Digital Lab 3:

Experiment2:

Stepper Motor Control Circuit

Date: 2023/10/12

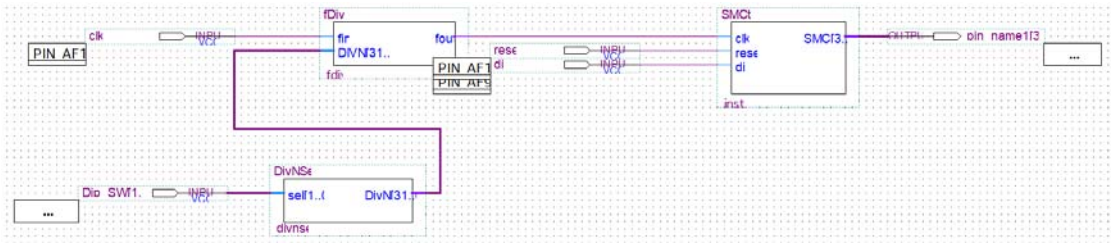
Class: 電機三全英班

Group: Group 11

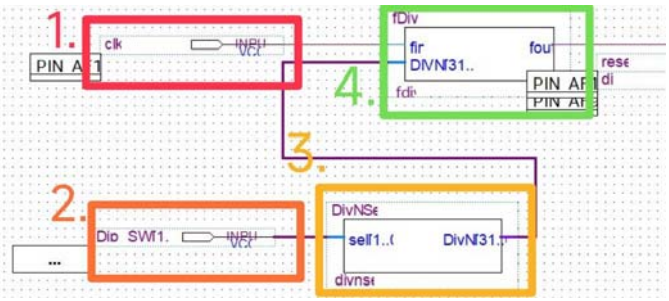
Name: B103105006 胡庭翊

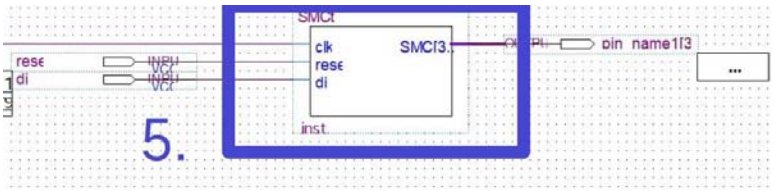
I. Block Diagram

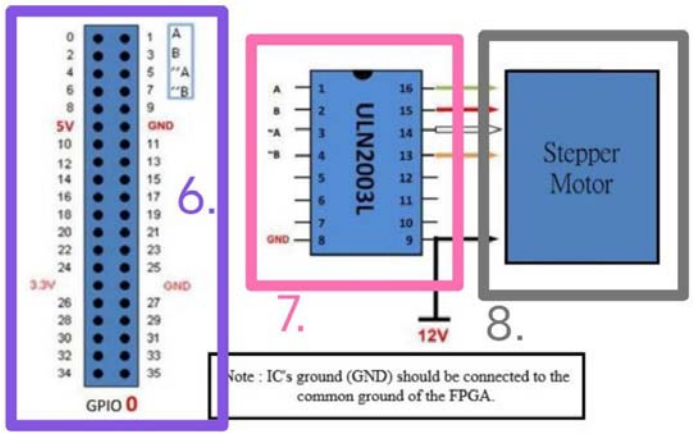
Structure:



Function:







Descriptions:

1. We use 50MHz that set by the board.
2. Use 2 dip switches to be the select of divisor selector.
3. Divisor Selector: Select the specific frequency to be used.
4. Changeable Frequency Divider: Convert the 50MHz input down into specific frequency.

5. State Control Machine: Control the state of Stepper Motor by the State Machine. It can be reset, and changed into inverse direction.
   When reset is 1, the state remains S0. Otherwise, when inv is 0, the states transition as follows, S1→S2→S3→S4. When inv is 1, the sequence reverses, S1→S4→S3→S2. (Refer to Figure 8 for reference.)
6. The output will be connected to GPIO board. It can be viewed as $A, B, \widetilde{A}, \widetilde{B}$.
7. The ULN2003L is an IC composed of seven sets of Darlington transistors, in order to enhance the output of the circuit, and divert the high current of the motor to the ground.
8. Step Motor: When driving the rotor, if two magnetic fields are in a mutually perpendicular structure, due to the law of magnets "like poles repel, unlike poles attract." A force will act on the rotor and cause it to rotate.

