



서울시립대학교
UNIVERSITY OF SEOUL

전자전기컴퓨터설계실험 II

디지털 시계 설계

김인철 교수님

2019440121 조성원

2022. 12. 13 (화요일)

Table of Contents

1. 개요

- 모듈 흐름도
- 기능별 설명

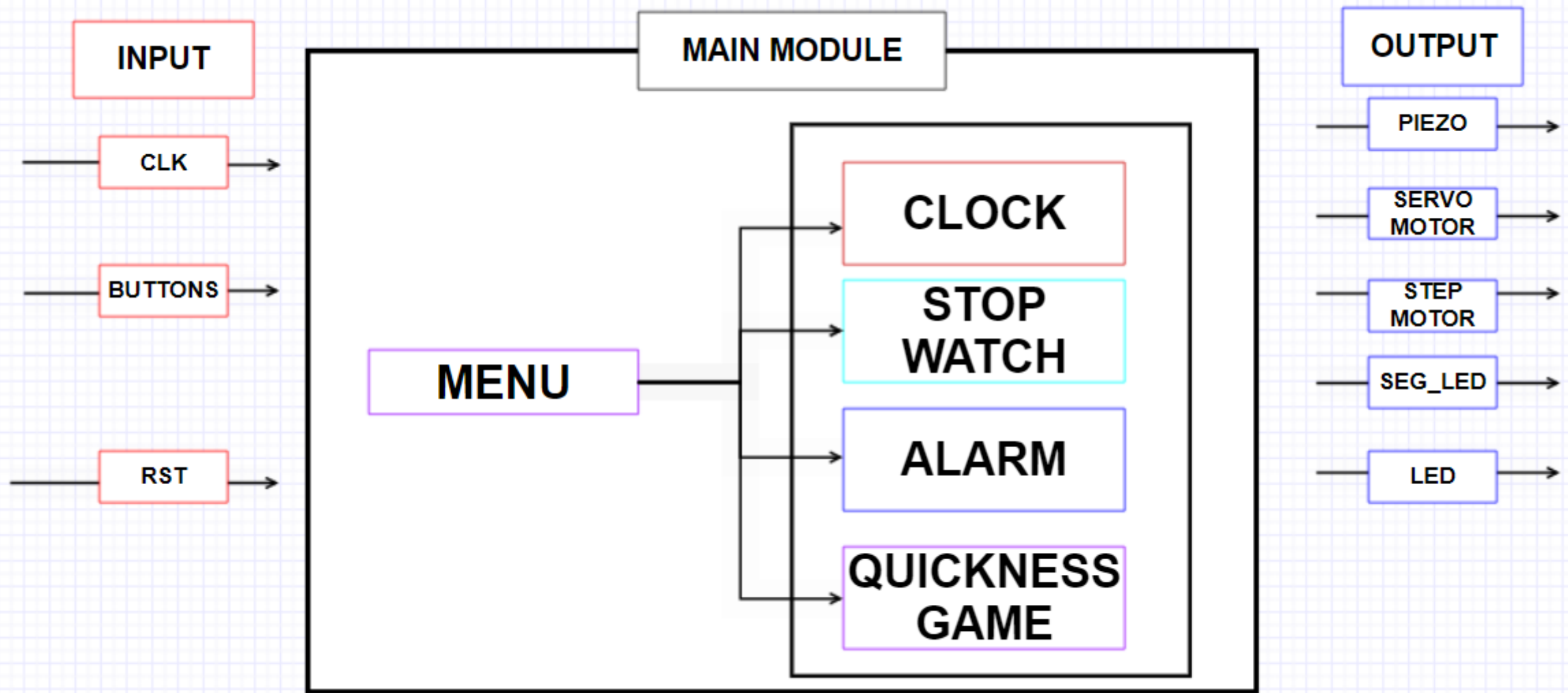
2. 기능 분석 및 구동 영상

- 시간 표시 및 조정 기능
- STOP WATCH기능
- 알람시계 기능
- 시간 측정을 이용한 게임 기능

3. 결론 및 구현 코드

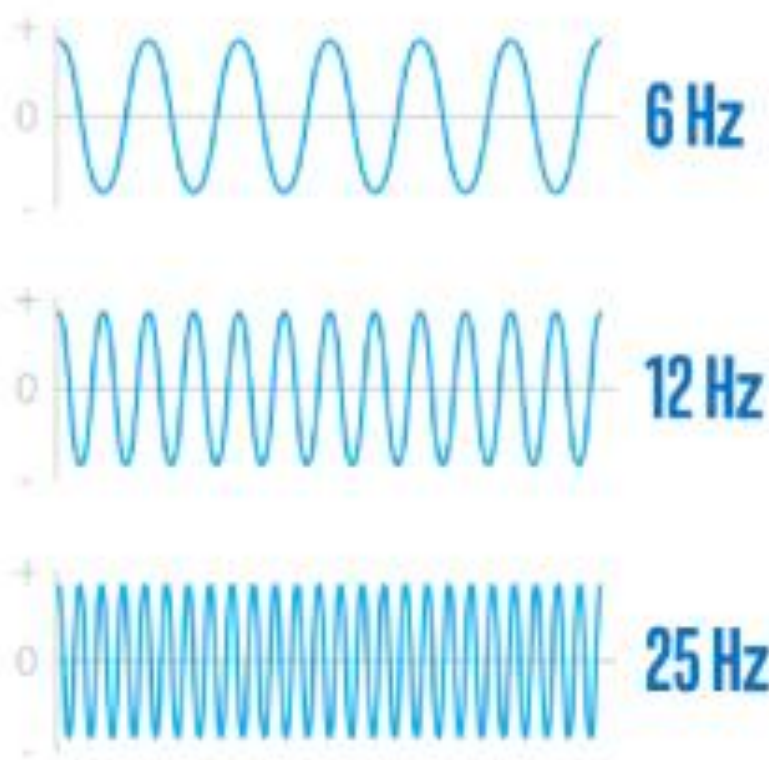
- Blog, Github

1. 개요 – 모듈 흐름도



(그림 1. 전체 시스템 BLOCK DIAGRAM)

1. 개요 – 기본 설명



(그림 2. 클럭 속도 설정의 중요성)

| 아래의 모듈 watch는 모듈 seg_decoder와 함께 1KHz 클럭을 받아들이고 분과 초를 표현하는 verilog HDL 코드의 예이다. 이코드를 모의실험하기 위한 간단한 testbench를

