计算机系统设计

课程实验报告

实验名称　相对简单 CPU 的设计

学生姓名　樊龙

学生学号　201608010325

专业班级　智能 1601

实验名称：相对简单 CPU 电路设计

实验目标：利用 VHDL 设计相对简单 CPU 的电路并验证。

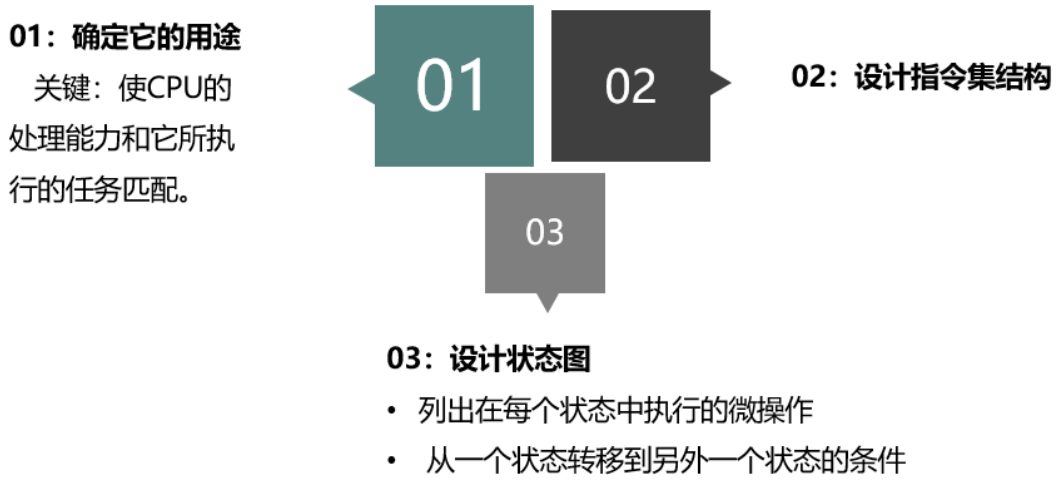
实验要求

- 采用 VHDL 描述电路及其测试平台
- 采用时序逻辑设计电路
- 采用从 1 累加到 n 的程序进行测试

实验内容

相对简单 CPU 的设计需求

CPU 设计步骤：



1.相对简单 CPU 的设计规格说明：

- 地址总线 16 位，数据总线 8 位
- 有一个 8 位累加寄存器 AC，一个 8 位通用寄存器 R，一个 1 位的零标志
- 有一个 16 位 AR 寄存器，一个 16 位程序计数器 PC，一个 8 位数据寄存器 DR，一个 8 位指令寄存器 IR，一个 8 位临时寄存器 TR

- 有 16 条指令，每条指令 1 个或 3 个字节，其中操作码 8 位。3 字节的指令有 16 位的地址

2. 指令集结构：

指令	指令码	操作
NOP	0000 0000	无
LDAC	0000 0001 Γ	AC←M[Γ]
STAC	0000 0010 Γ	M[Γ]←AC
MVAC	0000 0011	R←AC
MOV R	0000 0100	AC←R
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF (Z = 1) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF (Z = 0) THEN GOTO Γ
ADD	0000 1000	AC←AC + R，IF (AC + R = 0) THEN Z←1 ELSE Z←0
SUB	0000 1001	AC←AC - R，IF (AC - R = 0) THEN Z←1 ELSE Z←0
INAC	0000 1010	AC←AC + 1，IF (AC + 1 = 0) THEN Z←1 ELSE Z←0
CLAC	0000 1011	AC←0，Z←1
AND	0000 1100	AC←AC∧R，IF (AC∧R = 0) THEN Z←1 ELSE Z←0
OR	0000 1101	AC←AC∨R，IF (AC∨R = 0) THEN Z←1 ELSE Z←0
XOR	0000 1110	AC←AC⊕R，IF (AC⊕R = 0) THEN Z←1 ELSE Z←0
NOT	0000 1111	AC←AC'，IF (AC' = 0) THEN Z←1 ELSE Z←0

