

# 计算机体系结构的分类模型

沈绪榜<sup>1)</sup> 张发存<sup>1),2)</sup> 冯国臣<sup>1)</sup> 车得亮<sup>1)</sup> 王 光<sup>1)</sup>

<sup>1)</sup>(西安微电子技术研究所 西安 710054)

<sup>2)</sup>(西安理工大学计算机学院 西安 710048)

**摘 要** 根据计算机体系结构的发展,以指令流(instruction stream)计算、数据流(data stream)计算与构令流(configuration stream)计算的概念为基础,提出了一种新的计算机体系结构的分类模型。

**关键词** 指令流;数据流;构令流;软件;构件;流件;体系结构

中图法分类号 TP302

## The Classification Model of Computer Architectures

SHEN Xu-Bang<sup>1)</sup> ZHANG Fa-Cun<sup>1),2)</sup> FENG Guo-Chen<sup>1)</sup> CHE De-Liang<sup>1)</sup> WANG Guang<sup>1)</sup>

<sup>1)</sup>(Xi'an Microelectronic Technique Institute, Xi'an 710054)

<sup>2)</sup>(School of Computer, Xi'an University of Technology, Xi'an 710048)

**Abstract** According to the development of computer architectures, on the basis of the concept of Instruction Stream computation, Data Stream computation and Configuration Stream computation, this paper proposes a new classification model of computer architectures.

**Keywords** instruction stream; data stream; configuration stream; software; configware; flowware; architecture

## 1 引 言

从制造技术上讲,芯片的集成度是按摩尔定律成指数增长的,摩尔定律不是一个物理定律,而是一个经营管理的规律,目前已成为 IC 业制定规划的标准,只有达到这个标准,才能实现计算能力不变时,微处理器的价格和体积每 18 个月能减小 1 倍的目标.从设计技术上讲,芯片集成度的不断提高,促进了可重用的芯片实现形式的不断发展变化.如何改变和重新定义计算硬件的性质,这不但会影响软件和硬件的计算平台,而且也会影响计算机的应用领域.在多媒体、无线通信、数据通信与许多其它的嵌入式应用领域中,为了满足标准的不断更新和多标

准操作所带来的高吞吐量和超低功耗要求,可重构计算技术成了当前的研究热点.

重构计算已经为硬件设计开辟了一个新天地,可能导致计算“逻辑”的一次革命.硬件重构不仅能加快程序执行,而且也能调整计算平台.为了尽可能地降低可重构性的开销,设计者必须从硬件与软件两个方面考虑计算任务的优化划分,寻找可重构性与性能之间的平衡.人们估计到 2010 年,90% 以上的程序设计者都将是从事嵌入式系统的程序设计的,那时,可重构计算也将是程序设计者必须具备的基础知识之一.

可重构计算技术的发展,使计算机不仅有传统的基于指令流计算的体系结构和基于数据流计算的体系结构,而且有基于构令流的体系结构.所以,为

了更清楚地能对现代计算机的体系结构进行分类描述,需要有一种新的计算机体系结构分类模型.

本文以指令流(instruction stream)计算、数据流(data stream)计算与构令流(configuration stream)计算的概念为基础,研究了一种新的计算机体系结构的分类模型.

## 2 体系结构的发展

导致现在的多种可重用的 IC(Integrated Circuit)实现形式以及数字计算机体系结构发展的微电子技术,是从 1947 年晶体管的发明开始的,1947~1957 年是其研究与发展的 10 年;1957~1967 年开始了晶体管生产的商业化.当 1955 年晶体管用于计算机事业之后,便开始了微电子技术沿着 5 个重要方面演变的革命:晶体管缩小(shrinking transistor)、器件发明(device invention)、封装改进

(package improvement)、体系结构创新(architecture innovation)以及 CAD 设计(design by computer).按照 Makimoto's 的统计资料,IC 的实现形式是波浪式发展的,从标准化(standardization)形式到定制化(customization)形式再到标准化形式,以 10 年为周期周而复始地发展变化的,如图 1 中所示<sup>[1]</sup>.每一周期都有与 IC 集成度相适应的新的发展内容:芯片是 1958 年由 Kilby 发明的,1967~1977 年开始了定制芯片的商业化;微处理器是 1974 年由 Hoff 发明的,1977~1987 年开始了微处理器芯片商业化;1987~1997 年是定制逻辑芯片的 10 年,也许低功耗高集成度的 Flash 存储器件与可重构的逻辑器件预示着一个以用户编程(重构)能力为发展内容的 IC 标准化实现形式新周期的开始,使 IC 的标准化实现形式从可编程的芯片(处理器)形式,发展到可重构的芯片形式.

