

# 基于 VHDL 的复杂电路的 P-T 算法模型

刘丹非<sup>1</sup> 李曼义<sup>1</sup> 郭金怀<sup>2</sup>

(云南师范大学计算机科学系 物理系 昆明650092)<sup>1</sup>

(辽宁省监狱警察培训中心 辽宁122500)<sup>2</sup>

## Complicated Electric Circuit P-T Calculus Model Based on VHDL

LIU Dan-Fei<sup>1</sup> LI Man-Yi<sup>1</sup> GUO Jin-Huai<sup>2</sup>

(Department of Computer Science, Department of Physics, Yunan Normal University, Kunming 650092)<sup>1</sup>

(Liaoning Province Train Center for Prison Policeman, Liaoning 122500)<sup>2</sup>

**Abstract** When we design electric circuit with the hardware describe language VHDL, if the control of the electric circuit is more than to calculate, we can design electric circuit as a controller which is based on multiplexer and is divided into the space part and the time part. Electric circuit is synthesized and form CPLD or FPGA circuit by adjusting the P- T arithmetic model. We explain this method by designing the controller of CPU as a example.

**Keywords** VHDL, Control electric circuit, Time, Space

近年来,随着大规模专用集成电路 ASIC (Application-Specific Integrated Circuit) 的开发和研制,进行复杂时序电路的设计,用传统的中小规模 IC 方法进行电路和数字系统的设计已不能满足要求或几乎难以完成,迫切需提高设计效率和为设计者提供方便但又实用的设计平台。在这样的技术和需求背景下,能大大降低设计难度的 VHDL 设计方法正越来越广泛地被采用。硬件描述语言 VHDL (VHSIC Hardware Description Language), 就是可以描述硬件电路的功能、信号连接关系及定时关系的语言。它能比电路原理图更有效地表示硬件电路的特性,这种设计是电路行为级的设计。设计者根据 VHDL 的语法规则,对系统目标的逻辑行为进行描述,然后通过综合工具进行电路结构的综合、编译、优化,通过仿真工具进行逻辑功能仿真和系统时延的仿真,可在最短的时间内,设计出高效、稳定、符合设计要求的大规模或超大规模的逻辑电路。

在数字逻辑系统中,控制单元通常用传统的有限状态机 (FSM) 或时钟模式的时序电路来建模,每个控制步可以看作一种状态,与每一步相关性的转移条件指定了时序电路的次态和输出,它有两种形态, Moore 型和 Mealy 型。Moore 型有限状态机,是指输出仅为有限状态当前状态的函数;而 Mealy 型有限状态机,是指输出仅为有限状态当前状态和有限状态机输入值的函数,如图1。

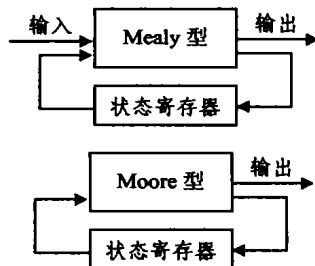


图1 有限状态机

## 1. 构造 P-T 算法模型

电路的控制多于电路中可能进行的算术运算,在电路的

行为描述中能把数据(输入)和控制操作(状态)分成两部分,同时在一个状态下,一组输入仅有一组确定的输出,这样在 P-T 算法模型的调度下使 P 算法和 T 算法成为统一模型。

对于一个时序复杂控制电路,依照数字逻辑电路的设计,在 VHDL 的电路行为描述中,可以把电路看成是空间(Place)和时间(Time)以及有相互制约关系的电路,空间部分是要产生控制电路的控制信号(控制码),而时间部分要把控制在时间上能分有限个状态,对于一个有限状态,它应该有“输入”,这个“输入”可以影响“状态转换”,而这个转换是一个状态转换函数,这个函数可以决定当前状态的次态。次态或新状态可被称为状态转换的输出状态。这个函数有有限个状态,每个状态又是输入的函数。这样,可以把上述问题看作时间和空间的函数,显然输出不仅决定于空间,也关联于时间,即 P-T 算法模型。

## 2. T 算法模型

时间	控制
s0	a1
	a3
	...
s1	a1
	a2
	a3
s2	a2
	a3
	...

