

## 关于测试单周期 CPU 的简单方法

（特别说明：本表每个同学都必须建立，根据需要添加代码，检查实验时，必须提供！）

### 1、测试程序段

地址	汇编程序	指令代码					16 进制数代码
		op(6)	rs(5)	rt(5)	rd(5)/immediate (16)		
0x00000000	addiu \$1,\$0,8	011111	00000	00001	0000 0000 0000 1000	=	7c010008
0x00000004	ori \$2,\$0,2	010000	00000	00010	0000 0000 0000 0010	=	40020002
0x00000008	add \$3,\$2,\$1	000000	00010	00001	0001 1000 0000 0000	=	00411800
0x0000000C	sub \$5,\$3,\$2	000010	00011	00010	0010 1000 0000 0000	=	08622800
0x00000010	and \$4,\$5,\$2	010001	00101	00010	0010 0000 0000 0000	=	44a22000
0x00000014	or \$8,\$4,\$2	010010	00100	00010	0100 0000 0000 0000	=	48824000
0x00000018	sll \$8,\$8,1	011000	00000	01000	0100 0000 0100 0000	=	60084040
0x0000001C	bne \$8,\$1,-2 (≠,转 18)	110001	01000	00001	1111 1111 1111 1110	=	c501fffe
0x00000020	slti \$6,\$2,4	011011	00010	00110	0000 0000 0000 1000	=	6c460008
0x00000024	slti \$7,\$6,0	011011	00110	00111	0000 0000 0000 0000	=	6cc70000
0x00000028	addi \$7,\$7,8	000001	00111	00111	0000 0000 0000 1000	=	04e70008
0x0000002C	beq \$7,\$1,-2 (=,转 28)	110000	00111	00001	1111 1111 1111 1110	=	c0e1fffe
0x00000030	sw \$2,4(\$1)	100110	00001	00010	0000 0000 0000 0100	=	98220004
0x00000034	lw \$9,4(\$1)	100111	00001	01001	0000 0000 0000 0100	=	9c290004
0x00000038	xor \$3,\$1,\$2	010011	00001	00010	0001 1000 0000 0000	=	4c221800
0x0000003C	sllv \$2,\$1,\$6	011001	00110	00001	0001 0000 0000 0000	=	64c11000
0x00000040	srlv \$3,\$1,\$6	000011	00110	00001	0001 1000 0000 0000	=	0cc11800
0x00000044	j 0x0000004C	111000	00000	00000	0000 0000 0001 0011	=	e0000013
0x00000048	addi \$10,\$0,10	000001	00000	01010	0000 0000 0000 1010	=	040a000a
0x0000004C	subu \$3,\$0,\$2	000111	00000	00010	0001 1000 0000 0000	=	1c021800
0x00000050	srl \$8,\$8,1	011110	00000	01000	0100 0000 0100 0000	=	78084040
0x00000054	addu \$3,\$0,\$2	001001	00000	00010	0001 1000 0000 0000	=	24021800
0x00000058	slt \$1,\$3,\$8	010101	00011	01000	0000 1000 0000 0000	=	54680800
0x0000005C	sltiu \$1,\$8,12	011101	01000	00001	0000 0000 0000 1100	=	7501000c
0x00000060	xori \$1,\$2,12	011010	00010	00001	0000 0000 0000 1100	=	6841000c
0x00000064	andi \$1,\$2,3	010110	00010	00001	0000 0000 0000 0011	=	58410003
0x00000068	halt	111111	00000	00000	0000 0000 0000 0000	=	FC000000

- 1、将指令代码初始化到指令存储器中，直接写入。
- 2、初始化 PC 的值，也就是以上程序段首地址 PC=0x00000000，以上程序段从 0x00000000 地址开始存放。
- 3、运行 Xilinx Vivado 进行仿真，看波形。
- 4、制作coe文件

指令存储器设计成8位，也可参照如下设计。

ROM：即指令存储器， CPU的指令文件即由ROM读取，只读。

input.txt程序中input.txt为指令代码文件，存成.coe或.txt文件。

```

`timescale 1ns / 1ps
module InstructionMemory(
    input[31:0] IAddr,
    input RW,
    output reg[31:0] IDataOut
);

    reg[7:0] InstMemory[255:0];

    initial begin        //此处为绝对地址，注意斜杠方向

        $readmemb("C:/Users/wukan/Documents/VIVADO/SingleCPU/input.txt",InstMemory);
//文件路径
    end
    always@(IAddr or RW)begin
        if(RW==0)begin
            IDataOut=
                {InstMemory[IAddr],InstMemory[IAddr+1],InstMemory[IAddr+2],InstMemory[IAddr+3]}
        };
    end
end
endmodule

