“计算机设计与实践”处理器实验设计报告

姓名：万晓珑

班级：1403104

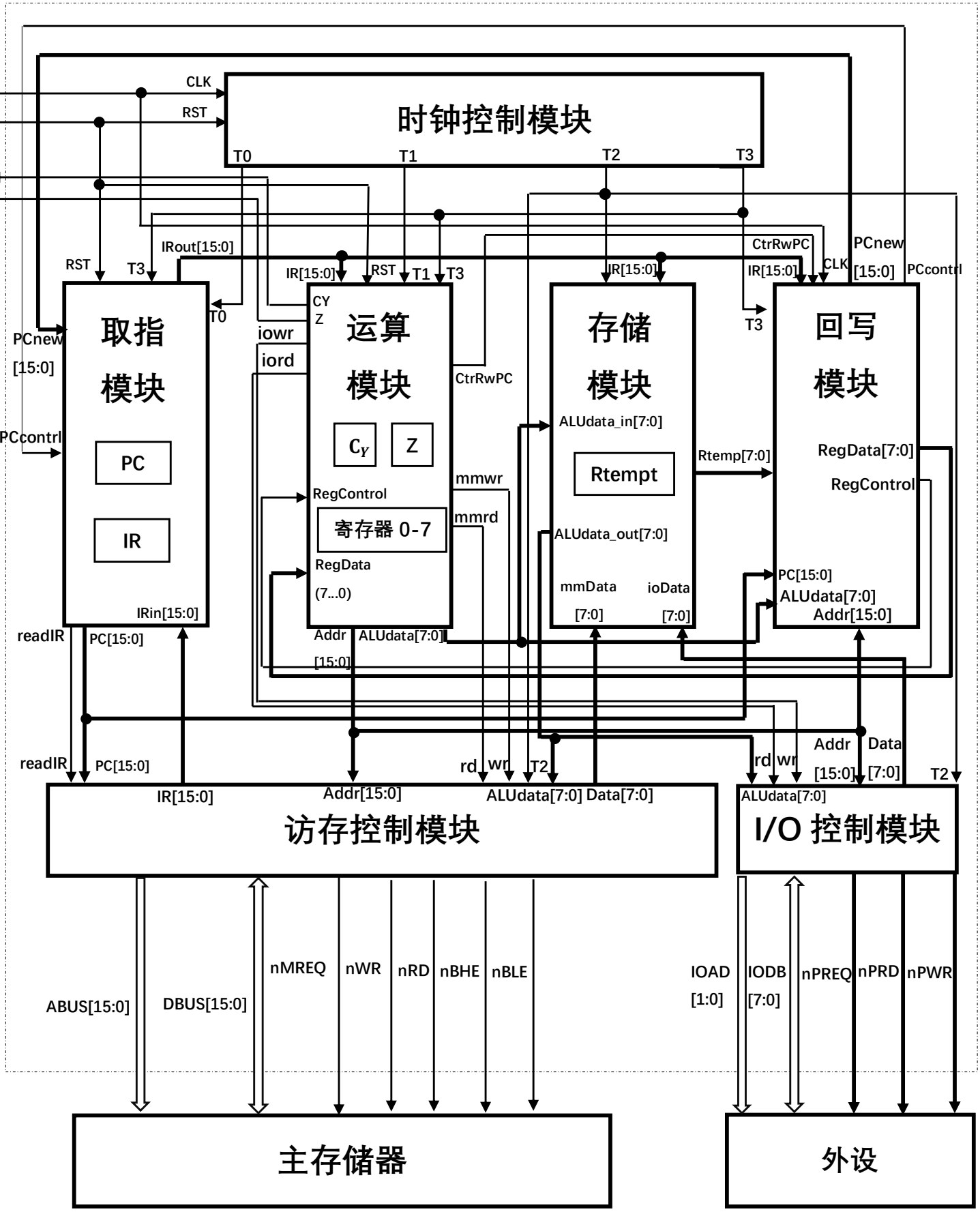
学号：1140310414

哈尔滨工业大学计算机学院

2016 年 7 月

目录

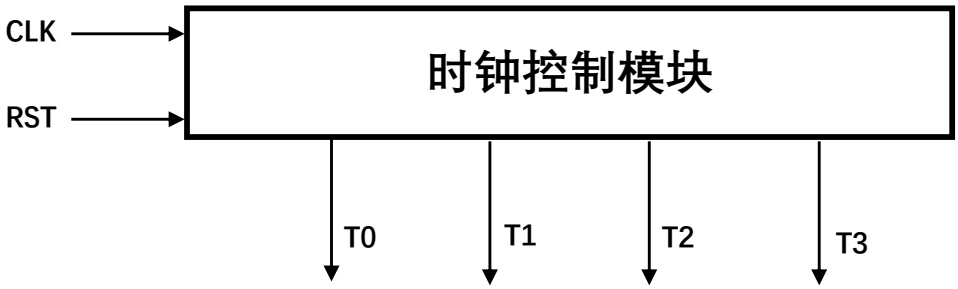
第一部分：详细设计整体框图



第二部分：各模块接口的详细说明

（1）时钟控制模块

1）设计框图



2）功能描述

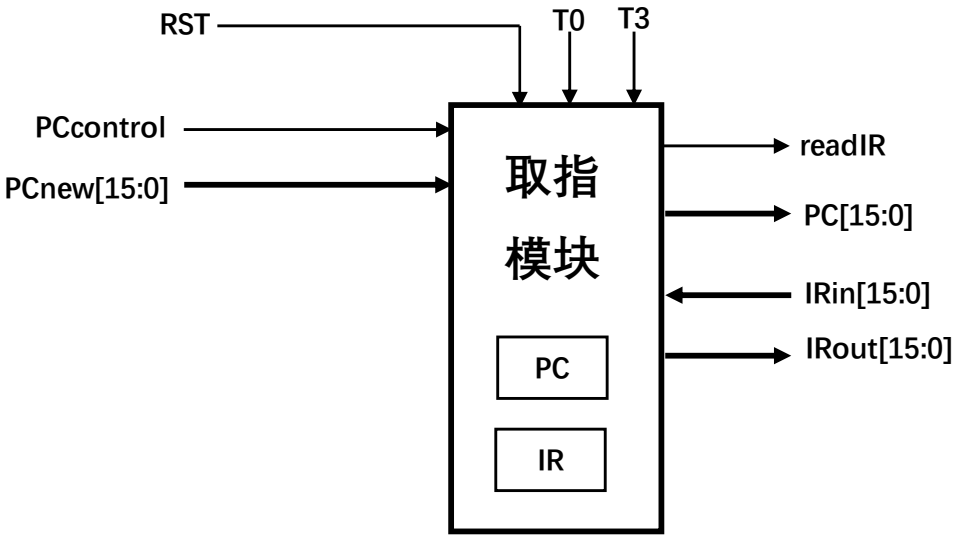
时钟控制模块负责产生节拍，用以控制其他模块的工作状态。

3）接口信号定义

信号名	位数	方向	来源/去向	意义
RST	1	I	处理器板	高电平复位
CLK	1	I	处理器板	系统时钟
T0	1	O	取指模块	控制取指过程
T1	1	O	运算模块	控制运算模块工作
T2	1	O	存储、访存控制、I/O 控制模块	控制访存、I/O 输入
T3	1	O	取指、运算、回写模块	控制回写过程

（2）取指模块

1）设计框图



2）功能描述

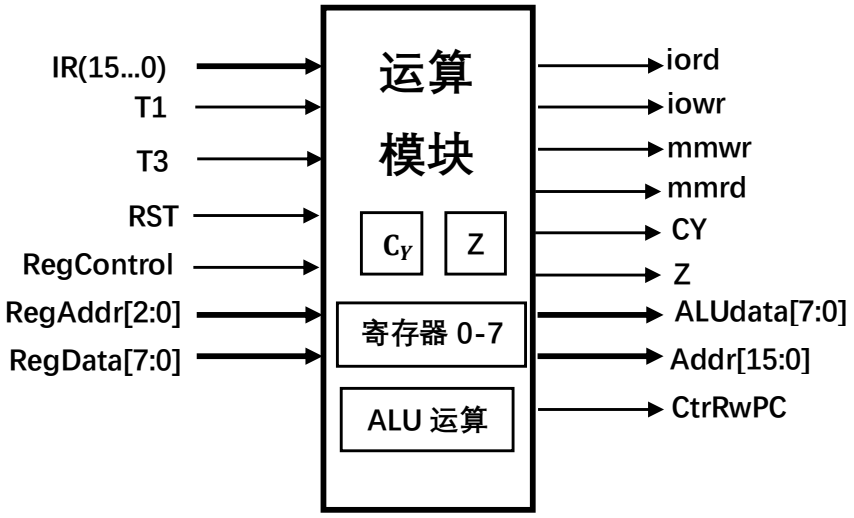
取指模块负责通过访存控制模块取指令，PC 的更新，以及 IR 的传递。

3）接口信号定义

信号名	位数	方向	来源/去向	意义
RST	1	I	处理器板	高电平复位
T0	1	I	时钟控制模块	控制取指过程开始工作
T3	1	I	时钟控制模块	控制回写 PC 过程
PCcontrol	1	I	回写模块	控制 PC 回写
PCnew	16	I	回写模块	新的 PC
readIR	1	O	访存控制模块	控制访存控制模块读 IR 值
PC	16	O	访存控制模块 回写模块	按 PC 中地址寻访 IR、计算 PC 的新值
IRin	16	I	访存控制模块	IR 值
IRout	16	O	运算、存储、回写模块	传输其他模块的 IR 值

（3）运算模块

1）设计框图



2）功能描述

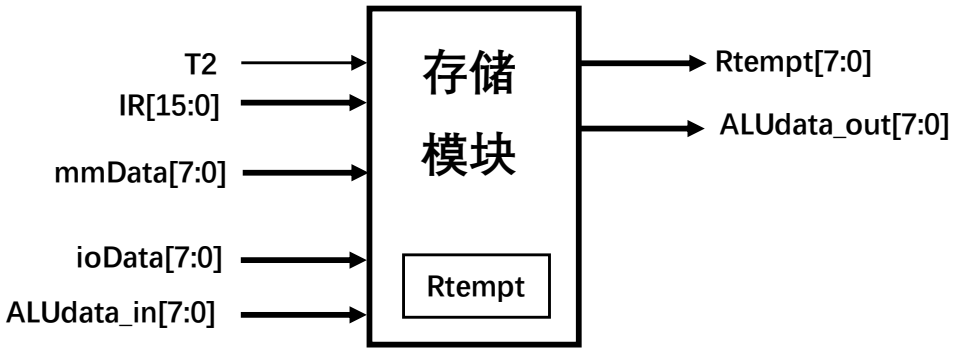
运算模块通过取指模块传输的 IR 计算出相应的结果地址或数据，并可以完成寄存器的更新。

