**实验报告**

**智能1602 邱勒铭 201608010702**

**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4的程序进行测试

**实验内容**

**相对简单CPU的规格说明**

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

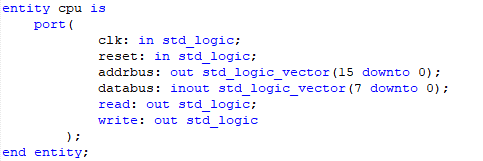
**相对简单CPU设计方案**

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

**CPU设计内容**

增加了FETCH4状态一共四个状态。取址包括四个状态，而译码体现为从FETCH4状态

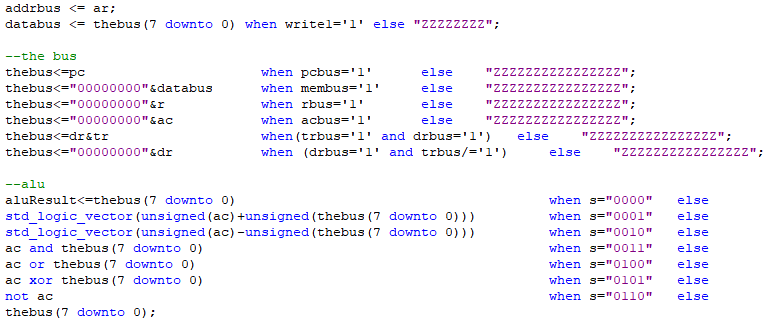
CPU的端口：



总体包含时钟信号，16位地址总线和8位数据总线，以及命令内存的read和write。

由于CPU是时序电路，我们需要知道当前状态和下一状态CPU的工作情况，因此要做好状态转移的工作。

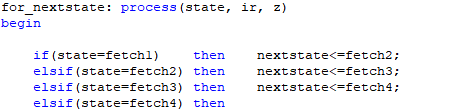
首先，CPU的地址总线和数据总线，以及ALU是非时序部分，将其单独写出：



之后是寄存器组，包括：程序计数器pc，地址寄存器ar，数据寄存器dr，指令寄存器ir，通用寄存器r，算术运算的专用寄存器ac，它们都是时序的，将其列为一组：

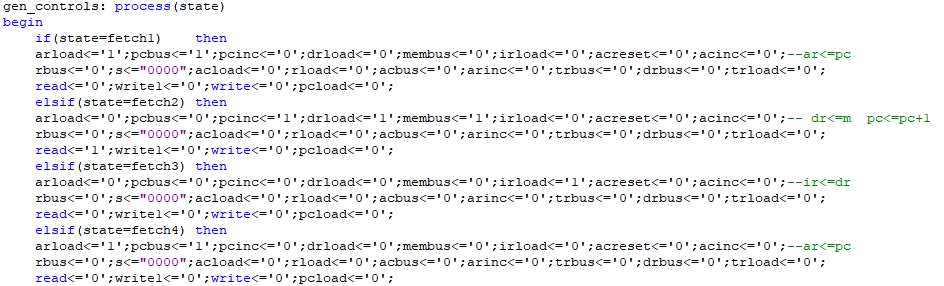


最后是状态转移组：



（部分）

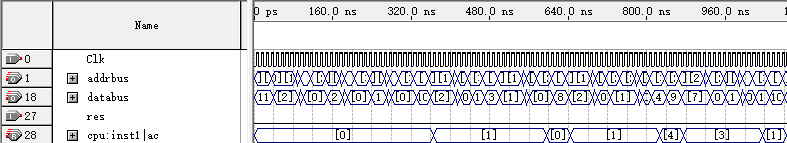
有了这些具体的转移逻辑和单个部件的控制关系，我们还需建立一个控制器来完成由指令到具体操作的转换：

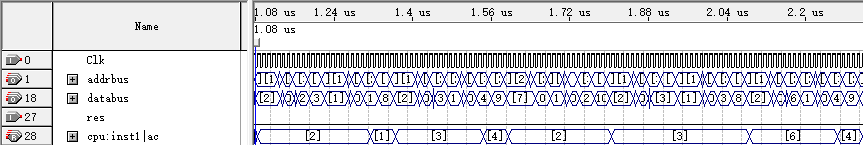


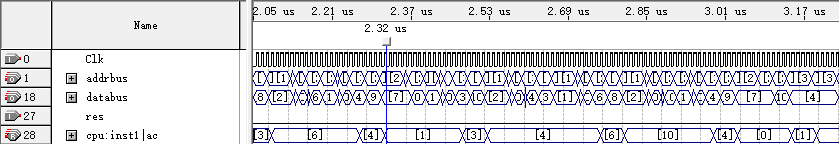
（局部）

**仿真结果**

可以看到仿真结果是正确的，到了0x24就不再变化了也就是36。







由ac可以追踪每次累加和加数，可以看到在2.85处结果是10，即累加结果。