



**中国科学院大学**  
University of Chinese Academy of Sciences

# **数 字 电 路**

## **实验报告**

**班级：**教 221

**组号：**----

**姓名：**唐嘉良

**学号：**2020K8009907032

**实验名称：**FIFO 实验

2021 年 12 月 9 日

# 一、实验目的

- 1、 熟悉 verilog 编程、调试
- 2、 熟悉 FIFO 工作原理
- 3、 实现功能较复杂的数字电路

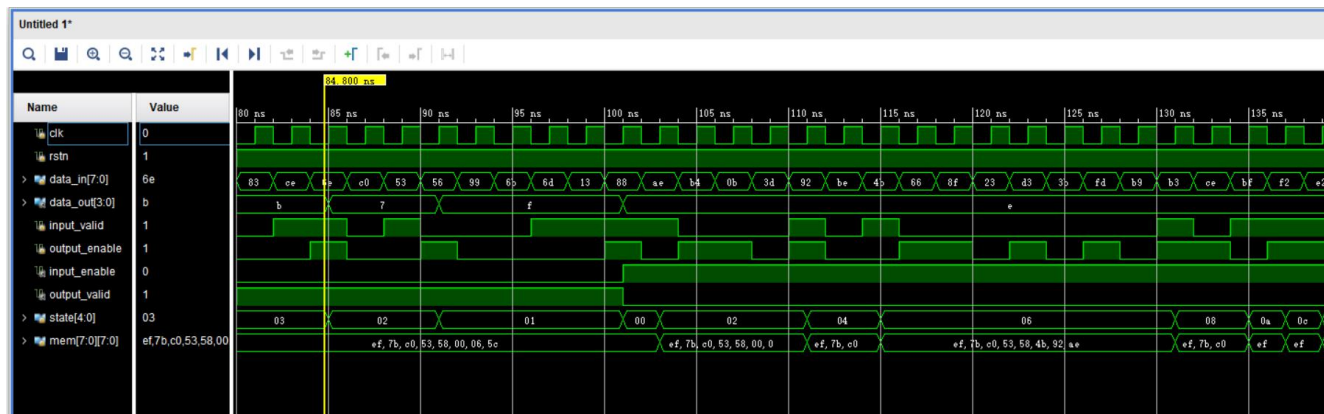
# 二、实验环境

本次实验我采用的是 vivado 2017.4 版本。

# 三、调试过程

FIFO





## 四、实验总结

在此次实验中，我更加熟悉 vivado 平台的操作流程，现在能够创建激励文件并进行调试。同时，通过构建同步 FIFO，我对 Verilog 语言的掌握程度大大提升，能够更加熟练自如地对数据进行基本的读写操作。

## 五、源代码

### FIFO

#### 源代码

```
module FIFO(
    input [0:0] clk,
    input [0:0] rstn,
    input input_valid,
    input output_enable,
    output reg input_enable,
    output reg output_valid,
    input [7:0] data_in,
    output reg [3:0] data_out
);
    reg [4:0] state;
    reg [7:0] mem [7:0];
    reg [2:0] write_addr;
    reg [3:0] read_addr;

    always @(posedge clk or negedge rstn)begin
        if(rstn == 0)begin
            write_addr <= 3'b0;
            input_enable<=1'b1;
            state<=0;
            output_valid<=1'b0;
            read_addr <=4'b0;
        end
    end
```

```

else if(rstn)
if(input_valid==1 && input_enable==1)
begin
if(state==16)
begin
end
else
begin
mem[write_addr][7:0]<=data_in;
state<=state+2;
write_addr<=write_addr+1;
end

end

end

always@(posedge clk or negedge rstn)begin
if((state==16      &&      input_valid==1      &&
input_enable==1) || (output_valid==1 && output_enable==1))
begin
if(read_addr[0]==0)begin
data_out<=mem[read_addr>>1][3:0];
read_addr<=read_addr+1;
state<=state-1;
end
else if(read_addr[0]==1)begin
data_out<=mem[read_addr>>1][7:4];
read_addr<=read_addr+1;
state<=state-1;
end
end

end

always@(posedge clk or negedge rstn)begin
if(input_valid==1&&input_enable==1)
begin
end
else if(output_valid==1&&output_enable==1)
begin
end
else
state<=state;
end

```

```

always@(posedge clk or negedge rstn)
if(state==16&&input_enable==1)
begin
output_valid<=1;
input_enable<=0;
end

always@(posedge clk or negedge rstn)
begin
if(state==1&&output_valid==1&&output_enable==1)
begin input_enable<=1;
output_valid<=0;
end
end
endmodule

```

## 激励文件

```

module test_FIFO(

);
reg clk,rstn;
reg [7:0] data_in;
wire [3:0] data_out;
reg input_valid;
reg output_enable;
wire input_enable;
wire output_valid;

FIFO inst_FIFO(
clk,
rstn,
input_valid,
output_enable,
input_enable,
output_valid,
data_in,
data_out);

always begin
#1 clk=~clk;
end

initial begin
clk=0;

```

```
end

initial begin
    #2 rstn=0;
    #2 rstn=1;
end

always begin
    #2;
    data_in<=$random()%255;
    input_valid<=$random()%2;
    output_enable<=$random()%2;
end

endmodule
```