

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

آزمایش 3

گزارش کار آزمایشگاه FPGA

نام استاد : دکتر ثامنی

ایمان استوار 9332366

علیرضا رحمان ستایش 9332679

# Arbitrary Waveform Generation

هدف طراحی ماژولی برای ساخت هرگونه موج دلخواه است .

طرح ( پیاده سازی: )

برای پیاده سازی از ماشین حالت دو حالت استفاده میکنیم به صورتی که دو **STATE** که حالت اول موج کمتر از **THERESHOLD** و حالت 2 زمانیست که موج بیشتر از **THERESHOLD** است .

مقدار **THERESHOLD** توسط **DIPSWTICH** تعیین میشود .

یک شمارنده 64 بیتی به نام **PULSNUM** مدام با فرکانس تقسیم شده توسط ماژول **FEC\_DIVIDER** با فرکانس 6MHZ می‌شمارد.

حال هنگامی که  $\{ADC\_A[9:6], 6'b000000\} > threshold$  ، **STATE0** به **STATE1** میرود و مقدار شمارنده را به عنوان **TOA** ذخیره میکنیم . مقدار **PA** را صفر قرار میدهیم که بعدا **MAX** پالس شروع شده را در آن بریزیم .

در هنگام انتقال از حالت 1 به 0 یعنی پایان پالس **PW** را محاسبه میکنیم .

انتقال از حالت 0 به 1:

```
if ({ADC_A[9:6],6'b000000}>threshold)begin
    if (state==1'b0)begin
        TOA <= pulsenum;
        PA <= 0;
    end
    state<=1'b1;
end
```

انتقال از حالت 1 به 0:

```

if ({ADC_A[9:6],6'b000000}<=threshold)begin
    if (state==1'b1)begin
        PW <= pulsenum-TOA;
    end
    state '1=>b0;
end

```

محاسبه PA:

```

if (state==1'b1)begin
    if (ADC_A > PA )
        PA <= ADC_A;
End

```

SCALE کردن PA , PW و نمایش آن ها توسط 7SEGMENT

```

if(PW>64'd30000) segment1in<=4'd9;
    else if (PW>64'd24000) segment1in<=4'd8;
    else if(PW>64'd20000) segment1in<=4'd7;
    else if(PW>64'd16000) segment1in<=4'd6;
    else if(PW>64'd12000) segment1in<=4'd5;
    else if(PW>64'd8000) segment1in<=4'd4;
    else if(PW>64'd4000) segment1in<=4'd3;
    else if(PW>64'd1000) segment1in<=4'd2;
    else if(PW>64'd600) segment1in<=4'd1;
    else segment1in<=4'd0;

if(PA>900) segment2in<=4'd9;

```

```
else if(PA>800) segment2in<=4'd8;  
else if(PA>700) segment2in<=4'd7;  
else if(PA>600) segment2in<=4'd6;  
else if(PA>500) segment2in<=4'd5;  
else if(PA>400) segment2in<=4'd4;  
else if(PA>300) segment2in<=4'd3;  
else if(PA>200) segment2in<=4'd2;  
else if(PA>100) segment2in<=4'd1;  
else segment2in<=4'd0;
```

به این طریق PA , PW را بازه بندی و قابل نمایش بر روی 7SEGMENT کردیم .

# کد کامل 1:

```
`timescale 1ns / 1ps
```

```
////////////////////////////////////////////////////////////////
```

```
// Company:
```

```
// Engineer:
```

```
//
```

```
// Create Date: 22:57:06 03/11/2017
```

```
// Design Name:
```

```
// Module Name: Top_module
```

```
// Project Name:
```

```
// Target Devices:
```

```
// Tool versions:
```

```
// Description:
```

```
//
```

```
// Dependencies:
```

```
//
```

```
// Revision:
```

```
// Revision 0.01 - File Created
```

```
// Additional Comments:
```

```
//
```

```
////////////////////////////////////////////////////////////////
```

```
module Top_module(
```

```
    input SysClk,
```

```
    output [ 6 : 0 ] segment2pin,
```

```
    output [ 6 : 0 ] segment1pin,
```

```
    input [7:0] DipSw,
```

```
    input Reset,
```

```
    //ADC
```

```

input [ 9:0 ] ADC_A,

output ADC_CLKIN,

output ADC_OE,

output SCLK,

output SDI      );

wire clockout;

reg [3:0] segment2in;

reg [3:0] segment1in;

reg state;

wire [31:0]pulsenum;

reg [63:0]TOA;

reg [63:0]PW ;

reg [9:0]PA;

reg count ;

reg [9:0]threshold ;

