同济大学计算机系

数字逻辑课程实验报告

学　　号 2350752

姓　　名 田思宇

专　　业 计算机科学与技术

授课老师 张冬冬

一、实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 D 触发器、JK 触发器以及 PC 寄存器的设计和仿真。
深入了解 D 触发器、JK 触发器、由 D 触发器构成的 PC 寄存器的原理。 学习使用 Verilog HDL 行为描述语言设计 D 触发器、JK 触发器、D 触发器构成的 PC 寄存器。
由于是第一个时序逻辑实验，我们还要学习如何编写时序逻辑的代码以及如何在 tb 文件中实现时钟信号的周期变化。

二、硬件逻辑图

三、模块建模

（该部分要求对实验中建模的所有模块进行功能描述，并列出各模块建模的 verilog 代码）

实验一：同步复位 D 触发器

功能描述：本模块实现的功能是同步复位 D 触发器，采用行为级语言描述，端口中 clk 是时钟信号，d 是输入，rst_n 是复位信号，q1、q2 代表两个互补的输出，首先用@(posedge CLK)表示所有信号要在时钟上升沿到来的时候有效，然后用 if else 语句实现逻辑判断，首先判断复位信号（因为低电平有效所以是!rst_n），如果是低电平就置 0，然后再判断输入，D=1 则置 1，D=0 则置 0。这里的赋值语句都是采用非阻塞赋值。

```
module Synchronous_D_FF(
input CLK,
input D,
input RST_n,
output reg Q1,
output reg Q2
);
always@(posedge CLK)
begin
if(!RST_n)
begin
Q1<=1'b0;
Q2<=1'b1;
end
else
begin
Q1<=D;
Q2<=~D;
end
end
```

```
endmodule
```

实验二：异步复位 D 触发器

功能描述：本模块实现的功能是异步复位 D 触发器，采用行为级语言描述，端口中 clk 是时钟信号，d 是输入，rst_n 是复位信号，q1、q2 代表两个互补的输出，首先用@(posedge CLK or negedge RST_n)来表示所有信号是 clk 上升沿或者 rst_n 的下降沿来临时有效，这样就可以保证复位信号来临时候就可以复位，无需等待时钟上升沿。然后用 if else 语句实现逻辑判断，首先判断复位信号（因为低电平有效所以是!rst_n），如果是低电平就置 0，然后再判断输入，D=1 则置 1，D=0 则置 0。这里的赋值语句都是采用非阻塞赋值。

```
module Asynchronous_D_FF(
input CLK,
input D,
input RST_n,
output reg Q1,
output reg Q2
);
always@(posedge CLK or negedge RST_n)
begin
if(!RST_n)
begin
Q1<=1'b0;
Q2<=1'b1;
end
else
begin
Q1<=D;
Q2<=~D;
end
end
endmodule
```

实验三：异步复位 JK 触发器

功能描述：本模块实现的是异步复位 JK 触发器，采用行为级语言描述，clk 代表时钟信号，J、K 代表两个输入，rst_n 是低电平有效的异步复位信号，Q1、Q2 是两个互补的输出。首先为了实现异步复位这一功能，使用@(posedge CLK or negedge RST_n)，然后再使用 if else 语句进行逻辑判断，如果复位信号为低电平，则复位 0、1，如果 J=0、K=0，那么就保持目前的 q1、q2；如果 J=1、K=0，那么就置 1；如果 J=0、K=1，那么就置 0；如果 J=1、K=1，那么取反，也就是 q1 与 q2 互换，这时候非阻塞赋值语句（<=）的作用就体现出来了。

```
module JK_FF(
input CLK,
input J,
input K,
input RST_n,
output reg Q1,
```

```
output reg Q2
);
always @(posedge CLK or negedge RST_n)
begin
if(!RST_n)
begin
Q1<=1'b0;
Q2<=1'b1;
end
else if(J==0&&K==0)
begin
Q1<=Q1;
Q2<=Q2;
end
else if(J==1&&K==0)
begin
Q1<=1'b1;
Q2<=1'b0;
end
else if(J==0&&K==1)
begin
Q1<=1'b0;
Q2<=1'b1;
end
else if(J==1&&K==1)
begin
Q1<=Q2;
Q2<=Q1;
end
end
endmodule
```

