

Lab #06: 7-Segment Controller

+ Introduction to Sequential Logic Design

Department of Computer Science
Yonsei University



Some of these slides contain material developed and copyrighted by:

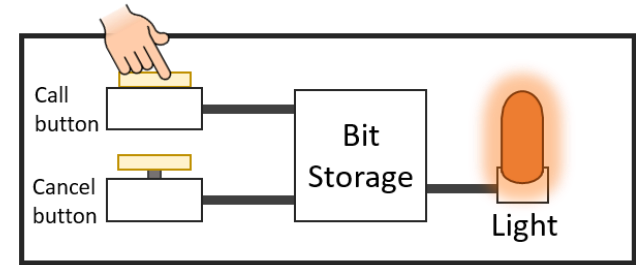
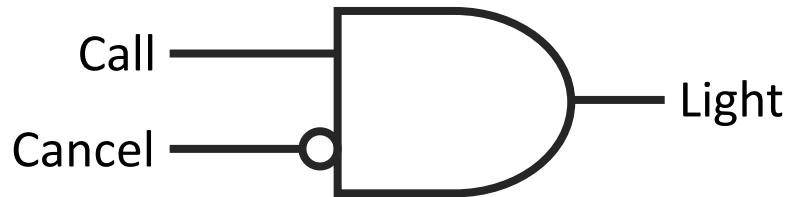
Prof. Shin-dug Kim (Yonsei), Prof. Anantha Chandrakasan (MIT), Prof. Steve Ward (MIT), Prof. John Wawrzynek (UC Berkeley), Junggeun Kim (Yonsei, CS Dept.),
Prof. Michael B. Taylor (UCSD), "Digital Design and Computer Architecture" by Prof. David Money Harris & Prof. Sarah L. Harris (Harvey Mudd College).

순차 논리회로 – Sequential Logic

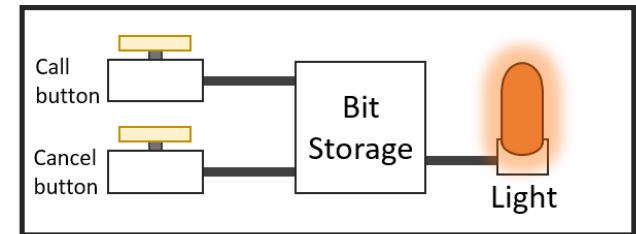
- Sequential circuit
 - Output 값이 현재의 input 값에 의해서만 결정되지 않음
(Combinational circuit의 경우, 현재의 input 값에 의해서만 output 값이 결정됨)
 - 과거 input 값들의 sequence에 의해서도 output이 영향을 받을 수 있음
 - 간단한 예: a circuit that counts up in binary
- 우리가 Sequential Logic을 배우면서 다룰 것들:
 - Flip-flop 설계/구현하기
 - Flip-flop들을 통해 Register(multi-bit storage) 구현하기
 - FSM(Finite State Machine)을 이용해 순차적인 동작을 하는 회로 구성하기
 - FSM을 한 시스템에 대한 controller로써 활용하기
 - ✓ A sequential circuit having a register and combinational logic

[예시] Bit Storage의 필요성

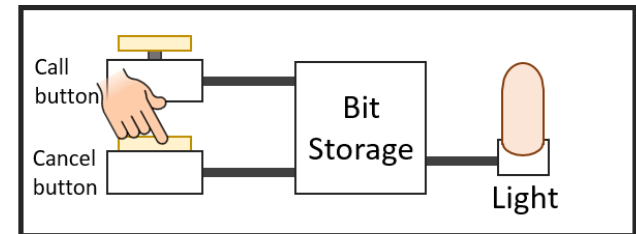
- 비행기 승무원 호출 버튼
 - Call 버튼을 누르면: 전등이 켜짐
 - ✓ 버튼을 누른 뒤 손을 떼더라도 전등은 계속 켜져 있음
 - Cancel 버튼을 누르면: 전등이 꺼짐
- Combinational Logic으로 이 기능을 구현할 수 있을까?



