第33卷 第10期 北京工业大学学报 Vol.33 No.10
2007年 10月 JOURNAL OF BEIJING UNIVERSITY OF TECHNOLOGY Oct. 2007

# 基于 Verilog HDL 的流水线模型机的设计与实现

易小琳，彭一凡
（北京工业大学 计算机学院，北京 100022）

**摘 要**：为了提高模型机指令执行的并行性，使用 Verilog HDL 并采取 top-down 设计方法，利用确定的有限状态自动机(DFA)理论，设计并实现了一台具有指令级并行性的流水线模型机的方案．阐述了该流水线模型机的 DFA 设计算法与 Verilog HDL 的实现方法，并给出了相应的仿真测试．测试结果证明，该模型机能并行处理 4 条指令，并具有预取指令和旁路功能．

**关键词**：流水线；Verilog HDL 描述；微处理器；确定的有限状态自动机
**中图分类号**：TP 303 **文献标识码**：A **文章编号**：0254－0037(2007)10－1096－06

提高计算机体系结构性能的发展方向之一是加快指令的解释过程．采用流水线技术的模型机可以并行处理多条指令，提高指令的执行效率，但是其设计复杂度明显高于顺序解释方式的简单模型机，而且所需要的暂存器更多[1]．Verilog HDL 可以使用户不需要直接接触相关硬件，只从电路的行为和结构来设计流水线内部结构[2-3]．本文使用 Verilog HDL 设计并实现了一个简单的静态流水线模型机．该模型机流水线采用 4 级流水，并且使用了旁路技术用于提高流水线的吞吐率[4-5]．模型机的设计采用了将确定的有限状态自动机(DFA)设计思想与 Verilog HDL 语言的特点相结合的方案[6]，成功地实现了流水线模型机运行过程的准确描述．此流水线模型机在 Altera 公司的 Quartus Ⅱ 6.0 Web Edition 上编译仿真，芯片为 Cyclone Ⅱ．计算机运行环境为 Intel (R) Pentium (R) Processor 1 400 MHz，768 MB 的内存．

## 1 模型机总体结构

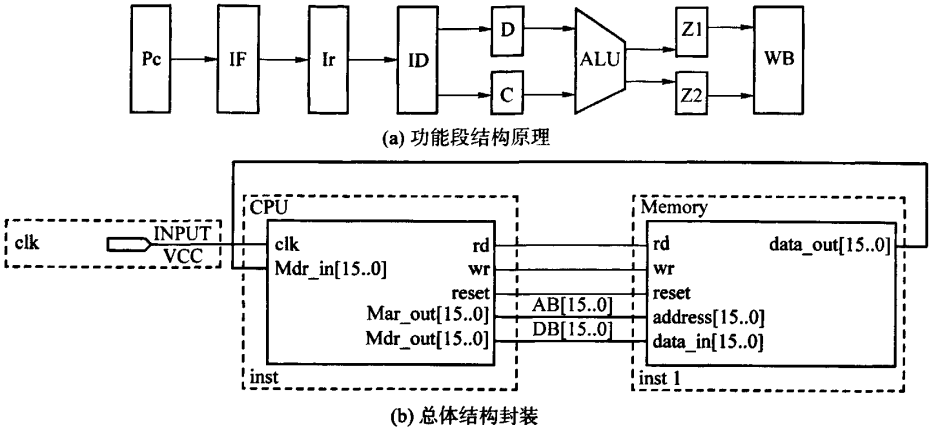
流水线模型机总体结构如图 1 所示[7-8]，主要由具有流水功能的 CPU 和 Memory(内存)组成．部件之间采用单总线数据通路结构．其中，信号 rd、wr 是脉冲触发信号，用于读写内存(每次读写 1 个字)，它们构成简单的控制总线；AB 是地址总线；DB 是数据总线．流水线模型机采用 4 段静态流水线，各段分别为取指(IF)、译码并取操作数(ID)、执行(EX)及写回(WB)．各段之间采用暂存器(组)相连．

### 1.1 寄存器及暂存器

本模型机使用的通用寄存器由 8 个 16 位寄存器组成．8 个 16 位指令寄存器堆(Ir)用于存放每个功能段正在执行的指令．16 位标志寄存器用于存储运算器的运算结果．16 位主存地址寄存器(Mar)作为 CPU 向系统地址总线 AB 发送地址信息的寄存器，根据该地址可访问主存．16 位主存数据寄存器(Mdr)用于存放 CPU 与主存之间传送的数据．

