Basys 3 클럭

Oscillators/ Clocks

The Basys 3 board includes a single 100MHz oscillator connected to pin W5. W5 is a MRCC input on bank 34). The input clock can drive MMCMs or PLLs to generate clocks of various frequencies and with known phase relationships that may be needed throughout a design. Some rules restrict which MMCMs and PLLs may be driven by the 100MHz input clock. For a full description of these rules and of the capabilities of the Artix-7 clocking resources, refer to the "7 Series FPGAs Clocking Resources User Guide" available from Xilinx https://docs.xilinx.com/v/u/en-US/ug472_7Series_Clocking.

Xilinx offers the Clocking Wizard IP core to help users generate the different clocks required for a specific design. This wizard properly instantiates the needed MMCMs and PLLs based on the desired frequencies and phase relationships specified by the user. The wizard will then output an easy to use wrapper component around these clocking resources that can be inserted into the user's design. The Clocking Wizard can be accessed from within IP Catalog, which can be found under the Project Manager section of the Flow Navigator in Vivado.

모든 동작은 조합회로로 표현, 필요한 부분에 DFF를 넣어줘 시간에 동기화시켜 줍니다.

카운터

```
module MyCnt(
    input CLK, //U18, BTNC
    output [3:0] LEDON //V19,U19,E19,U16
);
   reg [3:0] cnt;
   always @(posedge CLK)
      cnt = cnt + 1;
   assign LEDON = cnt;
endmodule
```

1. CLK 일반 핀에 연결해서 확인!
2. CLK clock 핀(W5)에 연결해서 확인 -> LED 밝기 확인! 1/2, 1/4, 1/8, 1/16

클럭 소스 테스트

```
module MyCnt(
    input CLK, //W5
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        cnt = cnt + 1;

    assign LEDON = cnt[26];

endmodule
```

시뮬레이션 해 볼것!
정확한 1초! 1

```
module MyCnt(
```

```
    input CLK, //W5
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        cnt = cnt + 1;
        if(cnt==100000000)
            cnt = 0;

    assign LEDON = cnt[26];

endmodule
```

정확한 1초! 2

```
module MyCnt(
    input CLK, //W5
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        if(cnt==100000000)
            cnt = 0;
        else
            cnt = cnt + 1;

    assign LEDON = cnt[26];

endmodule
```

```
module MyCnt(
    input CLK, //W5
    output LEDON //U16
    );

    reg [26:0] Q;
    wire [26:0] D;
```

```
    always @(posedge CLK)
        Q = D;

    assign D = Q + 1;

    assign LEDON = Q[26];

endmodule
```

RST 신호 추가

```
module MyCnt(
    input CLK, //W5
    input RST, //U18, BTNC
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        if(RST==1)
            cnt = 0;
        else
            cnt = cnt + 1;

    assign LEDON = cnt[26];

endmodule
```

1초 상한

```
module MyCnt(
    input CLK, //W5
    input RST, //U18, BTNC
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        if(RST==1)
```

```
            cnt = 0;
        else
        begin
            cnt = cnt + 1;
            if(cnt==100000000)
                cnt = 0;
        end

    assign LEDON = cnt[26];

endmodule
```

```
module MyCnt(
    input CLK, //W5
    input RST, //U18, BTNC
    output LEDON //U16
    );

    reg [26:0] cnt;

    always @(posedge CLK)
        if(RST==1)
            cnt = 0;
        else
        begin
            cnt = cnt + 1;
            if(cnt==100000000)
                cnt = 0;
        end

    assign LEDON = cnt<(100000000/2) ? 1 : 0; // 1Hz, 1:1

endmodule
```

```
assign LEDON = cnt<(100000000/20) ? 1 : 0; // 10Hz, 1:1
```

```
assign LEDON = cnt<(100000000/200) ? 1 : 0; // 100Hz, 1:1
```

```
assign LEDON = (cnt%(100000000/10))<(100000000/10)/10 ? 1 : 0; //10Hz, 1:9
```

```
assign LEDON = cnt%(100000000/100)<(100000000/100)/10 ? 1 : 0; //100Hz, 1:9
```

En 신호 추가

CNT 값 활용 부분 추가 : 클럭 분주기

```
module MyCnt(
    input CLK, //W5
    input RST, //U18, BTNC
    output LEDON //U16
    );

    reg [26:0] cnt;
    wire enTick;
    reg [31:0] cntTick;

    always @(posedge CLK)
        if(RST==1)
            cnt = 0;
        else
        begin
            cnt = cnt + 1;
            if(cnt==100000000)
                cnt = 0;
        end

    assign enTick = cnt%(100000000/1000)==0 ? 1 : 0;

    always @(posedge CLK)
        if(RST==1)
            cntTick = 0;
        else if(enTick==1)
            cntTick = cntTick + 1;

    assign LEDON = cntTick%1000 < 500 ? 1 : 0;

endmodule
```