



디지털논리회로실습

Lab 0. Practice

2017. 09. 13

HANYANG University

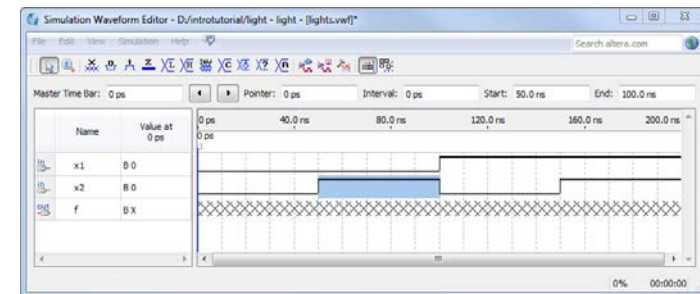
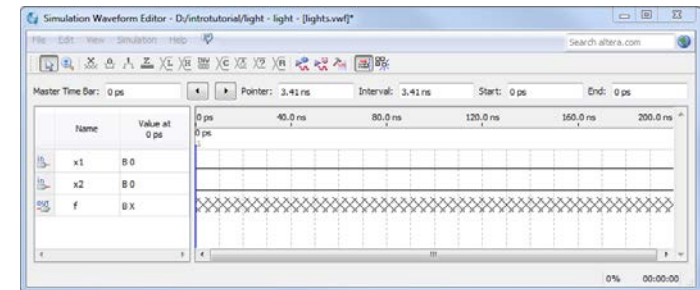
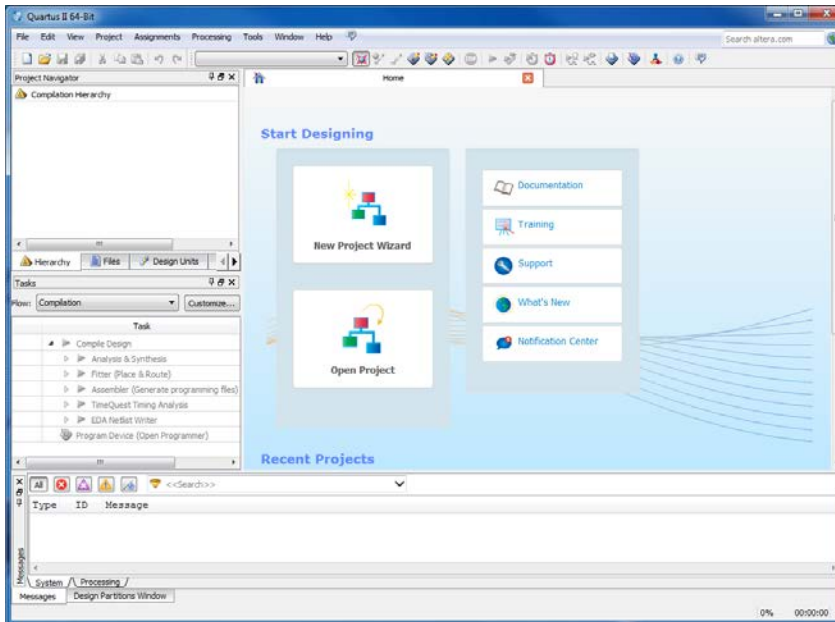
Contents

- 1. DE2 보드 개발환경 만들기.
- 2. Verilog HDL 코드를 DE2 보드에 올리기.
- 3. 기본 문법 실습.

