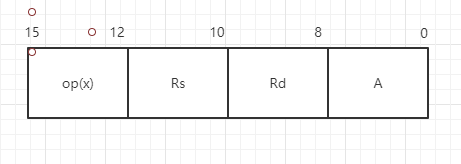
一、课设设计选图

A类题目：输入包含5个整数（有符号数）的数组M，输出最大负数的绝对值。

要求：必须使用RAM存储器读写数据，相应地需要设计对RAM存储器数据的读写指令，以及对RAM中数组操作必须的寄存器间接寻址方式。

2.嵌入式CISC模型计算机采用三数据总线结构的运算器，我们设计的测试代码共有14种操作码，四个通用寄存器，包含立即寻址、寄存器直接寻址、寄存器间接寻址3种寻址方式，采用定长16位RS型指令格式，所以设计的指令格式如下:

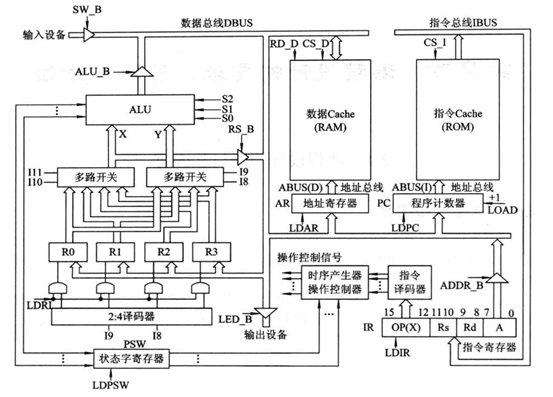


二、系统总体设计

2.1嵌入式CISI模型机系统总体设计

嵌入式CISC模型机的数据通路框图

:



2.2嵌入式CISC系统控制器的逻辑结构框图

三.模型机的指令系统和所有指令的指令格式。

寄存器选定：

|  |  |
| --- | --- |
| Rs或Rd | 选定的寄存器 |
| 0 0 | R0 |
| 0 1 | R1 |
| 1 0 | R2 |
| 1 1 | R3 |

数据的表示格式：

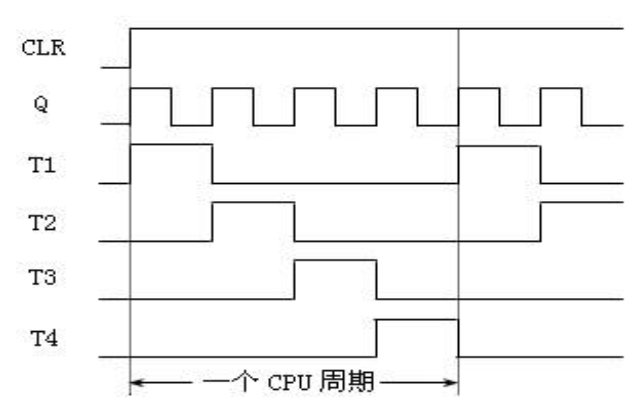
|  |  |
| --- | --- |
| 7 | 6 5 4 3 2 1 0 |
| 符号位 | 数据 |

所有指令格式与指令系统的设计：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 指令助记符 | 指令格式 | | | | 功能 |
| 15 14 13 12 | 11 10 | 9 8 | 7 6 5 4 3 2 1 |
| IN1 Rd | 0 0 0 1 | x x | Rd | x x x x x x x | 输入设备->RO |
| MOV Rd，im | 0 0 1 0 | x x | Rd | im | 立即数->Rd |
| STO1 Rs，(Rd) | 0 0 1 1 | Rs | Rd | x x x x x x x | (Rs)->(Rd) |
| INC Rd | 0 1 0 0 | xx | Rd | x x x x x x x | (Rd)+1->Rd 锁存标志位 |
| DEC Rd | 0 1 0 1 | x x | Rd | x x x x x x x | (Rd)-1->Rd 锁存标志位 |
| TEST Rd | 0 1 1 0 | x x | Rd | x x x x x x x | (Rd)&1 锁存标志位 |
| CMP Rs,Rd | 0 1 1 1 | Rs | RD | addr | Rs-Rd 锁存标志位 |
| JNZ addr | 1 0 0 0 | x x | x x | addr | 不为0，addr->PC |
| LAD (Rs)， Rd | 1 0 0 1 | Rs | Rd | x x x x x x x | （(Rs)）->Rd |
| JNS addr | 1 0 1 0 | x x | x x | addr | SF=0时，addr->PC |
| NEG | 1 0 1 1 | xx | Rd | x x x x x x x | |Rd|->Rd |
| OUT Rs | 1 1 0 0 | Rs | x x | x x x x x x x | (Rs)->输出设备 |
| MOV1 Rs,Rd | 1 1 0 1 | Rs | Rd | x x x x x x x | (Rs)->Rd |

