

动态计算机核心 PMC362 成功设计

姜咏江

对外经济贸易大学信息学院, 北京 (100013)

E-mail: accsys@126.com

摘要: PMC362 是程序能自动调度执行的动态计算机核。这种结构将各类多处理器设计成对指令无痕的, 将程序放置在环境条件完全具备的一级 Cache 级存储单位 MU 中, 通过 MU 的各种连接标志, 实现与不同类型处理器连接来完成程序不同的任务执行。这种设计对任意细划的处理器类型, 能够做到程序立即转移执行, 从而充分调动全体处理器工作。

关键词: CMP, 程序存储单位 (MU), 超并行程序执行, 动态计算机, 无痕处理器 (PU)

中图分类号: TP301; TP302; TP303

1 引言

用操作系统来管理程序执行一直是计算机系统的主流设计思想^[1], 这种思想不仅全面影响着软件设计, 而且也影响着计算机系统的硬件设计。

操作系统将程序看成“僵死”的东西, 因而程序是否能够到处理器上运行, 都完全需要操作系统来管理控制。由于操作系统自身也是程序, 为了实现管理控制, 需要特别为其设置高级别的运行权力, 因而操作系统会抢占处理器, 占据大量的机时。

在多处理器计算机设计中, 必然会产生处理器类型进一步细化问题, 例如, CPU、GPU、DSP 等, 这意味着程序在不同类型的处理器之间转移执行, 已经成为了不可避免的研究课题。这种情况下, 给操作系统怎样的地位是设计者要深刻思考的问题。如果仍然让操作系统来管理各处理器运行, 那么众多的处理器能否同时充分工作? 操作系统自身怎样运行? 它在哪个处理器上运行? 操作系统在不同的处理器上运行如何实现内部通信? 程序是否还必须以进程的方式运行? 会不会增加“死锁”的机率? 操作系统全面的软件干预是否会影响计算机的效率?

本文作者较早时间就认识到了操作系统在对多处理器计算机的不适应性, 于 2006 年设计了不用操作系统来管理程序在处理器上执行的计算机核心系统 PMC362^[2]。在该计算机系统中, 处理器被细分为多个种类 (参考图 1), 存储器也被分成了只与程序单独相关的存储块, 与程序计数器、标志寄存器和数据寄存器等, 形成程序运行的环境, 使其中的程序能够主动自己选择处理器执行, 通过硬件逻辑解决各种冲突, 既可排除死锁发生的可能性, 又可使全体处理器能够充分高效运行。

2 动态计算机 PMC362

全体程序都以主动的方式动态连接处理器运行的计算机, 我们称之为动态计算机。动态计算机秉承了作者较早的内存分块思想^[3], PMC362 就是这样一款设计成功的核心计算机。

2.1 PMC362 的结构

动态计算机 PMC362 的一般结构如图 1 所示。PU (Performance Unit) 是运算处理器, 叫执行单位, 有 3 个。CHL (Channel) 是 I/O 处理器, 有 2 个。T 是程序通信处理器, 也有 2 个。MU (Memory Unit) 是程序存储设备, 叫程序存储单位, 其中包含程序运行的环境, 共有 6 个。DV 是外部设备, 可有若干个。外部存储器是大容量存储设备, 放置任务程序和数据文件, 这种存储设备可以有多个。各设备之间采用关系型总线连接, 用逻辑开关控制总

线与设备的通断。

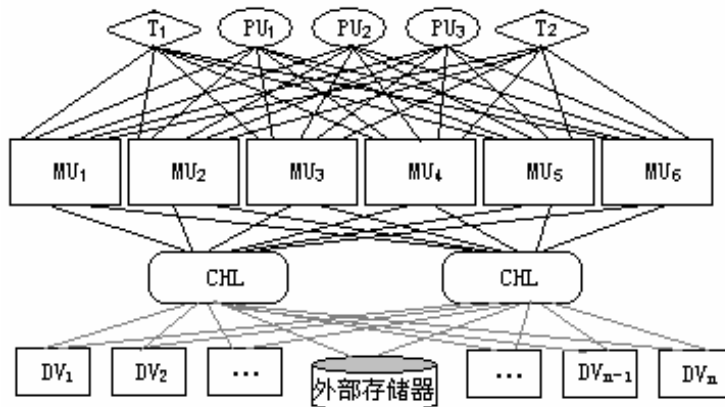


图 1 PMC362 的结构

在动态计算机核中，处理器 PU、T、CHL 都只负责指令执行，它们既不保留指令执行的结果，也不具有指令执行的初始条件。处理器都具有忙闲标志（T、CHL 的标志是双向的，PU 是单向的），它们靠这些标志来接受或拒绝 MU 或其他设备的连接。

