题目： 设计实现一个字长8位的CPU

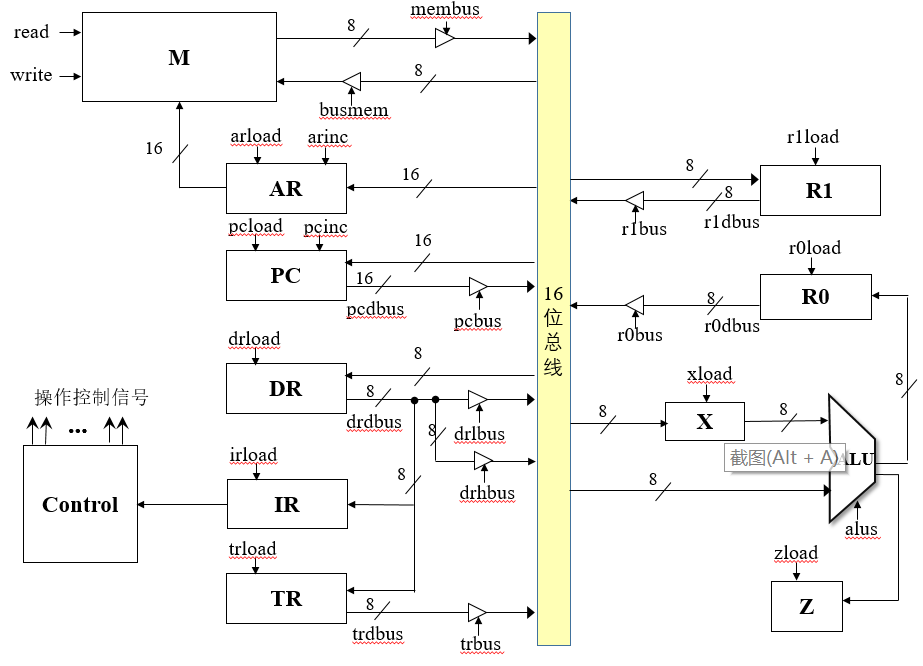
环境： Quartus II、ModelSim-Altera

## 内容：

设计实现一个字长8位的CPU系统，所设计的系统能调试通过，通过运行自行编写的程序进行仿真测试，通过检查程序结果的正确性来判断所设计计算机系统的正确性。

## 设计过程：

* 数据通路



* 指令系统：指令格式和功能、操作码编码

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 指令 | 格式 | 功能 |
| 1 | NOP | 00000000 | No Operation |
| 2 | ADD | 00010000 | R0<— R0+R1 |
| 3 | SUB | 00100000 | R0<— R0-R1 |
| 4 | AND | 00110000 | R0<— R0&&R1 |
| 5 | OR | 01000000 | R0<— R0||R1 |
| 6 | XOR | 01010000 | R0<— R0^R1 |
| 7 | INC | 01100000 | R0<— R0+1 |
| 8 | NOT | 01110000 | R0<— ~R0 |
| 9 | CLR | 10000000 | R0<—0 |
| 10 | SHR | 10010000 | R0<—R0/2 |
| 11 | MVR | 10100000 | R1<— R0 |
| 12 | JMP | 10110000 A | GOTO A |
| 13 | JPZ | 11000000 A | if(Z==1)GOTO A |
| 14 | JPNZ | 11010000 A | If(Z==0)GOTO A |
| 15 | LAD | 11100000 A | R0<— M[A] |
| 16 | STO | 11110000 A | M[A] <—R0 |