DE2 보드 개발환경 만들기

Review

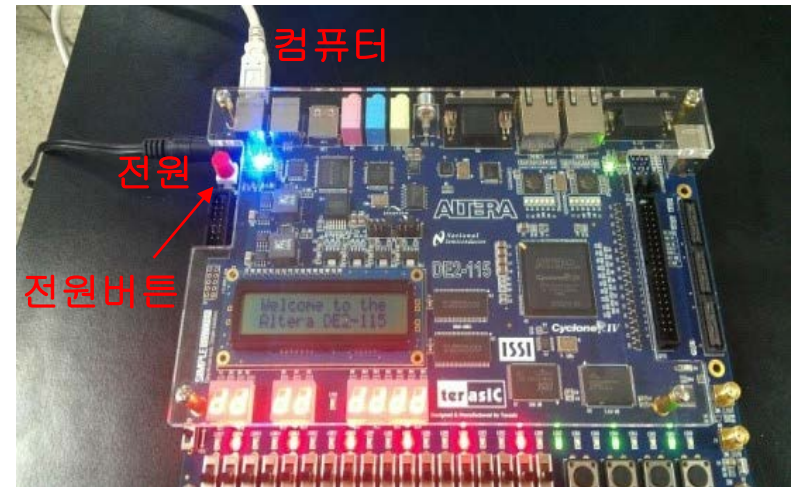
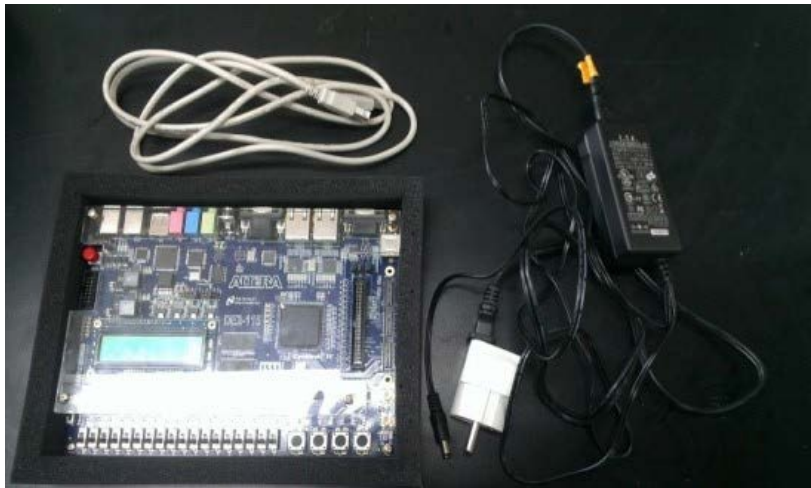
Quartus2 software 설치 및 설정



DE2 보드 개발환경 만들기

DE2 보드 구성

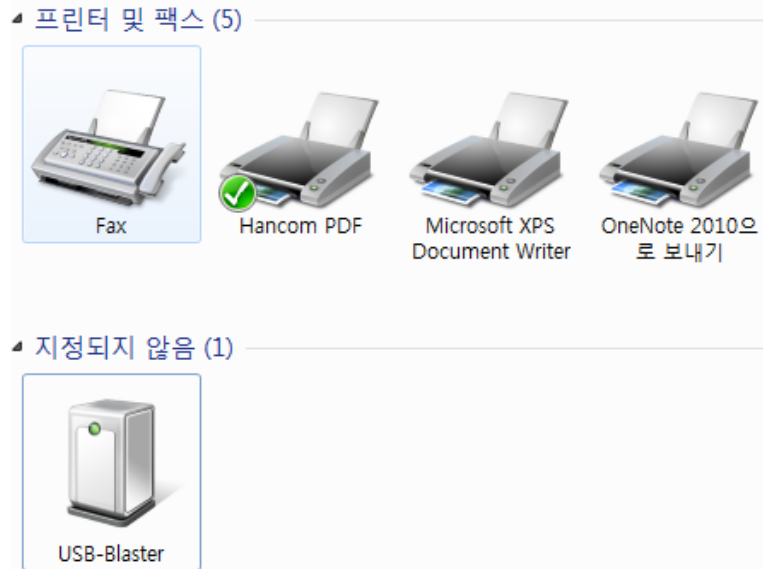
- DE2 보드, 전원, USB케이블



DE2 보드 개발환경 만들기

장치 설치

■ 제어판-장치 및 프린터



설치 순서

1. 아이콘 더블클릭
2. 팝업메뉴에서 하드웨어 클릭
3. 장치 기능 중 느낌표가 있는 부분 더블클릭
4. 왼쪽 하단에 설정변경 클릭
5. 상단 바의 드라이버 클릭
6. 드라이버 업데이트 클릭
7. 드라이버 소프트웨어 찾아보기 클릭
8. **quartus**가 설치된 경로 선택
예) **alteraxx.x**

다음 위치에서 드라이버 소프트웨어 검색:

C:\Waltera\W13.1

찾아보기(R)...

