

开发应用微计算机

迎接新的技术革命

第四届全国微计算机学术会议文集

上册

中国电子学会电子计算机学会微计算机分会
湖北省科学技术协会
中国船舶工业总公司第七研究院七〇九所

1984.5.武汉

第四届全国微计算机 学术会议文集

上 集

《文集》编辑组主要成员名单

刘象庆 柳开忠 李正保 程家骥 王仁华
张鸿海 柳浩然 刘胜厚 郭栋梁 张秋菊

一九八四年四月

中国电子学会电子计算机学会
微计算机分会领导成员名单

主任：何绍宗

付主任：李三立 徐培南 沈绪榜 于万源

常委：于万源 马庆乔 张翔 李三立 李晔

李名振 沈绪榜 何绍宗 胡宗藻 徐培南

梁湘 栾贵兴 童颀 孙乐之

委员：何玉珍 张长生 柳维长 刘德贵 石钟伍

余娟芬 顾庆祚 严仁惊 杜毅仁 穆孝芳

葛林根 陈国栋 李仲荣 曹正元 张福炎

朱逸芬 程锦松 林日钿 范棣华 寇化勋

陈兴叶 鲍家元 何延祥 张湘金 赖翔飞

王潼 刘诗嵩 李东信

秘书长：孙乐之（兼）

秘书：贾平良（骊山微电子有限公司科技处）

张宗瑚（航天部陕西管理局科技咨询服务部）

挂靠单位：骊山微电子有限公司

第四届全国微计算机学术会

议论文评审组成员名单

孙乐之	马庆乔	钟玉琢	黄伯臣	许庆贤	李本忍
童 颀	王长胤	鲍亦万	徐则琨	胡久清	王飞龙
黄俊杰	王振宇	黄麟翔	周 细	林 兵	王怀清
杜汉忠	唐敏雄	刘象庆	陈玉彬	高正中	刘帮汉
柳浩然	杨池芬	刘日升	何炎祥	罗 红	郭栋梁
李正保					

前 言

赵总理在《研究‘新的产业革命’和我们的对策》的讲话中指出：“对微计算机的利用一定要摆在重要位置，予以足够重视。”万里付总理在全国电子计算机和集成电路规划会议闭幕式讲话中指出：“我们要翻两番，实现四个现代化，非把电子计算机和集成电路搞上去不可，这个东西上不去，就实现不了现代化。”方毅同志在全国科技工作会议上也强调指出：“从认识上，政策上，措施上把微型机及其应用作为重点来突破。”党和国家领导同志多次讲到大力开发和应用电子计算机，特别是微计算机问题。这些讲话在全国激起了强烈的反响，对我们微机工作者是极大的鼓舞和鞭策。

微计算机价格低，性能好，易学易用。环境要求不高，维护更新容易，投资少，见效快。这一系列优点带来巨大的经济效益，社会效益，大开了人们的眼界，提高了人们的认识。其开发应用象雨后春笋，迅猛发展，神州大地上正在兴起一股强大的运用微机的热潮。中国电子学会电子计算机学会微计算机分会安排的第四届全国微计算机学术会议就是在这样的大好形势下召开的。

这次会议论文征集工作从1983年1月开始，到今年1月20日止，共收到论文684篇，是历届中最多的一次。1979年在陕西临潼召开的全国第一届微计算机学术交流会收到论文96篇，1981年在上海召开的第二届交流会收到162篇，1982年在广州召开的第三届交流会收到362篇。在数量上第四届超过了前三届的总和，学术水平也有很大提高。

获得论文征集的特大丰收，是这次会议的第一个特点。

这次会议贯彻“科学技术必须面向经济建设，经济建设必须依靠科学技术”的方针，以开发应用为重点可以说是又一特点。会议录用的487篇论文中，属于应用的共341篇，占总数的70%。应用范围之广，涉及交通运输、轻工仪表、纺织印染、机械制造、石油化工、地质探勘、采矿冶金、邮政通讯、广播电视、财贸金融、医疗卫生、文教体育、水文气象等行业部门。尤其可喜的是在农业上也开始应用计算机了，如棉花蚕茧收购、农业资源管理、病虫害预报等。在那些一贯靠手抄算盘打的部门，例如机关事务管理、人事档案、工资发放、计划调度等方面也在尝试采用计算机。反映出我国微机事业发展到了一个新时期。