图2 T 算法调度

假定控制可以分若干时间状态——若干步,在任意给定的时刻,必有惟一确定的状态,在某状态下必须依赖于外部输入的特定消息触发,才能引起状态转移或执行某种任务。在每一个时间状态中的控制操作取相同的时间进行,若各步为有限个状态 s0, s1, s2, ..., 不同的控制操作记作 a1, a2, a3, ...,

则它们的调度关系可用图2表示。

显然这是一个有限状态机,从一个状态进入另一个状态的条件取决于所执行的控制。

### 3. P 算法模型

每一个状态可以接受一组“输入”,这组“输入”的结果将导致一个“状态转换”的发生,也产生一组输出。状态转换基于“状态转换函数”,即由 T 算法模型决定。输出由 P 算法模型决定。假定电路可以分若干时间状态,在每一个时间状态中,电路的控制信号(输出 c1,c2,c3,c4,...)能看作是一组输入值的函数<sup>[1]</sup>,此算法可用图3来表示其关系。

输入	c1	c2	c3	c4	c5	...
a1	0	1	1	0	x	...
a2	x	1	0	0	1	...
a3	0	0	0	1	0	...
a4	x	x	1	0	1	...
...						

图3 P 算法调度

### 4. P-T 算法模型

综合 P 算法和 T 算法,而且在不同的状态下,输入相同,输出并不相同,P-T 算法模型可以用图4表示其调度关系。

时间	输入	c1	c2	C3	c4	c5	...
S0	a1	0	1	1	0	x	...
	a3	x	1	0	0	1	...
	...						
S1	a1	0	0	0	1	0	
	a2	x	x	1	0	1	
	a3	1	0	1	0	0	...
S3	a2	1	1	0	1	0	
	a3	0	0	1	1	1	
	...						

图4 P-T 算法调度

### 5. 应用举例

在清华大学计算机系的 TEC2000组成原理实验机扩展研发中<sup>[3]</sup>,将4片 Am2901组成的 ALU 以及控制器,改由 SPARTAN I (2.5vSRAM 工艺 FPGA)作开发芯片,沿用 TEC2000的指令系统及节拍<sup>[4]</sup>,按 P-T 算法模型,将控制分为8个状态,不定长指令周期,按图4的调度模型,用 CASE 语句形成控制输出。以下是 VHDL 代码表述。

#### 5.1 定义控制输入

```

CONSTANTS ADD:BIT_VECTOR := "00000000";
CONSTANTS SUB:BIT_VECTOR := "00000001";
CONSTANTS AND:BIT_VECTOR := "00000010";
-----ADD,SUB...TEC2000指令
    
```

#### 5.2 制作状态机<sup>[2]</sup>

```

TYPE STATE IS
(S0,S1,S2,S3,S4,S5,S6,S7);
SIGNAL PresentState : STATE;
SIGNAL NextState : STATE;
...
PROCESS (CP) -- PresentState -> NextState
    
```

```

BEGIN
IF CP'EVENT AND CP='1' THEN
PresentState <= NextState;
END IF;
END PROCESS SwitchToNextState;
PROCESS (PresentState)
BEGIN
CASE PresentState IS
WHEN S0 =>
NextState <= S1;
WHEN S1 =>
NextState <= S2;
WHEN S2 =>
IF A组指令 THEN
NextState <= S3;
ELSE
NextState <= S4; --B,C,D组指令
END IF;
WHEN S3 =>
NextState <= S1;
WHEN S4 =>
IF C组指令 THEN
NextState <= S6;
ELSE
NextState <= S5; --B,D组指令
END IF;
WHEN S5 =>
IF B组指令 THEN
NextState <= S1;
ELSE
NextState <= S6; --D组指令
END IF;
WHEN S6 =>
NextState <= S7;
WHEN S7 =>
NextState <= S1;
WHEN OTHERS =>
NextState <= NULL;
END CASE;
END PROCESS;
    
```

