中国科学技术大学计算机学院
《计算机组成原理》实验报告

实验题目：寄存器堆与存储器及其应用

学生姓名：钟书锐

学生学号：PB19000362

完成日期：2021.4.7

计算机实验教学中心制

2020 年 09 月

【实验题目】

LabH2_寄存器堆与存储器及其应用

【实验目的】

1.掌握 Vivado 的使用

2.熟悉 Verilog 的语法

4.了解如何通过 Vivado 产生电路图

5.熟悉综合、仿真与下载的步骤，掌握 Verilog 代码的调试

6.实现寄存器堆

7.实现 FIFO 队列

【实验环境】

硬件:

处理器：i7-10750H @ 2.60GHz 六核

显卡：RTX2060(6GB)

Nexys4DDR 开发板

操作系统:

WINDOWS10 家庭中文版

软件:

Vivado 2020.1

【实验内容】

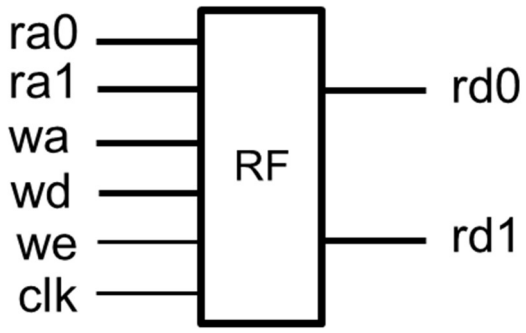
1. **寄存器堆（Register File）**

   a) clk：时钟