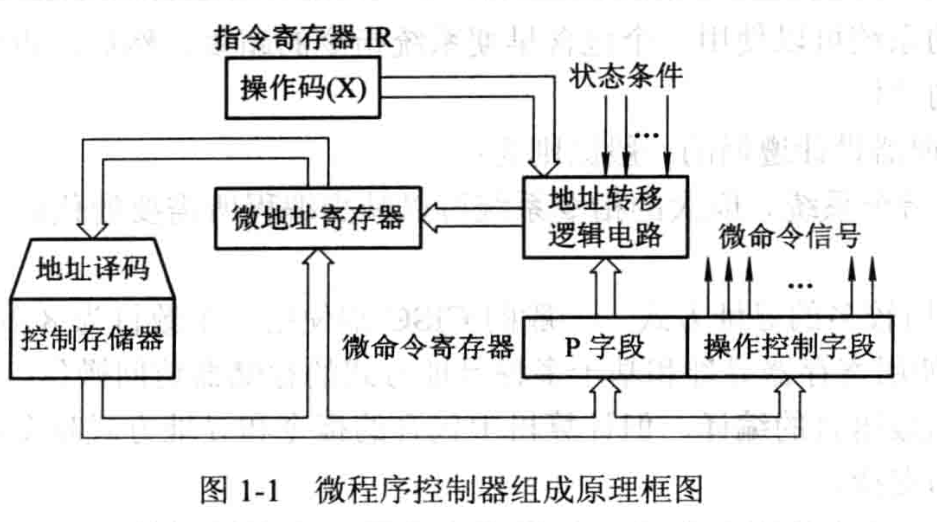
四、时序产生器电路设计

时序电路图：

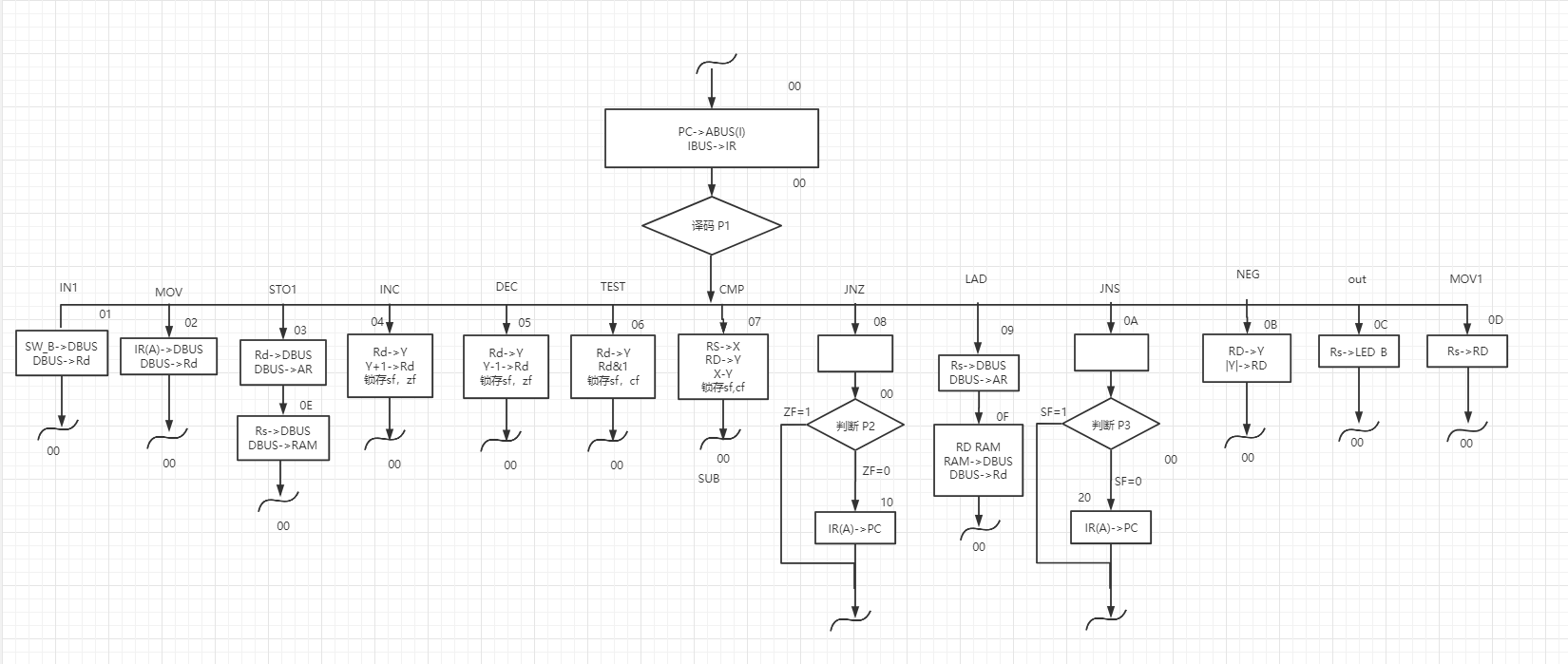


五、微程序流程图设计

5.1微程序控制器的设计



5.2微程序流程图



六、设计操作控制器单元

6.1微指令格式设计

指令流程图设计完成后，开始设计指令和微指令代码表，按照要求，CISC模型机系统使用的微指令采用全水平型微指令,微命令字长24位，其中微命令字段16位，P字段2位，后继微地址6位。

6.2微指令列表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 微地址 | LOAD | LDPC | LDAR | LDIR | LDRI | LDPSW | Rs\_B | S2 | S1 | S0 | ALU\_B | SW\_B | LED\_B | RD\_D | CS\_D | RAM\_B | CS\_I | ADDR\_B | P1 | P2 | P3 | μA5~μA0 |
| 00 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 000000 |
| 01 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 02 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 000000 |
| 03 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 001110 |
| 04 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 05 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 06 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 07 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 08 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 000000 |
| 09 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 001111 |
| 0A | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 000000 |
| 0B | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 0C | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 0D | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 0E | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 0F | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 000000 |
| 10 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 000000 |
| 20 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 000000 |