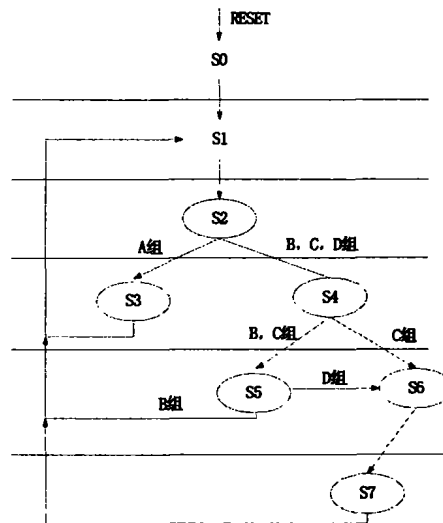


图5 TEC2000状态图

#### 5.3 生成输出逻辑

```

CASE STATE IS --形成34位控制信号 CON
WHEN S0 => CON <= "10001...";
WHEN S1 => CON <= "10001...";
WHEN S2 => CON <= "00100...";
WHEN S3 => CASE OP IS
WHEN ADD =>
CON <= "100"&SR&DR&...;
WHEN SUB =>
CON <= "100"&SR&DR&...;
WHEN AND =>
CON <= "100"&SR&DR&...;
-----SR,DR 为源和目的寄存器
END CASE;
WHEN S4 => CASE OP IS
WHEN JMPA => CON <= "100010...";
WHEN LDRR => CON <= "100"&SR&...;
...
END CASE;
WHEN S5 => CASE OP IS
WHEN JMPA => CON <= "001000..."; --
    
```

```

WHEN LDRR => CON<="0010000"&DR&"...";
...
WHEN S6 => CASE OP IS
  WHEN CALA => CON<="10000..."--;
WHEN S7 => CASE OP IS
  WHEN CALA => CON<="000010...";
WHEN OTHERS=> NULL;
END CASE;

```

**小结** 用 VHDL 语言设计复杂电路时,从大量的数据中能够分析出控制的重要方法是:电路中算术运算多,还是控制(多路选择)多,后者的优化和综合要求识别在转移内部和转移之间布尔简化的可能。上例中,CPU 的指令系统的每条指令均在一个状态时,产生所需控制信号,而每条指令所经历的状态是不同的。上述 P-T 算法模型是解决这类问题的一个较

为简洁的好方法。

### 参考文献

- 1 林敏. VHDL 数字系统设计与高层次综合第1版. 方颖立等,北京:电子工业出版社,2002. 1
- 2 卢毅. VHDL 与数字电路设计第1版. 赖杰,北京:科学出版社,2001. 4
- 3 王诚. 计算机组成与设计第1版. 北京:清华大学出版社,2002. 8
- 4 王诚. 计算机组成与设计实验指导第1版. 刘卫东等,北京:清华大学出版社,2002. 7

(上接第177页)

换组成,每一个 Givens 变换的获得需要5次算术运算和一次平方根运算,因此总共有  $\frac{5}{2}N^2 - \frac{5}{2}N$  次算术运算和  $\frac{1}{2}N^2 - \frac{1}{2}N$  次平方根运算。

综上所述,把三部分的计算量相加得总计算量为:  $\frac{19}{3}N^3 - 3N^2 - \frac{10}{3}N$  次算术运算和  $\frac{1}{2}N^2 - \frac{1}{2}N$  次平方根运算。

### 3.2 并行度

并行度的计算也要分为三个部分:Q 乘以 A 处,求  $R^{-1}$  处,以及  $R^{-1}$  乘以 Q 处。在 Q 乘以 A 处,这一步用  $2N-3$  个并行步完成,从算法的设计过程中易知每一步的并行度分别为  $1, 1, 2, 2, \dots, N/2-1, N/2-1, N/2, N/2-1, N/2-1, \dots, 2, 2, 1, 1$ ; 在求  $R^{-1}$  处,需要  $N-1$  个并行步完成,由其算法的设计易知并行度依次为  $1, 2, \dots, N-1$ ; 在  $R^{-1}$  乘以 Q 处,其算法设计的原理与 Q 乘以 A 时类似,也用  $2N-3$  个并行步完成,每步的并行度也分别为  $1, 1, 2, 2, \dots, N/2-1, N/2-1, N/2, N/2-1, N/2-1, \dots, 2, 2, 1, 1$ 。于是容易算得平均并行度为:  $(3N^2 - 3N) / (10N - 14)$ 。