1. Call button pressed – light turns on

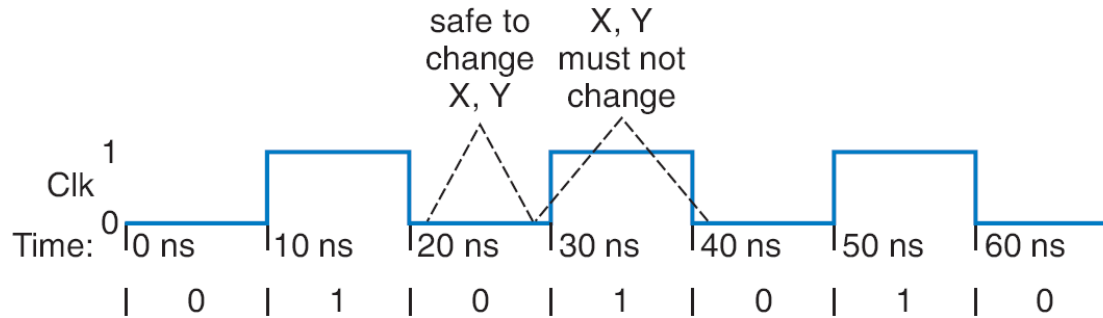


2. Call button released – light stays on



3. Cancel button pressed – light turns off

Clocks



- Clock period: 펄스 간의 시간 간격
 - 위의 예시의 경우: period = 20 ns
- Clock cycle: 특정한 시간의 흐름(간격)을 clock을 기준으로 나타냈을 경우
 - 위의 신호의 경우 3.5 clock cycle 만큼의 시간 경과를 보여준다
- Clock frequency: $1/\text{period}$
 - 위의 신호의 경우: frequency = $1 / 20(\text{ns}) = 50 \text{ (MHz)}$
 - ✓ $1\text{Hz} = 1/\text{s}$
 - ✓ $1\text{GHz} = 1/\text{ns}$

오늘의 실습

- Altera Quartus II 사용하기
 - Altera Quartus II 사용법
 - FPGA Programming
- Clock 활용하기
 - Clock을 배우기에 앞서서 – Quartz 시계와 수정 진동자의 원리/개념
 - Clock의 필요성 – 논리 회로의 지휘자
 - Clock 사용의 사례
 - ✓ Full Adder를 통해서
 - ✓ Verilog HDL로 구성한 Clock 예제
 - Always문의 심화적 활용법
 - 주파수 분주기의 간단한 개념
- FPGA Programming
 - Pin 설정 하기 – input과 output을 위한 설정
 - 7-Segment (FND) 장치와 Keypad의 동작 원리 짚고 넘어가기

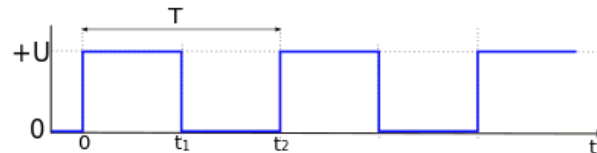
Quartus II 설치

- Quartus II는 Assignment #0에서 Modelsim을 다운로드 받은 곳에서 설치 파일을 구할 수 있습니다.
 - <http://dl.altera.com/13.0/?edition=web>
 - Quartus II Software
 - Cyclone II, Cyclone III, Cyclone IV device support도 다운로드
- 설치 파일 경로나 설치 대상의 경로에 한글이 있으면 안됩니다.
- Select Component 단계에서는 자신의 OS의 Architecture에 맞게 옵션을 설정해주세요. (“Quartus II Software 64-bit support”는 오직 64비트 체계의 시스템/OS에서만 구동 가능)
- 설치 후 최초 실행 시, “Run the Quartus II software”를 선택하세요.
- Quartus II와 ModelSim 연동
 - Tools->Options...의 EDA Tools Options에서 Modelsim-Altera 경로를 [Altera root 폴더]\[버전명]\modelsim_ase\win32aloem 으로 설정
- USB-blaster 문제 해결 방법:
 - <https://www.altera.com/support/support-resources/download/drivers/usb-blaster/dri-usb-blaster-vista.html>
- 자세한 사용법 및 설치 과정은 업로드 한 매뉴얼 참조