七、设计单元电路

7.1汇编语言源程序

|  |
| --- |
| MOV R0,00H  MOV R1,05H  INPUT:IN1 R2  STO1 R2,(R0)  INC R0  DEC R1  TEST R1  JNZ INPUT  MOV R0,00H  MOV R1,05H  LAD (R0),R3 :  JUDGE:LAD (R0),R2  TEST R2  JNS YES :判断是否为负数，不为负数跳转  CMP R3,R2  JNS YES ;R3大则跳转  MOV1 R2,R3 :把大的值放到R3  YES:INC R0  DEC R1  TEST R1  JNZ JUDGE  NEG R3  OUT R3 |

7.2机器语言源程序

**汇编语言源程序转换成机器语言源程**

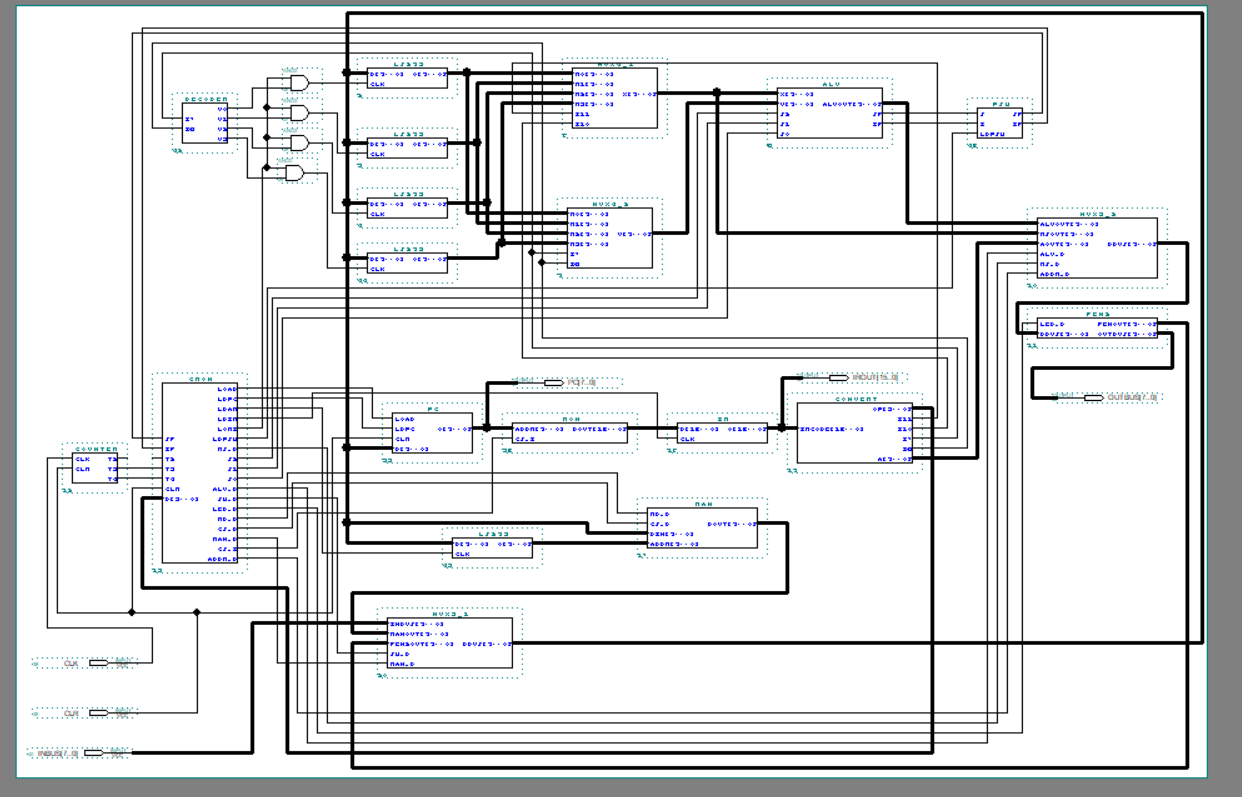
|  |  |  |  |
| --- | --- | --- | --- |
| 地址  （十六进制） | 汇编语言源程序 | 机器语言源程序  （二进制） | 机器语言源程序  （十六进制） |
| 00 | MOV R0,00H | 0010 0000 0000 0000 | 2000 |
| 01 | MOV R1,05H | 0010 0001 0000 0101 | 2105 |
| 02 | INPUT:IN1 R2 | 0001 0010 0000 0000 | 1200 |
| 03 | STO1 R2,(R0) | 0011 1000 0000 0000 | 3800 |
| 04 | INC R0 | 0100 0000 0000 0000 | 4000 |
| 05 | DEC R1 | 0101 0001 0000 0000 | 5100 |
| 06 | TEST R1 | 0110 0001 0000 0000 | 6100 |
| 07 | JNZ INPUT | 1000 0000 0000 0010 | 8002 |
| 08 | MOV R0,00H | 0010 0000 0000 0000 | 2000 |
| 09 | MOV R1,05H | 0010 0001 0000 0101 | 2105 |
| 0A | LAD (R0),R3 | 1001 0011 0000 0000 | 9300 |
| 0B | JUDGE:LAD (R0),R2 | 1001 0010 0000 0000 | 9200 |
| 0C | TEST R2 | 0110 0010 0000 0000 | 6200 |
| 0D | JNS YES | 1010 0000 0001 0001 | A011 |
| 0E | CMP R3,R2 | 0111 1110 0000 0000 | 7E00 |
| 0F | JNS YES | 1010 0000 0001 0001 | A011 |
| 10 | MOV1 R2,R3 | 1101 1011 0000 0000 | DB00 |
| 11 | YES:INC R0 | 0100 0000 0000 0000 | 4000 |
| 12 | DEC R1 | 0101 0001 0000 0000 | 5100 |
| 13 | TEST R1 | 0110 0001 0000 0000 | 6100 |
| 14 | JNZ JUDGE | 1000 0000 0000 1011 | 800B |
| 15 | NEG R3 | 1011 0 011 0000 0000 | B300 |
| 16 | OUT R3 | 1100 1100 0000 0000 | CC00 |