这次会议的第三个特点是论文在深度、广度、学术水平和实用价值上比前几届有较大提高。从会议交流论文及专家评审意见来看，突出一点是有较高实用价值。很多成果已投入使用，经过鉴定，具有较好的经济效益，达到了国内先进水平。例如有些项目使用微机后可提高工效十多倍至几十倍。葛洲坝工程局采用微机控制生产出的混凝土合格率达到99.94%。武汉制瓶厂在七〇九所等单位协助下使用微机控制玻璃制品生产后，耗油量下降4.4%，产品合格率由原来80%以下提高到85%以上，节约仪表投资费50%。这些事例生动地说明了微计算机在发展国民经济中的巨大作用。

近年来我国微机事业的发展取得了可喜的成绩。但必须清醒地认识到，和先进国家相比我们还处在十分落后的状态。首先是应用范围还不够广泛。有人统计国外微计算机应用类目有二万五千左右，而国内只有三一五百种。在发达国家，微计算机已渗透到各行各业各个领

域，而且正在潮水般地涌入机关、学校、家庭。相比之下，我们差距很大。其次在效益方面，虽然不少项目已经取得了相当可观的经济效益，社会效益，但从全面看，有些单位的计算机还没有充分发挥作用。有些推广应用的项目还需要狠抓经济效益。对于已经取得的科研成果还需要走出实验室，转化为生产力，加以工程化、商品化。我们微机工作者要和各行各业的专家们相结合，扎扎实实、艰苦工作，建立起适合我国特点的微机应用体系。第三要发展民族计算机工业。由于我国微电子工业落后，材料试剂设备不配套，集成电路发展慢，影响了微机的发展。据统计，国内97%以上的微计算机是进口的或用进口器件组装的，价格是国外市场的4—6倍。不能这样长此下去。必须采取措施，保护、发展民族计算机工业，研制生产出更多更先进的我国自己的集成电路、微计算机及外部设备。

我们从这次会议录用的论文中，选择了73篇，加上400余篇摘要和题录，编辑成这本文集，在会议召开之际出版，献给与会的各位专家和代表同志们。我们相信这册文集对开好这次会议及今后推广应用这些成果将产生积极的作用。

当前，国际上正在酝酿着以电子计算机为重要标志的新的技术革命。我们国家要在本世纪末实现国民经济翻两番的奋斗目标。我们每一个微机工作者，任重道远，大有可为。让我们团结起来，尽最大努力开发应用微计算机，迎接新的技术革命，为祖国四化建设作出应有的贡献。

中国电子学会 微计算机分会主任
电子计算机学会

何 绍 宗

1984年4月于西安

目 录

(上集)

第一部分 全文

A 类

- 1 从 VLSI 技术看微计算机的发展..... (1)
- 2 航天器控制系统的分布结构..... (11)
- 3 Intel 单片微处理机编程工具研制..... (22)
- 4 智能计时器——单片微机应用..... (33)
- 5 一种多微机构成的数据采集处理系统的设计与实现..... (37)
- 6 980/030 微计算机结构特点与开发技术研讨..... (49)
- 7 微机开发系统及其仿真器在游戏软件开发中的应用..... (62)
- 8 TDJ—1台式电子记账机的硬件设计..... (77)
- 9 控制用微型机系统的可靠性设计..... (85)
- 11 微机与微机市场..... (92)

B 类

- 39 高级报告生成系统..... (100)
- 40 一个面向微型机辅助机械设计的高级图形语言..... (109)
- 41 分布式操作系统 RMMX 80..... (118)
- 42 微机通用数据管理系统 GDMS 介绍..... (125)
- 43 VIDE 全屏幕编辑程序..... (133)
- 44 MCZ—1/70 微型机上的多用户文本编辑系统..... (141)
- 45 CP/M 族通讯软件的开发..... (150)
- 46 程序结构测试系统 PSTS..... (160)
- 47 软件移植的一个新技术
——软件工作台..... (170)
- 48 PASCAL/MTM 可作为软件工程支撑环境中的工具语言..... (182)
- 49 一种分布式实时多任务 OS 的实现与应用..... (194)
- 50 关系数据库管理系统 RDBMS (204)
- 52 一个支持微机数据库系统的文件存贮系统..... (212)
- 53 UCSD PASCAL 在 TRS—80 上的实现..... (218)

