实验报告

班级：智能 1601　　学号：201608010219　　姓名：罗枭鸿

实验题目：相对简单的 CPU 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

相对简单的 CPU 的设计需求

相对简单 CPU 的设计需求请详见课件，主要特征如下：

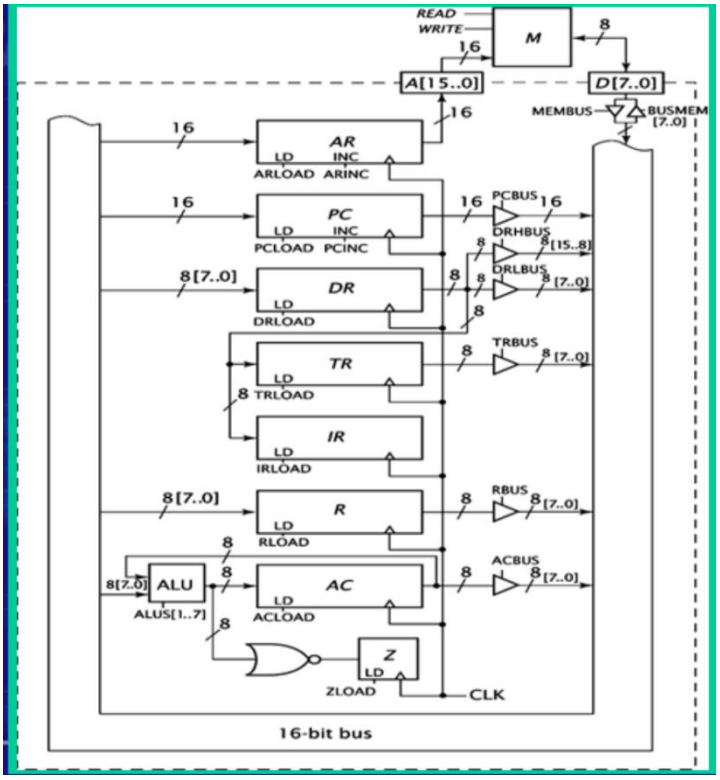
- 地址总线 16 位，数据总线 8 位
- 有一个 8 位累加寄存器 AC，一个 8 位通用寄存器 R，一个 1 位的零标志
- 有一个 16 位 AR 寄存器，一个 16 位程序计数器 PC，一个 8 位数据寄存器 DR，一个 8 位指令寄存器 IR，一个 8 位临时寄存器 TR
- 有 16 条指令，每条指令 1 个或 3 个字节，其中操作码 8 位。3 字节的指令有 16 位的地址

以下是 16 条指令对应操作码和操作：

指令	指令码	操作
NOP	0 0 0 0 0000	无
LDAC	0 0 0 0 0001 Γ	AC←M[Γ]
STAC	0 0 0 0 0010 Γ	M[Γ]←AC
MVAC	0 0 0 0 0011	R←AC
MOVR	0 0 0 0 0100	AC←R
JUMP	0 0 0 0 0101 Γ	GOTO Γ
JMPZ	0 0 0 0 0110 Γ	IF (Z=1) THEN GOTO Γ
JPNZ	0 0 0 0 0111 Γ	IF (Z=0) THEN GOTO Γ

ADD	0 0 0 0 1000	AC←AC+R , IF (AC+R=0) THEN Z←1 ELSE Z←0
SUB	0 0 0 0 1001	AC←AC-R , IF (AC-R=0) THEN Z←1 ELSE Z←0
INAC	0 0 0 0 1010	AC←AC+1 , IF (AC+1=0) THEN Z←1 ELSE Z←0
CLAC	0 0 0 0 1011	AC←0 , Z←1
AND	0 0 0 0 1100	AC←AC∧R , IF (AC∧R=0) THEN Z←1 ELSE Z←0
OR	0 0 0 0 1101	AC←AC∨R , IF (AC∨R=0) THEN Z←1 ELSE Z←0
XOR	0 0 0 0 1110	AC←AC⊕R , IF (AC⊕R=0) THEN Z←1 ELSE Z←0
NOT	0 0 0 0 1111	AC←AC' , IF (AC'=0) THEN Z←1 ELSE Z←0

CPU 的数据通路如下图所示：



相对简单的 CPU 的设计方案

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，FETCH4
3. 译码体现为从 FETCH4 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. 每条指令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