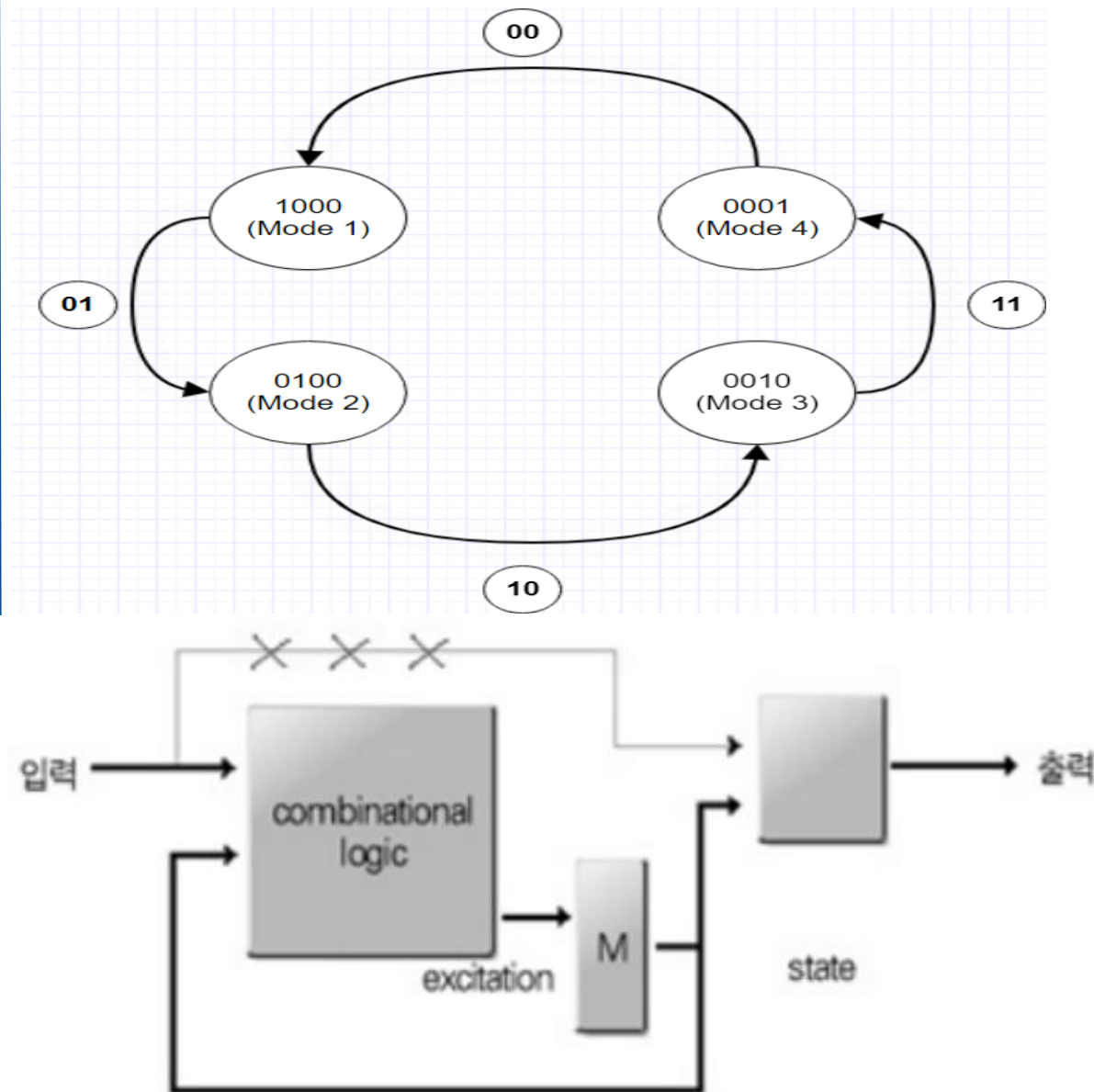
리고, 신호선에는 매 20msec마다 하나의 펄스를 보낸다. 펄스의 폭은 모터가 중간 지점, 즉 90도부터 어느 만큼 움직이는지를 결정한다. 예를 들어 만약 펄스 폭이 1.5msec이면 모터는 90도 위치로 움직이고, 펄스 폭이 0.6msec이면 0도 위치, 펄스 폭이 2.4msec이면 180도 위치로 움직인다. 즉, 펄스폭이 1msec 증가할 때마다 10도씩 증가한다.

(그림 3. 프로젝트 교안의 내용)

음계	2옥타브		3옥타브		4옥타브	
	f (Hz)	$10^6 / f/2$	f (Hz)	$10^6 / f/2$	f (Hz)	$10^6 / f/2$
도	130.8128	3822	261.6256	1911	523.2511	956
레	146.8324	3405	293.6648	1703	587.3295	851
미	164.8138	3034	329.6276	1517	659.2551	758
파	174.6141	2863	349.2282	1432	698.4565	716
솔	195.9977	2551	391.9954	1276	783.9909	638
라	220.0000	2272	440.0000	1136	880.0000	568
시	246.9417	2025	493.8833	1012	987.7666	506

(그림 4. 프로젝트 교안의 내용) (1MHz 기준)

1. 개요 – 기본 설명



(그림 5. Moore machine (finite state machine))

