2018

CPU设计报告

201608010629
冶占清
智能1602

目录

▶ 01

实验要求及设计

▶ 02

仿真结果

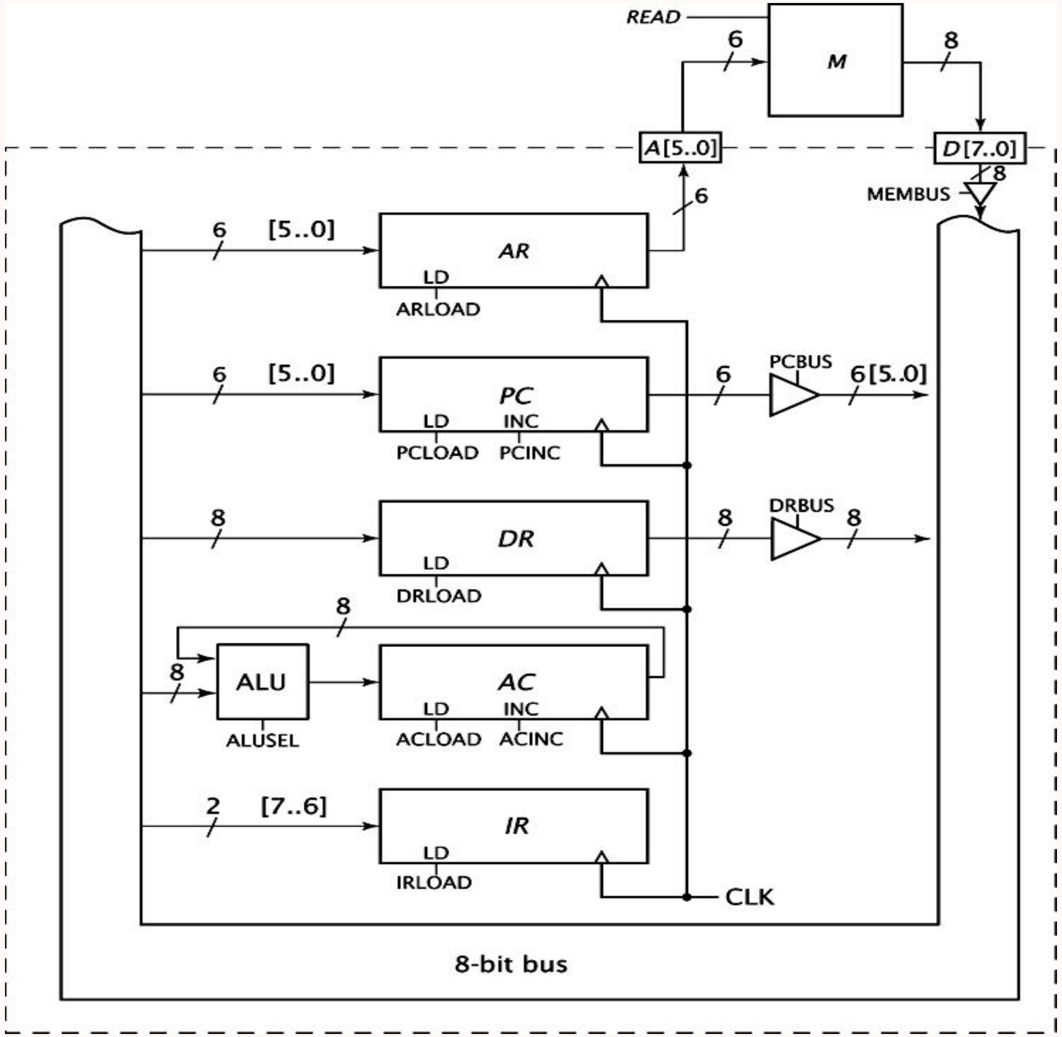
01实验要求及设计

设计规范：