存储单位 MU 中不仅包含程序存储器和数据存储器，而且包含程序运行的全部环境。也就是说，每个 MU 中都包含程序计数器、状态寄存器、数据寄存器等一系列决定程序执行的条件。如果对于单一处理器采用同时多线程 SMT^[4]方式，那么如图 2 所示，要对每个硬件线程设置运行环境。MU 中的旗帜表示只是负责与处理器联络，而内部的线程是在 MU 保持与处理器连接的情况下，完成 SMT 方式。

2.2 PMC362 主要部件功能特色

动态计算机 PMC362 是 16 位的，主要部件设计上有如下的特色：

(1) 处理器对指令无痕。为了让程序根据需求能快速地在不同处理器间变换，各处理器既不具有指令执行的初始信息，也不保留指令执行的结果，用忙闲标志实现与 MU 的动态连接。

(2) 存储单位完备。MU 既包括放置程序和数据的数据 SRAM 存储器，又包括程序执行的环境设备，还包括调度关联设备，能够在程序暂停执行的时候，通过各种标志信号实现自动调度。存储器采用哈弗结构，方便上下文的数据传递。

(3) 可控关系总线。设备间一律使用可控关系总线连接，保持连接需求的一致性，全体总线的通断施行标志信号控制。

(4) 标志信号控制避免发生死锁。由于 MU 的完备性和存储程序的单道性，各程序之

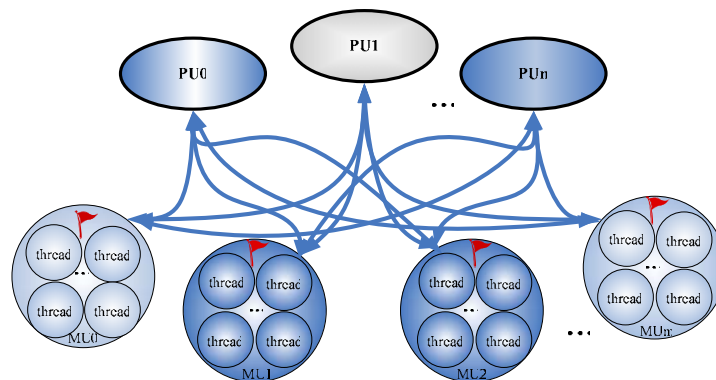


图 2 有线程的动态核连接结构

间不发生数据相关，共享设备加锁信号互斥使用，程序暂停一律释放共享资源。

(5) 并发并行同在。多处理器同时运行，体现了程序的同时并行执行状态，每个处理器上程序执行施行时间段控制，让等待执行的程序都有机会执行，保证任务的均衡完成，这是程序并发执行设计。

(6) 异步通信。相互关联装填到不同 MU 的程序，一律通过连接 T 处理器处理通信，受信暂停的 MU 能自动间隔一定时间连接 T 访问邮箱。

3 简单的程序设计

由于动态计算机把多程序同时执行放在了首位，也就是将程序级或线程级并行^[5]放在首位，并不刻意追求指令级并行^[5]。因而在一个程序内部，研究如何并行执行的设计问题就变得不重要了，况且这个问题可以使用硬线程的方法，在单个处理器的运行线程中加以解决。动态计算机能让简单传统方法设计的多个程序，同时在不同的处理器上执行。程序只要指出到什么类型的处理器执行，并写出需要执行的指令就可以实现。

例如，下面这段程序用两条指令就实现了从 PU 到 CHL，再从 CHL 到 PU 的转换，很方便地在不同处理器上完成了内存与外设之间的数据操作。

```
...
ADD  0023  ; 将累加器与 h0023 单元内容相加
STR  0024  ; 将累加器内容送 h0024 单元
CHL                ; 要求与输入输出处理器 CHL 连接
INCH 02 0025 0020 ; 将 02 设备的 h0020 地址数据读到 h0025 内存单元
PU                ; 要求与 PU 处理器连接
LDA  0024  ; 将 h0024 单元内容放入累加器
ADD  0025  ; 加上 0025 单元内容
STR  0025  ; 结果送回 0025 单元
...
```