3．一些寄存器

◆ 16 位的地址寄存器 AR：通过引脚 A[15..0]向存储器提供地址。

◆ 16 位的程序计数器 PC：存放的是将要执行的下一条指令的地址，或者指令需要的下一个操作数的地址。

◆ 8 位的数据寄存器 DR：通过 D[7..0]从存储器中接收指令和数据并且向存储器传送数据。

◆ 8 位的指令寄存器 IR：存放的是从存储器中取出来的操作码。

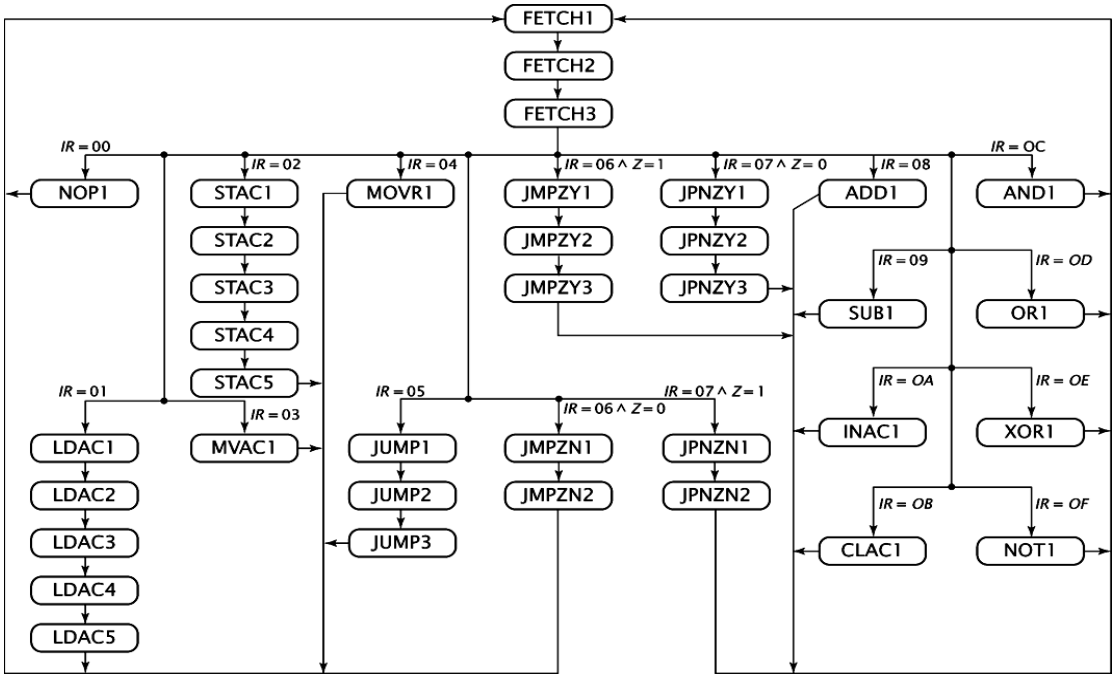
◆ 8 位的临时寄存器 TR：在指令执行过程中，临时存储数据。（程序员不能访问）

