

```

1  .text 0x0000
2  j      start                // 0
3  add $zero, $zero, $zero    // 4
4  add $zero, $zero, $zero    // 8
5  add $zero, $zero, $zero    // C
6  add $zero, $zero, $zero    // 10
7  add $zero, $zero, $zero    // 14
8  add $zero, $zero, $zero    // 18
9  add $zero, $zero, $zero    // 1C
10
11 start:
12 nor $at, $zero, $zero      // $1=FFFFFFFF
13 add $v1, $at, $at          // $3=FFFFFFFE
14 add $v1, $v1, $v1          // $3=FFFFFFFC
15 add $v1, $v1, $v1          // $3=FFFFFFF8
16 add $v1, $v1, $v1          // $3=FFFFFFF0
17
18 add $v1, $v1, $v1          // $3=FFFFFFE0
19 add $v1, $v1, $v1          // $3=FFFFFFC0
20 nor $s4, $v1, $zero        // $20=0000003F
21 add $v1, $v1, $v1          // $3=FFFFFF80
22 add $v1, $v1, $v1          // $3=FFFFFF00
23
24 add $v1, $v1, $v1          // $3=FFFFFFE0
25 add $v1, $v1, $v1          // $3=FFFFFFC0
26 add $v1, $v1, $v1          // $3=FFFFFF80
27 add $v1, $v1, $v1          // $3=FFFFFF00
28
29 add $v1, $v1, $v1          // $3=FFFFFE00
30 add $v1, $v1, $v1          // $3=FFFFFC00
31 add $v1, $v1, $v1          // $3=FFFF8000
32 add $v1, $v1, $v1          // $3=FFFF0000
33
34 add $v1, $v1, $v1          // $3=FFFE0000
35 add $v1, $v1, $v1          // $3=FFFC0000
36 add $v1, $v1, $v1          // $3=FFF80000
37 add $v1, $v1, $v1          // $3=FFF00000
38
39 add $v1, $v1, $v1          // $3=FFE00000
40 add $v1, $v1, $v1          // $3=FFC00000
41 add $v1, $v1, $v1          // $3=FF800000
42 add $v1, $v1, $v1          // $3=FF000000
43
44 add $v1, $v1, $v1          // $3=FE000000
45 add $v1, $v1, $v1          // $3=FC000000
46 add $a2, $v1, $v1          // $6=F8000000
47 add $v1, $a2, $a2          // $3=F0000000
48
49 add $a0, $v1, $v1          // $4=E0000000
50
51 add $t5, $a0, $a0          // $13=C0000000
52 add $t0, $t5, $t5          // $8=80000000
53
54 loop:
55 slt $v0, $zero, $at        // $2=00000001 针对ALU32位无符号数减
56 add $t6, $v0, $v0
57 add $t6, $t6, $t6          // $14=4
58 nor $t2, $zero, $zero      // $10=FFFFFFFF
59 add $t2, $t2, $t2          // $10=FFFFFFFE
60