* 指令执行：指令的执行流程图（状态图）、指令的执行节拍、指令执行的数据流和控制流

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 节拍 | 状态 | 微操作序列（数据流） | 控制信号（控制流） |
| 取指令 | T0 | fetch1 | AR<—PC | pcbus、arload |
| T1 | fetch2 | DR<—M，PC<—PC+1 | read、membus、drload、pcinc |
| T2 | fetch3 | IR<—DR，AR<—PC | irload、pcbus、arload |
| ADD | T3 | ADD1 | X<—R0 | r0bus、xload |
| T4 | ADD2 | R0<—X+R1 | r1bus、alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| SUB | T3 | SUB1 | X<—R0 | r0bus、xload |
| T4 | SUB2 | R0<—X-R1 | r1bus、alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| AND | T3 | AND1 | X<—R0 | r0bus、xload |
|  | T4 | AND2 | R0<—X&&R1 | r1bus、alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| OR | T3 | OR1 | X<—R0 | r0bus、xload |
| T4 | OR2 | R0<—X||R1 | r1bus、alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| XOR | T3 | XOR1 | X<—R0 | r0bus、xload |
| T4 | XOR2 | R0<—X^R1 | r1bus、alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| INC | T3 | INC1 | X<—R0 | r0bus、xload |
| T4 | INC2 | R0<—X+1 | alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| NOT | T3 | NOT1 | X<—R0 | r0bus、xload |
| T4 | NOT2 | R0<—~X | alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| CLR | T3 | CLR1 | R0<—0 | alus、r0load、zload |
| SHR | T3 | SHR1 | X<—R0 | r0bus、xload |
| T4 | SHR2 | R0<—X/2 | alus、r0load、zload（如果需要z标志寄存器，则有该控制信号） |
| MVR | T3 | MVR1 | R1<—R0 | r0bus、r1load |
| JMP | T3 | JMP1 | DR<—M，AR<—AR+1 | read、membus、drload、arinc |
| T4 | JMP 2 | TR<—DR，DR<—M | trload、read、membus、drload |
| T5 | JMP 3 | PC<—DR，TR | drhbus、trbus、pcload |
| JPZ  if(z==1) | T3 | JPZ1 | DR<—M，AR<—AR+1 | read、membus、drload、arinc |
| T4 | JPZ 2 | TR<—DR，DR<—M | trload、read、membus、drload |
| T5 | JPZ 3 | PC<—DR，TR | drhbus、trbus、pcload |
| JPZ  if(z!=1) | T3 | JPZ1 | PC<—PC+1 | pcinc |
| T4 | JPZ 2 | PC<—PC+1 | pcinc |
| JPNZ  if(z!=1) | T3 | JPNZ1 | DR<—M，AR<—AR+1 | read、membus、drload、arinc |
| T4 | JPNZ 2 | TR<—DR，DR<—M | trload、read、membus、drload |
| T5 | JPNZ 3 | PC<—DR，TR | drhbus、trbus、pcload |
| JPNZ  if(z==1) | T3 | JPNZ1 | PC<—PC+1 | pcinc |
| T4 | JPNZ 2 | PC<—PC+1 | pcinc |
| LAD | T3 | LAD1 | DR<—M，PC<—PC+1，  AR<—AR+1 | read、membus、drload、pcinc、arinc |
| T4 | LAD2 | TR<—DR，DR<—M，  PC<—PC+1 | trload、read、membus、drload、pcinc |
| T5 | LAD3 | AR<—DR，TR | drhbus、trbus、arload |
| T6 | LAD4 | DR<—M | read、membus、drload |
| T7 | LAD5 | R0<—DR | drlbus、r0load、alus |
| STO | T3 | STO1 | DR<—M，PC<—PC+1，  AR<—AR+1 | read、membus、drload、pcinc、arinc |
| T4 | STO2 | TR<—DR，DR<—M，  PC<—PC+1 | trload、read、membus、drload、pcinc |
| T5 | STO3 | AR<—DR，TR | drhbus、trbus、arload |
| T6 | STO4 | M<---R0 | r0bus、write、busmem |

* ALU：ALU的功能和选择运算的操作信号的逻辑表达式

1. module alu(alus,x, bus, dout);
2. input [3:0]alus;
3. input [7:0]x;
4. input [7:0]bus;
5. output [7:0]dout;
6. reg [7:0]dout;
7. always @(alus or x or bus)
8. begin
9. case(alus)
10. 4'b0000: dout=8'b00000000;
11. 4'b0001: dout=x + bus;
12. 4'b0010: dout=x - bus;
13. 4'b0011: dout=x & bus;
14. 4'b0100: dout=x | bus;
15. 4'b0101: dout=x ^ bus;
16. 4'b0110: dout=x + 8'b00000001;
17. 4'b0111: dout= ~ x;
18. 4'b1000: dout=x - 8'b00000001;
19. 4'b1001: dout=(x >> 1);
20. 4'b1010: dout=bus + 8'b00000000;
21. default:dout = 8'bx;
22. endcase
23. end
24. endmodule