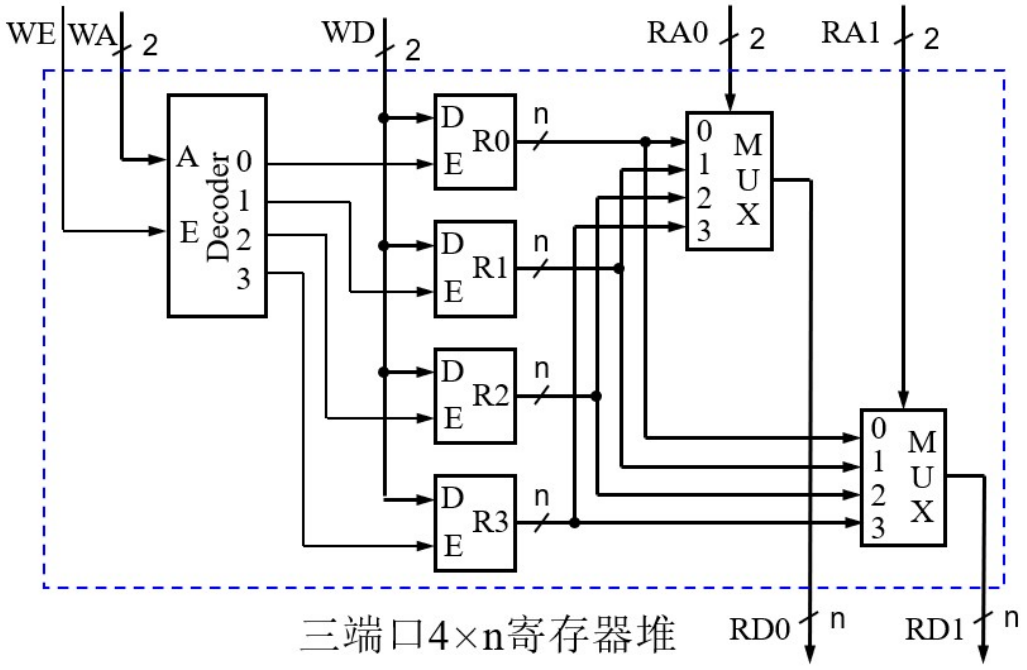
   b) ra0, rd0：异步读端口 0

   c) ra1, rd1：异步读端口 1

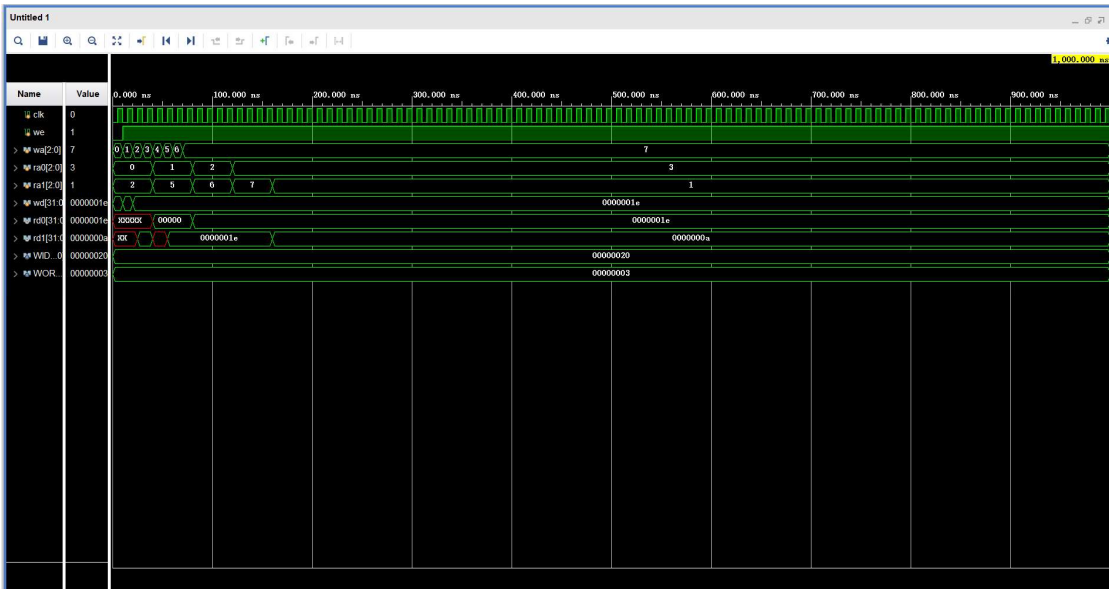
   d) wa, wd, we：同步写端口



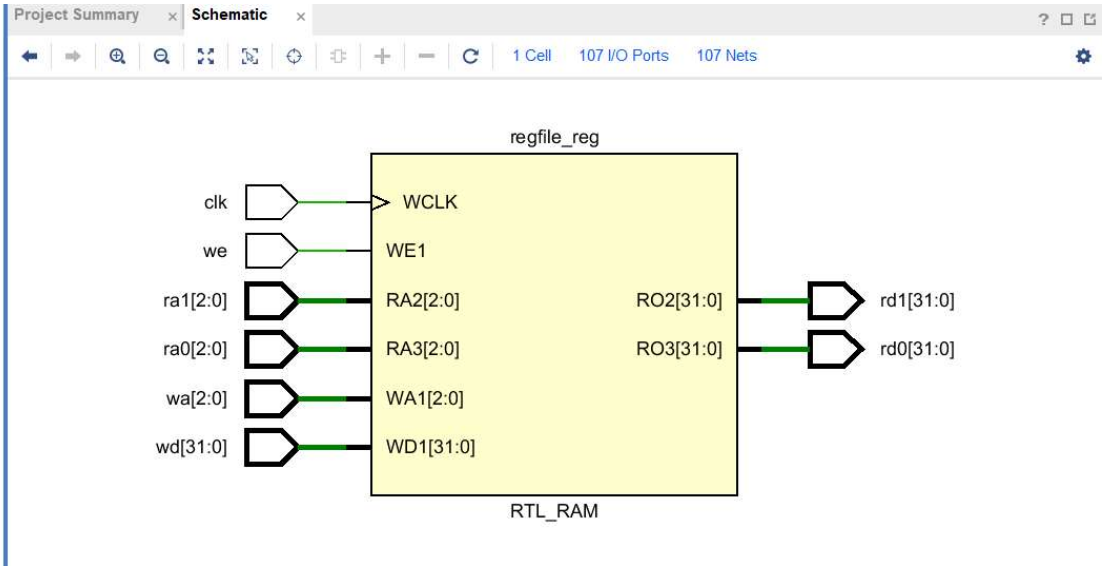
数据通路



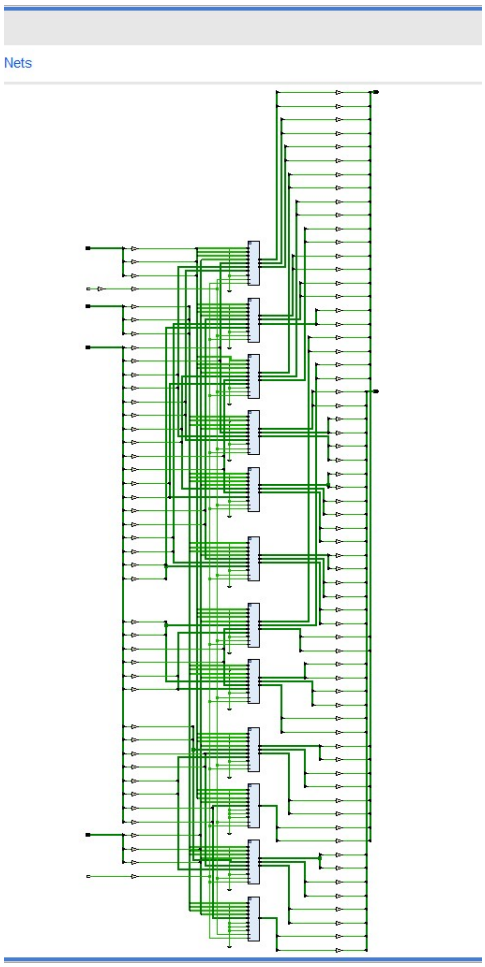
仿真结果


Untitled 1
Name
Value
clk
we
wa[2:0]
ra0[2:0]
ra1[2:0]
wd[31:0]
rd0[31:0]
rd1[31:0]
WID...
WOR...
0.000 ns
100.000 ns
200.000 ns
300.000 ns
400.000 ns
500.000 ns
600.000 ns
700.000 ns
800.000 ns
900.000 ns
1,000.000 ns