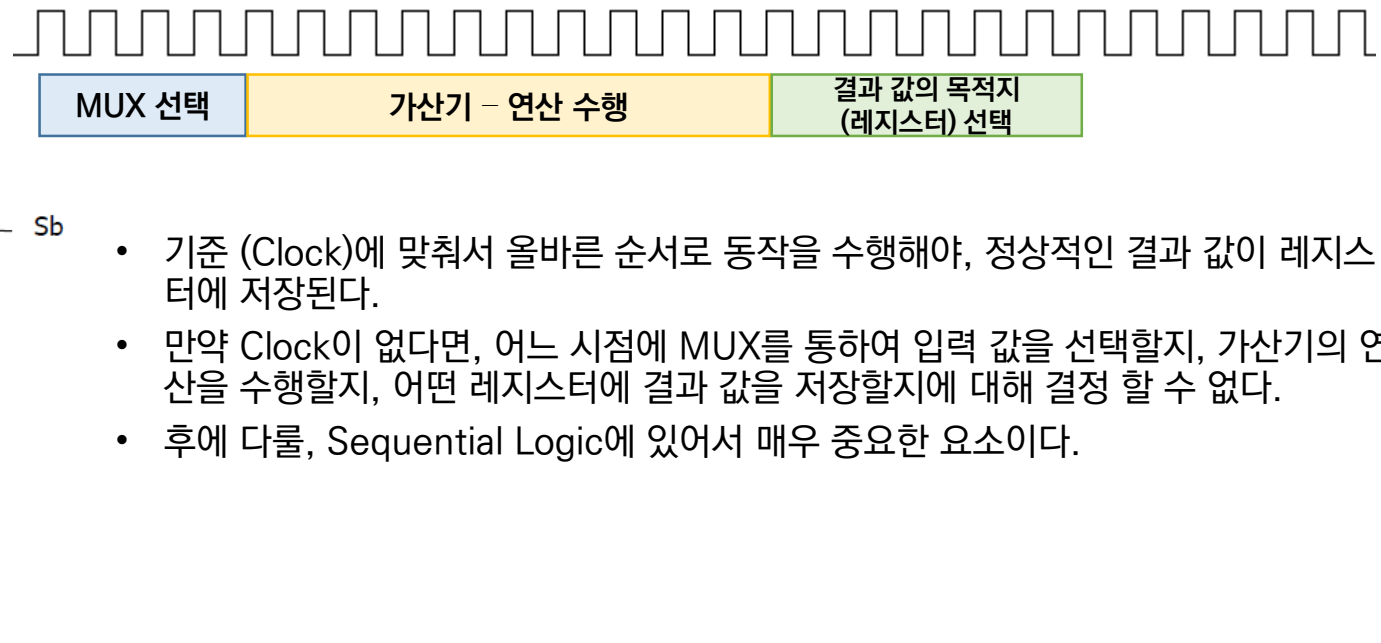
Clock의 필요성

• Clock의 필요성?

- 전자식 시계를 만드는 것에 clock이 필요하다는 것은 살펴보았다.
- 그렇다면, 우리가 배우는 논리 회로와는 무슨 상관이 있는 건가?
- Clock의 필요성: **동기적인 처리를 위한 기준**을 마련하기 위해 clock이 필요하다.