代码可以分 4 个文件进行编写：

rsisa.vhd、mem.vhd、cpu.vhd、comp.vhd

rsisa.vhd：声明每条指令对应的变量名。

mem.vhd：内存的代码，在这里声明了内存的大小、初始化内存（填入指令和数据），并规定了读写信号（write、read）有效时内存的动作。

cpu.vhd：相对简单的 CPU 的代码，在这里声明了 CPU 的内部组成、CPU 可能达到的各个状态，和 CPU 处于各个状态下的动作。

comp.vhd:使用 component 语句讲 cpu 和 mem 实例化，并连接起来，成为一个统一系统。

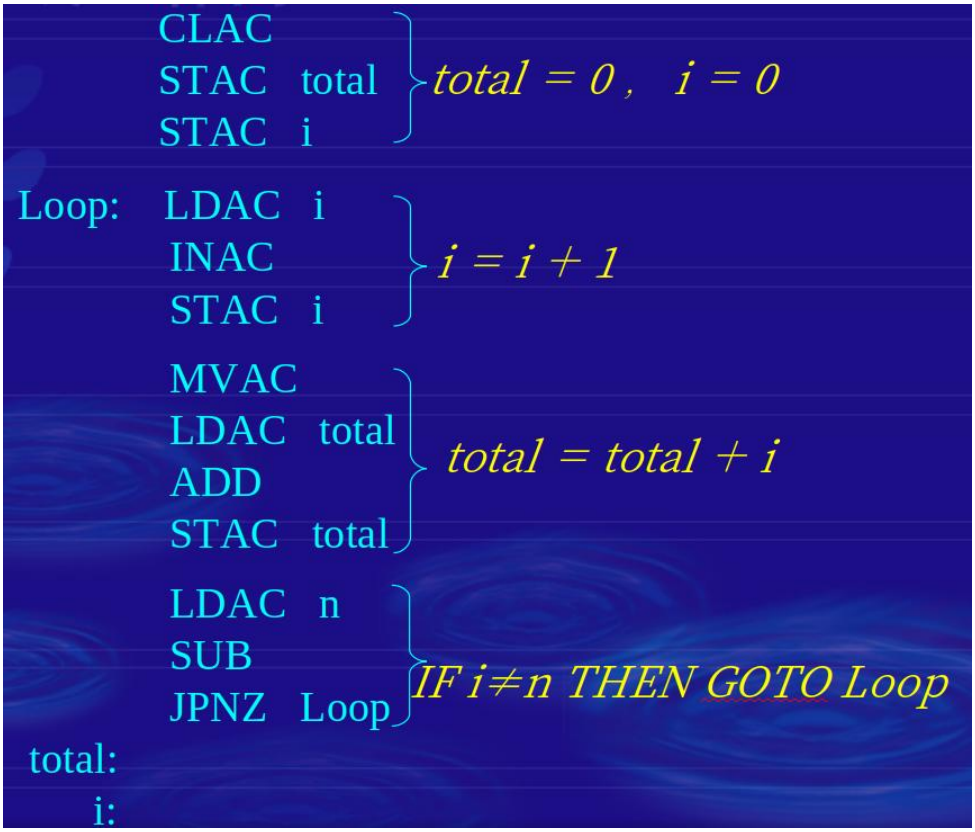
测试

相对简单 CPU 电路在如下机器上进行了测试：

部件	配置	备注
CPU 型号	core-i7 5500U	
内存	8GB	
操作系统	Ubuntu 18.04 LTS	中文版
综合软件	GHDL	
仿真软件	GHDL	
波形查看	GTKWave	

