

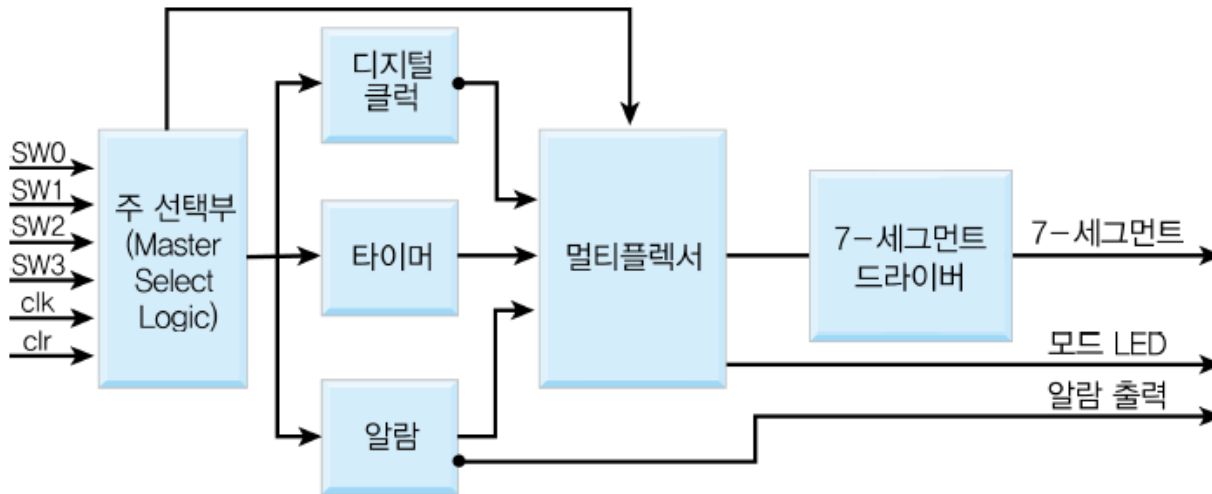


VLSI & System Lab.

논리회로 실험

Lecture 08

Project: 디지털 시계



[그림 6-1] 디지털 시계의 블록도

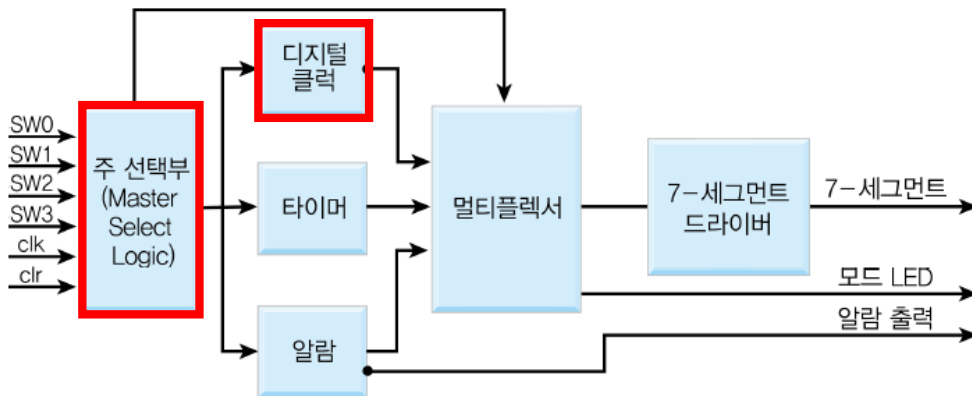
[표 6-1] 디지털 시계의 스위치와 동작 모드

	모드 선택(SW0)	선택 1(SW1)	선택 2(SW2)	선택 3(SW3)
0	시계 모드	현재 시간 출력		알람 ON/OFF
1	시간 설정	위치 선택 (Hour, Min)	시간 증가	
2	알람 설정	위치 선택 (Hour, Min)	시간 증가	
3	타이머(스톱위치)	스타트/스톱 (Start/Stop)	리셋	