• 예) 가산기



- 기준 (Clock)에 맞춰서 올바른 순서로 동작을 수행해야, 정상적인 결과 값이 레지스터에 저장된다.
- 만약 Clock이 없다면, 어느 시점에 MUX를 통하여 입력 값을 선택할지, 가산기의 연산을 수행할지, 어떤 레지스터에 결과 값을 저장할지에 대해 결정 할 수 없다.
- 후에 다룰, Sequential Logic에 있어서 매우 중요한 요소이다.

Clock의 사례 – FPGA Clock

- 진짜 시계를 만들어봅시다!
 - Verilog HDL을 통해 시계를 모델링 한 뒤, FPGA에 program하여 동작을 확인해봅시다!
 - 7-Segment 부분 코드는 공개하지 않습니다! (여러분들이 오늘 직접 설계하셔야 합니다.)

선언부

```

1 module clock(clk, fnd_data, fnd_ctrl);
2     input                clk;
3     output [7:0]         fnd_data;
4     output [5:0]         fnd_ctrl;
5     reg [7:0]            fnd_data;
6     reg [5:0]            fnd_ctrl;
7
8
9     reg [31:0]           cycle_cnt;
10
11     reg [6:0]            min;
12     reg [6:0]            sec;
13     reg [7:0]            tm_sec;
14
15     reg [3:0]            min_10, min_1, sec_10, sec_1, tm_sec_10, tm_sec_1;
16
17     reg                  fnd_clk;
18     reg [2:0]            fnd_state;
19     reg [3:0]            fnd_num;
20     reg [12:0]           fnd_cnt;
  
```

주파수 분주기

```

22 always @ (posedge clk)
23 begin
24
25     if (cycle_cnt == 0)          fnd_clk = 1;
26     else if (cycle_cnt == 32)    fnd_clk = 0;
27
28     cycle_cnt = cycle_cnt + 1;
29     if (cycle_cnt == 64)          cycle_cnt = 0;
30 end
  
```


Clock의 사례 – FPGA Clock

- 진짜 시계를 만들어봅시다!

시계 동작 부분

```

32  always @ (negedge fnd_clk)
33  begin
34
35      if (fnd_cnt == 5120)    begin          tm_sec = tm_sec + 1;    fnd_cnt = 0; tm_sec_10 = tm_sec / 10; tm_sec_1 = tm_sec % 10;          end
36      if (tm_sec == 100)      begin          tm_sec = 0; tm_sec_10 = 0; tm_sec_1 = 0; sec = sec + 1; sec_10 = sec / 10; sec_1 = sec % 10;          end
37      if (sec == 60)          begin          sec = 0;    sec_10 = 0; sec_1 = 0; min = min + 1; min_10 = min / 10; min_1 = min % 10;          end
38      if (min == 60)          begin          min = 0;    min_10 = 0; min_1 = 0;
39
40      if (fnd_state == 0)      begin fnd_ctrl = 6'b011111; fnd_num = min_10;          fnd_state = 1; end
41      else if (fnd_state == 1) begin fnd_ctrl = 6'b101111; fnd_num = min_1;          fnd_state = 2; end
42      else if (fnd_state == 2) begin fnd_ctrl = 6'b110111; fnd_num = sec_10;          fnd_state = 3; end
43      else if (fnd_state == 3) begin fnd_ctrl = 6'b111011; fnd_num = sec_1;          fnd_state = 4; end
44      else if (fnd_state == 4) begin fnd_ctrl = 6'b111101; fnd_num = tm_sec_10;      fnd_state = 5; end
45      else if (fnd_state == 5) begin fnd_ctrl = 6'b111110; fnd_num = tm_sec_1;      fnd_state = 0; end
46
47      case(fnd_num)
48
49
50
51
52
53      7-Segment로
54      시간을 표시하는 기능
55
56
57
58
59
60
61
62      fnd_cnt = fnd_cnt + 1;
63  end

