嵌入式SOC设计

-SOC功能测试C语言描述

浙江大学 计算机学院 施青松

武汉

2014年 11月11日

SOC功能测试程序

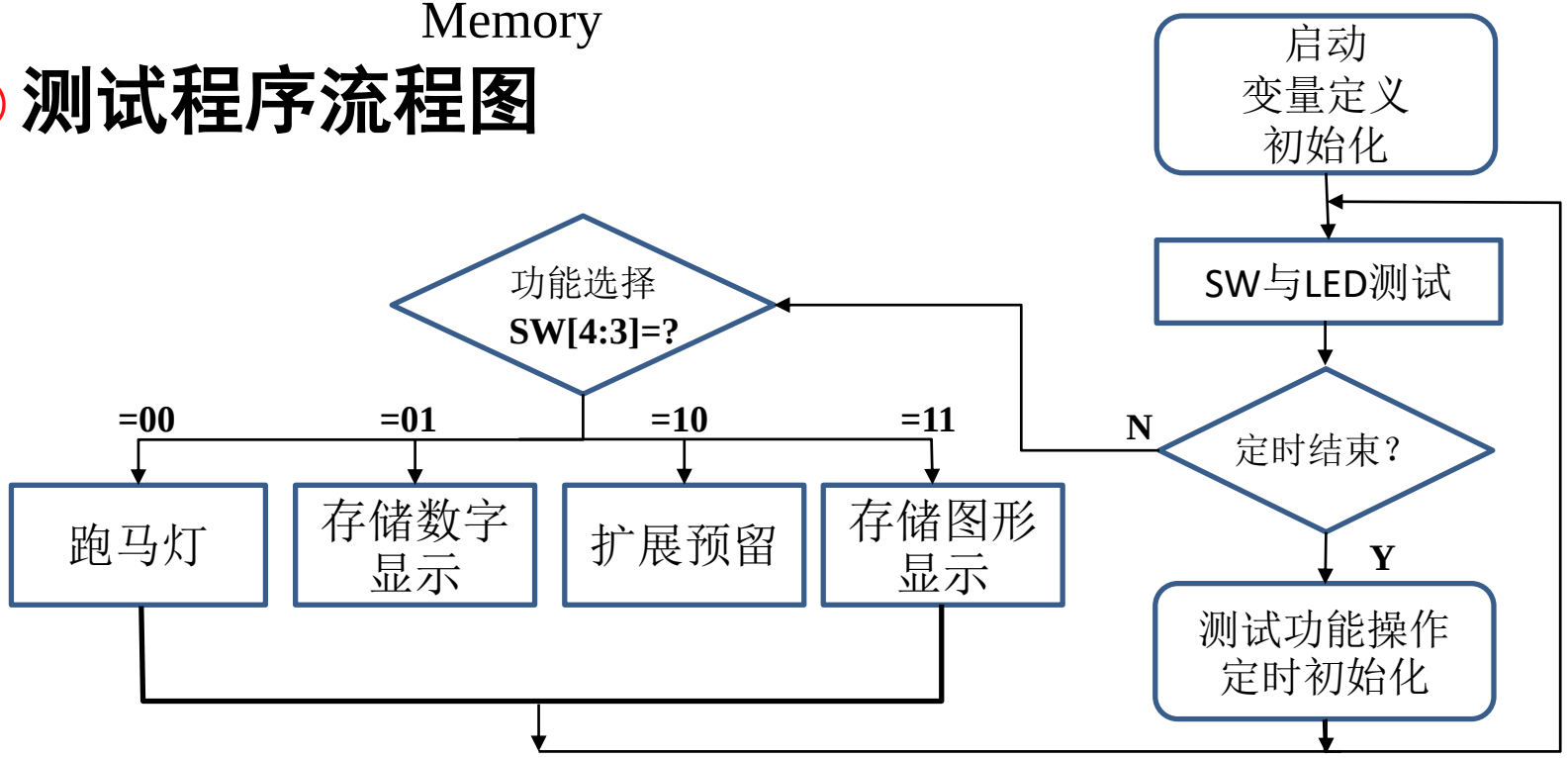
- **测试**
  - 设计测试
    - 功能测试、性能测试
    - 指令测试、通路测试、部件测试、IO测试等
    - 抽样测试、针对性测试
  - 产品测试
    - 完备性测试等
- **功能测试**
  - 选择主要功能设计测试程序
    - 实现验证功能粗调用
  - 选择针对性功能设计测试程序
    - 实现验证细节调试用

