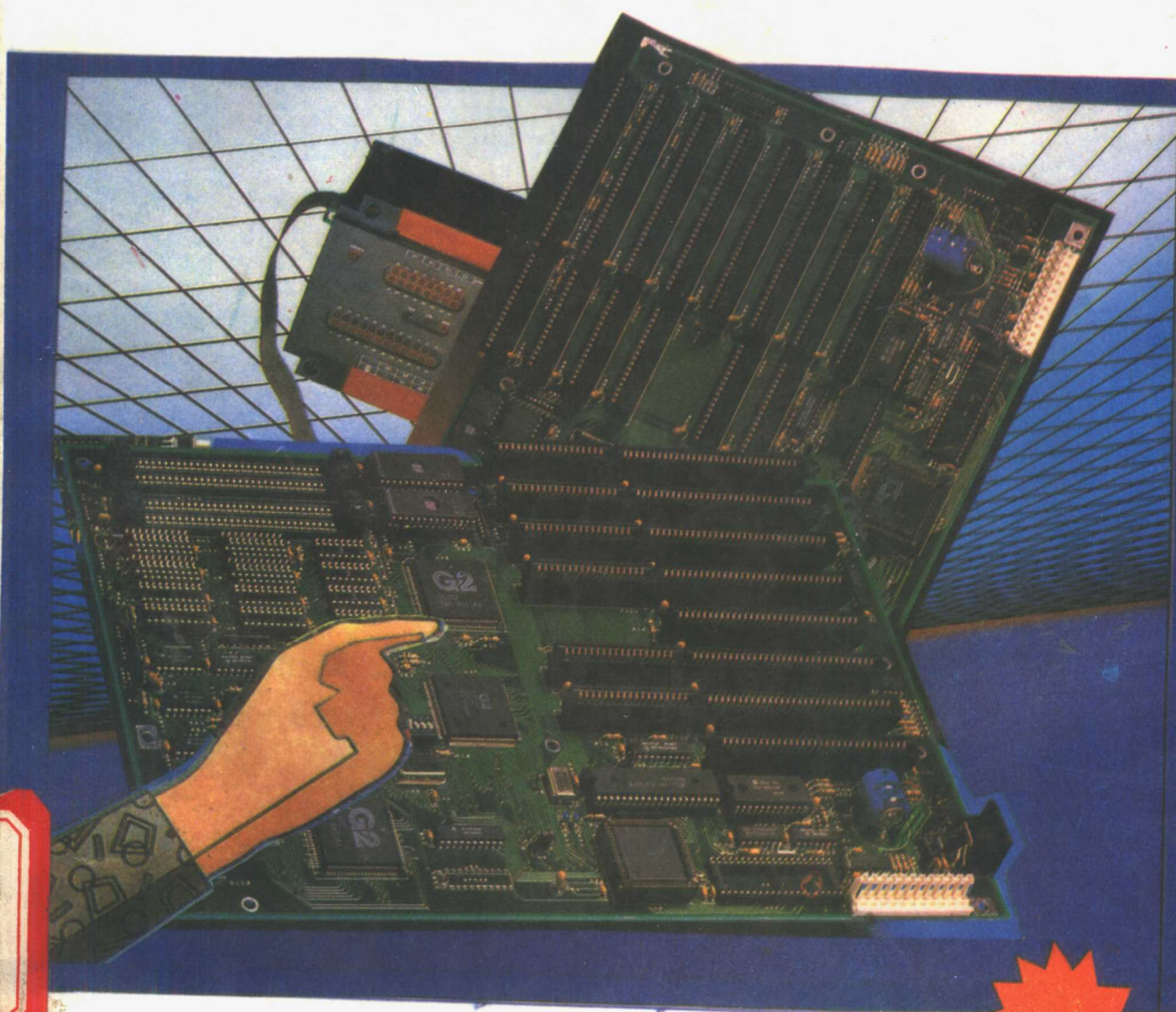


北京希望电脑公司计算机集成电路技术丛书

计算机科技人员必备工具书

最新计算机集成电路技术大全

—— CMOS BiCMOS 数据手册



海洋出版社

下
册

最新计算机集成电路技术大全

CMOS BiCMOS 数据手册 下册

11-31-90

北京希望电脑公司计算机集成电路技术丛书

计算机科技人员必备工具书

最新计算机集成电路技术大全 —— CMOS BiCMOS 数据手册

下 册

王文 魏林 沈齐 编译

海洋出版社

1991.5

内 容 简 介

本书是集 CMOS 和 BiCMOS 数据及产品信息大全的手册。共分上、中、下三册,下册包括八章,主要介绍了 RISC、模块、ECL、军品信息、桥式 MOS 系列产品、设计和编程工具、质量及产品可靠性以及产品的封装形式等。内容丰富,查阅方便,对计算机硬件设计人员,维修、使用人员本书是一本必备的工具书。

责任编辑: 阎世尊

最新计算机集成电路技术大全 CMOS BiCMOS 数据手册 (下册)

王文 魏林 沈齐 编译

海洋出版社出版(北京市复兴门外大街1号)