//메뉴 조정

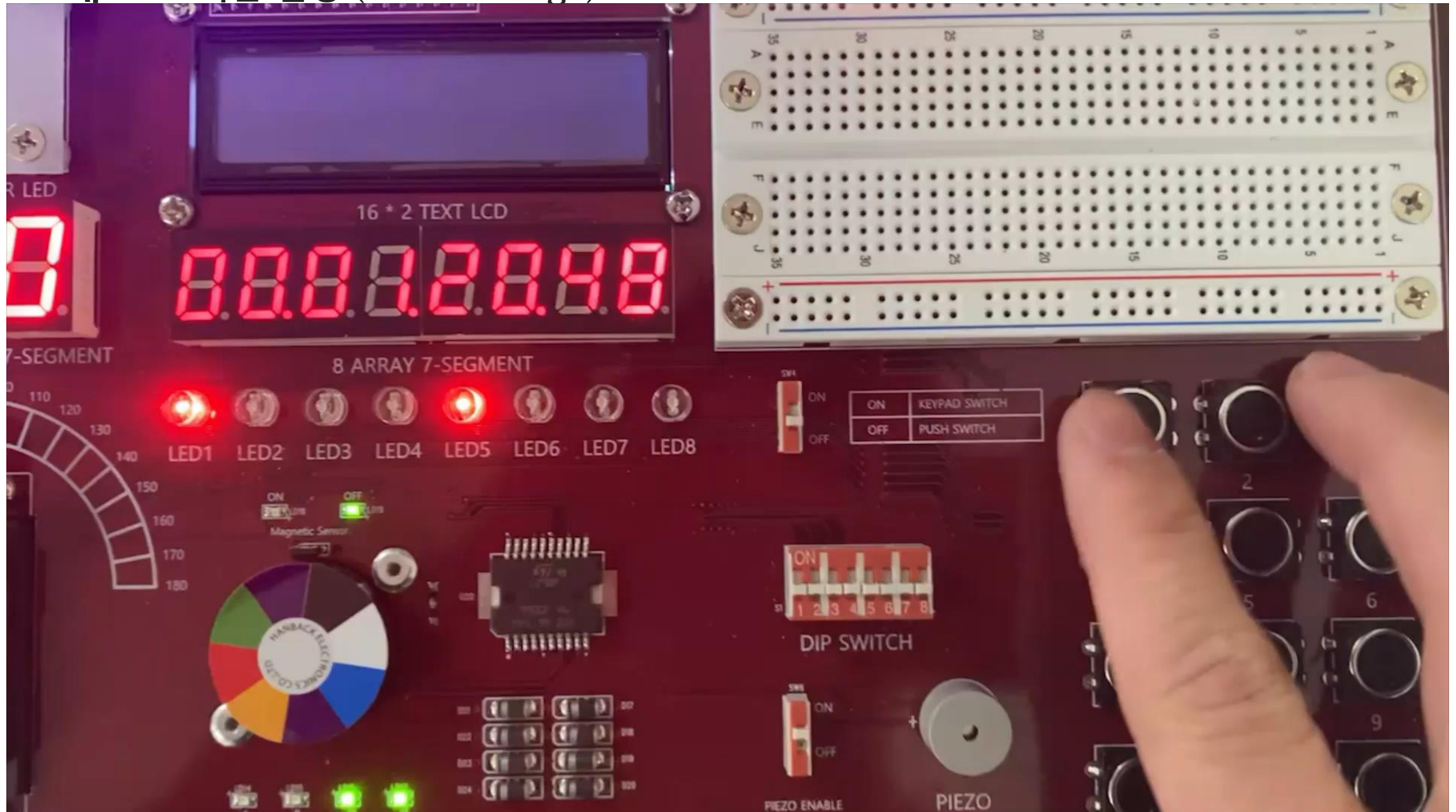
```
always@(posedge clk) begin
    if(menu_w ==1) begin
        state <= state +1;
    end
end
```

//led 조정

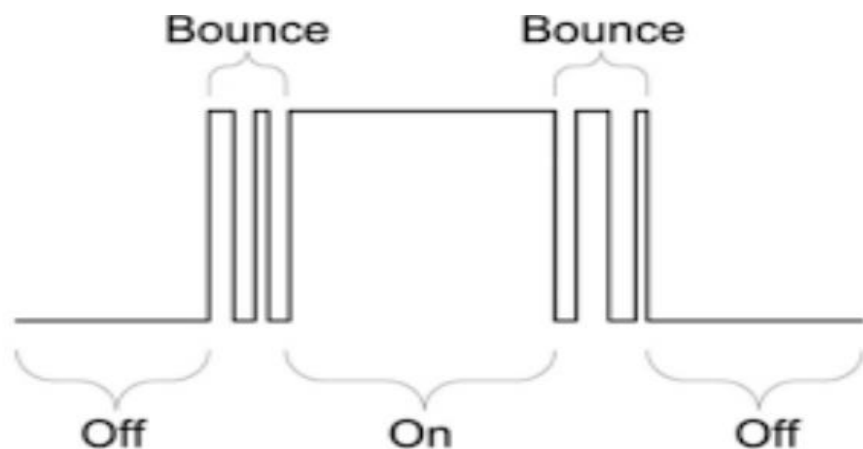
```
always@(posedge clk) begin
    case (state)
        2'b00 : led= 4'b1000; //정상 시계 + 조정가능
        2'b01 : led = 4'b0100; // 스톱와치
        2'b10 : led = 4'b0010; // 타이머
        2'b11 : led = 4'b0001; //리듬게임
    endcase
end
```

(그림 6. overflow와 one-hot vector로 메뉴표현)

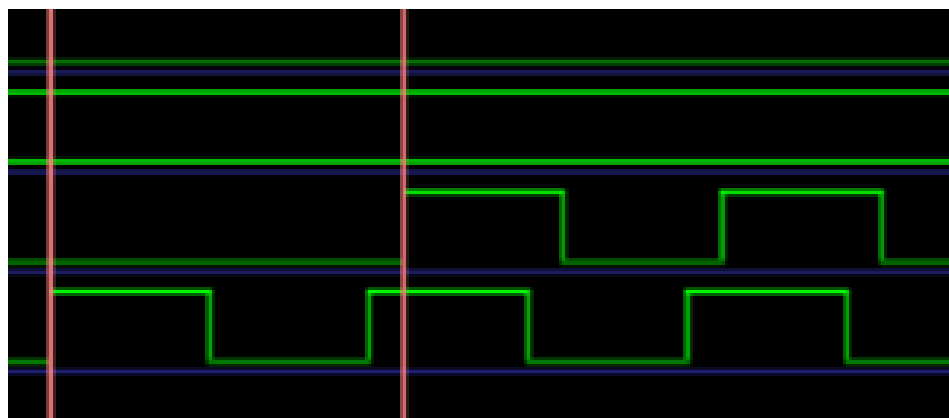
1. 개요 – 기본 설명 (menu change)



1. 개요 – 기본 설명



(그림 7. 채터링 효과)



(그림 8. 너무 빠른 클럭 속도)

```
module switch(clk, rst, in, out);  
    input clk, rst, in;  
    output reg out;  
    reg q;  
    always@(posedge clk) begin  
        if(rst) begin  
            q = 0;  
            out = 0;  
        end  
        else begin  
            out = in & (q^in);  
            q = in;  
        end  
    end  
endmodule
```

(그림 9. D F/F을 이용한 채터링 방지)

1. 개요 – MODE 1 설명 (시계)

- 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)
- 시간 조정 (hour, min, second)
- 낮/밤 나타내기 (D: 낮, N: 밤)
- 스텝모터와 초침의 동기화
- 시각(hour)을 통한 해, 달 위치 나타내기
- 초침에 따른 led 발광

1. 개요 – MODE 2 설명 (STOP WATCH)

- 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)
- 버튼을 이용한 GO/STOP 구현
- 초침에 따른 스텝모터 구현
- 초침에 따른 led 발광

1. 개요 – MODE 3 설명 (Alarm)

- 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

(단위 : Hz)

- 버튼을 이용한 시간 조정 구현

- 하울의 움직이는 성 알람 구현

- 음악에 따른 스텝모터 회전 구현

- 초침 및 음악에 따른 led 발광

옥타브 음계 \	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

(그림 10. 옥타브에 따른 음계 표현)

1. 개요 – MODE 4 설명 (Game)