相对简单 CPU 设计方案

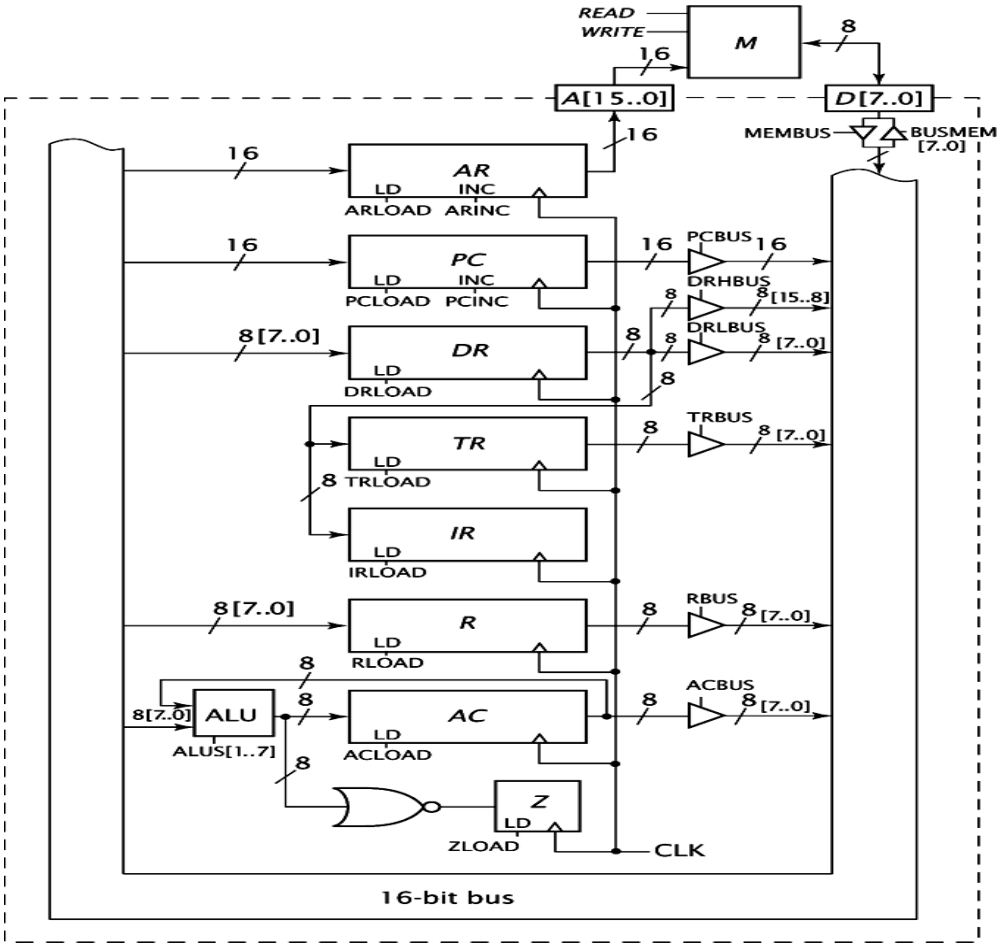
相对简单 CPU 的设计方案请详见课件，主要思路如下：

1. 指令执行过程分为取指、译码、执行三个阶段
2. 取指包括三个状态，FETCH1，FETCH2,FETCH3
3. 译码体现为从 FETCH3 状态到各指令执行状态序列的第一个状态
4. 执行根据指令的具体操作分为若干状态
5. 执行的最后一个状态转移到 FETCH1 状态
6. 控制器根据每个状态需要完成的操作产生相应的控制信号