```

```

initial begin

PA <=10'b0000000000;

TOA <= 64'b0;

state <= 0;

end

```

```

always @ (posedge SysClk) begin

```

```

    threshold={DipSw,2'b00};

    if ({ADC_A[9:6],6'b000000}>threshold)begin

        if (state==1'b0)begin

            TOA <= pulsenum;

            PA <= 0;

        end

        state<=1'b1;

    end
end

```

```

if ({ADC_A[9:6],6'b000000}<=threshold)begin
    if (state==1'b1)begin
        PW <= pulsenum-TOA;
    end
    state    <=1'b0;
end

```

```

if (state==1'b1)begin
    if (ADC_A > PA )
        PA <= ADC_A;
end

```

```

if(PW>64'd30000) segment1in<=4'd9;
else if (PW>64'd24000) segment1in<=4'd8;
else    if(PW>64'd20000) segment1in<=4'd7;
else    if(PW>64'd16000) segment1in<=4'd6;
else    if(PW>64'd12000) segment1in<=4'd5;
else    if(PW>64'd8000) segment1in<=4'd4;
else    if(PW>64'd4000) segment1in<=4'd3;
else    if(PW>64'd1000) segment1in<=4'd2;
else    if(PW>64'd600) segment1in<=4'd1;
else segment1in<=4'd0;

```

```

if(PA>900) segment2in<=4'd9;
else if(PA>800) segment2in<=4'd8;
else    if(PA>700) segment2in<=4'd7;
else    if(PA>600) segment2in<=4'd6;
else    if(PA>500) segment2in<=4'd5;
else    if(PA>400) segment2in<=4'd4;
else    if(PA>300) segment2in<=4'd3;
else    if(PA>200) segment2in<=4'd2;
else    if(PA>100) segment2in<=4'd1;
else segment2in<=4'd0;

```

end

```
//      BCD_Counter myBCD_Counter (  
//  .in(counter),  
//  .clk(SysClk),  
//  .out1(segment2in),  
//  .out0(segment1in)  
//  );
```

```
Decoder_7 segment1 (  
    .in(segment1in),  
    .out(segment1pin)  
);
```

```
Decoder_7 segment2 (  
    .in(segment2in),  
    .out(segment2pin)  
);
```

```
Fec_divider mynewfec(  
    .clock(SysClk),  
    .divnum(6'd2),  
    .clkout(clockout)  
);
```

```
ADC myADC (  
    .clk(clockout),  
    .in(ADC_A),  
    .ADC_CLKIN(ADC_CLKIN),  
    .SCLK(SCLK),  
    .ADC_OE(ADC_OE),  
    .SDI(SDI)
```



```
);
```

```
Binary_Counter mybinary (
```

```
    .clk(clockout),
```

```
    .resetbar(Reset),
```

```
    .enable(1'b1),
```

```
    .counter(pulsenum)
```

```
);
```

```
endmodule
```

## صورت مسئله:2

### Pulse detector with two threshold

هدف طراحی ماژولی برای تشخیص پالس موجی با پس زمینه نویز است .

و تشخیص  $POA, PW, TOA$  برای هر پالس

طرح ( پیاده سازی:)

برای پیاده سازی از ماشین حالت 4 حالت استفاده میکنیم به صورتی که دو **STATE** که حالت اول موج کمتر از  $THERESHOLD1$  و حالت 2 زمانیست که موج بیشتر از  $THERESHOLD1$  و کمتر از  $THERESHOLD2$  است و از حالت قبل حالت 1 بوده است . پس از آن اگر اندازه پالس از  $THERESHOLD2$  بیشتر شود به حالت 3 می رود و شروع پالس تشخیص داده میشود .