☒ 하위 폴더 포함(I)

Verilog HDL 코드를 DE2 보드에 올리기

Verilog HDL 코드를 보드에 올리는 과정

1. 새로운 프로젝트 생성

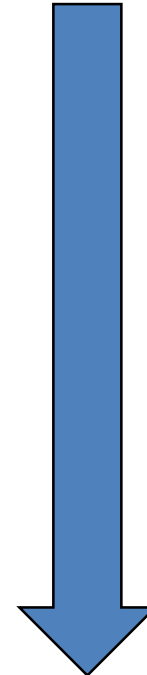
2. Verilog HDL 코드 작성

3. Verilog HDL 코드 컴파일

4. Pin Planner 설정

5. Verilog HDL 코드 컴파일

6. Programmer 를 통해 업로드



Verilog HDL 코드를 DE2 보드에 올리기

새 프로젝트 생성 및 코드 작성

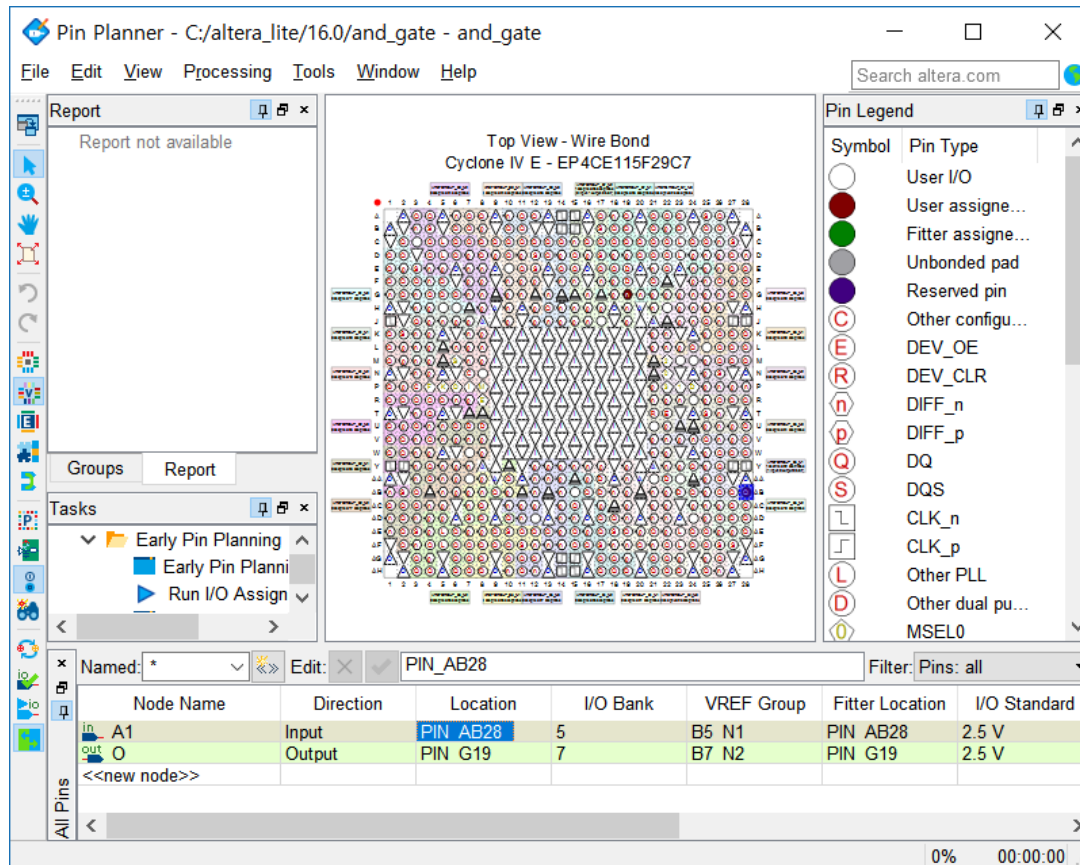
- 저번 시간에 실습했던 NOT_GATE 를 작성하고 컴파일
 (1~3번 단계)

```
1  module not _gate(A1, O);  
2      input A1;  
3      output O;  
4  
5      assign O = ~A1;  
6  
7  endmodule  
8
```

