

Final Project. 디지털 시계 만들기

2021 Fall Logic Design LAB

Department of Computer Science and Engineering

Seoul National University

디지털 시계 디자인

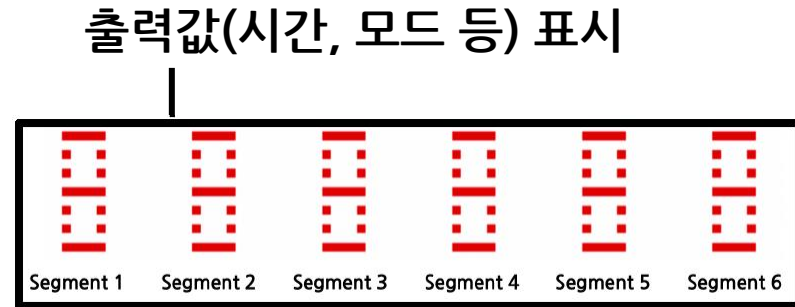
- 목표
 - 이론 시간과 실습 시간에 학습한 내용을 활용해 디지털 시계를 구현
- 기본 구현
 - 시계 기능
 - 현재 시간을 정확히 1초 마다 증가시킨다.
 - 임의의 시간과 분으로 시간을 설정한다.
 - 알람 기능
 - 알람시간 (시/분 단위)을 설정한다.
 - 알람으로 설정해놓은 시간이 되면 시계 기능에서 이를 표시한다.
 - 스톱워치 기능
 - 1/100초 단위까지 측정하는 스톱워치를 구현한다.

디지털 시계 디자인: 추가 구현

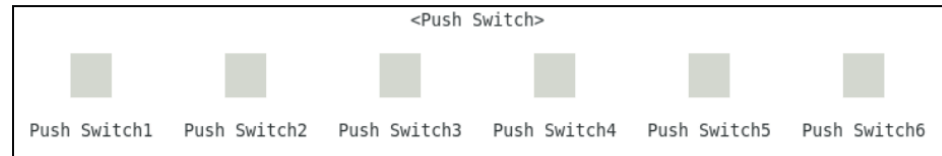
- 자유롭게 기본 구현 사항에 더해 구현 가능한 부분을 확장하여 구현
- 예시
 - 알람 snooze 기능
 - 5분 뒤에 다시 알람 울림
 - 스톱워치 lap 저장 기능
 - 진행 중인 스톱워치를 중단하지 않고 2개 이상의 lap을 저장
 - 세계시간 표시 기능
 - 현재 시간(서울)을 기준으로 특정 나라의 현재 시간 표시, LED를 이용해 나라(영국, 미국, 중국 등) 표현
 - 설정 시 비밀번호 사용 기능
 - 현재 시간을 설정하려면 비밀번호를 맞춘 상태이어야 함 (비밀번호 설정 관련 기능도 필요)
 - 그 외의 다양한 구현 가능
- 추가 구현 시 새로 구현된 기능이 기능적으로 새로운 구현이 아니거나 난이도가 낮은 경우 추가 구현으로 인정하지 않음 (예. 시간의 16진수 표시, push button 누를 때 LED 점등 효과)

디지털 시계 입출력 개요

- 7-segment LED (J1, 2, 3, 4, 5, 6)
 - 기본적으로 시간을 10진수로 표시
 - 추가 구현 시 모드 표현 등에 사용 가능



- Push Switch (SW1~6)
 - SW1(MODE): 모드 변경 버튼
 - SW2(SET): 설정 혹은 확인 버튼
 - SW3(OP1): 모드에 따른 기능1 수행
 - SW4(OP2): 모드에 따른 기능2 수행 or 기능이 없는 경우 추가 구현에 사용 가능
 - SW5(OP3): 모드에 따른 기능3 수행 or 기능이 없는 경우 추가 구현에 사용 가능
 - SW6(RESET): 디지털 시계의 RESET