مقدار شمارنده را به عنوان  $TOA$  ذخیره میکنیم . مقدار  $PA$  را صفر قرار میدهیم که بعدا  $MAX$  پالس شروع شده را در آن بریزیم .

سپس اگر پالس بیشتر از  $THERESHOLD1$  و کمتر از  $THERESHOLD2$  شود به حالت 4 میرویم و  $PW$  محاسبه میکنیم .

مقدار  $THERESHOLD$  ها توسط  $DIPSWITCH$  تعیین میشود .

یک شمارنده 64 بیتی به نام  $PULSNUM$  مدام با فرکانس تقسیم شده توسط ماژول  $FEC\_DIVIDER$  با فرکانس 6MHZ می شمارد.

پیاده سازی ماشین حالت :

```
if ({ADC_A[9:6],6'b000000} < threshold1)begin
    if (state==2'd1)begin
        state<=2'd0;
    end
    if (state==2'd2)begin
        state<=2'd3;
```

```

        end
        if (state==2'd3)begin//pulse ended
            state<=2'd0;
            PW <= pulsenum-TOA;
        end
    end
end

```

```

        if ({ADC_A[9:6],6'b000000}>=threshold1 && {ADC_A[9:6],6'b000000} <
threshold2)begin
            if (state==2'd0)begin
                state<=2'd1;
            end
            if (state==2'd2)begin
                state<=2'd3;
            end
        end
    end
end

```

```

        if ({ADC_A[9:6],6'b000000} >= threshold2)begin
            if (state==2'd0)begin
                state<=2'd1;
            end
            if (state==2'd1)begin//pulse detected
                state<=2'd2;
                TOA <= pulsenum;
                PA <= 0;
            end
        end
    end
end

```

```

        count=count+1;
    end
    if (state==2'd3)begin
        state<=2'd2;
    end
end

```

محاسبه PA:

```

if (state==1'b1)begin
    if (ADC_A > PA )
        PA <= ADC_A;
    End

```

**SCALE** کردن **PA** , **PW** و نمایش آن ها توسط **7SEGMENT**

```

if(PW>64'd30000) segment1in<=4'd9;
    else if (PW>64'd24000) segment1in<=4'd8;
    else if(PW>64'd20000) segment1in<=4'd7;
    else if(PW>64'd16000) segment1in<=4'd6;
    else if(PW>64'd12000) segment1in<=4'd5;
    else if(PW>64'd8000) segment1in<=4'd4;
    else if(PW>64'd4000) segment1in<=4'd3;
    else if(PW>64'd1000) segment1in<=4'd2;
    else if(PW>64'd600) segment1in<=4'd1;
    else segment1in<=4'd0;

```

```
if(PA>900) segment2in<=4'd9;  
else if(PA>800) segment2in<=4'd8;  
else if(PA>700) segment2in<=4'd7;  
else if(PA>600) segment2in<=4'd6;  
else if(PA>500) segment2in<=4'd5;  
else if(PA>400) segment2in<=4'd4;  
else if(PA>300) segment2in<=4'd3;  
else if(PA>200) segment2in<=4'd2;  
else if(PA>100) segment2in<=4'd1;  
else segment2in<=4'd0;
```

به این طریق PA , PW را بازه بندی و قابل نمایش بر روی 7SEGMENT کردیم .

## نتیجه گیری :

(بعد از program شدن برد)

پس از scale کردن PA ,PW هرکدام را به نسبت دادن به اعداد 0 تا 9 بر روی 7segmenta نمایش دادیم

با تغییر فرکانس یا همان pw و تغییر amplitude یا همان PA تغییرات مورد انتظار را روی 7segment ها دیدیم .  
وقتی که دامنه از threshold ها رد نمیشد تمام اعداد 0 میشدند .