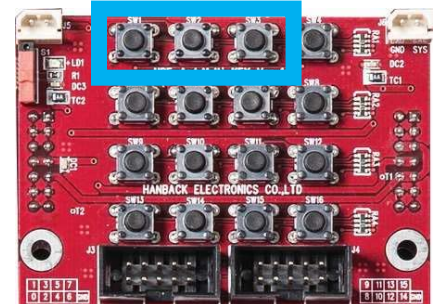


Practice 07

□ 디지털 클럭부 설계



[그림 6-1] 디지털 시계의 블록도



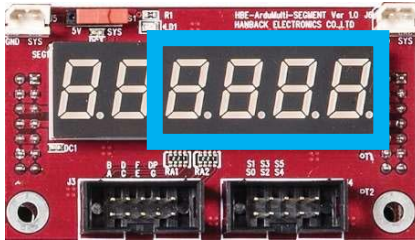
[표 6-1] 디지털 시계의 스위치와 동작 모드

모드 선택(SW0)	선택 1(SW1)	선택 2(SW2)	선택 3(SW3)
0	시계 모드	현재 시간 출력	
1	시간 설정	위치 선택 (Hour, Min)	시간 증가
2	알람 설정	위치 선택 (Hour, Min)	시간 증가
3	타이머(스톱워치)	스타트/스톱 (Start/Stop)	리셋

Practice 07

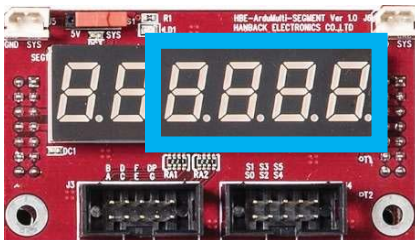
□ 디지털 클럭부 설계

- 시계 모드



0000
000 1
...
0059
0 100
...