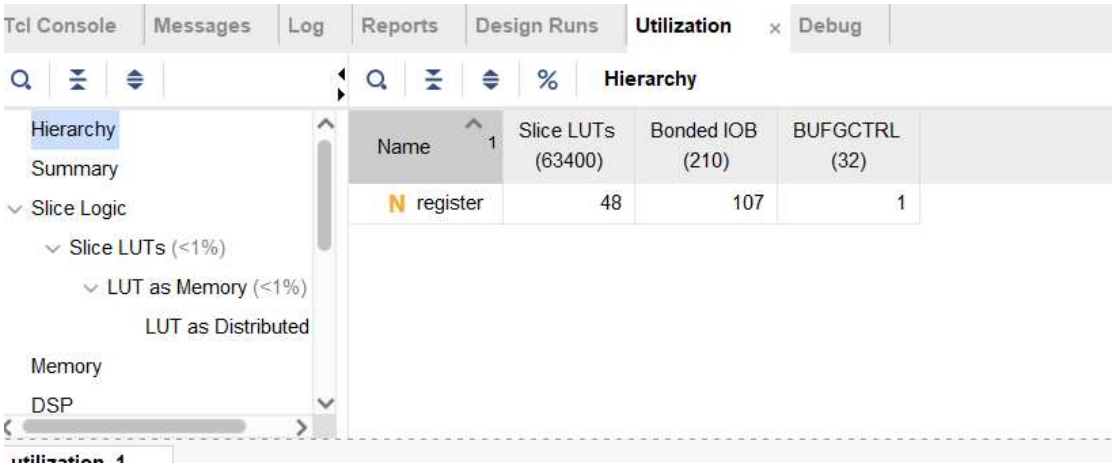
RTL 电路


Project Summary
Schematic
1 Cell
107 I/O Ports
107 Nets
regfile_reg
clk
we
ra1[2:0]
ra0[2:0]
wa[2:0]
wd[31:0]
WCLK
WE1
RA2[2:0]
RA3[2:0]
WA1[2:0]
WD1[31:0]
RO2[31:0]
RO3[31:0]
rd1[31:0]
rd0[31:0]
RTL_RAM

Synth 电路



资源占用



核心代码

```
module register
#(parameter WIDTH = 32,parameter WORD_LINE = 3)
(
input wire clk,
input wire we,
input wire [WORD_LINE-1:0] wa, ra0, ra1,
input wire [WIDTH-1:0] wd,
output wire [WIDTH-1:0] rd0, rd1
    );
reg  [WIDTH-1:0] regfile [0:(1 << WORD_LINE) - 1];
assign  rd0 = regfile [ra0],
        rd1 = regfile [ra1];
always @(posedge clk)
begin
        if (we)
            regfile [wa]  <=  wd;
end

endmodule
```

2. IP 核

Distributed

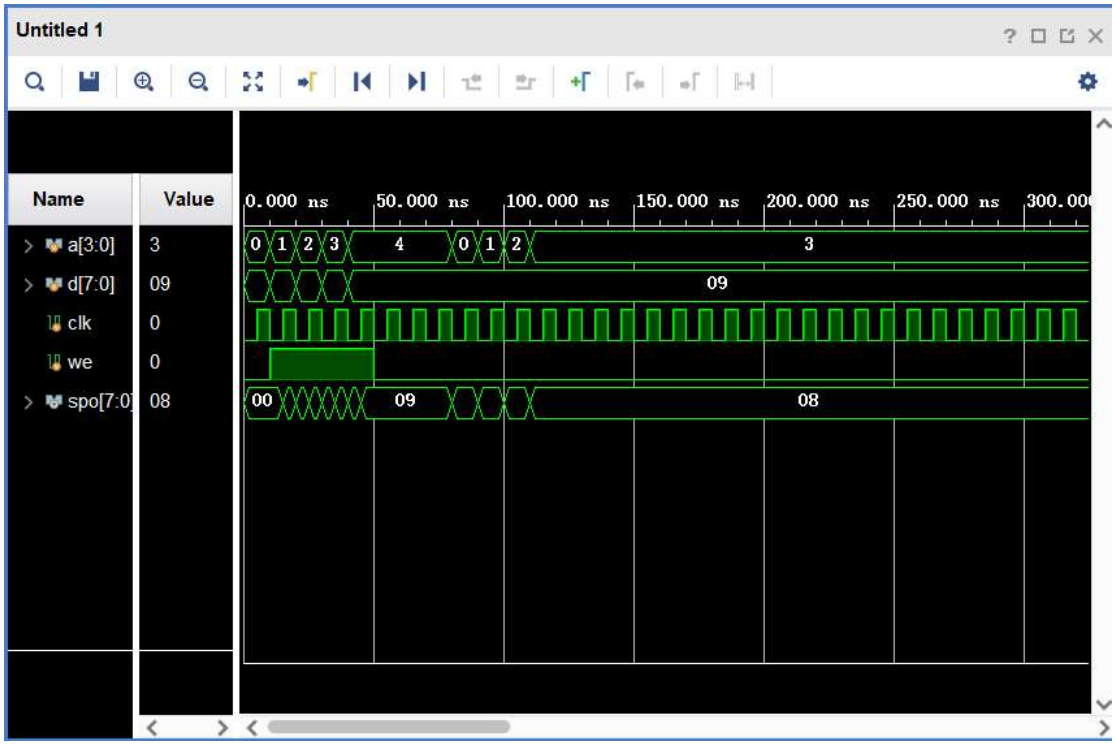
```
module IP_distributed(
    input wire [3 : 0] a,
    input wire [7 : 0] d,
    input wire clk,
    input wire we,
    output wire [7 : 0] spo
    );
dist_mem_gen_0   my_dist_mem (
  .a(a),
  .d(d),
  .clk(clk),
  .we(we),
  .spo(spo)
);
endmodule
```

