并行计算的 Petri 网建模和 FPGA 实现*

万 军[1a,2]，赵不贿[1a,1b]

(1. 江苏大学 a. 电气信息工程学院，b. 机械工业设施农业测控技术与装备重点实验室，江苏 镇江 212013；2. 常州大学 信息科学与工程学院，江苏 常州 213164)

摘 要：在分析现有 Petri 网建模及其 FPGA 实现方法的基础上，首先探讨了并行计算的 Petri 网建模方法，将并行计算任务分解成多个并行处理单元，用 IOPT 网为系统进行建模；然后提出了构造监控层和算法层的双层结构以实现模型到 FPGA 实现的具体映射，有效解决了 Petri 网模型中变迁只能表示简单加减运算的问题。通过矩阵乘法的应用示例，表明了上述方法的正确性和通用性。最后提出了进一步的研究方向。

关键词：并行计算；Petri 网；建模；FPGA

中图分类号：TP301　　**文献标志码**：A　　**文章编号**：1001-3695(2013)09-2660-04

doi:10.3969/j.issn.1001-3695.2013.09.025

Petri net modeling and FPGA implementation of parallel computing

WAN Jun[1a,2], ZHAO Bu-hui[1a,1b]

(1. *a. School of Electrical & Information Engineering, b. Key Laboratory of Facility Agriculture Measurement & Control Technology & Equipment of Machinery Industry, Jiangsu University, Zhenjiang Jiangsu* 212013, *China*; 2. *School of Information Science & Engineering, Changzhou University, Changzhou Jiangsu* 213164, *China*)

Abstract: In the analysis of existing methods of Petri net modeling and FPGA implementation of the models, this paper firstly discussed the Petri net modeling of parallel computing. This method included decomposing the parallel computing task into multiple parallel processing units, and modeling the system by using IOPT net. Then it proposed the double layer structure, which contained monitoring layer and of algorithm layer, in order to map the Petri net model to FPGA implementation. This implementation method was an effective solution to the problem that transitions of Petri nets model could only represent the simple addition and subtraction operations. A matrix multiplication application shows that the above methods are correct and generic. Finally, it gave further research directions.

Key words: parallel computing; Petri net; modeling; FPGA

0 引言

并行计算是指将顺序执行的计算任务分成可以同时执行的子任务，并行执行这些子任务，从而完成整个计算任务。并行计算由以下几个部分构成：并行计算的硬件平台、并行算法、并行程序设计和并行应用[1]。FPGA 器件以其丰富的硬件资源、细粒度并行能力、灵活的算法适应性和较低的功耗成为理想的可编程系统平台。近年来 FPGA 芯片的迅速发展以及其良好的可编程结构为实现并行计算提供了硬件保证。

Petri 网是一种用于系统的图形化建模工具，具有坚实的数学基础，它具有强大的描述异步、并发等能力，非常适合用于并行控制器的设计。国内外许多学者对 Petri 网的建模及其 FPGA 实现进行了深入的研究。文献[2～5]针对具体的应用分别提出了相关 Petri 网定义并给出了模型的 FPGA 实现方法。文献[2]通过时钟驱动逻辑来实施同步 Petri 网，针对 FPGA 内部结构中每个逻辑单元块输入数量较少的特点，定义了 K-基 Petri 网来完成 Petri 网模型到 FPGA 的实现。文献[3]探讨了一种异步控制器建模与 FPGA 实现的方法，用安全自主 Petri 网(SAPN)为离散事件系统建模，并利用电路元件映射完成 SAPN 的 FPGA 实现。文献[4,5]分别采用控制解释 Petri 网(CIPN)和信号解释 Petri 网(SIPN)对逻辑控制器建模，其中文献[4]提出了基于 Petri 网逻辑电路功能平行分解的方法，使 FPGA 中不同的可利用资源得到平衡使用，文献[5]完成了 SIPN 到 VHDL 的自动映射。为了方便进行 Petri 网模型到 VHDL 代码的转换，文献[6,7]都采用了 PNML 对 Petri 网模型进行描述。文献[6]中设计了 PN 基本构造块来模块化描述 PNML 中的元素，文献[7]开发了 PNML-VHDL 工具来完成 PNML 到 VHDL 代码的自动转换。国内也开展了 Petri 网模型的硬件实现研究，并给出了 Petri 网中 C/E 系统、P/T 系统和 T 时延 Petri 网的电路模型，模型中库所用触发器、变迁用门电路实现，是一种异步控制[8,9]。