- 시간 설정 모드



시계 모드



시간 설정 모드



초 (두자리) 선택



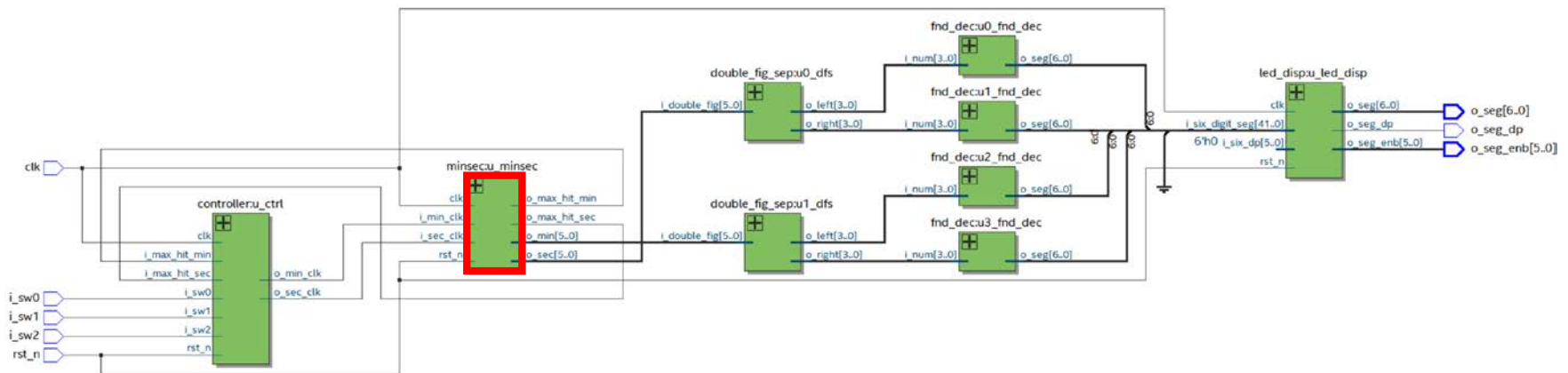
분 (두자리) 선택



선택된 파트의
숫자 증가

Practice 07

□ Top Module : top_hms_clock



- hourminsec

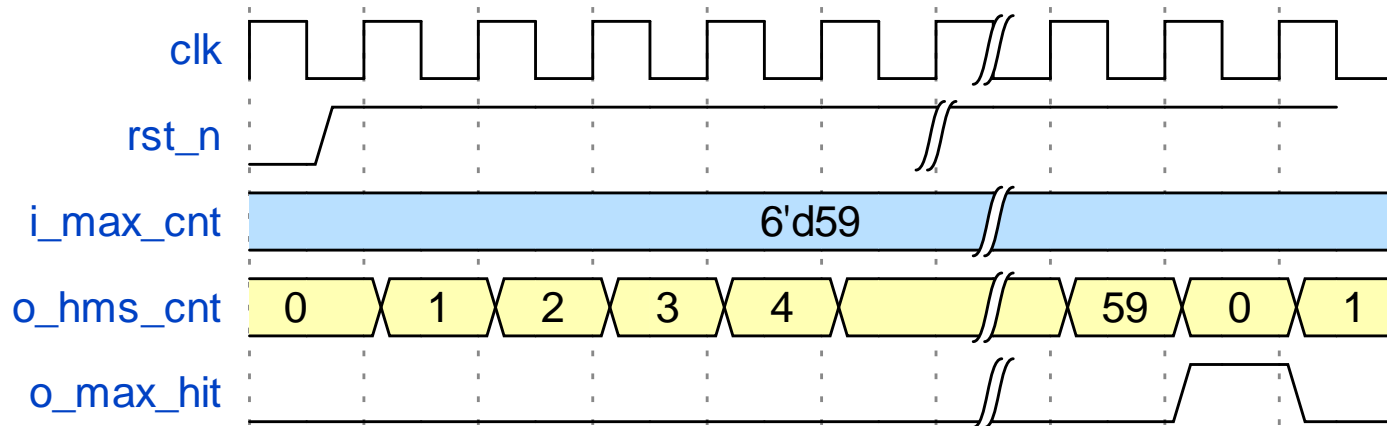
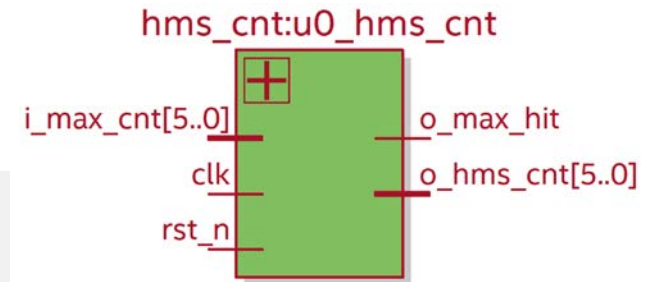
- 디지털 시계의 시:분:초 카운터 값 생성
- 분:초 부분이 59가 되었을 때 o_max_hit_sec/min 신호 생성

Practice 07

Module: hms_cnt

- wavedrom code

```
{signal: [{name: 'clk', wave: 'p.....|...'}, {name: 'rst_n', wave: '01.....|...', phase: 0.5}, {name: 'i_max_cnt', wave: '5.....|...', data: ["6'd59"]}, {name: 'o_hms_cnt', wave: '333333|333', data: ["0","1","2","3","4","", "59","0","1"]}, {name: 'o_max_hit', wave: '0.....|.10'}]}
```



Practice 07

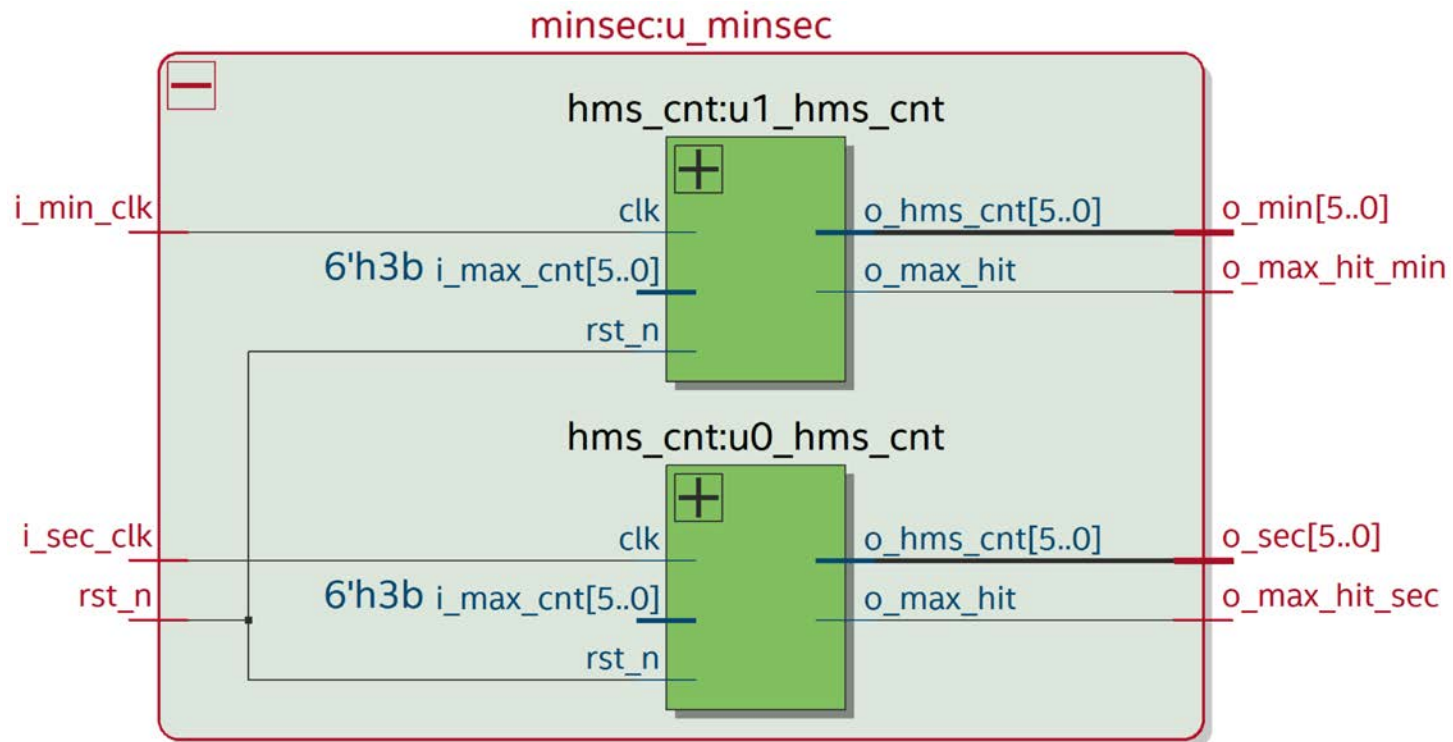
□ Module: hms_cnt

```
189 module hms_cnt(  
190     o_hms_cnt,  
191     o_max_hit,  
192     i_max_cnt,  
193     clk,  
194     rst_n);  
195  
196 output [5:0] o_hms_cnt    ;  
197 output      o_max_hit    ;  
198  
199 input  [5:0] i_max_cnt    ;  
200 input      clk            ;  
201 input      rst_n         ;  
202  
203 reg      [5:0] o_hms_cnt  ;  
204 reg      o_max_hit       ;  
  
205 always @(posedge clk or negedge rst_n) begin  
206     if(rst_n == 1'b0) begin  
207         o_hms_cnt <= 6'd0;  
208         o_max_hit <= 1'b0;  
209     end else begin  
210         if(o_hms_cnt >= i_max_cnt) begin  
211             o_hms_cnt <= 6'd0;  
212             o_max_hit <= 1'b1;  
213         end else begin  
214             o_hms_cnt <= o_hms_cnt + 1'b1;  
215             o_max_hit <= 1'b0;  
216         end  
217     end  
218 end  
219  
220 endmodule
```