54	解通讯网络的最大流问题.....	(234)
55	BASIC 结构化及其实现.....	(240)
56	PROLOG 语言在 BCM 微处理机上的实现.....	(252)
57	APPLE II 微型计算机新反汇编 软件.....	(260)
58	单板机微型机系统上运行的实时多任务执行软 件 RTMTE	(266)
59	国产16位微型机系统软件研制中的探索与收 获 ——TP86系统监控及汇编程序的功能、结构与特点.....	(276)
93	NEWDOS/80 操作系统里的汉字增配.....	(285)

C 类

94	岩石声波综合数据处理系统研究报告.....	(291)
95	微机在激光图象数据处理中的应用.....	(299)
96	在微计算机上建立土地资源面积文件管理系统的研究.....	(306)
97	北京 325 米气象塔微机数据处理系统.....	(315)
98	PERT 微型计算机软件系统.....	(323)
99	机车运用统计专用计算机———单板计算机用于机务统计.....	(332)
113	宾馆电脑管理系统的开发和应用.....	(345)
114	微型计算机在企业管理中的应用.....	(356)
115	用微型计算机实现煤炭生产调度管理.....	(363)
116	微型机科技计划管理系统.....	(369)
117	一个微型机成本管理系统的设计与实现.....	(379)
118	微型机中文销售合同管理系统.....	(389)
119	微型机办公室自动化系统的设计.....	(394)
120	微型机在仓库管理中的应用.....	(400)
121	西文图书检索系统 RESYS 的设计与实现.....	(410)
122	WD—TLX 微型机 图书流通管理系统.....	(419)
(381)		
(141)		
(041)		
(100)		
(070)		
(135)		
(134)		
(304)		
(212)		
(218)		

从 VLSI 技术看微计算机的发展

陕西微电子学研究所

沈绪榜

摘 要

本文从 VLSI 的性质出发,按体系结构,程序语言与应用复盖等三个方面,简述了 VLSI 技术对微计算机发展所起的影响;最后讨论了如何正确发展现在的 LSI 微计算机。

1. VLSI 的性质

为了弄清 VLSI 技术对微计算机发展的影响,先看看 VLSI 技术与现有 IC 技术的差别。由于过去 20 年的发展多少是连续的,LSI 与 VLSI 之间的界线是任意的,但有两个观点已日益明确。集成度主要是靠扩大芯片尺寸与减小线条宽度提高的(图 1a),所以,第一个划线的观点是最小线条宽度在 2μ 以下。因为超过这一点,光刻技术就变得困难,生产芯片所要求的设备变得非常昂贵。另一方面,芯片上元件数的增长,给其设计带来了前所未有的复杂性问题,调试非常困难。所以,第二个划线的观点是,集成度使复杂性超过了所有其它问题,一般指十万个元件以上。这时的芯片设计者将要求具有并行处理网的知识(图 1b)。的确,VLSI 技术就像一个“黑洞”,把系统中越来越多的部份“吞”到芯片上去。

用集成度来表示复杂性,将一般计算机的分级方法用到 VLSI 上(图 1b),是否合适还不清楚。但是,PCB 计算机的设计方法不适合于芯片计算机的设计,这是清楚的。因为对 LSI/VLSI 芯片来说,工程改变更花时间;芯片上信号不好观察;以及一个元件的尺寸和性能的改变,可能影响其周围的元件,设计的不同部份之间,交互更多。所以,规则的可程序设计的结构,使元件之间交互极少的设计方法学,对芯片计算机的设计是头等重要的。用 CAD 工具事先验证设计,用硅编译技术使设计是由证明没有错误的元件构成,这些措施都是伴随 VLSI 技术的发展而来的,是行之有效的解决复杂性问题的方法。

无论从性能和面积来讲,芯片计算机本身的规则性都是需要的。从性能上说,当最小线条宽度减小时,因为横截面积比长度减小得更快,互连电阻增加;而且仍然有许多互连线必须穿过整个芯片。结果是当最小线条宽度达到一定值时,互连线上的信号延迟时间将大于门延迟时间(图 2),对系统性能起主要作用。从面积上说,Donath 等指出[6],芯片上线条的总长度是集成度的函数(图 3),互连线的平均线长度近似地按(线路) $^{1/2}$ 增加,故由所有互连线所占的面积将按(线路) $^{1/2}$ 增加;而且会达到大部份硅片面积由互连线占用,而不是有源元件占用这样的地步。所以,芯片计算机的设计原则将必须从元件使用的优化,转向有效的互连线的安排上。下面将简述 VLSI 技术的这个性质,是如何影响微计算机的体系结构,程序语言与应用复盖等方面的变化的。