八、 CISC模型机顶层电路图

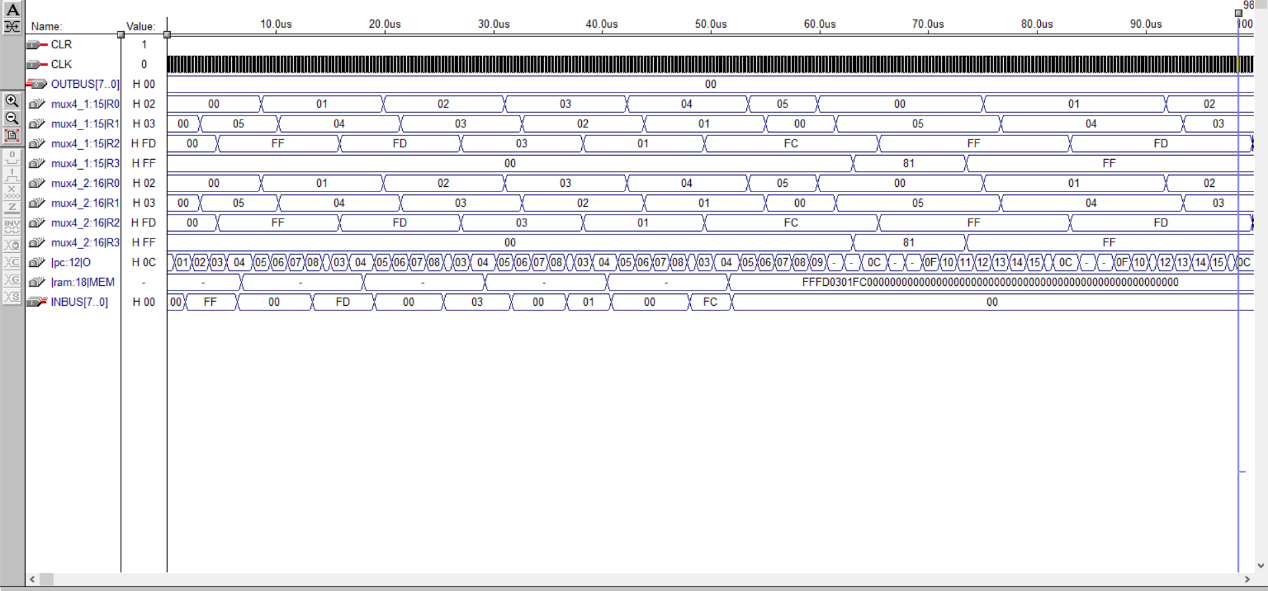
8.1微地址寄存器 AA

8.2微程序控制器CROM

8.3顶层电路图 top



九、仿真测试及结果分析



十、

**ALU功能表**

|  |  |  |  |
| --- | --- | --- | --- |
| S2 | S1 | S0 | 功能 |
| 0 | 0 | 0 | X+Y,修改ZF,SF |
| 0 | 0 | 1 | X-Y,锁存ZF,SF |
| 0 | 1 | 0 | Y+1,修改ZF,SF |
| 0 | 1 | 1 | Y-1,修改ZF,SF |
| 1 | 0 | 0 | 0-Y,锁存ZF,SF |
| 1 | 0 | 1 | Y-0锁存ZF,SF |
| 1 | 1 | 0 | ->Y锁存ZF,SF |

**PC功能表**

|  |  |  |
| --- | --- | --- |
| LOAD | LDPC | 功能 |
| 0 | 1 | 将总线上的内容装入pc |
| 1 | 1 | Pc+1 |
| 1 | 0 | 不装入也不计数 |

十故障分析解决

本次实验过程中遇到多次错误，在微程序指令设计与汇编程序的冲突，及顶层电路的冲突，在不断尝试中开始优化汇编语言来改变对微程序的局部设计以匹配顶层电路。

十一.总结

本次实验内容是设计一台嵌入式CISC模型机，这次试验历史一个小假期和两周在校时间，我们小组在管老师的指导下协作完成本次实验，并最终取得响应结果。在本次实验中我们大都第一次进行线上分工合作完成任务。第一次对线上分工的困难和便捷有相应的了解。同时也对体会到分工协作的好处。在本次实验过程中我们一起对实验进度进行协调发展，在实验前对整个任务进行规划，在任务规程中进行分化，极大提高了工作进度。本次实验我们更加深入的理解计算机工作原理，对计算机系统结构有了更清晰的认识。同时也查询VHDL等相关资料的学习，掌握对max2等软件的应用。在实验过程中我们小组通过讨论对CISC模型机有了更深入的认识，对其工作原理和信号传导更加明确。在试验过程中不断优化指令结构，对实验进行优化，最终在大家共同努力下得以完成。