海洋出版社发行 兰空印刷厂印刷

开本: 787×1092 1/16 印张: 84.75 字数: 470 千字

1991 年 5 月第一版 1991 年 5 月第一次印刷

印数: 1-3000 册

ISBN 7-5027-2249-1 / TP·38 定价: 16.00 元

前 言

本书是集 CMOS 和 BiCMOS 数据及产品信息大全的手册；通过阅读本书，不仅可以纵观 CMOS 和 BiCMOS 的所有技术数据，而且可以了解其产品信息及其它厂家生产的同类产品信息。本书既可供微机硬件人员使用，而且还可供工作站、小型机及大型机硬件人员及维修人员使用。

本书介绍的是美国 Cypress 半导体公司的产品。Cypress 公司是世界著名的半导体生产厂家之一，其产品覆盖面广、质量可靠，在世界半导体行业中占有重要地位。

全书总分上、中、下三册。上册主要介绍产品信息及 RAM；中册主要介绍 PROM、EPLD、FIFO、逻辑器件等；下册则涉及 RISC、模块、ECL、军品信息、桥式 MOS 系列产品、设计编程工具、质量及产品可靠性、温度数据及产品封装形式。本书对各芯片的介绍遵从一定的顺序，使得读者可以快速查阅所需技术数据，做到一目了然。

工欲善其事，必先利其器。了解完备的技术信息是硬件人员设计、维修、使用和购置硬件产品的先导；阅读此书，可使您做到“心中有数”。

本书的问世得到北京希望电脑公司秦人华老师的大力支持，在此表示衷心感谢！

编译者

1991.5

目 录

第七章 RISC 简介	1
CY7C601 32 位 RISC 处理器	8
CY7C602 浮点单元	21
CY7C604 高速缓存控制器和存储管理单元	29
CY7C605 高速缓存控制器和存储器管理单元	46
CY7C611 32-bit RISC 控制器	64
第八章 全定制存储块性能	75
CYM1220 64K×4 SRAM 存储块	78
CYM1240 256K×4 SRAM 存储块	79
CYM1400 32K×8 SRAM 存储块	80
CYM1420 128K×8 静态 RAM 存储块	82
CYM1421 128K×8 静态 RAM 存储块	83
CYM1422 128K×8 静态 RAM 存储块	85
CYM1423 128K×8 静态 RAM 存储块	90
CYM1441 256K×8 SRAM 存储块	92
CYM1460 512K×8 静态 RAM 存储块	97
CYM1461 512K×8 静态 RAM 存储块	103
CYM1464 512K×8 SRAM 存储块	109
CYM1540 256K×9 具有独立输入/输出口的缓冲存储块	110
CYM1610 16K×16 静态 RAM 存储块	115
CYM1611 16K×16 静态 RAM 存储块	117
CYM1620 64K×16 静态 RAM 存储块	124
CYM1621 64K×16 静态 RAM 存储块	126
CYM1622 64K×16 SRAM 存储块	132
CYM1623 64K×16 SRAM 存储块	137
CYM1624 64K×16 SRAM 存储块	139
CYM1641 256K×16 静态 SRAM 存储块	144
CYM1720 32K×24 SRAM 存储块	149
CYM1821 16K×32 静态 SRAM 存储块	154
CYM1822 16K×32 独立输入/输出口的静态 SRAM 存储块	161
CYM1830 64K×32 静态 SRAM 存储块	169
CYM1831 64K×32 静态 SRAM 存储块	175
CYM1841 256K×32 静态 RAM 存储块	180

CYM1910 16K×68 SRAM 存储块	187
CYM1911 16K×68 SRAM 存储块	192
CYM4210 级联 8K×9 FIFO	199
CYM4220 级联 16K×9 FIFO	201
第九章 ECL	203
CY10E301 / CY100E301 组合式 ECL 16P8 可编程逻辑器	203
CY10E302 / CY100E302 组合式 ECL 可编程逻辑器件	208
CY10E422 / CY100E422 256×4 ECL 静态 RAM	214
CY10E474 / CY100E474 1024×4 ECL 静态 RAM	220
CY1E484 / CY10E484 / CY100E484 4096×4 ECL 静态 RAM	229
CY1E494 / CY10E494 / CY100E494 16384×4 ECL 静态 RAM	230
第十章 军用产品概述	237
第十一章 Bridge MOS	251
第十二章 设计和编辑工具	253
QuicKpro	253
CY3101 PLD Toolkit	256
CY3200 PLDS-MAX+PLUS 设计系统	258
CY3300 QuicKpro II	262
第十三章 质量和可靠性	266
第十四章 封装	287

第七章 RISC 简介

简介

本章主要从总体上概括了 RISC 计算机结构的基本概念和优点，简要总结了由 Cypress 公司的 CY7C600 系列推出的 RISC 计算机的特性。

可缩尺的处理器结构

应用 Cypress 公司 CY7C600 系列开发的 RISC 结构简称为 SPARCTM。SPARC 意思是：可缩尺处理器结构，它适用于大规模、高性能机器以及小型机。术语“Scalable”是指芯片上最小走线宽度，走线越细，芯片速度越高，然而，由于某些芯片中结构相当复杂，芯片的设计尺寸并不能很好的缩尺（即无法严格按比例压缩），而 CY7C600 由于其设计的简单性，却能很好地进行比例缩尺。因此，随着较好的半导体制作工艺的不断完善，CY7C600 系统的工作速度将会越来越快，SPARC 是一种开放型结构，我们有理由相信 SPARC 凭其高明和富于竞争的设计特点，将建立起一套工业标准。有关设计方面的详细说明已经出版，其它的一些供应商也正在生产 SPARC 微处理器。