实验四：PC 寄存器

功能描述：本模块功能是一个 32 位的 pc 寄存器，底层电路原理是 32 个 D 触发器并行使用，本模块采用行为级语言描述，由于复位信号是异步且高电平有效，所以用@(posedge clk or posedge rst)来实现这一功能，因为复位信号不受 ena 使能信号的影响，所以首先判断复位信号，如果是高电平则输出是 32’b0。然后判断 ena 使能信号，如果 ena=1 那么允许 input 输入到寄存器，然后 output 输出，如果 ena=0，那么寄存器里面的值不会发生改变，不允许 input 输入。

```
module pcreg(
input clk,
input rst,
input ena,
input [31:0] data_in,
```

```
output reg [31:0] data_out
);
always@(posedge clk or posedge rst)
begin
if(rst)
begin
data_out<=32'b0;
end
else if(ena==1)
begin
data_out<=data_in;
end
else
begin
data_out<=data_out;
end
end
endmodule
```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

实验一：同步复位 D 触发器

```
`timescale 1ns/1ns
module d_fe_tb;
reg clk,d,rst;
wire q1,q2;
initial clk=0;
always #20 clk=~clk;
initial
begin
rst=0;
d=0;
#40//??????
rst=1;
d=1;
#40
d=0;
#40
rst=0;
end
Synchronous_D_FF uut(clk,d,rst,q1,q2);
endmodule
```

实验二：异步复位 D 触发器

```
`timescale 1ns/1ns
module d_fe_tb;
reg clk,d,rst;
wire q1,q2;
initial clk=0;
always #20 clk=~clk;
initial
begin
rst=0;
d=0;
#40//??????
rst=1;
d=1;
#40
d=0;
#40
rst=0;
end
Asynchronous_D_FF uut(clk,d,rst,q1,q2);
endmodule
```

实验三：异步复位 JK 触发器

```
`timescale 1ns/1ns
module jk_tb;
reg clk,j,k,rst;
wire q1,q2;
initial clk=0;
always #20 clk=~clk;
initial
begin
rst=0;
j=1;
k=0;
#40//??????
rst=1;
j=1;
k=0;
#40
j=0;
k=1;
#40
j=1;
```

```
k=1;
end
JK_FF uut(clk,j,k,rst,q1,q2);
endmodule
```

实验四：PC 寄存器

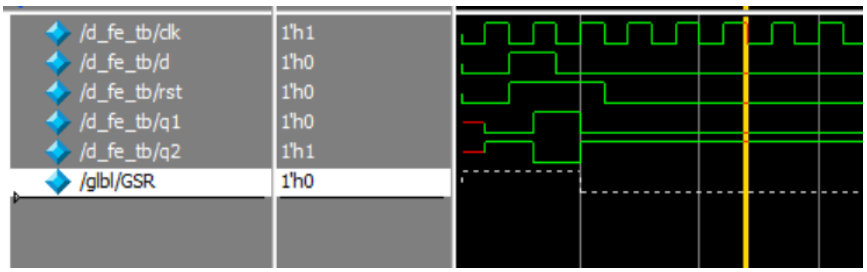
```
`timescale 1ns/1ns
module pcreg_tb;
reg clk,rst,ena;
reg [31:0] idata;
wire [31:0] odata;
initial clk=0;
always #20 clk=~clk;
initial
begin
ena=0;
rst=0;
idata=32'b0;
#40//??????
ena=1;
rst=1;
idata=32'b10101010101000000000000000000000;
#40
idata=32'b0;
#40
rst=0;
end
pcreg uut(clk,rst,ena,idata,odata);
endmodule
```