BLOCK

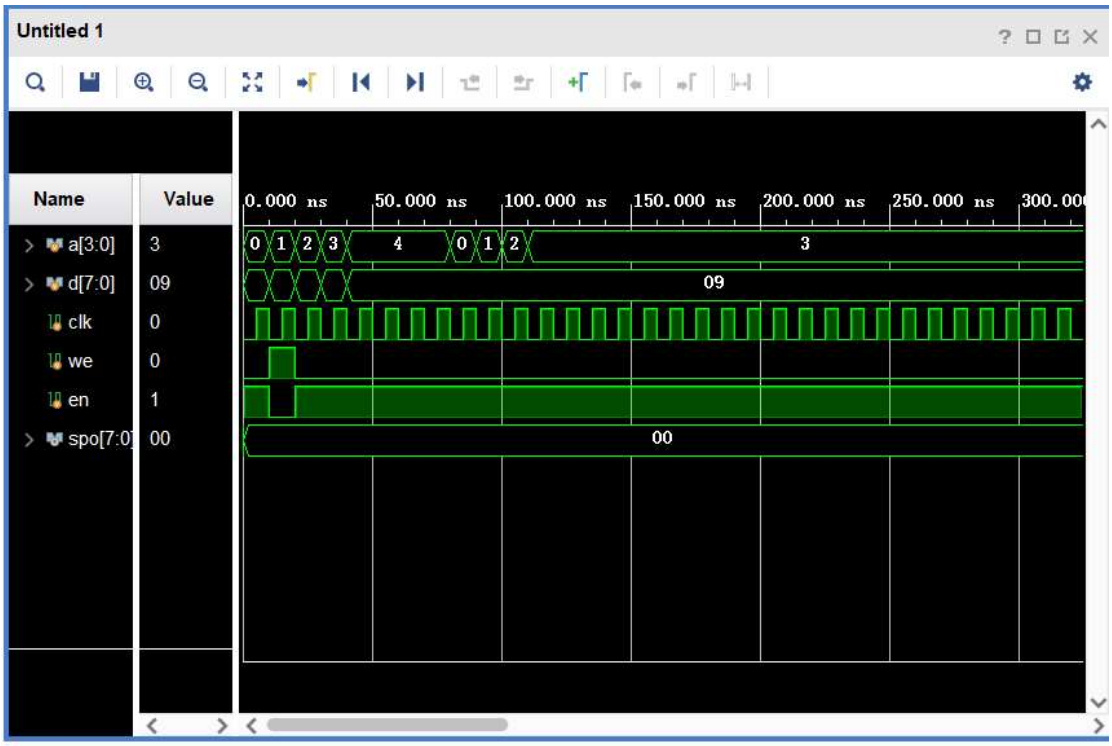
```
module IP_BLOCK(
    input wire [3 : 0] a,
    input wire [7 : 0] d,
    input wire clk,
    input wire we,
    input wire en,
    output wire [7 : 0] spo
    );
blk_mem_gen_0   my_blk_mem (.addra(a),.clka(clk),.dina(d),.douta(spo),.ena(en),.wea(we));
endmodule
```

