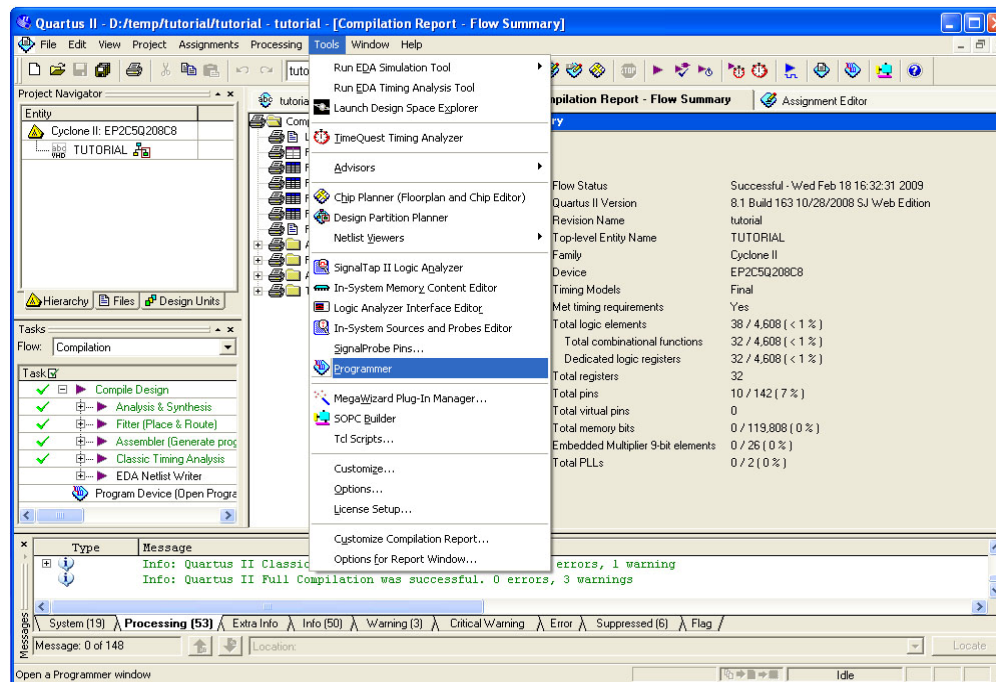


Fig. 6.3.17 Choosing **Programmer** in the **Tools** menu

Choose Programmer (Tools menu), A new CDF opens in the **Programmer** window automatically listing the **tutorial.sof** file as the current programming file. Since .sof file is for FPGA configuration, we need to change the programming file for the EPCS1 device.