```

```

61 loop1:
62     sw $a2, 0x4($v1) // 计数器端口: F0000004, 送计数常数$6=F8000000
63     lw $a1, 0x0($v1) // 读GPIO端口F0000000状态: {out0, out1, out2, D28-D20, LED7-L
64     add $a1, $a1, $a1 // 左移
65     add $a1, $a1, $a1 // 左移2位将SW与LED对齐, 同时D1D0置00, 选择计数器通道0
66     sw $a1, 0x0($v1) // $5输出到GPIO端口F0000000, 设置计数器通道counter_set=00端口、
67     add $t1, $t1, $v0 // $9=$9+1
68     sw $t1, 0x0($a0) // $9送$4=E0000000七段码端口
69     lw $t5, 0x0214($zero) // 取存储器20单元预存数据至$13, 程序计数延时常数
70
71 loop2:
72     lw $a1, 0x0($v1) // 读GPIO端口F0000000状态: {out0, out1, out2, D28-D20, LED7-L
73     add $a1, $a1, $a1
74     add $a1, $a1, $a1 // 左移2位将SW与LED对齐, 同时D1D0置00, 选择计数器通道0
75     sw $a1, 0x0($v1) // $5输出到GPIO端口F0000000, 计数器通道counter_set=00端口不变、
76
77     lw $a1, 0x0($v1) // 再读GPIO端口F0000000状态
78     and $t3, $a1, $t0 // 取最高位=out0, 屏蔽其余位送$11
79 // beq $t3, $t0, C_init // out0=0, Counter通道0溢出, 转计数器初始化, 修改7段码显示: C_
80     add $t5, $t5, $v0 // 程序计数延时
81     beq $t5, $zero, C_init // 程序计数$13=0, 转计数器初始化, 修改7段码显示: C_init
82
83 l_next: // 判断7段码显示模式: SW[4: 3]控制
84     lw $a1, 0x0($v1) // 再读GPIO端口F0000000开关SW状态
85     add $s2, $t6, $t6 // $14=4, $18=00000008
86     add $s6, $s2, $s2 // $22=00000010
87     add $s2, $s2, $s6 // $18=00000018(00011000)
88     and $t3, $a1, $s2 // 取SW[4: 3]
89     beq $t3, $zero, L20 // SW[4: 3]=00, 7段显示"点"循环移位: L20, SW0=0
90     beq $t3, $s2, L21 // SW[4: 3]=11, 显示七段图形, L21, SW0=0
91     add $s2, $t6, $t6 // $18=8
92     beq $t3, $s2, L22 // SW[4: 3]=01, 七段显示预置数字, L22, SW0=1
93     sw $t1, 0x0($a0) // SW[4: 3]=10, 显示$9, SW0=1
94     j loop2;
95
96 L20:
97     beq $t2, $at, L4 // $10=ffffffff, 转移L4
98     j L3
99
100 L4:
101     nor $t2, $zero, $zero // $10=ffffffff
102     add $t2, $t2, $t2 // $10=fffffffe
103
104 L3:
105     sw $t2, 0x0($a0) // SW[4: 3]=00, 7段显示点移位后显示
106     j loop2
107
108 L21:
109     lw $t1, 0x0260($s1) // SW[4: 3]=11, 从内存取预存七段图形
110     sw $t1, 0x0($a0) // SW[4: 3]=11, 显示七段图形
111     j loop2
112
113 L22:
114     lw $t1, 0x0220($s1) // SW[4: 3]=01, 从内存取预存数字
115     sw $t1, 0x0($a0) // SW[4: 3]=01, 七段显示预置数字
116     j loop2
117
118 C_init:
119     lw $t5, 0x0214($zero) // 取程序计数延时初始化常数
120     add $t2, $t2, $t2 // $10=fffffffc, 7段图形点左移

```

```

121 or $t2, $t2, $v0 // $10末位置1, 对应右上角不显示
122 add $s1, $s1, $t6 // $17=00000004, LED图形访存地址+4
123 and $s1, $s1, $s4 // $17=000000XX, 屏蔽地址高位, 只取6位
124 add $t1, $t1, $v0 // $9+1
125 beq $t1, $at, L6 // 若$9=ffffffff, 重置$9=5
126 j L7
127
128 L6:
129 add $t1, $zero, $t6 // $9=4
130 add $t1, $t1, $v0 // 重置$9=5
131
132 L7:
133 lw $a1, 0x0($v1) // 读GPIO端口F0000000状态
134 add $t3, $a1, $a1
135 add $t3, $t3, $t3 // 左移2位将SW与LED对齐, 同时D1D0置00, 选择计数器通道0
136 sw $t3, 0x0($v1) // $5输出到GPIO端口F0000000, 计数器通道counter_set=00端口不变、
137 sw $a2, 0x4($v1) // 计数器端口: F0000004, 送计数常数$6=F8000000
138 j l_next
139
140 .data 0x0200
141 .word 0xf0000000, 0x000002AB, 0x80000000, 0x0000003F, 0x00000001, 0xFFFF0000, 0x0000FFFF,
142 .word 0x00000000, 0x11111111, 0x22222222, 0x33333333, 0x44444444, 0x55555555, 0x66666666,
143 .word 0x88888888, 0x99999999, 0xaaaaaaaa, 0xbbbbbbbb, 0xcccccccc, 0xdddddddd, 0xeeeeeeee,
144 .word 0x557EF7E0, 0xD7BDFBD9, 0xD7DBFDB9, 0xDFCFFCFB, 0xDFCFBFFF, 0xF7F3DFFF, 0xFFFFDF3D,
145 .word 0xFFFFBCFB, 0xDFCFFCFB, 0xDFCFBFFF, 0xD7DB9FFF, 0xD7DBFDB9, 0xD7BDFBD9, 0xFFFF07E0,
146 .word 0x03bdf020, 0x03def820, 0x08002300, 0x00000000, 0x00000000, 0x00000000, 0x00000000,
147

```