各功能段之间的寄存器和暂存器及其功能见图 1．指令计数器(Pc)用于记录 IF 段将要执行的下条指令的地址．Ir 用于存放每个功能段正在执行的指令．暂存器 C、D 用于存放双操作数指令的 2 个操作数，如果是访存指令则 D 用于暂存存储器地址．暂存器 Z1、Z2 用于存放算术逻辑运算执行的结果．在此采用 2 个 16 位暂存器，其目的是为了便于进行乘除法的位扩展，并且其在发生先写后读(RAW)数据相关时能减少延迟．另外，Z2 还具有传递 D 中所存的存储器地址的作用．

收稿日期：2006-09-07．
作者简介：易小琳(1959－)，女，北京人，高级工程师．




(a) 功能段结构原理

(b) 总体结构封装

图 1　流水线模型机总体结构

Fig.1　The overview of the pipeline

### 1.2　内存储器

在指令执行过程中，可以从内存中读取数据，并将其通过数据总线 DB 传送到 CPU. 反之，也可将 CPU 中的数据通过 DB 写入到内存中. 在信号 rd 上升沿到来时，根据 address 端口地址从内存把数据读出，送到数据总线 DB；同理，在信号 wr 上升沿到来时，将 DB 上数据写入地址 address 所指向的内存单元.

### 1.3　时钟发生器

本文采用状态机设计方法，因此只需使用单一时钟频率控制整机运行. 本模型机外来时钟信号为 clk，周期为 20 ns. 另外，为了使各条流水线的运行规整，经调试选择 8 个时钟周期作为 1 个流水段运行周期.

### 1.4　指令代码格式

本模型机采用 6 种指令格式，分别为算逻运算指令、寄存器直接寻址指令、存储器读指令、存储器写指令、跳转指令和停机指令. 每条指令为 16 位，高 2 位用于区分算逻运算、转移、跳转和停机指令. 在转移指令中，13 位和 12 位用于细分不同的转移功能.

算逻运算指令和寄存器间数据传送指令的 7～5 位与 4～2 位分别用于存放双操作数的寄存器地址，其中 7～5 位同时为目的操作数的寄存器地址. 存储器读指令和存储器写指令的低 8 位用于存放存储器地址，10～8 位则存放另一个操作数的寄存器地址；读与写的区别通过操作码进行判断. 跳转指令的低 8 位用于设置新的程序计数器值，即对应于新的指令地址. 停机指令只有操作码，其功能是停止机器运转.

## 2　流水线的 DFA 设计

### 2.1　暂存器组织

本机为 4 段流水线，时空图如图 2 所示. 流水线模型机的各段之间采用暂存器进行联结，但并不是所有指令的执行过程都需要使用所有的暂存器，因此首先需要对各条指令在各个流水段中用到的暂存器数目进行归纳，以便进行数据相关处理和构建旁路. 另外，由于各条指令都需要 IF 段，即各条指令在 IF 段使用的暂存器和寄存器是相同的，分别为 Pc、Mar 和 Mdr，所以下文不再对各条指令的 IF 段进行讨论.

算逻运算指令在 ID 段需要取出双操作数，因此使用暂存器 C 和 D，其中，C 存放目的操作数，D 存放源操作数. 在 EX 段此种指令使用 Z1 用于存放运算结果. 在 WB 段对结果进行写回，所以需要对特定的

某一寄存器进行写操作,因此容易构成 RAW 相关.

寄存器数据传输指令在 ID 段和 EX 段均未使用暂存器,但由于在 WB 段也需要写回,因此与算逻运算指令在这一段的分析相似,易构成 RAW 相关.

| | | | | | | |
|---|---|---|---|---|---|---|
| 指令 $i$ | IF | ID | EX | WB | | |
| 指令 $i$+1 | | IF | ID | EX | WB | |
| 指令 $i$+2 | | | IF | ID | EX | WB |
| 指令 $i$+3 | | | | IF | ID | EX | WB |

图 2 流水线时空图

Fig.2 Pipeline space-time diagram

存储器读指令在 ID 段将存储器地址放入暂存器 D,在 EX 段又将该地址转移到 Z2 中. 这样,可以使 WB 段的逻辑行为变得规整,不仅简化了该段流水线的设计,而且既不耗费时钟周期,也不降低流水线的吞吐率. 此种指令在 WB 段需要进行访存操作,并对某一寄存器执行写操作. 前者会与 IF 段竞争存储器资源,后者可以造成 RAW 相关.

存储器写指令的 ID 段和 EX 段与读指令相同,不同之处在于 WB 段不会造成 RAW 相关,因为其只从寄存器堆中读取数据. 另外,由于所有的寄存器写操作都是在 WB 段完成,因此不会产生先读后写(WAR)相关.

