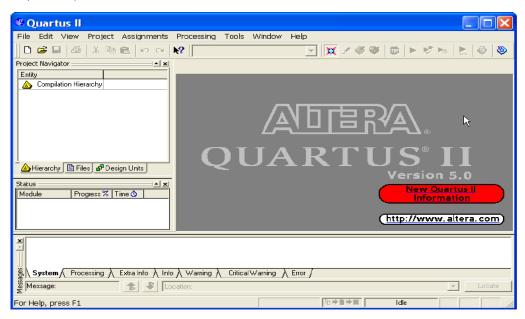
# Chương 1. Hướng dẫn thiết kế và thực hành môn học Hệ thống số trên Kit DE2

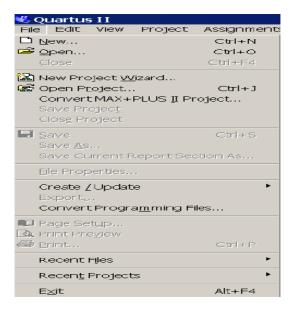
#### 1. Tạo một project trên Quartus II

Buróc 1. Start → Programs → Altera → Quartus II 7.2 → Quartus II 7.2 (32 -Bit)



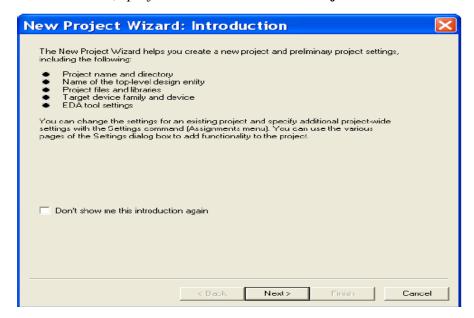
Hình 1.1 Màn hình chính của Quartus

Bước 2. Nhấn tab **File** trên màn hình chính



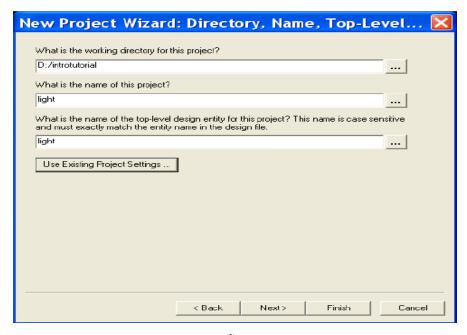
Hình 1.2 Tab File

Bước 3. Mở một project mới: File → New Project Wizard...



Hình 1.3 Tạo project

#### Bước 4. Nhấn **Next**



Hình 1.4 Chỉ đường dẫn và tên project

- Bước 5. Nhập đường dẫn thư mục của project (có thể tạo trước hoặc nếu chưa tạo sẽ được tự động tạo).
- Bước 6. Nhập tên của project.

### Digital\_System:LAB1

Bước 7. Nhập top-level của thiết kế cho project (nên cho giống tên của project).

Bước 8. Nhấn **Next** 

Bước 9. Nếu đường dẫn thư mục của project chưa được tạo trước:



Hình 1.5 Đường dẫn chưa tồn tại

#### Bước 10. Nhấn **Yes**



Hình 1.6 Add các file sử dụng trong project

#### Bước 11. Nhấn **Next**

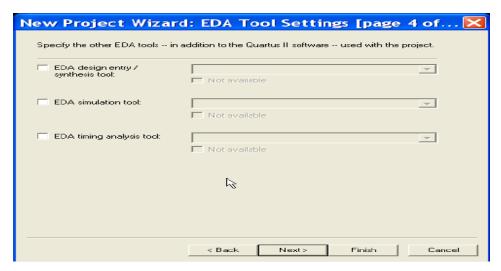


Hình 1.7 Chọn thiết bị FPGA

Bước 12. Chọn **Family : Cyclone II** 

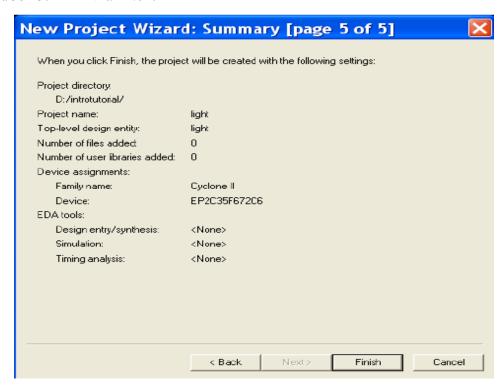
Bước 13. Chọn **Available devices : EP2C35F672C6** (Họ của Chip FPGA Cyclone II trên Kit DE2).