2. 体系结构: CISC, RISC 与 MISC

按照规则性的要求, 一个“好”的 VLSI 体系结构应当具有一个或多个下列特性:

(1)、是由少数几种简单的单元 (cell) 实现的; (2)、具有简单而规则的数据和控制路径, 使得单元能用一个局部而规则的互连网络连接起来, 长的不规则的通讯线极少; (3)、大量采用流水线与多处理, 达到大量单元是同时工作, 总的计算速效高。回顾现有的微计算机体系结构, 按其问世的先后, 大致可分成 CISC, RISC 与 MISC 三种情况。

CISC 体系结构: 受传统的影响, 微处理机的设计倾向也是使指令集合, 因而使其相应的体系结构, 越来越复杂。人们称之为 CISC (Complex Instruction Set Computer) 体系结构[4]。包含十万支晶体管以上的 VLSI 微处理机, 如 Intel 公司的 iAPX432, HP 公司的 32 位 μp , 贝尔实验室的 MAC 32 μp 等, 是众所周知的 CISC 型的。通常还把这些设计看成是“VLSI 处理机体系结构”的同义语。这种复杂性的后果是: 设计时间增长, 设计错误更多, 不适合 VLSI 技术实现, 不容易单片化。很不利于微计算机的迅速发展。

RISC 体系结构: 为了适应 VLSI 技术内在的规则性要求, Mead 与 Conway 的“Introduction to VLSI Systems”。以及他们所支持的 MPC (Multiprocessor Chip) 训练班与 Silicon foundries 等, 促进了新的微计算机体系结构的发展。1980 年春天开始了 RISC I 的设计[4]。RISC (Reduced Instruction Set Computer) 的基本思想是: 选择一个简单的指令集合与相应规则的处理机组织, 可以达到一个很高的指令速率, 从而总的处理能力很高。由于这种处理机结构规则, 控制逻辑只占 10%, 设计时间短, 44500 支晶体管的 RISC I 于 1981 年春天就完成了。更有意思的是, 性能上比 CISC 的优越[4]。11 个 HLL 基准程序的 49 个比较中, RISC I 比五台 CISC 型的机器快 1 到 2 倍, 只有一个比较是快于或等于 RISC I 的 (图 4); 五种机器执行汇编程序与 HLL 程序的时间比, 也是在 CISC 机器上 HLL 程序的时间比其对应的汇编程序的时间长得多 (图 5)。说明 RISC I 不仅体系结构更好, 而且更有利于执行 HLL 程序。因为编译实质上是大量的“情形分析” (Case Analysis), CISC 机器的指令与寻址方式多, 情形分析复杂。编译程序的设计者必须在编译程序的速度与目标程序的质量之间加以平衡, 不可能产生所有 CISC 指令所必需的情形分析。HLL 与指令之间的语义冲突, 可能妨碍指令的使用。所以, 对 HLL 仅提供初等要素 (primitives), 而不像 CISC 机器那样提供许多“解法” (solutions), 结构简单规则的 RISC 机器, 反而取得了性能上的优越性[5]。

MISC 体系结构: RISC 是将现有指令集合化简, 使指令少而译码非常简单。MISC (Minimal Instruction Set Computer) 吸收了这些思想, 指令集合是对高级语言的编译都非常有效来优选的。这种新型机器的代表就是 INMOS 公司将于 1984 年推出的 Transputer 系列。它的指令集合支持 Occam 高级语言 (图 b); 只有一种指令形式, 总是一个字节, 和机器的字长无关; 微处理机的结构非常简单, 没有寄存器地址空间, 整个机器也没有 I/O 地址空间, 只有单一的存储器地址空间, 使得编译程序的情形分析更加容易。编译的结果程序与汇编程序的效果一样[2]。因此, 不需要汇编语言, 机器语言对用户是透明的。32 位与 16 位机的指令集合是完全一样的。机器的名字 Transputer 是从 Transistor

与 computer 两个字派生来的, 它表示 Transputer 即是一个单片计算机, 又是一个像一支晶体管那样的硅元件。这个名字反映了它作为系统积木块的能力。的确, 它在体系结构上将处理机、存贮器与并发通讯接口等, 都安排在由250,000支晶体管组织的芯片上(图7)。使得就像过去能用更多的晶体管, 可以构成性能更强的计算机那样, 现在能用更多的微计算机 Transputer 组件, 可以构成性能更强的并行处理系统。