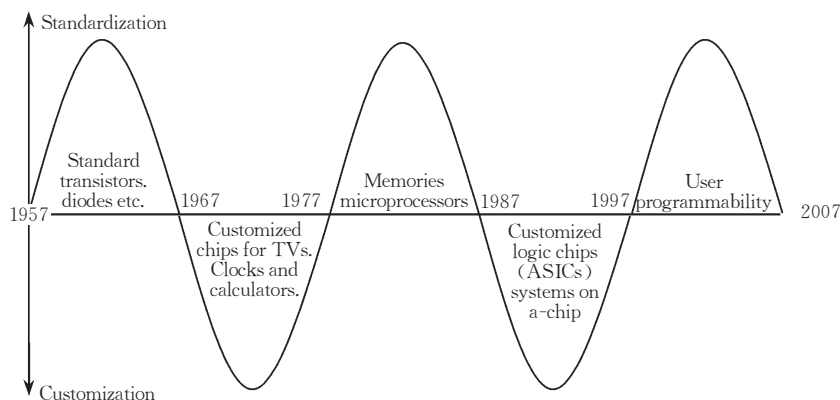


图 1 Makintoto 波形

### 2.1 基于指令流计算的体系结构

基于指令流计算的体系结构是目前主要的冯·诺依曼计算机体系结构.因为运算指令是由操作码与数据地址码两部分组成的,于是指令流包括了操作流与数据流,是按控制流原理(control flow paradigm)工作的,是应用最灵活的体系结构.

如图 1 中所示,20 世纪 60 年代后期,基于指令流计算的体系结构,是通过以 74/54 和 4000 系列为代表的 GPL(General-Purpose-Logic)可重用的定制芯片实现的.20 世纪 70 年代第一块存储器芯片与第一块处理器芯片问世之后,可重用性是以微处理器、存储器和 GPL 器件相结合的(称为嵌入式系统)硬件形式实现的,用户只需通过编写不同的应用软件就能实现不同的应用.20 世纪 70 年代后期,随着芯片集成度的进一步提高,便出现了流行的(称为 MCU(Micro-Controller Unit))的可重用的器件.

历史上,基于指令流的具体结构除了控制流体系结构外,还出现过非控制流体系结构.控制流体系结构是以过程语言(例如 BASIC, FORTRAN, PASCAL 和 C)的基本概念为基础的,基本的动作是赋值,语句执行是顺序控制的.非控制流体系结构则出现过 4 种:(1)以单赋值语言(例如, ID, LUCID, VAL 和 VALID)的基本概念为基础的,数据从一个语句“流”向另一个语句,语句的执行是数据驱动的,而且,标识符遵守单赋值规则,对应的是数据流体系结构;(2)以应用语言(例如: Puro, LISP, SASL 和 FP)的基本概念为基础的归约的体系结构;(3)以面向对象的语言(例如, SMALLTALK)的基本概念为基础的递归的体系结构;(4)以谓词逻辑语言(例如, PROLOG)的基本概念为基础的推理的体系结构.非控制流体系结构虽然有过多研究,但由于开销太大,没有得到实际应用,如今已没有继续研究.非

过程语言现在主要是在控制流计算机上,通过程序设计实现的,例如,面向对象的语言在控制流计算机上用程序实现是非常成功的.

2.2 基于数据流计算的体系结构

最简单的基于数据流计算的体系结构就是 ASIC 电路. 因为 ASIC 电路所要完成的功能是固定的,不需要操作码去定义其所要完成的功能,只需要用地址码去指明 ASIC 电路的输入数据的来源与输出结果的目的地,就能完成 ASIC 本身所定义的计算功能.

从 1987 年开始,美国卡内基·梅隆大学的 Kung 及其同事,通过研究算法计算与专用逻辑芯片体系结构的关系,提出了基于数据流计算的脉动阵列 (systolic array) 体系结构<sup>[2]</sup>. 脉动阵列是一个数据路径部件 DPU(Data Path Unit)的阵列,实际上是一个 PU 的阵列,如图 2 中所示. 它能完成大粒度的计算,例如,矩阵运算. 它的工作过程是:数据以一种有节奏的方式从数据存储器流出,通过流入 PU 阵列完成计算,计算结果从 PU 阵列流出,再流入数据存储器. 就像心脏收缩使血液进出心脏那样,作者遂称其为脉动阵列.