1．64字节的存储空间，每个字节是8位。

2．一个程序员可以访问的寄存器AC（8位累加器）

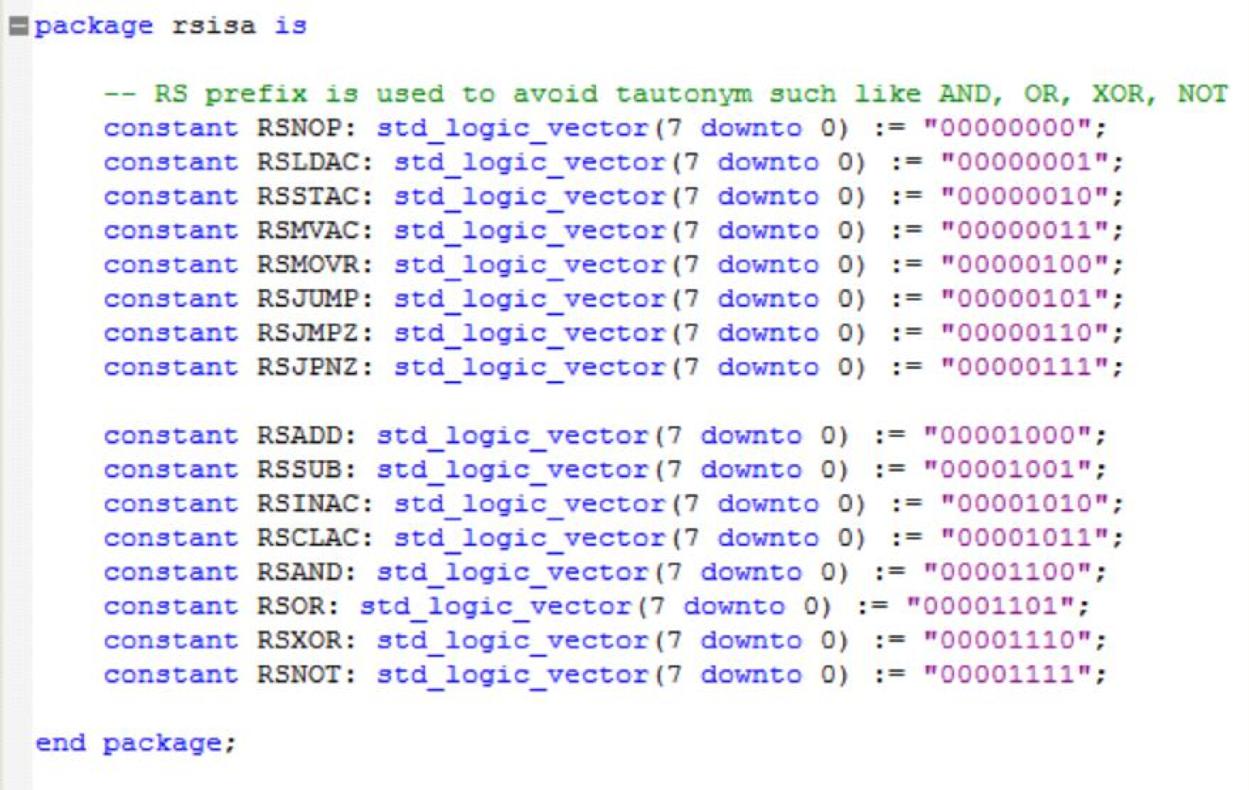
3．指令集（4条指令）



数据通路

01实验要求及设计

rsisa指令集:



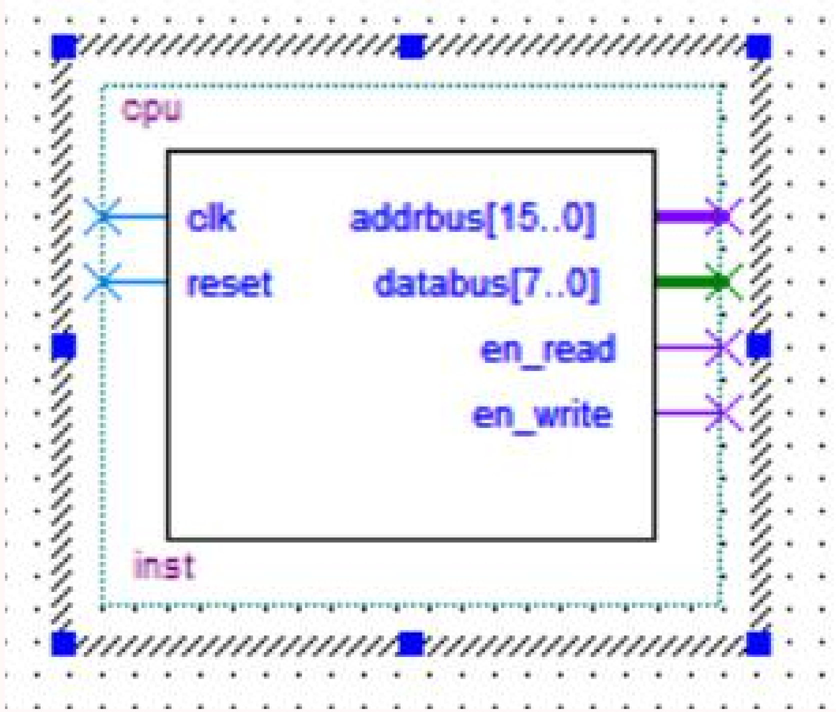
```
package rsisa is

    -- RS prefix is used to avoid tautonym such like AND, OR, XOR, NOT
    constant RSNOP: std_logic_vector(7 downto 0) := "00000000";
    constant RSLDAC: std_logic_vector(7 downto 0) := "00000001";
    constant RSSTAC: std_logic_vector(7 downto 0) := "00000010";
    constant RSMVAC: std_logic_vector(7 downto 0) := "00000011";
    constant RSMOVR: std_logic_vector(7 downto 0) := "00000100";
    constant RSJUMP: std_logic_vector(7 downto 0) := "00000101";
    constant RSJMPZ: std_logic_vector(7 downto 0) := "00000110";
    constant RSJPNZ: std_logic_vector(7 downto 0) := "00000111";

    constant RSADD: std_logic_vector(7 downto 0) := "00001000";
    constant RSSUB: std_logic_vector(7 downto 0) := "00001001";
    constant RSINAC: std_logic_vector(7 downto 0) := "00001010";
    constant RSCLAC: std_logic_vector(7 downto 0) := "00001011";
    constant RSAND: std_logic_vector(7 downto 0) := "00001100";
    constant RSOR: std_logic_vector(7 downto 0) := "00001101";
    constant RSXOR: std_logic_vector(7 downto 0) := "00001110";
    constant RSNOT: std_logic_vector(7 downto 0) := "00001111";

end package;
```

指令集
例如：

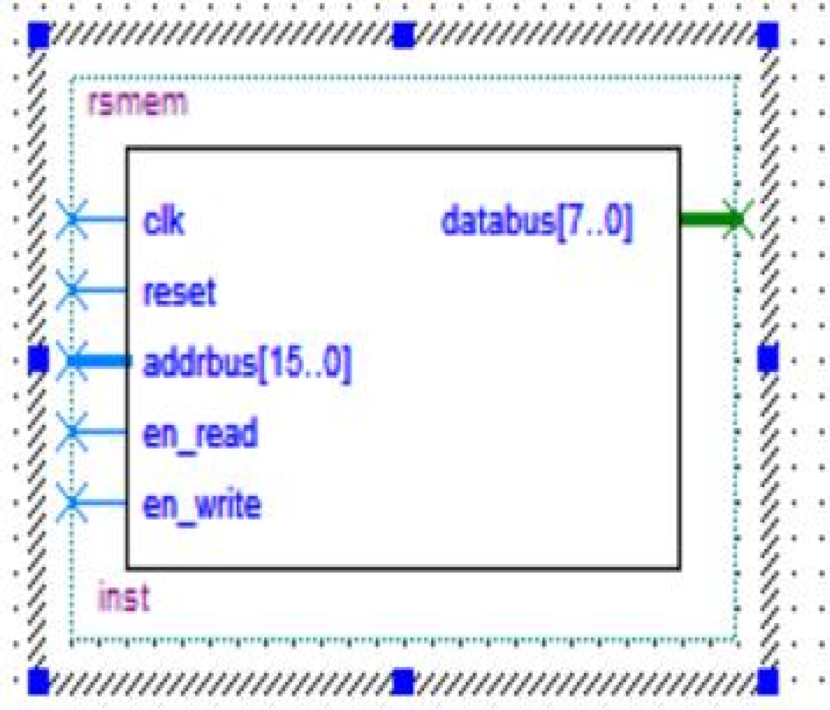
NOP	0000 0000
ADD	0000 1000
AND	0000 1100