Bước 14. Nhấn **Next** 



Hình 1.8 Thiết lập EDA tool

Bước 15. Nhấn **Next** 



Hình 1.9 Hoàn thành việc tạo project

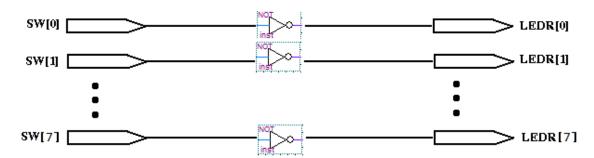
## Digital\_System:LAB1

Bước 16. Nhấn **Finish** để chở về màn hình chính.



Hình 1.10 Màn hình chính sau khi tạo project hoàn thành

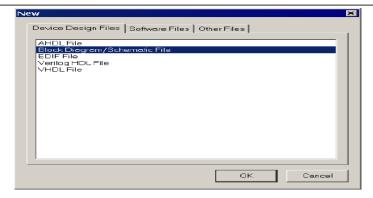
2. Thiết kế một mạch điện đơn giản như hình sau dùng Schematic trên Quartus II:



Hình 1.11 Thiết kế một mạch số đơn giản

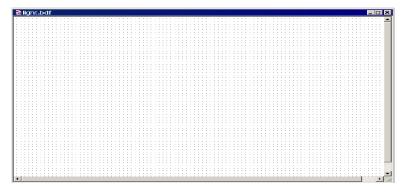
### 1.1.2.1 Mở trình thiết kế sử dụng schematic

Bước 1. Mở File → New



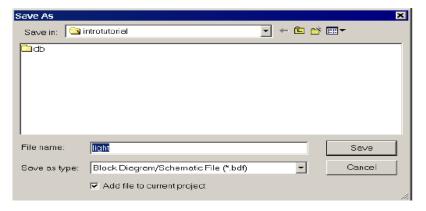
Hình 1.12 Chọn công cụ thiết kế

#### Bước 2. Chọn Block Diagram/Schematic File



Hình 1.13 Cửa sổ thiết kế mạch số

Bước 3. Save as file : **File → Save as** 



Hình 1.14 Lưu thiết kế

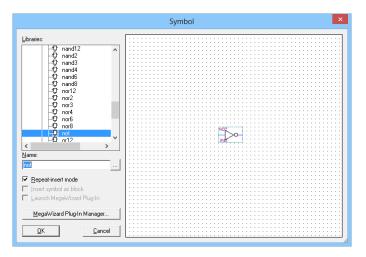
#### 1.1.2.2 Thiết kế mạch hệ thống số

#### Bước 1. Chọn và nhập cổng Logic

Graphic Editor cung cấp một số thư viện chứa những linh kiện điện tử, cho phép người sử dụng chọn và nhập vào schemtic. Nhấp đúp lên khoảng trống bên trong cửa sổ

### Digital\_System:LAB1

Graphic Editor hoặc nhấp lên biểu tượng trong thanh công cụ. Một cửa sổ như hình dưới xuất hiện

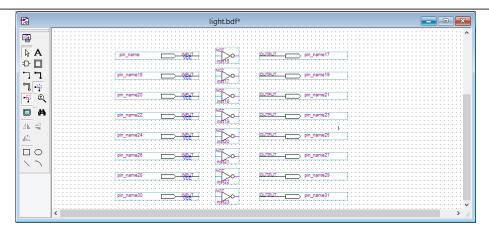


Hình 1.15 Chọn linh kiện

Từ cửa sổ này, ta có thể tìm và chọn những linh kiện hay cổng logic mà ta muốn để nhập vào cửa sổ Graphic Editor bằng cách sau khi chọn linh kiện thì ta nhấp nút **OK**. Thí dụ ta muốn nhập một cổng NOT ngõ vào, ta sẽ tìm và chọn not từ Library, sau đó nhấn **OK**, ta sẽ được một biểu tưởng cổng NOT xuất hiện trên cửa sổ Graphic Editor. Sử dụng chuột để di chuyển linh kiện đến vị trí mong muốn bằng cách nhấn chuột lên linh kiện và kéo rồi nhấp chuột để đặt nó xuống vị trí mới. Nếu muốn nhập một cổng NOT lần thứ hai, ta có thể làm như cách trên hoặc có thể copy từ biểu tượng đã có sẵn trên cửa sổ bằng cách nhấp phải chuột, kéo rê chuột để tạo ra một biểu tượng thứ hai. Ta cũng có thể xoay biểu tượng của linh kiện bằng việc sử dụng biểu tượng trên thanh công cụ.

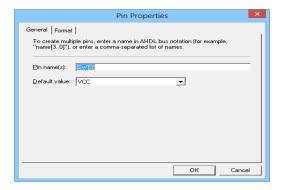
Bước 2. Gán ngõ vào và ngõ ra cho linh kiện:

Sau khi đã nhập linh kiện vào trong cửa sổ Graphic Editor, ta phải gán ngõ vào và ngõ ra cho linh kiện trong mạch điện. Qui trình cũng tương tự như tìm và nhập linh kiện, nhưng biểu tượng ngõ vào hay ngõ ra sẽ được tìm thấy trong thư viện **primitives/pin**. Trong hình dưới, ta sẽ nhìn thấy biểu tượng của ngõ vào và ngõ ra được gán vào chân của linh kiện.