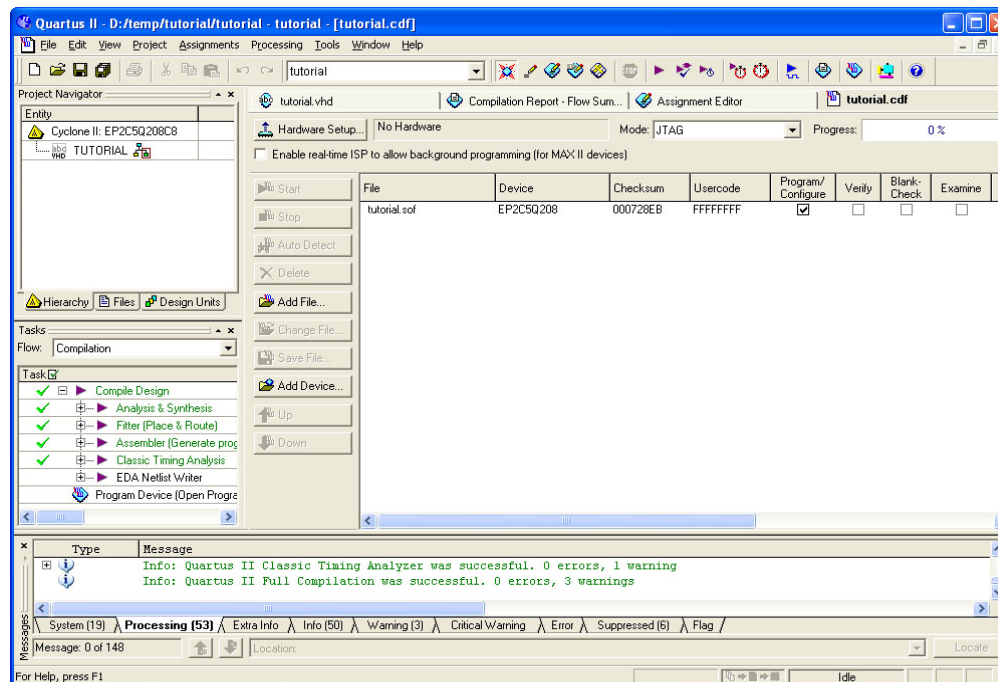


Fig. 6.3.18 Programmer window in the Tools menu

In the Mode list of the **Programmer** window, select **Active Serial Programming** and click **Yes** on the following dialogue box. Click **Add File** and select **tutorial.pof** as the programming file.

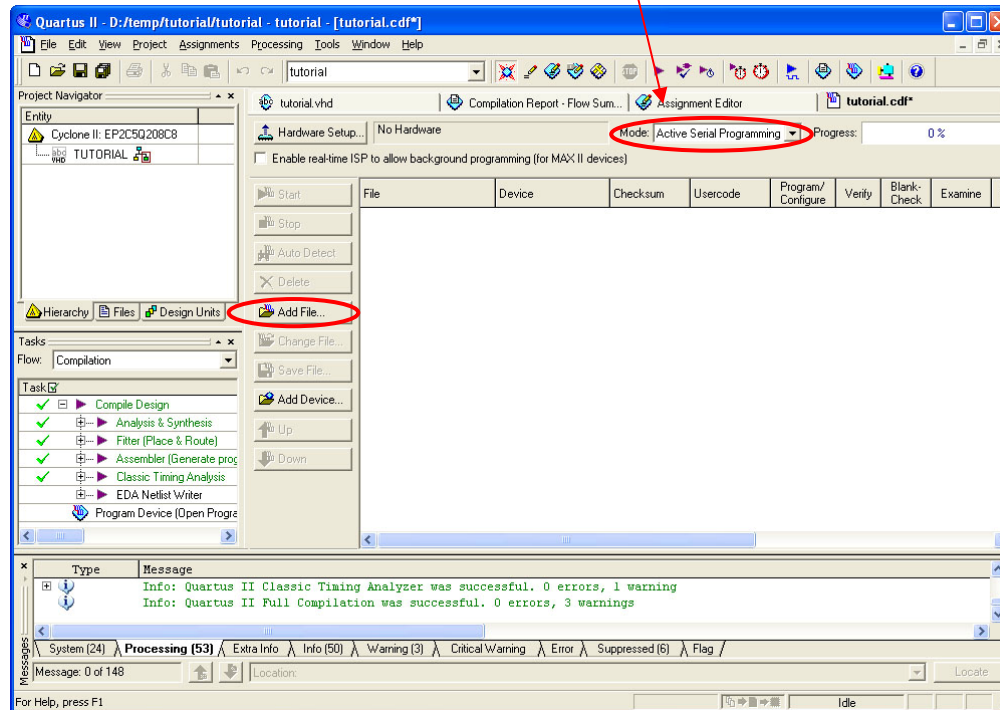
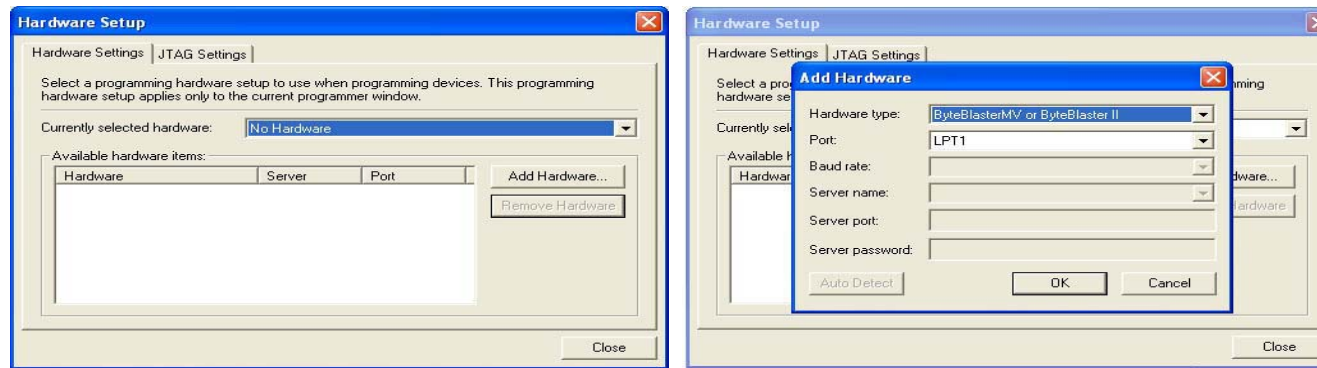


