# Final Project

2017-12-12
2016-17101 김종범

## 1. 구현 사항

"2017년 논리설계 Final Project: 디지털 시계 만들기" 스펙에 있는 모든 기본 구현 사항은 구현 완료하였다.

여기에 2개의 추가구현을 하였다. 첫 번째는 SW5를 이용하여 절전모드를 제작하였다. SW5를 누르면 절전모드가 토글되며, 기존 밝기의 1/2배의 밝기로 7-segment를 표시해준다. 두 번째는 모든 입력 버튼을 특정 시간 이상 꾹 누르고 있으면 특정 시간 단위로 계속 눌리는 것을 제작하였다. (이하, holding 이라고 부르겠다.)

위의 추가구현 이외에서 Programmable Watch 또한 제작 완료하였다. 위 모든 사항은 2017년 12월 12일 기준으로 모두 검사를 받은 상태이다.

## 2. 모듈 설명

ClockModulatorTo1Hz
ClockModulatorTo1KHz
ClockModulatorTo5Hz
ClockModulatorTo100Hz

위 4개의 모듈은 기존의 50MHz의 Clock을 각각 1Hz, 1KHz, 5Hz, 100Hz로 변환해주는 모듈이다. 1Hz는 기본적으로 Watch의 시간을 흐르게 할 때 쓰이고, 1KHz는 SNUBoard의 입력을 받을 때 Debouncer에 넣어준다. 5Hz는 깜박이는 속도를 결정할 때 쓰이메, 100Hz는 StopWatch 구현을 할 때 쓰인다.

*추가구현 : SW5가 눌리면 7-Segment의 밝기가 1/2가 된다.

```
assign display = (freeToggle & clk_50MHz? 42'b0 : brightDisplay);
always @ (posedge clk_50MHz or posedge reset) begin
        if(reset) begin
                freeToggle = 0;
        end else begin
                if(p_free) freeToggle = ~freeToggle;
        end
end
```

위는 밝기가 토클되는 추가구현을 만드는 코드이다, p_free 신호가 들어 올 때마다 freeToggle 신호가 NOT 되며, 최종적으로 출력되는 display는 freeToggle와 clk_50MHz가 모두 1인 경우에는 0을 반환하여 밝기를 낮추게 된다.

digitalWatch

digitalWatch의 총괄을 담당하는 모듈이다. 기본적으로 watchDisply, alarmDisplay, stopDisplay의 정보를 모으거나, 신호가 들어오면 적절한 모듈이 신호를 전해주는 역할을 한다. 위의 세 모듈에서 모든 정보를 이용하여 7-segment 6개에 display를 showHMStime 모듈을 이용하여 표시한다.

watchDisplay

세 모드 중에 시계 모드를 총괄하는 모듈이다. HMSCounter가 1초씩 증가하며 시간을 알려주고, 만약 Setting 모드게 들어가는 경우네는 settingDisplay와 HMSCounter의 로드기능을 이용하여 설정 및 세팅완료까지 해주는 모듈이다.

HMSCounter

기본적으로 1초마다 HMS(시간, 분, 초)를 알려주는 모듈이다. 특정상황인 경우 load=1로 만들면 특정 시간, 분, 초로 load할 수도 잇다.

alarmDisplay

세 모드 중에 알람 설정 모드를 총괄하는 모듈이다. memoryTime 모듈을 이용하여 기본적으로 시간을 기억하며, settingDisplay 모듈을 이용하여 설정 모드의 디스플레이를 얻을 수 있다.

memoryTime

HMSCounter에서 Counter를 제외한 모듈이라고 생각하면 된다. 특정 값을 기억하고 싶을 때 load=1 신호를 주며 값을 넣어주면 값을 기억한다.

settingDisplay

시계 모드와 알람 설정 모드에서 값을 설정하는 경우에 인풋을 받고 특정 값을 잘 설정하게 해주는 것을 담당하는 모듈이다.

stopDisplay

세 모드 중에 스톱워치 모드를 총괄하는 모듈이다. MSMSCounter 모듈과 입력을 적절히 판단하여 스톱워치 모드의 보드상태를 반환해준다.