Verilog HDL 코드를 DE2 보드에 올리기

Pin Planner

- 실제 보드의 입력과 출력에 해당하는 Pin을 설정하는 과정
- Assignments 메뉴에 Pin Planner 를 클릭하여 실행



Pin Planner - C:/altera_lite/16.0/and_gate - and_gate

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Groups Report

Tasks

- Early Pin Planning
- Early Pin Planni
- Run I/O Assign

Top View - Wire Bond
Cyclone IV E - EP4CE115F29C7

Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigne...
●	Fitter assigne...
○	Unbonded pad
●	Reserved pin
○	Other configu...
○	DEV_OE
○	DEV_CLR
○	DIFF_n
○	DIFF_p
○	DQ
○	DQS
○	CLK_n
○	CLK_p
○	Other PLL
○	Other dual pu...
○	MSEL0

Named: * Edit: PIN_AB28 Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
in A1	Input	PIN AB28	5	B5 N1	PIN AB28	2.5 V
out O	Output	PIN G19	7	B7 N2	PIN G19	2.5 V
<<new node>>						

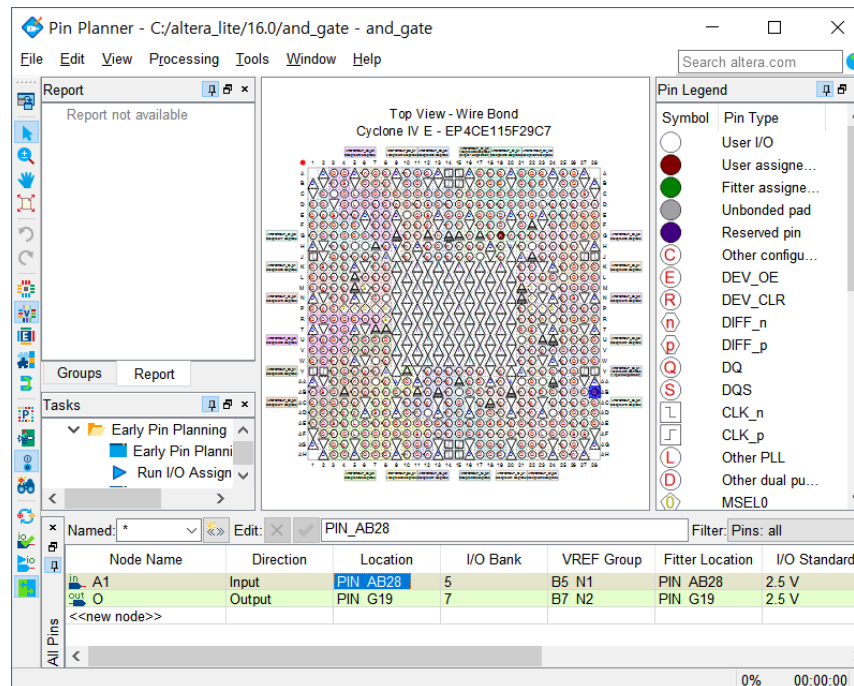
All Pins

0% 00:00:00

Verilog HDL 코드를 DE2 보드에 올리기

Pin Planner

- DE2 보드의 Pin Number 매뉴얼을 참조하여 Input과 Output에 대한 Pin을 설정한다
- Location란에 Pin 번호 입력
 - Example: Switch 0번을 Input으로 LEDR 0번을 Output으로 설정
- Pin 설정이 되었으면 Pin Planner를 닫고 compile을 한번 더 실행