- 6개의 색상 LED, 10개의 DIP switch 또한 기본 구현(알람) 또는 추가 구현에 사용

기본 구현: 디지털 시계의 초기 상태

- 입력

- 어떤 상태에 있더라도 SW6(RESET)을 누르게 되면 초기 상태로 돌아온다.
- 디지털 시계의 초기 상태
 - 모드: 시계 모드
 - RESET이 눌러져 있는 동안: 00시 00분 00초 유지
 - 설정한 알람 시간 등 모든 설정 값 초기화
- 디지털 시계 동작 시 무조건 초기조건으로 RESET을 누르고 시작

기본 구현: 시간 표시법 (1)

- 시계 기능과 알람 기능
 - 각각 현재의 시간과 알람 시간을 24시간 표현법과 12시간 표현법으로 표시
 - Leading zero 표현 (e.g. 1시 대신 01시로 표현)
- 24시간 표현법
 - 시, 분, 초 순서대로 2자리씩 출력

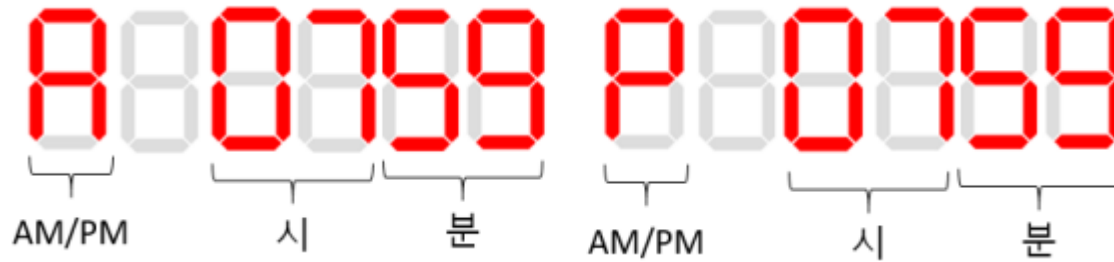


- 자정의 경우 오른쪽 사진과 같이 표현
(00시 00분 00초)



기본 구현: 시간 표시법 (2)