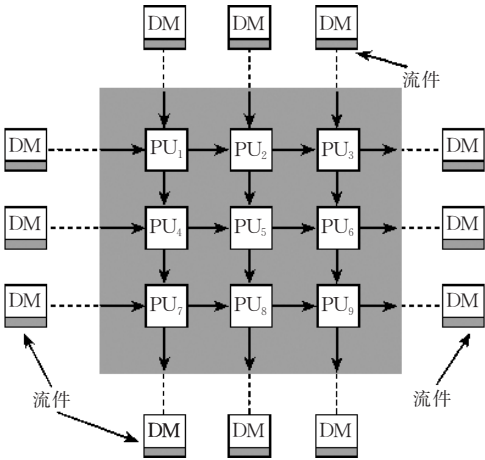


图 2 基于数据流计算的体系结构示意图

由于 PU 阵列的计算功能是事先按照算法的需要设计好的,不需要有控制 PU 阵列如何操作的命令,而只需要有控制数据存储器数据进出的 PU 阵列的命令,也就是对 PU 阵列来说,是按数据流计算的方式工作的,不需要有操作流的功能,而只需要有数据流的功能. 因此,控制 PU 阵列工作的命令,就没有像通常的指令那样有控制数据流与操作流的功能,而只有控制数据流的功能. 实质上,这就是现在人们称作流件(flowware)的起源.

流件只是用来定义数据进出 PU 阵列的端口与

时间步骤的,如图 2 中所示,与 PU 阵列的 12 个端口相联系的 12 个数据流,具体表明了流件的基本语义原理:从一个流件源产生的数据调度,决定在什么时候什么数据对象必须进入或离开哪一个 PU 阵列端口. 流件就是按照这种办法去安排嵌入式分布存储器的自动控制的数据存储器 DM(Data Memory)体,以产生所希望的数据流的.

指令流计算的指令流来自软件源,通过单一的计数器(program counter)控制执行,其数据流在指令的控制下流动;而数据流计算的数据流则来自流件源,在一个或多个数据计数器(data counters)控制下流动. 按照数据流体系结构的划分方案,允许每个自动控制的数据存储器体可以有一个数据计数器(如图 2 所示). 计数器是用来管理指令流的:读出下一条指令,改变指令地址,指令循环与嵌套,数据操作;指令地址计算与读出指令是有存储周期开销的;而数据计数器是用来管理数据流的:读出下一个数据,改变数据地址,数据循环与嵌套,但是没有数据操作;数据地址计算与读出数据的存储周期开销是可以避免的<sup>[3]</sup>.

文献<sup>[4]</sup>中的非控制流体系结构的数据流计算机已被淘汰,随着芯片集成度的提高,基于数据流计算的控制流体系结构得到了迅速发展.

2.3 基于指令流计算的体系结构

20 世纪 80 年代后期,随着芯片集成度的进一步提高,硬件从不可重构的实现形式,发展到了以 PLD 与 FPGA 为代表的各种细粒度的静态可重构的(reconfigurable)硬件实现形式,进一步提高了硬件的可重用性. 静态可重构实质上就是用户在使用 FPGA 之前,要事先定义 FPGA 中的细粒度单元的功能以及重构这些单元之间的互联关系. 最简单的基于指令流计算的体系结构,就是这种可重构的 FPGA 电路,也就是现在称之为变件(morphware)的电路. 它的结构一旦改变好之后,则只需要控制该电路的输入数据的来源与输出结果的目的地,就能完成重构所事先赋予 FPGA 的计算功能,它也是按数据流计算的方式工作的. 因此,指令流计算的效率接近于数据流计算的效率,但是,指令流计算的可重构性则带来了类似指令流计算那样的灵活性.

如图 1 中所示,20 世纪 90 年代,出现了粗粒度的 FPGA 芯片,实际上,就是一种基于总线的 SoC 动态可重构的硬件实现形式,重构是通过现在人们所说的构件(configuware)实现的. 动态可重构就是在使用过程中,可以通过构件定义 FPGA 的功能.

大约在 1999 年,人们又开始了基于网络的 SoC(又称为 NoC(Network on Chip))的动态可重构硬件实现形式的研究<sup>[5]</sup>.

现在的基于构令流计算的体系结构是采用粗粒度的变件(morphware)动态实现的:它可以是一个可重构的处理部件 rPU(reconfigurable Processing Unit),也可以是一个 rPU 阵列,如图 3 中所示. 构令流来自存放在构令存储器 CM(Configuration Memory)中的构件源,通过构令管理器(configuration manager)控制执行. 它是用来改变 rPU 阵列的硬件结构以及形成进出 rPU 阵列的数据流互连关系的<sup>[1]</sup>,也是按照数据流计算的方式工作的:构令执行之后,只要该构令的数据流还在进行,硬件结构以及互连关系是不变化的.