```

制作 coe 文件，对 rom 进行初始化。

coe 文件的格式一般如下：数据之间以逗号隔开，最后一个数据用分号；

第一行、第二行都是标准格式，就第一行等号后面的数字需要改变，这个数字代表该.coe 文件里数据采用什么进制，有 2 进制、10 进制和 16 进制 3 种。

如果采用系统提供的 IP 核方式生成 Insrom，采用如下方法对 rom 进行初始化。

生成和导入 Xilinx ROM/RAM 的初始化文件.COE

这里比较关键，要导入 ROM 的值。

在 LOAD INIT FILE 项上打勾，点击 LOAD FILE.....出现下图，要你选择一个文件。

这个文件就是你要放在 ROM 中的数据，文件的后缀名是.COE。你可以自己建一个空的文本文件，然后把后缀名该为.COE 即可。

## Rom汇编程序设计、代码

### 1. 汇编代码main:

```

addi $s1,$zero,85 // U
sw $s1,0($s0)
addi $s1,$zero,50 // 2
sw $s1,4($s0)
addi $s1,$zero,48 // 0
sw $s1,8($s0)
addi $s1,$zero,49 // 1
sw $s1,12($s0)
addi $s1,$zero,51 // 3
sw $s1,16($s0)
addi $s1,$zero,49 // 1
sw $s1,20($s0)
addi $s1,$zero,51 // 3
sw $s1,24($s0)
addi $s1,$zero,55 // 7
sw $s1,28($s0)
addi $s1,$zero,54 // 6
sw $s1,32($s0)
addi $s1,$zero,56 // 8
sw $s1,36($s0)
add $4,$2,$3 // 将$4 = $2 + $3
lw $4,4($2) //读出$2偏移16个字节的内容的到$4
sw $2,8($2) //将$2写入$2偏移32字节的RAM
sub $2,$4,$3 // $2=$4-$3
or $2,$4,$3 // $2=$4 | $3
and $2,$4,$3 // $2=$4 & $3
slt $2,$4,$3 // $4<$3, $2 = 1
beq $4,$3,exit //$4 == $3, 退出
j main
exit:lw $2,0($3)
j main

```

## 2. 机器码

[00400024]	20110055	addi \$17, \$0, 85	; 2: addi \$s1,\$zero,85
[00400028]	ae110000	sw \$17, 0(\$16)	; 3: sw \$s1,0(\$s0)
[0040002c]	20110032	addi \$17, \$0, 50	; 4: addi \$s1,\$zero,50
[00400030]	ae110004	sw \$17, 4(\$16)	; 5: sw \$s1,4(\$s0)
[00400034]	20110030	addi \$17, \$0, 48	; 6: addi \$s1,\$zero,48
[00400038]	ae110008	sw \$17, 8(\$16)	; 7: sw \$s1,8(\$s0)
[0040003c]	20110031	addi \$17, \$0, 49	; 8: addi \$s1,\$zero,49
[00400040]	ae11000c	sw \$17, 12(\$16)	; 9: sw \$s1,12(\$s0)
[00400044]	20110033	addi \$17, \$0, 51	; 10: addi \$s1,\$zero,51
[00400048]	ae110010	sw \$17, 16(\$16)	; 11: sw \$s1,16(\$s0)
[0040004c]	20110031	addi \$17, \$0, 49	; 12: addi \$s1,\$zero,49
[00400050]	ae110014	sw \$17, 20(\$16)	; 13: sw \$s1,20(\$s0)
[00400054]	20110033	addi \$17, \$0, 51	; 14: addi \$s1,\$zero,51
[00400058]	ae110018	sw \$17, 24(\$16)	; 15: sw \$s1,24(\$s0)
[0040005c]	20110037	addi \$17, \$0, 55	; 16: addi \$s1,\$zero,55
[00400060]	ae11001c	sw \$17, 28(\$16)	; 17: sw \$s1,28(\$s0)
[00400064]	20110036	addi \$17, \$0, 54	; 18: addi \$s1,\$zero,54
[00400068]	ae110020	sw \$17, 32(\$16)	; 19: sw \$s1,32(\$s0)
[0040006c]	20110038	addi \$17, \$0, 56	; 20: addi \$s1,\$zero,56
[00400070]	ae110024	sw \$17, 36(\$16)	; 21: sw \$s1,36(\$s0)
[00400074]	00432020	add \$4, \$2, \$3	; 22: add \$4,\$2,\$3
[00400078]	8c440004	lw \$4, 4(\$2)	; 23: lw \$4,4(\$2)
[0040007c]	ac420008	sw \$2, 8(\$2)	; 24: sw \$2,8(\$2)
[00400080]	00831022	sub \$2, \$4, \$3	; 25: sub \$2,\$4,\$3
[00400084]	00831025	or \$2, \$4, \$3	; 26: or \$2,\$4,\$3
[00400088]	00831024	and \$2, \$4, \$3	; 27: and \$2,\$4,\$3
[0040008c]	0083102a	slt \$2, \$4, \$3	; 28: slt \$2,\$4,\$3
[00400090]	10830002	beq \$4, \$3, 8 [exit-0x00400090];	29: beq \$4,\$3,exit
[00400094]	08100009	j 0x00400024 [main]	; 30: j main
[00400098]	8c620000	lw \$2, 0(\$3)	; 31: lw \$2,0(\$3)
[0040009c]	08100009	j 0x00400024 [main]	; 32: j main

### 3. 制作coe文件

MEMORY\_INITIALIZATION\_RADIX=16;

MEMORY\_INITIALIZATION\_VECTOR=

20110055

ae110000

20110032

ae110004

20110030

ae110008  
20110031  
ae11000c  
20110033  
ae110010  
20110031  
ae110014  
20110033  
ae110018  
20110037  
ae11001c  
20110036  
ae110020  
20110038  
ae110024  
00432020  
8c440004  
ac420008  
00831022  
00831025  
00831024  
0083102a  
10830002  
08000000  
8c620000  
08000000

Sim.v

仿真模块。

```
`timescale 1ns / 1ps  
module Sim;  
    reg CLK; //时钟信号  
    reg Reset; //置零信号
```

```

SingleCPU scpu(CLK,Reset);
initial begin
    CLK=    0;
    Reset=  0;    //刚开始设置 pc 为 0

    #50;    //等待 Reset 完成
    CLK=    !CLK;    //下降沿，使 PC 先清零
    #50;
    Reset=  1;    //清除保持信号
    forever #50 begin    //产生时钟信号，周期为 50s
        CLK=    !CLK;
    end
end
endmodule

```