01实验要求及设计

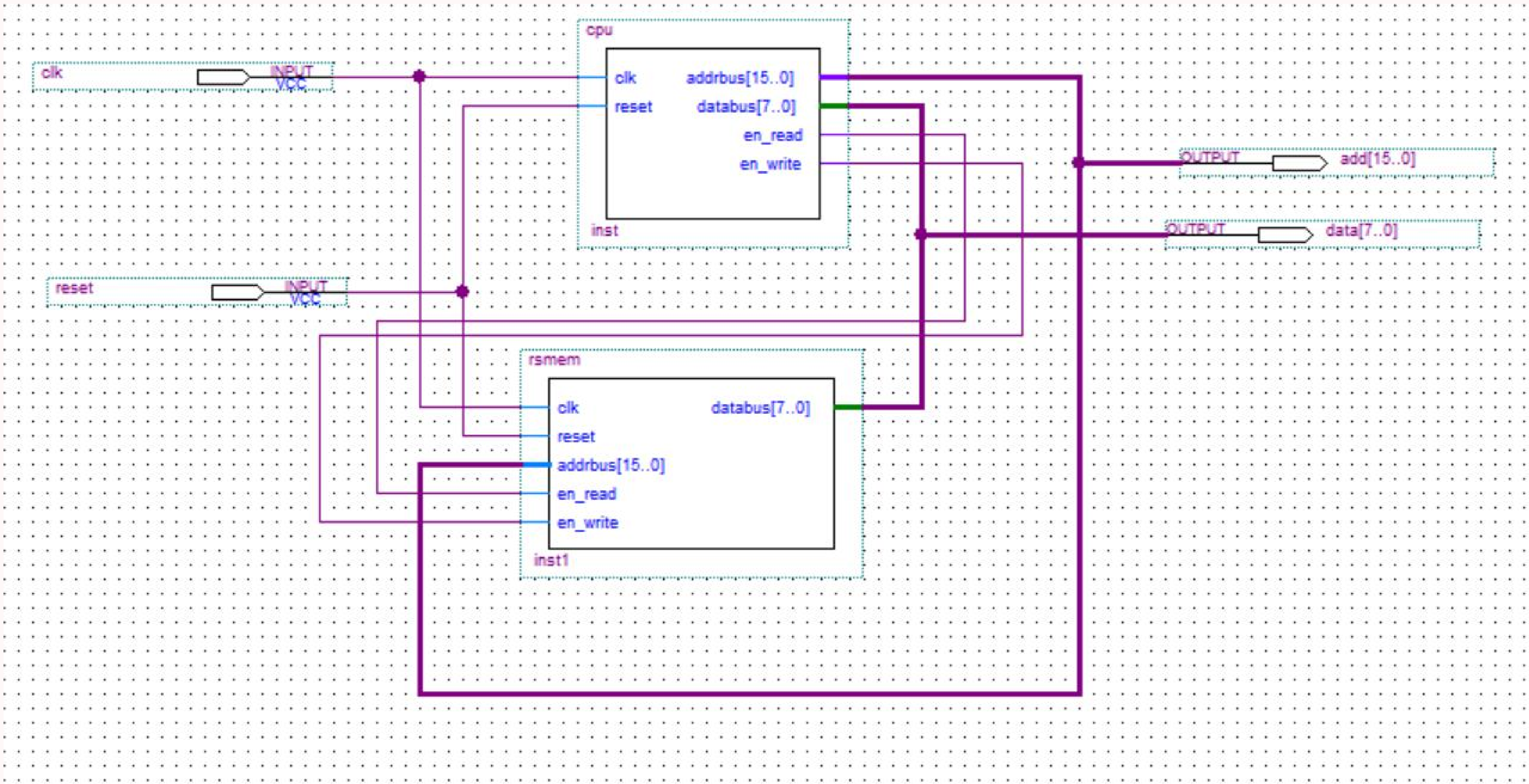
cpu:



mem:



01实验要求及设计


cpu
clk
INPUT
VCC
clk
reset
addrbus[15..0]
databus[7..0]
en_read
en_write
inst
OUTPUT
add[15..0]
OUTPUT
data[7..0]
reset
INPUT
VCC
rsmem
clk
reset
addrbus[15..0]
en_read
en_write
databus[7..0]
inst1

01实验要求及设计

取址，译码:

FETCH1：AR←PC

FETCH2：DR←M，PC←PC+1

FETCH3：IR←DR

FETCH4：AR←PC

执行:

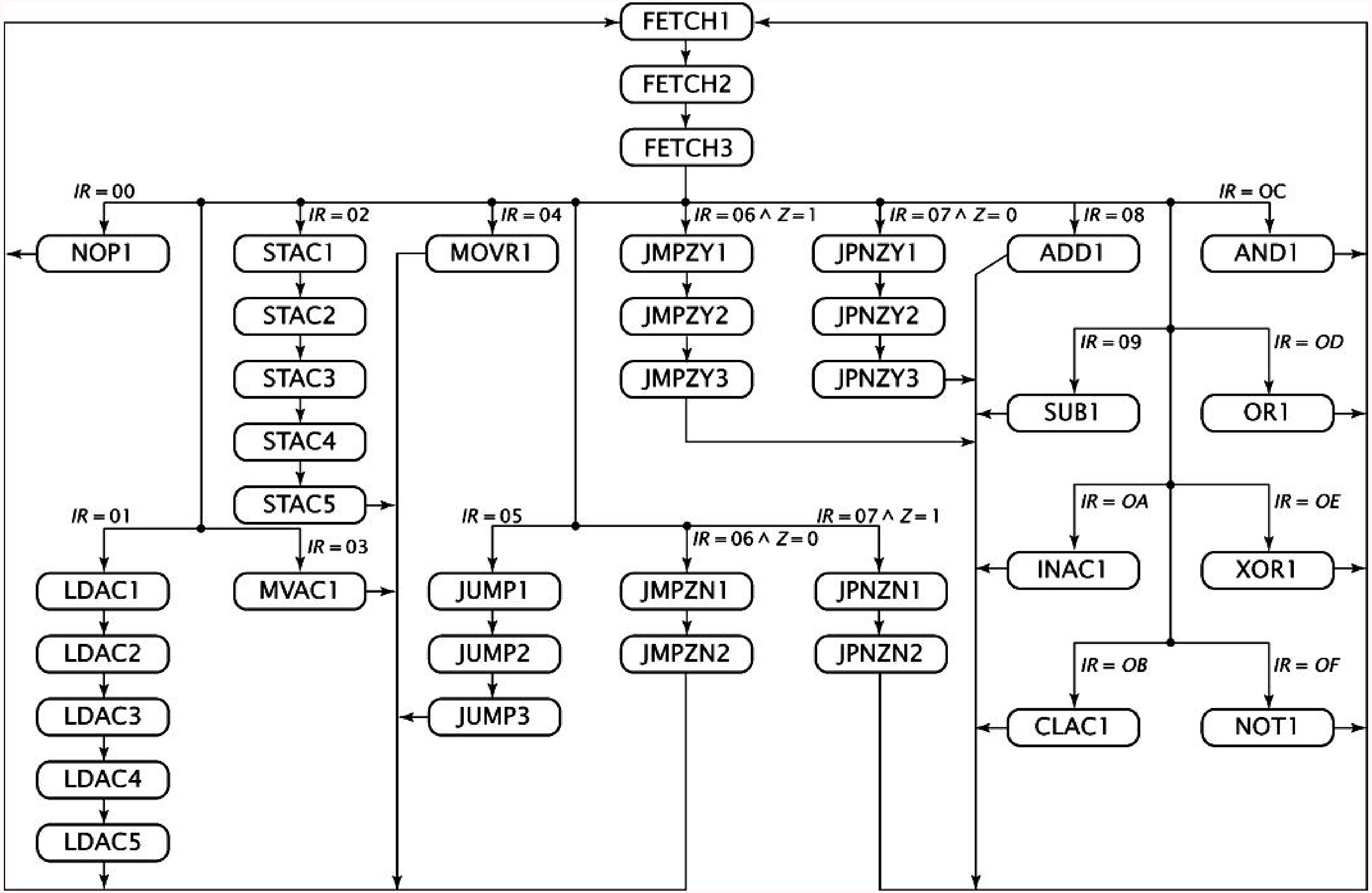
NOP指令

LDAC指令:操作码 地址的低半部分 地址的高半部分

功能：从存储器中获得地址，然后从存储器中获得数据，并把数据装载到累加器中。

....