- 게임 시작/종료 버튼 구현
- 난수 생성 및 난수에 따른 게임모드 설정
- 순발력 시간 측정
- 게임 상태에 따른 소리 구현
- LCD 구현

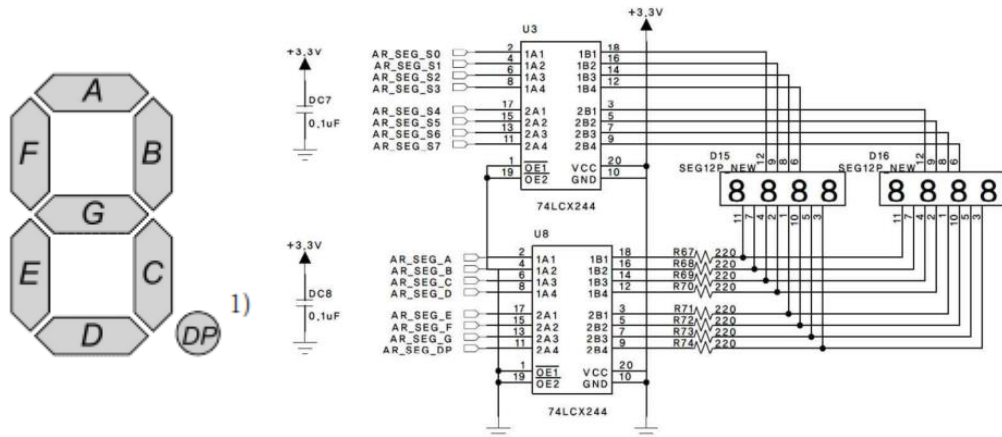
2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second : 100milli-s : 10milli-s)

A. LCD조정하기

```
seg_decoder1 u1(sec[0], seg_s_one);
seg_decoder u2(sec[1], seg_s_ten);
seg_decoder1 u3(min[0], seg_m_one);
seg_decoder u4(min[1], seg_m_ten);
seg_decoder1 u5(hour[0], seg_h_one);
seg_decoder u6(hour[1], seg_h_ten);
seg_decoder u7(m_min, seg_m_m);
seg_decoder u8(m_sec, seg_m_mm);
```

(그림 11. 모듈 불러오기)



```
module seg_decoder1(bcd, seg_data);
    input [3:0] bcd;
    output [7:0] seg_data;
    reg [7:0] seg_data;
    always @(bcd) begin
        case(bcd)
            4'H0 : seg_data=8'b1111_1101; // 'HFC
            4'H1 : seg_data=8'b0110_0001; // 'H60
            4'H2 : seg_data=8'b1101_1011; // 'HDA
            4'H3 : seg_data=8'b1111_0011; // 'HF2
            4'H4 : seg_data=8'b0110_0111; // 'H66
            4'H5 : seg_data=8'b1011_0111; // 'HB6
            4'H6 : seg_data=8'b1011_1111; // 'HBE
            4'H7 : seg_data=8'b1110_0001; // 'HE0
            4'H8 : seg_data=8'b1111_1111; // 'HFE
            4'H9 : seg_data=8'b1111_0111; // 'HF6
            4'Ha : seg_data=8'b1110_1110; // 'HEE
            4'Hb : seg_data=8'b0011_1110; // 'H3E
            4'Hc : seg_data=8'b1001_1100; // 'H9C
            4'Hd : seg_data=8'b0111_1010; // 'H7A
            4'He : seg_data=8'b1001_1110; // 'H9E
            4'Hf : seg_data=8'b1000_1110; // 'H8E
        endcase
    end
endmodule
```

(그림 12. .표시가 있는 lcd 표현)

```
module seg_decoder(bcd, seg_data);
    input [3:0] bcd;
    output [7:0] seg_data;
    reg [7:0] seg_data;
    always @(bcd) begin
        case(bcd)
            4'H0 : seg_data=8'b1111_1100; // 'HFC
            4'H1 : seg_data=8'b0110_0000; // 'H60
            4'H2 : seg_data=8'b1101_1010; // 'HDA
            4'H3 : seg_data=8'b1111_0010; // 'HF2
            4'H4 : seg_data=8'b0110_0110; // 'H66
            4'H5 : seg_data=8'b1011_0110; // 'HB6
            4'H6 : seg_data=8'b1011_1110; // 'HBE
            4'H7 : seg_data=8'b1110_0000; // 'HE0
            4'H8 : seg_data=8'b1111_1110; // 'HFE
            4'H9 : seg_data=8'b1111_0110; // 'HF6
            4'Ha : seg_data=8'b0000_0000; // 'HEE
            4'Hb : seg_data=8'b0011_1110; // 'H3E
            4'Hc : seg_data=8'b1001_1100; // 'H9C
            4'Hd : seg_data=8'b0111_1010; // 'H7A
            4'He : seg_data=8'b1001_1110; // 'H9E
            4'Hf : seg_data=8'b1000_1110; // 'H8E
        endcase
    end
endmodule
```

(그림 13. .표시가 없는 lcd 표현)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

B) 버튼을 이용한 시간 조정 (시 나타내기)

```
if(state==00) begin
    if(h_up==1) begin
        if(hour[0]==1 & hour[1] ==1) begin
            hour[0] <= 0;
            hour[1] <=0;
        end
        else begin
            if(hour[0] ==9)begin
                hour[1] <=hour[1] +1;
                hour[0] <= 0;
            end
            else begin
                hour[0] <= hour[0] +1;
            end
        end
    end
end
```

(그림 14. 버튼을 이용해 시 증가)

```
else if(h_down)begin
    if(hour[1]==0)begin
        if(hour[0]==0)begin
            hour[0] <=0;
            hour[1] <=0;
        end
        else hour[0] <= hour[0] -1;
    end
    else begin
        hour[1] <= hour[1] -1;
        hour[0] <= 9;
    end
end
```

(그림 15. 버튼을 이용해 시 감소)

```
else begin
    if(cnt==99999& min[1]==5& min[0]==9 & sec[1]==5 & sec[0]==9)begin
        if(hour[0]==1 & hour[1] ==1) begin
            hour[0] <= 0;
            hour[1] <=0;
        end
        else begin
            if(hour[0] ==9)begin
                hour[1] <=hour[1] +1;
                hour[0] <= 0;
            end
            else begin
                hour[0] <= hour[0] +1;
            end
        end
    end
end
```

(그림 16. 기본 상태의 시 계산)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

B) 버튼을 이용한 시간 조정 (분 나타내기)

```
if(state==00) begin
    if(m_up==1) begin
        if(min[1]==5 && min[0]==9)begin
            min[1] <=0;
            min[0] <=0;
        end
        else begin
            if(min[0]==9) begin
                min[0] <=0;
                min[1] <= min[1] +1;
            end
            else min[0] <= min[0] +1;
        end
    end
end
```