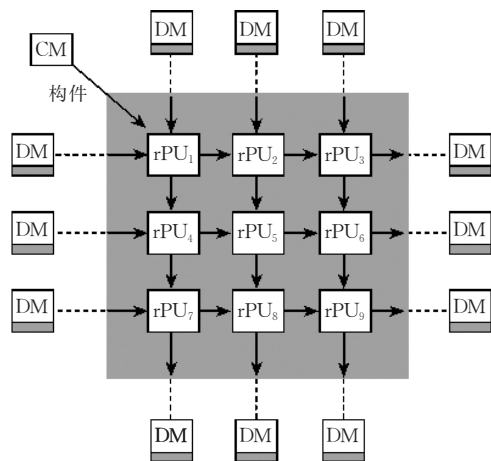


图 3 基于构令流计算的体系结构示意图

随着芯片集成度的提高,基于构令流计算的体系结构已成为当今的研究热点. PACP 公司研制的名为 XPP(eXtreme Processing Platform)的动态可重构的 IP 核,已用于一种可重构的 SoC 芯片中<sup>①</sup>. 该动态可重构的 IP 核由 64 个可重构的 24 位的 ALU 模块、16 个可重构的 RAM 模块以及 8 个可重构的 I/O 模块组成,在构令管理器的控制下工作. 根据应用的计算要求不同,在 SoC 芯片上可以包含一个或多个动态可重构 XPP 这样的 IP 核.

### 3 体系结构的分类模型

众所周知,由 Flynn 提出的基于指令流计算的体系结构的分类方案,共用了 6 个分类概念:IM (Instruction Memory), DM (Data Memory), CU (Control Unit), PU (Processing Unit), IS (Instruction Stream) 与 DS (Data Stream); 只有一个分类层

次,共有 4 种体系结构类别: SISD 体系结构、SIMD 体系结构、MISD 体系结构与 MIMD 体系结构. Flynn 的体系结构分类方案是在 1966 年提出来的<sup>[6]</sup>,在当时的硬件水平条件下,提出的指令流与数据流的概念以及把指令流和指令控制下的数据流结合在一起,对计算机体系结构的分类,对计算机体系结构的发展起到了促进作用. 由于 Flynn 的分类方案非常简洁,后来便出现了一些更细化的体系结构分类方案. 例如,1989 年 Dasgupta 提出了一个多层次的分类方案<sup>[7]</sup>,可以详细地刻画现有机器的内体系结构. 由于其分类字符是以计算机内体系结构中的典型部件结构为依据的,因此,如果再出现新的典型部件结构,则这种分类方案就无法描述了. 这些方案都没有像 Flynn 提出的只以典型逻辑概念为基础的体系结构分类方案那样,受到广泛的关注和引用.

芯片集成度与 CMOS 掩膜版价格的指数上升,促进了可重用的芯片实现形式的迅速发展. 以 IP 核为基础的设计方法学以及用来完成功能与体系结构协同设计的动态可重构平台的使用,有效地提高了 SoC 设计的效率,增加了 SoC 功能的可调整性,使更多可重构的硬件集成到 SoC 的体系结构中,除了基于指令流计算的体系结构之外,还形成了基于数据流计算与构令流计算的体系结构. 在实际计算机的体系结构中,除了 Flynn 所提出的 IM, DM, CU, PU, IS 与 DS 等 6 个分类概念之外,还出现了构令存储器 CM(Configuration Memory)、可重构的处理部件 rPU(reconfigurable Processing Unit) 以及构令流 CS(Configuration Stream) 等 3 个新的概念. 针对这种情况,本文提出了一种新的计算机体系结构分类模型及其分类表示方法.

#### 3.1 一种新的体系结构分类模型

这里提出的计算机体系结构的分类模型,是以指令流、数据流与构令流的概念为基础的,是以分别称为 I (Instruction stream) 体系结构、D (Data stream) 体系结构和 C (Configuration stream) 体系结构为坐标轴来建立的(如图 4 中所示),在此称为 IDC 体系结构分类模型. 这 3 个正交的坐标轴是用来定义体系结构所在的空间的. I 轴只表示基于指令流计算的体系结构, D 轴只表示基于数据流计算的体系结构, C 轴只表示基于构令流计算的体系结构. 每个轴的箭头方向表示该类体系结构的实现程度.