```

FPGA – Pin 설정하기

- Input / Output에 대한 Pin을 할당해주어야 합니다



[Key Matrix]	
KEY_COL0	L10
KEY_COL1	M14
KEY_COL2	M12
KEY_COL3	M11
KEY_ROW0	N15
KEY_ROW1	N16
KEY_ROW2	M15
KEY_ROW3	M16

[LED]	
LED0	D3
LED1	D4
LED2	C4
LED3	D5
LED4	C5
LED5	D6
LED6	C6
LED7	D7

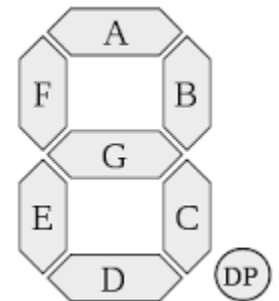
[Push button]	
PB0	E1
PB1	E2
PB2	D1
PB3	D2

[7-Segment]	
S5	A7
S4	B8
S3	A8
S2	B9
S1	A9
S0	A10
D7	A3
D6	B4
D5	B7
D4	A6
D3	A5
D2	B6
D1	B5
D0	A4

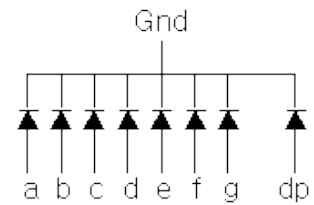
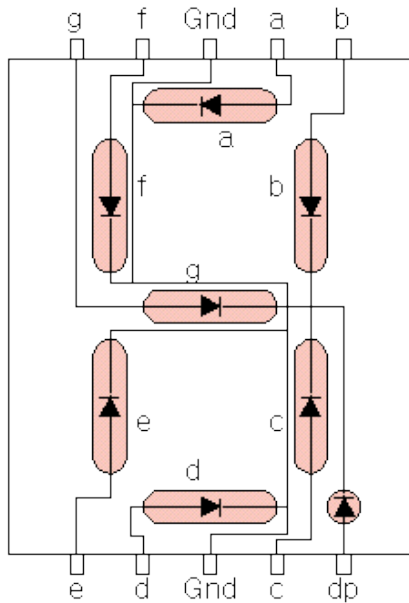
Selection