(그림 17. 버튼을 이용한 분 증가)

```
else if(m_down==1)begin
    if(hour[0]==0&&hour[1]==0)begin
        if(min[1]==0)begin
            if(min[0]==0)begin
                min[0]<=0;
                min[1] <=0;
            end
            else min[0] <= min[0] -1;
        end
        else begin
            min[1]<=min[1]-1;
            min[0] <= 9;
        end
        else min[0] <= min[0] -1;
    end
end
end
end
```

(그림 18. 버튼을 이용한 분 감소)

```
else begin
    if(cnt==99999 && sec[1]==5 && sec[0]==9) begin
        if(min[1]==5 && min[0]==9)begin
            min[1] <=0;
            min[0] <=0;
        end
        else begin
            if(min[0]==9) begin
                min[0] <=0;
                min[1] <= min[1] +1;
            end
            else min[0] <= min[0] +1;
        end
    end
end
```

(그림 19. 기본 상태의 분 계산)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

B) 버튼을 이용한 시간 조정 (초 나타내기)

```
if(state==00)begin
    if(s_up==1)begin
        if(sec[1]==5 && sec[0]==9)begin
            sec[1] <=0;
            sec[0] <=0;
        end
        else begin
            if(sec[0] ==9) begin
                sec[0] <=0;
                sec[1] <= sec[1] +1;
            end
            else sec[0] <= sec[0] +1;
        end
    end
end
```

(그림 20. 버튼을 이용한 초 증가)

```
else if (s_down==1)begin
    if(sec[1]==0) begin
        if(sec[0]==0&min[0]==0 &&min[1]==0&&hour[0]==0&hour[1]==0)begin
            sec[1] <=5;
            sec[0] <=9;
        end
        else begin
            sec[0] <= sec[0]-1;
        end
    end
    else begin
        if(sec[0] ==0)begin
            sec[1] <= sec[1] -1;
            sec[0] <= 9;
        end
        else sec[0] <= sec[0] -1;
    end
end
```

(그림 21. 버튼을 이용한 초 감소)

```
else begin
    if(cnt==99999) begin
        if(sec[1]==5 && sec[0]==9)begin
            sec[1] <=0;
            sec[0] <=0;
        end
        else begin
            if(sec[0] ==9) begin
                sec[0] <=0;
                sec[1] <= sec[1] +1;
            end
            else sec[0] <= sec[0] +1;
        end
    end
end
```

(그림 22. 기본 상태의 초 계산)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second : 100milli-s : 10milli-s)

B) 버튼을 이용한 시간 조정 (100milli-second 나타내기)

```
else begin
    if(state==2'b00) begin
        if(cnt%10000 ==0) begin
            if(m_min ==9) begin
                m_min <= 0;
            end
        else begin
            m_min <= m_min +1;
        end
    end
end

else if(state==2'b01) begin
```

(그림 23. 기본 상태의 micro_second 계산)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second : 100milli-s : 10milli-s)

B) 버튼을 이용한 시간 조정 (10milli-second 나타내기)

```
//m_sec
always@(posedge clk)begin
    if(rs) m_sec = 0;
    else begin
        if(state==2'b00) begin
            if((cnt%1000)==0) begin
                if(m_sec ==9) begin
                    m_sec <= 0;
                end
            else begin
                m_sec <= m_sec +1;
            end
        end
    end
end
```

(그림 23. 기본 상태의 micro_second 계산)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

C) 낮/밤 나타내기

// day night flag 설정

```
always@(posedge clk) begin
```

```
    if(rst) flag= 0;
```

```
    else begin
```

```
        if(state==2'b00)begin
```

```
            if(cnt==99999& min[1]==5& min[0]==9 & sec[1]==5 & sec[0]==9 &
```

```
            hour[0]==1 & hour[1]==1) begin
```

```
                flag = ~flag; // 0 : day 1 :night
```

```
            end
```

```
        else begin
```

```
            if(min[1]==5 & min[0]==9 & hour[1]==1&hour[0]==1&m_up==1)begin
```

```
                flag =~flag;
```

```
            end
```

```
        else if(hour[1]==1 & hour[0] ==1 & h_up==1)begin
```

```
            flag= ~flag;
```

```
        end
```

```
    end
```

```
end
```

(그림 24. 낮/밤 설정 코드)

//밤낮 표현

```
always@(posedge clk) begin
```

```
    if(rst) day = 8'b1111_1100;
```

```
    else begin
```

```
        if(flag==0) day = 8'b1111_1100;
```

```
        else day = 8'b1110_1100;
```

```
    end
```

```
end
```

```
always@(posedge clk)begin
```

```
    if(rst) seg_data2 = 8'b1111_1100; // 7segment 에 연결
```

```
    else begin
```

```
        seg_data2 = day;
```

```
    end
```

```
end
```

(그림 25. 7-segment output 설정 코드)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

D) 초침과 스텝모터 동기화

ADC	파일 폴더
DAC	파일 폴더
FPGA	파일 폴더
Piezo	파일 폴더
Servo Motor	파일 폴더
SRAM	파일 폴더
UART	파일 폴더
USBtoSerial	파일 폴더
Voltage Meter	파일 폴더

(그림 26. combo-II 데이터 시트)

```
always@(posedge clk)begin
    if(rst) motor_state <=0;
    else begin
        if(state==2'b00)begin
            if(rst) motor_state <=0;
            else begin
                if(cnt==99999)begin
                    motor_state <= motor_state +1;
                end
                else motor_state <= motor_state;
            end
        end
    end
end
```

(그림 27. 초침에 동기화한 설정 코드)

```
//motor 각도조정
always@(posedge clk)begin
    case(motor_state)
        2'd0 : smotor = 4'b0011;
        2'd1 : smotor = 4'b0110;
        2'd2 : smotor = 4'b1100;
        2'd3 : smotor = 4'b1001;
    endcase
end
```

(그림 28. 스텝 모터 제어)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

E) 시각(hour)을 통한 해, 달 위치 나타내기

// 시간 조정

```
always @(posedge clk)
begin
    if(rst) motor_cnt <=0;
    else begin
        if(state==2'b00)begin
            if(motor_cnt >=1999) motor_cnt <=0;
            else motor_cnt = motor_cnt+1;
        end
    end
end
```

(그림 29. 신호 주기 설정하기)

```
always @(motor_cnt, tmp) begin
    if (motor_cnt < tmp) servo = 1;
    else servo = 0;
end
```

(그림 30. 주기에 따라 신호 넣기)

// 모터제어 1

```
always@(posedge clk)begin
    if(rst) a<=0;
    else begin
        if(state==2'b00)begin
            if(a>99999 + 6*60)begin
                a<=0;
            end
            else a <= a +1;
        end
    end
end
```