最后我们小组对给予指导的管老师表示衷心感谢。

十二、参考文献

附录，各单元器件VHDL代码

Addr:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY ADDR IS

PORT(

I15,I14,I13,I12:IN STD\_LOGIC;

CF,ZF,SF,T4,P1,P2,P3:IN STD\_LOGIC;

SE5,SE4,SE3,SE2,SE1,SE0:OUT STD\_LOGIC

);

END ADDR;

ARCHITECTURE A OF ADDR IS

BEGIN

SE5<=NOT((P3 AND ZF AND T4) OR (P3 AND NOT(SF) AND T4));

SE4<=NOT(P2 AND NOT(ZF) AND T4);

SE3<=NOT(P1 AND I15 AND T4);

SE2<=NOT(P1 AND I14 AND T4);

SE1<=NOT(P1 AND I13 AND T4);

SE0<=NOT(P1 AND I12 AND T4);

END A;

Alu:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY ALU IS

PORT(

X:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

Y:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

S2,S1,S0:IN STD\_LOGIC;

ALUOUT:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0);

CF,ZF,SF:OUT STD\_LOGIC

);

END ALU;

ARCHITECTURE A OF ALU IS

SIGNAL XX,YY,TEMP:STD\_LOGIC\_VECTOR(8 DOWNTO 0);

BEGIN

PROCESS

BEGIN

IF(S2='0' AND S1='0' AND S0='0') THEN -- add

XX<='0'&X;

YY<='0'&Y;

TEMP<=XX+YY;

ALUOUT<=TEMP(7 DOWNTO 0);

CF<=TEMP(8);

SF<=TEMP(7);

IF(TEMP="100000000" OR TEMP="000000000") THEN

ZF<='1';

ELSE

ZF<='0';

END IF;

ELSIF(S2='0' AND S1='0' AND S0='1') THEN -- cmp

ALUOUT<=X-Y;

IF(X<Y) THEN

CF<='1';

ZF<='0';

SF<='1';

ELSIF(X=Y) THEN

CF<='0';

ZF<='1';

SF<='0';

ELSE

CF<='0';

ZF<='0';

SF<='0';

END IF;

ELSIF(S2='0' AND S1='1' AND S0='0') THEN --INC

YY<='0'&Y;

TEMP<=YY+1;

ALUOUT<=TEMP(7 DOWNTO 0);

CF<=TEMP(8);

SF<=TEMP(7);

IF(TEMP="100000000"OR TEMP="000000000") THEN

ZF<='1';

ELSE

ZF<='0';

END IF;

ELSIF(S2='0' AND S1='1' AND S0='1') THEN --DEC

YY<='0'&Y;

TEMP<=YY-1;

ALUOUT<=TEMP(7 DOWNTO 0);

CF<=TEMP(8);

SF<=TEMP(7);

IF(TEMP="100000000"OR TEMP="000000000") THEN

ZF<='1';

ELSE

ZF<='0';

END IF;

ELSIF(S2='1' AND S1='0' AND S0='0') THEN --NEG

YY<='0'&Y;

TEMP<=0-Y;

ALUOUT<=TEMP(7 DOWNTO 0);

CF<=TEMP(8);

SF<=TEMP(7);

IF(TEMP="100000000"OR TEMP="000000000") THEN

ZF<='1';

ELSE

ZF<='0';

END IF;

ELSIF(S2='1' AND S1='0' AND S0='1') THEN --TEST

YY<='0'&Y;

TEMP<=YY-0;

ALUOUT<=TEMP(7 DOWNTO 0);

CF<=TEMP(8);

SF<=TEMP(7);

IF(TEMP="100000000"OR TEMP="000000000") THEN

ZF<='1';

ELSE

ZF<='0';

END IF;

ELSIF(S2='1' AND S1='1' AND S0='0') THEN --Rd->BUS

ALUOUT<=Y;

ELSE

ALUOUT<="00000000";

CF<='0';

ZF<='0';

SF<='0';

END IF;

END PROCESS;

END A;

Comtrom:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY CONTROM IS

PORT(

ADDR:IN STD\_LOGIC\_VECTOR(5 DOWNTO 0);

UA:OUT STD\_LOGIC\_VECTOR(5 DOWNTO 0);

D:OUT STD\_LOGIC\_VECTOR(20 DOWNTO 0)

);

END CONTROM;

ARCHITECTURE A OF CONTROM IS

SIGNAL DATAOUT:STD\_LOGIC\_VECTOR(26 DOWNTO 0);

BEGIN

PROCESS(ADDR)