3）接口信号定义

信号名	位数	方向	来源/去向	意义
T1	1	I	时钟控制模块	控制运算过程开始工作
T3	1	I	时钟控制模块	控制回写寄存器过程
IR	16	I	取指模块	指令
RST	1	I	处理器板	高电平复位
RegControl	1	I	回写模块	控制寄存器回写
RegAddr	3	I	回写模块	回写寄存器地址
RegData	8	I	回写模块	回写寄存器数据
iord	1	O	I/O 控制模块	控制 I/O 读
iowr	1	O	I/O 控制模块	控制 I/O 写
mmrd	1	O	访存控制模块	控制存储器读
mmwr	1	O	访存控制模块	控制存储器写
CY	1	O	FPGA	进位、借位信号
Z	1	O	FPGA	零标志
ALUdata	8	O	存储模块、回写模块	ALU 运算数据结果
Addr	16	O	访存控制、I/O 控制、回写模块	ALU 运算地址结果
CtrRwPC	1	O	回写模块	控制 JZ 回写

（4）存储模块

1）设计框图



2）功能描述

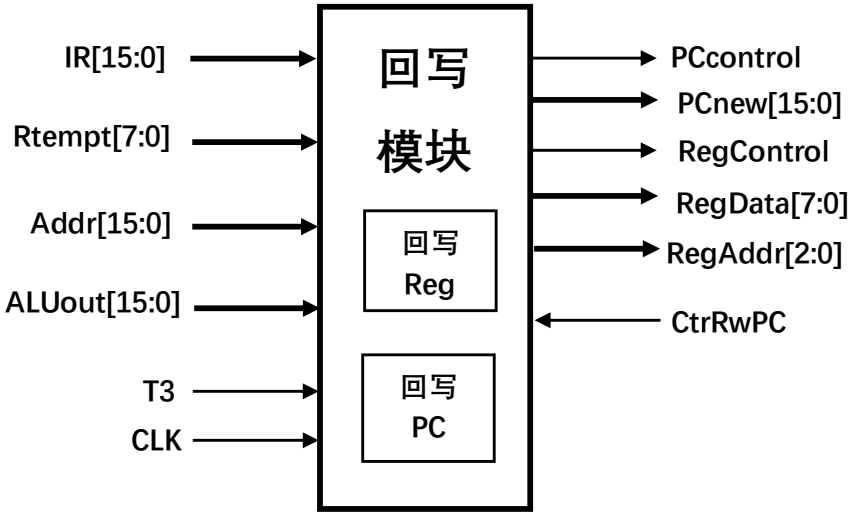
存储模块负责接受访存控制模块和 I/O 控制模块传输回的数据，并传送给回写模块。

3）接口信号定义

信号名	位数	方向	来源/去向	意义
T2	1	I	时钟控制模块	控制存储模块开始工作
IR	16	I	取指模块	指令
mmData	8	I	访存控制模块	访存控制模块读出的数据
ioData	8	I	I/0 控制模块	I/O 端口输入数据
Rtempt	8	O	回写模块	传输数据
ALUdata_in	8		运算模块	ALU 运算数据结果
ALUdata_out	8		访存控制、I/O 控制	传输 ALU 运算数据结果

（5）回写模块

1）设计框图



2）功能描述

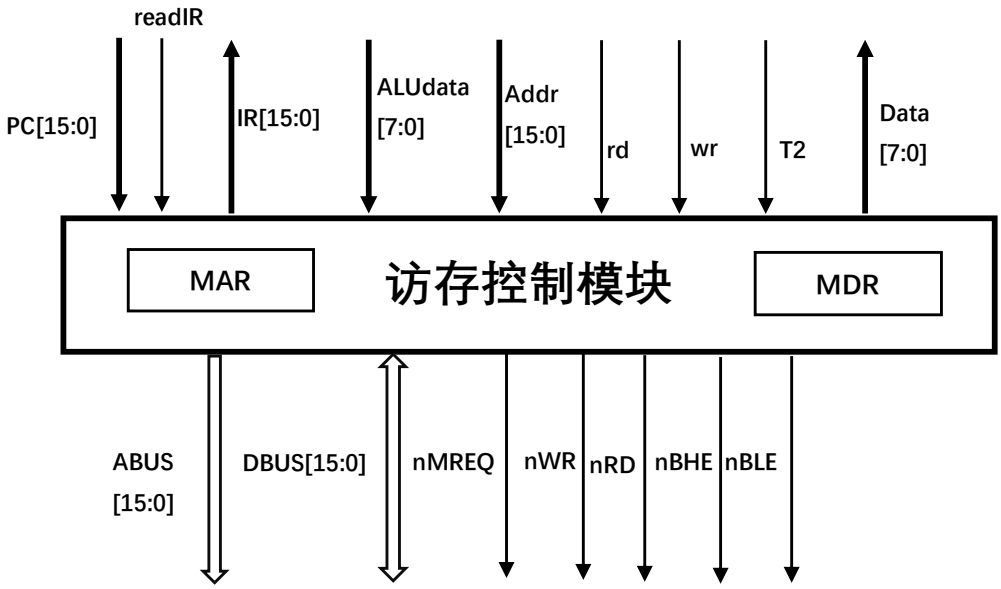
回写模块包括回写寄存器和回写 PC 两部分构成，回写寄存器通过 ALU 运算出的数据结果传输给运算模块回写数据，回写 PC 通过 ALU 算出的地址结果回写

3）接口信号定义

信号名	位数	方向	来源/去向	意义
T3	1	I	时钟控制模块	控制回写模块开始工作
CLK	1	I	时钟控制模块	PC 赋值控制
IR	16	I	取指模块	指令
Rtempt	8	I	存储模块	接收数据
ALUdata	8	I	运算模块	ALU 运算数据结果
Addr	16	I	运算模块	ALU 运算地址结果
PCcontrol	1	O	取指模块	控制 PC 回写
PCnew	16	O	取指模块	PC 新值
RegControl	1	O	运算模块	控制寄存器回写
RegData	8	O	运算模块	寄存器回写值
RegAddr	3	O	运算模块	寄存器回写地址
CtrRwPC	1	I	运算模块	控制 JZ 回写

（6）访存控制模块

1）设计框图



2）功能描述

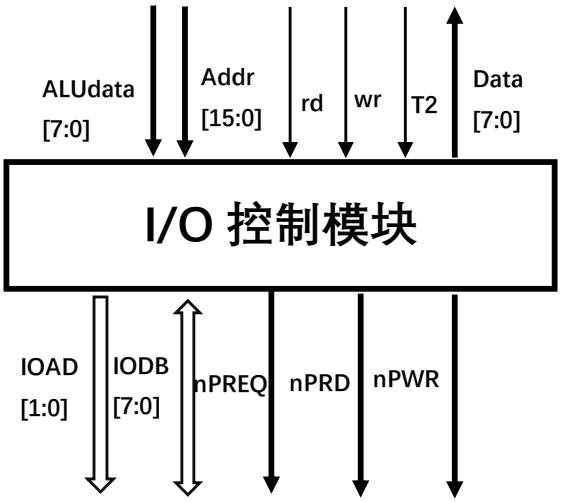
访存控制模块功能为寻址、控制 CPU 访问主存、将访存得到的数据传输给存储模块。

3）接口信号定义

信号名	位数	方向	来源/去向	意义
PC	16	I	取指模块	指令地址
readIR	1	I	取指模块	控制访存控制模块寻指令
IR	16	O	取指模块	指令
ALUdata	8	I	存储模块	ALU 运算数据结果
Addr	16	I	运算模块	ALU 运算地址结果
T2	1	I	时钟控制模块	控制访存工作开始
rd	1	I	运算模块	控制存储器读
wr	1	I	运算模块	控制存储器写
Data	8	O	存储模块	将数据传输给存储模块
ABUS	16	O	主存储器	地址总线
DBUS	16	I/O	主存储器	数据总线
nMREQ	1	O	主存储器	存储器片选
nWR	1	O	主存储器	存储器写
nRD	1	O	主存储器	存储器读
nBHE	1	O	主存储器	高字节访问允许
nBLE	1	O	主存储器	低字节访问允许