指令 CHL 和 PU 的作用仅仅是设立连接请求标志，程序执行这些指令后都会与原连接的处理器断开，如果需要连接的处理器有空闲的，就会立即连接执行程序，不然会等待。所有等待执行的程序都会按照硬件优先顺序进行连接，不会发生连接冲突。

4 程序同时执行的自动调度

使用操作系统实现程序调度是十分复杂的过程。动态计算机程序调度简单，集中体现在程序能够“自动”在多个处理机之间调度，而且不需要任何所谓的系统程序干预。这种调度方式能最大限度地提高计算机的工作效率。动态计算机中，各种设备都设有工作或请求连接的标志线，整机通过高层的控制逻辑，就可以让 MU 中的程序自动调度运行起来。

系统初始，除了固化有程序的 MU 之外，所有的 MU 都将请求连接标志指向 CHL，如果没有外部任务程序请求执行，CHL 会拒绝与这些 MU 连接。外设与 MU 的连接也是动态的，如果外设中的任务程序请求执行，空闲的 CHL 将与之连接，并且立即与空闲的 MU 连接，将任务程序传输给 MU，同时将 MU 中的程序上下文寄存器填写好，以备后继程序段能够准确装填，连续执行。装填好程序段的 MU 会自动与 CHL 断开，并将连接申请标志将指向 PU，进入程序的动态执行过程。

MU 之间会依据一定的优先顺序与各处理器连接，各类处理器也有连接优先顺序，这样

可以避免连接冲突。装载了程序段的 MU 会依据程序执行需要, 自动寻找不同类型的处理器连接或断开, 在转换或完成程序段执行时, 通过指令填写好应该连接的处理器标志, 以达到选择或周而复始执行程序的过程。

MU 内的存储器容量是有限的, 超长的程序需要分段进入 MU 去执行, 为不使任务错位, 保证上下文共享存储在数据存储器中的数据资源, 编译程序要在每个装填段的末尾, 添加设置连接 CHL 标志指令, 并用保存在 MU 中的程序段编号控制次序, 在全部程序段执行完成后, 清空程序上下文寄存器, 申请指向 CHL, 以便新的程序能够进入 MU 执行。

由于处理器无痕的特点, 在同类型处理器中, MU 与任何一个处理器连接都可以执行程序, 这样可以充分发挥全体处理器的作用, 最大限度地实现程序同时执行。动态核同时执行的多个程序, 不必刻意去编写专门的并行程序, 普通传统编程方法得到的程序, 都可以投入程序并行执行状态, 因而称这种程序并行执行为超并行程序执行。

相互关联的多个程序装入不同的 MU 中执行, 如果它们之间需要数据通信, 可以通过指令将地址信息填入 MU 的通信地址寄存器保存, 然后转向与通信处理器 T 连接。发信的 MU 在与 T 连接后, 通过 T 将发送地址和数据送到邮箱。接收数据的 MU 再通过 T 查找信件, 如果邮箱没有需要的信件, 相隔一定时间, 再次通过 T 查找。

邮箱、外部设备和网络端口设备都可以有多个, 不执行程序任务时, 它们与处理器总是断开的, 只有程序任务需要连接这些处理器时, 才会根据需要连接, 并同时设置好忙标志, 以防连接冲突。输入输出、通信等过程, 一般要同时实现三种连接: MU 与处理器连接; MU 与使用的其他设备连接; 处理器与这个其他设备连接。动态核结构要求, 只要有一个连接不成立, 立即断开其他两方的连接, 这样可以保证不发生“死锁”。

5 PMC362 程序并行执行仿真

这里借助于 Altera 公司的 EDA 软件 Quartus II^[6]的波形仿真器, 将 6 个程序在 PMC362 上执行的动态过程仿真结果展示出来, 借此说明动态计算机程序自动调度执行的优越性。

图 2~图 5 是 6 个程序 a、b、c、d、e、f 动态并行执行的时序仿真截图。图中 outa~outf 是 6 个程序的输出, chl0、chl1 是 I/O 处理器, pu0、pu1、pu2 是运算处理器。处理器连接优先级为 chl0、chl1、pu0、pu1、pu2, 而程序的优先顺序是 a、b、c、d、e、f。图中 6 位二进制数从低向高分别指示 a、b、c、d、e、f, 为 1 的位置表示该程序正在左面标注的处理器上运行。例如, pu2 的值是 001000, 表示程序 d 正在处理器 pu2 上执行。