01实验要求及设计


FETCH1
FETCH2
FETCH3
IR = 00
IR = 02
IR = 04
IR = 06 ∧ Z = 1
IR = 07 ∧ Z = 0
IR = 08
IR = OC
NOP1
STAC1
MOVR1
JMPZY1
JPNZY1
ADD1
AND1
STAC2
JMPZY2
JPNZY2
IR = 09
IR = OD
STAC3
JMPZY3
JPNZY3
SUB1
OR1
STAC4
IR = OA
IR = OE
STAC5
IR = 01
IR = 05
IR = 07 ∧ Z = 1
IR = 03
IR = 06 ∧ Z = 0
LDAC1
MVAC1
JUMP1
JMPZN1
JPNZN1
INAC1
XOR1
LDAC2
JUMP2
JMPZN2
JPNZN2
IR = OB
IR = OF
LDAC3
JUMP3
CLAC1
NOT1
LDAC4
LDAC5

01实验要求及设计

IR寄存器：

ADD指令：（IR=00）

ADD1: DR←M

ADD2: AC←AC＋DR

AND指令：（IR=01）

AND1: DR←M

ND2: AC←AC∧DR

JMP指令：（IR=10）

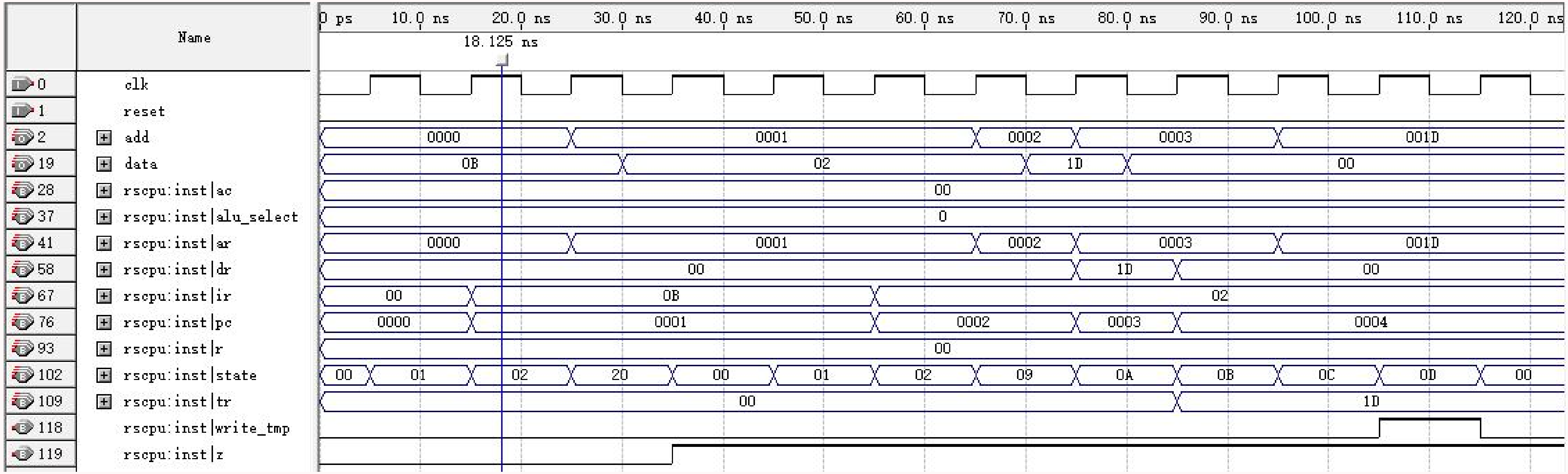
JMP1: PC←DR[5..0]