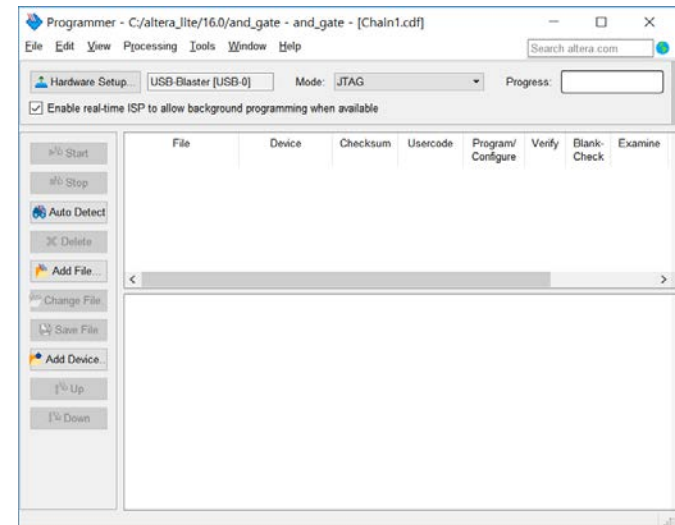
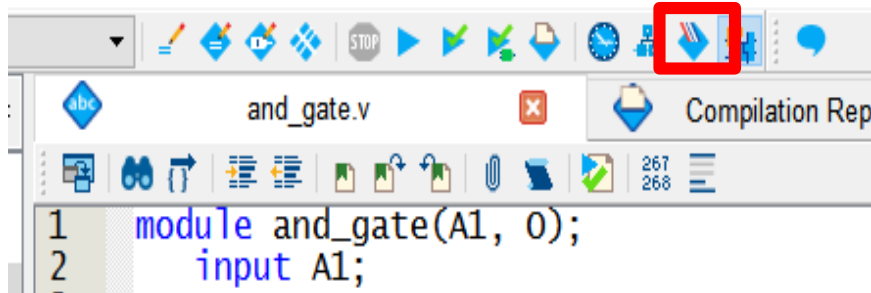


Verilog HDL 코드를 DE2 보드에 올리기

Programmer

- Verilog HDL 코드를 DE2 보드에 올리는 과정
 - ⚠ 주의: PIN 설정이 잘못되어있으면 오작동을 일으킴
- Tools-Programmer 로 실행하거나 아래 보이는 빨간색 박스 안의 버튼을 눌러 실행

