# 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

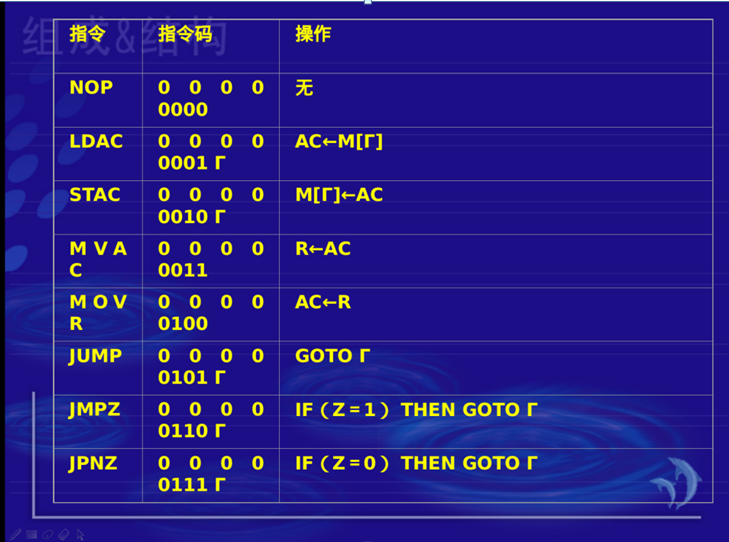
**实验内容**

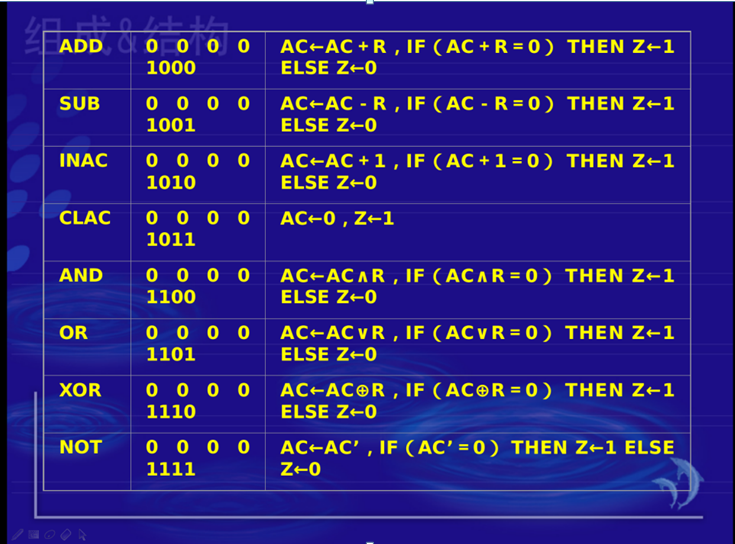
**相对简单的CPU的设计需求**

相对简单CPU的设计需求请详见课件，主要特征如下：

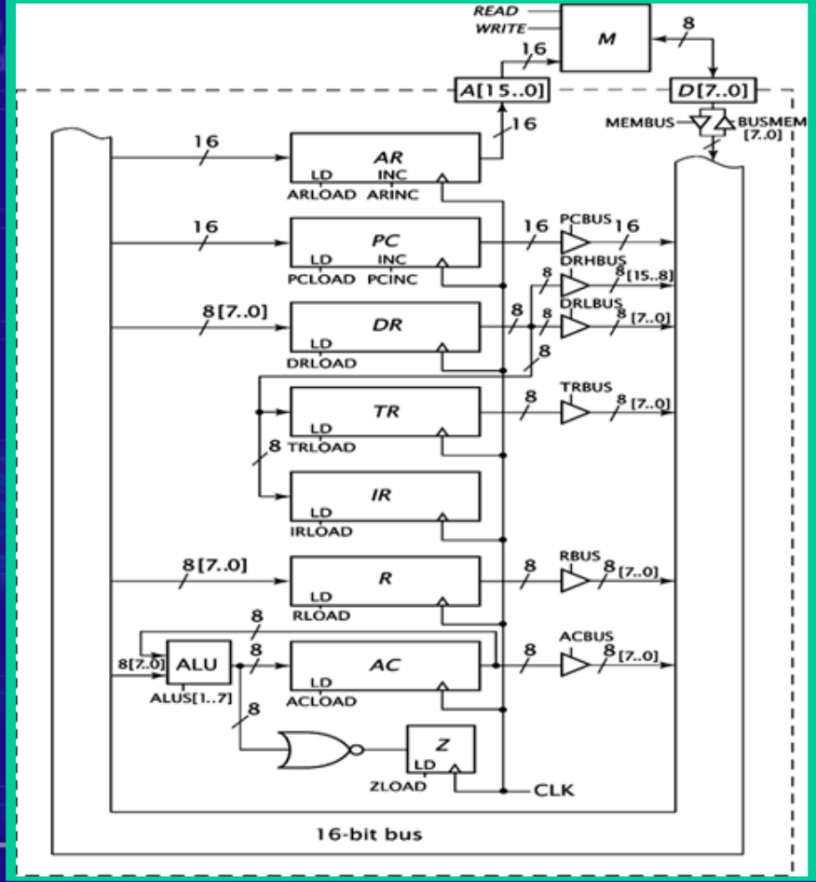
* 地址总线16位，数据总线8位
* 有一个8位累加寄存器AC，一个8位通用寄存器R，一个1位的零标志
* 有一个16位AR寄存器，一个16位程序计数器PC，一个8位数据寄存器DR，一个8位指令寄存器IR，一个8位临时寄存器TR
* 有16条指令，每条指令1个或3个字节，其中操作码8位。3字节的指令有16位的地址