- 12시간 표현법
 - AM/PM, 시, 분 순서대로 출력하고 초 단위는 생략

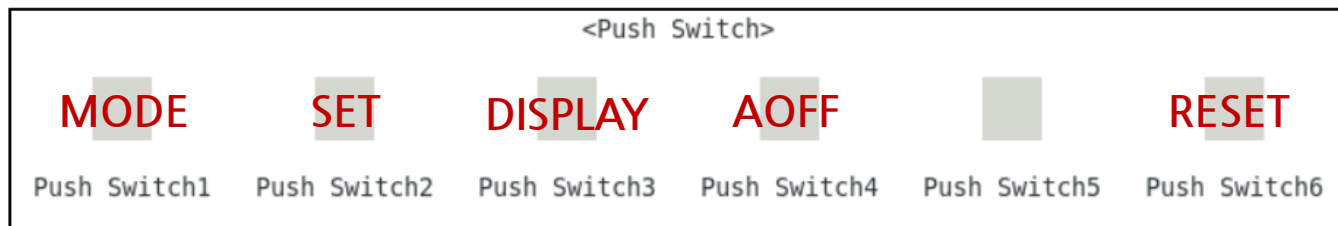


- 정오는 PM 12시 00분, 자정은 AM 12시 00분으로 표현



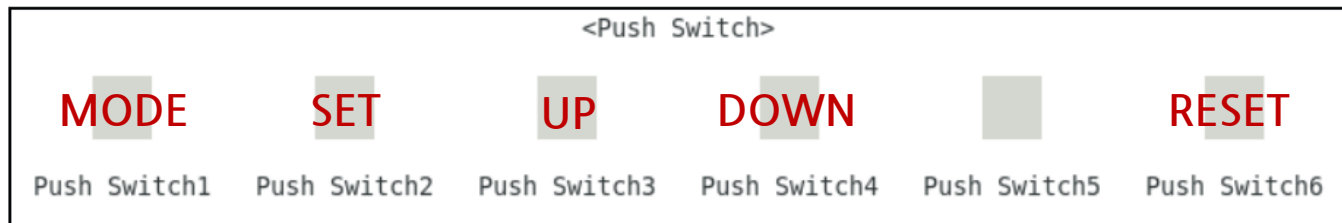
기본 구현: 시계 기능

- 시계 표현 방법
 - 시간 표현은 12시간/24시간 표시법으로 표현
- 입력
 - MODE: 현재의 시계가 진행되는 상태에서 알람 시간 표시 모드로 진입
 - SET: 시계의 시간 설정으로 진입
 - DISPLAY: 12/24시간 표시법 간의 변환
 - AOFF: 알람이 울리고 있는 경우 알람을 끄
 - 울리지 않은 경우에는 아무 동작 없음
 - RESET: 디지털 시계의 초기 상태로 돌아감



기본 구현: 시계 시간 설정 기능 (1)

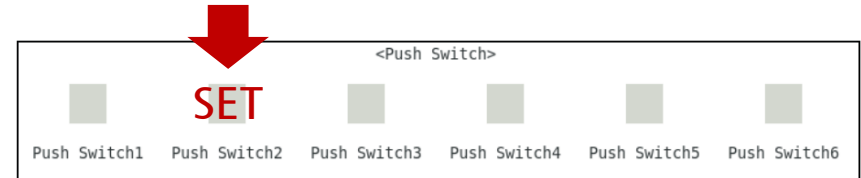
- 시계 시간 설정 표현 방법
 - 시간 표현은 시간 설정으로 진입하기 전 시계의 표시 설정에 따라 표시
 - 24시간 표현법에서 진입할 경우 24시간 표현법으로 표시
 - 12시간 표현법에서 진입할 경우 12시간 표현법으로 표시
- 입력
 - MODE: 현재의 시간 설정이 저장되고 시계가 진행되는 상태에서 알람 설정 모드로 진입
 - SET: 현재 설정 중인 항목을 저장
 - UP: 현재 설정 중인 항목의 값을 올림 (예. 1시 → 2시)
 - DOWN: 현재 설정 중인 항목의 값을 내림 (예. 2시 → 1시)
 - RESET: 디지털 시계의 초기 상태로 돌아감



기본 구현: 시계 시간 설정 기능 (2)

■ 동작: 12시간 표현법

- 12시간 표현법: AM/PM - 시 - 분 순서대로 설정
- 현재 설정 중인 항목은 점멸
 - UP/DOWN을 누르면 점멸하고 있는 부분만 변경
 - 분 항목까지 설정하고 SET을 누르면 시계로 돌아옴
 - 설정한 시간의 00초부터 정확하게 시작



시계 모드로 복귀

SET

기본 구현: 시계 시간 설정 기능 (3)

- 동작: 24시간 표현법
 - 24시간 표현법: 시 - 분 순서대로 설정 (초는 00으로 고정)
 - 현재 설정 중인 항목은 점멸
 - UP/DOWN을 누르면 점멸하고 있는 부분만 변경됨
 - 분 항목까지 설정하고 SET을 누르면 시계로 돌아옴
 - 설정한 시간의 00초부터 정확하게 시작



기본 구현: 알람 기능 (1)

■ 시계 표현 방법

- 시간 표현은 앞서 설명한 12시간/24시간 표현법으로 표현



알람이 없는 경우



알람이 있는 경우(12시간 표현법 가정)

■ 알람 표시 방법

- 설정된 알람과 시계의 현재 시간이 일치할 경우 알람이 울림
- 시계 모드에서 AOFF를 눌러야만 알람이 정지함
- 시간이 지나도 AOFF가 눌리지 않으면 계속 알람이 울림
- 알람이 울리는 것은 모든 7segment가 점멸하는 방식으로 기본 구현



