

1. 기본 구현사항

<모듈 구조>

Clock, Calendar, Alarm, Stopwatch, Display Control 모듈이 각 기능을 담당한다.

Mode Control : 네 모듈의 상위 구조로, isUsing 변수를 활용해 각 모듈이 사용중일 때 해당 모듈의 아웃풋을 최종 아웃풋에 할당한다. (Overlapping 문제로 clock + alarm + display control , calendar + stopwatch 로 분리하였다.)

<구조 디자인>

모듈에서 변수의 값이 변화하는 시점이 많다. 중요한 사실은 변수의 값이 복수의 always 문 내에서 변화할 수 없다는 것이다. 따라서 한 모듈이 커다란 always @ (posedge clk) 문 내에서 동작하도록 디자인하였다. Input button 을 인식해 변수 값을 증감시켜야 할때도 always @ (posedge input button) 문을 지양하고, always @ (posedge clk) 문 내에서 채터링 문제를 해결하여 input button을 인식하였다.

<채터링 문제의 해결 - 추가구현>

Input button 에 의해 변수의 값이 특정값으로 지정되는 경우가 있는데, 해당 경우는 채터링이 일어나도 크게 문제가 되지 않는다. 하지만 Up / down button 등 특정 경우에는 input이 여러번 catch되면 잘못된 결과가 발생한다. 채터링을 해결하기 위해 디바운서 모듈 사용이 아닌 '개별 모듈 내 input sampling 주기 조절' 을 택하였다.

예시) set button 의 채터링 문제 해결

```
// always @ ( posedge clk ) 내부에서
if( setcounter < buttonfreq )
    begin
        setcounter = setcounter + 1;
    end
else
    begin
        setcounter = 0;
        if(set==1)
            begin
                // 이 부분에 set이 눌러졌을 때의 동작 기술
            end
        else begin end
    end
end
```

매우 빠른 Clk 주기마다 1씩 증가하는 setcounter가 특정값 (buttonfreq) 에 도달할 때만 set button input을 sampling 할 수 있게 함으로써 모듈 내에서 자체적으로 button 의 input sampling 주기를 조절할 수 있게 디자인하였다. clk 주기보다 훨씬 긴 주기로 sampling 하기 때문에 input이 진동하여도 한번의 입력으로 인식할 수 있다.

2. 모듈 설명

<clock>

```
reg is12; // 12시간 표시법 or 24시간 표시법 결정
reg [1:0] setStatus;
// setStatus는 세팅 상태를 나타내는 변수 이다.

always @ (posedge clkBoard)
begin
    // 알람 모듈과 교류하기 위한 로직

        if((alarmon==1)&&(up==1))
        begin
            alarmoff=1;
        end
        else
        begin
            alarmoff=0;
        end

case(setStatus)

// setStatus 값에 따라 표시모드, 시간 변경모드, 분
// 변경 모드 등으로 바뀐다.

    0:    // clock display mode
    begin
        ...

        if( counter1 < secfreq )
        begin
            counter1 = counter1 + 1;
        end
        else
        begin
            sec = sec+1;
// counter가 freq와 같아지면 1초 증가
            counter1 = 0;
        end

    end

    ...

// clock 로직

    if (sec > 59)
    begin
        sec = 0;
        m = m + 1;
    ... 이후 변수 값에 따라 display 할당
```

<calendar>

```
always @ (posedge clkBoard)
begin

case(setStatus)
    0:    // calendar display mode
        begin

            if( setcounter < buttonfreq )
            begin
                setcounter = setcounter + 1;
            ...

// 채터링 문제를 해결하여 set 버튼 입력에 따라 설정
// 모드 변경

// calendar 로직으로 경계점 처리

            if( y > 99 )
            begin
                y = 0;
            end
            else begin end

            if( y < 0 )
            begin
                y = 99;
            end
            else begin end

            if( m==1 || m==3 || m==5 ||
m==7 || m==8 || m==10 ||
m==12 ) //31days
            begin
                if(day>31)
                begin
                    day=1;
                end
                else begin end
                if(day<1)
                begin
                    day=31;
                end
                else begin end
            end
            else begin end

            ...

이후 변수 값에 따라 display 할당
```