以下是16条指令对应操作码和操作：





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。

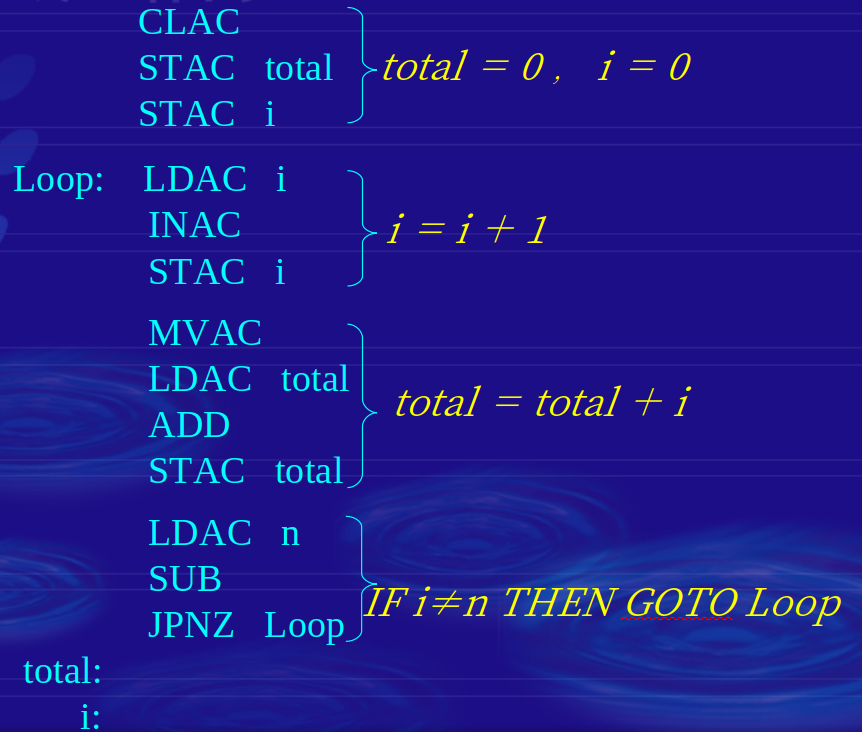
**测试**

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

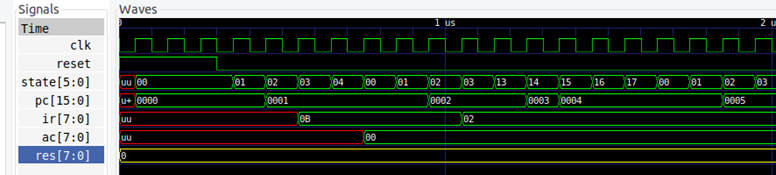
reset：重置信号

state：状态

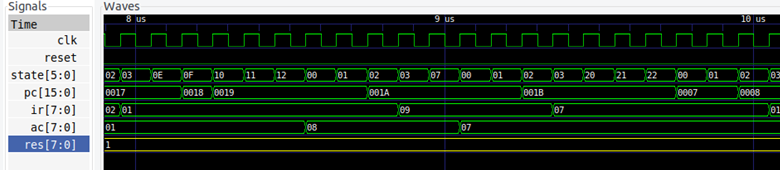
pc、ir、ac：CPU中的三个关键寄存器

res：复制total的变量，用来查看total的值

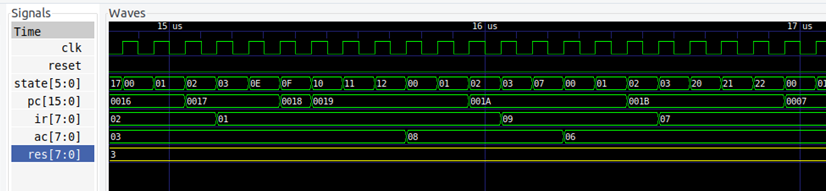