根据指令执行过程得到状态之间的转换如下：



根据每个指令的状态转换，数据之间的传输关系构建数据通路如下：



内存设计：

例如计算 1+2+3+4+5 的值，指令执行过程如下：

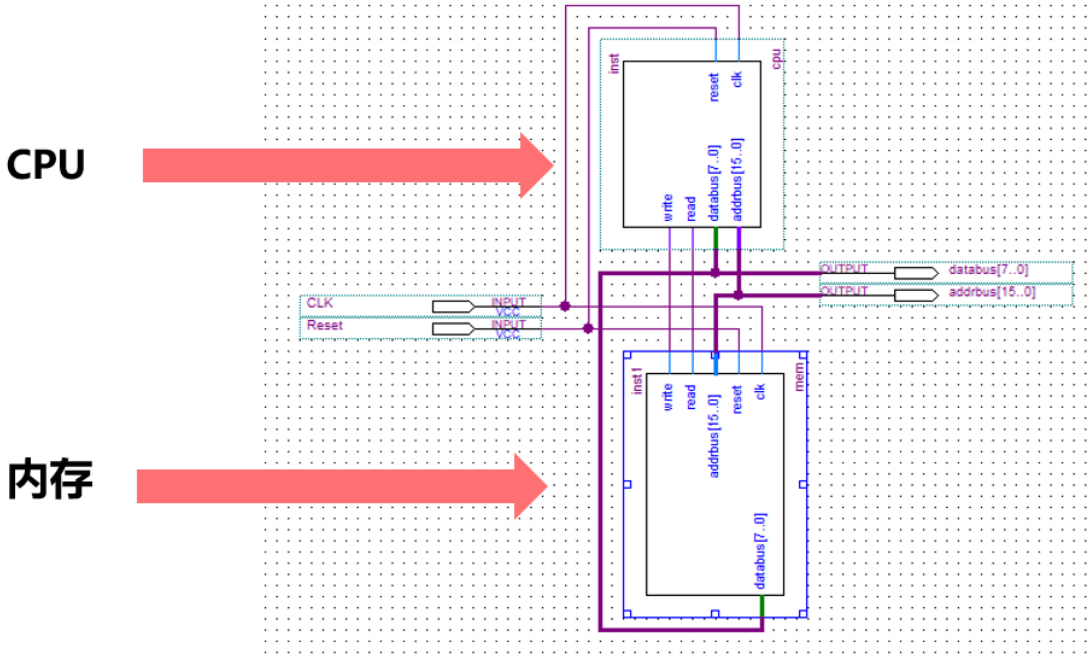
指令	1st Loop	2nd Loop	3rd Loop	4th Loop	5th Loop
CLAC	AC＝0				
STAC total	total＝0				
STAC i	i＝0				
LDAC i	AC＝0	AC＝1	AC＝2	AC＝3	AC＝4
INAC	AC＝1	AC＝2	AC＝3	AC＝4	AC＝5
STAC i	i＝1	i＝2	i＝3	i＝4	i＝5
MVAC	R＝1	R＝2	R＝3	R＝4	R＝5
LDAC total	AC＝0	AC＝1	AC＝3	AC＝6	AC＝10
ADD	AC＝1	AC＝3	AC＝6	AC＝10	AC＝15
STAC total	total＝1	total＝3	total＝6	total＝10	total＝15
LDAC n	AC＝5	AC＝5	AC＝5	AC＝5	AC＝5
SUB	AC＝4, Z＝0	AC＝3, Z＝0	AC＝2, Z＝0	AC＝1, Z＝0	AC＝0, Z＝0
JPNZ Loop	JUMP	JUMP	JUMP	JUMP	NO JUMP

所以根据要实现的功能，mem 里面保存的内容如下：

```
signal memdata: memtype(4095 downto 0) := (
0 => RSCLAC,
1 => RSSTAC,--m[total_total]<=ac    total=0
2 => std_logic_vector(to_unsigned(total_addr, 8)),
3 => X"00",
4 => RSSTAC,--m[i_addr]<=ac    i=0
5 => std_logic_vector(to_unsigned(i_addr, 8)),
6 => X"00",
7 => RSLDAC,  -- loop    --ac<=m[i_addr]    ac=i
8 => std_logic_vector(to_unsigned(i_addr, 8)),
9 => X"00",
10 => RSINAC, --ac++
11 => RSSTAC, --i=ac
12 => std_logic_vector(to_unsigned(i_addr, 8)),
13 => X"00",
14 => RSMVAC, --r=ac
```

```
15 => RSLDAC, --ac=total
16 => std_logic_vector(to_unsigned(total_addr, 8)),
17 => X"00",
18 => RSADD, --ac=ac+r
19 => RSSTAC, --total=ac
20 => std_logic_vector(to_unsigned(total_addr, 8)),
21 => X"00",
22 => RSLDAC, --ac=n
23 => std_logic_vector(to_unsigned(n_addr, 8)),
24 => X"00",
25 => RSSUB, --ac=ac-r
26 => RSJPNZ, --
27 => std_logic_vector(to_unsigned(loop_addr, 8)),
28 => X"00",
29 => X"00",  -- total
30 => X"00",  -- i
31 => "00000100",  -- n
others => RSNOP
```