什么是 RISC

RISC 是简化指令集计算机的简略缩词，它的风格是强调计算机结构的简单性和高效性。RISC 设计是从足够的不可缺少的指令集开始的。具有代表性的如用一些简单操作可以完成几乎所有的计算操作。RISC 机器的速度比那些具有传统复杂指令组的计算机结构的机器速度快 2-5 倍。而且，RISC 机器由于其较简洁的设计，较易使用，因而具有更短的研制开发周期。

RISC 是由汇编语言向高级语言演化的产物。汇编语言程序有时会用到复杂的机器指令，而高级语言编译程序则很少用到。例如：大多数 C 编译器中使用了 CISC 机器中可用指令的 30% 左右。研究表明，一个典型程序中，大约 80% 运算只需要处理器中指令集的 20% 左右。

RISC 使 UNIX 操作系统在软件上能实现的功能可用硬件得以实现。UNIX 系统证明了操作系统能做到简单又有用。硬件研究得到了相同的结论，半导体技术的不断进步降低了消耗在处理过程和存储器上的开销。因而，过于复杂的指令组成了影响系统性能提高的不利用因素。RISC 机器的设计者通过机器结构和编译器器设计的密切配合来努力简化硬件实现。

设计过程的每一步骤，设计者都必须反省：某一特点对计算机性能的影响程度究竟如何以及实现时的花费是否合算等。每一个附加的功能，无论它在独立条件下是多么方便有用，它的引入总会导致其它执行速度的降低。

RISC 结构的目的在于通过执行一些软件系统中不常用的从而实现一个设计的最大有效速度，其中包括那些能获得净性能增益的纯硬件指标。这种性能增益情况是通过对一些大规模高级语言程序进行的

详细研究而获得的。

RISC 系统通过提供一些建筑来提高性能。这些模块能适当组合达配来实现高级而不必运用过多的而又复杂的指令。

RISC 结构:

下面是 RISC 结构的一些典型特性, 包括 CY7C600 的设计:

- 单周期的执行。

大多数指令的执行在一个机器的周期中完成。

- 无微代码的硬连线控制。

微代码的使用会增加复杂性和每条指令的执行周期数。

- 装入 / 存储, 寄存器到寄存器的设计。

所有的计算指令均用到寄存器。存储器的访问仅在装入和存储指令时进行。

- 具有极少寻址方式的简单的固定格式指令。

所有指令都是单字长 (典型的为 32bit), 且具有很少的寻址方式。

- 流水线结构。

指令组的设计允许几条指令在同一时间同时执行。

- 高性能存储器。

RISC 机器至少有 32 个寄存器和大容量高速缓冲存储器。

- 到软件的功能迁移。

只有那些在某种程序上提高了性能的特性才能在硬件上加以实现, 程序中包括一系列用于完成复杂的简单指令而不再是复杂指令本身。

- 编译程序可视的简单、高效的指令流水线。

例如, 在下一条指令执行后, 转移指令有效, 它允许在当前指令执行时提取下一条指令。

增强性能的关键在于单周期执行以及使周期尽可能的短。RISC 结构的许多特点, 如: 装入 / 存储以及寄存器到寄存器的设计都有获得利于单周期执行的实现, 另一方面, 简单的固定格式指令减少了指令译码时间, 从而有更短的执行周期。注意其中的某些特性, 尤其是流水线法和高性能存储器在超级计算机的设计上已有多年的应用历史, 不同之处则在于在 RISC 结构中, 这些应用已被集成于一个具有简单指令组, 无微代码的处理器中。把函数功能从运行时间移到编译时间来完成亦能提高性能。在编译时间中计算出的函数不需要在程序每次运行中作进一步计算, 而且, 编译程序能重新安排流水线指令的顺序, 安排好寄存器到寄存器之间的操作以便较好地存储计算结果。

新的简化设计标准已经制定:

- 指令应尽量简单, 除非有充分的理由使其复杂化。作为参考: 一条增加周期 10% 的新指令至少要使总的执行周期总数目降低 10% 左右。

• 微代码通常不如硬连线指令序列执行快。把软件微代码化并没有带来什么好处, 这只能使之更加难以修改和调整。

• 固定格式指令及流水线执行比程序的长度更为重要, 随着存储器越来越便宜, 速度越来越快, 空间 / 时间的权衡应以有利于时间的安排为主, 减少存储空间不再会降低运行时间。

- 编译技术应采用简单的指令来产生较复杂的指令, 而不是象 70 年代做的那些用错综复杂的微码

化的指令代替一些简单的指令，优化的编译程序能形成简单、快速的指令序列，而不是复杂的高级代码，操作数可保存在寄存中以进一步提高速度。

RISC 在速度方面的优点

采用任何一种已给定的基准，一台特定的计算机的性能函数 P 与基准的指令计数值 I ，每条指令所需的时钟周期数 C 以及与时钟速率 S 的倒数均成反比关系。假设一台 RISC 机器在机器的时钟条件下运行，那么， S 是相同的，而每条指令执行所需要的时钟周期数对 RISC 机器而言其值在 1.3~1.7 之间，对传统机器则在 4~10 之间。这使得 RISC 机器具有更多功能较强的指令，对于同一个程序，RISC 机器就必须执行更多条的指令，较为典型的是多执行 10% 到 30% 的指令。既然 RISC 机器要多执行 10% 到 30% 的指令，而每条指令执行又快 3~6 倍，因而，执行一个规模较大的程序，它们比传统机器的执行速度快 2~5 倍。