Practice 07

Module: minsec



Practice 07

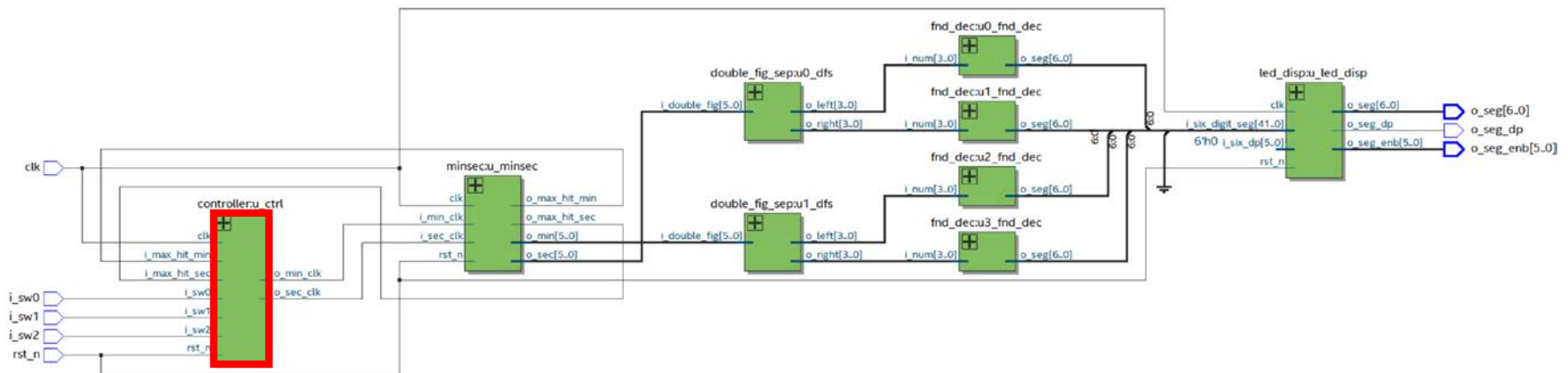
□ Module: minsec

```
360 module minsec( o_sec,
361                 o_min,
362                 o_max_hit_sec,
363                 o_max_hit_min,
364                 i_sec_clk,
365                 i_min_clk,
366                 clk,
367                 rst_n);
368
369 output [5:0] o_sec      ;
370 output [5:0] o_min      ;
371 output      o_max_hit_sec ;
372 output      o_max_hit_min ;
373
374 input      i_sec_clk      ;
375 input      i_min_clk      ;
376
377 input      clk             ;
378 input      rst_n           ;
379
380 hms_cnt      u0_hms_cnt(
381                 .o_hms_cnt      ( o_sec      ),
382                 .o_max_hit      ( o_max_hit_sec ),
383                 .i_max_cnt      ( 6'd59      ),
384                 .clk             ( i_sec_clk   ),
385                 .rst_n           ( rst_n      ));
386
387 hms_cnt      u1_hms_cnt(
388                 .o_hms_cnt      ( o_min      ),
389                 .o_max_hit      ( o_max_hit_min ),
390                 .i_max_cnt      ( 6'd59      ),
391                 .clk             ( i_min_clk   ),
392                 .rst_n           ( rst_n      ));
393
394 endmodule
```