<alarm>

```
always @ (posedge clkBoard)
begin

// 현재 알람시간 일치 여부를 clock 모듈에 보내기 위
한 로직

    if( (hasalarm==1)&&(m==am)&&(h==ah) )
    begin
        alarmon=1;
    end
    else begin end

    if( alarmoff==1 )
    begin
        alarmon=0;
    end
    else begin end

    alarmonout = alarmon;

    ...

case(setStatus)
0:
    ...

// set status 값에 따라 모드 변경
// 이후 변수 값에 따른 display 할당
```

<stopwatch>

```
always @ (posedge clkBoard)
begin

    ...

    if( counter1 < 500000 )
    begin
        counter1 = counter1 + 1;
    end
    else
    begin
        counter1 = 0;
        msec = msec+1;
    end

    ...

// 100분의 1초마다 msec가 1씩 증가하고 이후에 나오
는 시계 로직에 의해 sec와 min도 증가하게 된다.
```

<display control>

```
always @ (posedge clkBoard)

    ...

    if(isUsing == 4)
    begin
        is12out = ~is12in;
    end
    else

// 12시간 모드, 24시간 모드 변경
```

<mode control>

```
Clock c1(...);
displayControl dc1(...);
```

// 모듈 인스턴스

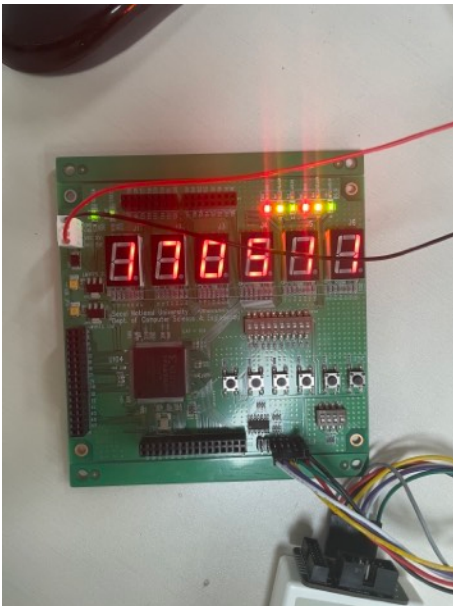
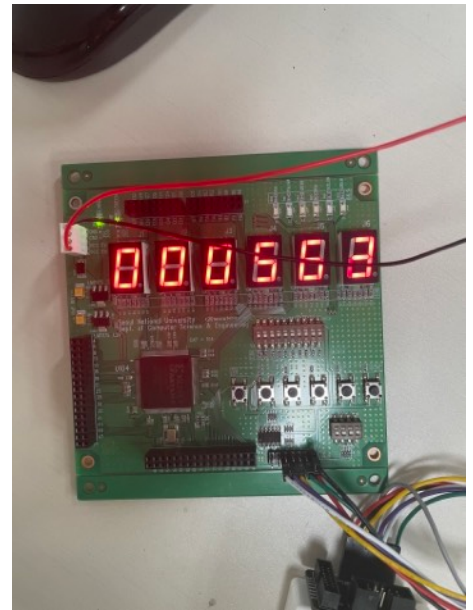
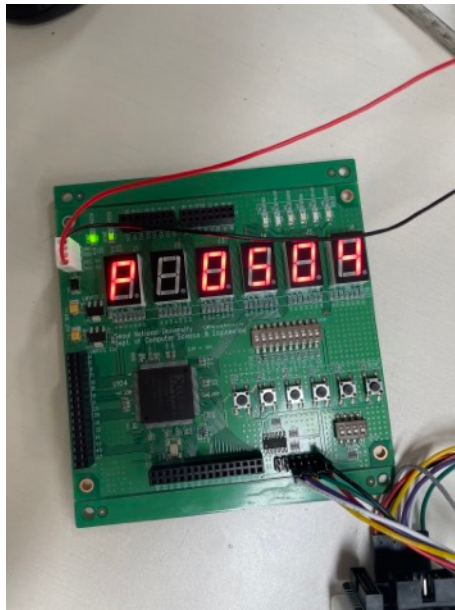
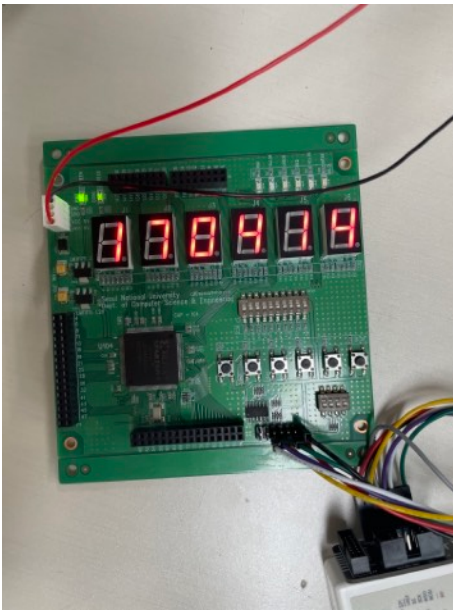
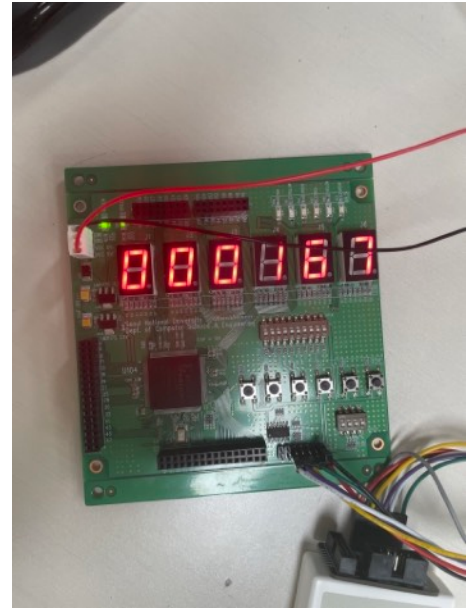
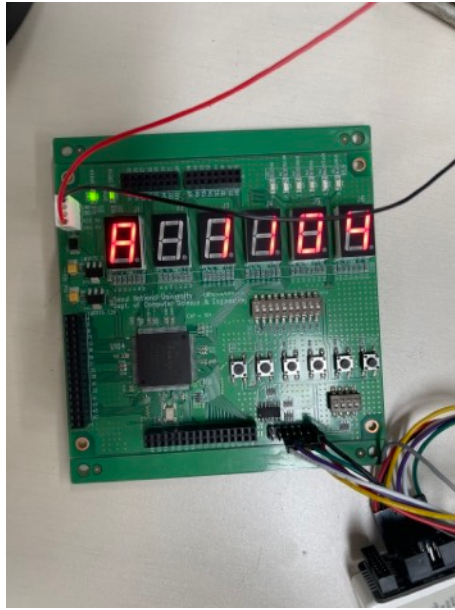
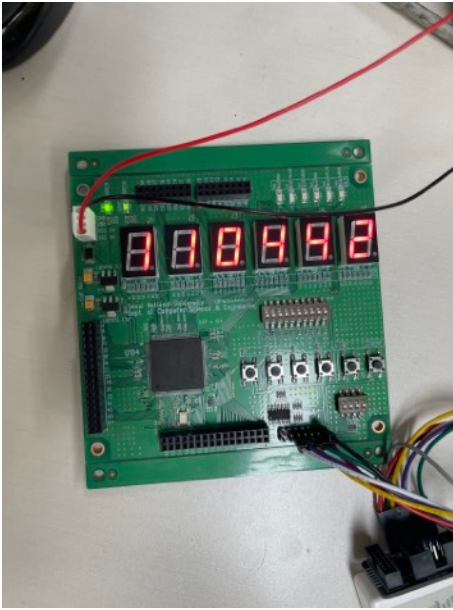
```
always @ (posedge clkBoard)
begin

    if( modecounter < buttonfreq )
    begin
        modecounter = modecounter + 1;
    end
    else
    begin
        modecounter = 0;
        if(mode==1)
        begin
            if(isUsing < 4)
                isUsing = isUsing + 2;
            else
                isUsing = 0;
        end
    end
end

case(isUsing)
0:
begin
    d1 = clockd1;
    d2 = clockd2;
    d3 = clockd3;
    d4 = clockd4;
    d5 = clockd5;
    d6 = clockd6;

    ...
```

3. 실행 사진



시계기능
알람기능
12시간 / 24시간 변환
달력기능
요일 표시 기능
스톱워치 기능
점멸 기능

등의 정상 작동을 확인하였다.