这个仿真的时钟周期是 80ns。从图 2 可以看到, 在系统启动后 40ns 处 a 程序就在 pu0 上执行了, 并且接下来的 2 个时钟周期, b 程序到 pu1 上执行, c 程序在 pu2 处理器上执行了。约到 2.88us 的时候, a 程序断开与 pu0 的连接, 自动转到空闲的输入输出处理器 chl0 上执行, 而且就在这个时钟周期内, d 程序立即与刚与 a 程序断开的处理器 pu0 连接执行了。

由于没有空闲的 pu 处理器, 因而 e 程序和 f 程序一直在等待, 因为它们的连接优先级低, 不能抢在其他程序前面连接执行。

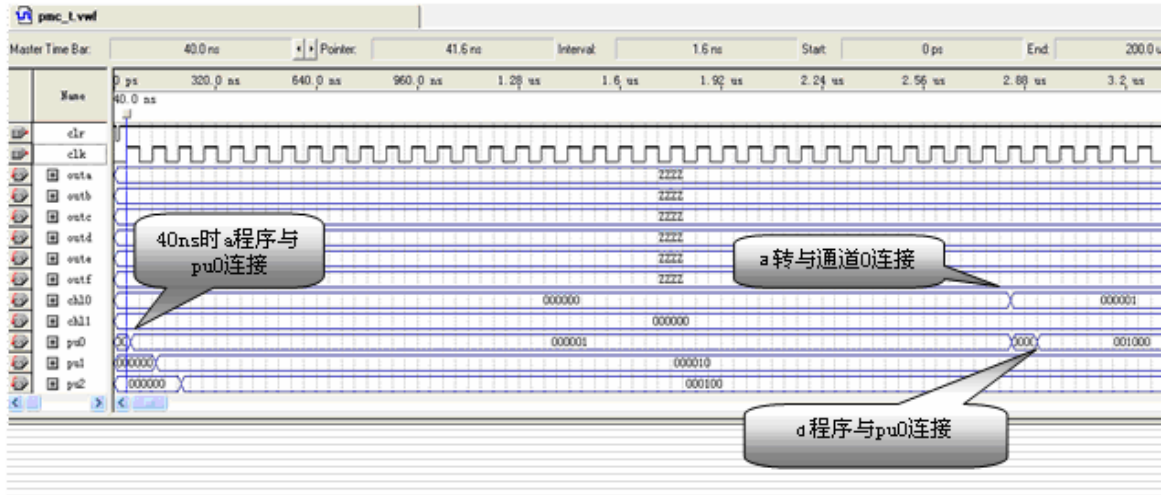


图 2 初始执行与 a 程序转换

从图 3可见，约到 4.40us 的时候，b 程序需要输入输出，断开了与 pu1 的连接，而与空闲的 ch11 连接执行，等待的 e 程序优先级高于 f 程序，所以它马上与 pu1 连接，进入程序执行。此时 5 个处理器都在工作，形成了 5 个程序并行执行的局面。