Data

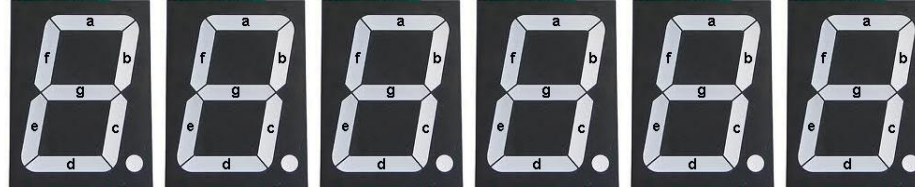
DP
G
F
E
D
C
B
A



FPGA – 7-Segment 장비와 Keypad

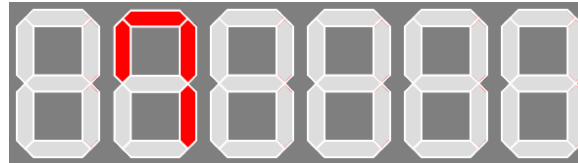


common-cathode type

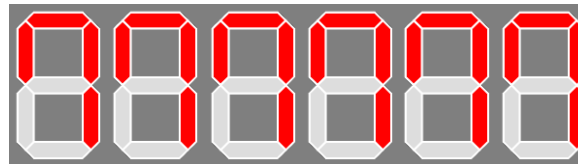


Selection 신호의 경우 0이면 해당 자릿수의 7-Segment를 작동

예) $\text{fnd_selection} = 6'b101111;$
 $\text{fnd_data} = 8'b11100100;$

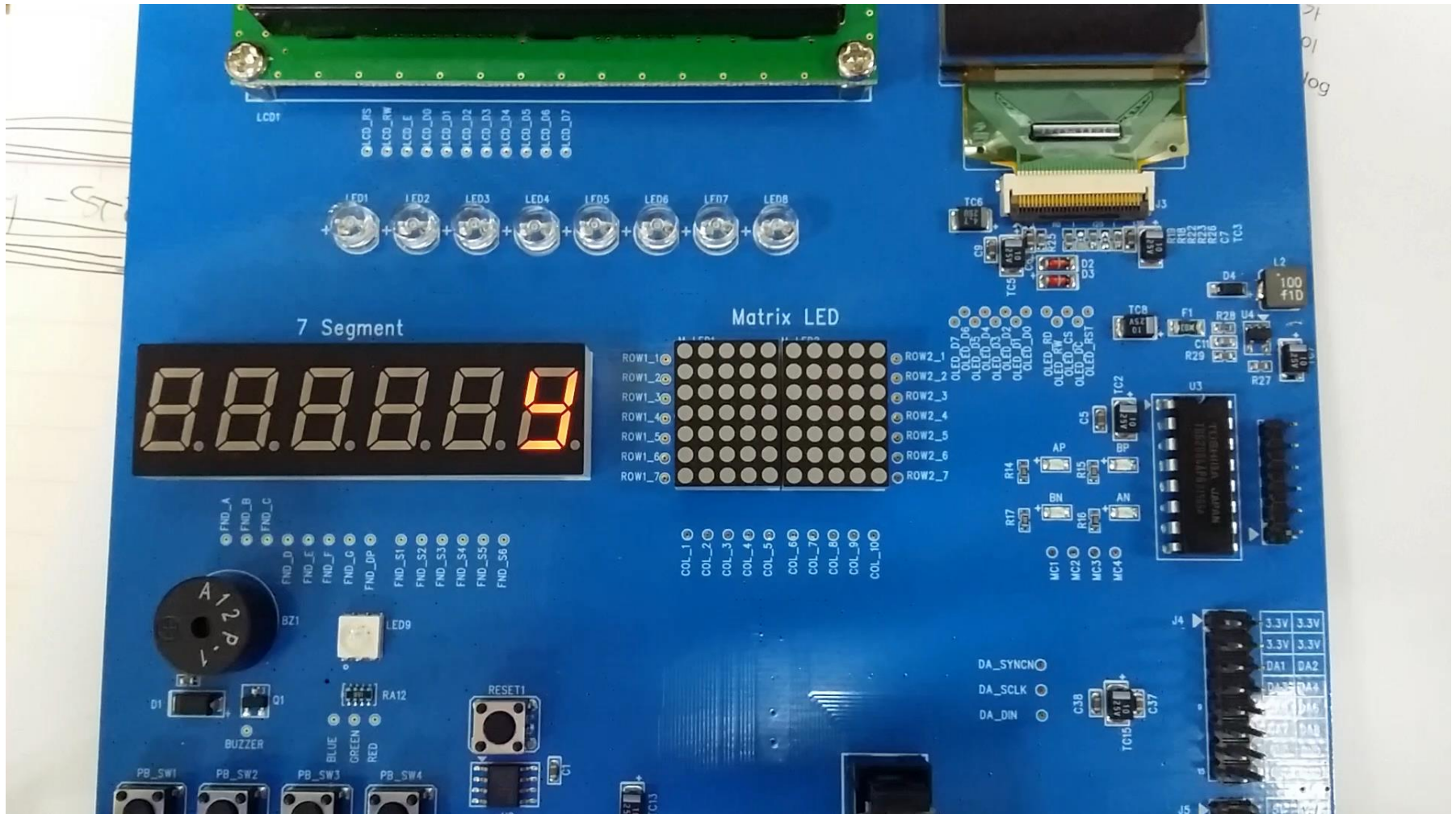