BEGIN

CASE ADDR IS

WHEN "000000" => DATAOUT<="110100100011111101100000000"; --取值

WHEN "000001" => DATAOUT<="100010100010111111000000000"; --IN1

WHEN "000010" => DATAOUT<="100010100011111110000000000"; -- MOV

WHEN "000011" => DATAOUT<="101000111001111111000001110"; --STO1

WHEN "000100" => DATAOUT<="100011101001111111000000000"; --INC

WHEN "000101" => DATAOUT<="100011101101111111000000000"; --DEC

WHEN "000110" => DATAOUT<="100001110111111111000000000"; --TEST

WHEN "000111" => DATAOUT<="100001100111111111000000000"; --CMP

WHEN "001000" => DATAOUT<="100000100011111111010000000"; --JNZ

WHEN "001001" => DATAOUT<="101000000011111111000001111"; --LAD

WHEN "001010" => DATAOUT<="100000100011111111001000000"; --JNS

WHEN "001011" => DATAOUT<="100011110001111111000000000"; --NEG

WHEN "001100" => DATAOUT<="100000000011011111000000000"; --OUT

WHEN "001101" => DATAOUT<="100010000011111111000000000"; --MOV1

WHEN "001110" => DATAOUT<="100000000011100111000000000"; --写ram

WHEN "001111" => DATAOUT<="100010100011110011000000000"; --读ram

WHEN "010000" => DATAOUT<="010000100011111110000000000"; -- IR->PC(ZF)

WHEN "100000" => DATAOUT<="010000100011111110000000000"; -- IR->PC(SF)

WHEN OTHERS => DATAOUT<="100000100011111111000000000";

END CASE;

UA(5 DOWNTO 0)<=DATAOUT(5 DOWNTO 0);

D(20 DOWNTO 0)<=DATAOUT(26 DOWNTO 6);

END PROCESS;

END A;

Comvert:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY CONVERT IS

PORT(

IRCODE:IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

OP:OUT STD\_LOGIC\_VECTOR(3 DOWNTO 0);

I11,I10,I9,I8:OUT STD\_LOGIC;

A:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END CONVERT;

ARCHITECTURE A OF CONVERT IS

BEGIN

OP<=IRCODE(15 DOWNTO 12);

I11<=IRCODE(11);

I10<=IRCODE(10);

I9<=IRCODE(9);

I8<=IRCODE(8);

A<=IRCODE(7 DOWNTO 0);

END A;

Counter:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY COUNTER IS

PORT(

CLK,CLR:IN STD\_LOGIC;

T2,T3,T4:OUT STD\_LOGIC

);

END COUNTER;

ARCHITECTURE A OF COUNTER IS

SIGNAL X:STD\_LOGIC\_VECTOR(1 DOWNTO 0);

BEGIN

PROCESS(CLK,CLR)

BEGIN

IF(CLR='0') THEN

T2<='0';

T3<='0';

T4<='0';

X<="00";

ELSIF(CLK'EVENT AND CLK='1') THEN

X<=X+1;

T2<=(NOT X(1)) AND X(0);

T3<=X(1) AND (NOT X(0));

T4<=X(1) AND X(0);

END IF;

END PROCESS;

END A;

Decoder:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY DECODER IS

PORT(

I9,I8:IN STD\_LOGIC;

Y0,Y1,Y2,Y3: OUT STD\_LOGIC

);

END DECODER;

ARCHITECTURE A OF DECODER IS

BEGIN

PROCESS

BEGIN

IF(I9='0' AND I8='0') THEN

Y0<='1';

Y1<='0';

Y2<='0';

Y3<='0';

ELSIF(I9='0' AND I8='1') THEN

Y0<='0';

Y1<='1';

Y2<='0';

Y3<='0';

ELSIF(I9='1' AND I8='0') THEN

Y0<='0';

Y1<='0';

Y2<='1';

Y3<='0';

ELSE

Y0<='0';

Y1<='0';

Y2<='0';

Y3<='1';

END IF;

END PROCESS;

END A;

F1:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY F1 IS

PORT(

UA5,UA4,UA3,UA2,UA1,UA0:IN STD\_LOGIC;

D:OUT STD\_LOGIC\_VECTOR(5 DOWNTO 0)

);

END F1;

ARCHITECTURE A OF F1 IS

BEGIN

D(5)<=UA5;

D(4)<=UA4;

D(3)<=UA3;

D(2)<=UA2;

D(1)<=UA1;

D(0)<=UA0;

END A;

F2:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY F2 IS

PORT(

D:IN STD\_LOGIC\_VECTOR(5 DOWNTO 0);

UA5,UA4,UA3,UA2,UA1,UA0:OUT STD\_LOGIC

);

END F2;

ARCHITECTURE A OF F2 IS

BEGIN

UA5<=D(5);

UA4<=D(4);

UA3<=D(3);

UA2<=D(2);

UA1<=D(1);

UA0<=D(0);

END A;

F3:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY F3 IS

PORT(

D:IN STD\_LOGIC\_VECTOR(3 DOWNTO 0);

I15,I14,I13,I12:OUT STD\_LOGIC

);

END F3;

ARCHITECTURE A OF F3 IS

BEGIN

I15<=D(3);

I14<=D(2);

I13<=D(1);

I12<=D(0);

END A;

Fen2:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY FEN2 IS

PORT(

LED\_B:IN STD\_LOGIC;

DBUS:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

FENOUT,OUTBUS:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END FEN2;

ARCHITECTURE A OF FEN2 IS

BEGIN

PROCESS

BEGIN

IF(LED\_B='0') THEN

OUTBUS<=DBUS;

ELSE

FENOUT<=DBUS;

END IF;

END PROCESS;

END A;

Ir:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY IR IS

PORT(

D:IN STD\_LOGIC\_VECTOR(15 DOWNTO 0);