Hình 1.16 Các linh kiên đã được chon

Sau khi gán ngõ vào và ngõ ra cho linh kiện, ta phải đặt tên cho chúng. Để đặt tên, ta nhấp đúp vào từ **pin\_name** của ngõ vào hay ngõ ra. Một hộp thoại như hình sau sẽ xuất hiện



Hình 1.17 Đặt tên pin cho thiết kế

Nhập tên cho chân linh kiện vào ô Pin name(s), rồi nhấn OK.

Bước 3. Kết nối linh kiện

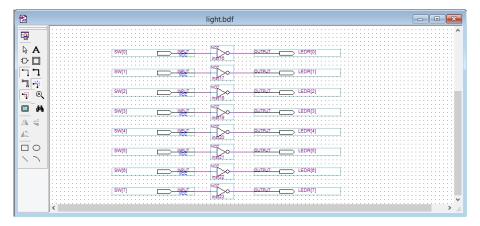
Những linh kiện trong mạch phải được kết nối bằng dây điện. Nhấn chọn biểu

tượng trên thanh công cụ để kích hoạt **Orthogonal Node Tool**. Di chuyển con trở đến đầu của chân linh kiện, nhấn và giữ chuột trái và kéo cho đến khi đường dây chạm vào chân của linh kiện nào mà mình muốn kết nối tới. Chú ý, dấu chấm đen nhỏ thể hiện cho sự kết nối giữa hai đường dây dẫn.

Với qui trình tương tự, ta sẽ kết nối cho toàn bộ mạch điện sao cho đúng với chức năng hoạt động mà ta mong muốn. Nếu trong quá trình kết nối dây, ta kết nối sai dây dẫn nào đó, ta có thể xóa dây dẫn đó đi bằng cách nhấn chọn dây dẫn đó rồi nhấn phím

**Delete (Del)** trên bàn phím. Sau khi hoàn thành kết nối dây, ta nhấn biểu tượng dễ để kích hoạt chức năng Select and Smart Drawing Tool. Bây giờ ta có thể sắp đặt lại vị trí

của mạch điện sao cho dễ nhìn bằng cách chọn linh kiện hoặc dây dẫn và di chuyển chúng đến một vi trí thích hợp hơn. Thí dụ, hình dưới đây là một mạch điện hoàn chỉnh:



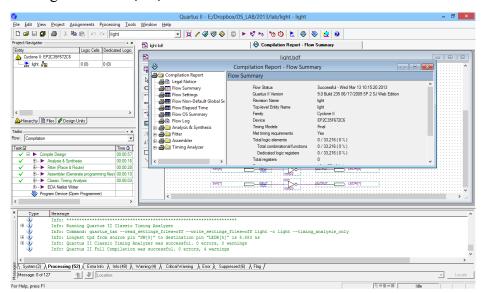
Hình 1.18 Thiết kế hoàn chỉnh

Sau khi mạch điện được hoàn chỉnh, ta nhớ lưu lại. Một file thiết kế sẽ được lưu dưới định dạng .bdf. Thí dụ **light.bdf** 

#### . Trình biên dich

Với dữ liệu vào là file định dạng .bdf (light.bdf), nhiều công cụ trong phần mềm Quartus II được dùng để phân tích, tổng hợp mạch đã được thiết kế ở phần trên, rồi sau đó sẽ tạo ra một file thực thi dùng để nạp lên FPGA. Những công cụ được sử dụng trong quá trình này được gọi là trình biên dịch. Để thực thi quá trình biên dịch, ta thực hiện các bước sau:

Bước 1. Chọn: **Processing** → **Start Compilation** hoặc nhấn chọn biểu tượng trên thanh công cụ. Sau khi quá trình biên dịch được hoàn tất, một bảng báo cáo được tạo ra như hình dưới



Hình 1.19 Cửa sổ trình biên dịch report

Để xem lại quá trình biên dịch, ta chọn : Processing > Bước 2.

Compilation Report hoặc nhấn chọn biểu tượng trên thanh công cu.