但迄今为止，文献中描述的方法都可以归结为用 P/T 系统对实际系统进行建模的，变迁所代表的运算只能表示简单加减运算，不能代表复杂的运算，对不同的变迁发生采取不同的算法，这是 P/T 系统实现的难点所在。Petri 网本身是一种并发模型，只要赋予变迁或库所运算的含义即可代表并行计算的模型。本文结合 Petri 网的建模能力以及 FPGA 的并行计算优势，首先探讨了并行计算的 Petri 网建模方法，然后提出了 Petri

收稿日期：2013-01-20；**修回日期**：2013-03-11　　**基金项目**：国家自然科学基金资助项目(61070058)；江苏高校优势学科建设工程资助项目(苏政办发〔2011〕6 号)

作者简介：万军(1978-)，男，江西南昌人，博士研究生，主要研究方向为 Petri 网、嵌入式系统(13815074795@139.com)；赵不贿(1957-)，男，江苏高淳人，教授，博导，主要研究方向为 DEDS、EDA、Petri 网等.

网模型到 FPGA 实现的具体过程,并给出了一个应用实例,最后指明了进一步的研究方向。

1　并行计算的 Petri 网建模

1.1　IOPT 网简介

1)P/T 系统

六元组 $R=(S,T;F,K,W,M_0)$ 称为 P/T 系统,其中 $N=(S,T;F)$ 是一个有向网,简称网,其中 S 称为网系统 R 的库所集,T 称为变迁集,F 称为流关系,满足:

a)$S\cap T=\varnothing$。

b)$S\cup T\neq\varnothing$。

c)$F\subseteq S\times T\cup T\times S$(× 为笛卡尔积)。

d)$\mathrm{dom}(F)\cup\mathrm{cod}(F)=S\cup T$。其中 $\mathrm{dom}(F)=\{x\mid\exists y:(x,y)\in F\}$,$\mathrm{cod}(F)=\{y\mid\exists x:(x,y)\in F\}$ 分别为 F 的定义域和值域。

K、W 和 M 依次为 N 上的容量函数、权函数和标志。$K:S\to\mathrm{IN}\cup\{\omega\}$,$W:F\to\mathrm{IN}$,$M:S\to\mathrm{IN0}$。$\mathrm{IN0}=\{0,1,2,\cdots\}$,$\mathrm{IN}=\{1,2,\cdots\}$。$M_0$ 是初始标志。

2)IOPT 网

IOPT 网是普通 P/T 系统的扩展,其规范了网和环境之间的接口,从而方便为控制器建模。IOPT 网在普通 P/T 网的基础上增加了以下特性:

a)测试弧和测试弧权值;

b)变迁具有优先权;

c)变迁具有信号哨(输入信号的布尔函数);

d)变迁关联有输入事件和输出事件;

e)库所具有输出信号。

IOPT 网中的变迁发生条件为:当一个变迁有发生权且与之关联的外部条件为真时(输入事件和输入信号哨都为真),变迁发生。此外,如果一个变迁在所有变迁当中有最高的优先权,这个变迁与其他变迁冲突时,它能够发生。IOPT 网的形式化定义可参见文献[10]。

1.2　并行计算的 Petri 网建模

针对 FPGA 平台,并行计算采用的并行处理结构包括流水线结构和阵列结构两种。流水线结构将计算任务分解成多个子任务,由多个阶段依次完成,多组数据依次进入流水线电路同时进行不同阶段的计算,如图 1(a)所示。阵列结构由多个并行工作的运算处理单元(processing element,PE)构成[11],这些单元可同时接收整体数据的多个部分进行并行计算,也可以依次处理部分数据,直到完成全部数据的处理。PE 本身可以是简单的组合电路,也可以是复杂的时序电路。图 1(a)和(b)所示分别为流水线和阵列结构示意图。

本文采用数据驱动的阵列结构[12],即将并行计算任务分解成多个 PE,当某 PE 中所需的各数据有效时进行计算,多组并行排列的 PE 可同时激活。PE 本身可以是简单的串行处理,也可以包含复杂的并行处理。为并行计算所建立的同步 Petri 网模型如图 2 所示,其中输入库所 $A_1,A_2,\cdots,A_n$ 等与并行计算中输入数据的集合相联系,库所中的托肯表示具体的计算数据。变迁与某个具体的操作相联系,且隐含着与系统时钟同步,其中变迁 T 表示 N 个 PE 处理完成后所做的结束操作。为了图形的简洁,每个 PE 中的变迁所包含的输入库所集和输出库所集仅有一个元素。