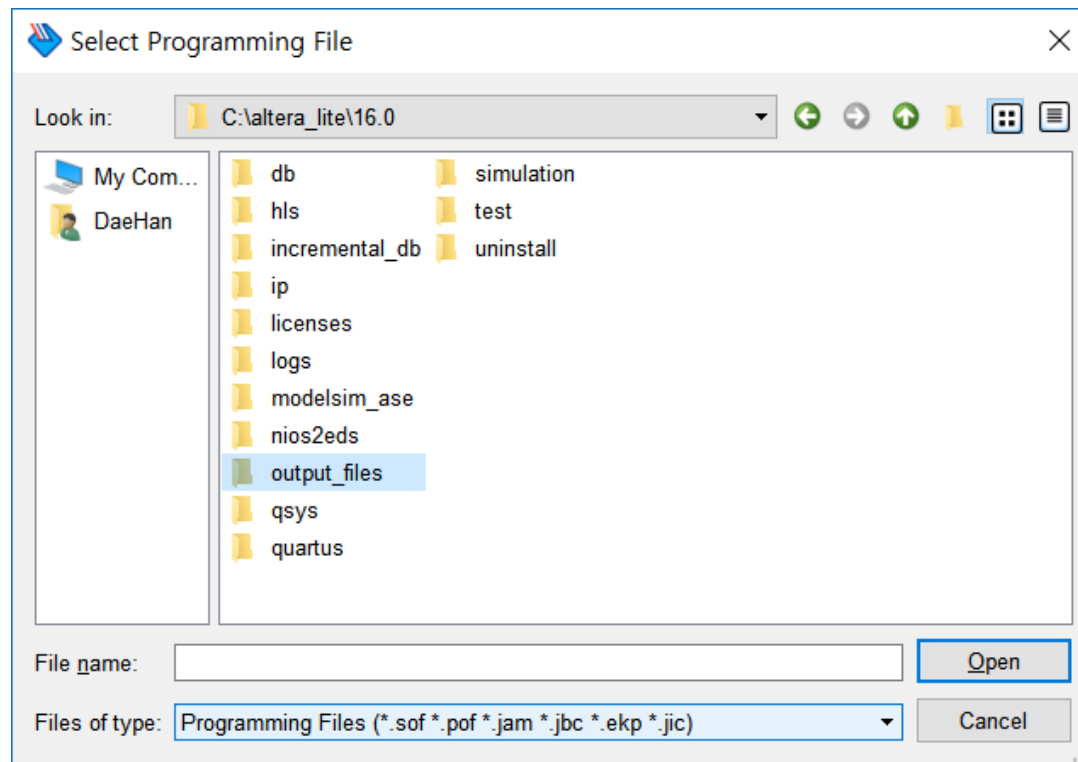
Tools Window Help



Verilog HDL 코드를 DE2 보드에 올리기





Programmer 실행 순서

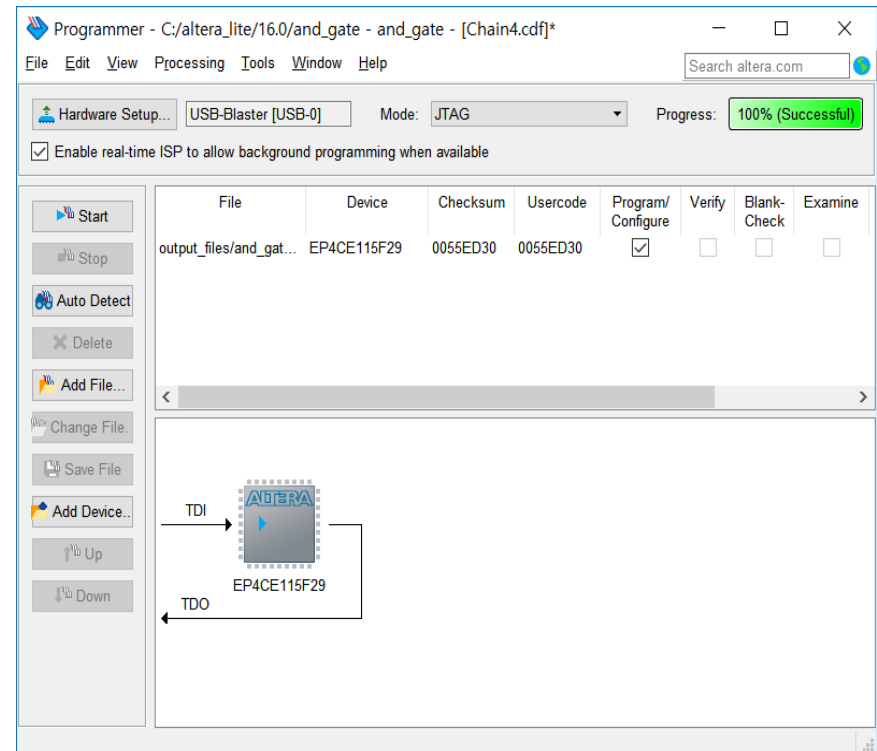
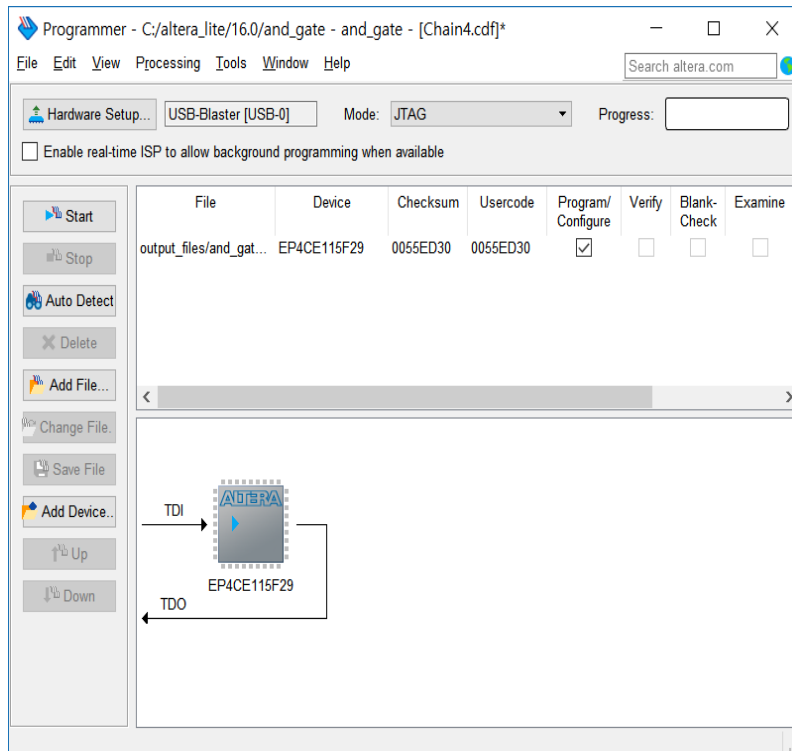
- 1. Add File... 클릭
- 2. Pop up 메뉴에서 'output_files' 폴더 선택
- 3. 현재 만든 프로젝트를 선택한 뒤 'Open' 버튼 클릭