仿真结果

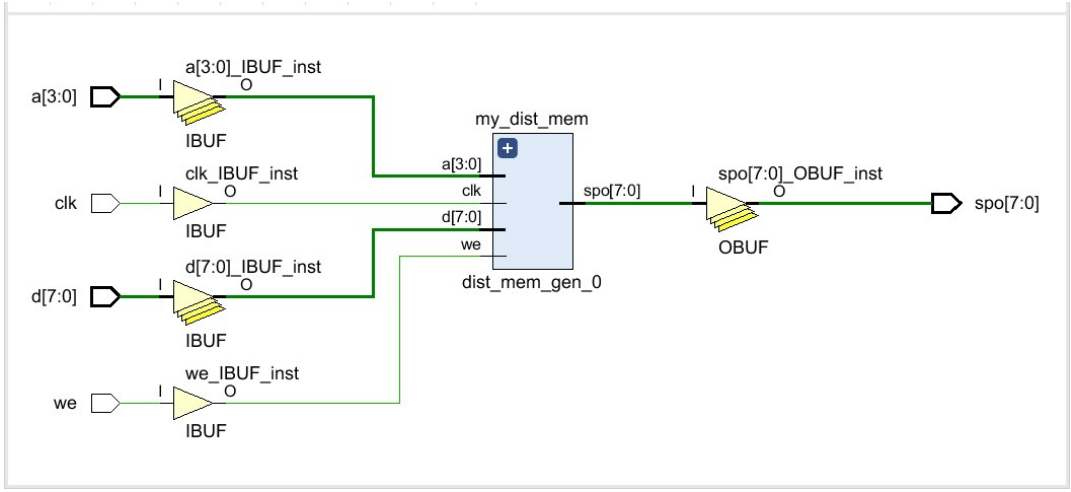
Distributed

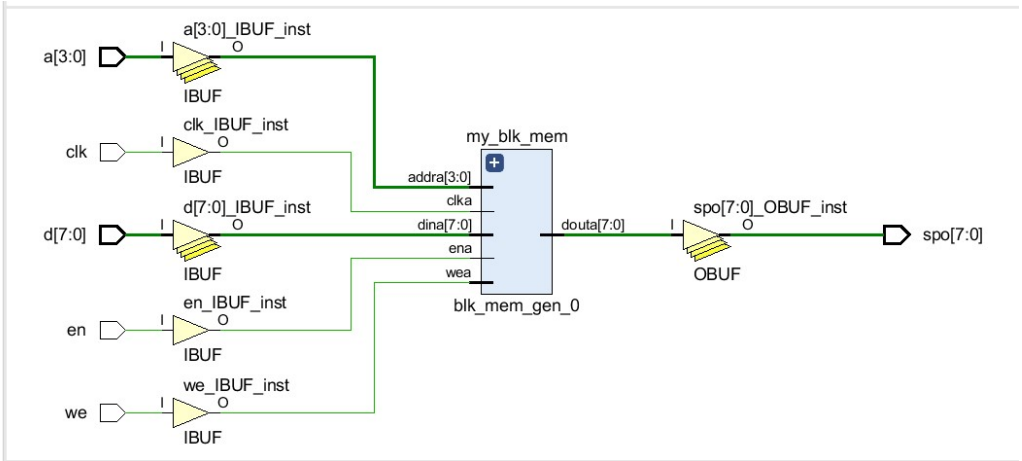


BLOCK

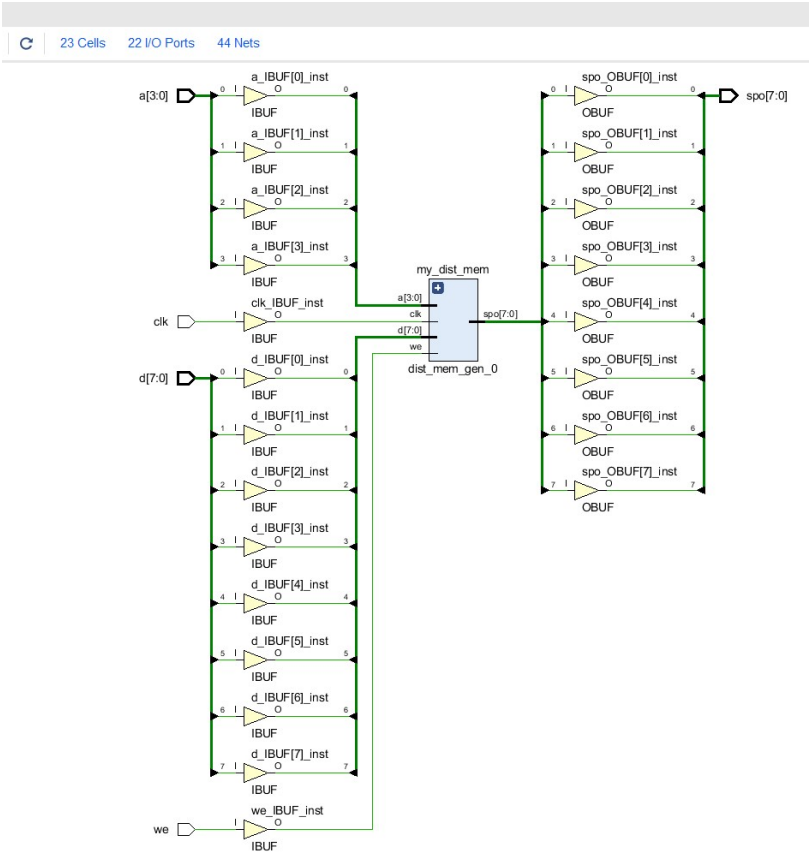

Untitled 1
Name
Value
a[3:0]
3
d[7:0]
09
clk
0
we
0
en
1
spo[7:0]
00
0.000 ns
50.000 ns
100.000 ns
150.000 ns
200.000 ns
250.000 ns
0 1 2 3 4 0 1 2 3
09
00

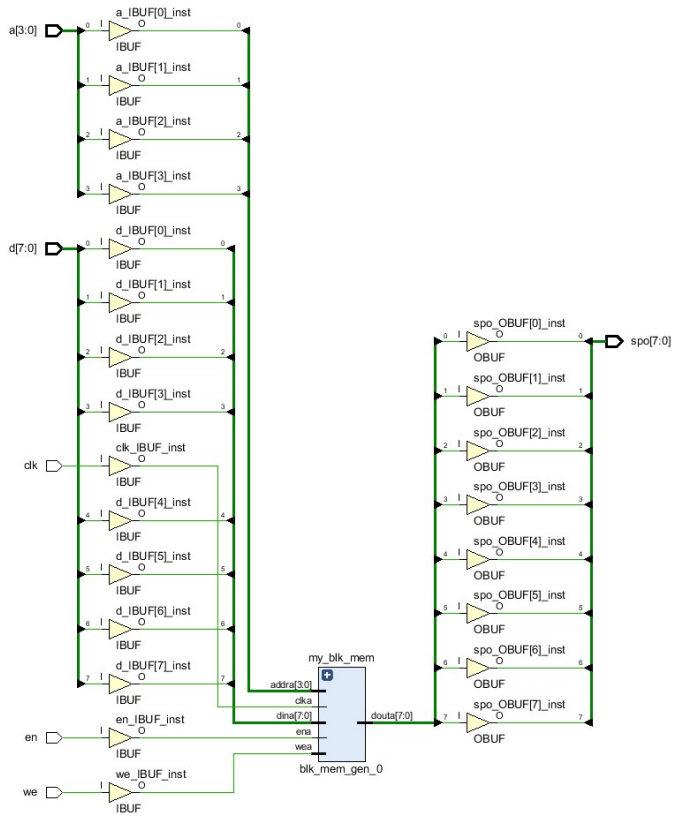
RTL 电路


a[3:0]
a[3:0]_IBUF_inst
IBUF
clk
clk_IBUF_inst
IBUF
d[7:0]
d[7:0]_IBUF_inst
IBUF
we
we_IBUF_inst
IBUF
my_dist_mem
a[3:0]
clk
d[7:0]
we
spo[7:0]
dist_mem_gen_0
spo[7:0]_OBUF_inst
OBUF
spo[7:0]


a[3:0]
a[3:0]_IBUF_inst
IBUF
clk
clk_IBUF_inst
IBUF
d[7:0]
d[7:0]_IBUF_inst
IBUF
en
en_IBUF_inst
IBUF
we
we_IBUF_inst
IBUF
my_blk_mem
addra[3:0]
clka
dina[7:0]
ena
wea
douta[7:0]
blk_mem_gen_0
spo[7:0]_OBUF_inst
OBUF
spo[7:0]