另外一种选择；**PC←AR**

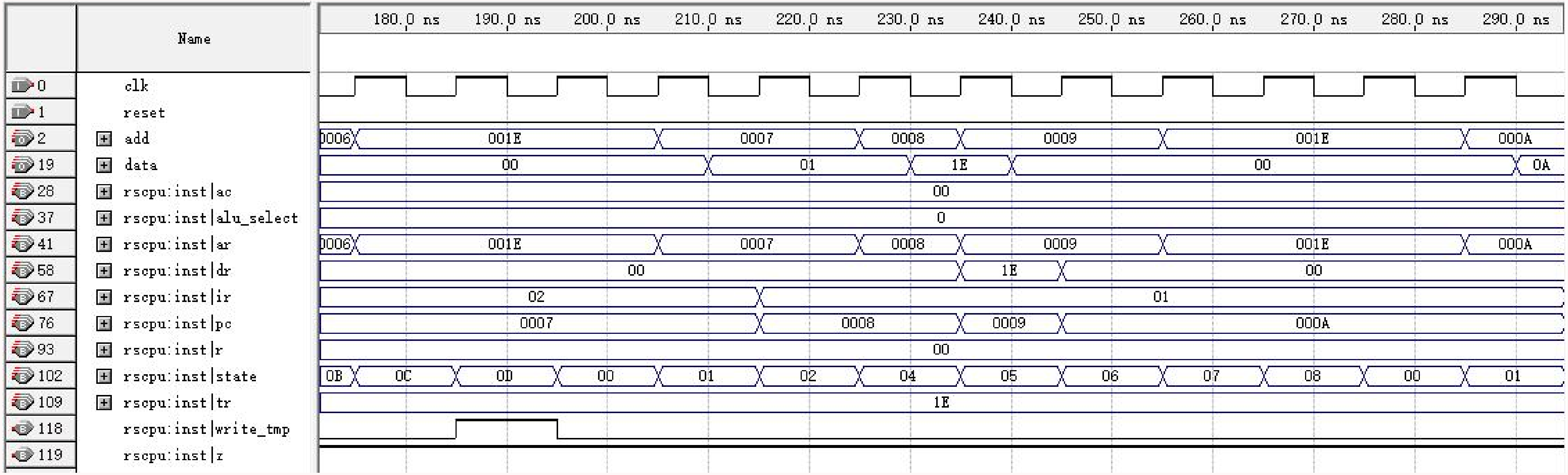
NC指令：（IR=11）

INC1: AC←AC+1

02仿真结果


Name
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
50.0 ns
60.0 ns
70.0 ns
80.0 ns
90.0 ns
100.0 ns
110.0 ns
120.0 ns
18.125 ns
0 clk
1 reset
2 add 0000 0001 0002 0003 001D
19 data 0B 02 1D 00
28 rscpu:inst|ac 00
37 rscpu:inst|alu_select 0
41 rscpu:inst|ar 0000 0001 0002 0003 001D
58 rscpu:inst|dr 00 1D 00
67 rscpu:inst|ir 00 0B 02
76 rscpu:inst|pc 0000 0001 0002 0003 0004
93 rscpu:inst|r 00
102 rscpu:inst|state 00 01 02 20 00 01 02 09 0A 0B 0C 0D 00
109 rscpu:inst|tr 00 1D
118 rscpu:inst|write_tmp
119 rscpu:inst|z

02仿真结果


Name
180.0 ns
190.0 ns
200.0 ns
210.0 ns
220.0 ns
230.0 ns
240.0 ns
250.0 ns
260.0 ns
270.0 ns
280.0 ns
290.0 ns
0 clk
1 reset
2 add 0006 001E 0007 0008 0009 001E 000A
19 data 00 01 1E 00 0A
28 rscpu:inst|ac 00
37 rscpu:inst|alu_select 0
41 rscpu:inst|ar 0006 001E 0007 0008 0009 001E 000A
58 rscpu:inst|dr 00 1E 00
67 rscpu:inst|ir 02 01
76 rscpu:inst|pc 0007 0008 0009 000A
93 rscpu:inst|r 00
102 rscpu:inst|state 0B 0C 0D 00 01 02 04 05 06 07 08 00 01
109 rscpu:inst|tr 1E
118 rscpu:inst|write_tmp
119 rscpu:inst|z

谢谢观看

Thank you for reading my report Thank you for reading my reportThank you for reading my report