SOC功能粗调测试框图

◎ 功能抽样

- 基本指令：lw、sw、add、sub、and、or、slt、beq、j
- 基本部件：CPU、MIO_BUS、Display、GPIO-LED、GPIO-SW Memory

◎ 测试程序流程图



SOC功能测试代码：C语言

```
        long int *GPIO_Port, *Display_Port, *Counter_Port;
        long int A[16] =……, B[16] =……;
main(){     long int GPIO, Display, Counter, Ctimer, Memory, Count_Soft;
            long int i=0, N_4, N_left_1, Temp, Dot;
            goto Start;

 Start:     long int *GPIO_Port    =  0xF000000;
            long int *Display_Port =  0xE000000;
            Counter_Port = GPIO_Port + 4;
            Counter = 0x0000003F;
            Ctimer     = 0xF8000000;          //计数常数
            N_left_1 = 0x80000000;
            N_4        = 0x00000004;          //字内字节数，字地址+1=字节+4
            Dot        = 0xFFFFFFFE;          //最低位为0的数，在七段码的右上角显示一个点

 loop:      Counter_Port = Ctimer;            //送硬件计数时间常数
            GPIO           = *GPIO_Port;      //读GPIO状态：SW状态
            GPIO           = GPIO <<2;        //SW和LED对齐
            GPIO_Port    = GPIO;              //SW状态送LED显示
            Display_Port = Display + 1;       //显示值输出七段显示器
 ……
```

启动、变量定义与初始化

功能判及测试功能代码

```
        Count_Soft    = A[5];                    //取存储器常量，用于软件计数初值
loop1:  GPIO          = *GPIO_Port;              //读GPIO状态：SW状态
        GPIO          = GPIO <<2;                //SW和LED对齐
        *GPIO_Port    = GPIO;                    //SW状态送LED显示
        GPIO          = *GPIO_Port;              //再读GPIO状态：状态判断用
        Temp          = GPIO & N_left_1; //与80000000相与,取最高位=out0,屏蔽其余位
        //if (Temp ==N_left_1)  goto C_init; //硬件计数溢出out0=1,计数器通道0溢出,转C_init
        Count_Soft    = Count_Soft + 1;          //程序计数延时(加1)
        if (Count_Soft==0) goto C_init;          //若程序计数溢出转C_init

Next:   GPIO          = *GPIO_Port;    //延时未到继GPIO SW状态：判断显示内容SW[4:3]=?
        Temp          = GPIO  & 0x00000018;
        if(Temp ==0x00000000) goto L20;          //SW[4:3]=00,7段显示“点”左移跑马灯
        if(Temp ==0x00000018) goto L21;          // SW[4:3]=11，七段显示B[i]预存七段图形
        if(Temp ==0x00000008) goto L22;          // SW[4:3]=01，七段显示A[i]预置16进制数
        *Display_Port = Display ;                // SW[4:3]=10，输出显示值+1， SW0=1
        goto loop1;
L20:    if(Dot ==0xFFFFFFFF) Dot =  0xFFFFFFFE; // 跑马灯全灭置Dot= 0xFFFFFFFE
        *Display_Port = Dot ;                    // SW[4:3]=00，显示跑马灯
        goto loop1;
……
```