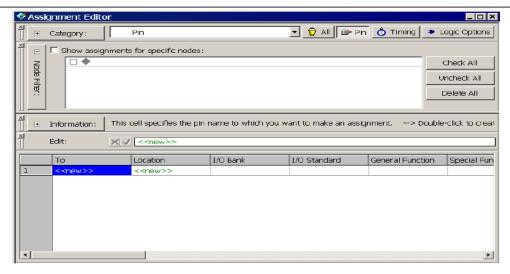
#### 1.1.2.3 Message window

Phần mềm Quartus II sẽ hiển thi thông tin trong suốt quá trình biên dịch trên cửa sổ Message widow. Nếu sơ đồ mạch điện được thiết kế trong phần Graphic Editor hoàn toàn đúng, thì một thông báo "The compilation was successful" được hiện thi. Trong trường hợp quá trình biên dịch xuất hiện lỗi thì có nghĩa đã có lỗi xảy ra trong quá trình thiết kế trên Graphic Editor. Mỗi thông báo tương ứng với một lỗi được tìm thấy sẽ xuất hiện trên cửa sổ **Message**. Nhấp đúp vào thông báo lỗi đó ta sẽ biết rõ hơn về lỗi đã xảy ra trên mạch điện. Tương tư, trình biên dịch cũng thông báo một số cảnh báo "Warning". Ngoài ra ta cũng có thể tìm hiểu thêm thông tin về lỗi cũng như cảnh báo bằng cách nhấn chon vào thông báo đó rối nhấn phím F1 trên bàn phím.

#### 3. Gán pin

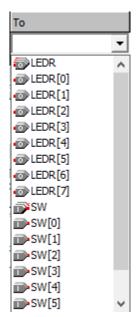
Vì ta chưa thực hiện gán pin trên FPGA cho linh kiến trong mạch điện đã thiết kế ở trên nên khi thực hiện biên dịch thì trình biên dịch Quartus II đã gán chân của linh kiên với pin của FPGA một cách ngẫu nhiên. Tuy nhiên, giả sử trong thiết kế cổng XOR đơn giản ở trên, sau khi thiết kế được biên dịch và nạp lên FPGA, ta muốn hai ngõ vào SW[0] tới SW[7] được điều khiển bởi hai switch SW0 tới SW7 còn kết quả ngõ ra LEDR[0] tới LEDR[7] sẽ được thể hiện trên led LEDR0 tới LEDR7 (các SW và LED được ghi trên Kit). Mặt khác ta biết switch SW0 được kết nổi cổ định với pin N25 của FPGA, tương tư vậy switch SW1 được kết nối cố định với pin N25 của FPGA và tương tự các chân khác trong file pin của Altera. Để thực hiện được điều đó ta phải gán chân linh kiện trên mạch (SW[0] tới SW[7] và LEDR[0] tới LEDR[7]) với pin tương ứng trên FPGA (N25, N26...). Để gán pin ta thực hiện các bước sau

Chọn **Assignments > Pins**, một cửa sổ như hình dưới sẽ xuất hiện Bước 1.



Hình 1.20 Cửa sổ mapped pin giữa thiết kế và FPGA

Bước 2. Trong mục **Category** chọn **Pin**. Nhấp đúp lên mục **<<new>>>** trong cột **To**. Một cửa sổ như hình dưới xuất hiện



Hình 1.21 Cửa số gán pin

Bước 3. Nhấn chọn x1 để gán pin trước, tiếp đến nhấp đúp lên mục ngay bên phải của SW[0] trong cột **Location**, một cửa sổ như hình dưới sẽ xuất hiên

Location	I/O Bank		I/O Standard		General Function	Special	F ^
PIN_N25			3.3-V LVTTL				
PIN_N2	I/O Bank 2	Dedicate	ed Clock	CLK0, LVDS	CLKOp, Input		٨
PIN_N9	I/O Bank 2	Row I/O	)	LVDS31p			
PIN_N18	I/O Bank 5	Row I/C	)	LVDS110p			
PIN_N20	I/O Bank 5	Row I/C	)	LVDS124p			
PIN_N23	I/O Bank 5	Row I/C	)	LVDS126p, [	DPCLK7/DQS0R/CQ1R		
PIN_N24	I/O Bank 5	Row I/C	)	LVDS126n			
PIN_N25	I/O Bank 5	Dedicate	ed Clock	CLK4, LVDS	CLK2p, Input		
PIN_N26	I/O Bank 5	Dedicate	ed Clock	CLK5, LVDS	CLK2n, Input		
PIN_P1	I/O Bank 1	Dedicate	ed Clock	CLK3, LVDS	CLK1n, Input		
PIN_P2	I/O Bank 1	Dedicate	ed Clock	CLK2, LVDS	CLK1p, Input		
PIN_P3	I/O Bank 1	Row I/C	)	LVDS26p, D	PCLK1/DQS1L/CQ1L#		
PIN_P4	I/O Bank 1	Row I/C	)	LVDS26n			
PIN_P6	I/O Bank 1	Row I/O	)	LVDS22n			
PIN_P7	I/O Bank 1	Row I/O	)	LVDS22p			
PIN_P9	I/O Bank 2	Row I/C	)	LVDS31n			
PIN_P17	I/O Bank 6	Row I/C	)	LVDS130n			٧

Hình 1.22 Cửa sổ liệt kê danh sách pin của FPGA

- Bước 4. Ta nhấp chọn PIN\_N25.
- Bước 5. Tương tự, ta gán pin cho chân ngõ vào SW[1] tới pin PIN\_N26, và các chân còn lại tới các pin tương ứng. Sau khi gán pin hoàn tất, ta sẽ được như hình dưới