（7）I/O 控制模块

1）设计框图



2）功能描述

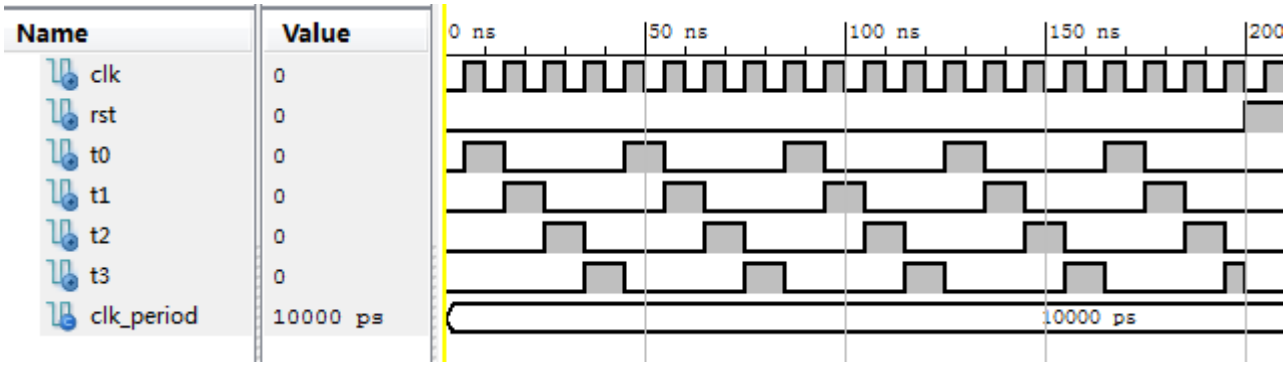
I/O 控制模块用于控制 CPU 与 I/O 接口之间的读写。

3）接口信号定义

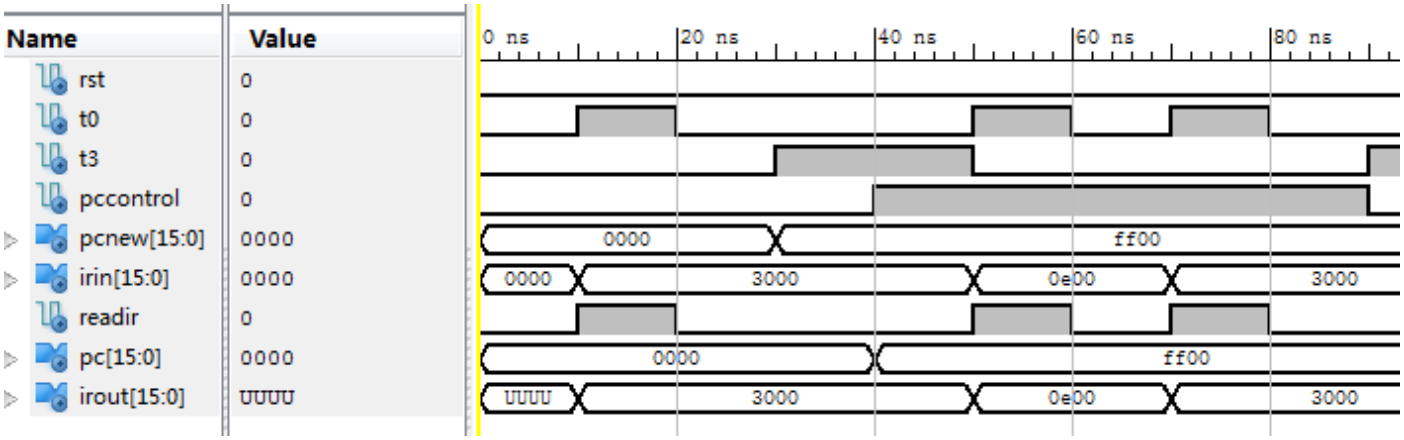
信号名	位数	方向	来源/去向	意义
ALUdata	8	I	存储模块	ALU 运算数据结果
Addr	16	I	运算模块	ALU 运算地址结果
T2	1	I	时钟控制模块	控制访存工作开始
rd	1	I	运算模块	控制 I/O 读
wr	1	I	运算模块	控制 I/O 写
Data	8	O	存储模块	将数据传输给存储模块
IOAD	2	O	FPGA	地址总线
IODB	8	I/O	FPGA	数据总线
nPREQ	1	O	FPGA	外设访问允许
nPRD	1	O	FPGA	外设读
nPWR	1	O	FPGA	外设写