① PACT Technologies. The XPP White Paper Release 2.1. <http://www.pactxpp.com>



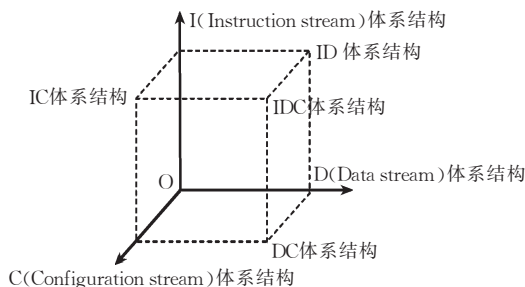


图4 IDC体系结构分类模型

在 IDC 体系结构分类模型中,两个轴定义的平面用来表示具有两类流计算的体系结构. ID 平面用来表示具有指令流计算与数据流计算两种硬件实现形式的计算机体系结构. IC 平面用来表示具有指令流计算和构令流计算的体系结构. DC 平面用来表示具有数据流计算和构令流计算的体系结构.

实际上更多的体系结构是分布在 IDC 体系结构分类模型的三维空间中的,属于具有指令流计算、数据流计算与构令流计算的计算机体系结构.

### 3.2 体系结构的分类表示

如前所述,在图4所示的体系结构分类模型中,体系结构可以看做是属于一维的坐标轴上的,二维的坐标平面上的以及三维的坐标空间上的.为了使这3种情形以及由多种基本体系结构组合而成的复杂体系结构的分类表示清楚起见,体系结构的分类采用了众所周知的 e-mail 地址的类似表示方法.

#### 3.2.1 坐标轴上的体系结构分类表示

对在坐标轴 I 上表示的基于指令流计算的体系结构来说,采用 Flynn 分类的体系结构表示方法,共有 SISD, SIMD, MISD 与 MIMD 4 种基本体系结构分类表示.但是,现代的计算机并不都是单一的 SISD, SIMD, MISD 或 MIMD 的体系结构,而有些是这些体系结构的组合.例如,由一个 SISD 的 RISC 处理器作为宿主机,与一个 SIMD 的 PE 阵列作为协处理器组成的嵌入式图像处理的 SoC 芯片,就是两种体系结构的组合,用 SISD · SIMD 表示就更为准确一些.这种组合表示是需要的,因为在现代的微处理器中,为了更好地支持多媒体应用,也开始增加一些 SIMD 指令,MMX 指令就是一个众所周知的例子.这种微处理器所构成的计算机,其体系结构用 SISD · SIMD 组合表示就更准确些.这样一来,在坐标轴 I 上就可以有  $15(C_1^1 + C_4^2 + C_4^3 + C_1^4)$  种体系结构的类别表示.

对在坐标轴 D 上表示的基于数据流计算的体系结构来说,其数据流是从流件源来的,如图2中所

示,是没有操作流而只有数据流的.所以,只有单数据流与多数据流两种基本情形,分别用 SD (Single Data stream architecture) 与 MD (Multiple Data stream architecture) 表示.例如,图2中的脉动阵列体系结构,就可以用 MD 表示.故在坐标轴 D 上只有  $3(C_2^1 + C_2^2)$  种体系结构类别.

对在坐标轴 C 上表示的基于构令流计算的体系结构来说,也是按数据流计算的方式工作的,其数据流是由构令流决定的,换句话说,构令不仅用来改变并决定硬件的逻辑结构,而且用来改变并决定数据的流动路径.因此,与 I 体系结构类似,也有 4 种基本分类表示: SCSD (Single Configuration Single Data stream), SCMD (Single Configuration Multiple Data stream), MCSD (Multiple Configuration Single Data stream) 和 MCMD (Multiple Configuration Multiple Data stream). 例如,图3中的 C 体系结构就可以用 SCMD 表示.因此,利用这 4 种基本表示,总共可以有 15 种体系结构类别.

#### 3.2.2 坐标平面上的体系结构分类表示

对在坐标平面 ID 上表示的基于指令流计算与数据流计算的体系结构来说,则可以用坐标轴 I 上与坐标轴 D 上的体系结构表示方法的组合来表示,故有  $45(15 \times 3)$  种体系结构类别(不包括坐标轴上的情形,下同).例如,由一个 SISD 的 RISC 处理器作为宿主机,与一个如图2中所示的基于数据流的脉动阵列作为协处理器组成的嵌入式 SoC 芯片,就是这两种体系结构的组合,可以用 SISD @ MD 表示.实际上,在指令流计算机的输入输出接口逻辑中,数据流一般是在输入输出指令控制下,也就是程序计数器的控制下流动的,但为了提高计算的并行性,也出现了在 DMA 或 FIFO 等部件控制下,也就是在数据计数器控制下流动、按数据流计算的情形.