由于在 RISC 机器中用一些简单的指令代替了一条复杂指令从而降低了指令代码密度，所以在某种程序上，RISC 机器中编译后比传统机器中编译后的程序规模要大。所有的 SPARC 指令都为 32bit 长，而传统机中一些指令却要短一些。实际上，执行的指令数并不象增大了的程序规模所预示的那样。例如，一个窗口型寄存器文件常简化调用/返回序列，使得现场转换的开销得以减少。

CY7C600 结构

SPARC CPU 包括：一个进行基本处理的 CY7C601 的整数单元 (IU) 和一个用于浮点数计算的 CY7C602 浮点单元 (FPU)。CY7C602 是一个与 SPARC 兼容的浮点单元。典型的基于 CY7C600 的计算机有一个存储器管理单元 (MMU) 和一个用于存储指令和数据的大范围虚拟地址空间高速缓冲存储器。它们均由 82bit 数据和指令总线连接。整数和浮点单元均并行操作。FPU 由一套浮点算术单元进行浮点数的计算，CY7C600 结构还提供一个与其它协处理器相连接的接口。

指令类别

- CY7C600 结构大约有 50 条整数指令。CY7C600 指令大致分为七个基本类：
- 装入和存储指令（访问存储器的唯一途径）

这些指令用两个寄存器或用一个寄存器一个常数 (const) 来计算所需的存储器的地址，半字长访问必须对准 2 字节边界，字访问必须对准 4 字节边界，双字访问则必须对准 8 字节边界。这些对准限制大大加快了存储器的访问速度。

- 算术/逻辑/移位指令

这些指令计算出两个源操作数的运算结果，并把它们存入寄存器中，它们可完成算术、逻辑或移位操作。

- 浮点数和协处理器指令

这些指令包括浮点数的计算指令，浮点数寄存器操作指令及涉及到协处理器的指令。当需要时，浮点数操作指令与 IU 指令及其它浮点数操作指令可并行执行。这种并发对程序页来说是透明的。

- 控制转移指令

这包括调转、调用、陷阱和转移指令。控制转移常被延迟到下一条指令执行完之后，这样才能保证每次控制转移发生时指令流水线不会空。因则，编译程序可以优化延迟的转移处理。

- 读/写控制寄存器指令

这些指令包括读、写各种控制寄存器内容的指令，通常，这些指令本身指明了源和目的地。

- 人工智能指令

包括带有标记符的算术指令，用原文指令、带标记符指令对实现人工智能语言如 LISP 是很有用的，因为，标记符能自动地给软件释程序提供算术操作数的数据类型。

- 多重处理指令

包括两条用于存储器中完成信号的指令：一个是 Atomic Load/store 无符号位字节，它从存储器中取出一个字节，然后把该存储器单元为全 1，另一条指令是 SWAP，它完成寄存器和存储器单元间内容的交换。两条指令均是“atomic”，即不允许中断的。

寄存器窗口

CY7C600 设计中提高系统性能的一条独一无二的特色是它的重叠寄存器窗口，由于程序的调用而放入寄存器中的结果可以自动变成为被子程序所用的操作数，这样就减少了对主存的装入/存储指令的需要。

根据结构上的具体情况，可能有 2~32 个寄存器窗口，每一个窗口都有一个工作寄存器和 8 个全局寄存器。基本应用包括有 8 个寄存器窗口，每个窗口有 24 个寄存器（其中 8 个重叠，故仅记为 16 个），加 8 个全局寄存器，共 136 个寄存器。最近的研究表明：CY7C600 系统中（但不是其它商用 RISC 机器）的寄存器窗口和带标记符算术功能足以提供要求 AI 语言（如 LISP 和 Smalltalk）的专家系统的开发所需要的出色性能。

陷阱和中断

CY7C600 设计能支持各种陷阱和中断，它们由一张能支持 128 种硬件陷阱和 128 种软件陷阱的表格来完成。尽管浮点数指令能和整数指令并发执行，但浮点数由于 FPU 从上述所提的表格中提供了所用指令的地址，因而，它的陷阱是较为精确的。

保护

一些 CY7C600 指令具有特权性，只有在处理器处于临控程序方式时才被执行。这种指令的执行确保用户程序不会意外地修改相对于外围设备的机器状态。

CY7C600 设计也提供了存储保护，这一点对多任务操作的顺利进行是必须的。存储器保护使得用户程序不可能破坏系统，其它用户程序及它们自身。

开放型结构

开放型结构之优点

CY7C600 设计是第一代开放型 RISC 结构,也是少数开放型 CPU 结构中的一种。由于标准化使用户有可能在竞争激烈的市场上买到性能价格比最好的硬、软件,所以标准化产品比专卖品显示出更多的好处。集成电路由数家相互竞争中的半导体厂商提供,而软件则由系统制造商提供。但如果用户受限于某些专卖处理器时,这些好处就会失去。

RISC 结构,特别是 CY7C600 设计,由于它们相对的简单性,所以实现起来比较容易,由于他们有很短的设计周期,RISC 机器能很迅速地采用新技术。

CY7C600 系统能支持:

- C 语言和 UNIX 操作系统。
- 数值应用 (使用 FORTRAN 语言)
- 人工智能语言与专家系统应用,使用诸如 LISP 及 PROLOG 等 AI 语言。

支持 C 语言比较容易,大多数现代硬件结构均能支持 C 语言,关键的特性是字节的可寻址性。数值应用需要快速的浮点数操作,人工智能应需要大的地址空间和数据类型可交换性。具有流水线浮点数操作能力的浮点处理器实现了数值应用所需要的高性能。

对于人工智能和专家系统应用,CY7C600 系统提供了带标记符指令和字对准。由于象 LISP 和 PROLOG 语言常是解释执行的,字对准性能使解释程序对整数和同类型指针的操作和交换变得更为容易,在带标记符指令中,每个操作数的低两位确定了操作数的类型,如果操作数是一个指针变量,则大多数时间进行存储器访问。语言解释程序能把操作数放在适当的寄存器中,从而大大地改善了程序的环境。

CY7C600 结构并不要求有存储器管理单元 (MMU),但 SPARC 结构规定了一个高性能的 MMU 单元。相同的处理器将被不同类型的机器所使用,例如:带有嵌入应用码的单用户机器就不需要 MMU。相反,一台分时的多任务机器,例如传统的 UNIX 工作站,就需要分页 MMU。进一步说:一个多任务处理机如向量机或 hypercube 需要专门的存储器管理。CY7C600 结构可以针对不同应用目的采用不同 MMU 配置来完成,这并不会影响用户软件。

CY7C600 机器及其它 RISC 机器

CY7C600 设计比其它 RISC 结构更接近于 Berkeley's RISC-II 结构。与 RISC-II 结构相同,为了减少装入/存储指令数,它使用了寄存器窗口,CY7C600 结构允许 32 个寄存器窗口,但早期设计只有 8 个窗口。带标记指令是从应用 RISC-II 结构开发的 SOAR: Smalltalking on A RISC 处理器中获得的。(SOAR 在实现了 RISC-II 后在 Berkeley 公司得到了发展)。

CY7C600 系统为优化的浮点数处理性能而设计。能支持单精度,双精度和扩展精度操作数及其操作。这些均遵循 ANSI/IEEE754 浮点数标准。高浮点数处理性能来自于 IU 和 FPU 的并发操作性。当浮点单元 FPU 进行计算时,整数单元便同时并行装载和存放浮点操作数。如果产生错误(如浮点数异常),浮点单元就会精确地确定陷阱发生的位置,经整数单元自行处理后继续执行。而且,浮点数单元有一套内部指令队列。当整数单元处理不相关功能时,就可执行这些指令。

CY7C600 系统可提供很高的性能,结构的灵活性使得未来的系统所传送的性能比原型机要超出许多倍,而且,结构的开放是它能最迅速地利用技术进步,完善自己的体系。

CY7C600 产品系列

CY7C600 的设计提供了实现高性能计算机和控制器的完整方案。该系列包括：整数单元，浮点控制器，浮点数处理器，高速缓冲存储器控制器和存储器管理单元以及高速缓存器，数据 RAM 等成员。

SPARC 处理器系列由执行所有非浮点数操作的 CY7C601 整数单元和 CY7C608 浮点数控制器 FPC 组成。CY7C608 浮点数控制器与 CY7C609 浮点数处理器相连接 IU 并行运行来处理浮点数支算。同时还支持第二类通用协处理器的接口。IU 通过一个 32 位地址总线和一个 32 位数据/指令总线访问外存储器。在典型的数据处理应用中，IU 和 FPU 常由一高性能 CY7C604 高速缓存控制器和一存储器管理单元以及由 CY7C657 高速缓存 RAM 构成的高速缓冲存储器相联，协调工作。在许多专用控制器应用场合 IU 利用自带的高速内存就能实现独立工作。

CY7C601 整数单元

IU 是一种基本的处理单元，它能执行除浮点数操作以外的所有指令。CY7C601IU 包括一个大型 136×32 的三端口寄存器文件。这个寄存器文件分为 8 个窗口，每个窗口包括 24 个工作寄存器并可访问 8 个相同的全局寄存器。处理器状态寄存器中的当前窗口指针 (CWP) 始终指示着当前工作窗口。

当处理器调用于程序时，CWP 减量，当处理器返回时，CWP 增量。每个窗口的寄存器分成三类：输入寄存器，输出寄存器和本地寄存器。每个窗口与他所毗邻的窗口共同分享它的输出和输入寄存器。前一窗口的输出寄存器是下一个窗口的输入寄存器。全局寄存器适合于所有窗口，而本地寄存器只对各自的窗口有效。当最后一个窗口的输出寄存器成为第一个窗口的输入寄存器时，所有窗口就构成了一个循环的整体。

IU 提供用户和临督程序两种以支持多任务操作系统。一些指令属于特权指令，只有在处理器处于临控程序方式时才能执行。从用户方式转到临控程序方式需要实现硬件中断或执行陷阱指令。

IU 即支持异步陷阱（中断）又支持同步陷阱错误条件和陷阱指令。陷阱使得控制转移给表格范围内的一个偏移量，该表的基址由陷阱基址寄存器确定，偏移量是陷阱类型的函数。陷阱发生在当前指令产生任何为程序员可见的变化之前，因而，可以认为它能在指令运行期间产生。