Fig. 6.3.19 Programmer window after the mode change

When the Programmer window is first invoked, “No Hardware” message would be shown at the right of the **Hardware Setup...** button. Then you need to set up programming hardware first. Click **Hardware Setup** and the **Hardware Setup** dialog box appears as shown Fig. 6.3.20(a)



(a)

(b)

Fig. 6.3.20 (a) **Hardware Setup** and (b) **Add Hardware** dialog box.

Now, click **Add Hardware** and the **Add Hardware** dialog Box appears as shown Fig. 6.20(b). In the **Hardware type** list, select **ByteBlasterMV or ByteBlasterII** and select **LPT1** in the Port list. Click **OK** in the **Add Hardware** dialog Box and click **Close** in the **Hardware Setup** dialog Box. Then save the current hardware setup information by choosing **Save** (File menu).

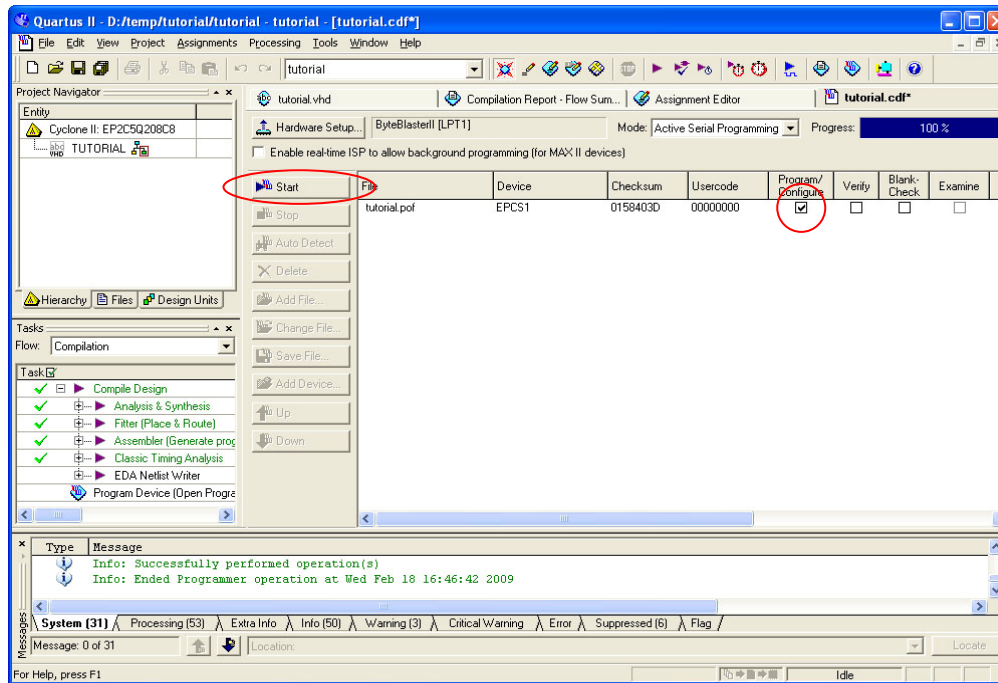


Fig. 6.3.21 Programmer window when the programming procedure was completed

Select **Program/Configure** check box of the **tutorial.pof** file, and click **Start**. “Successfully performed operation” message in the **System** tab, shows the success in the programming the FPGA device. **Remove** the flat cable of the QB-Altera-AT board from the