对在坐标平面 IC 上表示的基于指令流计算与构令流计算的体系结构来说,则可以用坐标轴 I 上与坐标轴 C 上的体系结构表示方法的组合来表示,故有  $225(15 \times 15)$  种体系结构类别.例如,由一个 SISD 的 RISC 处理器作为宿主机,与一个如图3中所示的基于构令流的 rPU 阵列作为协处理器组成的嵌入式 SoC 芯片,就是这两种体系结构的组合,可以用 SISD @ SCMD 表示.实际上,动态可重构的概念现在已经用到微处理器设计中,将处理器设计成由指令流与构令流两种流组成<sup>[8]</sup>,是属于 IC 平面上的体系结构.

对在坐标平面 DC 上表示的基于数据流计算与

构令流计算的体系结构来说,则可以用坐标轴 D 上与坐标轴 C 上的体系结构表示方法的组合来表示,故有  $45(3 \times 15)$  种体系结构类别. 例如,由一个可重构的 RISC 处理器作为宿主机,与一个图 2 中的 Systolic Array 作为协处理器所构成的 SoC 芯片,就是坐标平面 DC 上的一种体系结构,可以用 SCSD@MD 表示.

### 3.2.3 坐标空间中的体系结构表示

对在坐标空间 IDC 中表示的基于指令流、数据流与构令流的体系结构来说,则可以用坐标轴 I 上、坐标轴 D 上与坐标轴 C 上的体系结构表示方法的组合来表示,可以有  $675(15 \times 3 \times 15)$  种体系结构类别(不包括坐标轴上的与坐标平面上的情形). 例如,嵌入式 SoC 可以由一个 SISD 的 RISC 处理器作为宿主机,与一个如图 2 中所示的脉动阵列作为协处理器,还有一个如图 3 中所示的 rPU 阵列作为协处理器组成的计算机体系结构,就是一个坐标空间 IDC 中的体系结构,可以用 SISD@MD@SCMD 表示.

## 4 讨 论

随着芯片集成度的上升与应用领域的扩展,如图 1 所示,1977~1987 年是通过指令编程的 IC 标准化实现形式的 10 年;1987~1997 年是逻辑芯片商业化的 10 年,SoC 主要是面向应用领域设计的;现在微电子技术正在从 Mega Age 向 Giga Age 过渡,1997~2007 年将是通过构令重构的 IC 标准化实现形式的 10 年. 从而使组成现代计算机的芯片可分为 4 类:除了存储器芯片与处理器芯片外,还有逻辑芯片与模拟芯片. 对于所有 3 种体系结构来说,都是要用到存储、通信与处理的功能的,其中,存储功能是相同的,以用来存储数据以及软件、流件或构件. 差异主要体现在通信功能与处理功能的体系结构上.

下面将进一步讨论三维坐标分类法的必要性、合理性与完备性.

### 4.1 三维坐标分类法的必要性

基于指令流计算的体系结构,因为指令一般是由操作码和数据地址码两部分组成的,软件(software)就是由这些指令组成的. 因此,基于指令流计算的体系结构是最基本的、最灵活的体系结构,但不是最高效的体系结构. 其体系结构的特点主要体现在处理器芯片的通用计算方式以及指令(也就是程序计数器)控制下的存储器与处理器之间的数据通

信功能上. 有 CISC, RISC 与 DSP 3 种处理器. 模拟芯片完成嵌入式计算机与物理世界的接口功能,其数据流可以在指令(也就是程序计数器)控制下流动,按指令流计算的体系结构设计;也可以在部件(也就是数据计数器)控制下流动,按数据流计算的体系结构设计.

基于数据流计算的体系结构则正好相反,因为流件(flowware)可以看作是只有访问地址与地址操作,而没有数据操作的命令组成的. 例如,从图 2 中可以看出,数据是直接通过脉动阵列来完成计算的,所以它是最快的,但是,是面向单个算法设计的,最不灵活. 其体系结构的特点主要体现在逻辑芯片的专用计算方式以及部件(也就是数据计数器)控制下的存储器与逻辑芯片之间的数据通信功能上. 不仅有与处理器差别很大的 ASIC 电路,而且有只完成一种固定功能的类似处理器的电路,例如,由 ZarLink 公司研制的完成 FFT 计算的 PDSP 芯片. 这些逻辑芯片所完成的功能,都比处理器芯片功能的粒度大. 因为处理器受指令字长的限制,每条指令一般都只能完成加、减、乘、除中的一种运算(只有乘加指令是例外),而在上述 FFT 逻辑芯片中,完成的是固定的 1024 点或 4096 点的复数蝶式运算功能,每个复数蝶式运算包括 4 次乘法、3 次加法与 3 次减法. 因此,数据流计算有比指令流计算粒度大的高效性,但没有指令流计算粒度小的灵活性,彼此是不能取代的.