3. 程序语言、顺序与并发

随着 VLSI 技术的发展, 微计算机的体系结构除了更加面向高级语言外, 还提供多机并行处理的能力。特别是由许多相同的自含存贮器与并发通讯接口的处理机, 如 Transputer, 构成多处理机系统时, 将变得处理速度更快, 存贮容量更大, 可靠而经济。为了使多机并行处理系统能有效地使用, 需要有相应的程序语言以描述并发进程, 以及它们之间的通讯与同步问题。

因为并行处理不是微计算机才有的事, 在顺序语言的基础上, 提出过许多办法以适应硬件的并行处理能力。为了描述并发进程, 广泛采用了加关键字的办法如 FORK, JOIN 等; 而用得最普遍的通讯方法是采用公共存贮器。然而, 用它设计程序时会产生严重的问题, 而且在某些硬件实现中会很经济(如用 crossbar switches), 或不可靠(如用 glitches)。对于同步提出的方法就更多: semaphores, events, conditional critical regions 等, 都证明是适合于相应的用途的。

最近, INMOS 公司在研制 Transputer 单片机的同时, 开发了支持与并发与通讯的 Occam 语言, 试图用更简单的与硬件相对应的办法, 描述并发进程, 解决处理机之间的通讯与同步问题。Occam 语言的基本进程有: 赋值进程, 输入进程, 输出进程与等待进程等。为了控制进程的执行次序, 除了 WHILE 与 IF 之外, Occam 语言还定义了 SEQ, PAR 与 ALT 三个基本构造(图8)。从 PAR 构造可以看出 Occam 语言描述并发进程的两个附加原则: 一是各种初始构造, 例如顺序执行的语句或进程的并行集合, 可以组合在一起; 二是用空格多少给编译程序说明一条语句的上文, 而不是用附加关键字来模糊语言的清晰度。采用 PAR 构造说明 Occam 语言更好的反映了硬件的并行处理能力。在 Occam 语言中, 是借助发送与接收信息, 而不是公共存贮器, 来实现进程之间的通讯的。正如用一个数据路径连接彼此通讯的一对硬件那样, Occam 语言提供一个通道(Channel)来连接一对彼此通讯的进程。事实上, 两个异步通讯的硬件通常使用信号交换协议, 作为彼此传送数据的手段, 也作为通讯是否准备就绪的给信号手段。当发送器想传送一个数据时, 如果接收器未准备好, 发送器就等待; 当接收器已准备好时就传送数据。然后, 发送器与接收器恢复其原有的任务。如果在发送器准备好之前, 接收器就要求接收, 也会出现类似的等待情况。两个并发的 Occam 进程就正是以这种方式通讯和同步的。Occam 语言自然地描述并发与通讯的能力, 使它不限于用在传统的体系结构中, 也可以用在新的并行体系结构中。例如, 一个 $n \times n$ 的完成矩阵相乘的心脏收缩阵列(systolic Array), 每个处理元及整个阵列的 Occam 描述将是简单的(图9), 不需要操作系统支持[3]。

用 Occam 语言编写应用程序时, 任何复杂的功能都可以作为进程来程序设计, 用起来

就像顺序语言中的函数与过程那样,这有点像 C 语言和 BCPL 语言。因为 Occam 语言和基本硬件基础紧密相关,它也是实现复杂软件系统的有效手段。作为系统性能的高级设计语言,它在更高的抽象级上,完成布尔代数同样的功能:如果一个门电路能看成一个满足布尔逻辑规则的电子元件,那么,Transputer 就好比一个满足 Occam 规则的可程序设计的元件。

4. 应用复盖:品种与数量

传统上都是靠不同的系列品种来复盖不同的应用领域的。VLSI 技术的复杂性,以及越来越高的应用要求,都倾向于以较少的品种,用不同的数量复盖不同的应用领域。Transputer 系列体现了后一种思想。它只有一种 16 位的 IMST222,一种 32 位的 T424,以及 IMS G213 图示处理机,与 IMS212 磁盘处理机。此外,就是配套的存储器品种。它充分利用问题中内在的并行性,用 Transputer 流水线或阵列来扩大应用领域[7]。所以,它能复盖整个微处理机的应用范围,从价廉的微控制器,到超级计算机,到第五代计算机系统。例如,64 点,256 点,与 1024 点的 FFT 等可以分别用 6 个 8 个与 10 个 Transputer 的流水线完成;数据库数据的搜索可以用流水线或阵列解决。其它应用,如图象处理,有限元分析,矩阵运算,电话交换系统,容错系统,以及人工智能等都可以很自然地使用 Transputer 网。Transputer 的体系结构保证连网时是直接互连的,不需要任何附加逻辑。