예-2) $\text{fnd_selection} = 6'b000000;$
 $\text{fnd_data} = 8'b11100100;$



YONSEI 전광판

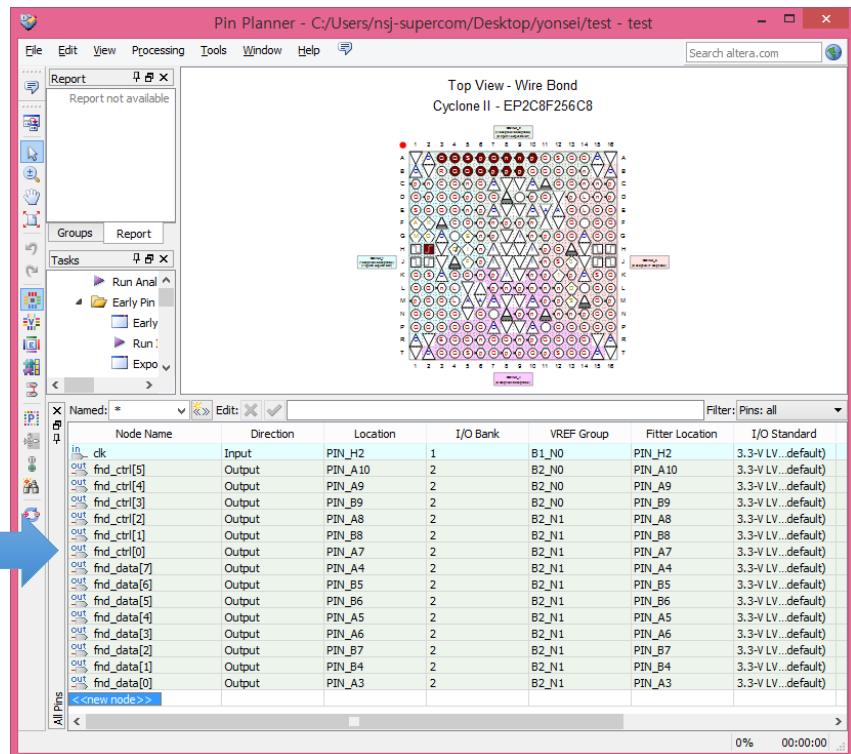
- 시연 동영상



설치 순서

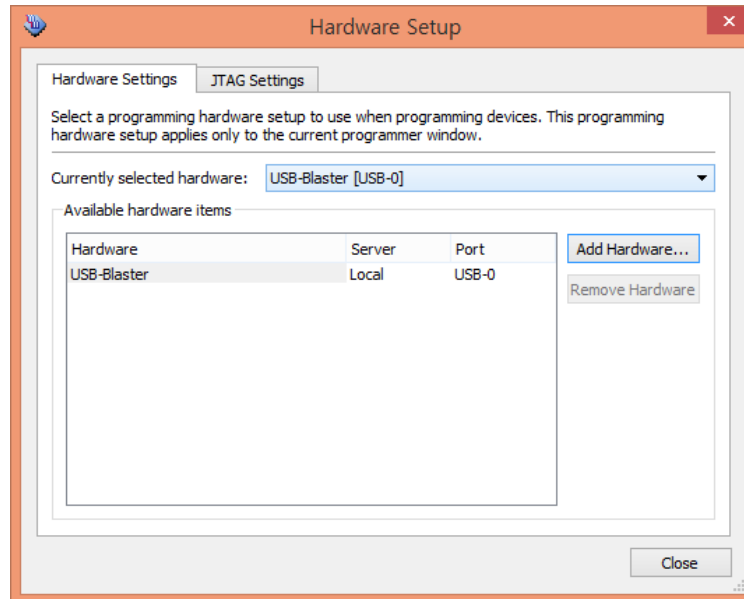
- Quartus II에서 모듈 설계
- 컴파일
- Simulation하여 정상 동작 하는지 검증
- Pin Assignment
- FPGA에 Program

- Assignment → Pin Planner 클릭
- Node Name: 설계한 모듈의 Input/Output
- Location: 실제 FPGA의 pin 번호
- 설정이 완료되면 프로젝트 저장 후 컴파일