MSMSCounter

HMSCounter와 기능은 같으나, 100Hz(0.01초) 단위로 증가하는 Counter이다. 각각의 값을 MSMS(분, 초, millisecond)로 반환한다.

showHMStime

digitalWatch 모듈에서 모두 모아진 출력해야 할 7-segment 값 정보와, 어디를 깜빡여야하는지 여부가 입력으로 들어오면 그에 맞게 7-segment 값을 반환한다. 값 계산을 위해 get00_99to가 쓰이며, 6개 각각의 표시할 7-segment 값을 알기위해 showFourUnsigned가 사용된다

get00_99to

0부터 99사이의 값이 들어오면 십의 자리와 일의 자리로 나누어서 값을 반환한다. 특정값(111, 110)

같은 경우에는 값을 표시하지 않거나, -를 표시한다. 이는 leading Zero와 알람이 설정되지 않았을 때 쓰인다.

showFourUnsigned

0부터 9까지의 수가 들어오면 7-Segment로 반환하고, 10이나 11이 들어오면 표시하지 않거나, -를 표시한다. 이는 leading Zero와 알람이 설정되지 않았을 때 쓰인다.

handableWatch

handableWatch는 SNUBoard로 입력을 줄 때 사용하는 Module이다. (즉, Programmable Watch가 아닌 경우에 사용한다.) 아래의 toPulseInput을 이용하여 pulse 신호로 입력을 변환한 뒤, digitalWatch 모듈에 넣어준다.

toPulseInput

스위치를 이용한 입력을 Debouncer과 PulseGenerator 모듈을 이용하여, 채터링도 방지하며, 푸시 버튼 입력이 여러 번 반영되지 않게 변환하여 반환해주는 모듈이다.

Debouncer

스위치의 상태가 변하는 순간 발생하는 채터링(chattering)을 방지하는 모듈이다.

PulseGenerator

높은 주파수의 clock를 이용하기 때문에, 푸시 버튼 입력이 여러 번 반영되지 않게 기존의 입력은 Pulse로 변환해주는 역할을 한다.

*추가구현 : 특정 시간 이상 누르면 holding을 지원하게 수정하였다.

```
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
	case (cstate)
		cs_0: next_cstate = (clk_5Hz == 1'b1) ? cs_01 : cs_0;
		cs_11:next_cstate = (clk_5Hz == 1'b1) ? cs_11 : cs_0;
		cs_01:next_cstate = (clk_5Hz == 1'b1) ? cs_11 : cs_0;
		default:next_cstate = cs_0;
	endcase
	case (counter)
		cnt_0: nextCounter = (state == s_11) ? (cstate == cs_01 ? cnt_1 : cnt_0) : cnt_0;
		cnt_1: nextCounter = (state == s_11) ? (cstate == cs_01 ? cnt_2 : cnt_1) : cnt_0;
		cnt_2: nextCounter = (state == s_11) ? (cstate == cs_01 ? cnt_3 : cnt_2) : cnt_0;
		cnt_3: nextCounter = (state == s_11) ? (cstate == cs_01 ? cnt_4 : cnt_3) : cnt_0;
		cnt_4: nextCounter = (state == s_11) ? cnt_5 : cnt_0;
		cnt_5: nextCounter = (state == s_11) ? (cstate == cs_01 ? cnt_4 : cnt_5) : cnt_0;
		default: nextCounter = s_0;
	endcase
end
```

위 코드는 추가구현을 도입하였을 때 PulseGenerator의 상태계산하는 부분의 코드이다. 기본적으

로 clk_5Hz이 4번이상 posedge가 된 경우에 holding이 시작되고, clk_5Hz의 01인 상황에서만 상태가 변경되는 것을 이용하여 일정 시간 동안 입력이 되는 것을 구현할 수 있었다.

*추가구현 : 아래 모듈들은 ProgrammableWatch를 제작하기 위해 만들어진 모듈들이다.

ProgrammableWatch

ProgrammableWatch는 다른 Board의 InstructionMemory에 들어 있는 순서대로 실행하는 모듈이다. 모듈 안에 HandShaker와 Decoder를 이용하여 통신을 한 뒤, Instruction 번호를 Decode하여 pulse 신호의 입력으로 변환하여 digitalWatch 모듈에 넘겨준다.

HandShaker

req와 ack 와이어를 통해서, 4-Cycle handsahking 통신을 진행하는 모듈이다.

Decoder

통신을 통해 받아온 Instruction 번호를 decode하여 입력 Pulse 신호를 전달한다.

3. 추가 구현 설명 및 스펙

추가구현1 : 밝기가 1/2이 되는 것은 Clock 신호가 1일 때에만 display를 내보내 주게 해서 만들었다. 토글형식으로 진행되며, reset 버튼을 누를시 밝은 상태로 초기화 된다.

추가구현2 : Holding을 지원한다. 기존에는 버튼을 꾹 누르고 있으면 단 한번만 입력신호가 가는 형태였는데, 특정 시간 이상 버튼을 꾹 누르게 되면 특정 시간마다 버튼이 한번 씩 눌리는 효과를 얻을 수 있다. 이는 시간 Setting이나 알람 Setting시 유용하게 사용될 수 있는 기능이라고 할 수 있다.

추가구현3 : Programmable Watch를 제작하였다. HandShaker 측에서 값을 전달 받으면 4초 정도의 Delay를 발생시켜서 완성하였다. Decoder는 단지 받아온 Instruction 번호를 적절히 decode하여 입력 신호를 만드는 역할을 한다.