Verilog HDL 코드를 DE2 보드에 올리기

Programmer 실행 순서

-  4. 3번 작업 후 start 버튼이 활성화됨
-  5. Device name을 확인
-  6. Program/Configure 에 체크를 확인
-  7. Start 버튼을 눌러 실행



기본 문법 실습

N bit input & output

- 다음과 같이 선언하면 N 비트를 할당함

















```
module muxx(SW, LED);  
  
    input [15:0]SW;  
    output [15:0]LED;  
  
endmodule
```

Highlighted Pins

Named: *

⚙

Edit:

	Node Name	Direction	Loc
	SW[15]	Input	
	SW[14]	Input	
	SW[13]	Input	
	SW[12]	Input	
	SW[11]	Input	
	SW[10]	Input	
	SW[9]	Input	
	SW[8]	Input	
	SW[7]	Input	
	SW[6]	Input	
	SW[5]	Input	
	SW[4]	Input	
	SW[3]	Input	
	SW[2]	Input	
	SW[1]	Input	
	SW[0]	Input	
	<<new node>>		

기본 문법 실습

■ N bit input & output

- 다음과 같이 Input값을 Output 에 N:1로 할당 하거나 직접 값을 넣을 수 있음

```
input  [4:0]SW1,SW2;  
output [15:0]LED;
```

```
assign LED[4:0]=SW1; //gate  
assign LED[9:5]=~SW2; //not gate
```

```
assign LED[15:10]=6'b100101; //assign
```