II. State Control Machine(SMCrtl)

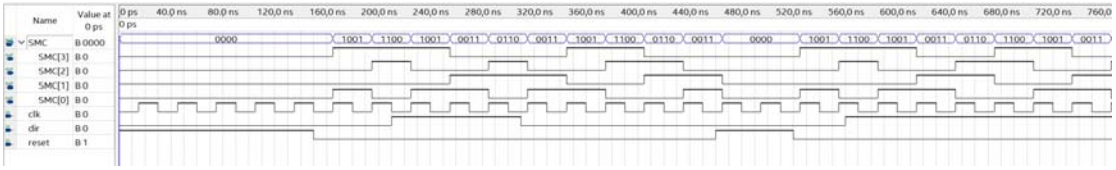
A. Verilog Code and Comment

```
module SMCtrl (clk, reset, dir, SMC);  //state control machine

input clk, reset, dir;  //three input variables. while input clk is the output of fDiv(speed control clock)
output reg[3:0] SMC; //4 bits output of registor SMC
reg[2:0] ns, cs;  //3 bits output of ns and cs

//latch: when clk is posege, always assign ns to cs
always@(posedge clk)
   cs <= ns;

//next stage generator: when the value of reset, dir, or cs changes:
//if reset=1, assign ns into 0 of 3bits in decimo number
//else if cs=0: ns assign to 1;
//      if cs=1: if dir=1, assign ns to 2; if dir=0, assign ns to 4
//      if cs=2: if dir=1, assign ns to 3; if dir=0, assign ns to 1
//      if cs=3: if dir=1, assign ns to 4; if dir=0, assign ns to 2
//      if cs=4: if dir=1, assign ns to 1; if dir=0, assign ns to 3
//      otherwise, assign ns to 0
always@(reset or dir or cs)
   if(reset) ns <= 3'd0;
   else
      case(cs)
      3'd0: ns <= 3'd1;
      3'd1: ns <= (dir)?3'd2:3'd4;
      3'd2: ns <= (dir)?3'd3:3'd1;
      3'd3: ns <= (dir)?3'd4:3'd2;
      3'd4: ns <= (dir)?3'd1:3'd3;
      default: ns <= 3'd0;
      endcase

//output decoder: when the value of cs changes:
//if cs=0: assign the output SMC into 4 bits of binary number 0000
//if cs=1: assign the output SMC into 4 bits of binary number 1001
//if cs=2: assign the output SMC into 4 bits of binary number 0011
//if cs=3: assign the output SMC into 4 bits of binary number 0110
//if cs=4: assign the output SMC into 4 bits of binary number 1100
//otherwise, assign the output SMC into 4 bits of binary number 0000
always@(cs)
   case(cs)
   3'd0: SMC <= 4'b0000;
   3'd1: SMC <= 4'b1001;
   3'd2: SMC <= 4'b0011;
   3'd3: SMC <= 4'b0110;
   3'd4: SMC <= 4'b1100;
   default: SMC <= 4'b0000;
   endcase

endmodule
```

B. Simulation

When reset is set to 1 and the value of clk changes, SMC=0000. Else, SMC will continuing be assigned into specific value, leading the outcome of clockwise rotation or anticlockwise

rotation.



III. Changeable Frequency Divider(fDiv)

A. Verilog Code and Comment

```
module fDiv (fin, DIVN, fout);   //changeable frequency devidor

input fin;  //1 bit input
output fout;   //1 bit output
input [31:0] DIVN;   //32bits of input DIVN
wire [31:0] _DIVN;   //declare 32bits of wire _DIVN

reg[31:0] count; //a 32 bits registor count
reg fout; //1 bit registor fout

assign _DIVN = {1'b0, DIVN[31:1]}; //_DivN equals to DivN devided by 2

always@(posedge fin) //posedge input
begin
    count <= (count>=DIVN)?32'd1:count+1;
    //when count is larger or equal to DivN, set count to 1 in Decimal
    //otherwise count+1
    fout <= (count<=_DIVN)?1'b0:1'b1;
    //when count is lesser than _DivN, set fout to Binary 0;
    //otherwise when it is larger or equal, set fout to binary 1.
end
endmodule
```

B. Simulation

The value of DIVN will be assigned according to the output of divisor selector, and since we set _DIVN to DIVN divided by two, the frequency of the output (fout) would be half of the input (fin).

IV. Divisor selector (DivNSel)

A. Verilog Code and Comment

```
module DivNSel(sel, DivN); //Divisor selector
input [1:0] sel;  //2bits of input sel
output [31:0] DivN;  //32bits of output DivN, it will be sent to frequency divider
reg [31:0] DivN;  //32bits of registor

//when the value of sel changes:
//     if sel=00: assign DivN into 64000 in 32bits decimal number
//     if sel=01: assign DivN into 128000 in 32bits decimal number
//     if sel=10: assign DivN into 256000 in 32bits decimal number
//        else: assign DivN into 512000 in 32bits decimal number
always @(sel)
    case(sel)
        2'b00: DivN <= 32'd512000;
        2'b01: DivN <= 32'd1024000;
        2'b10: DivN <= 32'd2048000;
        default: DivN <= 32'd4096000;
    endcase
endmodule

```

B. Simulation

There are four set frequency: 512000, 1024000, 2048000, and 4096000. As input changes, the output would also be changed and sent to the frequency divider.

V. Reflection

In our recent electrical engineering experiment, we utilized Verilog to design and control a stepper motor. The experiment incorporated components like a Divisor Selector, Changeable Frequency Divider, and a state control machine. With the experience gained from our first attempt, this experiment proceeded much more smoothly.

I realized that preparing the necessary code before class and then implementing it on the board during the lab session could significantly enhance the efficiency of the experiment. This approach can save time and allow for a more focused and productive laboratory experience.