分别用 VHDL 编写好 CPU 和 mem 的代码之后生成顶层图的形式，然后用连线的方式讲两个模块综合起来。



测试平台

Windows10　　Quartus 9.0

测试输入

我们采用从 1 累加到 n 的程序作为测试输入：

用相对简单CPU编程计算1 + 2 + ... + n。

算法步骤 ：

1：total＝0，i＝0

2：i＝i＋1

3：total＝total＋i

4：IF i≠n THEN GOTO 2

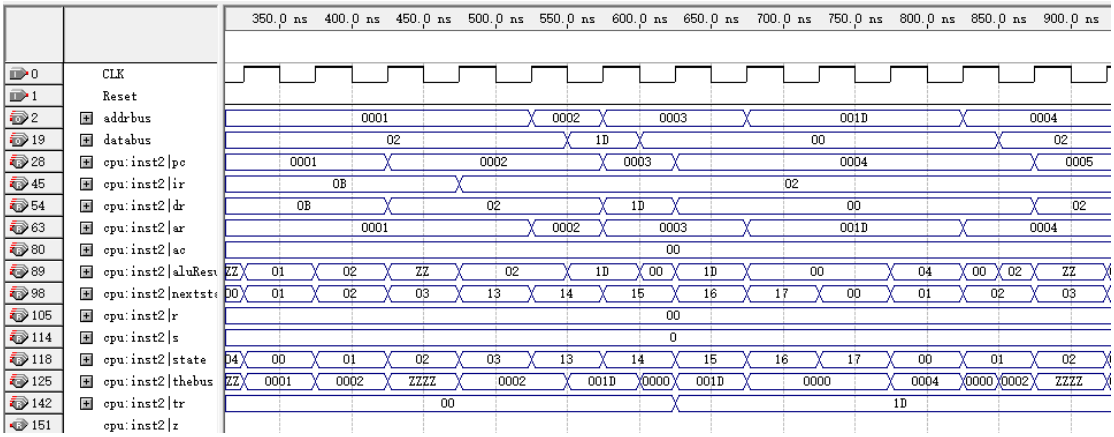
实现这一算法的相对简单CPU的代码如下：

```
        CLAC
        STAC  total      } total＝0，i＝0
        STAC  i
Loop:   LDAC  i
        INAC             } i＝i＋1
        STAC  i
        MVAC
        LDAC  total      } total＝total＋i
        ADD
        STAC  total
        LDAC  n
        SUB              } IF i≠n THEN GOTO Loop
        JPNZ  Loop
total:
i:
```