(그림 31. 기본 설정의 시간에 따른 각도 조정)

2. 기능 분석 및 영상 – Mode1

1) 시간 나타내기 00:00:00:0 (hour:min:second: 100milli-s : 10milli-s)

E) 시각(hour)을 통한 해, 달 위치 나타내기

```
// 모터제어 1
always@(posedge clk)begin
    if(rst) a<=0;
    else begin
        if(state==2'b00)begin
            if(a>99999 * 60*60)begin
                a<=0;
            end
            else a <= a +1;
        end
    end

//모터 제어 2
always @(posedge clk)
if (rst) tmp = 150; // 00시 기준
else begin
    if(a>99999 * 60 *60)begin
        tmp <= tmp+1;
    end

    else begin
        if(flag ==0)begin //
            case(hour[0])
                4'd5: tmp = 75; //저녁 5시
                4'd4: tmp = 90;
                4'd3: tmp = 105;
                4'd2: tmp = 120;
                4'd1: tmp = 135;
                4'd0: if(hour[1]==1)begin
                    tmp =120;
                end
                else tmp = 150;

                4'd6: tmp = 60; //저녁 6시

                4'd7: tmp = 75;
                4'd8: tmp = 90;
                4'd9: tmp = 105;

            endcase
        end
        else begin
            case(hour[0])
                4'd9: tmp = 195;
                4'd8: tmp = 210;

            endcase
        end
    end
end

4'd9: tmp = 195;
4'd8: tmp = 210;
4'd7: tmp = 225;

4'd6: tmp = 240; //새벽 6시
4'd5: tmp = 225;
4'd4: tmp = 210;
4'd3: tmp = 195;
4'd2: tmp = 180;
4'd1: tmp = 165;

4'd0: if(hour[1]==1)begin
    tmp = 180;
end
else tmp = 150;

endcase
end
end
```

(그림 32. 낮/밤, 시각 상태를 고려한 각도 설정)

2. 기능 분석 및 영상 – Mode 2

2) Stop Watch 기능

A) 버튼을 이용한 STOP & MOVE

```
//스탑와치 flag 세우기
always@(posedge clk)begin
    if(rst) s_flag <=0;
    else if(s_start)begin
        s_flag <=1;
    end
    else if(s_stop) begin
        s_flag <=0;
    end
    else begin
        s_flag <= s_flag;
    end
end
```

(그림 33. stop_watch_flag를 이용해 시간 제어)

B) 초침에 따른 스텝모터

```
else if(state==2'b01 & s_flag==1)begin
    if(rst) motor_state <=0;
    else begin
        if(cnt==99999)begin
            motor_state <= motor_state +1;
        end
        else motor_state <= motor_state;
    end
end
```

(그림 34. flag상태에 따른 스텝모터 제어)

2. 기능 분석 및 영상 – Mode 3

3) Alarm Clock

A) 버튼을 이용한 Alarm ON/OFF 설정

```
//alarm start 알리기
always@(posedge clk)begin
    if(rst) alarm<=0;
    else begin
        if(state==2'b10)begin
            if(s_start ==1)begin //알람시작 : *버튼
                alarm <=1;
            end
            else if(alarm ==1 && alarm_end==1)begin
                alarm=0;
            end
        end
    end
end
```

(그림 35. 알람 start 설정)

```
// alarm 끝난거 알리기
always@(posedge clk)begin
    if(rst) alarm_end<=0;
    else begin
        if(state==2'b10)begin
            if(rst) alarm_end<=0;
            else if(alarm==1&sec[0]==0&sec[1]==0&min[0]==0&min[1]==0&hour[1]==0&hour[0]==0)begin
                alarm_end<=1;
            end
            else alarm_end<=alarm_end;
        end
    end
end
```

(그림 36. 알람 end 설정)

3) Alarm Clock

```
always@(posedge clk)begin
    if(rst) begin
        piezo_cnt <=0;
        piezo_time <=0;
        melody <=0;
    end
    else begin
        if(state==2'b10)begin
            if(alarm_end ==1)begin
                //음나오는 시간 조정 0.5초
                if(piezo_time<50000)begin
                    piezo_time <= piezo_time+1;
                    if(piezo_cnt >= piezo_half)begin
                        piezo <= !piezo;
                        piezo_cnt <= 0;
                    end
                    else piezo_cnt = piezo_cnt+1;
                end
            end
        end
    end
end
```

```

end
else begin
    if(melody==38)begin
        piezo_time <=0;
        melody <= 0;
    end
    else begin
        piezo_time <=0;
        melody<= melody+1;
    end
end

end

end

else if(state==2'b11)begin
    if(game_flag ==1)begin
        if(piezo_cnt >= piezo_half)begin
            piezo <= !piezo;
            piezo_cnt <= 0;
        end
        else piezo_cnt = piezo_cnt+1;
    end
end

end

end

end

```

(그림 38. piezo 알람 설정)

27

2. 기능 분석 및 영상 – Mode 3

3) Alarm Clock

C) 주파수 변환을 통한 알람 음계 설정

(단위 : Hz)

옥타브 음계	1	2	3	4	5	6	7	8
C(도)	32.7032	65.4064	130.8128	261.6256	523.2511	1046.502	2093.005	4186.009
C#	34.6478	69.2957	138.5913	277.1826	554.3653	1108.731	2217.461	4434.922
D(레)	36.7081	73.4162	146.8324	293.6648	587.3295	1174.659	2349.318	4698.636
D#	38.8909	77.7817	155.5635	311.1270	622.2540	1244.508	2489.016	4978.032
E(미)	41.2034	82.4069	164.8138	329.6276	659.2551	1318.510	2637.020	5274.041
F(파)	43.6535	87.3071	174.6141	349.2282	698.4565	1396.913	2793.826	5587.652
F#	46.2493	92.4986	184.9972	369.9944	739.9888	1479.978	2959.955	5919.911
G(솔)	48.9994	97.9989	195.9977	391.9954	783.9909	1567.982	3135.963	6271.927
G#	51.9130	103.8262	207.6523	415.3047	830.6094	1661.219	3322.438	6644.875
A(라)	55.0000	110.0000	220.0000	440.0000	880.0000	1760.000	3520.000	7040.000
A#	58.2705	116.5409	233.0819	466.1638	932.3275	1864.655	3729.310	7458.620
B(시)	61.7354	123.4708	246.9417	493.8833	987.7666	1975.533	3951.066	7902.133

(그림 39. 음계에 따른 주파수)