# کد کامل 2:

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 22:57:06 03/11/2017
// Design Name:
// Module Name: Top_module
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////
module Top_module(
    input SysClk,

    output [ 6 : 0 ] segment2pin,
    output [ 6 : 0 ] segment1pin,
    input [7:0] DipSw,
    input Reset,
    //ADC
    input [ 9:0 ] ADC_A,
    //input [ 9:0 ] ADC_A,
```

```
output ADC_CLKIN,  
output ADC_OE,  
output SCLK,  
output SDI    );
```

```
wire clockout;  
reg [3:0] segment2in;  
reg [3:0] segment1in;  
reg [1:0]state;  
wire [63:0]pulsenum;  
reg [63:0]TOA;  
reg [63:0]PW ;  
reg [9:0]PA;  
reg [920:0]count ;  
reg [9:0]threshold1 ;  
reg [9:0]threshold2 ;
```

```
//assign ADC_A=ADC_A;  
initial begin  
PA <=10'b0000000000;  
TOA <= 64'b0;  
state <= 2'b0;  
count=10'b0;  
end
```

```
always @ (posedge SysClk) begin
```

```
threshold1={DipSw[3:0],6'b00};  
threshold2={DipSw[7:4],6'b00};
```

```
if ({ADC_A[9:6],6'b000000} < threshold1)begin  
    if (state==2'd1)begin  
        state<=2'd0;  
    end  
    if (state==2'd2)begin  
        state<=2'd3;
```



```

        end

        if (state==2'd3)begin//pulse ended

            state<=2'd0;

            PW <= pulsenum-TOA;

        end
    end

    if ({ADC_A[9:6],6'b000000}>=threshold1 && {ADC_A[9:6],6'b000000} < threshold2)begin

        if (state==2'd0)begin

            state<=2'd1;

        end

        if (state==2'd2)begin

            state<=2'd3;

        end

    end

    if ({ADC_A[9:6],6'b000000} >= threshold2)begin

        if (state==2'd0)begin

            state<=2'd1;

        end

        if (state==2'd1)begin//pulse detected

            state<=2'd2;

            TOA <= pulsenum;

            PA <= 0;

            count=count+1;

        end

        if (state==2'd3)begin

            state<=2'd2;

        end

    end

    if (state==2'd2)begin

        if (ADC_A > PA )

            PA <= ADC_A;

        end
    end

```

```

if(PW>64'd9000) segment1in<=4'd9;
else if (PW>64'd8000) segment1in<=4'd8;
else    if(PW>64'd7000) segment1in<=4'd7;
else    if(PW>64'd6000) segment1in<=4'd6;
else    if(PW>64'd5000) segment1in<=4'd5;
else    if(PW>64'd4000) segment1in<=4'd4;
else    if(PW>64'd3000) segment1in<=4'd3;
else    if(PW>64'd2000) segment1in<=4'd2;
else    if(PW>64'd1000) segment1in<=4'd1;
else segment1in<=4'd0;

```

```

if(PA>1000) segment2in<=4'd9;
else if(PA>950) segment2in<=4'd8;
else    if(PA>900) segment2in<=4'd7;
else    if(PA>850) segment2in<=4'd6;
else    if(PA>800) segment2in<=4'd5;
else    if(PA>750) segment2in<=4'd4;
else    if(PA>700) segment2in<=4'd3;
else    if(PA>650) segment2in<=4'd2;
else    if(PA>600) segment2in<=4'd1;
else segment2in<=4'd0;

```

```

end

```

```

/*BCD_Counter myBCD_Counter (
.in(PW[]),
.clk(SysClk),
.out1(segment2in),
.out0(segment1in)
);*/

```

```
Decoder_7 segment1 (  
    .in(segment1in),  
    .out(segment1pin)  
);
```

```
Decoder_7 segment2 (  
    .in(segment2in),  
    .out(segment2pin)  
);
```

```
Fec_divider mynewfec(  
    .clock(SysClk),  
    .divnum(6'd2),  
    .clkout(clockout)  
);
```

```
ADC myADC (  
    .clk(clockout),  
    .in(ADC_A),  
    .ADC_CLKIN(ADC_CLKIN),  
    .SCLK(SCLK),  
    .ADC_OE(ADC_OE),  
    .SDI(SDI)  
);
```

```
Binary_Counter mybinary (  
    .clk(clockout),  
    .resetbar(Reset),  
    .enable(1'b1),  
    .counter(pulsenum)  
);
```

```
endmodule
```