QB-EP2C208-BK board and the 8 LEDs, from LED1 to LED8 on the QB-EP2C208-BK board will start to turn on and off in turn. Let's turn off the power switch of the board and turn it on again. You can see the FPGA works same as before.

Although a simulation procedure is not shown in this tutorial, it is very important procedure for a hardware designer. Thus users must learn how to make a simulation of their own designed hardware referring to QuartusII handbook. Moreover, SignalTapII embedded logic analyzer of the QuartusII software is fully available in the QB-EP2C208-BK board. Thus you can observe the signals of the designed hardware easily in the QB-EP2C208-BK board as shown in Fig. 6.3.22.

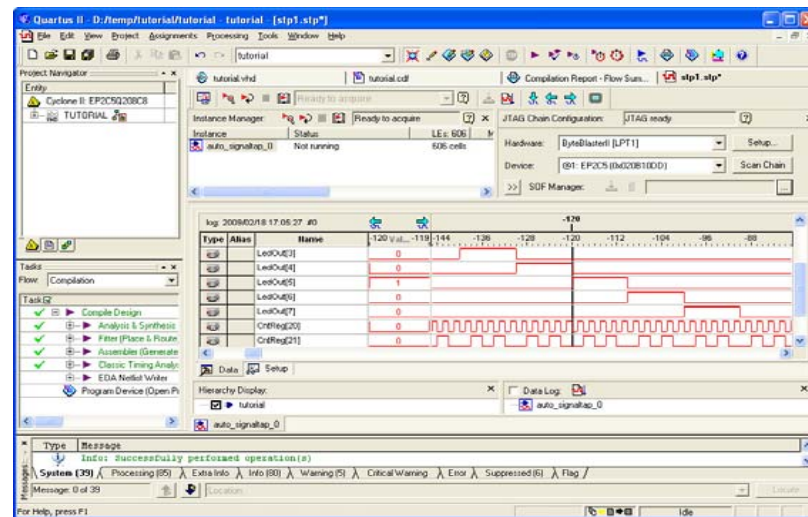


Fig. 6.3.22 Real time signal observation with the embedded **SignalTap II** Logic Analyzer

---

The Quartus II handbook by Altera will be helpful to you to learn the **SignalTap II** logic analyzer.

---

## 7 Useful Links

- How to use Quartus II
  - ✓ [http://www.altera.com/literature/manual/intro\\_to\\_quartus2.pdf](http://www.altera.com/literature/manual/intro_to_quartus2.pdf)
  - ✓ [http://www.altera.com/literature/hb/qts/quartusii\\_handbook.pdf](http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf)
  
- Cyclone II Device specification
  - ✓ [http://www.altera.com/literature/hb/cyc2/cyc2\\_cii5v1.pdf](http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1.pdf)
  
- Free HDL tutorial site
  - ✓ <http://www.doulos.com/knowhow/>
  - ✓ <http://www.vhdl-online.de/tutorial/>
  
- Recommended Books
  - ✓ Douglas J. Smith, “Hdl Chip Design: A Practical Guide for Designing, Synthesizing & Simulating Asics & Fpgas Using Vhdl or Verilog,” Doone Publications, 1996
  - ✓ Bob Zeidman, and Robert M. Zeidman, “Verilog Designer’s Libaray,” Prentice Hall, 1999
  - ✓ Weng Fook Lee, “VHDL Coding and Logic Synthesis with Synopsys,” Academic Press, 2000
  - ✓ Roth, Charles H., Jr, “Digital System Design Using VHDL”, Thomson Learning, 1998
  - ✓ Wakerly, John F., “Digital Design : Principles and Practices” Prentice Hall, 1990