Practice 07

□ Top Module : top_hms_clock



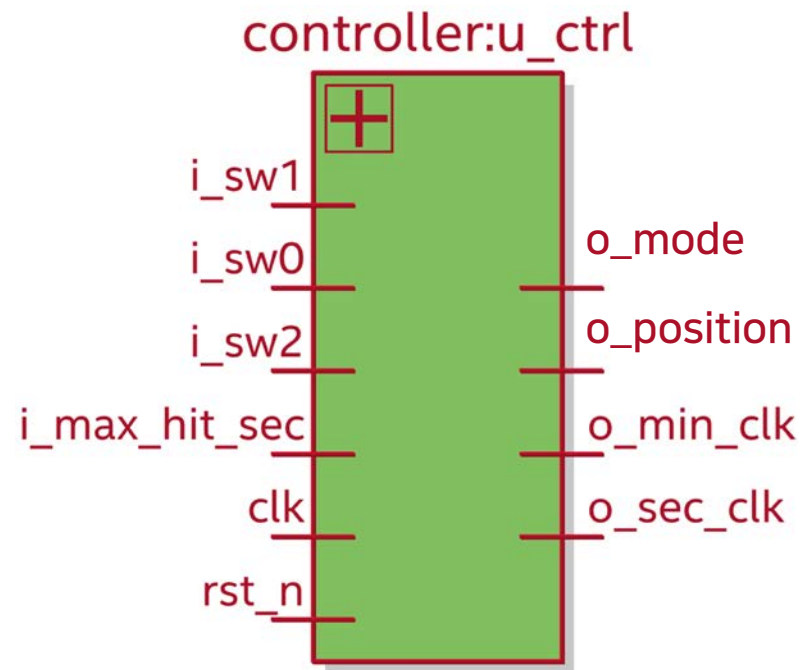
- controller

- 시계의 모드 (mode) 및 설정 부분 (position) 제어 : i_sw0 ~ i_sw2
- 시:분:초 카운터를 위한 클럭 제공 (o_min_clk, o_sec_clk)

Practice 07

□ Module: controller

- 2개의 nco (numerical controlled oscillator)
 - debounce 용 clock 생성
 - 디지털 시계용 clk_1hz 생성
- 모드/설정부분 제어
 - o_mode 신호
 - o_position 신호
- Debounce 모듈
 - 스위치의 bounce 현상 제거



Practice 07

Module: controller

```
276 parameter    MODE_CLOCK = 1'b0    ;
277 parameter    MODE_SETUP = 1'b1    ;
278
279 parameter    POS_SEC = 1'b0        ;
280 parameter    POS_MIN = 1'b1        ;
281
307 reg          o_mode                ;
308 always @(posedge i_sw0 or negedge rst_n) begin
309     if(rst_n == 1'b0) begin
310         o_mode <= MODE_CLOCK;
311     end else begin
312         o_mode <= o_mode + 1'b1;
313     end
314 end
```

```
315
316 reg          o_position              ;
317 always @(posedge i_sw1 or negedge rst_n) begin
318     if(rst_n == 1'b0) begin
319         o_position <= POS_SEC;
320     end else begin
321         o_position <= o_position + 1'b1;
322     end
323 end
```



시계 모드
↕
시간 설정 모드



초 (두자리) 선택
↕
분 (두자리) 선택



선택된 파트의
숫자 증가



Practice 07

Module: controller

```
332 reg          o_sec_clk          ;
333 reg          o_min_clk          ;
334 always @(*) begin
335     case(o_mode)
336     MODE_CLOCK : begin
337         o_sec_clk = clk_1hz;
338         o_min_clk = i_max_hit_sec;
339     end
340     MODE_SETUP : begin
341         case(o_position)
342         POS_SEC : begin
343             o_sec_clk = ~i_sw2;
344             o_min_clk = 1'b0;
345         end
346         POS_MIN : begin
347             o_sec_clk = 1'b0;
348             o_min_clk = ~i_sw2;
349         end
350     endcase
351 end
352 endcase
353 end
```