测试输入

我们采用从 1 累加到 n（n 设置成 8）的程序作为测试输入：



测试记录

波形截图如下所示，其中的信号：

clk：时钟信号

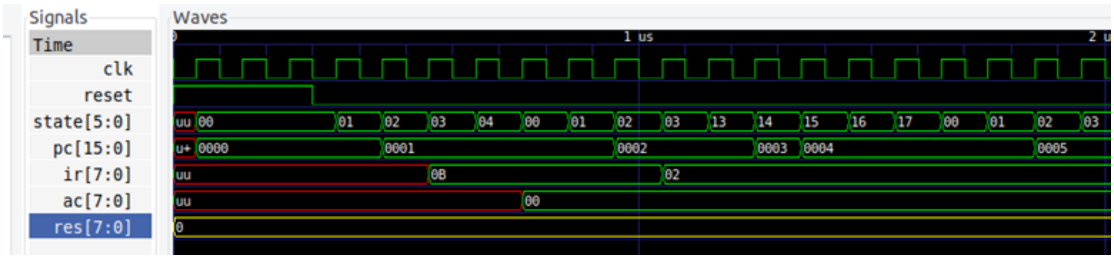
reset：重置信号

state：状态

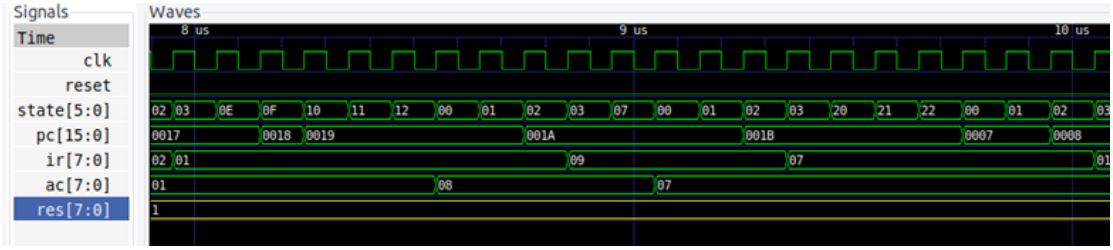
pc、ir、ac：CPU 中的三个关键寄存器

res：复制 total 的变量，用来查看 total 的值

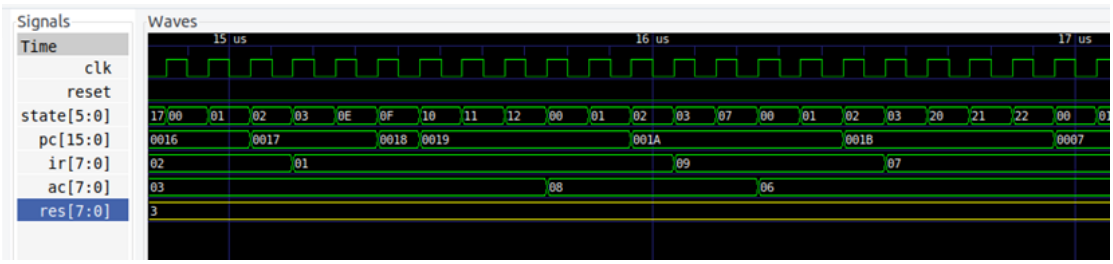
初始状态，res 等于 0：



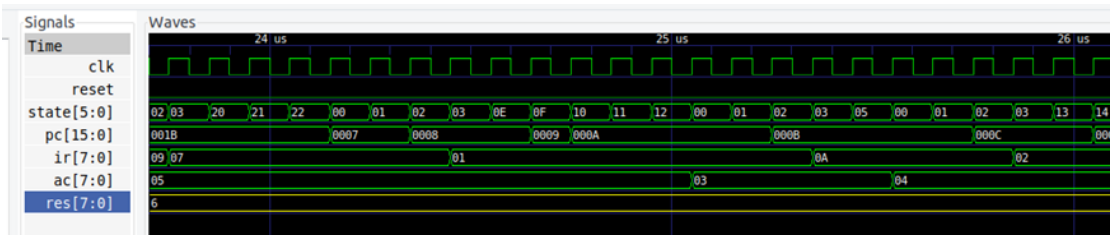
res 等于 1：