293, 391, 493, 587,
587, 523, 493, 440,
493, 391, 493, 587,

783, 783, 783, 880,
693, 659, 698, 440,
554, 698, 880, 783,

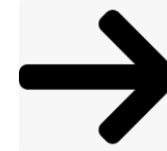
698, 659, 698, 783,
698, 659, 587, 523,
493, 523, 587, 523,

391, 440

170 127 101 85
85 95 101 113
101 127 101 85

63 63 63 56
72 75 71 113
90 71 56 63

71 75 71 63
71 75 85 95
101 95 85 95
127 113



(그림 40. 클럭 주파수에 따른 변환 결과)

2. 기능 분석 및 영상 – Mode 3

3) Alarm Clock

D) 알람 멜로디 설정 (하울의 움직이는 성)

```
always@(posedge clk)begin
    if(rst) piezo_half = 0;
    else begin
        //노래 너무빨리 재생 금지
        if(state==2'b10 & alarm_end==1)begin
            case(melody)
                6'd1 : piezo_half = 12'd170;
                6'd2 : piezo_half = 12'd127;
                6'd3 : piezo_half = 12'd101;
                6'd4 : piezo_half = 12'd85;
                6'd5 : piezo_half = 12'd85;
                6'd6 : piezo_half = 12'd95;
                6'd7 : piezo_half = 12'd101;
                6'd8 : piezo_half = 12'd113;
                6'd9 : piezo_half = 12'd101;
                6'd10 : piezo_half = 12'd127;
                6'd11 : piezo_half = 12'd101;
                6'd12 : piezo_half = 12'd85;
                6'd13 : piezo_half = 12'd63;
                6'd14 : piezo_half = 12'd63;
```

(그림 41. piezo 알람 설정)

```
        6'd16 : piezo_half = 12'd56;
        6'd17 : piezo_half = 12'd72;
        6'd18 : piezo_half = 12'd75;
        6'd19 : piezo_half = 12'd71;
        6'd20 : piezo_half = 12'd113;
        6'd21 : piezo_half = 12'd90;
        6'd22 : piezo_half = 12'd71;
        6'd23 : piezo_half = 12'd56;
        6'd24 : piezo_half = 12'd63;
        6'd25 : piezo_half = 12'd71;
        6'd26 : piezo_half = 12'd75;
        6'd27 : piezo_half = 12'd71;
        6'd28 : piezo_half = 12'd63;
        6'd29 : piezo_half = 12'd71;
        6'd30 : piezo_half = 12'd75;
        6'd31 : piezo_half = 12'd85;
        6'd32 : piezo_half = 12'd95;
        6'd33 : piezo_half = 12'd101;
        6'd34 : piezo_half = 12'd95;
        6'd35 : piezo_half = 12'd85;
        6'd28 : piezo_half = 12'd63;
        6'd29 : piezo_half = 12'd71;
        6'd30 : piezo_half = 12'd75;
        6'd31 : piezo_half = 12'd85;
        6'd32 : piezo_half = 12'd95;
        6'd33 : piezo_half = 12'd101;
        6'd34 : piezo_half = 12'd95;
        6'd35 : piezo_half = 12'd85;
        6'd36 : piezo_half = 12'd95;
        6'd37 : piezo_half = 12'd127;
        6'd38 : piezo_half = 12'd113;
    endcase
end
```

(그림 42. 하울의 움직이는 성 주파수 설정)

2. 기능 분석 및 영상 – Mode 3

3) Alarm Clock

E) 음악 박자 및 초침에 따른 스텝모터 구동

```
if(state==2'b10 & alarm_end==1)begin
    if(rst) motor_state <=0;
    else begin
        if(cnt==99999)begin
            motor_state <= motor_state +1;
        end
        else motor_state <= motor_state;
    end
end
end
```

(그림 43. 노래에 따른 스텝모터 설정)

```
always@(posedge clk)begin
    case(motor_state)
        2'd0 : smotor = 4'b0011;
        2'd1 : smotor = 4'b0110;
        2'd2 : smotor = 4'b1100;
        2'd3 : smotor = 4'b1001;
    endcase
end
```

(그림 44. 주기에 따른 스텝모터 설정)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

A) 게임 설명

- Start 버튼을 누르고 버튼 입력 시 시간 측정 진행
- Lcd에 뜨는 1에 해당하는 버튼을 눌러 0으로 만들면 성공
- 누가 더 빨리 1의 숫자를 없애는 가 → 승리 조건

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

B) 게임 모드 설정

```
always@(posedge clk)begin
    if(rst) random_case <=0;
    if(state==2'b11)begin
        if(random_case==99)begin
            random_case <= 0;
        end
        random_case <=random_case +1;
    end
end

always@(posedge clk) begin
    if(rst) rhy=0;
    if(state==2'b11)begin
        if(s_start)begin
            rhy <= (random_case/10);
        end
    end
end
```

(그림 45. 난수 생성을 통해 게임 모드 설정이 랜덤하게 진행되게 함)

```
//리듬생성
always@(posedge clk)begin
    if(state==2'b11)begin
        if(rst) game_number<=0;
        if(s_start==1)begin
            case(rhy)
                4'd0: game_number=6'b111101;
                4'd1: game_number= 6'b101111;
                4'd2: game_number= 6'b111111;
                4'd3: game_number= 6'b101001;
                4'd4: game_number =6'b110111;
                4'd5: game_number= 6'b101101;
                4'd6: game_number= 6'b111001;
                4'd7: game_number= 6'b101111;
                4'd8: game_number= 6'b101011;
                4'd9: game_number= 6'b111100;
            endcase
        end
    end
end
```

(그림 46. 난수에 따른 10개의 게임 난이도)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

C) 게임 시작 설정

```
//game_flag
always@(posedge clk)begin
    if(rst) game_flag = 0;
    else begin
        if(state==2'b11)begin
            if(s_start==1)begin
                game_flag=1;
            end
        else begin
            if(sec[0]==0&sec[1]==0&min[1]==0&min[0]==0& hour[1]==0 & hour[0]==0)begin
                game_flag =0;
            end
            else if(game_time[5] ==9 &game_time[4]==9 & game_time[3]==9) begin
                game_flag = 0;
            end
        end
    end
end
end
end
end
end
```

(그림 47. flag를 통한 게임 시작 / 종료 설정)