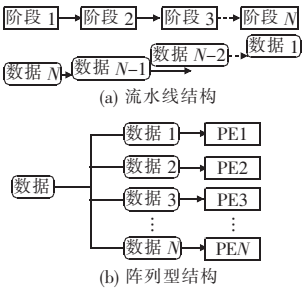

图 1　流水线型电路和并行阵列型电路

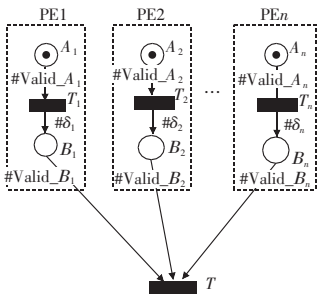

图 2　并行计算的 IOPT 模型

图 2 中库所旁的以"#"开始的字符串表示输出信号,变迁旁放置的以"#"开始的字符串表示信号哨。模型中变迁的激发条件为:

a)如果 $\delta_i\uparrow\xi=1$,则变迁 T_i 有发生权。其中:ξ 表示系统的时钟信号,$\uparrow\xi$ 表示时钟信号的上升沿(也可用下降沿);δ_i 表示与变迁 T_i 相关联的信号哨,当它为 0 时,与之相关联的变迁不允许发生,而为 1 时,与之相关联的变迁所有输入库所都有一个托肯时允许发生。

b)有发生权的变迁可以立即发生。

c)多个有发生权的变迁不发生冲突时能同时发生。

变迁发生时,对于变迁的前集中的库所,若用普通弧与变迁相连,则库所失去托肯,表示对应的数据已使用从而无效。若库所对应的数据被其他操作所共用,则采用测试弧与变迁相连。变迁后集的库所获得托肯,表示产生结果数据。为了指明数据的有效性,以便为后续变迁的发生条件所用,在 PE 内设置对应的输出信号,变迁发生同时也将修改输出信号的值。

2　并行计算的 FPGA 实现

并行计算的 FPGA 实现采用如图 3 所示的双层结构,具体包括:

a)监控层。该层由 IOPT 模型表示,根据并行计算的流程,实现所需的逻辑控制。

b)算法层。该层通过所设计的 FPGA 电路,实施相应的操作,产生计算的数据。

c)输出信号。监控层输出控制信号,控制算法层中的运算电路。

在该结构下,监控层相当于控制决策层,而算法层则实现具体的算法。针对不同的并行计算算法,可构建不同的监控模型,调动相应的算法模块在算法层进行组合。因此具有通用性和灵活性。

根据以上所述的实现结构,并行计算的 FPGA 实现过程即为将 IOPT 模型中的库所和变迁映射到相应的硬件描述。

库所在 IOPT 模型中代表 PE 的输入数据和输出结果,对应到算法层中的数据存储单元。具体实现时根据数据的类型和范围,映射成相应的 VHDL 数据类型。库所相关的输出信号表示数据的有效状态信息,用一位寄存器表示。

变迁表示某具体操作。根据 1.2 节中变迁发生条件的描述,系统在每个时钟周期检测变迁相关的信号哨,若条件满足则发生。因此变迁对应到算法层中某运算电路,变迁发生代表运算电路的执行,两者通过变迁发生时产生的输出信号进行通信。

Petri 网模型转换到 VHDL 描述过程以变迁为核心,将变迁及其前集和后集库所作为转换块。在图 4 中,库所 A、B 代

表两操作数,A 操作数有效时变量 Valid_A 置 1,B 操作数有效时变量 Valid_B 置 1。当 Valid_A 和 Valid_B 同为 1,且 Valid_C 为 0 时,变迁 T 发生,运算结果输出到 C 寄存器,且信号 Valid_C 置 1。Valid_C 为 1 后 T 不能再次激发。图中 A 、B 到 T 之间弧类型为测试弧,表明操作数被其他变迁共享。变迁 T 发生时通过输出事件 StartOperation 产生算法层电路的控制信号。