	То	Location	I/O Bank	I/O Standard	General Function	Special Function	Reserved
4	<b>■</b> SW[3]	PIN_AE14	7	3.3-V LVTTL	Dedicated Clock	CLK12, LVDSCLK6n, I	
5	<b>■</b> SW[4]	PIN_AF14	7	3.3-V LVTTL	Dedicated Clock	CLK13, LVDSCLK6p, I	
6	■ SW[5]	PIN_AD13	8	3.3-V LVTTL	Dedicated Clock	CLK14, LVDSCLK7n, I	
7	■ SW[6]	PIN_AC13	8	3.3-V LVTTL	Dedicated Clock	CLK15, LVDSCLK7p, I	
8	<b>■</b> SW[7]	PIN_C13	3	3.3-V LVTTL	Dedicated Clock	CLK10, LVDSCLK5n, I	
9	DLEDR[0]	PIN_AE23	7	3.3-V LVTTL	Column I/O	LVDS151n	
10	■ LEDR[1]	PIN_AF23	7	3.3-V LVTTL	Column I/O	LVDS151p	
11	■LEDR[2]	PIN_AB21	7	3.3-V LVTTL	Column I/O	LVDS152n	
12	■ LEDR[3]	PIN_AC22	7	3.3-V LVTTL	Column I/O	LVDS152p	
13	■ LEDR[4]	PIN_AD22	7	3.3-V LVTTL	Column I/O	LVDS153n	
14	■ LEDR[5]	PIN_AD23	7	3.3-V LVTTL	Column I/O	LVDS153p	
15	■ LEDR[6]	PIN_AD21	7	3.3-V LVTTL	Column I/O	LVDS154n	
16	■LEDR[7]	PIN_AC21	7	3.3-V LVTTL	Column I/O	LVDS154p, CDPCLK3/	
17	< <new>&gt;</new>	< <new>&gt;</new>					

Hình 1.23 Cửa sổ sau gán pin

- Bước 6. Lưu lại kết quả gán pin: **File → Save**
- Bước 7. Ta phải biên dịch lại thiết kế ở trên với kết quả gán pin này vì như ta đã nói ở trên, vì quá trình biên dịch ở trên, trình biên dịch Quartus II chỉ gán pin một cách ngẫu nhiên nên sẽ không đúng với yêu cầu thiết kế của ta, do đó ta phải gán lại pin cho đúng với yêu cầu rồi phải chạy lại quá trình biên dịch. Lúc này trình biên dịch Quartus II sẽ sử dụng những pin mà ta đã gán cho chân của mạch điện trong thiết kế để phân tích, tổng hợp và tạo ra một file để thực thi việc nạp xuống cho FPGA.

Ngoài ra ta cũng có một cách khác để gán pins cho design, đặc biệt là rất hữu ích trong thiết kế mà có nhiều chân, ta không thể ngồi gán pin cho từng chân được vì sẽ tốn

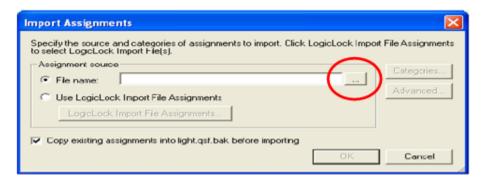
nhiều thời gian, Quartus II cung cấp một phương pháp giúp ta gán nhiều pin vào hoặc gỡ nhiều pin ra cùng một lúc bằng một file có định dạng đặc biệt dùng cho mục đích này đó là định dạng .CSV. Format của file này như sau

Nếu ta dùng Microsoft Excel, thì ta sẽ có format như sau:

7	То	Location
8	SW[0]	PIN_N25
9	SW[1]	PIN_N26
10	SW[2]	PIN_P25
11	SW[3]	PIN_AE14
12	SW[4]	PIN_AF14
13	SW[5]	PIN_AD13
14	SW[6]	PIN_AC13
15	SW[7]	PIN_C13
16	LEDR[0]	PIN_AE23
17	LEDR[1]	PIN_AF23
18	LEDR[2]	PIN_AB21
19	LEDR[3]	PIN_AC22
20	LEDR[4]	PIN_AD22
21	LEDR[5]	PIN_AD23
22	LEDR[6]	PIN_AD21
23	LEDR[7]	PIN_AC21

Hình 1.24 Dùng Microsoft Excel để tạo file gán pin

Sau khi tạo file có format như trên, ta sẽ thực hiện việc gán pin như sau
 Bước 1. Chọn Assignments -> Import Assignments, một hộp thoại như hình dưới xuất hiện



Hình 1.25 Import file gán pin

Bước 2. Click **button ...** , chỉ đường dẫn của file ta vừa tạo ở trên. Rồi nhấn **OK**.