---

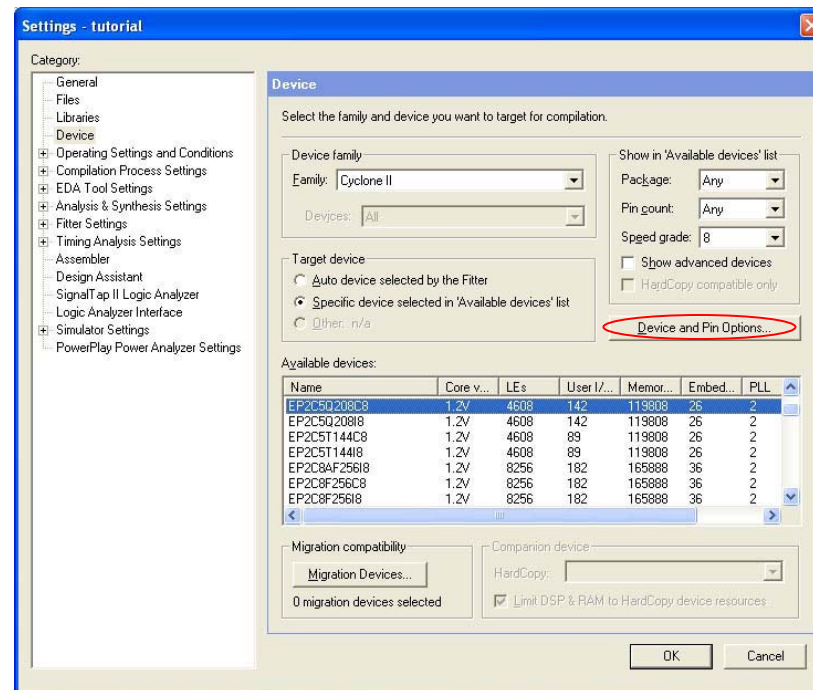
## 8 Appendix

As you can see in QB-EP2C208-BK Main Board feature table on page 7 in this document, the QB-EP2C208-BK board has a EPCS1 configuration ROM. In order to make Quartus II generate proper configuration file for EPCS1, you should select the proper configuration ROM (EPCS1) for your configuration ROM.

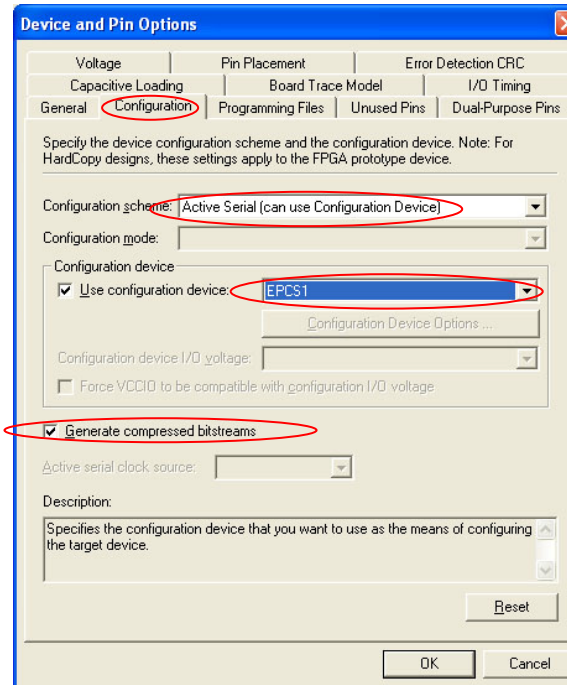
In order to do this, please choosing Assignment/Device... on the menu as shown below.



Then you can see the device setting dialog box as shown below.



Now please click the **Device and Pin Options..** button to show the **Device and Pin Options** dialog box.



Finally, select EPCS1 as the configuration device and check “Generate compressed bitstreams”. Now you can generate configuration ROM file for EPCS1 by just recompile your project.