初始状态，res等于0：



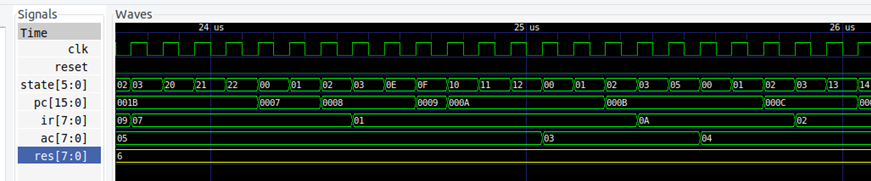
res等于1：



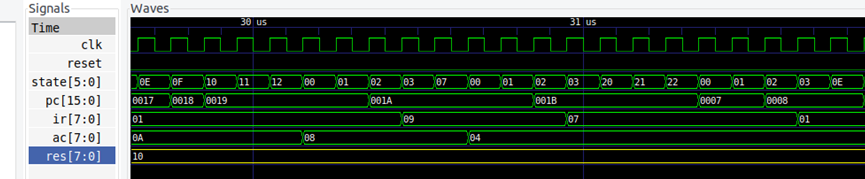
res等于3



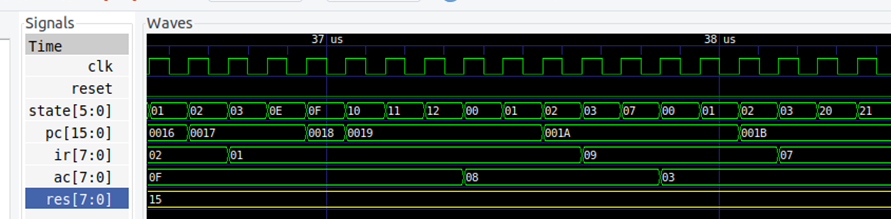
res等于6：



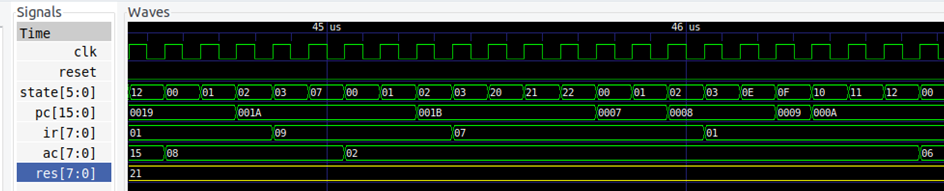
res等于10：



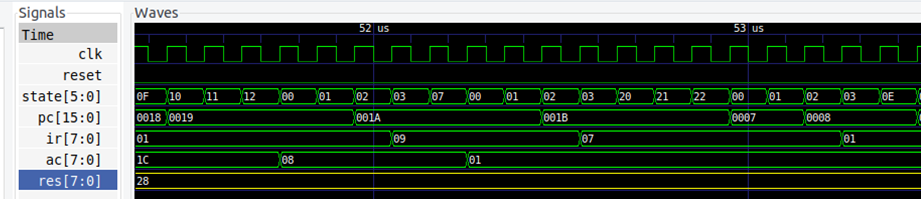
res等于15



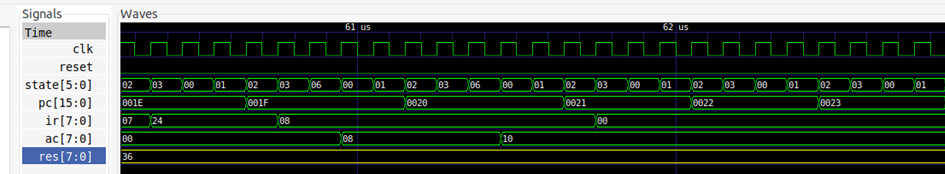
res等于21：



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。