* 控制器：所有的控制信号的逻辑表达式

1. assign pcbus=fetch1||fetch3;
2. assign r0bus=add1||sub1||and1||or1||xor1||inc1||not1||shr1||mvr1||sto4;
3. assign r1bus=add2||sub2||and2||or2||xor2;
4. assign drlbus=lad5;
5. assign drhbus=lad3||sto3||jmp3||(jpz3&&z)||(jpnz3&&(!z));
6. assign trbus=jmp3||(jpz3&&z)||(jpnz3&&(!z))||lad3||sto3;
7. assign membus=fetch2||jmp1||jmp2||(jpz1&&z)||(jpz2&&z)||(jpnz1&&(!z))||(jpnz2&&(!z))||lad1||lad2||lad4||sto1||sto2;
8. assign busmem=sto4;
9. assign r0load=add2||sub2||and2||or2||xor2||inc2||not2||clr1||shr2||lad5;
10. assign r1load=mvr1;
11. assign zload=add2||sub2||and2||or2||xor2||inc2||not2||shr2||clr1;
12. assign xload=add1||sub1||and1||or1||xor1||inc1||not1||shr1;
13. assign arload=fetch1||fetch3||lad3||sto3;
14. assign drload=fetch2||lad1||lad2||lad4||sto1||sto2||jmp1||jmp2||(jpz1&&z)||(jpz2&&z)||(jpnz1&&(!z))||(jpnz2&&(!z));
15. assign irload=fetch3;
16. assign trload=lad2||sto2||jmp2||(jpz2&&z)||(jpnz2&&(!z));
17. assign pcload=jmp3||(jpz3&&z)||(jpnz3&&(!z));
18. assign read=fetch2||lad1||lad2||lad4||sto1||sto2||jmp1||jmp2||(jpz1&&z)||(jpz2&&z)||(jpnz1&&(!z))||(jpnz2&&(!z));
19. assign write=sto4;
20. assign pcinc=fetch2||lad1||lad2||sto1||sto2||(jpz1&&(!z))||(jpz2&&(!z))||(jpnz1&&z)||(jpnz2&&z);
21. assign arinc=lad1||sto1||jmp1||(jpz1&&z)||(jpnz1&&(!z));

## 实验结果：

写出一段由自己所设计的指令系统构成的程序，并以注释方式注明每条指令的操作码助记符及执行后的结果

给出该程序运行的仿真结果（截图展示），并附以简单文字说明

|  |  |  |  |
| --- | --- | --- | --- |
| 地址 | 指令 | R0 | R1 |
| 0 | NOP | 00000000 | 00000000 |
| 1 | INC R0 | 00000001 | 00000000 |
| 2 | MVR R1<--R0 | 00000001 | 00000001 |
| 3 | ADD | 00000010 | 00000001 |
| 4 | XOR | 00000011 | 00000001 |
| 5 | NOT R0 | 11111100 | 00000001 |
| 6 | SUB | 11111011 | 00000001 |
| 7 | SHR R0 | 01111101 | 00000001 |
| 8 | NOT R0 | 10000010 | 00000001 |
| 9 | OR | 10000011 | 00000001 |
| 10 | AND | 00000001 | 00000001 |
| 11 | INC R0 | 00000010 | 00000001 |
| 12 | SUB | 00000001->00000000 | 00000001 |
| 13 | JPNZ | 00000001->00000000 | 00000001 |
| 14 | 12 |  | 00000001 |
| 15 |  | 00000001 |
| 16 | JPZ | 00000000 | 00000001 |
| 17 | 20 |  | 00000001 |
| 18 |  | 00000001 |
| 19 | INC R0 |  | 00000001 |
| 20 | NOT R0 | 11111111 | 00000001 |
| 21 | STO | 11111111 | 00000001 |
| 22 | A |  | 00000001 |
| 23 |  | 00000001 |
| 24 | CLR | 00000000 | 00000001 |
| 25 | JMP | 00000000 | 00000001 |
| 26 | 29 |  | 00000001 |
| 27 |  | 00000001 |
| 28 | INC R0 |  | 00000001 |
| 29 | LAD | 11111111 | 00000001 |
| 30 | A |  | 00000001 |
| 31 |  | 00000001 |