SW状态循环显示

三种定时选择

四种显示

测试功能与定时初始化代码：四种显示

```
L21:   Memory = B[i];
       *Display_Port = Memory;        //SW[4:3]=11，显示预存七段图形
       goto loop1;

L22:   Memory = A[i+8];               //字地址是8，字节地址是0x20
       *Display_Port = Memory;        //SW[4:3]=01，显示预存16进制数， SW0=1
       goto loop1;
-------------------------------------------------------------------
C_init:                               //延时结束，修改显示值和定时/延时初始化
       Count_Soft    = A[5]; //取存储器常量，用于软件计数初值
       Dot = Dot  <<1;
       Dot = Dot +1;
       i     =( i +1)&0x0000000f;    // 数组下标+1，内存地址+4
       Display = Display + 1;         //当前显示值 +1
       if(Display ==0xFFFFFFFF) Display = 5; //显示全F,则置5
       GPIO          = GPIO_Port;    //读GPIO状态：SW状态
       GPIO           = GPIO <<2;    //SW和LED对齐
       *GPIO_Port    = GPIO;                    //SW状态送LED显示
       *Counter_Port = Ctimer;       //送硬件计数时间常数
       goto Next;
}
```

数组A、B初始化值

A[ ] =

{0xf0000000, 0x000002AB, 0x80000000, 0x0000003F, 0x00000001,
0xFFF70000, 0x0000FFFF, 0x80000000, 0x00000000, 0x11111111,
0x22222222, 0x33333333, 0x44444444, 0x55555555, 0x66666666,
0x77777777, 0x88888888, 0x99999999, 0xAAAAAAAA, 0xBBBBBBBB,
0xCCCCCCCC, 0xDDDDDDDD, 0xEEEEEEEE, 0xFFFFFFFF, 0x00000000};

B[ ] =

{0x557EF7E0, 0xD7BDFBD9, 0xD7DBFDB9, 0xDFCFFCFB, 0xDFCFBFFF,
0xF7F3DFFF, 0xFFFFDF3D, 0xFFFF9DB9, 0xFFFFBCFB, 0xDFCFFCFB,
0xDFCFBFFF, 0xD7DB9FFF, 0xD7DBFDB9, 0xD7BDFBD9, 0xFFFF07E0,
0x007E0FFF, 0x03bdf020, 0x03def820, 0x08002300, 0x00000000};

浙江大学 ZHEJIANG UNIVERSITY

嵌入式SOC设计

-SOC功能测试MIPS程序

浙江大学 计算机学院 施青松

武汉

2014年 11月11日

SOC功能测试MIPS代码 启动与常数子代码

```
#baseAddr 0000
        j  start;                          //0
        add $zero, $zero, $zero;   //4
        add $zero, $zero, $zero;   //8
        add $zero, $zero, $zero;   //C
        add $zero, $zero, $zero;   //10
        add $zero, $zero, $zero;   //14
        add $zero, $zero, $zero;   //18
        add $zero, $zero, $zero;   //1C
start:
        nor $at, $zero, $zero; //r1=FFFFFFFF
        add $v1, $at, $at;  //r3=FFFFFFFE
        add $v1, $v1, $v1;  //r3=FFFFFFFC
        add $v1, $v1, $v1;  //r3=FFFFFFF8
        add $v1, $v1, $v1;  //r3=FFFFFFF0
        add $v1, $v1, $v1;  //r3=FFFFFFE0
        add $v1, $v1, $v1;  //r3=FFFFFFC0,
        nor $s4, $v1, $zero; //r20=0000003F
        add $v1, $v1, $v1;  //r3=FFFFFF80
        add $v1, $v1, $v1;  //r3=FFFFFF00
        add $v1, $v1, $v1;  //r3=FFFFFE00
       add $v1, $v1, $v1;  //r3=FFFFFC00
        add $v1, $v1, $v1;  //r3=FFFFF800
        add $v1, $v1, $v1;  //r3=FFFFF000
        add $v1, $v1, $v1;  //r3=FFFFE000
        add $v1, $v1, $v1;  //r3=FFFFC000
        add $v1, $v1, $v1;  //r3=FFFF8000
        add $v1, $v1, $v1;  //r3=FFFF0000
        add $v1, $v1, $v1;  //r3=FFFE0000
        add $v1, $v1, $v1;  //r3=FFFC0000
        add $v1, $v1, $v1;  //r3=FFF80000
        add $v1, $v1, $v1;  //r3=FFF00000
        add $v1, $v1, $v1;  //r3=FFE00000
        add $v1, $v1, $v1;  //r3=FFC00000
        add $v1, $v1, $v1;  //r3=FF800000
        add $v1, $v1, $v1;  //r3=FF000000
        add $v1, $v1, $v1;  //r3=FE000000
        add $v1, $v1, $v1;  //r3=FC000000,左移26位
        add $a2, $v1, $v1;  //$a2:r6=F8000000计数常数
        add $v1, $a2, $a2;  //r3=F0000000，GPIO地址
        add $a0, $v1, $v1;  //r4=E0000000,七段码显示地址
        add $t5, $a0, $a0;  //$t5:r13=C0000000
        add $t0, $t5, $t5;  //r8=80000000,仅最高位为1
```