점멸

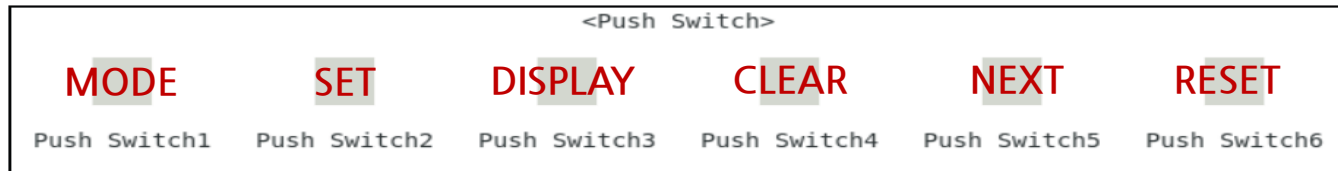
■ 최대 10개의 알람을 저장할 수 있어야 함

기본 구현: 알람 기능 (2)

■ 입력

- Push switch

- MODE: 현재의 알람 기능이 지속되는 상태에서 스톱워치 모드로 진입
- SET: **알람 시간 설정**으로 진입 (*알람 시간 설정 방식은 시계 시간 설정과 동일*)
- DISPLAY: 12/24시간 표시법 간의 변환
- CLEAR: 현재 설정되어 있는 알람을 비활성화
- NEXT: 여러 개의 알람이 사용중일 경우 번호순으로 다음 알람의 정보를 출력
- RESET: 디지털 시계의 초기 상태로 돌아감



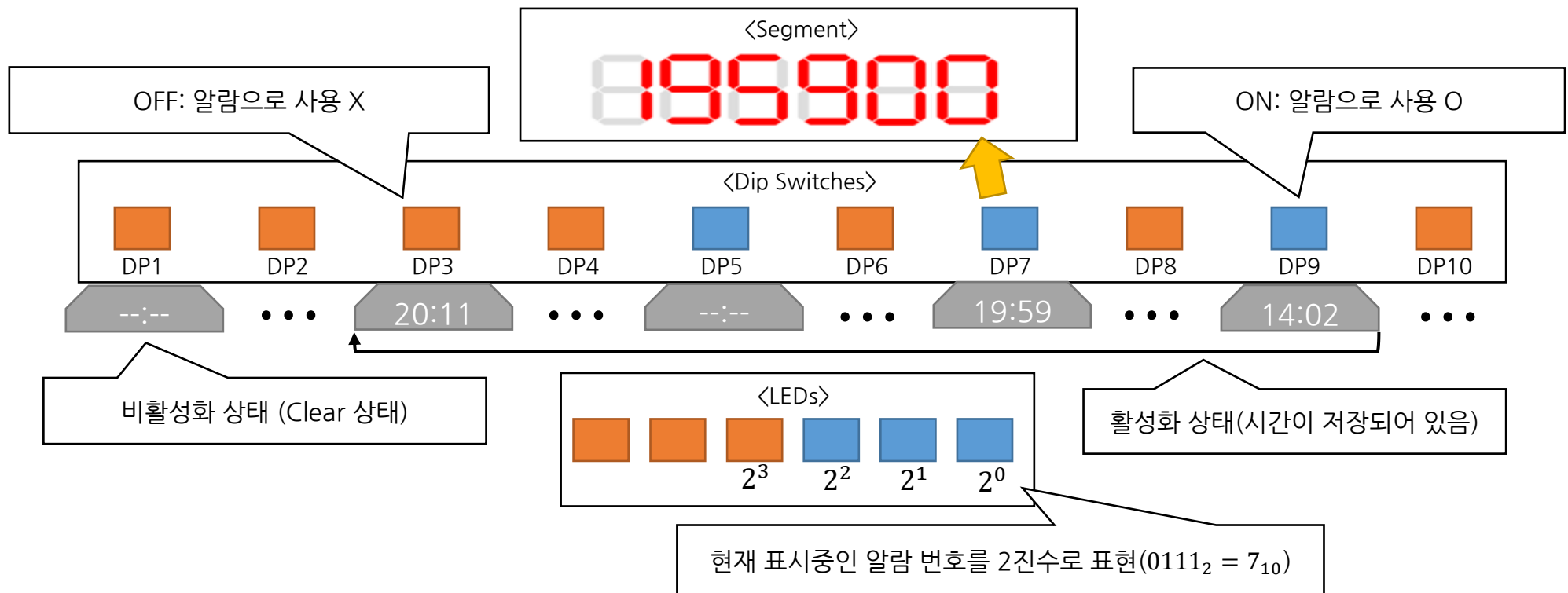
- Dip Switch

- ON: 알람 번호에 해당하는 알람들을 사용
- OFF: 알람 번호에 해당하는 알람들을 사용하지 않음

기본 구현: 알람 기능 (3)

