

测试题 2

修改思路

先修改 `alu.v`，加入汉明距离的相关计算 `aluc` 控制信号。这里使用的方法是将 `add` 的控制信号限定为 `0001`，而把汉明距离计算指令的控制信号设定为 `1001`。如下所示。

```
1  ...
2  4'b0001: s = a & b;           //0001 AND
3      4'b1001:
4      s = (a[0] ^ b[0]) +
5          (a[1] ^ b[1]) +
6          (a[2] ^ b[2]) +
7          (a[3] ^ b[3]) +
8          (a[4] ^ b[4]) +
9          (a[5] ^ b[5]) +
10         (a[6] ^ b[6]) +
11         (a[7] ^ b[7]) +
12         (a[8] ^ b[8]) +
13         (a[9] ^ b[9]) +
14         (a[10] ^ b[10]) +
15         (a[11] ^ b[11]) +
16         (a[12] ^ b[12]) +
17         (a[13] ^ b[13]) +
18         (a[14] ^ b[14]) +
19         (a[15] ^ b[15]) +
20         (a[16] ^ b[16]) +
21         (a[17] ^ b[17]) +
22         (a[18] ^ b[18]) +
23         (a[19] ^ b[19]) +
24         (a[20] ^ b[20]) +
25         (a[21] ^ b[21]) +
26         (a[22] ^ b[22]) +
27         (a[23] ^ b[23]) +
28         (a[24] ^ b[24]) +
29         (a[25] ^ b[25]) +
30         (a[26] ^ b[26]) +
31         (a[27] ^ b[27]) +
32         (a[28] ^ b[28]) +
33         (a[29] ^ b[29]) +
34         (a[30] ^ b[30]) +
35         (a[31] ^ b[31]);       //1001 hamming distance
36  ...
```

然后修改 `sc_cu.v`，增加对新增指令的解码。如下所示。

```
1  ...
2  wire i_hamming = r_type & func[5] & func[4] & ~func[3] &
3      ~func[2] & ~func[1] & func[0];           // 110001, new!
4  ...
5
6  ...
```

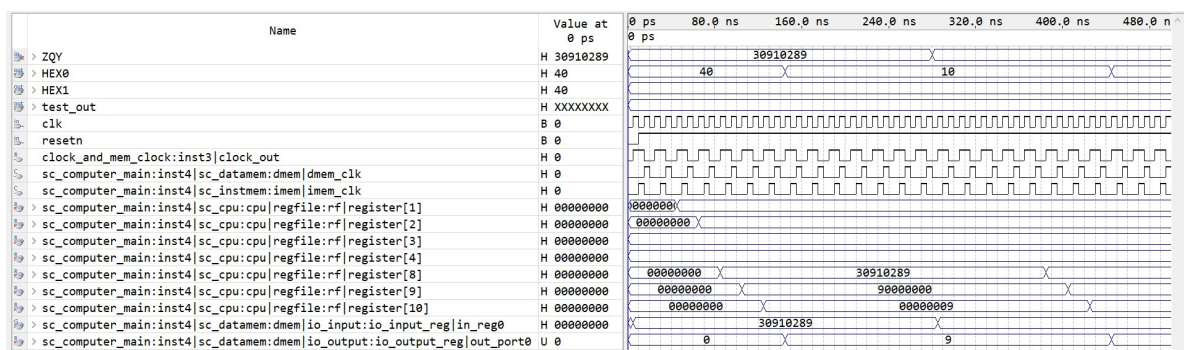
```

7  assign wreg = i_add | i_sub | i_and | i_or  | i_xor |
8              i_sll | i_srl | i_sra | i_addi | i_andi |
9              i_ori | i_xori | i_lw | i_lui  | i_jal | i_hamming; //
    new!
10
11  assign aluc[3] = i_sra | i_hamming; // new!
12      assign aluc[2] = i_sub | i_or | i_srl | i_sra | i_ori | i_lui;
13      assign aluc[1] = i_xor | i_sll | i_srl | i_sra | i_xori | i_lui;
14  assign aluc[0] = i_and | i_or | i_sll | i_srl | i_sra | i_andi | i_ori |
    i_hamming; // new!
15  ...

```

再编写编译指令，先以 add 为基准，然后手动修改 .mif 文件以正确解码。再参照测试题 1 修改波形即可。

波形仿真结果



其中 in_reg0 对应 90h 口，out_port0 对应 80h 口。

汉明距离计算

理论结果如下。

R8: 00110000100100010000001010001001

R9: 10010000000000000000000000000000

汉明距离：9。可以看出，与上面的实验结果是相符合的。

汇编代码

如下所示。

```

1  main: addi $1, $0, 144 # in_port (named zqy)
2        addi $2, $0, 128 # out_port
3  loop: lw  $8, 0($1)      # input in_port to $8
4        sll  $9, $8, 28    # left shift by 28 bits
5        add  $10, $8, $9   # find hamming distance
6        sw   $10, 0($2)    # output the result to out_port
7        j    loop

```

生成指令存储器 .mif 后手动修改它，得到：

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	20010090	20020080	8C280000	00084F00	01095031	AC4A0000	08000002	00000000

数据存储器 .mif 和测试题 1 相同，不改变。