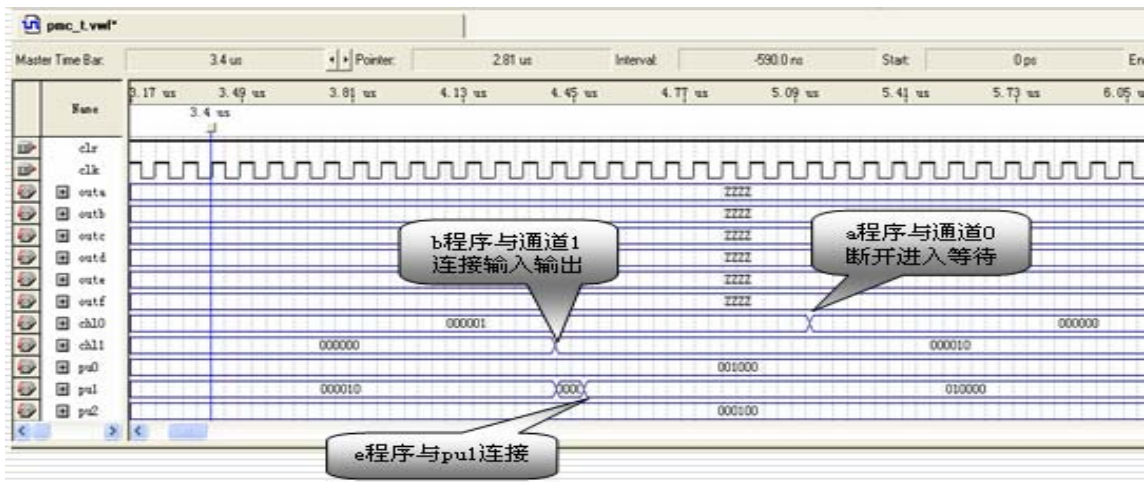


图 3 b 程序与 a 程序的转换

约在 5.10us 的时候，a 程序完成输入输出，与 pu0 断开，虽然又需要与 pu 连接，但由于没有空闲的 pu 处理器，a 程序也只好进入了等待状态。这时没有程序要求与 I/O 处理器连接，ch10 暂时空闲。

从图 4可以看到，约到 6.70us 的时候，b 程序与 ch11 处理器断开，由于没有空闲的 pu，也只得进入等待状态。约 7.10us 的时候，d 程序需要连接 ch1 处理器，于是依据优先级，它与 ch10 连接，而空闲出来的 pu0 处理器又被优先级高度的 a 程序抢先连接，而优先级低的 f 程序还得与 b 程序一起等待，此时 ch11 处理器空闲。

约到 9.60us 的时候，d 程序断开了与 ch10 的连接，要求与 pu 处理器连接，但没有空闲的 pu 处理器，所以 d 程序也进入了等待。此时处理器 ch10、ch11 都空闲，而 pu 处理器都在工作，并行执行的程序只有 3 个。

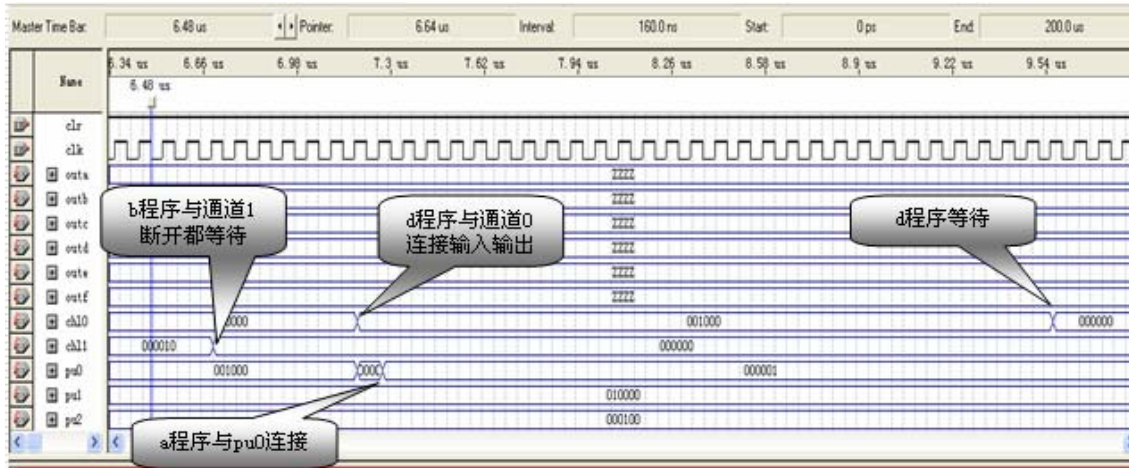


图 4 b、d、a 程序的转换

从图 5 可以看到，约在 10.10us 的时候，a 程序完成结果输出，并在 10.50us 的时候结束程序，断开了与 pu0 的连接。等待的程序中 b 程序优先级高，因而它立即与 pu0 连接了，d 程序和 f 程序等待。