CLK:IN STD\_LOGIC;

O:OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0)

);

END IR;

ARCHITECTURE A OF IR IS

BEGIN

PROCESS(CLK)

BEGIN

IF(CLK'EVENT AND CLK='1') THEN

O<=D;

END IF;

END PROCESS;

END A;

Is273: LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY LS273 IS

PORT(

D:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

CLK:IN STD\_LOGIC;

O:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END LS273;

ARCHITECTURE A OF LS273 IS

BEGIN

PROCESS(CLK)

BEGIN

IF(CLK'EVENT AND CLK='1')THEN

O<=D;

END IF;

END PROCESS;

END A;

Mcommand:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY MCOMMAND IS

PORT(

T2,T3,T4:IN STD\_LOGIC;

O:IN STD\_LOGIC\_VECTOR(20 DOWNTO 0);

P1,P2,P3,LOAD,LDPC,LDAR,LDIR,LDRI,LDPSW,RS\_B,S2,S1,S0,ALU\_B,SW\_B,LED\_B,RD\_B,CS\_D,RAM\_B,CS\_I,ADDR\_B:OUT STD\_LOGIC

);

END MCOMMAND;

ARCHITECTURE A OF MCOMMAND IS

SIGNAL DATAOUT:STD\_LOGIC\_VECTOR(20 DOWNTO 0);

BEGIN

PROCESS(T2)

BEGIN

IF(T2'EVENT AND T2='1') THEN

DATAOUT(20 DOWNTO 0)<=O(20 DOWNTO 0);

END IF;

P3<=DATAOUT(0);

P2<=DATAOUT(1);

P1<=DATAOUT(2);

ADDR\_B<=DATAOUT(3);

CS\_I<=DATAOUT(4);

RAM\_B<=DATAOUT(5);

CS\_D<=NOT(NOT DATAOUT(6) AND T3);

RD\_B<=NOT(NOT DATAOUT(7) AND (T2 OR T3));

LED\_B<=DATAOUT(8);

SW\_B<=DATAOUT(9);

ALU\_B<=DATAOUT(10);

S0<=DATAOUT(11);

S1<=DATAOUT(12);

S2<=DATAOUT(13);

RS\_B<=DATAOUT(14);

LDPSW<=DATAOUT(15) AND T4;

LDRI<=DATAOUT(16) AND T4;

LDIR<=DATAOUT(17) AND T3;

LDAR<=DATAOUT(18) AND T3;

LDPC<=DATAOUT(19) AND T4;

LOAD<=DATAOUT(20);

END PROCESS;

END A;

Mmm:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MMM IS

PORT(

SE:IN STD\_LOGIC;

CLK:IN STD\_LOGIC;

D:IN STD\_LOGIC;

CLR:IN STD\_LOGIC;

UA:OUT STD\_LOGIC

);

END MMM;

ARCHITECTURE A OF MMM IS

BEGIN

PROCESS(CLK,CLR,SE)

BEGIN

IF(CLR='0') THEN

UA<='0';

ELSIF(SE='0') THEN

UA<='1';

ELSIF(CLK'EVENT AND CLK='1') THEN

UA<=D;

END IF;

END PROCESS;

END A;

Mux3-1:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MUX3\_1 IS

PORT(

INBUS,RAMOUT,FEN2OUT:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

SW\_B,RAM\_B:STD\_LOGIC;

DBUS:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END MUX3\_1;

ARCHITECTURE A OF MUX3\_1 IS

BEGIN

PROCESS

BEGIN

IF(SW\_B='0') THEN

DBUS<=INBUS;

ELSIF(RAM\_B='0') THEN

DBUS<=RAMOUT;

ELSE

DBUS<=FEN2OUT;

END IF;

END PROCESS;

END A;

Mux3-2:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MUX3\_2 IS

PORT(

ALUOUT,RSOUT,AOUT:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

ALU\_B,RS\_B,ADDR\_B:STD\_LOGIC;

DBUS:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END MUX3\_2;

ARCHITECTURE A OF MUX3\_2 IS

BEGIN

PROCESS

BEGIN

IF(ALU\_B='0') THEN

DBUS<=ALUOUT;

ELSIF(RS\_B='0') THEN

DBUS<=RSOUT;

ELSIF(ADDR\_B='0') THEN

DBUS<=AOUT;

ELSE

DBUS<="00000000";

END IF;

END PROCESS;

END A;

Mux4-1:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MUX4\_1 IS

PORT(

I11,I10:IN STD\_LOGIC;

R0,R1,R2,R3:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

X:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END MUX4\_1;

ARCHITECTURE A OF MUX4\_1 IS

BEGIN

PROCESS

BEGIN

IF(I11='0' AND I10='0') THEN

X<=R0;

ELSIF(I11='0' AND I10='1') THEN

X<=R1;

ELSIF(I11='1' AND I10='0') THEN

X<=R2;

ELSE

X<=R3;

END IF;

END PROCESS;

END A;

Mux4-2:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY MUX4\_2 IS

PORT(

R0,R1,R2,R3:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

I9,I8:IN STD\_LOGIC;

Y:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END MUX4\_2;

ARCHITECTURE A OF MUX4\_2 IS

BEGIN

PROCESS

BEGIN

IF(I9='0' AND I8='0') THEN

Y<=R0;

ELSIF(I9='0' AND I8='1') THEN

Y<=R1;