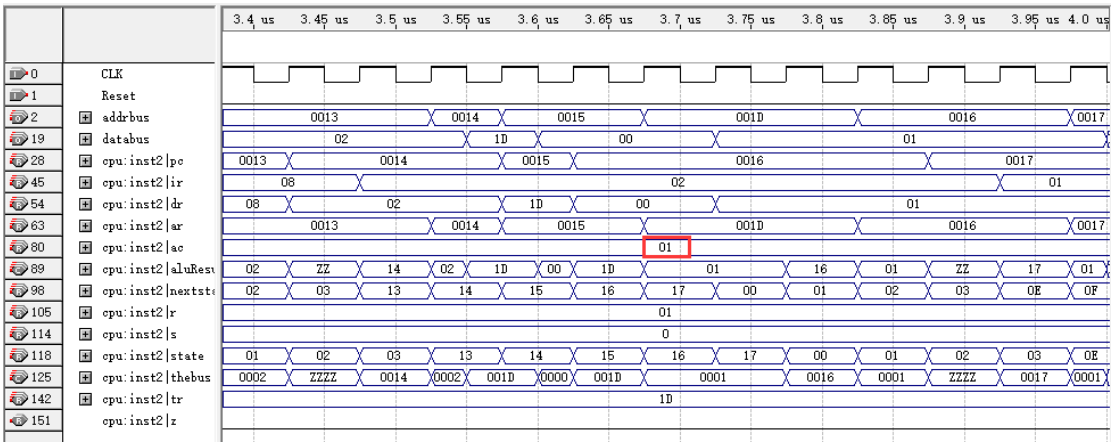
测试记录

相对简单 CPU 运行测试程序的 PC 寄存器、IR 寄存器、AC 累加器等信号波形截图如下：

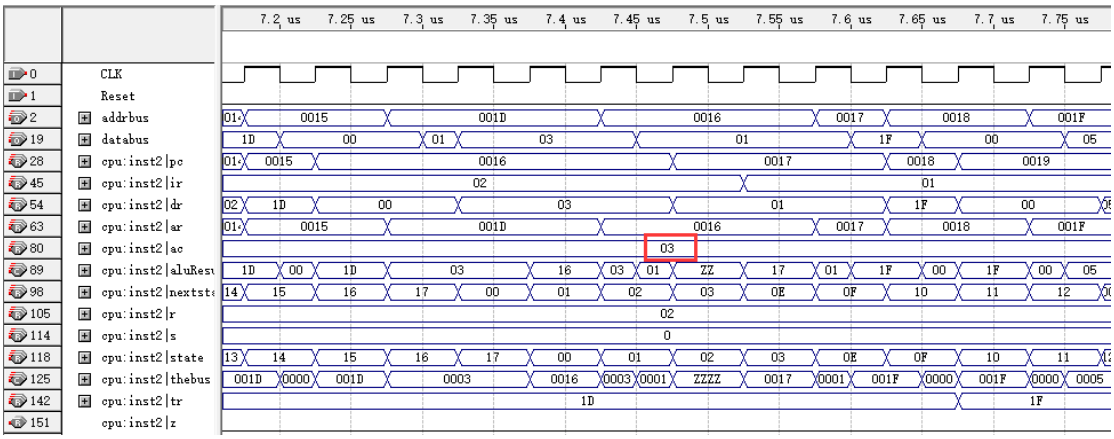
初始状态时各个寄存器的状态如下图所示：



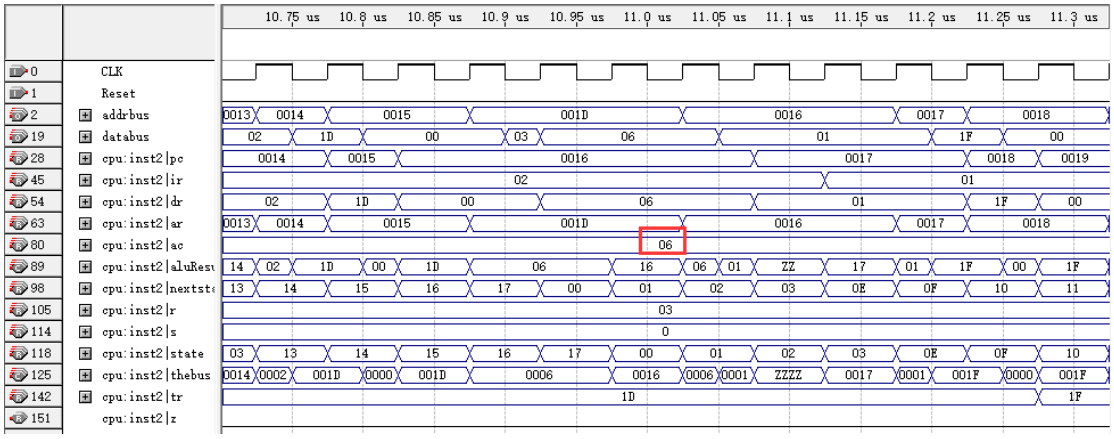
当 total 的值为 1 时各个寄存器的值如下：



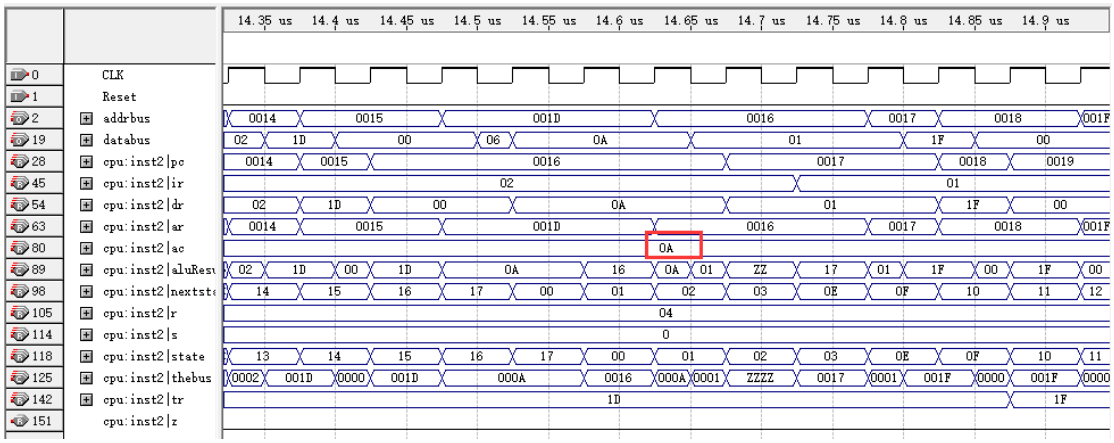
当 total 的值为 3 时各个寄存器的值如下：



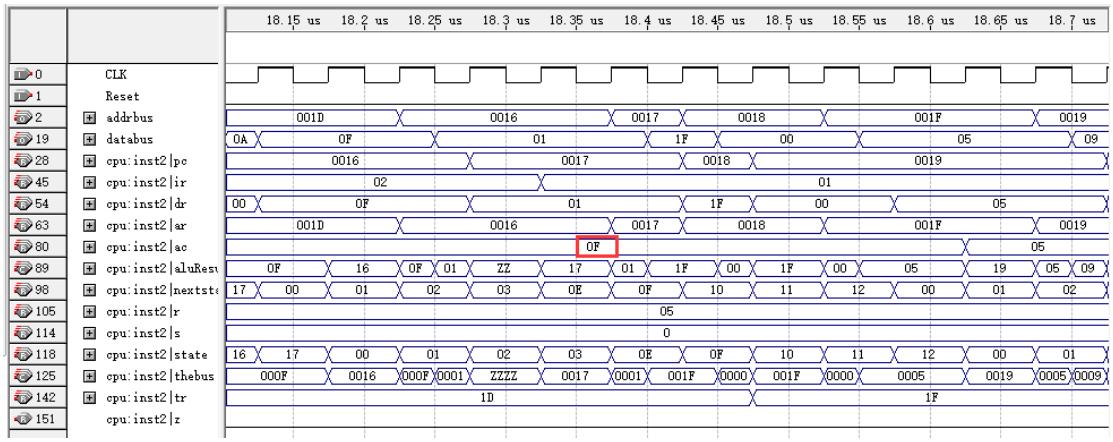
当 total 的值为 6 时各个寄存器的值如下：