CY7C602 浮点单元

CY7C602 浮点数单元为 7C600 系统提供了高性能的功能。它能与 CY7C601 并发操作运行。CY7C602 对存储器进行访问的所有地址和控制信号均由 CY7C601 提供。浮点指令由 CY7C601 提供地址并由 CY7C601 和 CY7C602 并行编译。但只有在指令被从 CY7C601 来的一个信号激活后才开始由 CY7C602 执行。挂起的和当前执行的 FP 指令被置放在片上队列中，此时 IU 继续执行其它非浮点指令。

CY7C602 有一个 32×32 位的数据寄存器文件用于浮点操作。这些寄存器的内容在 CY7C601 运用浮点装入/存储指令的控制下，从外存中传送进来，或被送到外围存储器中。浮点装入/存储期间对数据进行访问所需要的地址和控制信号由 CY7C601 提供，而数据则由 CY7C602 发送和接收。尽管 CY7C602 与 CY7C601 并发执行，但从一个包含有浮点运行程序的运行结果来看就如同指令是顺序执行的一样。

CY7C604 高速缓存控制器和存储器管理单元。

CY7C604 高速缓存控制器和存储器管理单元 (CMU) 为请求分页式虚拟存储器环境提供了硬件支持, 这些虚拟存储器 CY7C601 服务。CY7C604 遵循为存储器管理所确定的 SPARC 结构, 页大小定为 4K 字节, CMU 把从处理器中获得的 32 位虚拟地址转换为 36 位物理地址, 并提供写通过 (write-through) 和缓冲拷贝返回 (copy-back) 高速缓存方式。片上的现场寄存器能多达 4096 个现场 (contexts)。

高速地址查寻由片上的转换后缓冲器 (TLB) 提供。每一个入口包含一个 4K 字节页的虚拟物理地址映射。如果在 TLB 入口中发现虚拟地址的匹配, 该入口中的物理地址转换量就会传送给 CMU 的输出端口, 如果来自于处理器的虚拟地址在 CMU 中没有的相应入口, CMU 就会利用片上硬件且访问一个常驻主存的三级页表自动实现虚拟地址间的转换。每个匹配的 TLB 入口的防止外界干扰情况被自动查询, 并把它作为存储器异常报告给整数单元。

CY7C604 也为 2048 个高速缓存地址标记提供存储功能。这些标志是为一个具有 32 字节行的 64K 高速缓存器而设置的。标志入口可以在直接读和写, 在正常的操作中来自处理器的虚拟地址的低 11 位, 即 15~5 位, 用来选取 CY7C604 的标志入口, 并且其中的 16 位内容在芯片上与高 16 比特处理器地址位比较以决定高速缓冲存储器是否包含所需的数据或指令。高速缓存器的 hit / miss 比较被各种内容的保护检查所确证, 并输出结果。流水线访问由芯片上寄存器支持, 这种片上寄存器从处理器获得地址和数据。

CY7C604 也包含系统所需要的逻辑以实现由 SPARC 指令组提供的一字长和半字长写入功能。高速缓存器标志更新也由于 "miss" 特性所特有的自动页面更新而变得简单。这种特性减少了标志更新时对处理器访问。

为多处理器服务的 CY7C605 高速缓存控制器和存储器管理单元。

CY7C605 高速缓存控制器和存储器管理单元 MMU 是对 CY7C604 的扩充。它应用于多处理器系统。CY7C605 提供了一个与 CY7C604 相同 SPARC 标准的 MMU。还增加了一个增强的高速缓存控制器, 它可以帮助总线临控和维护一个多处理器高速缓存所需要的一致协议。CY7C605 提供了一个双高速缓存标志存储器。该存储器允许 CY7C605 在执行总线临控的同时支持 CY7C601 对高速缓冲存储器实现访问。CY7C605 高速缓冲存储器一致协议是建立在 IEEE Future bus 之上的。IEEE Future bus 被认为是一种既能维持高速缓存器的一致性又不致于降低处理器性能的高级协议。

CY7C605 支持数据的直接输入。这种功能是指 CY7C605 支持的高速缓存器直接提供修正数据, 给其它向它发出请求的高速缓冲存储器, 而无需光通过主存储器来提供修正数据。CY7C605 也支持存储器映象。在直接数据插入执行期间, 存储器映象允许存储器系统自动更新。这种特性允许多处理器系统在单一总线上完成发出请求高速缓存器和主存储器的更新。CY7C605 管脚与 CY7C604 兼容。这个特性允许一个系统通过修改操作系统和把 CY7C604 替换成 CY7C605 而把单处理器提高为多处理器系统。

CY7C157 高速缓存器数据 RAM

CY7C157 16k×16 静态 RAM 的设计易于与其他芯片接口，并为 CY7C600 处理器提供了最大性能。该 RAM 具有可寄存的地址输入、带锁存的数据输入和输出以及自定时写脉冲。这使得 CY7C601 整数单元所需的高速缓存器大大的简化。该器使用一个时钟控制地址寄存器的装入，数据输入锁存，数据输出锁存，流水线控制锁存和芯片使能寄存器。该芯片使能信号在时钟控制下进入寄存器，然后以流水线方式通过一个控制寄存器从而调整输出使能。该流水线设计允许这样一种高速缓存器存在，该高速缓存器是作为 CY7C601 整数单元内部指令流水线的扩充，因而极大地提高了其性能，写使能是边缘触发和自定时的，所以免除了用户用外围逻辑电路产生精确写脉冲的必要。一个分离的步输出使能信号提供一个写入期间的输出禁止或是允许其它器件访问总线。