初次运行设置代码

```
loop:
    slt $v0,$at,$zero;          //因0>FFFFFFFFF(符号数),故$v0:r2=00000001，则生成常数1
    add $t6, $v0, $v0;
    add $t6, $t6, $t6;          //$t6:r14=4,为获得常数，字地址+1=字节+4
    nor $t2, $zero, $zero;      //$t2:r10=FFFFFFFF, 即~0
    add $t2, $t2, $t2;          //$t2:r10=FFFFFFFE，最低位为0的数，在七段码的右上角显示一个点

   sw  $a2, 4($v1); //计数器端口:F0000004，送计数
    lw  $a1, 0($v1); //读GPIO端口F0000000状态://{out0，out1，out2，D28-D20,LED7-LED7，BTN3-BTN0,SW7-SW0}
   //out0=1,表示计数器0溢出。
    add $a1, $a1, $a1;  //左移
    add $a1, $a1, $a1;  //$a1左移2位,将SW与LED输出对齐
    sw  $a1, 0($v1);     //将新$a1:r5写到GPIO端口F0000000,写到LED
    add $t1, $t1, $v0;  //$t1：r9=r9+1
    sw  $t1, 0($a0);    //$t1：r9送r4=E0000000七段码端口
    lw  $t5, 14($zero);  //取存储器第20单元预存数据至$t5:r13,程序计数延时常数
```

功能判断子代码

```
loop1:
      lw  $a1, 0($v1);        //读GPIO端口F0000000状态，同前。
      add $a1, $a1, $a1;
      add $a1, $a1, $a1;      //左移2位将SW与LED对齐
      sw  $a1, 0($v1);        //再将新$a1:r5写到GPIO端口F0000000,写到LED
      lw  $a1, 0($v1);        //再读GPIO端口F0000000状态
      and $t3,$a1,$t0;        //与80000000相与，即：取最高位=out0,屏蔽其余位
      // beq  $t3,$t0,C_init;  //硬件计数。out0=1,Counter通道0溢出,转C_init
      add $t5, $t5, $v0;    //程序计数延时(加1)
      beq $t5, $zero,C_init;  //若程序计数$t5:r13=0,转C_init
l_next:                       // 延时未到，继续：判断7段码显示模式：SW[4:3]
      lw  $a1, 0($v1);        //再读GPIO端口F0000000开关SW
      add $s2, $t6, $t6;      //因$t6:r14=4, 故$s2:r18=00000008
      add $s6, $s2, $s2;      //$s6:r22=00000010
      add $s2, $s2, $s6;      //$s2:r18=00000018(00011000)
      and $t3, $a1, $s2;      //取SW[4:3]到$t3
      beq $t3, $zero, L20;    //SW[4:3]=00,7段显示"点"左移反复
      beq $t3, $s2, L21;      //SW[4:3]=11，显示七段图形
      add $s2, $t6, $t6;      //$s2:r18=8
      beq $t3, $s2, L22;      //SW[4:3]=01,七段显示预置数字
      ……
```

SW状态循环显示

三种定时选择

测试功能子代码：四种显示

```
        sw  $t1, 0($a0);            //SW[4:3]=10，显示输出“当前显示数+1”，SW0=1
        j loop1;

L20:
        beq $t2, $at, L4;           //r10=0xFFFFFFFF,当前显示全黑转移L4
        j L3;
L4:
        nor $t2, $zero, $zero;      //r10=0xFFFFFFFF
        add $t2, $t2, $t2;          //r10=0xFFFFFFFE
L3:
        sw  $t2, 0($a0);            //SW[4:3]=00,7段显示点移位后显示
        j loop1;
L21:
        lw  $t1, 60($s1);           //SW[4:3]=11，从内存取预存七段图形
        sw  $t1, 0($a0);            //SW[4:3]=11，显示七段码图形
        j loop1;
L22:
        lw  $t1, 20($s1);           //SW[4:3]=01，从内存取预存数字
        sw  $t1, 0($a0);            //SW[4:3]=01,七段显示预置数字
        j loop1;
```