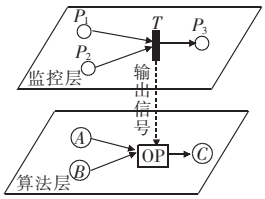

图 3 并行计算的 FPGA 实现结构

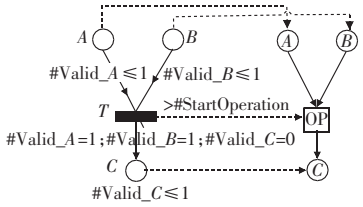

图 4 转换块及其算法层映射示意

图 4 中转换块对应的 VHDL 实现代码示意如下,其中信号 operateBZ 即为图 4 中输出事件 StartOperation 发生后的输出信号。

```
Valid_A,Valid_B,Valid_C:INOUT STD_LOGIC;
operateBZ:OUT STD_LOGIC;
process(clk)
begin
if rst'event and rst = '0' then
   Valid _C <= '0';
   operateBZ <= '0';
elseif clk'EVENT and clk = '0' then
    if(Valid_A = '1' and Valid _B = '1' and Valid _C = '0') then
     operateBZ <= '1';
     Valid _C <= '1';
    end if;
end if;
end process;
```

图 4 中算法层对应的 VHDL 实现代码示意如下,具体操作可通过调用相应 IP 核实现,输入信号 operateBZ 即为转换块中 T 变迁的输出信号,用于控制操作结果数据输出。

```
A,B:IN STD_LOGIC_VECTOR(7 DOWNTO 0);
C : OUT STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL C1:STD_LOGIC_VECTOR(15 DOWNTO 0);
Operate_ins :Operate PORT MAP(//IP 核例化,完成相应计算
clk⇒clk,rst⇒rst,
In1⇒A,In2⇒B,Out⇒C1);
process(operateBZ)
begin
   if(operateBZ'EVENT and operateBZ = '1') then
      C≤C1;//计算结果
   end if;
end process;
```

3 应用实例

矩阵计算是科学和工程应用的核心问题,其在信号处理、图像处理和无线通信系统等领域有着广泛的应用。目前,使用 FPGA 实现矩阵计算的研究比较广泛,但还面临着硬件编程、并行算法设计、硬件结构优化等挑战[13]。本文以矩阵乘法作为应用实例来突出建模和 FPGA 实现的方法。不失一般性,实例中仅考虑整数乘法。

对于矩阵乘法 $C = A \times B$,其中 A、B 和 C 分别为 $M \times L$、$L \times N$ 和 $M \times N$ 维矩阵,被定义成

$$C_{ij} = \sum_{k=1}^{L} A_{ik} B_{jk} \quad 1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N$$

矩阵乘的算法描述如下所示,其中矩阵 C 的行索引为 i,列索引为 j,求和操作采用下标 k。

```
for i = 1 to M do
   for j = 1 to N do
      for k = 1 to L do
         C[i,j] = C[i,j] + A[i,k] × B[k,j];
      end for
   end for
end for
```

对上述算法进行分析易知时间复杂度为 $M \times L \times N$,即为 $O(n^3)$。针对上述算法,可以构造包含 $M \times N$ 个 PE 的并行矩阵乘法器,同时实现矩阵 C 中每个元素的计算。该矩阵乘法器的时间复杂度降为 $O(n^2)$。

3.1 矩阵乘法的 Petri 网模型

1)顶层模型

假设矩阵乘法的 2 个输入矩阵为 3 阶方阵,建立的矩阵乘法器包含 9 个并行处理单元。PE 在开始运算之前需要先通过总线串行读入 18 个输入矩阵元素,送入相应寄存器中。图 5 为 3 阶方阵乘法的顶层 Petri 网模型示意图,其中:PE 用宏库所表示,宏库所 P_0 完成输入矩阵元素的串行输入;start 信号用于驱动 PE 执行,count 用于 P_0 内控制输入矩阵元素个数。P_0 完成后输出启动信号 start 置位,从而驱动 PE1 ~ PE9 开始并行计算。PE 计算结束后,相应的输出信号置位。图中当 Finish1 ~ Finish9 均置位时,表明本次矩阵乘法运算结束,T_1 发生,所有条件信号均置为无效值。当 count 值置 1 后且外部输入允许信号 InputEnable 有效后可激发下一次计算过程。