五、实验结果

（该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图））

实验一：

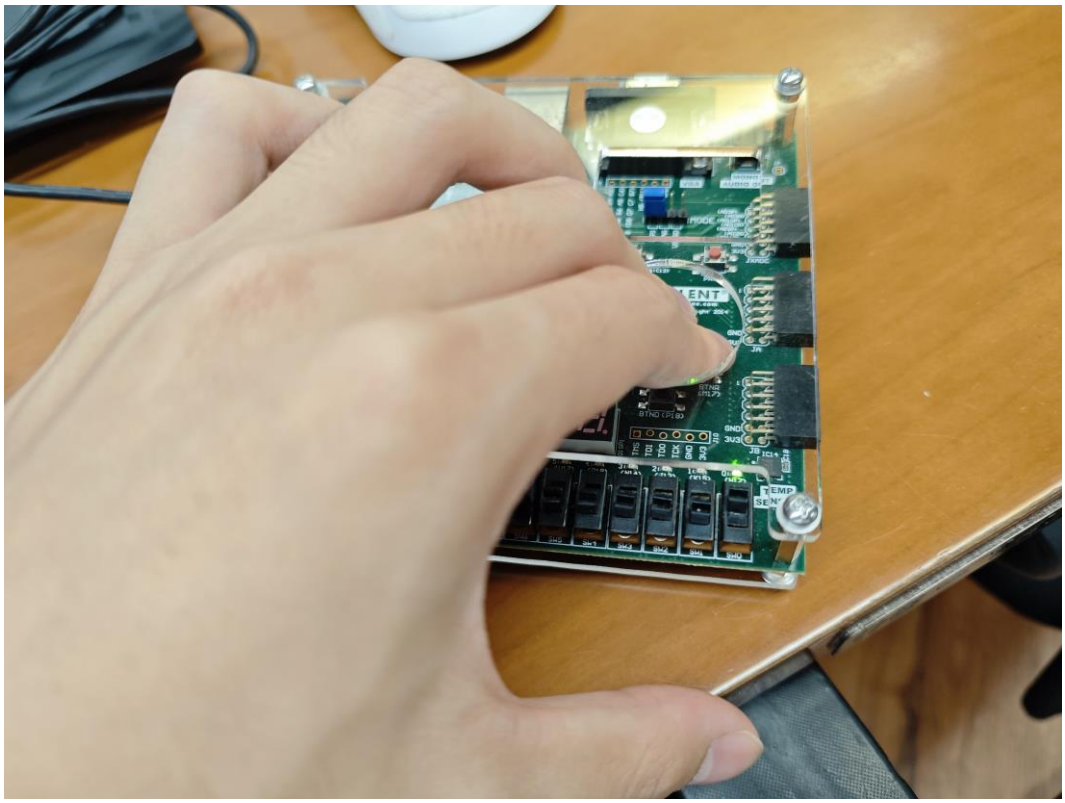
modelsim 仿真波形图



下板结果：

1、 刚下板时候，由于是同步复位，所以先按一下时钟信号按钮，让信号复位，从图片上可以看出，虽然 D=1，但是复位信号低电平（有效），所以置 0。

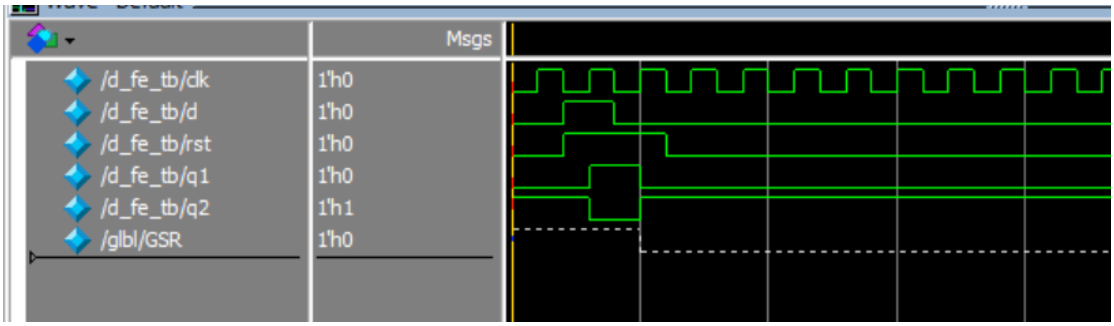
2、 这时候让复位信号为高电平，D=1，clk 上升沿信号来临，可以看到置 1 了