Synth 电路


23 Cells
22 I/O Ports
44 Nets
a[3:0]
a_IBUF[0]_inst
a_IBUF[1]_inst
a_IBUF[2]_inst
a_IBUF[3]_inst
IBUF
clk
clk_IBUF_inst
d[7:0]
d_IBUF[0]_inst
d_IBUF[1]_inst
d_IBUF[2]_inst
d_IBUF[3]_inst
d_IBUF[4]_inst
d_IBUF[5]_inst
d_IBUF[6]_inst
d_IBUF[7]_inst
we
we_IBUF_inst
my_dist_mem
a[3:0]
clk
d[7:0]
we
spo[7:0]
dist_mem_gen_0
spo_OBUF[0]_inst
spo_OBUF[1]_inst
spo_OBUF[2]_inst
spo_OBUF[3]_inst
spo_OBUF[4]_inst
spo_OBUF[5]_inst
spo_OBUF[6]_inst
spo_OBUF[7]_inst
OBUF
spo[7:0]


a[3:0]
a_IBUF[0]_inst
a_IBUF[1]_inst
a_IBUF[2]_inst
a_IBUF[3]_inst
IBUF
d[7:0]
d_IBUF[0]_inst
d_IBUF[1]_inst
d_IBUF[2]_inst
d_IBUF[3]_inst
clk
clk_IBUF_inst
d_IBUF[4]_inst
d_IBUF[5]_inst
d_IBUF[6]_inst
d_IBUF[7]_inst
en
en_IBUF_inst
we
we_IBUF_inst
my_blk_mem
addra[3:0]
clka
dina[7:0]
ena
wea
douta[7:0]
blk_mem_gen_0
spo_OBUF[0]_inst
spo_OBUF[1]_inst
spo_OBUF[2]_inst
spo_OBUF[3]_inst
spo_OBUF[4]_inst
spo_OBUF[5]_inst
spo_OBUF[6]_inst
spo_OBUF[7]_inst
OBUF
spo[7:0]

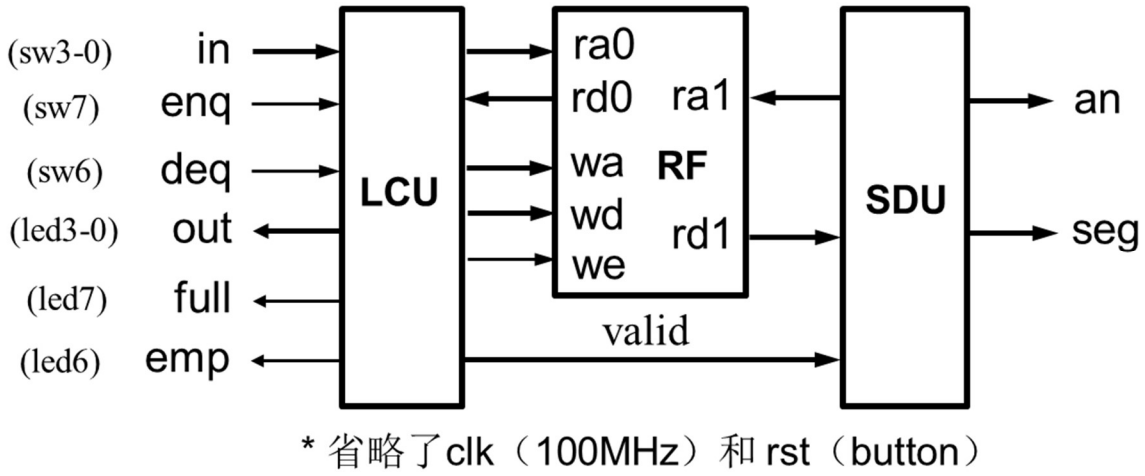
资源占用

Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)
IP_distributed	8	8	22
my_dist_mem (dist_mem_gen_0)	8	8	0

Name	Block RAM Tile (135)	Bonded IOB (210)
IP_BLOCK	0.5	23
my_blk_mem (blk_mem_gen_0)	0.5	0