Để thuận tiện cho người sử dụng Altera đã cung cấp một file CSV có tên DE2\_pin\_assignments, file này liệt kê tất cả các pin của FPGA, có format như sau:

Location
PIN_N25
PIN_N26
PIN_P25
PIN_AE14
PIN_AF14
PIN_AD13
PIN_AC13
PIN_C13
PIN_B13
PIN_A13
PIN_N1
PIN_P1

Hình 1.26 File gán pin tao sẵn bởi Altera

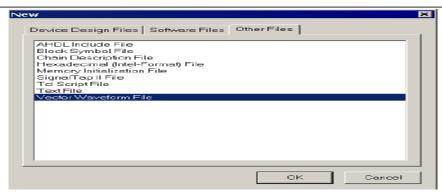
Nếu ta muốn sử dụng file có sẵn này vào việc gán pin cho thiết kế của ta thì một yêu cầu bắt buộc khi ta đặt tên cho chân linh kiện phải trùng với tên trong cột **To** của file này. Thí dụ, nếu ta muốn hai chân ngõ vào của cổng XOR được điều khiển bởi hai Switch 0 và Switch 1 trên Kit DE2 thì ta phải đặt tên cho hai chân này lần lượt là SW[0], SW[1] như trong cột **To** của file này. Do đó ta phải tham khảo file này trước khi đặt tên cho chân linh kiện để khi gán pin ta sẽ rất thuận tiện đó là không phải tạo file.csv nữa mà chỉ cần Import file có sẵn này vào thôi.

Sau khi gán pin xong, ta biên dịch lại.

- **♣** Re- compiling design : **Processing → Start Compilation**
- **♣** Review Compilation report : **Processing → Compilation Report**

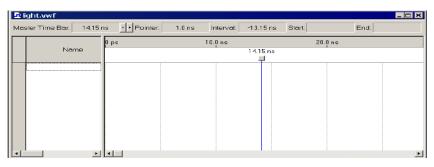
#### 4. Mô phỏng mạch đã thiết kế:

Buốc 1. Tạo input waveform : **File → New → Other Files → Vector**Waveform File



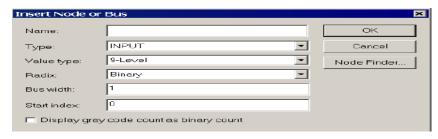
Hình 1.27 Tạo waveform

#### Bước 2. Nhấn **OK**



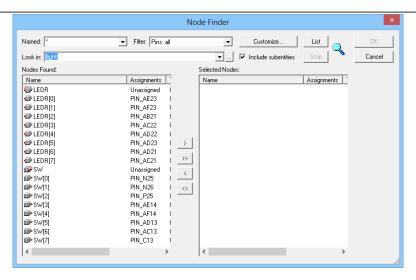
Hình 1.28 Cửa sổ tạo waveform

- Bước 3. Chọn thời gian thực hiện mô phỏng : **Edit** → **End Time**
- Bước 4. Nhập thời gian thực hiện mô phỏng.
- Buốc 5. Fit windown : View → Fit in Windown
- Bước 6. Tạo waveform cho inputs : Edit → Insert Node or Bus



Hình 1.29 Nhập tên signal của thiết kế

#### Bước 7. Chọn Node Finder



Hình 1.30 Dùng chức năng Node Finder

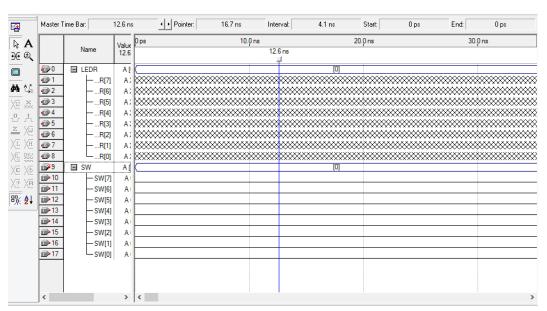
Bước 8. Chon Filter: Pins: all

Bước 9. Nhấn button **List** 

Bước 10. Chọn signal bên **Nodes found** ; nhấn >> để chuyển sang bên

#### **Selected Nodes**

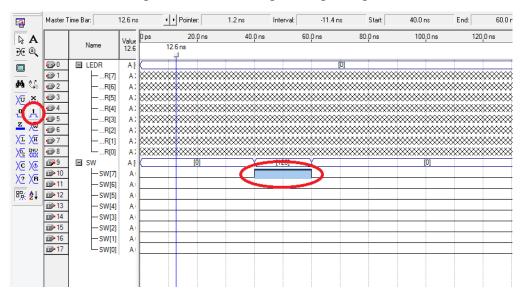
Bước 11. Nhấn **OK** 



Hình 1.31 Tạo input waveform

- Bước 12. Chọn một input signal bằng cách nhấp chuột vào signal đó.
- Bước 13. Chọn biểu tượng mũi tên con trỏ
- Bước 14. Di chuyển con trỏ sang màn hình waveform.