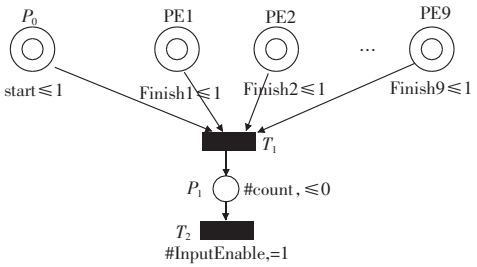

图 5 3 阶矩阵乘法的 Petri 网模型

2)PE 单元模型

图 6 是 PE 的 Petri 网模型示意图,具体实现如下:

$$C[1,1] = A[1,1] * B[1,1] + A[1,2] * B[2,1] + A[1,3] * B[3,1]$$

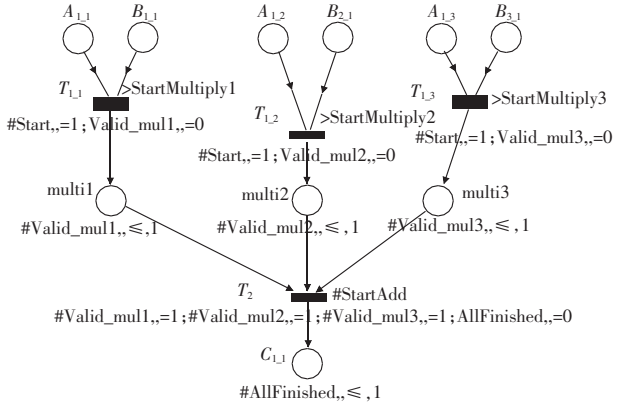

图 6 矩阵乘法的 PE 单元模型

此 PE 模型又包括三个并行的乘法转换块和一个加法转换块。Start 信号有效则 T_{1_1}、T_{1_2} 和 T_{1_3} 将同时发生,分别产生 StartMultiply1、StartMultiply2 和 StartMultiply3 事件,通过调用乘法 IP,分别获得 $A[1,1] * B[1,1]$、$A[1,2] * B[2,1]$ 和 $A[1,3] * B[3,1]$ 的计算数据,并置位相应输出信号;在 Valid_mul1、Valid_mul2 和 Valid_mul3 均有效后,T_2 发生,产生 Star-

tAdd 事件来执行加法操作,得到运算结果 $C[1,1]$,且置位输出信号 AddFinished,表明本次矩阵乘法运算结束。通过这种 PE 的阵列结构,可以完成任意维数的矩阵乘法并行计算。

3)数据输入模型

图 7 为数据串行输入的模型。图中源变迁在时钟信号的同步下,据信号 count 值依次发生,数据从外部总线读入到相应库所。经过 18 个时钟周期后,18 个库所均获得相应托肯。库所 B_{3_3} 获得托肯后,start 信号置位,表明矩阵元素输入完毕。

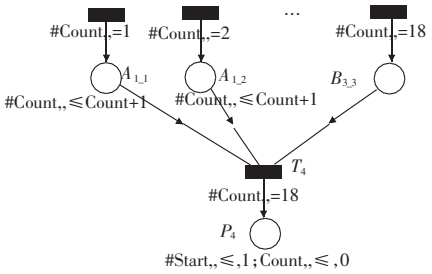

图 7　数据串行输入模型

3.2　模型的仿真

参照 3.1 节建立的矩阵乘法模型,选用 Xilinx Spartan3E 系列 XC3S500E 芯片来实现设计。并行处理单元中的乘法器使用 Xilinx 公司提供的乘法 IP 核,加法器使用定制的三输入 IP 核,一次乘法操作和加法操作可在一个时钟周期内完成。该实例具体实现的乘法表达式为

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}\times\begin{bmatrix}10&11&12\\13&14&15\\16&17&18\end{bmatrix}=\begin{bmatrix}84&201&318\\90&216&342\\96&231&366\end{bmatrix}$$

仿真结果如图 8 所示。从图中可以看出,data_in_en 信号启动后,矩阵数据经过串行输入至内部寄存器占用 18 个时钟周期,输入数据后经过 2 个时钟周期矩阵运算产生计算结果。图中显示了图 6 建立的 PE 单元模型的 FPGA 计算结果,其中 flag1、flag2 和 flag3 分别为图 6 模型中 StartMultiply1、StartMultiply2 和 StartMultiply3 事件所对应的输出信号,addbz 是 StartAdd 事件所对应的输出信号。