↑
시계 모드
↕
시간 설정 모드



↑
초 (두자리) 선택
↕
분 (두자리) 선택

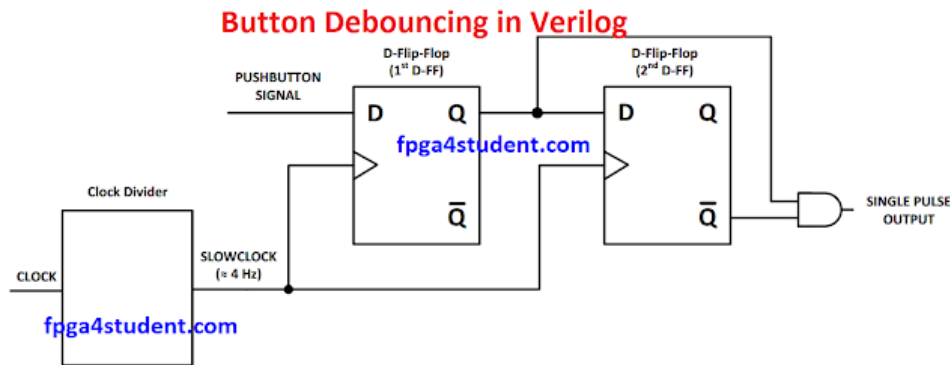


↑
선택된 파트의
숫자 증가

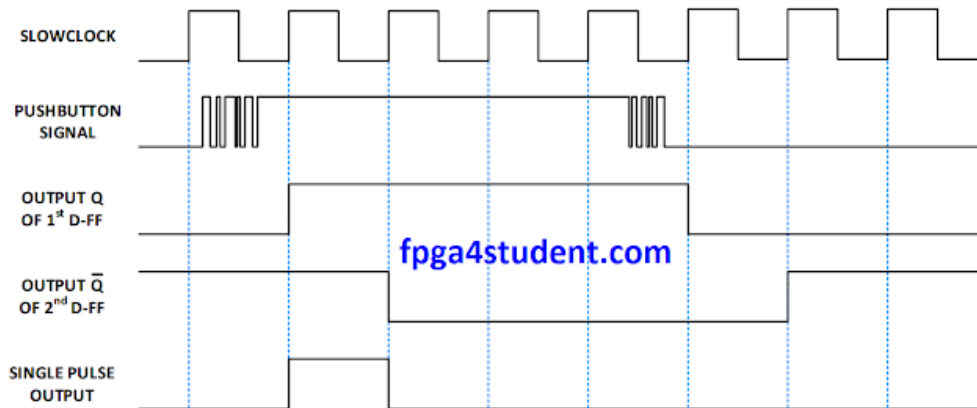
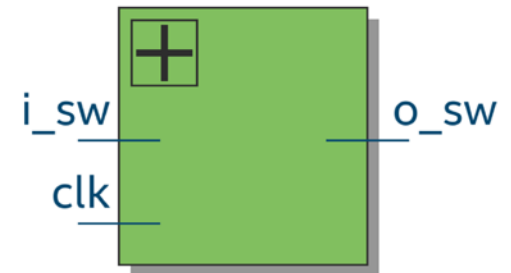


Practice 07

Module: debounce



debounce:u2_debounce



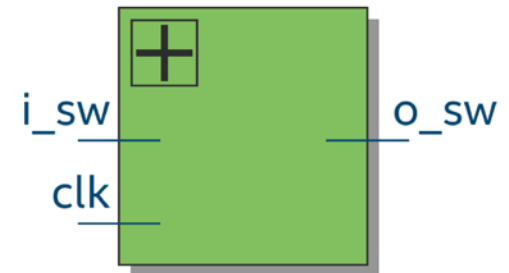
Practice 07

Module: debounce

```
222 module debounce(  
223     o_sw,  
224     i_sw,  
225     clk);  
226 output      o_sw      ;  
227  
228 input       i_sw      ;  
229 input       clk       ;  
230  
231 reg         dly1_sw    ;  
232 always @(posedge clk) begin  
233     dly1_sw <= i_sw;  
234 end
```

```
236 reg         dly2_sw    ;  
237 always @(posedge clk) begin  
238     dly2_sw <= dly1_sw;  
239 end  
240  
241 assign      o_sw = dly1_sw | ~dly2_sw;  
242  
243 endmodule
```

debounce:u2_debounce



실습 내용

□ 디지털 시계 (분:초) 부분 설계 - Debounce 적용 전

- GitHub에 제공된 소스코드 사용 - 다른 모듈 건드리지 말 것
- top module(top_hms_clock) 만 채워서 설계
- 이번 실습은 Test Bench 생략