Bước 15. Nhấn và giữ chuột và kéo rê (left) trong một khoảng thời gian (giả sử ta muốn trong khoảng thời gian từ 40ns -> 60 ns , SW0 signal có giá trị "1", thì ta nhấn , giữ và rê chuột trong khoảng thời gian từ 40ns -> 60ns.



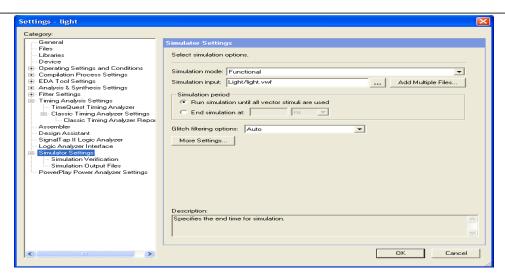
Hình 1.32 Tạo mức logic "1"

Bước 16. Nhấn **button "1"** phía bên trái màn hình

		$\overline{}$	<u> </u>	20.0		0.0		CO O		2.0	100		- 12	
	Name	value	0 ps	20.0 ns	s 4	0.0 ns		60.0 ns	8	).0 ns	100	0 ns	12	),0 ns
	Ivaille	12.6	14	2.6 ns										
<b>⊚</b> 0	■ LEDR	Α[							[0]					
<u></u> 1	⊢R[7]	A)	******	******	****	****	*****	******	*****	*****	******	****	******	****
<u></u> 2	R[6]	A)	<b> </b>	******	****	***	*****	*******	*****	******	*****	****	*****	***
<b></b> 3	—R[5]	A)	<b>******</b>	******	********	XXXX	*****	*******	*****	******	******	****	******	****
<b>₯</b> 4	R[4]	A)	XXXXXX	*****	****	XXXX	*****	******	*****	******	*****	****	*****	<b>****</b>
<u>∞</u> 5	—R[3]	A)	<b>******</b>	*****	*********	₩₩	*****	******	*****	*****	*****	****	*****	⋘
<b>∞</b> 6	—R[2]	A)	<b>******</b>	******	********	****	*****	******	*****	*****	*****	****	*****	***
<u>₀</u> 7	—R[1]	A)	XXXXXX	******	*********	****	*****	*******	*****	******	*****	*****	*****	****
<u>∞</u> 8	∟ <sub>R[0]</sub>	A)	XXXXXX	******	****	****	***	******	*****	******	*****	****	*****	****
- 10	- c:	Αſ		[0]		$\propto$	[128]	_x				[0]		
9	■ SW	_ ^ L												
	- SW[7]	-				┚								
<b>→</b> 10		-												
<b>→</b> 10 <b>→</b> 11	-SW[7]	A		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1										
<ul><li>10</li><li>11</li><li>12</li></ul>	— SW[7] — SW[6]	A A												
10 11 12 13 14	— SW[7] — SW[6] — SW[5]	A A		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1										
10 11 12 13 14 15	— SW[7] — SW[6] — SW[5] — SW[4]	A A A		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1										
10 11 12 13 14 15 16	— SW[7] — SW[6] — SW[5] — SW[4] — SW[3]	A A A A												
<ul> <li>→ 10</li> <li>→ 11</li> <li>→ 12</li> <li>→ 13</li> <li>→ 14</li> <li>→ 15</li> <li>→ 16</li> <li>→ 17</li> </ul>	— SW[7] — SW[6] — SW[5] — SW[4] — SW[3] — SW[2]	A A A A												
10 11 12 13 14 15 16	— SW[7] — SW[6] — SW[5] — SW[4] — SW[3] — SW[2] — SW[1]	A A A A												

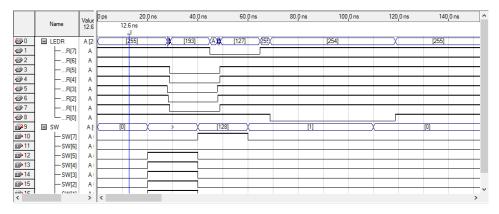
Hình 1.33 Mức logic "1" đã được tạo

- Bước 17. Tương tự cho những tín hiệu inputs khác, không tạo waveform cho outputs (XXX).
- Buớc 18. Save File Waveform : File → Save As
- Bước 19. Thiết lập thực hiện mô phỏng : **Assignments** → **Setting**



Hình 1.34 Thiết lập chế chộ simulation

- Bước 20. Chọn Simulator Settings
- Buốc 21. Chọn Simulation mode: Functional / Timing
- Bước 22. Chỉ đường dẫn của input waveform vừa tạo.
- Bước 23. Nhấn **OK**
- Buốc 24. Tạo simulation netlist : **Processing → Generate Functional Simulation Netlist**
- Bước 25. Chạy mô phỏng : **Processing** → **Start Simulation.**
- Bước 26. Quan sát waveform của Output và debug nếu có lỗi.