### 3.3 通讯复杂度

按照前文所述的并行求逆算法,在假设有充分多的处理机的条件下来分析其通讯复杂度。同样分三个部分来分析:第一部分,计算 QA 时,共在  $2N-3$  个并行步内完成,设第  $i$  个并行步内的并行度为  $D_i$ ,则在每个并行步中有  $D_i \cdot 2 \cdot 8N$  个字节的数据需要通讯,故这一步中的通讯复杂度为  $\sum_{i=1}^{2N-3} (D_i \cdot T_{SR} + T_W \cdot D_i \cdot 2 \cdot 8N \cdot d)$ 。其中  $T_{SR}$  为发送处理机的发送开销和接收处理机的接收开销之和,  $T_W$  为网络带宽的倒数,即每字节传输时间,  $d$  为处理机间的平均通讯距离;第三部分,计算  $R^{-1}Q$  时,采用同第一部分相同的并行策略,通讯复杂度也相同;第二部分,计算  $R^{-1}$  时,共在  $N-1$  个并行步内完成,第  $i$  步的并行度为  $i$ ,每个并行步中共有  $i$  行  $i \cdot 8N$  个字节的数据需要通讯,故这一步中的通讯复杂度为  $\sum_{i=1}^{N-1} (i \cdot T_{SR} + T_W \cdot i \cdot 8N \cdot d)$ 。综上所述可得总通讯复杂度为:

$$T = 2 \cdot \sum_{i=1}^{2N-3} (D_i \cdot T_{SR} + T_W \cdot D_i \cdot 2 \cdot 8N \cdot d) + \sum_{i=1}^{N-1} (i \cdot T_{SR} + T_W \cdot i \cdot 8N \cdot d)$$

### 3.4 效率及加速比

若记以上算出的矩阵求逆算法的总计算复杂度为  $W_2$ , 总

通讯复杂度为  $T_2$ , 则其算法的效率可通过式  $E_2 = W_2 / (W_2 + T_2)$  计算出来。而加速比则为  $\alpha_2 = P \cdot E_2$ , 其中  $P$  为处理机的台数。

### 4. 模拟结果及分析

模拟环境: Intel pentium166MMX 软件平台: SCO Unix OpenServer 5. 20 PVM 3. 3110M 局域网。

有关矩阵求逆的模拟计算时间情况见表1。

表1 矩阵求逆模拟程序计算时间比较

矩阵阶数	串行时间(秒)	模拟4节点并行		模拟8节点并行		模拟16节点并行	
		时间(秒)	效率	时间(秒)	效率	时间(秒)	效率
64	1	15	27%	35	23%	70	23%
128	2	31	27%	68	24%	138	23%
256	10	68	59%	139	58%	278	58%

从表中计算时间的发展规律来看,当矩阵阶数增加一倍时,串行计算时间增加接近10倍,而并行计算时间只增加近两倍。因此可以预见的是,当矩阵阶数从256进一步增大,即计算量进一步增大,并行效率还应当能有适当提高。

**结束语** 本文提出一种基于 LogP 简化模型的矩阵求逆并行算法,该算法被用在 L 阵实现信号的频率、DOA 和极化的联合估计的并行处理中收到了良好的效果。实践验证了本算法的有效性。

### 参考文献

- 1 李晓梅,等. 可扩展的并行算法设计与分析. 国防工业出版社,1999
- 2 Dowling E M, Fu Z, Drafz R S. HARP: An Open Architecture for Parallel Matrix and Signal Process. IEEE Transactions on Parallel and Distributed Systems, Oct. 1993
- 3 Tsay J-C, Chang P-Y. Design of Efficient Regular Arrays for Matrix Multiplication by Two-Step Regularization. IEEE Transactions on Parallel and Distributed Systems, Feb. 1995
- 4 Hwang K, Xu Z W. Scalable Parallel Computers for Real-Time Signal Processing. IEEE Signal Processing Magazine, 1996
- 5 李晓梅,蒋增荣. 并行算法. 湖南科学技术出版社,1992
- 6 陈国良. 并行算法的设计与分析. 中国科学技术出版社,1993
- 7 李晓梅,等. 面向结构的并行算法——设计与分析. 国防科技大学出版社,1996
- 8 张林波,迟学斌,等. 网络并行计算与分布式编程环境. 科学出版社,1996