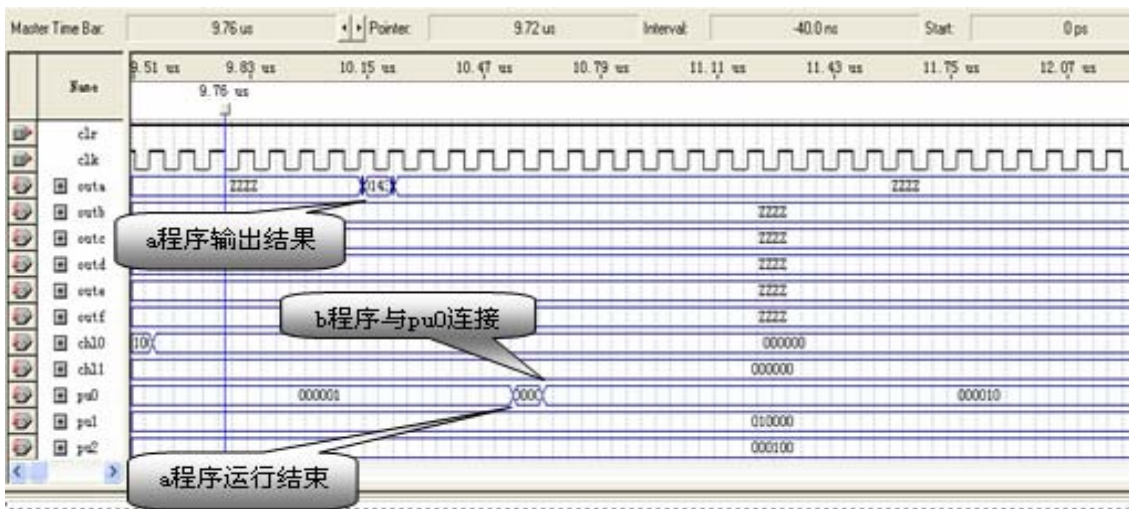


图 5 a 程序输出与结束

从 PMC362 的部分仿真截图可以看到动态计算机程序调度执行的自动性，这种快捷的程序调度不用其他程序干涉，这不仅节约了转移的时间，而且能够实现各种类型处理器充分的工作。

6 下载到开发板上运行检验

动态计算机核 PMC362 于 2006 年设计成功，除了进行设计仿真之外，还下载 Altera 公司的 DE1 电子电路开发板上进行了实际运行测试。作者用本机指令系统编写了操作管理程序，输出延时子程序等，还用 Verilog HDL 硬件设计语言编写了各设备驱动程序。经过开发板上的下载运行测试，运行稳定，效果良好。

7 结论

PMC362 动态计算机核心的设计成功，说明了个人设计计算机核心器件的可行性，软硬件一体化设计的实用性^[7]。动态计算机为传统程序实现同时并行执行创造了实例。这不仅说

明了不用操作系统管理多程序同时执行的现实性，而且也对片上多处理器 CMP 的设计，提供了先进的设计方法。动态计算机核心的 PU-MU-CHL 结构，不会出现“存储墙”等^[8]问题，还会有效减少存储缓存层次，在片上多处理器设计领域，具有广阔的发展前途。

参考文献

- [1] 汤子赢, 哲凤屏, 汤小丹. 计算机操作系统. 西安, 西安电子科技大学出版社, 2003.6, p1-4.
- [2] 姜咏江. 一种 PU-MU-CHL 结构计算机核心设计. 中华人民共和国国家知识产权局 200610072761.3, 2006.12.25.
- [3] 姜咏江. 内存分块并行计算机研究. 计算机应用与软件, 2005, 10: p3-4.
- [4] 陈国良, 吴俊敏, 张隆兵等. 并行计算机体系结构. 北京, 高等教育出版社, 2002.9, p435-446.
- [5] John L.Hennessy, David A.Patterson. 计算机系统结构. 北京, 电子工业出版社, 2007.8, p45-51, p134-138.
- [6] Altera. Quartus II. [HTTP://www.Altera.com.cn](http://www.Altera.com.cn)
- [7] David Money Harris, Sarah L.Harris. 数字设计和计算机系统结构. 北京, 机械工业出版社, 2008.1.
- [8] 胡武伟, 李国杰. 纳米级工艺对微处理器设计的挑战与机遇. 信息技术快报, 第 6 卷第 2 期.

THE DYNAMIC core OF COMPUTER PMC362

Jiang Yongjiang

School of Information Technology & Management Engineering, University of International Business and Economics, Beijing (100013)

Abstract

PMC362 is dynamic core of the computer. In this computer all program can schedule self and execute on the all Performace Unit(PU). Research the many core, all PU can't record anything to instruction in the dynamic core. As the First-level Cache, Memory Unit(MU) consist of with all environmental condition devices, that program in it. The MU can connect to any PU with the signal and do it's task. Any procedure can shift to other PU immediately. This kind of design can realize all processor work fully.

Keywords: Many Core; Memory Unit(MU); Supper parallel program execution; Dynamic Computer; Performace Unit(PU)

作者简介: 姜咏江 1945 年生。中国计算机学会高级会员。计算机设计专家，中嵌联盟特聘专家。研究计算机理论与设计。成功设计了多种计算机核结构。个人制作了教学计算机 PMC110 等。