前两种体系结构都是基于不可重构的硬件实现形式的;也就是说,硬件(芯片)的结构是用户无法改变的. 而基于构令流计算的体系结构,则主要是让用户通过构令(configuration)来改变(重构)芯片的结构. 其体系结构的特点主要体现在可重构芯片所提供的灵活的计算方式与通信方式上. 所以,基于构令流计算的体系结构具有类似指令流计算体系结构的灵活性;一旦硬件的操作功能与数据的流动路径形成之后,它就是按数据流计算的方式工作的,所以,它又具有接近基于数据流计算的体系结构的高效性,是指令流计算与数据流计算所不能做到的. 当构令流计算的体系结构是以 IP 核为基础的 SoC 芯片时,IP 核的结构以及 IP 核之间的互连结构都可以重构,按构令流计算的方式工作. 特别是在高性能数字信号处理应用领域中,可重构的体系结构已经成了研究的热点.

### 4.2 三维坐标分类法的合理性

正是以逻辑概念而不是以部件结构为依据使

Flynn 分类具有简洁性、实用性和稳定性. 这种按逻辑概念分类方法的合理性就在于逻辑概念的不同并不等于要求物理实现上的不同. 例如, 指令存储器与数据存储器只是一种逻辑概念的区分, 而物理实现上则可能就是不分开的; 又例如, 指令流与数据流也只是一种逻辑概念的区分, 实际上这两种信息流可能是共享总线的; 再例如, 指令流计算、数据流计算与构令流计算虽是 3 个不同的概念, 而物理实现上可以同时设计到一个处理器中<sup>[9]</sup>. 所以, 这里就顺理成章地采用了以逻辑概念为依据的计算机体系结构的分类方法.

从解释能力上来说, Flynn 分类只用了指令流与在指令控制下的数据流的逻辑概念, 因而只能描述算法时间映射(temporal mapping)到软件的基于指令流的并行计算. 但是, 由于微电子技术按照摩尔预言的速度发展, 现代并行计算机的体系结构已经超出了指令流与指令控制下的数据流相结合的概念以及 Flynn 分类的范围. 需要采用新出现的逻辑概念, 扩充计算机体系结构的分类范围. 如前所述, 一是需要采用现在新出现的基于数据流计算的流件(flowware)概念, 以描述算法空间映射(spatial mapping)到流件的基于数据流的并行计算; 二是由于 20 世纪 80 年代后期开始的可重构计算技术, 除了指令流(instruction stream)计算与数据流(data stream)计算之外, 现在又出现了构令流(configuration stream)计算的概念. 在此基础上, 除了软件(software)与流件(flowware)的概念之外, 还出现了构件(configware)的概念<sup>[3]</sup>. 需要采用这些新出现的逻辑概念, 以描述算法空间映射(spatial mapping)到构件的基于构令流的并行计算.

分类表示法的合理性在于: (1) 分类表示符是在指令流计算概念的 4 个众所周知的分类表示符 SISD, SIMD, MISD 与 MIMD 的基础上, 类似地提出了数据流计算概念的 2 个分类表示符 SD 与 MD 以及构令流计算概念的 4 个分类表示符 SCSD, SCMD, MCSD 与 MCMD 而形成的, 通过组合表示法, 可以表示多达  $1023(33+315+675)$  种体系结构类别, 能够对现代计算机的体系结构, 特别是对以 SoC 芯片为基础的计算机体系结构进行更清楚的分类描述. (2) 分类表示采用了众所周知的 e-mail 地址的类似构成方法, 按照上述 10 个分类表示符, 在一维的体系结构中, 最短的体系结构表示将是只用到 1 个分类表示符的表示, 最长的体系结构表示则将是用到 4 个分类表示符的表示; 在二维的体系结构中,

最短的体系结构表示将是只用到 2 个分类表示符的表示, 最长的将是用到 8 个分类表示符的表示; 在三维的体系结构中, 最短的将是只用到 3 个分类表示符的表示, 最长的则是将用到 10 个分类表示符的表示. 对于同一应用任务的多种体系结构设计, 分类表示符用得越多, 则说明体系结构的方案越复杂; 而分类表示符用得越少的体系结构方案, 则从设计、测试与应用的角度来说, 可能是最可取的. 因此, 三维坐标的分类表示方法不仅简洁实用, 而且有助于判断体系结构方案的可取性.