### 2.2 IF 段的设计与实现

IF 段首先对指令寄存器堆的空间进行判断,判断其是否仍有空间读取下一条指令. 如果没有剩余的空间,则延迟 1 个周期再次进行判断.

由前面的分析可知,IF 段与 WB 段会构成资源相关,因此需要判定该相关是否存在,即判断当前 WB 段运行的指令是否为存储器读或写指令. 若是该 2 种指令,则向后延迟 1 个周期;否则就可以安全地进行取指操作. 在执行完取指操作之后,IF 段对该条指令进行识别,判断它是否为跳转指令,如果是,则更新 Pc 寄存器的数据. IF 段的算法如下.

IF1:如果 Ir 有空间,转移到 IF2;否则转移到 IF5.

IF2:如果 WB 段访存,转移到 IF5;否则转移到 IF3.

IF3:取指.

IF4:如果当前指令为跳转指令,则判断 ID 和 EXE 是否为算逻运算指令,如果是,则延迟 1 个操作周期后转移到 IF5;否则更新 Pc 后转移到 IF5.

IF5:延迟到本周期结束,转回 IF1.

### 2.3 ID 段的设计与实现

ID 段是整个流水线中最复杂的一段,它涉及到与 EX 段和 WB 段的冲突,是模型机的核心. 该段首先判断当前指令是否为算逻运算指令,只有此类指令在 ID 段才可能发生冲突和相关,其他指令或者读取 Ir 中的地址,或者跳过,均不影响流水线流程. 在当前指令为算逻运算指令时,则进行下一步判断. 此时根据流水线当前的不同状态分为 3 类.

1) ID 段仅与 EX 段重叠,重叠过程可参见图 3. 此时判断在 EX 段运行的指令类别,如果 EX 段运行的是算逻运算指令,则辨别其目的操作数是否为当前指令的源操作数或者目的操作数,二者居其一则构成 RAW 相关,于是向后延迟 1 个周期;否则从指令寄存器中取出双操作数. 如果 EX 段运行的是寄存器直接寻址指令,则只判断其目的操作数是否与当前的源操作数或者目的操作数相关;若相关,直接将 EX 段运行的指令的源操作数内容存入 C 或者 D,另一个操作数则从当前指令所给出的寄存器中取出. 如此便实现了旁路技术,并可以节省 1 个周期. 如果 EX 段运行的是存储器读指令,则判断其存入的寄存器是否与当前指令所需访问的寄存器相关,一旦相关则构成 RAW,延迟 1 个周期.

| | | | | | |
|---|---|---|---|---|---|
| 指令 $i$ | IF | ID | EX | WB | |
| 指令 $i$+1 | | IF | ID | EX | WB |

图 3 ID 段与 EX 段重叠示意图

Fig.3 The overlapping of ID and EX

2) ID 段仅与 WB 段重叠,重叠过程可参见图 4. 此时判断在 WB 段运行的指令类别. 如果 WB 段运行的是算逻运算指令,则辨别其目的操作数是否为当前指令的源操作数或者目的操作数,二者居其一则直

接将 WB 段运行指令的 Z1 内容存入 C 或者 D，另一个操作数从当前指令所指定的寄存器中取出．如果 WB 段运行的是寄存器直接寻址指令或者存储器读指令，则与 EX 段相应的处理方法相同．

3) ID 段与 EX 段、WB 段重叠，重叠过程可参见图 5．此时需要同时考虑在 EX 段与 WB 段运行的指令．如果 EX 段运行的是算逻运算指令，则辨别其目的操作数是否为当前指令的源操作数或者目的操作数，二者居其一则构成 RAW 相关，于是向后延迟 1 个周期；否则，没有相关，此时则继续判断当前指令是否与 WB 段构成冲突．如果在 WB 段执行算逻运算指令且构成冲突，则直接将 Z1 内容存入暂存器 C 或者 D；若 WB 段运行的是寄存器直接寻址指令且构成冲突，则将原操作数存入 C 或者 D；若 WB 段运行的是存储器读指令且构成冲突，则延迟 1 个周期．如果 EX 段运行的是寄存器直接寻址指令且构成冲突，同样判断 WB 段的 3 种情况．每种情况的处理方法与上面相同．如果 EX 段运行的是存储器读指令且构成冲突，则把当前指令直接延迟 1 个周期后再作判断．第 3 种情况是 EX 段与 ID 段不发生冲突，而冲突只发生在 ID 段与 WB 段之间，这时其处理方法与第 2 种情况相同．

| 指令 $i$ | IF | ID | EX | WB | | |
|---|---|---|---|---|---|---|
| 指令 $i$+1 | | 停顿 | IF | ID | EX | WB |