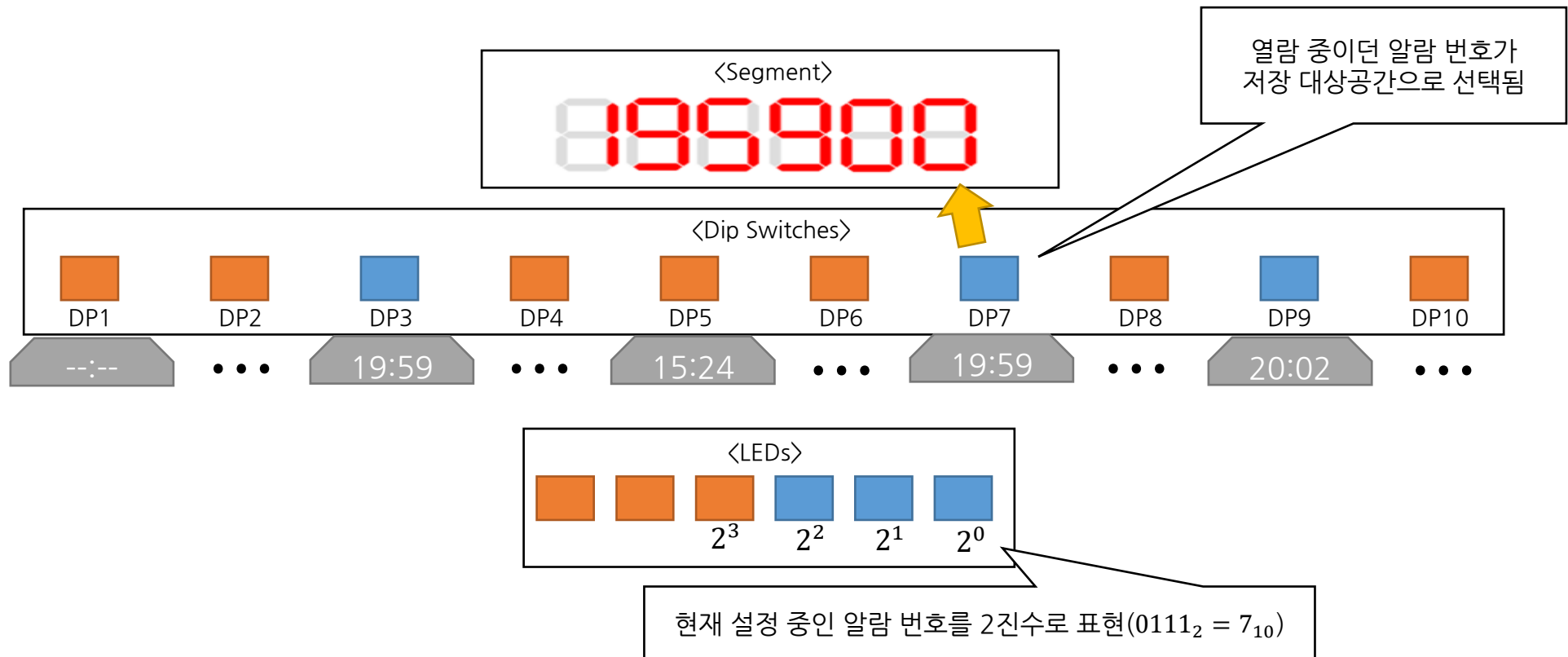
■ 알람 시간 표시 상세

- LED를 이용해 현재 모니터링 중인 알람 번호 출력 (이진수 출력: 1=LED ON, 0=LED OFF)
- 처음 알람 시간 표시 모드에 진입했을 때는 ON 되어 있는 알람 중 가장 아래 알람 표시
- 아무 알람도 ON되어 있지 않은 경우 알람이 없는 경우로 출력