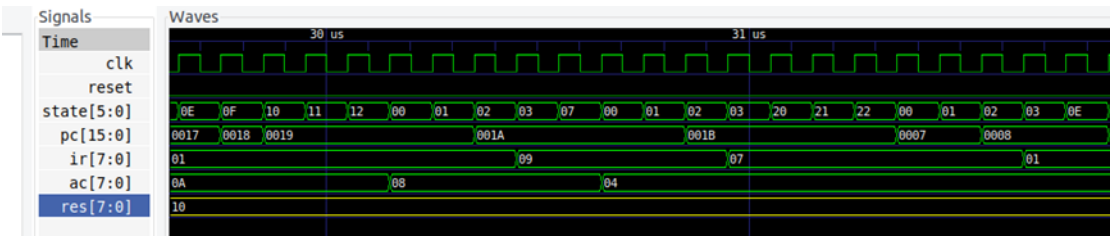
res 等于 3



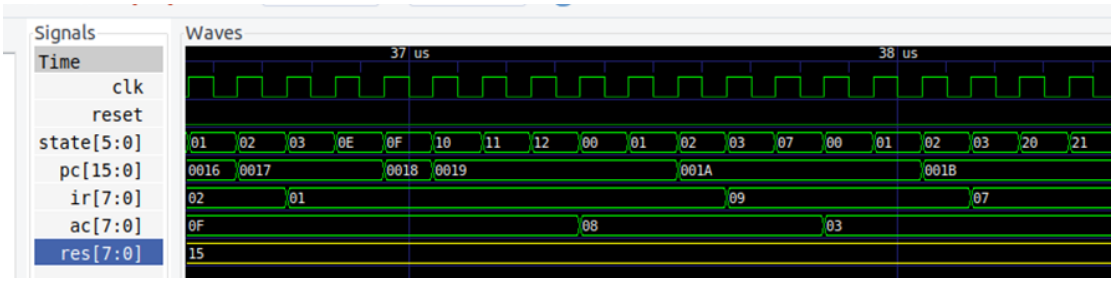
res 等于 6：



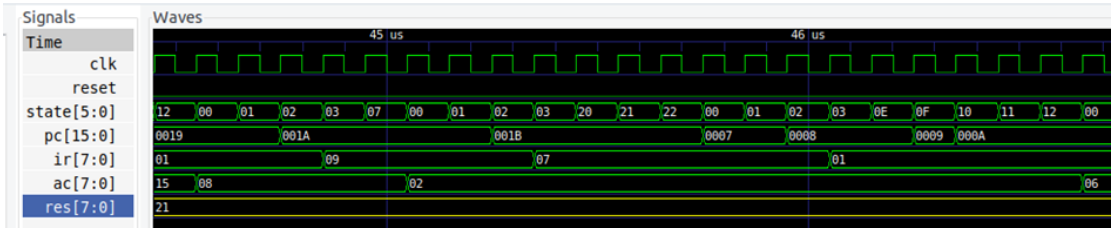
res 等于 10：



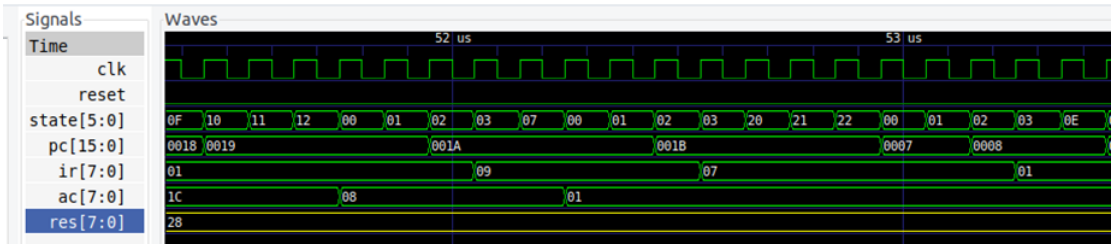
res 等于 15



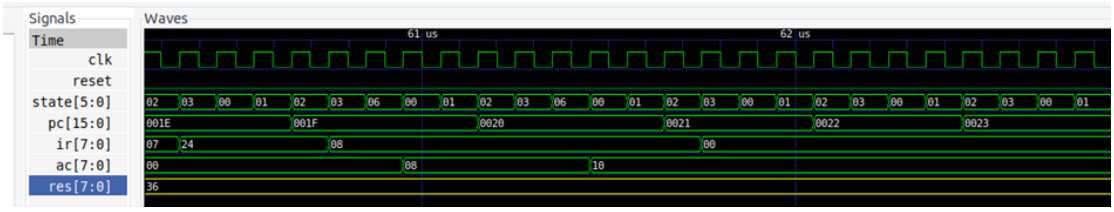
res 等于 21：



res 等于 28：



res 等于 36：



当 res 变为 36 之后不在发生变化。由于 n 设置成了 8，所以：totoal=0+1+2+3+4+5+6+7+8=36

仿真结果正确。