第三部分：系统仿真波形

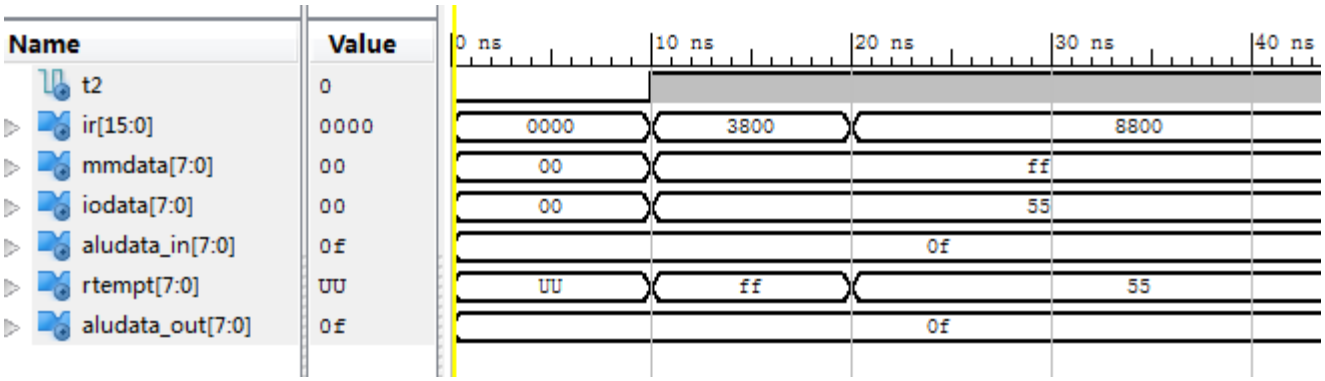
（1）时钟控制模块



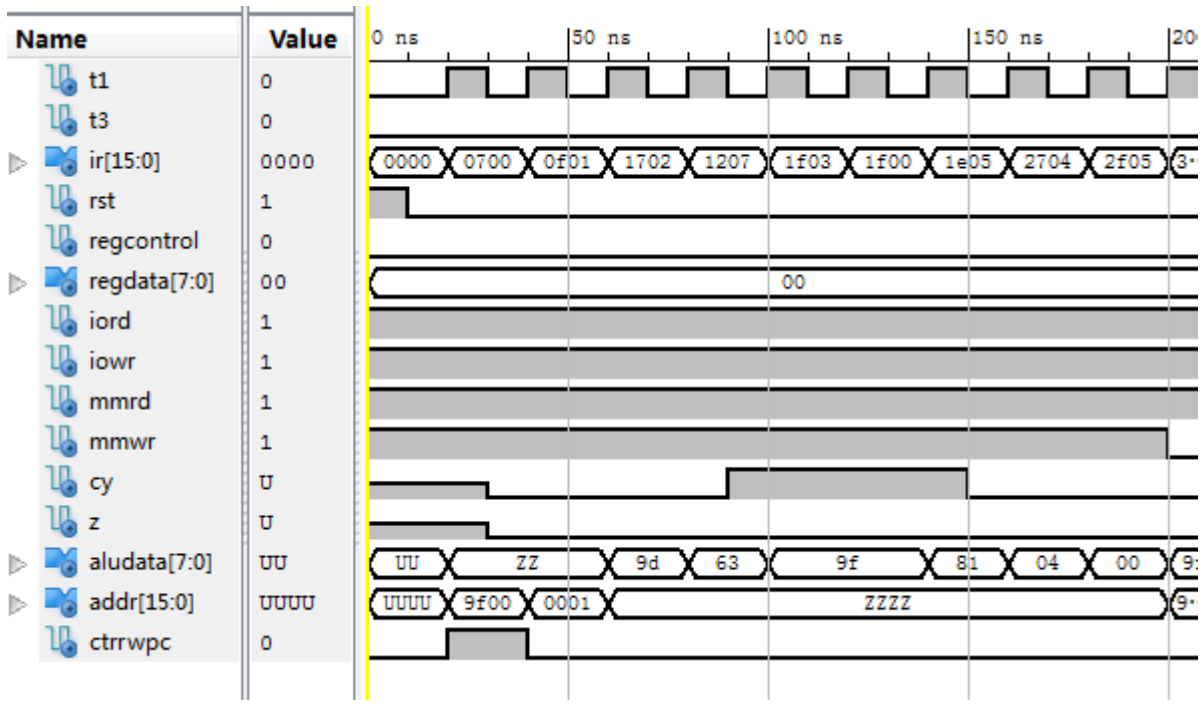
（2）取指模块

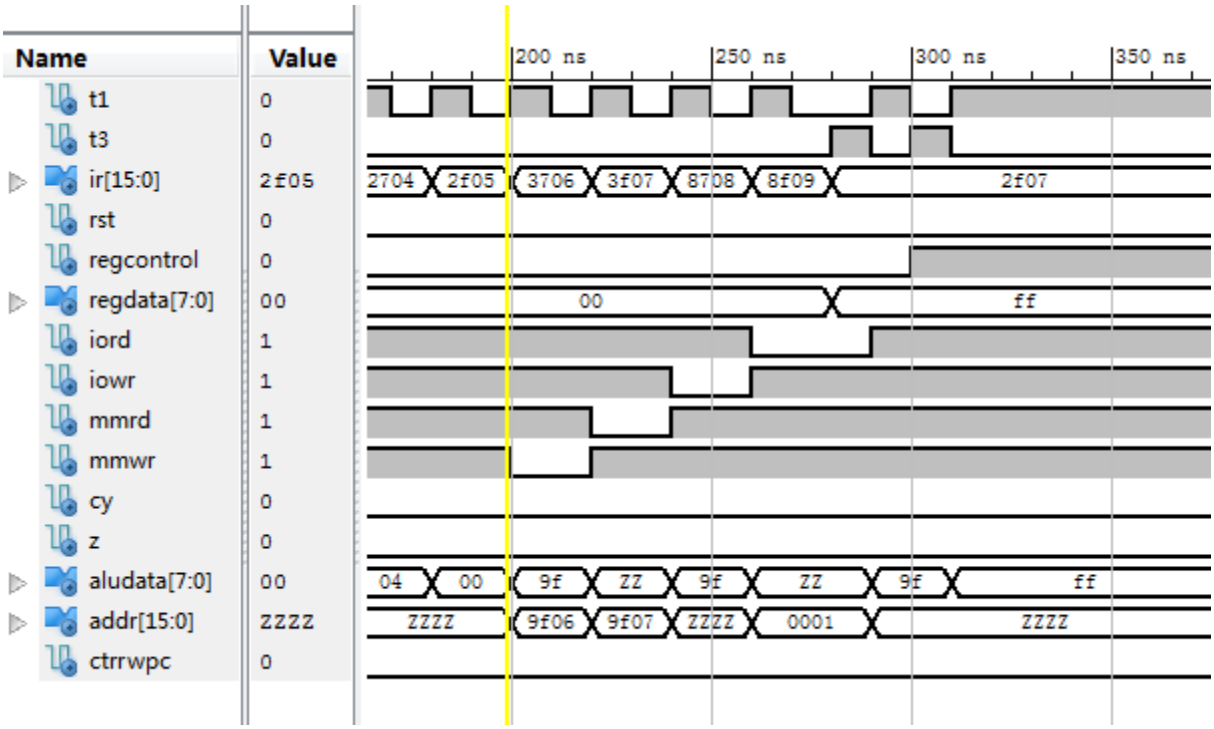


（3）存储模块

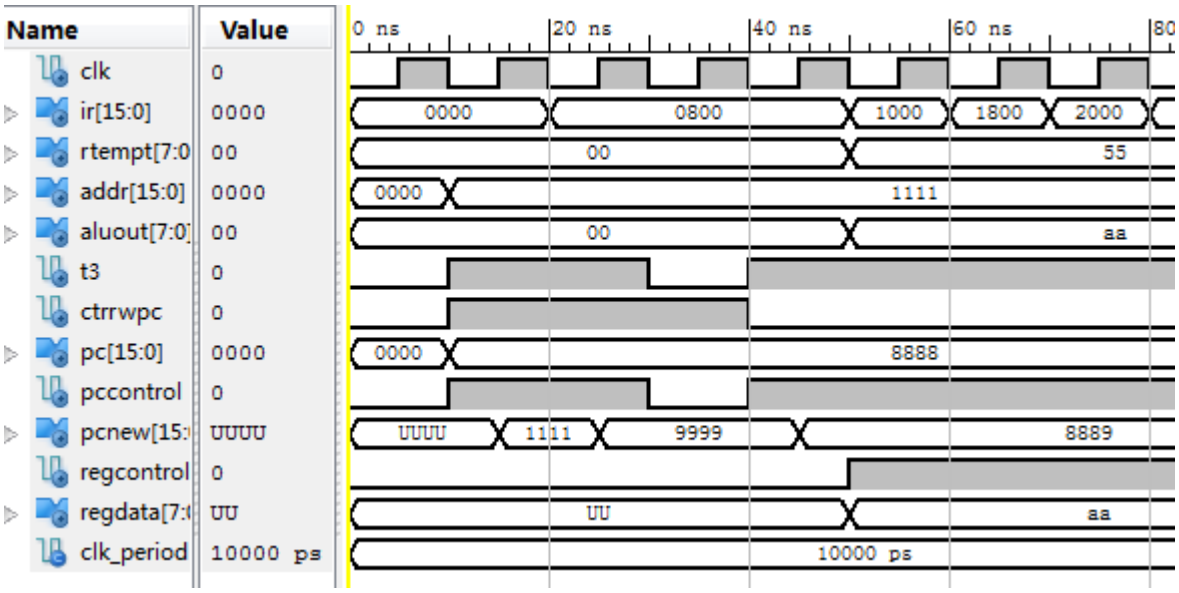


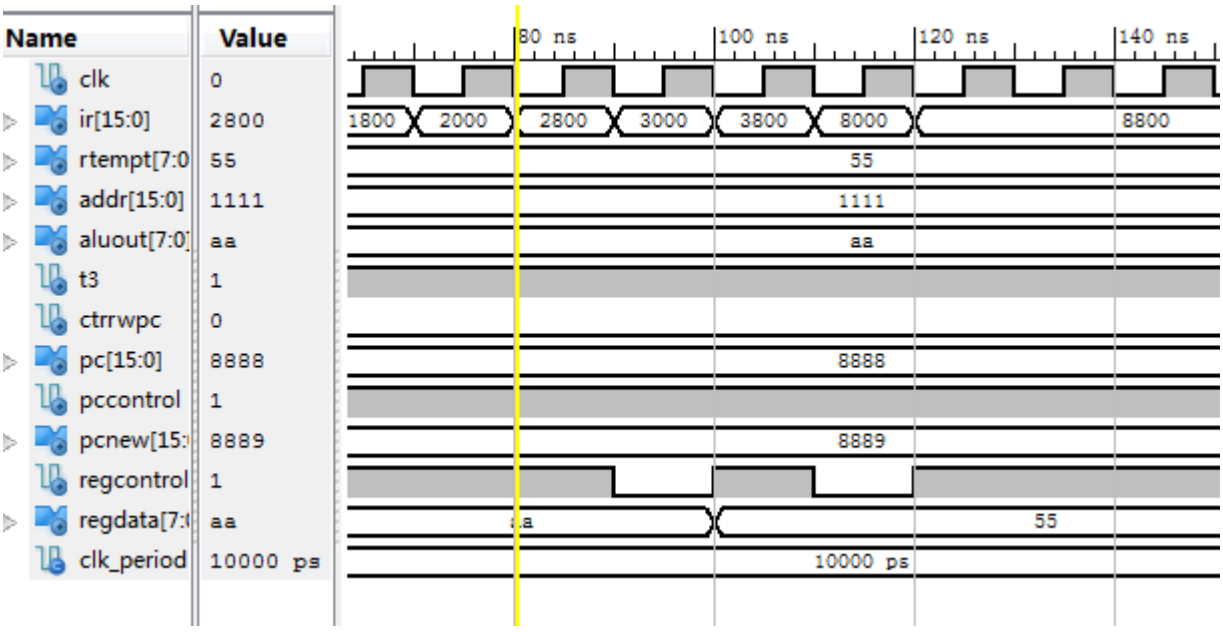
（4）运算模块


Name
Value
t1
0
t3
0
ir[15:0]
0000
rst
1
regcontrol
0
regdata[7:0]
00
iord
1
iowr
1
mmrd
1
mmwr
1
cy
U
z
U
aludata[7:0]
UU
addr[15:0]
UUUU
ctrrwpc
0
0 ns
50 ns
100 ns
150 ns
0000
0700
0f01
1702
1207
1f03
1f00
1e05
2704
2f05
00
UU
ZZ
9d
63
9f
81
04
00
UUUU
9f00
0001
ZZZZ

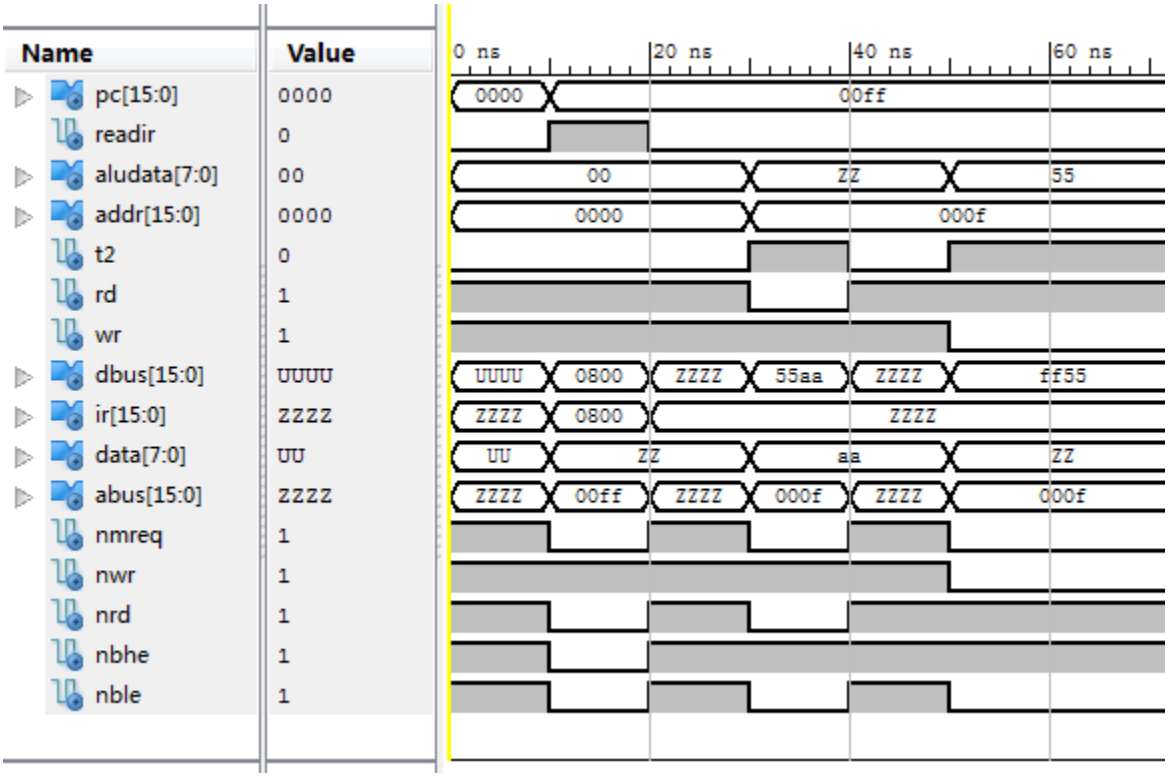

Name
Value
t1
0
t3
0
ir[15:0]
2f05
rst
0
regcontrol
0
regdata[7:0]
00
iord
1
iowr
1
mmrd
1
mmwr
1
cy
0
z
0
aludata[7:0]
00
addr[15:0]
ZZZZ
ctrrwpc
0
200 ns
250 ns
300 ns
350 ns
2704
2f05
3706
3f07
8708
8f09
2f07
00
ff
04
00
9f
ZZ
9f
ZZ
9f
ff
ZZZZ
9f06
9f07
ZZZZ
0001
ZZZZ