1. //以下为testbench中的测试代码
2. #50 D = 8'b00000000;//0,nop
3. #100 A1 = 0;
4. #300 A1 = 1;
5. #50 D = 8'b01100000;//1,inc
6. #100 A1 = 0;
7. #300 A1 = 1;
9. #50 D = 8'b10100000;//2,mvr r1<-r0
10. #100 A1 = 0;
11. #300 A1 = 1;
13. #50 D = 8'b00010000;//3,add
14. #100 A1 = 0;
15. #300 A1 = 1;
17. #50 D = 8'b01010000;//4,xor
18. #100 A1 = 0;
19. #300 A1 = 1;
21. #50 D = 8'b01110000;//5,not
22. #100 A1 = 0;
23. #300 A1 = 1;
25. #50 D = 8'b00100000;//6,sub
26. #100 A1 = 0;
27. #300 A1 = 1;
29. #50 D = 8'b10010000;//7,shr
30. #100 A1 = 0;
31. #300 A1 = 1;
33. #50 D = 8'b01110000;//8,not
34. #100 A1 = 0;
35. #300 A1 = 1;
37. #50 D = 8'b01000000;//9,or
38. #100 A1 = 0;
39. #300 A1 = 1;
41. #50 D = 8'b00110000;//10,and
42. #100 A1 = 0;
43. #300 A1 = 1;
44. #50 D = 8'b01100000;//11,inc
45. #100 A1 = 0;
46. #300 A1 = 1;
48. #50 D = 8'b00100000;//12,sub
49. #100 A1 = 0;
50. #300 A1 = 1;
52. #50 D = 8'b11010000;//13,jpnz
53. #100 A1 = 0;
54. #300 A1 = 1;
56. #50 D = 8'b00001100;//14,jpnz地址低八位
57. #100 A1 = 0;
58. #300 A1 = 1;
60. #50 D = 8'b00000000;//15,jpnz地址高八位
61. #100 A1 = 0;
62. #300 A1 = 1;
64. #50 D = 8'b11000000;//16,jpz
65. #100 A1 = 0;
66. #300 A1 = 1;
68. #50 D = 8'b00010100;//17,jpz地址低八位
69. #100 A1 = 0;
70. #300 A1 = 1;
72. #50 D = 8'b00000000;//18,jpz地址高八位
73. #100 A1 = 0;
74. #300 A1 = 1;
76. #50 D = 8'b01100000;//19,inc
77. #100 A1 = 0;
78. #300 A1 = 1;
80. #50 D = 8'b01110000;//20,not
81. #100 A1 = 0;
82. #300 A1 = 1;
84. #50 D = 8'b11110000;//21,sto
85. #100 A1 = 0;
86. #300 A1 = 1;
88. #50 D = 8'b00000000;//22,sto地址低八位
89. #100 A1 = 0;
90. #300 A1 = 1;
92. #50 D = 8'b00001111;//23,sto地址高八位
93. #100 A1 = 0;
94. #300 A1 = 1;
96. #50 D = 8'b10000000;//24,clr
97. #100 A1 = 0;
98. #300 A1 = 1;
100. #50 D = 8'b10110000;//25,jmp
101. #100 A1 = 0;
102. #300 A1 = 1;
104. #50 D = 8'b00011101;//26,jmp地址低八位
105. #100 A1 = 0;
106. #300 A1 = 1;
108. #50 D = 8'b00000000;//27,jmp地址高八位
109. #100 A1 = 0;
110. #300 A1 = 1;
112. #50 D = 8'b01100000;//28,inc
113. #100 A1 = 0;
114. #300 A1 = 1;
116. #50 D = 8'b11100000;//29,lad
117. #100 A1 = 0;
118. #300 A1 = 1;
120. #50 D = 8'b00000000;//30,lad地址低八位
121. #100 A1 = 0;
122. #300 A1 = 1;
124. #50 D = 8'b00001111;//31,lad地址高八位
125. #100 A1 = 0;
126. #300 A1 = 1;



## 

先执行空指令，再自加后吧R0的值赋值给R1，后依次进行加法，异或，非，减法，右移，非，或，与，自加，减法运算，下一步指令是JPNZ，此时z=！R0=0，返回到地址12所指的SUB指令继续做减法，R0减为0后，z=1，JPNZ不继续跳转，继续执行下一步操作JPZ，此时z=1，跳转到地址20所指的非操作，随后将运算完的R0的值存到地址A（0f00）中，对R0清零，无条件跳转到地址29所指的载入操作，从地址A（0f00）中取出数重新存入R0中。