3. **FIFO**

数据通路



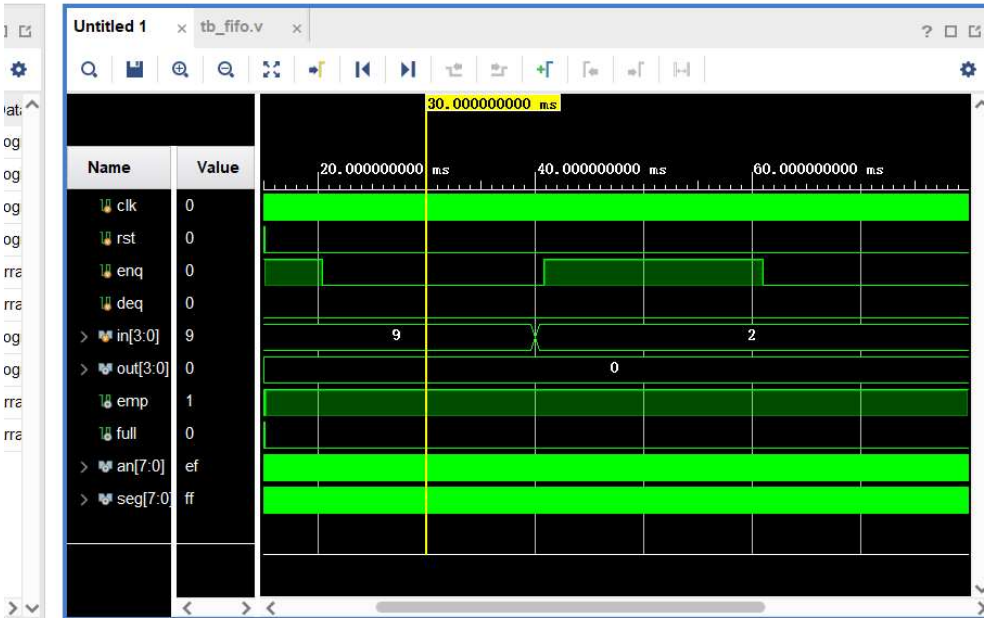
逻辑设计

LCU:逻辑设计单元

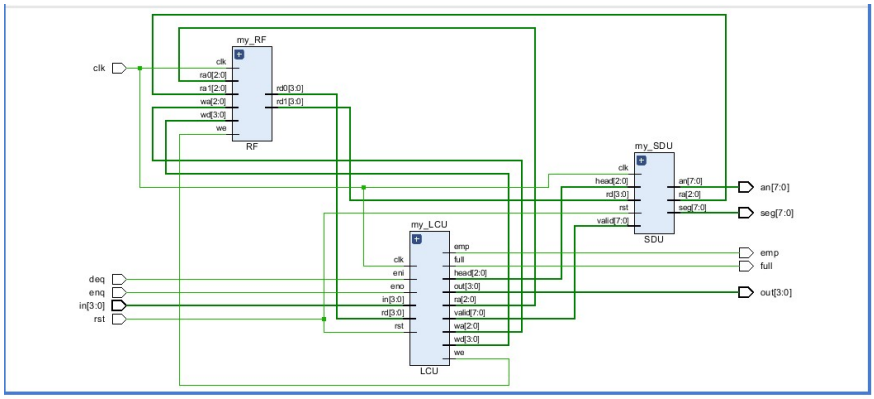
RF:寄存器堆

SDU:显示单元

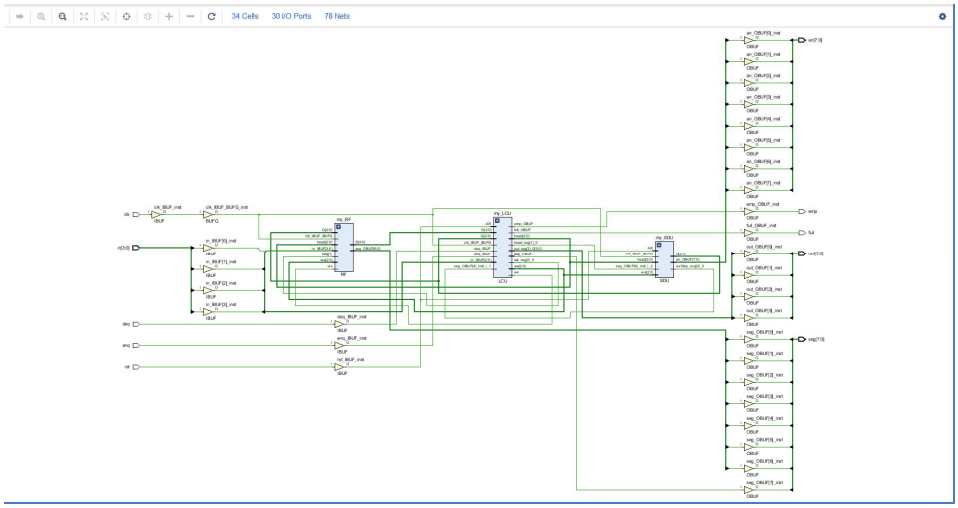
仿真结果



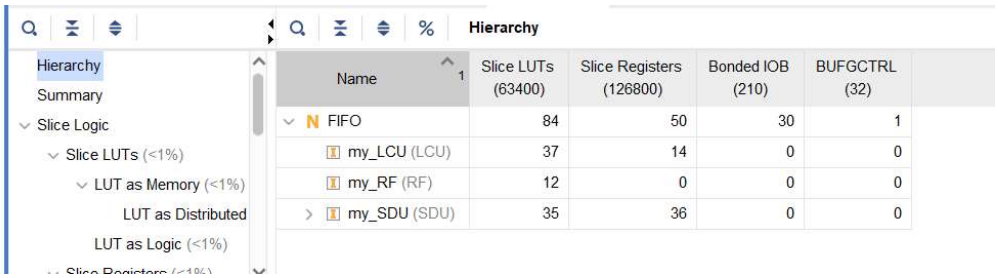
RTL 电路



Synth 电路



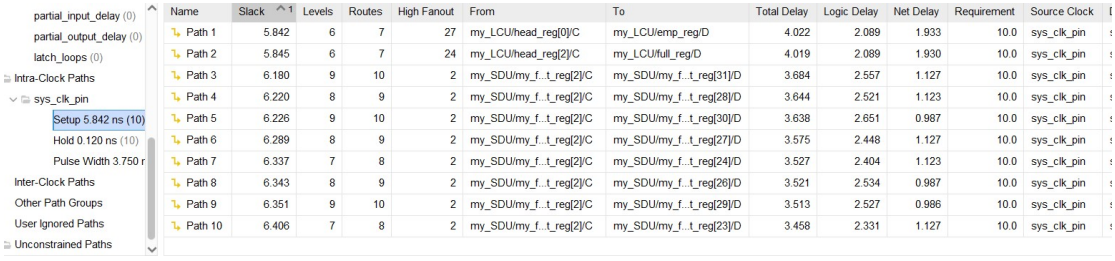
资源占用



Hierarchy

Hierarchy	Name	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
Summary	N FIFO	84	50	30	1
Slice Logic	my_LCU (LCU)	37	14	0	0
Slice LUTs (<1%)	my_RF (RF)	12	0	0	0
LUT as Memory (<1%)	my_SDU (SDU)	35	36	0	0
LUT as Distributed					
LUT as Logic (<1%)					

时间性能分析



partial_input_delay (0)
partial_output_delay (0)
latch_loops (0)
Intra-Clock Paths
sys_clk_pin
Setup 5.842 ns (10)
Hold 0.120 ns (10)
Pulse Width 3.750 r
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	5.842	6	7	27	my_LCU/head_reg[0]/C	my_LCU/emp_reg/D	4.022	2.089	1.933	10.0	sys_clk_pin
Path 2	5.845	6	7	24	my_LCU/head_reg[2]/C	my_LCU/full_reg/D	4.019	2.089	1.930	10.0	sys_clk_pin
Path 3	6.180	9	10	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[31]/D	3.684	2.557	1.127	10.0	sys_clk_pin
Path 4	6.220	8	9	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[28]/D	3.644	2.521	1.123	10.0	sys_clk_pin
Path 5	6.226	9	10	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[30]/D	3.638	2.651	0.987	10.0	sys_clk_pin
Path 6	6.289	8	9	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[27]/D	3.575	2.448	1.127	10.0	sys_clk_pin
Path 7	6.337	7	8	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[24]/D	3.527	2.404	1.123	10.0	sys_clk_pin
Path 8	6.343	8	9	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[26]/D	3.521	2.534	0.987	10.0	sys_clk_pin
Path 9	6.351	9	10	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[29]/D	3.513	2.527	0.986	10.0	sys_clk_pin
Path 10	6.406	7	8	2	my_SDU/my_f...t_reg[2]/C	my_SDU/my_f...t_reg[23]/D	3.458	2.331	1.127	10.0	sys_clk_pin

核心代码

主模块

```
module FIFO(
    input clk, rst,  //时钟（上升沿有效）、同步复位（高电平有效）
    input enq,       //入队列使能，高电平有效
    input [3:0] in, //入队列数据
    input deq,       //出队列使能，高电平有效