(5) 回写模块

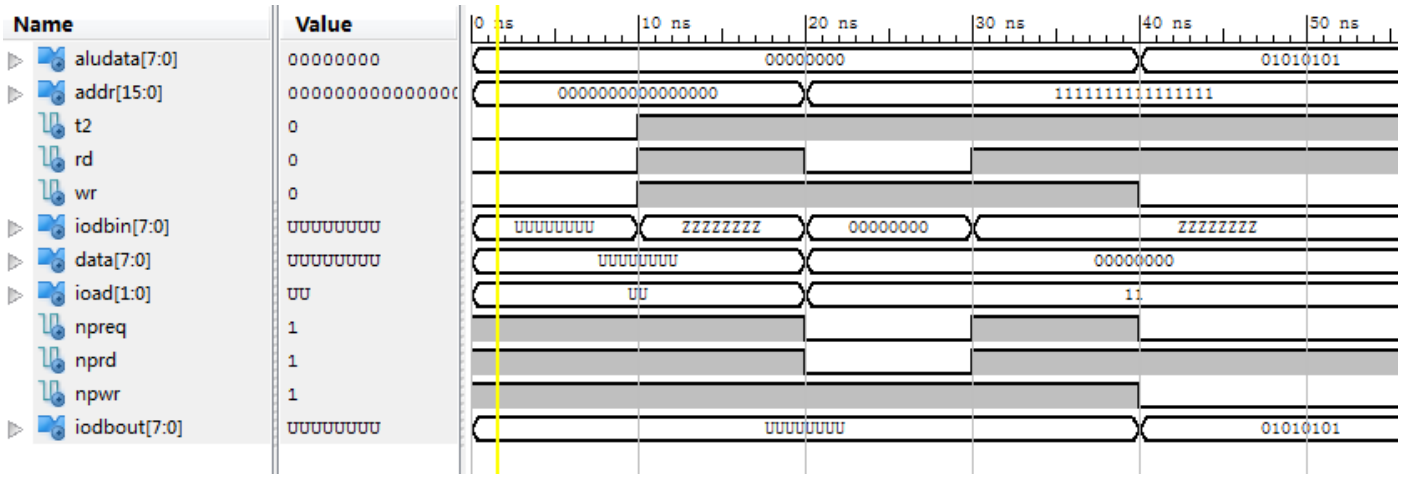

Name
Value
clk
0
ir[15:0]
0000
rtempt[7:0
00
addr[15:0]
0000
aluout[7:0]
00
t3
0
ctrrwpc
0
pc[15:0]
0000
pccontrol
0
pcnew[15:
UUUU
regcontrol
0
regdata[7:
UU
clk_period
10000 ps
0 ns
20 ns
40 ns
60 ns
0000
0800
1000
1800
2000
00
55
1111
aa
8888
1111
9999
8889
10000 ps


Name
Value
clk
0
ir[15:0]
2800
rtempt[7:0
55
addr[15:0]
1111
aluout[7:0]
aa
t3
1
ctrrwpc
0
pc[15:0]
8888
pccontrol
1
pcnew[15:
8889
regcontrol
1
regdata[7:
aa
clk_period
10000 ps
80 ns
100 ns
120 ns
140 ns
1800
2000
2800
3000
3800
8000
8800
55
1111
aa
8888
8889
55
10000 ps