□ FPGA

- 스위치의 Bounce 현상에 대해 관찰
- 코드를 수정하여 Debounce 적용 후 스위치 테스트



실습 내용

□ Quiz

- 오른쪽 코드에서 i_sw2를 누르는 순간이 아닌 때는 순간 숫자가 증가하게 하려면?
 - README.md에 작성 후 제출

```
332 reg          o_sec_clk          ;
333 reg          o_min_clk          ;
334 always @(*) begin
335     case(o_mode)
336         MODE_CLOCK : begin
337             o_sec_clk = clk_1hz;
338             o_min_clk = i_max_hit_sec;
339         end
340         MODE_SETUP : begin
341             case(o_position)
342                 POS_SEC : begin
343                     o_sec_clk = ~i_sw2;
344                     o_min_clk = 1'b0;
345                 end
346                 POS_MIN : begin
347                     o_sec_clk = 1'b0;
348                     o_min_clk = ~i_sw2;
349                 end
350             endcase
351         end
352     endcase
353 end
```



실습 내용

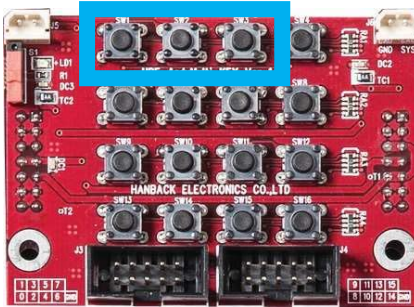
□ Project Guide : 질의응답 불가

- 시:분:초에 대한 디지털 시계 완성
- 설정 모드에서 7-segment의 dp를 활용한 설계
 - 예) 초 설정 시 - 초 부분의 dp led를 점등
- Blink 모드 개발
 - 설정 모드에서 설정 부분을 깜빡이게 Display

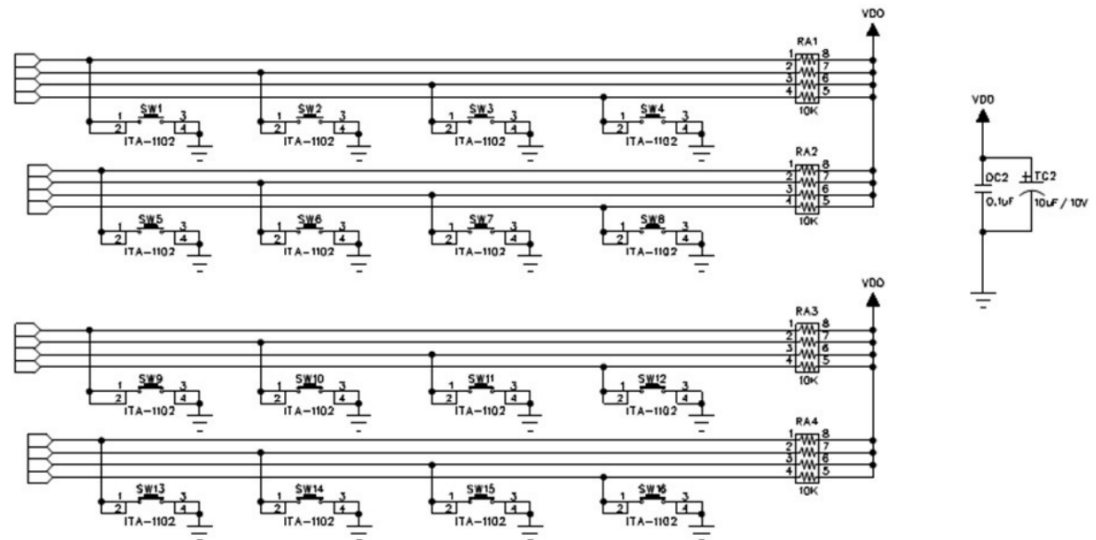


Key Switch 관련

□ Pin Assignment



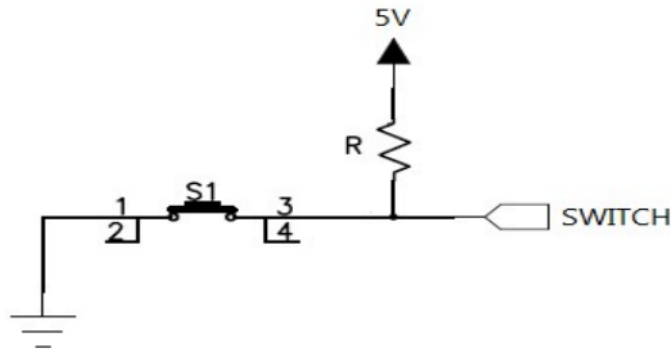
Key(2)	GPIO#0 26	E11	SW 3 : S2
Key(1)	GPIO#0 25	D9	SW 2 : S1
Key(0)	GPIO#0 24	C9	SW 1 : S0