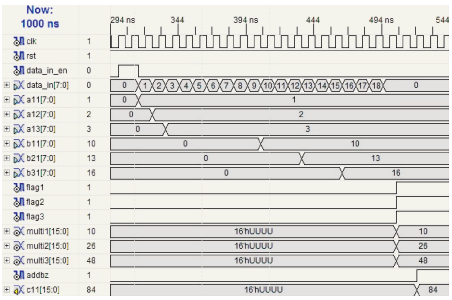

图 8　并行矩阵乘法功能

4　结束语

本文充分利用了 Petri 网的异步、并行建模能力和 FPGA 并行计算的能力,提出了一种并行计算的 Petri 网建模和 FPGA 实现方法。所述方法使用了 IOPT 网对并行计算流程进行建模,具体实施时将库所映射成计算数据,变迁映射成具体算法,调用不同的 IP 核即可实现各种计算。因此设计的方法具有通用性。下一步将在以下几个方面作深入的研究:

a)建模与实现工具设计。

现有的 Petri 网建模软件在描述诸如实数、负数等数据类型,以及权值可变的弧等方面显得力不从心。按照所建立的并行计算的 Petri 模型具体到 FPGA 实现时,尚缺乏一套自动化工具完成 Petri 网模型到 VHDL 程序映射。为此设计一套从规范级到实现级上提供完整服务的工具将非常重要和必要。所设计工具需要提供如下功能:(a)使用专用的编辑工具完成 Petri 网模型的建立,在建立过程中能灵活地配置库所对应的数据类型,选择变迁对应的 IP 算法;(b)实现 Petri 网模型到 VHDL 程序的自动转换;(c)调用 Modelsim 仿真软件仿真完成 VHDL 程序的仿真功能;(d)动态维护 IP 算法库,可增加和修改 IP 核。

b)异步模型设计。

本文建模采用的 IOPT 网仍为同步 Petri 网,即所建立的系统存在一个指定的全局时钟,变迁的条件必须在全局时钟信号到来时判别发生条件有效才能发生。若引入握手协议,用请求和应答信号来替代全局时钟,这样每个操作的完成与否不再受限于时钟周期。为了完成异步模型在 FPGA 的实现,需要开发具体的异步电路单元库。

参考文献:

[1] 陈国良,孙广中,徐云,等. 并行计算的一体化研究现状与发展趋势[J]. 中国科学,2009,54(8):1043-1049.

[2] CHANG N,KWON W H. FPGA-based implementation of synchronous Petri nets[C]//Proc of the 22nd IEEE International Conference on Industrial Electronics, Control, and Instrumentation. [S. l.]:IEEE Press,1996:469-474.

[3] UZAM M,KOC I B,GELEN G,*et al*. Asynchronous implementation of discrete event controllers based on safe automation Petri nets[J]. International Journal of Advanced Manufacturing Technology,2009,41(5-6):595-612.

[4] BUKOWIEC A,ADAMSKI M. Synthesis of Petri nets into FPGA with operation flexible memories[C]//Proc of the 15th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems. Piscataway:IEEE Computer Society,2012:16-21.

[5] WAGNER F,MÜNCH P,LIU S,*et al*. Development process for dependable high-performance controllers using Petri nets and FPGA technology[C]//Proc of the 1st IFAC Workshop on Dependable Control of Discrete Systems. 2007:139-144.

[6] KUBÁTOVÁ H. Direct implementation of Petri net based model in FPGA[C]//Proc of International Workshop on Discrete-Event System Design. Zielona Gora:University of Zielona Gora,2004:31-36.

[7] GOMES L,COSTA A,BARROS J P,*et al*. From Petri net models to VHDL implementation of digital controllers[C]//Proc of the 33rd Annual Conference of the IEEE Industrial Electronics Society. Piscataway:IEEE Computer Society,2007:94-99.

[8] 赵不贿,景亮,严仰光. Petri 网的硬件实现[J]. 软件学报,2002,13(8):1652-1657.

[9] 赵不贿,严仰光,陆继远,等. 控制电路基于 Petri 网的图形化设计与仿真研究[J]. 电路与系统学报,2005,10(5):54-58.

[10] COMES L,BARROS J P,COSTA A,*et al*. The input-output place-transition Petri net class and associated tools[C]//Proc of the 5th IEEE International Conference on Industrial Informatics. Piscataway:IEEE Computer Society,2007:509-514.

[11] 徐甲同,李学干. 并行处理技术[M]. 西安:西安电子科技大学出版社,1999:51.

[12] 邬贵明,窦勇,王森. 面向数据驱动处理器阵列的自动综合[J]. 计算机工程与科学,2009,31(A1):42-45.

[13] 邬贵明. FPGA 矩阵并行算法与结构[D]. 长沙:国防科学技术大学,2011.