6
비트수 +진법

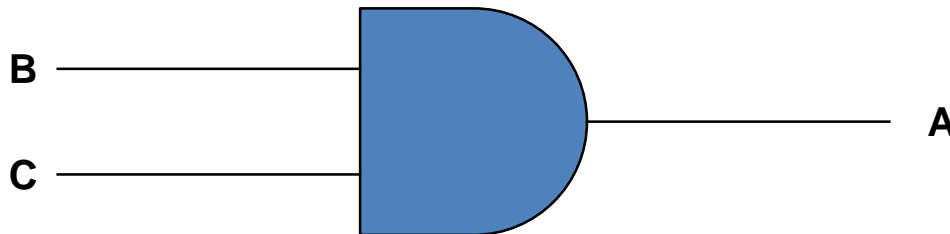
기본 문법 실습

■ wire

- 표현하고 싶은 node들을 단지 물리적인 연결
- 기본문법

🔑 wire <변수명>;


```
1 wire      A, B, C, D, E; // simple 1-bit wide wires
2 wire [8:0] Wide;        // a 9-bit wide wire
3 reg I;
4
5 assign A = B & C;        // using a wire with an assign statement
```



기본 문법 실습

reg

- wire하고는 달리 그 신호에 새로운 값이 할당되기 전 까지 현재의 값을 그대로 유지
- 기본문법

 reg <변수명>;

```
module muxx(SW1, SW2, SW3, LED);  
    input SW1,SW2,SW3;  
    output LED;  
  
    wire a,b,c;  
    reg d;  
  
    assign a=SW1;  
    assign b=SW2;  
    assign c=SW3;  
  
    always @(posedge c) begin  
        d = a & b;  
    end  
  
    assign LED=d;  
  
endmodule
```


기본 문법 실습


always

- 단독으로 쓰이면 '항상' 실행, always뒤에 @이 붙으면 @뒤에 조건이 '참'일 때만 실행
- 기본문법

```
 always @ (<이벤트 리스트>) begin
```

```
...
```

```
end
```

```
 always @ (A or B) begin  
  if(A>B)  
    Y = A;  
  else  
    Y = B;  
  end  
|
```

기본 문법 실습

Clk

- 50Mhz의 신호를 발생



- 기본문법

```
module muxx(SW1, Clk, LED);  
  
    input Clk;  
    input [1:0]SW1;  
    output [3:0]LED;  
    reg [3:0]a,b;  
  
    always @(posedge Clk) begin  
        if(SW1==2'b00) begin  
            a=4'b0000;  
            b=4'b1111;  
        end  
        else if(SW1==2'b01)  
            a=4'b0001;  
        else if(SW1==2'b10)  
            a=4'b0010;  
        else  
            a=4'b0111;  
    end  
  
    assign LED=a;  
  
endmodule
```

기본 문법 실습

IF 문 1

- 조건을 설정하여 조건에 따라 다른 결과를 출력
- 기본 문법 형태
 - ➡ (조건1)? 결과1: (조건2)? 결과2: (조건3)..... : 결과;

```
input [1:0]SW1;  
output [3:0]LED;  
  
assign LED[3:0] = (SW1==2'b00)? 4'b0000:  
                  (SW1==2'b01)? 4'b0001:  
                  (SW1==2'b10)? 4'b0011:  
                  4'b0111; //SW1==2'b11
```

기본 문법 실습

IF 문 2

- 조건을 설정하여 조건에 따라 다른 결과를 출력
- 기본 문법 형태

```
always @(posedge clk) begin
```

```
    if(SW1==2'b00) begin
```

```
        a=4'b0000;
```

```
        b=4'b1111;
```

```
    end
```

```
    else if(SW1==2'b01)
```

```
        a=4'b0001;
```

```
    else if(SW1==2'b10)
```

```
        a=4'b0010;
```

```
    else
```

```
        a=4'b0111;
```

```
end
```

```
assign LED=a;
```

2줄 이상일 경우 **begin ~ end** 를 사용

연습문제

- SW0~3 LEDR0을 이용해 다음과 같이 동작하는 논리 회로를 작성
 - SW2가 Positive 일 때 SW0,1이 AND Gate로 동작 Negative 일 때 OR Gate로 동작해서 LEDR0으로 출력
 - SW3이 positive일 때만 논리회로가 동작함