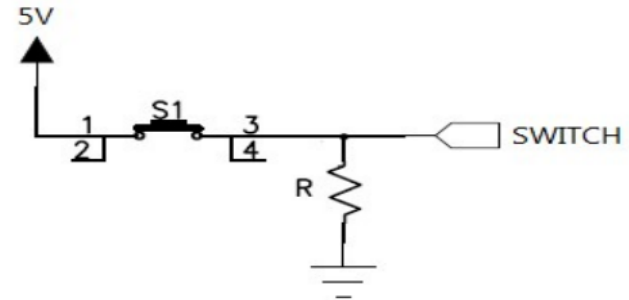
Key Switch 관련

□ Pull-up Switch & Pull-down Switch

그림 7 - 2 | 회로 구성에 따른 스위치 동작



Pull up resistor

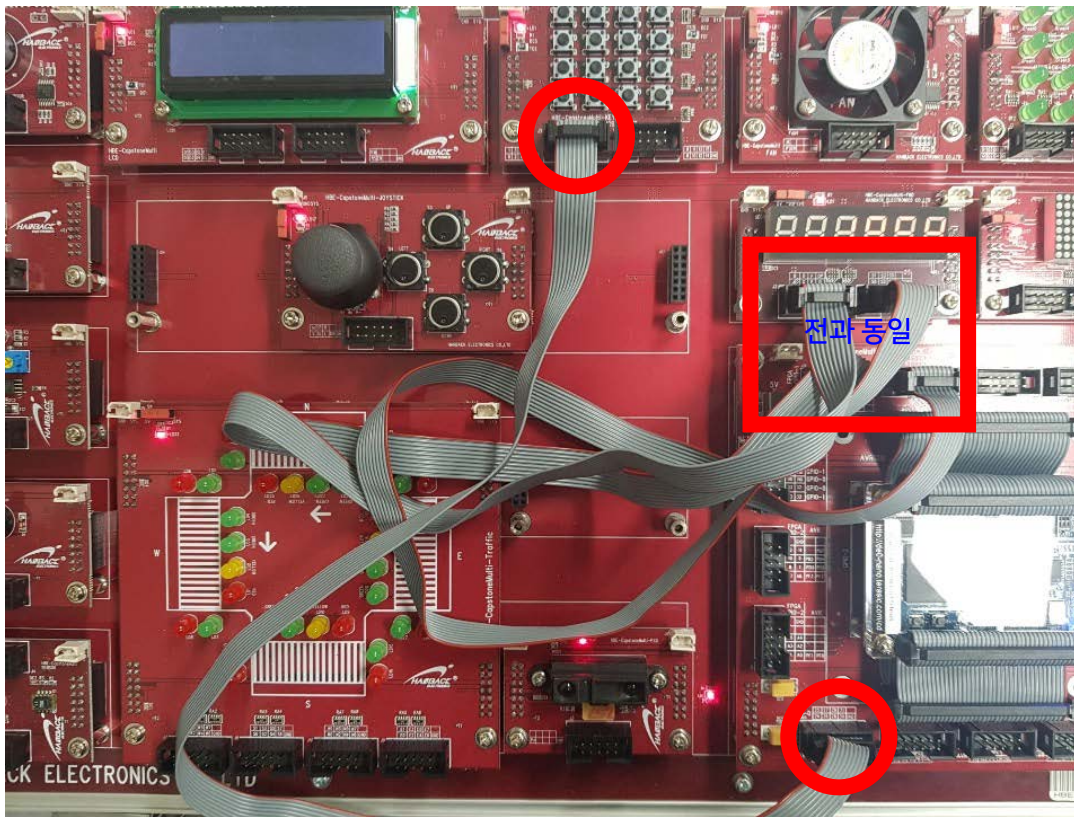


Pull down resistor



케이블 연결

□ Pull-up Switch & Pull-down Switch



README.md

github.com/woongchoi-sookmyung/LogicDesign 참조