#### 4.3 三维坐标分类法的完备性

图 4 所示的计算机体系结构的分类方案是否完备可以从两个方面来说明: (1) 基于指令流计算、数据流计算与构令流计算的必要性与充分性. 从上面的分析可以看出, 除了指令流计算之外, 数据流计算与构令流计算是完全必要的. 从硬件的操作功能与数据的流动路径的组合关系来看, 除了指令流计算之外, 再加上数据流计算与构令流计算, 也是充分的. (2) 计算智能(computational intelligence), 也就是软计算(soft-computation)有关的仿生硬件, 会不会带来新的计算机体系结构? 十多年前产生的仿生硬件的研究, 也就是计算智能与可重构硬件的结合. 仿生硬件(Bioinspired Hardware /Biologically Inspired Hardware), 就是在电路的设计或运行中, 采用了生物学原理<sup>[10]</sup>. 因此, 采用可重构硬件的概念是必要的. 由于硬件只有可重构与不可重构两种情形, 因此, 已经包含这两种情形的图 4 所示的计算机体系结构分类模型, 也是充分的.

计算机的体系结构的分类模型只是用来对计算机体系结构进行分类的, 而设计计算机时, 所需要的是计算机的体系结构模型, 又叫做计算机的技术模型. 众所周知, 基于指令流计算的体系结构模型有 ISA 模型, 那么, 基于数据流计算与基于构令流计算的体系结构, 是否分别会有相应的 DSA 与 CSA 模型? 这将另文进行讨论.

#### 参 考 文 献

- 1 Manners D., Makimoto T.. Living with the Chip. London: Chapman & Hall, 1995
- 2 Kung H. T.. Why systolic architecture? Computer, 1987, 15: 37~46
- 3 Hartenstein R.. Data-stream-based computing: Models and architectural resources. Kaiserslautern University of Technology, Germany. <http://hartenstein.de>

4

Trelearen P. C. *et al.*. Data-driven and demand-driven computer architecture. *ACM Computing Surveys*, 1982, 14(1): 93~143

5

Jantsch A. , Tenhunen H. . *Networks on Chip*. Boston: Kluwer Academic Publishers, 2003

6

Flynn M. J. . Very high speed computing systems. *Proceedings of the IEEE*, 1966, 54: 1901~1909

7

Dasgupta S. . *Computer Architecture——A Modern Synthesis*. New York: John Wiley & Sons, 1989

8

Heysters P. M. , Smit J. , Smit G. J. M. , Havinga P. J. M. . Exploring energy-efficient reconfigurable architectures for DSP algorithms. In: *Proceedings of the PROGRESS 2000 Workshop*, 2000, 37~46

9

Enzler R. . Architectural trade-offs in dynamically reconfigurable processors. Swiss Federal Institute of Technology (ETH), Zurich: Technology Report 15423, 2004. <http://www.ife.ee.ethz.ch/~enzler/pub/phdthesis04.html>

10

Deprettere F. eds. . *Embedded Processor Design Challenges*. Berlin Heidelberg: Springer-Verlag, 2002



**SHEN Xu-Bang**, born in 1933, member of Chinese Academy of Sciences, Ph. D. supervisor. He has long been engaged in the research of embedded computers and their chips.

**ZHANG Fa-Cun**, born in 1966, associate professor. His

research interests focus on embedded computer architecture.

**FENG Guo-Chen**, born in 1975, Ph. D. candidate. His research interests focus on embedded computer architecture.

**CHE De-Liang**, born in 1975, Ph. D. candidate. His research interests focus on embedded computer architecture.

**WANG Gang**, born in 1973, Ph. D. candidate. His research interests focus on embedded computer architecture.

Background

With the development of reconfigurable computing techniques, for computer architectures, there are not only traditional instruction stream computation based architectures and data stream computation based architectures, but also configurable stream computation based architectures. The reconfigurable hardware can not only speed up the program execution, but also adjust the computation platforms. The authors are engaged in the research of function and architecture co-

design of SoC. In order to give a clear classifiable description of modern computer architectures, it is necessary to have a new classification model of computer architectures. On the basis of the concepts of Instruction Stream computation, Data Stream computation and Configuration Stream computation, a new classification model of computer architectures is proposed in this paper.