3、然后再将 D=0，让复位信号为高电平，clk 上升沿来临时，置 0 了。

实验二：

modelsim 仿真波形图

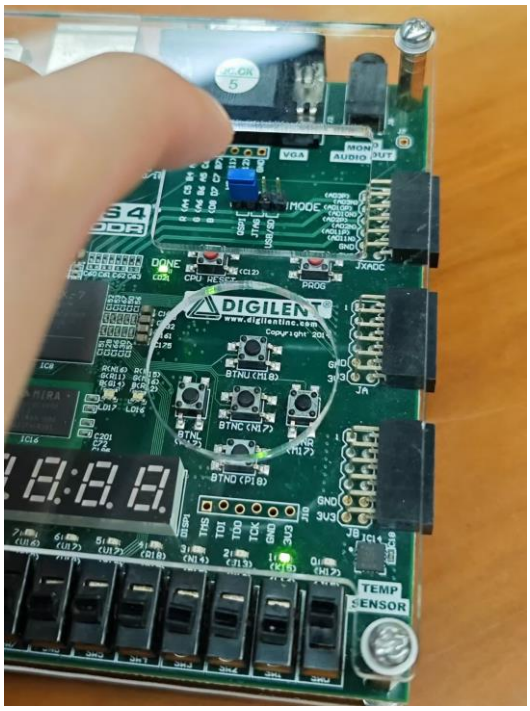


下板结果

1、这是刚下板时候的结果，由于是异步复位并且复位信号是低电平有效，所以刚下板就自动复位，置 0.

2、这时候让 D=1，复位信号高电平（无效），当 clk 上升沿来临时，置 1.

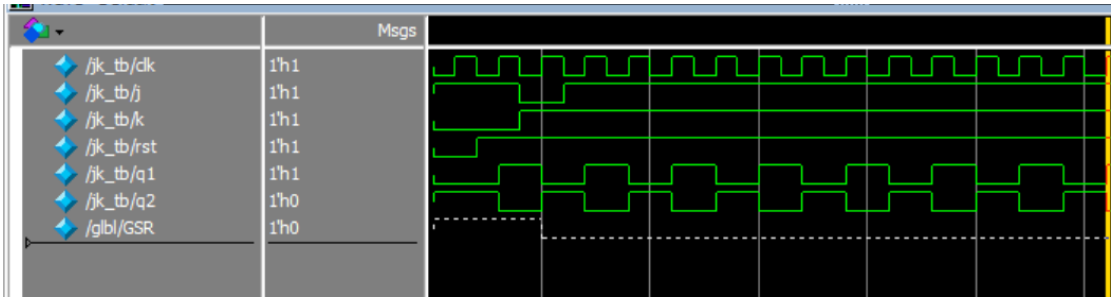
3、这时候松开复位信号的按钮，由于异步复位，所以又复位置 0 了。



4、可以看到就算 clk 来临且 d=1，也不会置 1，因为这时候复位信号始终有效

实验三：

modelsim 仿真波形图



下板结果

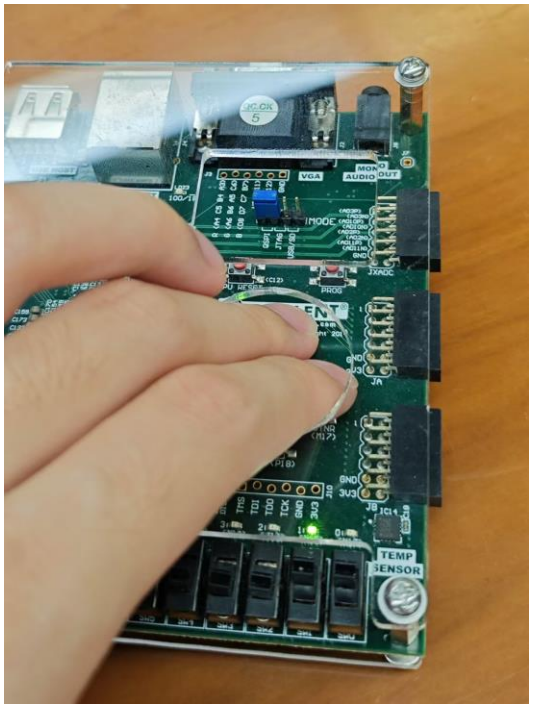
1、首先由于复位信号是低电平有效，所以下板之后就自动复位了，置 0.