기본 구현: 알람 기능 (4)

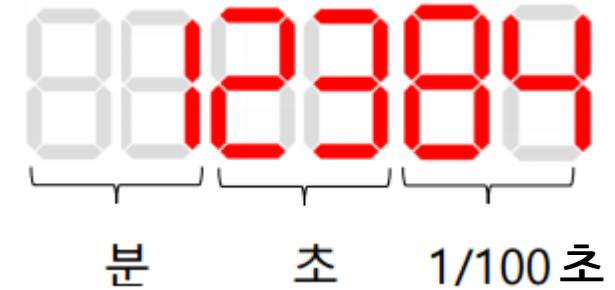
- 알람 시간 설정 상세 (표시 모드에서 set 버튼을 눌러 진입)
 - LED를 이용해 현재 모니터링 중인 알람 번호 출력 (이진수 출력: 1=LED ON, 0=LED OFF)
 - 알람 시간 중복 설정 허용됨 (AOFF시 해당 알람들 모두 꺼짐)
 - 시간 설정 방법은 시계 모드에서의 설정 방법과 동일



기본 구현: 스톱워치 기능

■ 스톱워치 표현 방법

- 앞의 두 자리는 분, 중간 두 자리는 초, 나머지 두 자리는 1/100초 단위
- 분/초 단위는 Leading Zero 제거, 1/100초 단위는 Leading Zero 미 제거

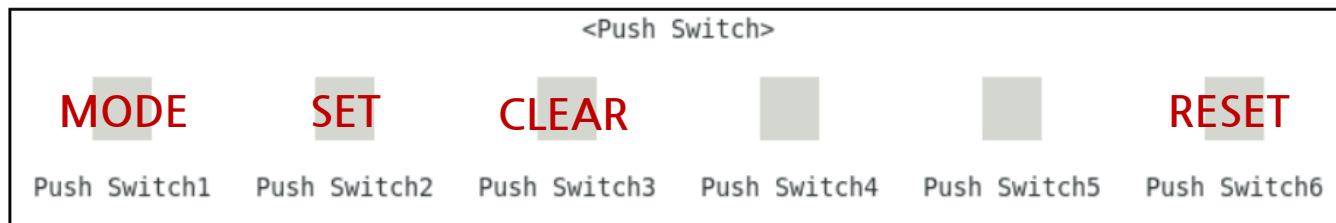


■ 입력

- MODE: 현재의 스톱워치가 진행되는 상태에서 시계 모드로 진입
- SET: 스톱워치를 시작하거나 일시 정지
- CLEAR: 스톱워치가 정지한 상태에서 시간을 00분 00.00초로 초기화
- RESET: 디지털 시계의 초기 상태로 돌아감

■ 동작

- 59분 59.99초에 다다른 경우 더 이상 진행하지 않고 멈추고, 이후 SET을 눌러도 스톱워치가 진행하지 않음



유의 사항

- 구현 시 참고사항
 - 자세한 구현 세부는 ETL에 같이 공지된 스펙 문서 및 예시 시뮬레이터파일을 참조
 - 불충분하다고 생각하는 혹은 추가하고 싶은 스펙이 있다면 스스로 스펙을 정하고 이를 구현
 - 스스로 정한 스펙이 있다면 이를 보고서에 작성하여 제출
- 권장사항
 - 과제의 복잡도가 높으므로 **모듈화** 하여 구현을 시작할 것
 - 각 모듈 마다 **시뮬레이션**을 통해 동작을 확인할 것

평가: 배점 (1)

■ 기본 구현

기능	세부	배점
시계 기능	1초 단위의 시/분/초 시계 기능	70 점
알람 기능	시계의 시간에 따라 알람을 설정/울리는 기능	90 점
스톱워치 기능	1/100초 단위의 스톱워치 기능	40 점
총점		200 점

■ 추가 구현 (최대 추가 점수: 50점)