Hình 1.35 Waveform sau khi chạy mô phỏng

### 5. Programming mạch đã thiết kế lên FPGA :

- Bước 1. Kết nối Kit DE2 với máy tính qua cổng USB-Blaster (phải cài đặt driver trước ).
- Bước 2. Bật nguồn Kit DE2.
- Có 2 mode cho việc programming : JTAG và Active Serial modes

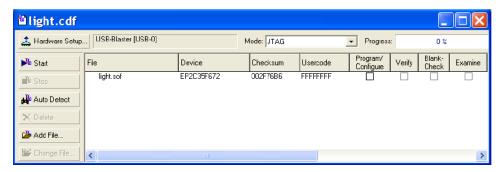
### Digital\_System:LAB1

**.** 

**♣** JTAG mode

Bước 3. Trên Kit DE2, chuyển Switch **RUN/PROG** về vị trí **RUN** 

Bước 4. Trên màn hình chính Quantus II, chọn **Tools** → **Programmer** 



Hình 1.36 Nạp thiế kế lên FPGA

Bước 5. Nhấn **Hardware Setup**, chọn **USB-Blaster[USB-0]** (Chú ý : phải cài đặt driver cho USB-Blater trước).



Hình 1.37 Thiết lập cổng giao tiếp giữa kit DE2 và Computer

Bước 6. Nhấn **Close** 

Bước 7. Chọn **Mode JTAG** 

Bước 8. Nhấn **Add File**, chỉ đường dẫn đến **File**.**sof** (được tạo ra khi chạy Compilation).

Bước 9. Check box **Program/Configure** 



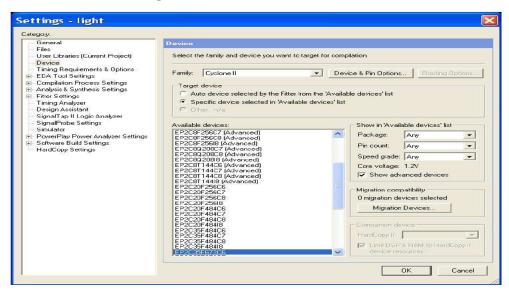
Hình 1.38 Chọn cấu hình nạp thiết kế

Bước 10. Nhấn **Start**.

Bước 11. Quan sát trên Kit DE2, bật và tắt các switch SW0 tới SW1 và quan sát LEDR tương ứng.

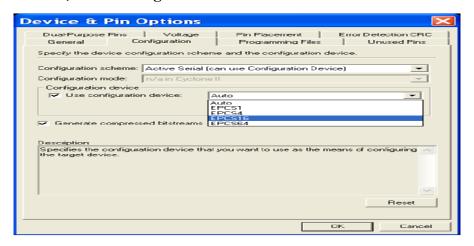
♣ Active Serial Mode :

Bước 12. Chọn **Assignments** → **Devide** 



Hình 1.39 Thiết kế chế độ nạp lên FPGA bằng AS mode

- Bước 13. Chọn Family: Cyclone II
- Buốc 14. Chon Available devices: EP2C35F672C6
- Bước 15. Nhấn **Device & Pin Option**
- Bước 16. Chọn **Tab Configuration**



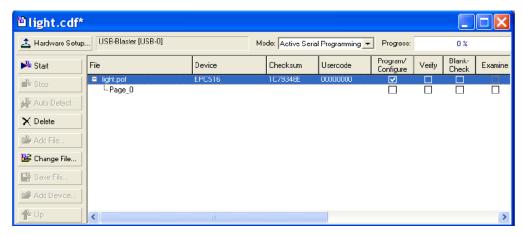
Hình 1.40 Chọn loại ROM tương ứng

- Bước 17. Chọn **Configuration device** : **EPCS64** (họ EPPROM trên Kit DE2 , dùng để lưu chương trình để nạp cho FPGA mỗi khi power on).
- Bước 18. Tương tự JTAG ở những bước kế tiếp:
- Bước 19. Trên Kit DE2, chuyển Switch **RUN/PROG** về vị trí **RUN**
- Bước 20. Trên màn hình chính Quantus II, chọn **Tools → Programmer**
- Bước 21. Chọn **Hardware Setup : USB-Blaster[USB-0]**

### Digital\_System:LAB1

Bước 22. Chọn **Mode : Active Serial Programming** 

Bước 23. Nhấn **Add File**, chỉ đường dẫn đến **File .pof** (File được tạo ra trong quá trình chạy Compilation).



Hình 1.41 Chọn file thiết kế .pof

- Buốc 24. Check box **Program/Configure.**
- Bước 25. Nhấn **Start** để programming chương trình cho EPPROM.
- Bước 26. Nhấn Phím **Restart** trên Kit DE2,
- Bước 27. Quan sát trên Kit DE2, switch SW0 tới SW7 và quan sát LED.