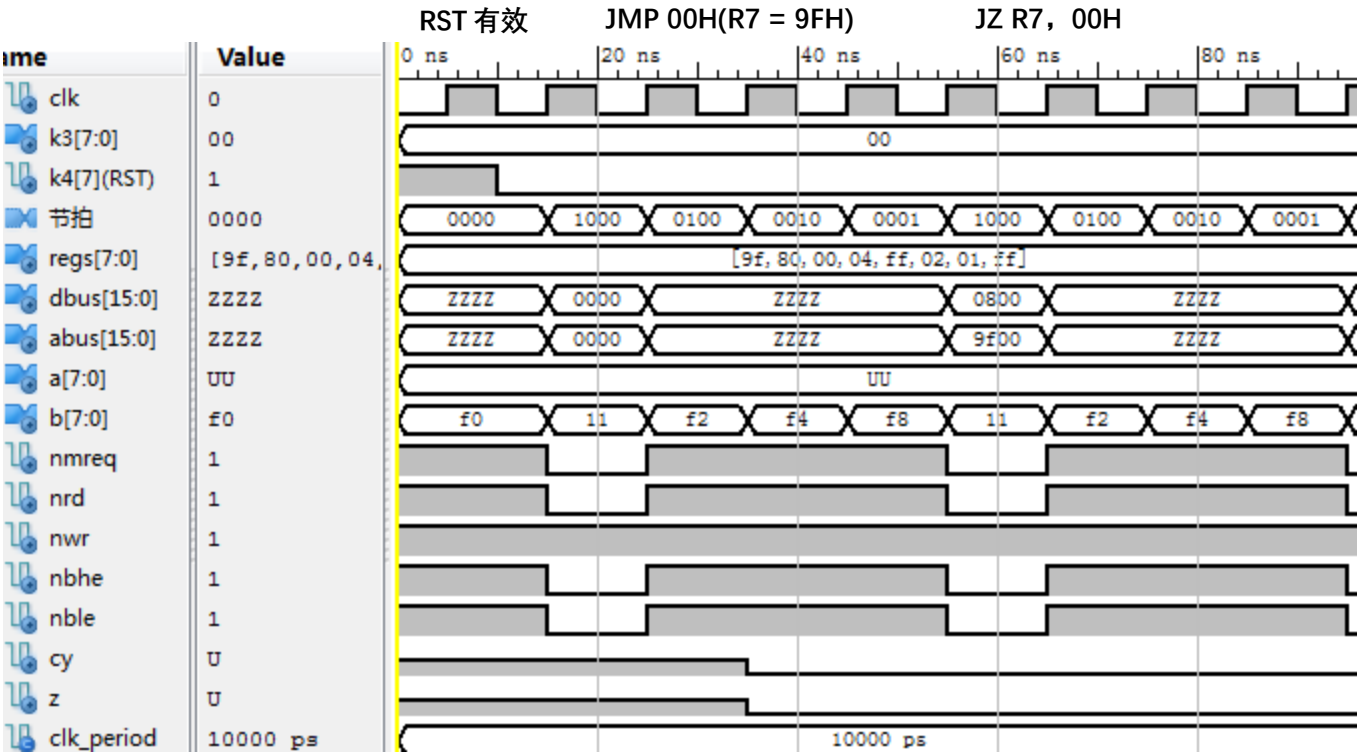
（6）访存控制模块

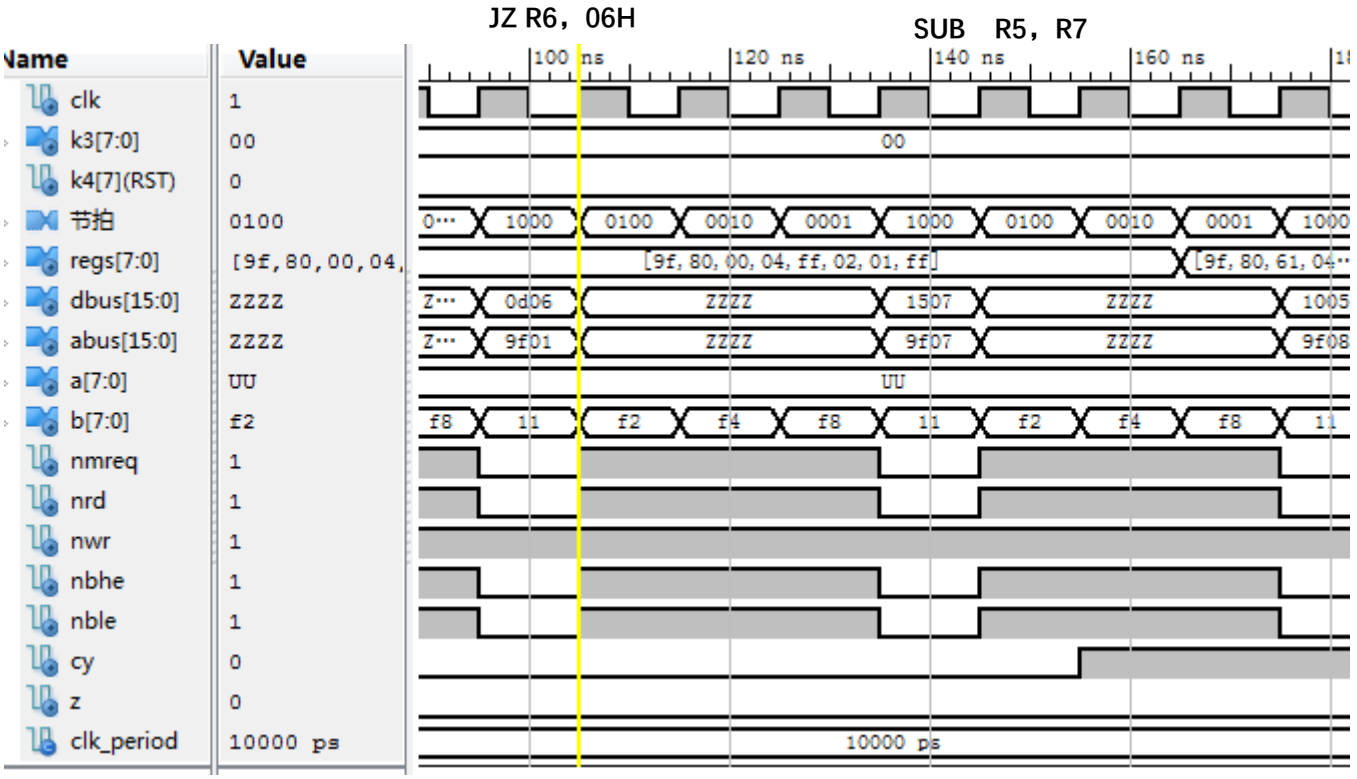


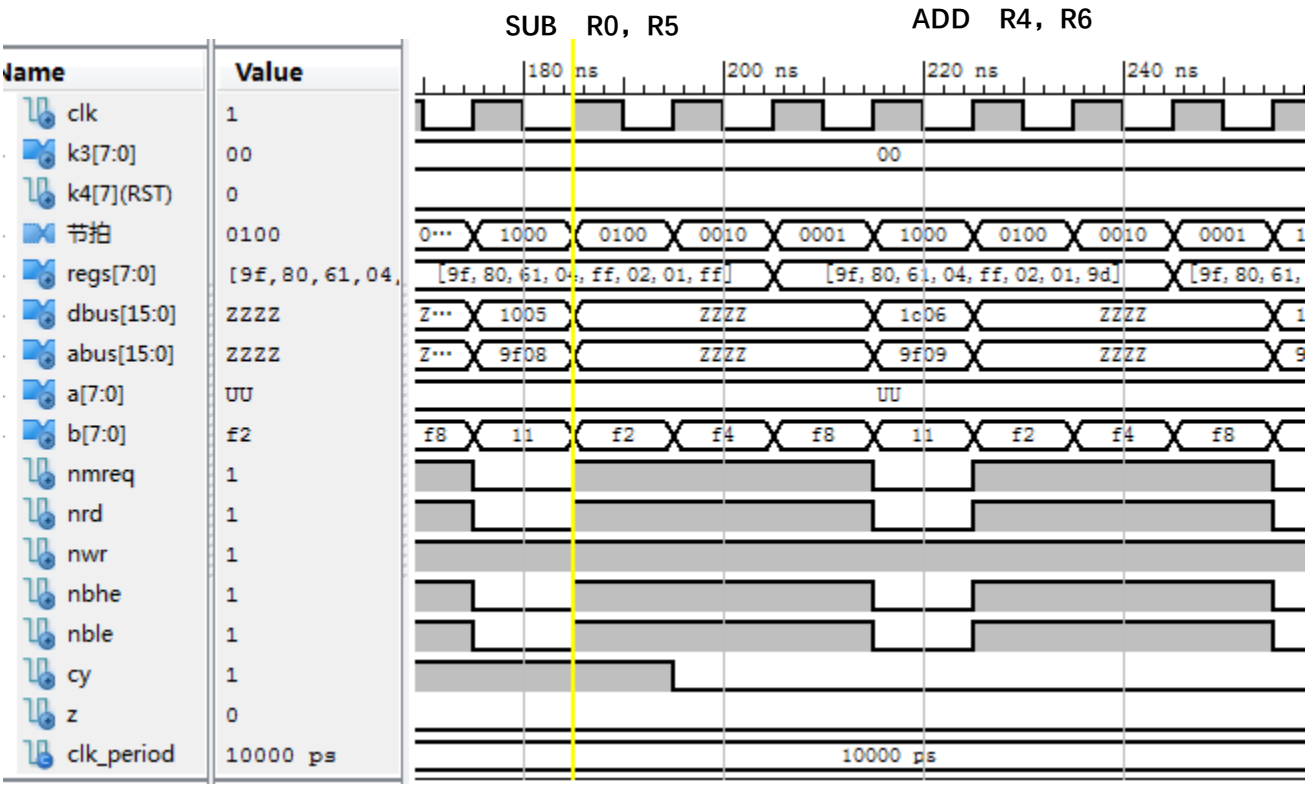
（7）I/O 控制模块

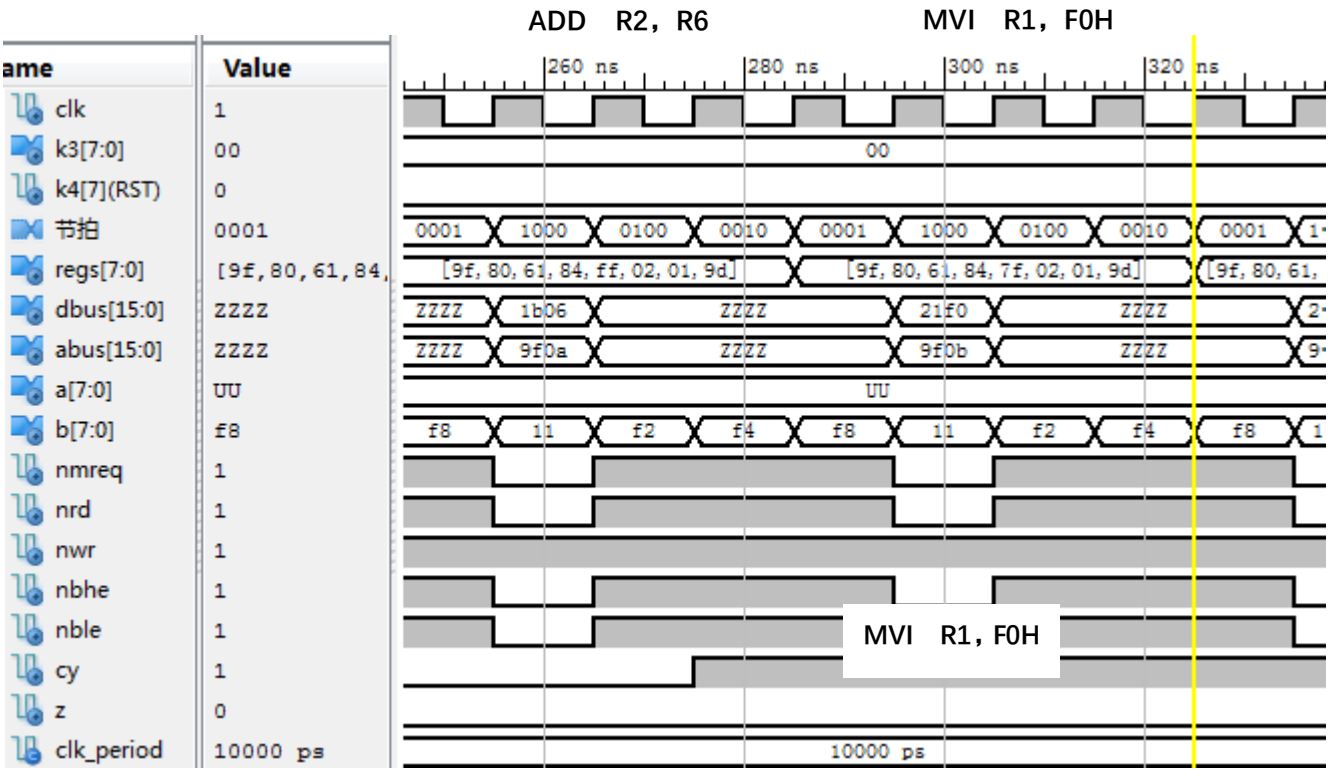


(8) 总模块仿真波形

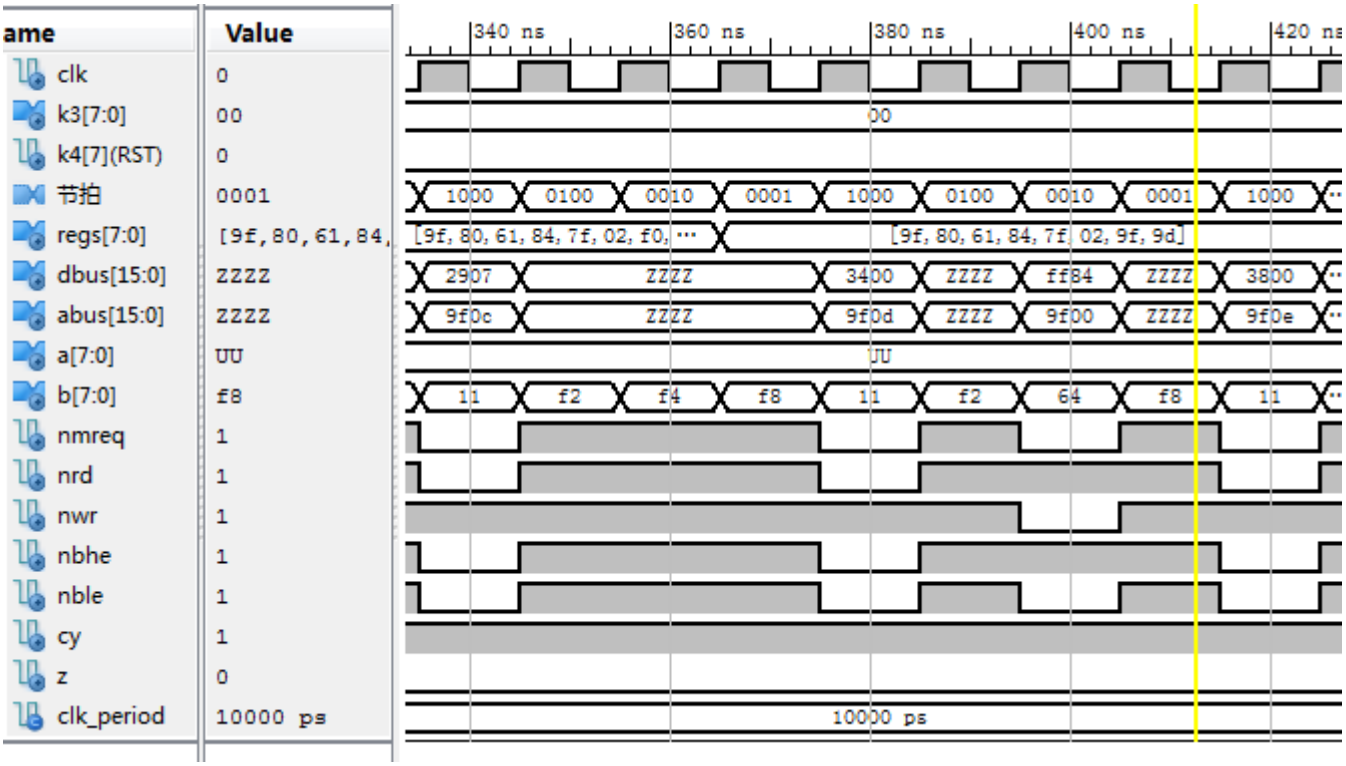

RST 有效
JMP 00H(R7 = 9FH)
JZ R7，00H
Value
clk 0
k3[7:0] 00
k4[7](RST) 1
节拍 0000
regs[7:0] [9f,80,00,04,
dbus[15:0] ZZZZ
abus[15:0] ZZZZ
a[7:0] UU
b[7:0] f0
nmreq 1
nrd 1
nwr 1
nbhe 1
nble 1
cy U
z U
clk_period 10000 ps
0 ns
20 ns
40 ns
60 ns
80 ns
[9f, 80, 00, 04, ff, 02, 01, ff]
10000 ps


JZ R6，06H
SUB R5，R7
Value
clk 1
k3[7:0] 00
k4[7](RST) 0
节拍 0100
regs[7:0] [9f,80,00,04,
dbus[15:0] ZZZZ
abus[15:0] ZZZZ
a[7:0] UU
b[7:0] f2
nmreq 1
nrd 1
nwr 1
nbhe 1
nble 1
cy 0
z 0
clk_period 10000 ps
100 ns
120 ns
140 ns
160 ns
[9f, 80, 00, 04, ff, 02, 01, ff]
[9f, 80, 61, 04
10000 ps