图 4　ID 段与 WB 段重叠示意图

Fig.4　The overlapping of ID and WB

| 指令 $i$ | IF | ID | EX | WB | | |
|---|---|---|---|---|---|---|
| 指令 $i$+1 | | IF | ID | EX | WB | |
| 指令 $i$+2 | | | IF | ID | EX | WB |

图 5　ID 段与 EX 段、WB 段重叠示意图

Fig.5　The overlapping of ID, EX and WB

### 2.4　EX 段的设计与实现

由图 2 的流水线时空图可知，EX 段不需要判断是否与其他段发生冲突．这是因为，首先本模型机采用顺序流水线方式，所以 IF 段与 ID 段不会与本段发生冲突；其次暂存器 C 与 D 用于暂存操作数，即使它们会与前面指令发生冲突，由于在 ID 段已经解决，所以也不会与 WB 段冲突．由上所述，本段只根据指令进行算逻运算或者传递存储器地址．算法如下．

EX1：如果当前指令为算逻运算指令，计算 C oprt D 存入 Z1，转移到 EX3．

EX2：如果当前指令为访存指令，则置 Z2 为 D 的值，转移到 EX3．

EX3：延迟到本周期结束，转回 EX1．

### 2.5　WB 段的设计与实现

作为流水线的最后一个功能段，自然不需要处理冲突问题，而只需根据指令完成不同的写回操作．这里不再对其赘述．拟定算法如下．

WB1：如果是算逻运算指令，将 Z1 写入相应的寄存器，转移到 WB5．

WB2：如果是寄存器间数据传输指令，将源寄存器的内容写入目的寄存器，转移到 WB5．

WB3：若是存储器读指令，以 Z2 为地址访存后，将数据写入目的寄存器，转移到 WB5．

WB4：如果是存储器写指令，以 Z2 为地址，把源寄存器的数据写入存储器，转移到 WB5．

WB5：延迟到本周期结束，转回 WB1．

### 2.6　指令寄存器堆控制

在 Ir 的设计过程中，引入数据结构中循环队列的思想，设置了 8 个 16 位寄存器，采用先进先出的方式．另外，设置变量 head 和 tail 作为该队列的头尾指针，并且认为 tail + 1 与 head 相等时该队列已满．其中，IF 段控制队列的尾指针不断地向该队列压入新的指令；WB 段控制队列的头指针，当其运行完毕后将该条命令从队列中清除．ID 段与 EX 段也各自有自己的指针，控制相关指令的流程．

## 3　仿真结果

内存单元存入下列指令序列：

```
mem[4'h0]<=16'h7000;//MOV AX,[00]
mem[4'h1]<=16'h7301;//MOV BX,[01]
mem[4'h2]<=16'h7102;//MOV CX,[10]
mem[4'h3]<=16'h7203;//MOV DX,[11]
mem[4'h4]<=16'h0108;//ADD AX, DX
mem[4'h5]<=16'h0164;//ADD BX, CX
mem[4'h6]<=16'h0160;//ADD BX, AX
mem[4'h7]<=16'h6300;//MOV [00], BX
mem[4'h8]<=16'h800f;//JMP 0f
mem[4'hf]<=16'hc000;//HALT
```

得到仿真结果如图 6 所示.