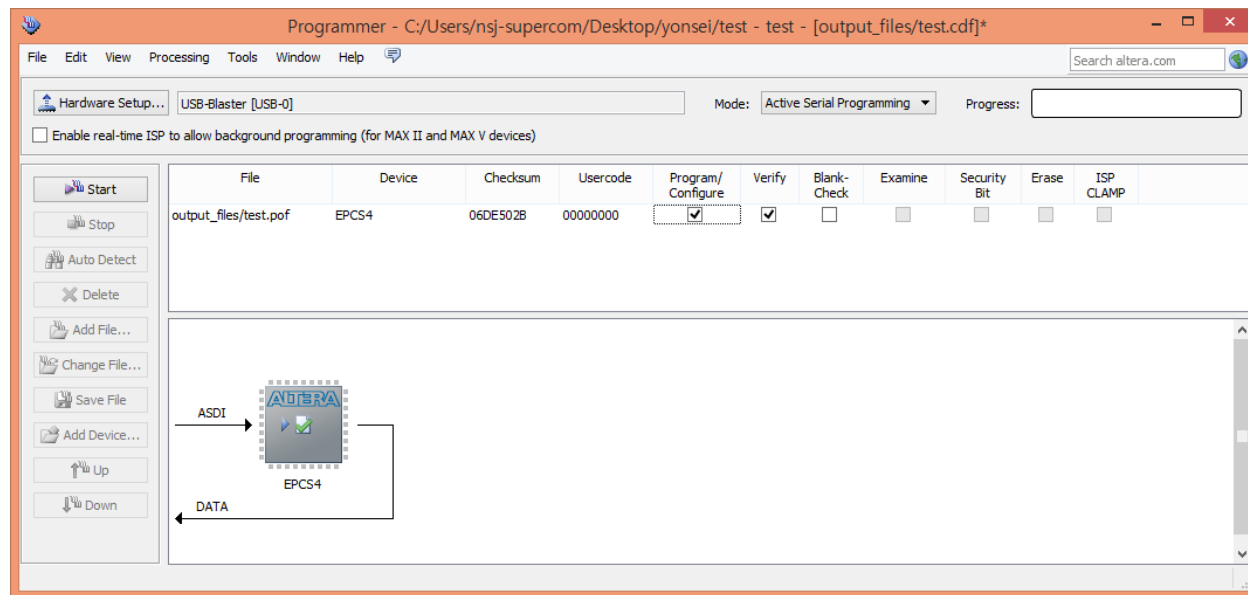
Program POF file into FPGA

- Tools -> Programmer 클릭
- 좌측 상단에 “Hardware Setup”을 클릭
- 아래 화면과 같은 창이 표시되면 “Currently selected hardware”를 “USB-Blaster [USB-X]”을 선택한 후 Close 클릭



Program POF file into FPGA

- “Add File…”을 클릭한 후 program할 POF파일을 선택
- 아래 화면과 같이 “Program/Configure”, “Verify”를 체크
- Start를 클릭하여 program 시작
- 완료되면 보드를 재시작하여 결과를 확인



CNDI eCube4412 페리보드 핀 배열

[LED]	
LED0	D3
LED1	D4
LED2	C4
LED3	D5
LED4	C5
LED5	D6
LED6	C6
LED7	D7

[Key Matrix]	
KEY_COL0	L10
KEY_COL1	M14
KEY_COL2	M12
KEY_COL3	M11
KEY_ROW0	N15
KEY_ROW1	N16
KEY_ROW2	M15
KEY_ROW3	M16

[Push button]	
PB0	E1
PB1	E2
PB2	D1
PB3	D2

[Buzzer]
J4

[Clock: 16MHz]
H2

[7-Segment]			
S0	A7	Selection	
S1	B8		
S2	A8		
S3	B9		
S4	A9		
S5	A10		
D0	A3	Data	a
D1	B4		b
D2	B7		c
D3	A6		d
D4	A5		e
D5	B6		f
D6	B5		g
D7	A4		DP