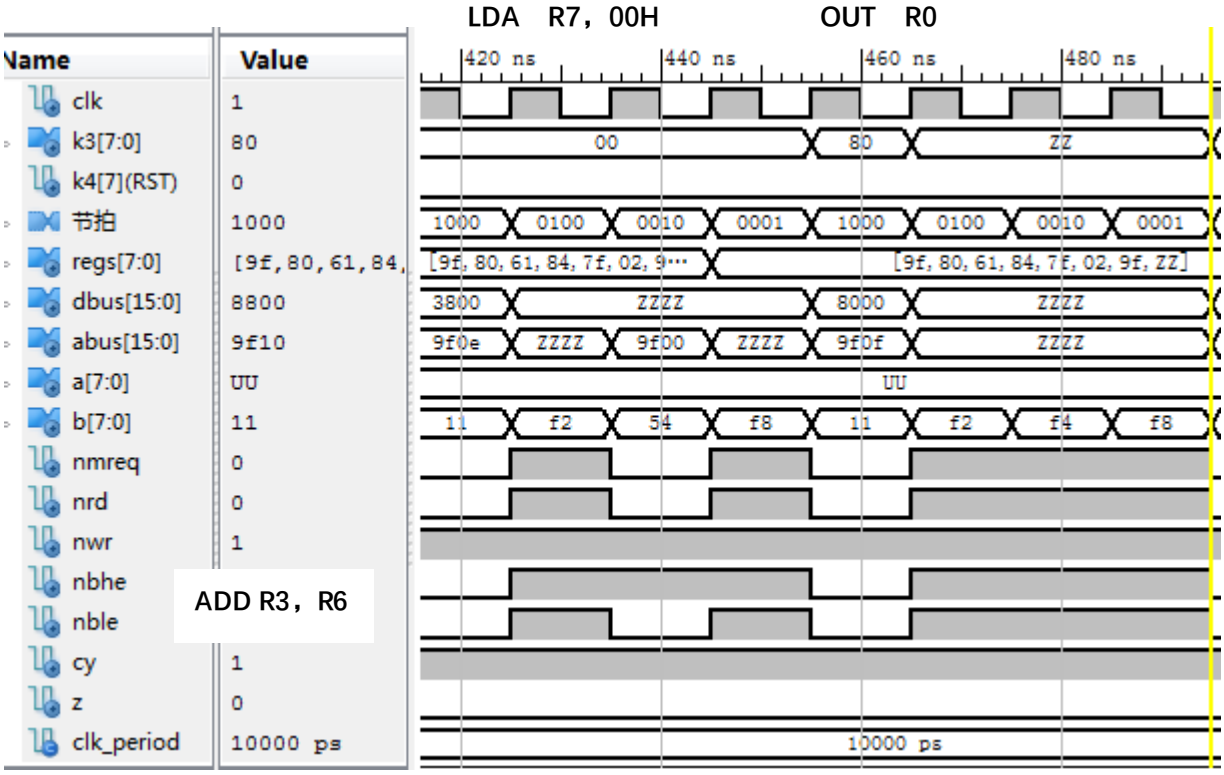

SUB R0，R5
ADD R4，R6
Name
Value
clk
1
k3[7:0]
00
k4[7](RST)
0
节拍
0100
regs[7:0]
[9f,80,61,04,
dbus[15:0]
ZZZZ
abus[15:0]
ZZZZ
a[7:0]
UU
b[7:0]
f2
nmreq
1
nrd
1
nwr
1
nbhe
1
nble
1
cy
1
z
0
clk_period
10000 ps
180 ns
200 ns
220 ns
240 ns
00
0100
0010
0001
1000
[9f, 80, 61, 04, ff, 02, 01, ff]
[9f, 80, 61, 04, ff, 02, 01, 9d]
1005
1c06
9f08
9f09
UU
f8
11
f2
f4
10000 ps


ADD R2，R6
MVI R1，F0H
Name
Value
clk
1
k3[7:0]
00
k4[7](RST)
0
节拍
0001
regs[7:0]
[9f,80,61,84,
dbus[15:0]
ZZZZ
abus[15:0]
ZZZZ
a[7:0]
UU
b[7:0]
f8
nmreq
1
nrd
1
nwr
1
nbhe
1
nble
1
cy
1
z
0
clk_period
10000 ps
260 ns
280 ns
300 ns
320 ns
00
0001
1000
0100
0010
[9f, 80, 61, 84, ff, 02, 01, 9d]
[9f, 80, 61, 84, 7f, 02, 01, 9d]
1b06
21f0
9f0a
9f0b
UU
f8
11
f2
f4
MVI R1，F0H
10000 ps

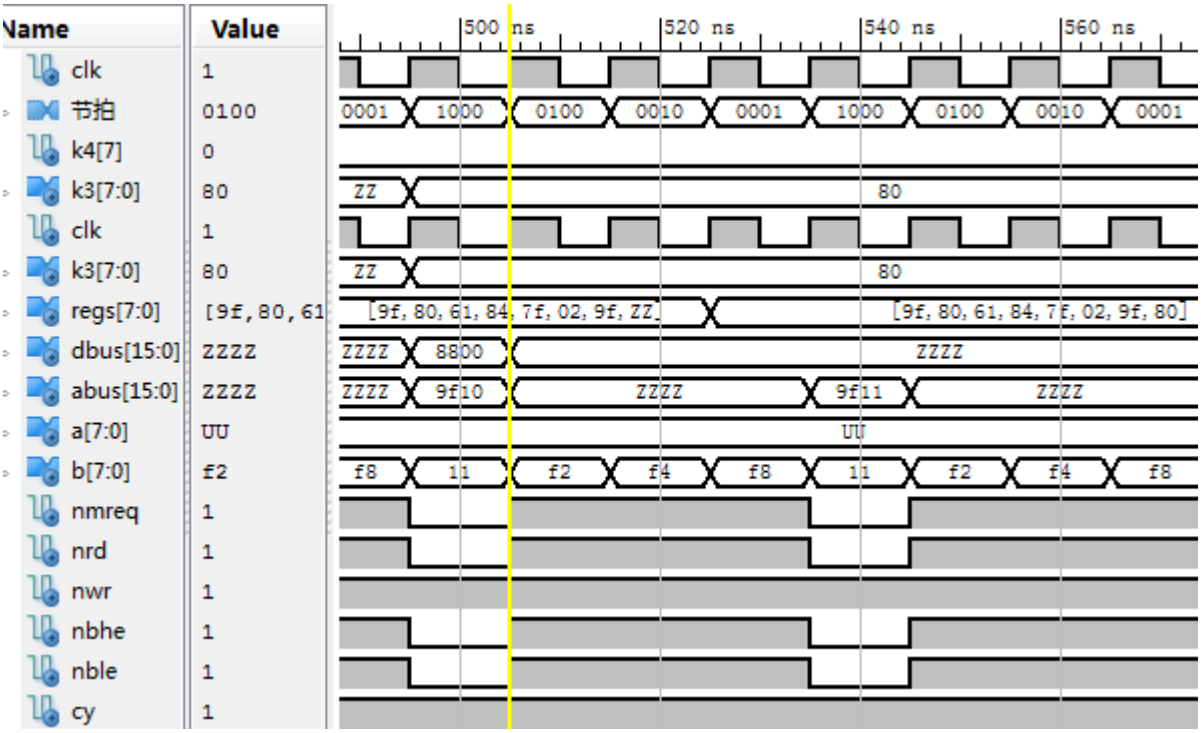
MOV R1，R7　　STA R4，00H



LDA R7，00H　　OUT R0



IN R0


Name
Value
clk
1
节拍
0100
k4[7]
0
k3[7:0]
80
clk
1
k3[7:0]
80
regs[7:0]
[9f,80,61
dbus[15:0]
ZZZZ
abus[15:0]
ZZZZ
a[7:0]
UU
b[7:0]
f2
nmreq
1
nrd
1
nwr
1
nbhe
1
nble
1
cy
1
500 ns
520 ns
540 ns
560 ns
0001
1000
0100
0010
0001
1000
0100
0010
0001
ZZ
80
ZZ
80
[9f, 80, 61, 84, 7f, 02, 9f, ZZ]
[9f, 80, 61, 84, 7f, 02, 9f, 80]
ZZZZ
8800
ZZZZ
ZZZZ
9f10
ZZZZ
9f11
ZZZZ
UU
f8
11
f2
f4
f8
11
f2
f4
f8