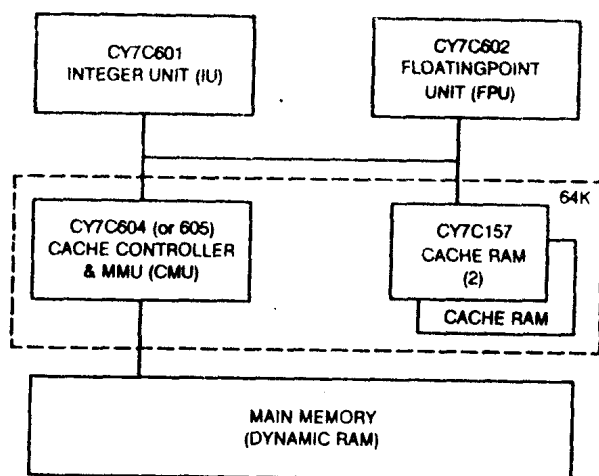


图 1 全系统方框图

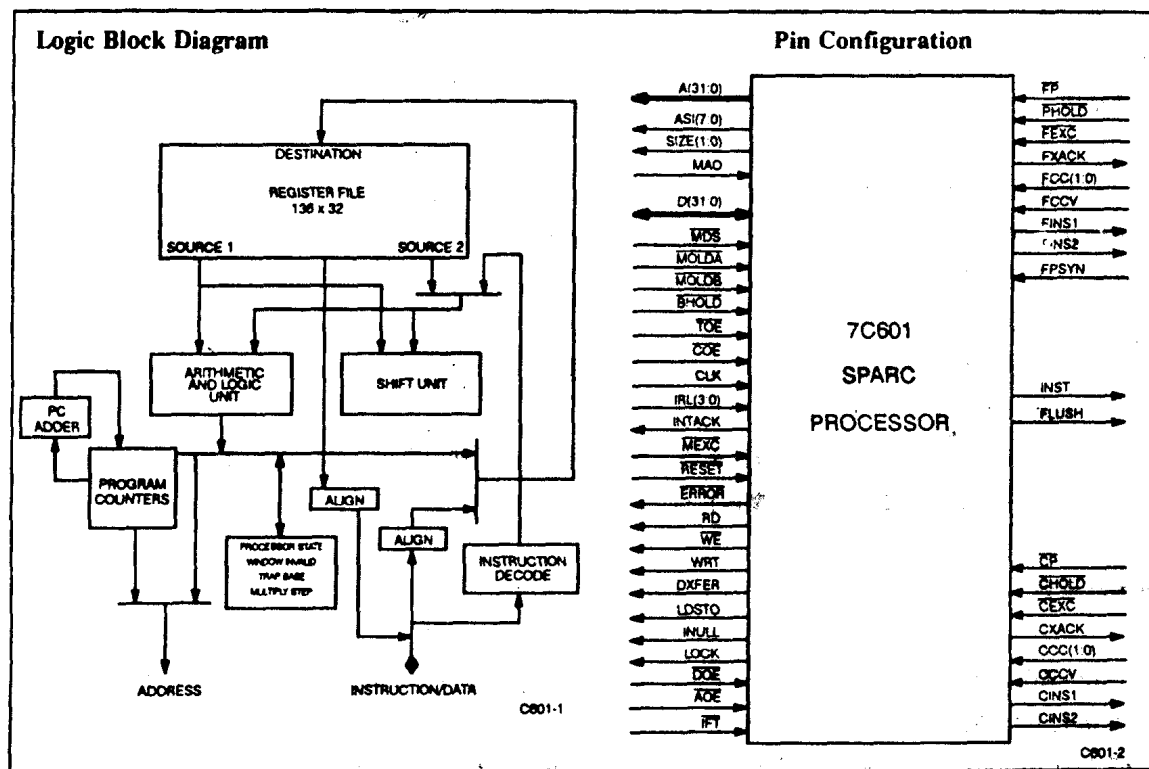
SPARC™, SUN™, 和 NFS™ 是 SUN 微机系统注册商标, UNIX™ 是 AT&T Bell 实验室注册商标, VAX 是数字设备公司的注册商标。

CY7C601 32 位 RISC 处理器

特性

- 简化指令集计算机结构
- 简单格式指令
- 大多数指令为单周期指令
- 高性能
- 25-, 33-和 44MHz 时钟频率可分别产生 18、24 和 29MIPS。
- 快速中断响应
- 四级流水线

- 大型窗口寄存器文件
- 136 个的 32 位寄存器
- 寄存器可用于八个窗口作为低级过程的开销，其中每个窗口含 24 个寄存口。
- 寄存器也可作为寄存器组来完成现场切 / 换。
- 支持多重处理器过程。
- 大范围虚拟地址空间。
- 32 位虚拟地址总线。
- 8 位地址空间识别总线
- 硬件流水线监锁装置
- 支持多任务处理
- 用户方式临控方式
- 特权指令
- 对人工智能系统的支持。
- 为用户定义的协处理器提供高性能接口
- 允许并行执行浮点数指令的 FPU 接口
- 0.8 微米 CMOS 技术
- 207 管脚网格阵列封装或 208 方型扁平封装。