定时子代码

```
C_init:                          //延时结束，修改显示值和定时/延时初始化
    lw  $t5, 14($zero);          //取程序计数延时初始化常数
     add $t2, $t2, $t2;          //$t2:r10=fffffffc，7段图形点左移
     or  $t2, $t2, $v0;          //$t2:r10末位置1，对应右上角不显示
     add $s1, $s1, $t6;          //$t6:r14=00000004，LED图形访存字地址加1
     and $s1, $s1, $s4;          //与3F相与，留下后6位。$s1:r17=000000XX，//屏蔽地址高位
     add $t1, $t1, $v0;          //r9+1
     beq $t1, $at, L6;           //若r9=ffffffff,重置r9=5
     j L7;
L6:
     add $t1, $zero, $t6;        //r9=4
     add $t1, $t1, $v0;          //重置r9=5
L7:
     lw  $a1, 0($v1);            //读GPIO端口F0000000状态
     add $t3, $a1, $a1;
     add $t3, $t3, $t3;          //左移2位将SW与LED对齐，同时
     sw  $t3, 0($v1);            //r5输出到GPIO端口F0000000
     sw $a2,4($v1);              // 计数器端口:F0000004，送计数常数
     j  l_next;
```

ROM初始数据-.coe

I9_mem.coe初始数据

```
memory_initialization_radix=16;
memory_initialization_vector=
08000008, 00000020, 00000020, 00000020, 00000020, 00000020, 00000020, 00000020,
00000827, 00211820, 00631820, 00631820, 00631820, 00631820, 00631820, 0060a027,
00631820, 00631820, 00631820, 00631820, 00631820, 00631820, 00631820, 00631820,
00631820, 00631820, 00631820, 00631820, 00631820, 00631820, 00631820, 00631820,
00631820, 00631820, 00631820, 00631820, 00633020, 00c61820, 00632020, 00846820,
01ad4020, 0001102a, 00427020, 01ce7020, 00005027, 014a5020, ac660004, 8c650000,
00a52820, 00a52820, ac650000, 01224820, ac890000, 8c0d0014, 8c650000, 00a52820,
00a52820, ac650000, 8c650000, 00a85824, 01a26820, 11a00017, 8c650000, 01ce9020,
0252b020, 02569020, 00b25824, 11600005, 1172000a, 01ce9020, 1172000b, ac890000,
08000036, 11410001, 0800004d, 00005027, 014a5020, ac8a0000, 08000036, 8e290060,
ac890000, 08000036, 8e290020, ac890000, 08000036, 8c0d0014, 014a5020, 01425025,
022e8820, 02348824, 01224820, 11210001, 0800005f, 000e4820, 01224820, 8c650000,
00a55820, 016b5820, ac6b0000, ac660004, 8c650000, 00a85824, 0800003e;
```

RAM初始数据-.coe

- D_mem.coe初始数据

```
memory_initialization_radix=16;
memory_initialization_vector=
F0000000, 000002AB, 80000000, 0000003F, 00000001, FFF70000,
0000FFFF, 80000000, 00000000, 11111111, 22222222, 33333333,
44444444, 55555555, 66666666, 77777777, 88888888, 99999999,
aaaaaaaa, bbbbbbbb, cccccccc, dddddddd, eeeeeeee, FFFFFFFF,
557EF7E0, D7BDFBD9, D7DBFDB9, DFCFFCFB, DFCFBFFF, F7F3DFFF,
FFFFDF3D, FFFF9DB9, FFFFBCFB, DFCFFCFB, DFCFBFFF, D7DB9FFF,
D7DBFDB9, D7BDFBD9, FFFF07E0, 007E0FFF, 03bdf020, 03def820,
08002300;
```

- 下载和仿真均可用

红色数据是LED图形


Thank you!