第四部分：系统管脚定义的 UCF 文件

```
NET "K0(0)" LOC = "P103";
NET "K0(1)" LOC = "P102";
NET "K0(2)" LOC = "P101";
NET "K0(3)" LOC = "P100";
NET "K0(4)" LOC = "P97";
NET "K0(5)" LOC = "P96";
NET "K0(6)" LOC = "P95";
NET "K0(7)" LOC = "P94";
NET "K1(0)" LOC = "P87";
NET "K1(1)" LOC = "P86";
NET "K1(2)" LOC = "P85";
NET "K1(3)" LOC = "P84";
NET "K1(4)" LOC = "P82";
NET "K1(5)" LOC = "P81";
NET "K1(6)" LOC = "P80";
NET "K1(7)" LOC = "P79";
NET "K2(0)" LOC = "P73";
NET "K2(1)" LOC = "P72";
NET "K2(2)" LOC = "P71";
NET "K2(3)" LOC = "P70";
NET "K2(4)" LOC = "P66";
NET "K2(5)" LOC = "P65";
NET "K2(6)" LOC = "P64";
NET "K2(7)" LOC = "P63";
NET "K3(0)" LOC = "P47";
NET "K3(1)" LOC = "P48";
NET "K3(2)" LOC = "P49";
NET "K3(3)" LOC = "P50";
NET "K3(4)" LOC = "P53";
NET "K3(5)" LOC = "P54";
NET "K3(6)" LOC = "P55";
NET "K3(7)" LOC = "P56";
NET "K4(0)" LOC = "P41";
NET "K4(1)" LOC = "P40";
NET "K4(2)" LOC = "P39";
NET "K4(3)" LOC = "P38";
NET "K4(4)" LOC = "P36";
NET "K4(5)" LOC = "P35";
NET "K4(6)" LOC = "P34";
NET "K4(7)" LOC = "P33";
NET "A(0)" LOC = "P110";
NET "A(1)" LOC = "P111";
NET "A(2)" LOC = "P203";
NET "A(3)" LOC = "P177";
```

```
NET "A(4)" LOC = "P184";
NET "A(5)" LOC = "P178";
NET "A(6)" LOC = "P152";
NET "A(7)" LOC = "P147";
NET "B(0)" LOC = "P125";
NET "B(1)" LOC = "P124";
NET "B(2)" LOC = "P109";
NET "B(3)" LOC = "P108";
NET "B(4)" LOC = "P107";
NET "B(5)" LOC = "P99";
NET "B(6)" LOC = "P93";
NET "B(7)" LOC = "P78";
NET "S0(0)" LOC = "P223";
NET "S0(1)" LOC = "P222";
NET "S0(2)" LOC = "P221";
NET "S0(3)" LOC = "P220";
NET "S0(4)" LOC = "P218";
NET "S0(5)" LOC = "P217";
NET "S0(6)" LOC = "P216";
NET "S0(7)" LOC = "P215";
NET "S1(0)" LOC = "P235";
NET "S1(1)" LOC = "P234";
NET "S1(2)" LOC = "P232";
NET "S1(3)" LOC = "P231";
NET "S1(4)" LOC = "P230";
NET "S1(5)" LOC = "P229";
NET "S1(6)" LOC = "P228";
NET "S1(7)" LOC = "P224";
NET "S2(0)" LOC = "P7";
NET "S2(1)" LOC = "P6";
NET "S2(2)" LOC = "P5";
NET "S2(3)" LOC = "P4";
NET "S2(4)" LOC = "P3";
NET "S2(5)" LOC = "P238";
NET "S2(6)" LOC = "P237";
NET "S2(7)" LOC = "P236";
NET "S3(0)" LOC = "P19";
NET "S3(1)" LOC = "P18";
NET "S3(2)" LOC = "P17";
NET "S3(3)" LOC = "P13";
NET "S3(4)" LOC = "P12";
NET "S3(5)" LOC = "P11";
NET "S3(6)" LOC = "P10";
NET "S3(7)" LOC = "P9";
NET "S4(0)" LOC = "P28";
NET "S4(1)" LOC = "P27";
```

```
NET "S4(2)" LOC = "P26";
NET "S4(3)" LOC = "P25";
NET "S4(4)" LOC = "P24";
NET "S4(5)" LOC = "P23";
NET "S4(6)" LOC = "P21";
NET "S4(7)" LOC = "P20";
NET "S5(0)" LOC = "P74";
NET "S5(1)" LOC = "P68";
NET "S5(2)" LOC = "P67";
NET "S5(3)" LOC = "P57";
NET "S5(4)" LOC = "P52";
NET "S5(5)" LOC = "P46";
NET "S5(6)" LOC = "P42";
NET "S5(7)" LOC = "P31";
NET "ABUS(0)" LOC = "P153";
NET "ABUS(1)" LOC = "P154";
NET "ABUS(2)" LOC = "P155";
NET "ABUS(3)" LOC = "P156";
NET "ABUS(4)" LOC = "P157";
NET "ABUS(5)" LOC = "P173";
NET "ABUS(6)" LOC = "P174";
NET "ABUS(7)" LOC = "P175";
NET "ABUS(8)" LOC = "P176";
NET "ABUS(9)" LOC = "P186";
NET "ABUS(10)" LOC = "P185";
NET "ABUS(11)" LOC = "P208";
NET "ABUS(12)" LOC = "P207";
NET "ABUS(13)" LOC = "P206";
NET "ABUS(14)" LOC = "P205";
NET "ABUS(15)" LOC = "P187";
NET "DBUS(0)" LOC = "P160";
NET "DBUS(1)" LOC = "P161";
NET "DBUS(2)" LOC = "P162";
NET "DBUS(3)" LOC = "P163";
NET "DBUS(4)" LOC = "P167";
NET "DBUS(5)" LOC = "P168";
NET "DBUS(6)" LOC = "P169";
NET "DBUS(7)" LOC = "P170";
NET "DBUS(8)" LOC = "P202";
NET "DBUS(9)" LOC = "P201";
NET "DBUS(10)" LOC = "P200";
NET "DBUS(11)" LOC = "P199";
NET "DBUS(12)" LOC = "P195";
NET "DBUS(13)" LOC = "P194";
NET "DBUS(14)" LOC = "P193";
NET "DBUS(15)" LOC = "P192";
```

```
NET "nMREQ" LOC = "P159";
NET "nWR" LOC = "P171";
NET "nRD" LOC = "P188";
NET "nBHE" LOC = "P189";
NET "nBLE" LOC = "P191";
NET "CLK" LOC = "P92";
```

第五部分：处理器功能测试程序，包括助记符和二进制代码

助记符　操作数　　二进制代码　　指令(16)　　地址　　备注

++

MVI　　R0,FFH　　0010 0000 1111 1111　　20FFH　　0000H　　R0<-FFH

MVI　　R1,01H　　0010 0001 0000 0001　　2101H　　0001H　　R1<-01H

MVI　　R2,01H　　0010 0010 0000 0010　　2202H　　0002H　　R2<-02H

MVI　　R3,03H　　0010 0011 0000 0011　　2303H　　0003H　　R3<-03H

MVI　　R4,A0H　　0010 0100 1010 0000　　24A0H　　0004H　　R4<-A0H

MVI　　R5,00H　　0010 0101 0000 0000　　2500H　　0005H　　R5<-00H

MVI　　R6,AAH　　0010 0110 1010 1010　　26AAH　　0006H　　R6<-AAH

MVI　　R7,00H　　0010 0111 0000 0000　　2700H　　0007H　　R7<-00H

==========

测试 MVI 指令

OUT　　R6　　1000 0110 0000 0000　　8600H　　0008H　　输出 1010 1010

==========

测试 MOV 指令

MOV　　R6,R7　　0010 1110 0000 0111　　2E07H　　0009H　　R6<-R7　　**R6--00H

OUT　　R6　　1000 0110 0000 0000　　8600H　　000AH　　输出 0000 0000

==========

测试 ADD 指令　---CY　0->1　　Z 0->1　　此时 CY=0 Z=0

ADD　　R0，R1　　0001 1000 0000 0001　　1801H　　000BH　　R0<-R0+R1+CY**CY<-1 Z<-1 R0<-00H

OUT　　R0　　1000 0000 0000 0000　　8000H　　000CH　　输出 0000 0000

==========

测试 JZ　指令

JZ　　R4，15H　　0000 1100 0001 0101　　0C15H　　000DH　　不跳转 **PC+1

==========

测试 ADD 指令　---CY　1->0　　Z　1->0　　此时 CY=1 Z=1

ADD　　R0，R1　　0001 1000 0000 0001　　1801H　　000EH　　R0<-R0+R1+CY **CY<-0 Z<-0 R0<-02H

OUT　　R0　　1000 0000 0000 0000　　8000H　　000FH　　输出 0000 0010

==========

测试 SUB 指令　---CY　0->1　　此时 CY=0

SUB　　R1,R2　　0001 0001 0000 0010　　1102H　　0010H　　R1<-R1-R2-CY **CY<-1 R1<-FFH

OUT　　R1　　1000 0001 0000 0000　　8100H　　0011H　　输出 1111 1111

==========

测试 SUB 指令　---CY　1->0　　Z 0->1　　此时 CY=1 Z=0

SUB　　R3,R2　　0001 0011 0000 0010　　1302H　　0012H　　R3<-R3-R2-CY **CY<-0 Z<-1 R3<-00H

OUT　　R3　　1000 0011 0000 0000　　8300H　　0013H　　输出 0000 0000

==========

测试 JMP 指令

JMP	32H	0000 0000 0011 0010	0032H	0014H	跳转到地址 0034H **PC<-0034H

==========

测试 JZ 指令

JZ	R7,02H	0000 1111 0000 0010	0F02H	0032H	向后跳两步

==========

测试 STA 指令

MVI	R0,F0H	0010 0000 1111 0000	20F0H	0034H	R0<-F0H
MVI	R1,01H	0010 0001 0000 0001	2101H	0035H	R1<-01H
STA	R0,A0H	0011 0000 1010 0000	30A0H	0036H	将 R0 的值赋给地址 00A0H **R0->[00A0H]

==========

测试 LDA 指令

OUT	R1	1000 0001 0000 0000	8100H	0037H	输出 0000 0001
LDA	R1,A0H	0011 1001 1010 0000	39A0H	0038H	将地址 00A0H 的值赋给 R1 **R1<-F0H
OUT	R1	1000 0001 0000 0000	8100H	0039H	输出 1111 0000

==========

测试 IN 指令

IN	R5	1000 1101 0000 0000	8D00H	003AH	通过外设输入
OUT	R5	1000 0101 0000 0000	8500H	003BH	

==========

测试 JZ 指令

JZ	R7，FFH	0000 1111 1111 1111	0FFFH	003CH	向前跳一步

==========

第六部分：设计、调试、波形、下载过程中遇到的问题及解决方法

JZ 不明什么时候回写 PC+X，什么时候回写 PC+1

问题描述：

缺少相应控制信号，不能确定如何回写 JZ 的值。

解决方法：

增加控制信号 CtrRwPC 当需要 JZ 回写 PC+X 时 CtrRwPC = ‘1’ 有效，否则 CtrRwPC = ‘0’ 无效。

PC 多次加 1

问题描述：

在元件例化后，总体波形的检查时，发现 PC 无限次加一。

解决方法：

通过对代码的研究，发现 process 中电平触发，由于敏感信号的原因进程会被多次调用，导致 PC 无限次加一。改用时钟沿触发，避免电平触发带来的错误。

93 号错误

问题描述：

在 13.4 工程更换 9.1i 工程时，报错。

解决方法：

增加全局时钟，更改设置。

Out 输出时无论是什么数据输出均是高阻

问题描述：

Out 指令测试时，输出均为 1111 1111。

解决方法：

将代码中寄存器回写赋高阻删除。

第七部分：实验体会

在本次实验的过程中我完成了通过 VHDL 语言编程、FPGA 下载等过程，实现了给定系统的处理器设计。

通过这次的简单 RISC 处理器的设计，我更加深入的理解了 VHDL 语言，能够通过串并行语句进行硬件设计。在使用 process 语句过程中，我感受到了设计的奥秘。process 敏感信号表中添加或删除一个信号，对于设计的硬件电路实物图具有很大的影响。在本次设计中我曾今多次出现过未将所有信号都加入敏感信号表中，导致仿真波形出现了很多除可预测的问题。在添加了敏感信号之后，问题解决。顺序语句的设计的过程中，我深刻地体会了语句的顺序性执行和时间发生并行性。