		7C601-40	7C601-33	7C601-25
Maximum Operating Current (mA)	Commercial	650	600	600
	Military			600

功能

CY7C601 整数单元是 SPARC 标准 32 位 RISC 处理高速 CMOS 器件。RISC 结构使得每个机器周期执行一条指令的处理器设计成功成为可能。CY7C601 支持浮点数接口和协处理器接口。这些接口允许浮点指令、协处理器指令并行执行。

CY7C601 SPARC 处理器具有以下特性:

- 简单指令格式

所有的指令都是 32 位长, 并与存储器 32 位边界对齐。三个基本的指令格式特片统一了操作码和地址段的放置。

- 寄存器密集结构

大多数指令在两个寄存器或在一个寄存器与一个常数间执行, 并且把结果放在第三个寄存器中。只有装入和存储指令能访问片外存储器。

- 大型窗口寄存文件

处理器有 136 个片内 32 位寄存器, 这 136 个寄存器包括 8 个重叠的寄存器组 (每组 24 个) 和 8 个全局寄存器。这种结构允许编译程序通过子程序调用来高速缓存本地值, 并提供一个基于寄存器的参数传递结构。

- 延迟的控制转移

处理器总是在控制转移之后才取下一条指令, 并根据控制转移指令中某一位的状态来确定是执行还是取消它。这个特性允许编译程序在一个延迟的控制转移之后重建调整代码以放置一条有用指令, 因而更有利于处理器的流水作业。

- 并行浮点数支行

浮点数指令能够相互并行执行或与其它非浮点数指令一同并行执行。

- 快速中断响应

中断输入在每一个时钟周期中进行采样, 并在 1 到 3 个周期内完成应答。中断服务程序的第一条指令能在接到中断请求后的 6~8 个周期内执行。

7C600 系列

SPARC 处理器系列组成如下: 一个执行所有非浮点操作的 CY7C601 整数单元, 一个与 CY7C601 并发执行浮点数算术运算的浮点单元 CY7C602, 该处理器也为第二类运用协处理器提供支持。CY7C601 通过一条 32 位地址总线和一条 32 位数据/指令总线对外部存储器进行访问。在典型的数据处理应用中, CY7C601 和 CY7C602 同一个高性能 CY7C604 存储器管理单元 (MMU) 以及高速缓存控制器和一个由 CY7C157 16K 字节 $\times$ 16 的高速缓存 RAM 实现的高速缓存器共同作业。在许多专用控

制器应用中, CY7C601 只需要利用自带的高速存储器就可实现独立工作。

协处理器接口

CY7C601 是基本的处理器件, 它执行指令集中除浮点数操作以外的所有指令。CY7C601 和 CY7C602 并行操作。CY7C602 识别出浮点数指令, 并把它们放在一个队列中, 与此同时 CY7C601 继续执行非浮点数指令。如果 CY7C602 遇到一条不适合置于队列中的指令, CY7C602 就会控制 CY7C601 处于保持状态直到该指令存储完毕; CY7C602 包括一组供操作的寄存器。这些寄存器中的内容通过浮点数装入/存储指令, 在 CY7C601 的控制下从外部存储器传送进来或送往外部存储器。处理器联锁装置使得浮点数操作的并发性对于编译程序或汇编语言处理程序具有隐蔽性。一个包含着浮点数计算的程序运行的结果就如同指令是顺序执行的一样。

寄存器

CY7C601 包含着一个大型 136×32 三端口寄存器文件。该寄存器文件被分成八个窗口, 每个窗口有 24 个工作寄存器且均可访问 8 个全局寄存器。处理器状态寄存器中的当前窗口指针 (CWP) 随时记录着当前有效的各窗口状态。当处理器调用子程序时, CWP 完成减量操作, 当处理器返回时, CWP 就会增量。每个窗口的寄存器分为输入寄存器、输出寄存器和局部寄存器。8 个全局寄存器为所有窗口共同分享, 在每个窗口中以寄存器 0~7 出现。寄存器 8~15 作为输出寄存器, 寄存器 16~23 作为局部寄存器, 寄存器 24~31 作为输入寄存器。每个窗口与毗邻窗口共同分享他的输入和输出寄存器。前一个窗口的输出寄存器是当前窗口的输入寄存器, 当前窗口的输出寄存器是下一个窗口的输入寄存器。全局寄存器适合于所有窗口, 而局部寄存器对每个窗口而言均是独享的, 当第 7 个窗口的输出寄存器成为第 0 个窗口的输入寄存器时, 窗口就连接成一个首尾相接的环形整体。

支持多任务处理

CY7C601 通过提供用户方式/监控方式为多任务操作提供支持。一些指令具有特权性, 它们只有在处理器处于监控方式时才被执行。从用户方式转换成监控方式需要产生一个硬件中断或执行陷阱指令。

中断和陷阱

CY7C601 支持异步陷阱 (中断) 和同步陷阱 (错误条件和中断指令)。陷阱使得控制转移给某表格范围内的一个偏移量。该表的基址由陷阱基址寄存器确定, 偏移量是陷阱类型的函数。陷阱发生在当前指令产生任何为程序员可见的变化之前, 因而可以认为它能在指令运行周期产生。

指令集总括

指令分为如下五个基本类型

1. 装入和存储指令