D) 게임 시간 설정

```

else if(state==2'b11)begin
    if(rst) cnt<=0;
    else begin
        if(game_flag==0) cnt<=0;
        else begin
            if(game_flag==1)begin
                if(cnt==99999) begin
                    cnt <= 0;
                end
                else cnt<=cnt+1;
            end
        end
    end
end
end
end

//game time 정하기
always@(posedge clk)begin
    if(rst) begin
        game_time[0]<=0;
        game_time[1]<=0;
        game_time[2]<=0;
        game_time[3]<=0;
        game_time[4]<=0;
        game_time[5]<=0;
    end
    else begin
        if(state==2'b11)begin
            if(game_flag==1&&cnt==99999)begin
                if(game_time[5] ==9 &game_time[4]==9 & game_time[3]==9)begin
                    game_time[0]<=0;
                    game_time[1]<=0;
                    game_time[2]<=0;
                    game_time[3]<=0;
                    game_time[4]<=0;
                    game_time[5]<=0;
                end
            end
            game_time[5] = game_time[5] +1;
        end
    end
end

end
game_time[5] = game_time[5] +1;
end

```

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

E) 게임 시작 중 설정

```
else if(state==2'b11)begin
    if(game_flag ==1)begin
        if(piezo_time<20000)begin
            piezo_time <= piezo_time+1;
            if(piezo_cnt >= piezo_half)begin
                piezo <= !piezo;
                piezo_cnt <= 0;
            end
            else piezo_cnt = piezo_cnt+1;
        end
    else begin
        if(melody==38)begin
            piezo_time <=0;
        end
        else begin
            piezo_time <=0;
        end
    end
end
end
```

(그림 50. 게임 시작 시 진행음 발생)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

F) 게임 종료 후 설정

```
else if(state==2'b11)begin
    if(rst) m_min<=4'b1010;
    else begin
        m_min <= 4'b1010;
    end
end
```

(그림 51. 게임종료 후 m_min 자리의 설정)

```
module seg_decoder(bcd, seg_data);
    input [3:0] bcd;
    output [7:0] seg_data;
    reg [7:0] seg_data;
    always @(bcd) begin
        case(bcd)
            4'H0 : seg_data=8'b1111_1100; // 'HFC
            4'H1 : seg_data=8'b0110_0000; // 'H60
            4'H2 : seg_data=8'b1101_1010; // 'HDA
            4'H3 : seg_data=8'b1111_0010; // 'HF2
            4'H4 : seg_data=8'b0110_0110; // 'H66
            4'H5 : seg_data=8'b1011_0110; // 'HB6
            4'H6 : seg_data=8'b1011_1110; // 'HBE
            4'H7 : seg_data=8'b1110_0000; // 'HE0
            4'H8 : seg_data=8'b1111_1110; // 'HFE
            4'H9 : seg_data=8'b1111_0110; // 'HF6
            4'Ha : seg_data=8'b0000_0000; // 'HEE
            4'Hb : seg_data=8'b0011_1110; // 'H3E
            4'Hc : seg_data=8'b1001_1100; // 'H9C
            4'Hd : seg_data=8'b0111_1010; // 'H7A
            4'He : seg_data=8'b1001_1110; // 'H9E
            4'Hf : seg_data=8'b1000_1110; // 'H8E
        endcase
    end
endmodule
```

(그림 52. INPUT 값이 10일 때의 7-SEG 설정)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

F) 게임 종료 후 설정 (1000mm, 100mm second 부분 설정)

```
else if(state==2'b11)begin
    if(s_start==1)begin
        hour[0] <= game_number[4];
        hour[1] <= game_number[5];
    end
    else begin
        if(sec[0]==0&sec[1]==0&min[1]==0&min[0]==0&hour[1]==0&hour[0]==0)begin
            hour[0] <=game_time[4];
            hour[1]<=game_time[5];
        end
        else if(game_flag==1)begin
            if(m_down==1) hour[0]<=0;
            else if(s_down==1) hour[1] <=0;
            else begin
                hour[0] <= hour[0];
                hour[1] <= hour[1];
            end
        end
    end
end
end
end
end
```

(그림 53. 1s, 100mm-> hour 부분에 나타내기)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

F) 게임 종료 후 설정 (10mm, 1mm second 부분 설정)

```
// state 11
else if(state==2'b11)begin
    if(s_start==1)begin
        min[0] <= game_number[2];
        min[1] <= game_number[3];
    end
    else begin
        if(s_stop==1&sec[0]==0&sec[1]==0&min[1]==0&min[0]==0&hour[1]==0&hour[0]==0)begin
            min[0] <=game_time[2];
            min[1]<=game_time[3];
        end
        else if(game_flag==1)begin
            if(s_up==1) min[0]<=0;
            else if(h_down==1) min[1] <=0;
            else begin
                min[0] <=min[0];
                min[1] <= min[1];
            end
        end
    end
end
end
```

(그림 54. 10mm, 1mm -> min 부분에 나타내기)

2. 기능 분석 및 영상 – Mode 4

4) 시간을 이용한 순발력 게임

F) 게임 종료 후 설정 (0.1mm, 0.01mm second 부분 설정)

```
//state 11
else if(state==2'b11)begin
    if(s_start==1)begin
        sec[0] <= game_number[0];
        sec[1] <= game_number[1];
    end
    else begin
        if(sec[0]==0&sec[1]==0& min[0] ==0 & min[1]==0&hour[0]==0 & hour[1]==0)begin
            sec[0] <=game_time[0];
            sec[1]<=game_time[1];
        end
        else if(game_flag==1)begin
            if(h_up==1) sec[0]<=0;
            else if(m_up==1) sec[1] <=0;
        end
    end
end
end
end
```

(그림 55. 0.1mm, 0.01mm 설정-> second 부분에 나타내기)

3. 결론 및 구현 코드

A) 구현 결과

기본 시계 기능

- 시간 나타내기
- 시간 조정하기

추가 기능

- Alarm 기능
- 스톱와치 기능
- 게임기능

3. 결론 및 구현 코드

B) 구현 코드내용

글 제목

[서울시립대 전기전자설계실험 II] verilog를 이용한 디지털 시계

2022 SeoulTech 지능형 창작로봇대회

2022 공학페스티벌 캡스톤 디자인 경진대회

2022 서울시립대 창의공학설계 경진대회

영화 <올빼미> (6)

디지털 설계

[서울시립대 전기전자설계실험 II] verilog를 이용한 디지털 시계

고릴라 38분 전

URL 복사

로그인

• Verilog를 이용한 디지털 시계 구현

가능

• 메뉴를 통한 4가지 모드 설정

A. Mode 1

• 시간 나타내기 (시, 분, 초, 100ms)

• 시간 조정 (시, 분, 초)

• 낮, 밤 나타내기 (7-segment)



seongwon_uos

Top Repositories

New

Find a repository...

- xswzaq789/water_float_Robot
- Worldconqueror/data_structure
- Worldconqueror/cctv_toy_project
- Worldconqueror/pyramids-permutation

<https://blog.naver.com/airplon>

<https://github.com/Worldconqueror>

감사합니다

20194440121 조성원
Verilog를 이용한 디지털시계 만들기



서울시립대학교
UNIVERSITY OF SEOUL