ELSIF(I9='1' AND I8='0') THEN

Y<=R2;

ELSE

Y<=R3;

END IF;

END PROCESS;

END A;

Pc:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY PC IS

PORT(

LOAD,LDPC,CLR:IN STD\_LOGIC;

D:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

O:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END PC;

ARCHITECTURE A OF PC IS

SIGNAL QOUT:STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

PROCESS(LDPC,CLR,LOAD)

BEGIN

IF(CLR='0') THEN

QOUT<="00000000";

ELSIF(LDPC'EVENT AND LDPC='1') THEN

IF(LOAD='0') THEN

QOUT<=D;

ELSE

QOUT<=QOUT+1;

END IF;

END IF;

END PROCESS;

O<=QOUT;

END A;

Psw:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

ENTITY PSW IS

PORT(

LDPSW:IN STD\_LOGIC;

CF\_IN,ZF\_IN,SF\_IN:IN STD\_LOGIC;

CF,ZF,SF:OUT STD\_LOGIC

);

END PSW;

ARCHITECTURE A OF PSW IS

BEGIN

PROCESS(LDPSW)

BEGIN

IF(LDPSW'EVENT AND LDPSW='1') THEN

CF<=CF\_IN;

ZF<=ZF\_IN;

SF<=SF\_IN;

END IF;

END PROCESS;

END A;

Ram:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY RAM IS

PORT(

CS\_D,RD\_D:IN STD\_LOGIC;

DIN:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

ADDR:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

DOUT:OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0)

);

END RAM;

ARCHITECTURE A OF RAM IS

TYPE MEMORY IS ARRAY(0 TO 31) OF STD\_LOGIC\_VECTOR(7 DOWNTO 0);

BEGIN

PROCESS(CS\_D)

VARIABLE MEM:MEMORY;

BEGIN

IF(CS\_D'EVENT AND CS\_D='0') THEN

IF(RD\_D='0')THEN

MEM(CONV\_INTEGER(ADDR(4 DOWNTO 0))):=DIN;

ELSE

DOUT<=MEM(CONV\_INTEGER(ADDR(4 DOWNTO 0)));

END IF;

END IF;

END PROCESS;

END A;

Rom:

LIBRARY IEEE;

USE IEEE.STD\_LOGIC\_1164.ALL;

USE IEEE.STD\_LOGIC\_ARITH.ALL;

USE IEEE.STD\_LOGIC\_UNSIGNED.ALL;

ENTITY ROM IS

PORT(

DOUT:OUT STD\_LOGIC\_VECTOR(15 DOWNTO 0);

ADDR:IN STD\_LOGIC\_VECTOR(7 DOWNTO 0);

CS\_I:IN STD\_LOGIC

);

END ROM;

ARCHITECTURE A OF ROM IS

BEGIN

DOUT<="0010000000000000" WHEN ADDR="00000000" AND CS\_I='0' ELSE --MOV R0,00H

"0010000100000101" WHEN ADDR="00000001" AND CS\_I='0' ELSE --MOV R1,05H

"0001001000000000" WHEN ADDR="00000010" AND CS\_I='0' ELSE --INPUT:IN1 R2

"0011100000000000" WHEN ADDR="00000011" AND CS\_I='0' ELSE --STO1 R2,(R0)

"0100000000000000" WHEN ADDR="00000100" AND CS\_I='0' ELSE --INC R0

"0101000100000000" WHEN ADDR="00000101" AND CS\_I='0' ELSE --DEC R1

"0110000100000000" WHEN ADDR="00000110" AND CS\_I='0' ELSE --TEST R1

"1000000000000010" WHEN ADDR="00000111" AND CS\_I='0' ELSE --JNZ INPUT

"0010000000000000" WHEN ADDR="00001000" AND CS\_I='0' ELSE --MOV R0,00H

"0010000100000101" WHEN ADDR="00001001" AND CS\_I='0' ELSE --MOV R1,05H

"1001001100000000" WHEN ADDR="00001010" AND CS\_I='0' ELSE --LAD (R0),R3

"1001001000000000" WHEN ADDR="00001011" AND CS\_I='0' ELSE --JUDGE:LAD (R0),R2

"0110001000000000" WHEN ADDR="00001100" AND CS\_I='0' ELSE --TEST R2

"1010000000010001" WHEN ADDR="00001101" AND CS\_I='0' ELSE --JNS YES

"0111111000000000" WHEN ADDR="00001110" AND CS\_I='0' ELSE --CMP R3,R2

"1010000000010001" WHEN ADDR="00001111" AND CS\_I='0' ELSE --JNS YES

"1101101100000000" WHEN ADDR="00010000" AND CS\_I='0' ELSE --MOV1 R2,R3

"0100000000000000" WHEN ADDR="00010001" AND CS\_I='0' ELSE --YES:INC R0

"0101000100000000" WHEN ADDR="00010010" AND CS\_I='0' ELSE --DEC R1

"0110000100000000" WHEN ADDR="00010011" AND CS\_I='0' ELSE --TEST R1

"1000000000001011" WHEN ADDR="00010100" AND CS\_I='0' ELSE --JNZ JUDGE

"1011001100000000" WHEN ADDR="00010101" AND CS\_I='0' ELSE --NEG R3

"1100110000000000" WHEN ADDR="00010110" AND CS\_I='0' ELSE --OUT R3

"0000000000000000";

END A;