- 추가 구현은 난이도와 구현 개수에 따라, **50점 내에서 차등 부여**
- 추가 구현으로 얻은 점수는 **가산 점**으로 총점에 더하는 방식으로 평가됨
(e.g. 기본 구현 190점, 추가 구현 30점 획득한 경우 최종 점수는 220점)
- 시뮬레이터의 Segment, Push Switch, DIP Switch 및 LED 모두 사용 가능

평가: 배점 (2)

- 추가 구현 점수로 인해 최종 점수가 200점을 넘길 경우, 초과된 점수는 **이전 Assignment들에서 감점된 점수를 Make up하는 점수로 활용됨**
 - 논리설계 Assignment는 Assignment 1, 2, 3, 5와 Mini Project, Final Project로 구성됨
 - 예시) 최종 점수가 220점인 경우, 20점 초과함. 지난 Assignment들에서 총 감점된 점수가 50점일 때, 초과된 20점을 사용해 최종적으로 30점 감점된 것으로 처리됨
- 논리설계 Assignment 총 합계 만점을 초과한 경우, **초과된 점수를 제하고 만점으로 처리**
 - 예시) 논리설계 Assignment 총 합계 만점을 N점이라고 가정하였을 때, 논리설계 Assignment 총 합계 점수가 (N+10)점인 학생의 경우, 10점은 버려지고 최종 점수는 N점으로 처리됨

보고서 제출 및 평가 방법

- 제출 기한: **12월 19일(일요일) 23시 59분 전까지** ETL에 업로드
- 프로젝트 폴더와 보고서를 압축하여 제출
 - 보고서
 - 구현 사항, 모듈 설명, 추가 구현 설명/스펙 포함 (4장 내외)
예) 모듈 연결, state minimization 등 설계 사항
 - 프로젝트 폴더
 - Cleanup Project Files 하여 제출
- **성적 입력 기한 사정 상 Bonus Day 사용 불가**
- **Copy 적발 시 F**

Help Desk

- Q & A Session

- 시간: 12월 2(목), 9(목) 19시 00분 ~ 19시 50분, 16(목) 19시 00분 ~ 20시 50분
- 장소: Zoom 세션 (온라인 진행), 해당 날 링크 공지
- 공지되는 링크를 통해 대기실에서 대기하고 있으면 한 학생 씩 입장해 질의 응답하는 방식으로 진행
- **Q & A Session 중 출석 확인 없음**
- 그 외의 시간에는 질의응답 게시판이나 조교 e-mail (tas0221@davinci.snu.ac.kr)을 이용해 문의