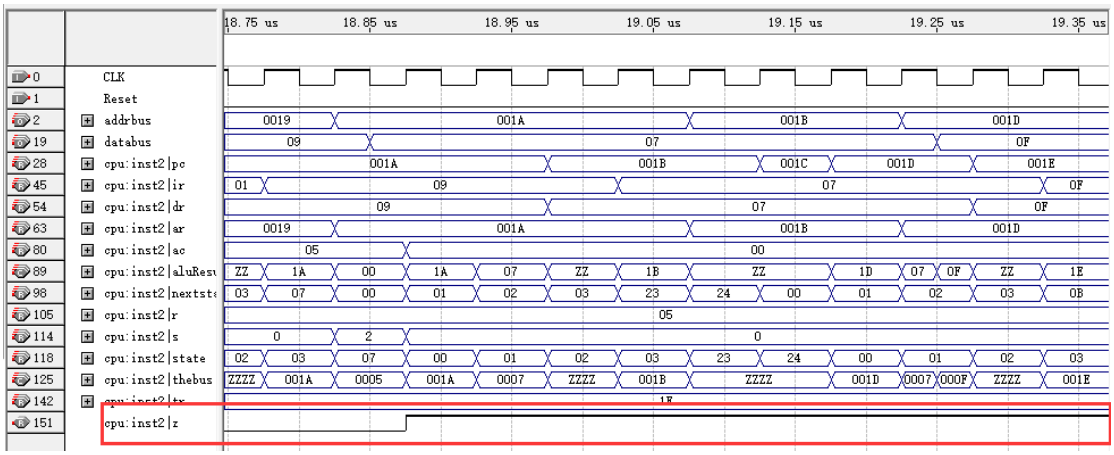
当 total 的值为 10 时各个寄存器的值如下：



当 total 的值为 15 时各个寄存器的值如下：



当计算完 1+2+3+4+5 的值之后，Z 的值由 0 变为 1，计算过程结束。



分析和结论

从测试记录来看，对各个寄存器的值以及状态之间的转换做了分析，相对简单 CPU 实现了对测试程序指令的读取、译码和执行，得到的运算结果正确。根据分析结果，可以认为所设计的相对简单 CPU 实现了所要求的功能，完成了实验目标。这次实验让我了解了设计 CPU 的一般步骤，更加掌握了 VHDL 语言，通过这次试验，学会了将一个大的工程分成很多小的模块，分别实现了模块之后，在对这些模块进行测试，然后将这些模块综合在一起，实现最终的目标。