    output [3:0] out,   //出队列数据
    output [7:0] an,    //数码管选择
    output [7:0] seg,   //数码管数据
    output wire emp,    //空队标志
    output wire full    //满队标志
    );

    wire [2:0] ra0;
    wire [3:0] rd0;
    wire [2:0] ra1;
    wire [3:0] rd1;
    wire [2:0] wa;
    wire [3:0] wd;
    wire we;
    wire [7:0] valid;
    wire [2:0] head;

    LCU my_LCU(clk, rst, enq , deq , in, rd0, ra0, we, wa, wd, out, emp, full, head, valid);
    RF #(4,3) my_RF(clk, we, wa, ra0, ra1, wd, rd0, rd1);
    SDU my_SDU(clk, rst, head, valid, rd1, ra1, an, seg);

endmodule
```

控制模块

```
    always @ (posedge clk, posedge rst) begin
        if (rst) prevEno <= 0;
        else if (clk) prevEno <= eno;
    end
    assign enoPosedge = (~prevEno & eno) ? 1 : 0;

    reg prevEni;
    wire eniPosedge;
    always @ (posedge clk, posedge rst) begin
        if (rst) prevEni <= 0;
        else if (clk) prevEni <= eni;
    end
    assign eniPosedge = (~prevEni & eni) ? 1 : 0;

    reg [2:0] tail;
    always @ (posedge clk, posedge rst) begin
        if (rst) tail <= 7;
        else if (clk) begin
            if (eniPosedge && !full && in <= 9) tail <= tail - 1;
        end
    end

    always @ (posedge clk, posedge rst) begin
        if (rst) head <= 7;
        else if (clk) begin
            if (enoPosedge && !emp) head <= head - 1;
        end
    end
```

```
always @ (posedge clk, posedge rst) begin
    if (rst) begin
        emp <= 1;
        full <= 0;
    end
    else if (clk) begin
        if (enoPosedge && (head - 1 == tail || tail == 7 && head == 0)) emp <= 1;
        else if (eniPosedge && in <= 9) emp <= 0;

        if (eniPosedge && (tail - 1 == head || head == 7 && tail == 0) && in <= 9) full <= 1;
        else if (enoPosedge) full <= 0;
    end
end

assign ra = head;
always @ (posedge clk, posedge rst) begin
    if (rst) out <= 0;
    else if (clk) begin
        if (enoPosedge && !emp) out <= rd;
    end
end
```

【结果分析】

能够实现出队入队

设计为循环队列

能够显示空队与满队

【实验总结】

复习了 VERILOG 语法与 vivado 使用

掌握时序逻辑与组合逻辑的基本实现

掌握了数码管显示的方法

【实验建议】

建议多提前几天放出题目