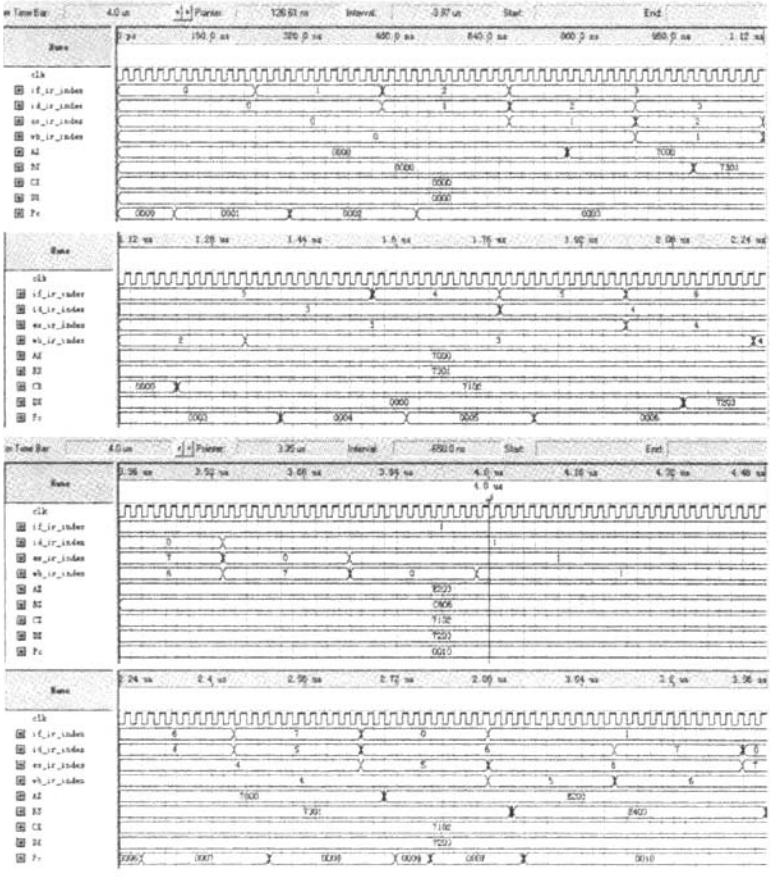


图 6 仿真测试波形

Fig. 6 The simulation of the pipeline

图 6 中,if_ir_index、id_ir_index、ex_ir_index、wb_ir_index 表示 4 个功能段当前执行指令的序号;AX、BX、CX、DX 是 4 个寄存器;Pc 是程序计数器. 从图 6 中可以看出,第 3 条指令的取指阶段延迟 3 个周期,这是因为发生了 IF 段与 WB 段的访存冲突;第 4 条指令的 ID 段延迟 2 个周期,因为在 DX 段发生了读写相关;第 6 条指令的 ID 段延迟 1 个周期,因为发生 BX、AX 的读写相关,但是在加入旁路以后使得延迟从 2 个周期减少到 1 个周期.

## 4 结束语

本流水线模型机采用 top-down 设计方法，利用 Verilog HDL 和确定的有限状态自动机理论实现了简单流水线模型机的功能．本模型机的总体设计合理，便于进行进一步的优化和现有机器指令的扩充．另外由于本模型机的每个模块不仅是可仿真的，也是可综合的，所以从物理意义上说，它与实际的模型机没有本质区别，即可以在分配管脚后通过相应的试验平台将其烧制在可编程逻辑芯片中．

通过使用 Verilog HDL 进行流水线模型机的设计、实现及仿真测试，可以很方便地对设计出的机器的各项性能指标进行评价，为高性能的流水线机设计提供了实现及评价的便利方案．

**参考文献：**

[1] SHEN J P, LIPASTI M H. 现代处理器设计[M]. 张承义, 邓宇, 王蕾, 译. 北京：电子工业出版社, 2004.

[2] JIANG Jiang. Research and implementation on instruction control pipeline in general-purpose EPIC microprocessor[J]. Mini-Micro Systems, 2006, 27(9): 1661-1664.

[3] 夏宇闻. Verilog 数字系统设计教程[M]. 北京：北京航空航天大学出版社, 2003.

[4] SRINIVASAN S T, RAJWAR Ravi, AKKARY H, et al. Continual flow pipelines[C]// Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems. New York: ACM Press, 2004: 107-109.

[5] HAMACHER Carl, VRANESIC Zvonko, ZAKY Safwat. Computer organization[M]. 5th ed. Beijing: China Machine Press, 2002.

[6] DRAGAN Milicev, ZORAN Jovanovic. Formal model of software pipelining loops with conditions[C]// Proceedings of the International Parallel Processing Symposium. Washington, D C: IEEE Computer Society, 1997: 554-558.

[7] 郑纬民，汤志忠．计算机系统结构[M]．北京：清华大学出版社，2004.

[8] PATTERSON D A, HENNESSY J L. 计算机组成与设计[M]. 第 3 版. 北京：机械工业出版社, 2006.

# Design and Implementation of a Pipeline Model Machine Based on Verilog HDL

YI Xiao-lin, PENG Yi-fan

(College of Computer Science, Beijing University of Technology, Beijing 100022, China)

**Abstract**: In order to raise parallelism of executing instructions by model machine, this paper introduces the schema of designing a pipeline model machine. Using Verilog HDL, a pipeline model machine with parallelism of instructions which is combined with top-down method and DFA is implemented. This paper describes the schema and some algorithms of the pipeline model machine and simulates this machine in the end. The simulation results show that the model machine can process 4 instructions at the same time, and has the per-formances of pre-fetching instructions and bypassing.

**Key words**: pipeline processing; Verilog HDL descriptions; microprocessor chips; deterministic finite automation (DFA)