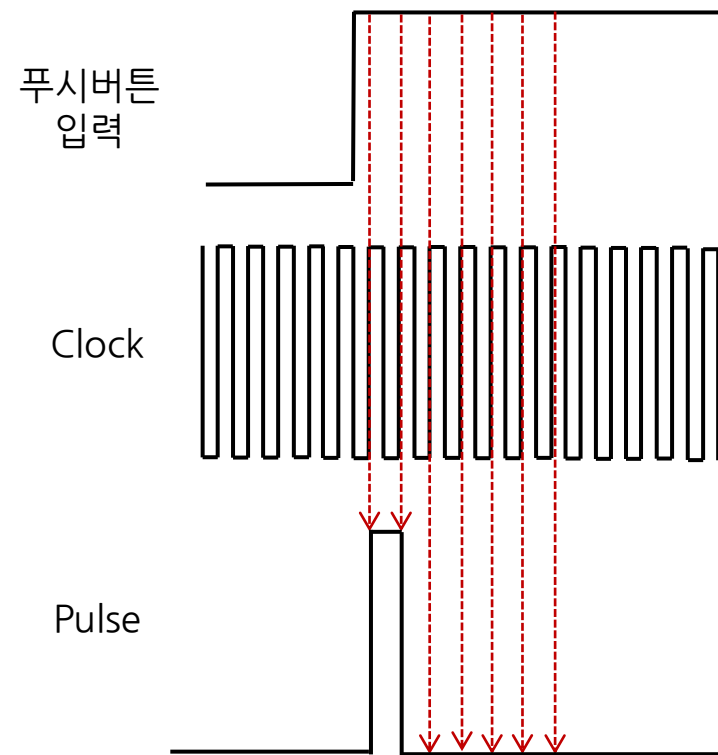
시뮬레이터 관련 추가 유의 사항

Pulse Generator

- 높은 주파수의 clock에 synchronous한 FSM 설계
 - 푸시 버튼 입력이 **여러 번** 반영되는 문제가 발생

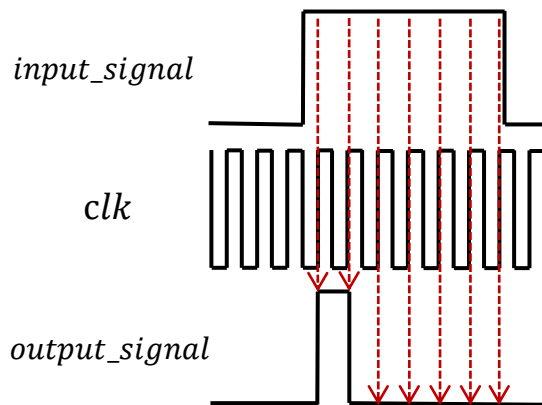
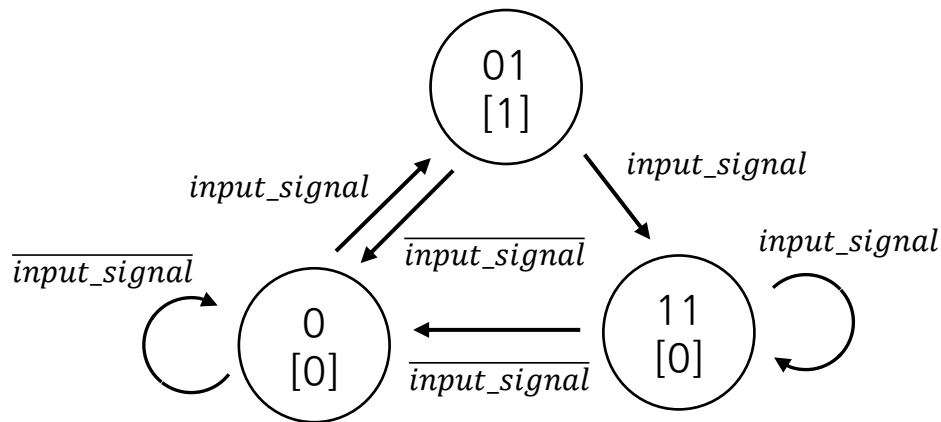
- Pulse Generator

- 긴 시간 동안의 푸시 버튼 입력을
한 cycle 동안의 입력으로 변환



Pulse Generator FSM

- 다음과 같은 FSM의 설계로 문제를 해결



```

module PulseGenerator (
    input clk,
    input input_signal,
    input reset,
    output output_signal
);
    parameter s_0 = 2'd0;
    parameter s_11 = 2'd1;
    parameter s_01 = 2'd2;

    reg[1:0] state; reg[1:0] next_state;

    assign output_signal = (state == s_01) ? 1'b1 : 1'b0;

    always @(*) begin
        case (state)
            s_0:
                next_state = (input_signal == 1'b1) ? s_01 : s_0;
            s_11:
                next_state = (input_signal == 1'b1) ? s_11 : s_0;
            s_01:
                next_state = (input_signal == 1'b1) ? s_11 : s_0;
            default:
                next_state = s_0;
        endcase
    end

    always @(posedge clk) begin
        state <= (reset == 1'b1) ? s_0 : next_state;
    end
endmodule
    
```


실습 – 4-bit binary up counter 구현

- 목표
 - 4-bit binary up counter를 2MHz clock에서 동작하도록 구현
- 실험 내용
 - ClockModulator, PulseGenerator 구현
 - 구현한 모듈들을 이용해 4-bit binary up counter 구현
 - SNU Logic Design 보드에 올려 동작 검증
- 제출 사항: 없음

실습 – 4-bit binary up counter 구조

- 푸시 버튼을 한 번 누를 때 마다 카운터가 1 씩 증가하도록 구현