2、这时候 J=1,K=0,复位信号高电平（无效），时钟来临时置 1，如图所示。

3、当 J=0,K=1 时候，并且复位信号无效，置 0

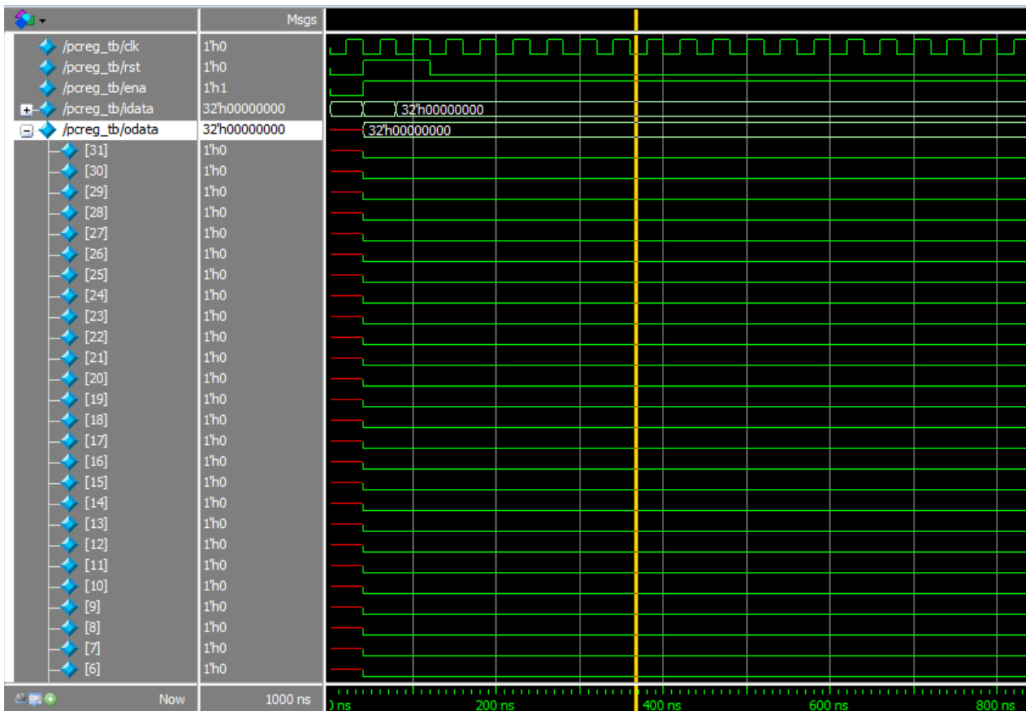
4、紧接着让 J=1，K=1，复位信号无效，时钟信号来临时候，输出信号反转，如图所示变成了 q1=1，q2=0；

5、继续让 J=1，K=1，复位信号无效，时钟信号来临时候，输出信号反转，如图所示变成了 q1=0，q2=1；



实验四：

modelsim 仿真波形图


Msgs
/pcreg_tb/clk
/pcreg_tb/rst
/pcreg_tb/ena
/pcreg_tb/idata
/pcreg_tb/odata
[31]
[30]
[29]
[28]
[27]
[26]
[25]
[24]
[23]
[22]
[21]
[20]
[19]
[18]
[17]
[16]
[15]
[14]
[13]
[12]
[11]
[10]
[9]
[8]
[7]
[6]
1'h0
1'h1
32'h00000000
Now
1000 ns
200 ns
400 ns
600 ns
800 ns