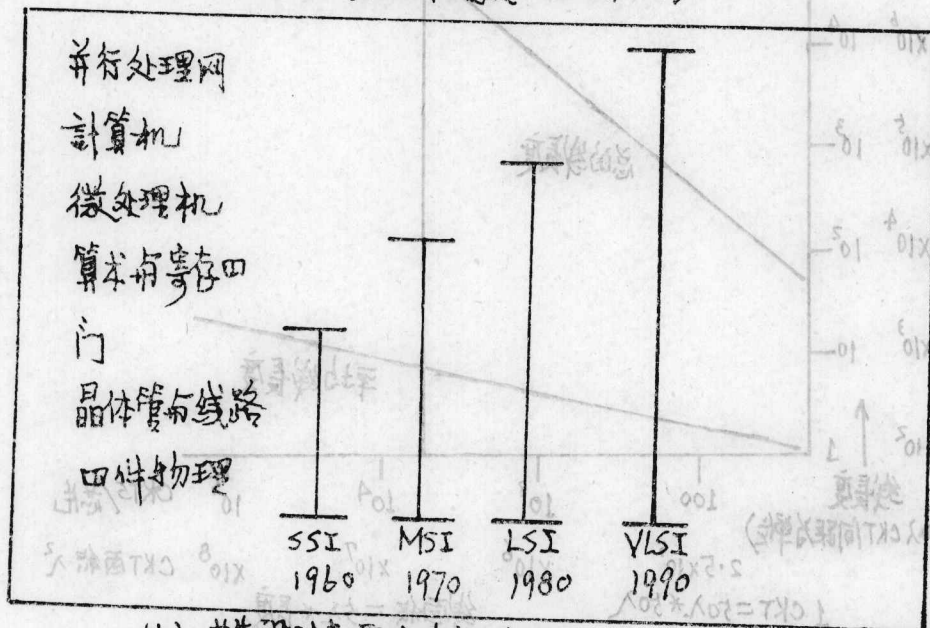
5. 如何发展低档 LSI 微计算机

以上是国际上 VLSI 微计算机的发展情况。从性能、实现与发展来看,对于发展现在的低档 LSI 微计算机,有些什么可以借鉴的地方呢?就性能上讲(图 10),Transputer 机运算速度快,代码密度高,使用方便(不用汇编程序语言,编译效率高,对实时应用程序也可直接使用 HLL)。从实现上讲(图 10),Transputer 机的处理机芯片面积小,效率高,16 位的 S14 处理机只用 8500 支晶体管[1],32 位将只增加三分之一,是目前唯一能用低档 LSI 实现的 32 位单片微处理机。而且 32 位与 16 位的指令格式与指令集合完全一样,便于设计与开发。就发展上讲,Transputer 单片计算机属于第五代微计算机体系结构,减轻了 Von Neumann 结构的“瓶颈”问题,可以直接连成各种大小的阵列。现在的低档 LSI 技术可以用单板实现它,将来的 VLSI 技术用单片实现它,在研制、使用与发展上都有连续性继承性。在 IC 技术迅速发展的今天,为了把我们的低档 LSI 微计算机的体系结构,选择得比当前的工艺现状看得更远些,使它能比当前的工艺存在更久些,使它也是为了将来而设计的,研究一下 Transputer 计算机,也许是有点现实意义与深意义的。因此,本文对它作了较多的介绍。

由于作者水平有限,对 VLSI 芯件计算机的设计也刚刚开始学习,不妥之处,敬请批评指正。

時間	芯片尺寸 a	最小線寬度 b	芯片上晶體管數目 $(a/b)^2/50$
1960	1mm X 1mm	30μ	22
1970	2.5mm X 2.5mm	10μ	1,250
1980	6mm X 6mm	3μ	80,000
1990	10mm X 10mm	0.5μ	8,000,000

(a). IC 的進展 (最小線寬度小於 2μ 時)



(b). 芯片設計者要求的專業知識

(當要求性壓倒所有其它問題時)

125-1

圖 1: 什麼是 VLSI?

10mm 金屬線互連的情況					
λ	5μ	2.5μ	0.5μ	0.1μ	
R	25Ω	100Ω	2.5K	62.5K	
C	4pF	2pF	0.4pF	0.08pF	
RC	100ps	200ps	1ns	5ns	
門延遲	5ns	2.5ns	0.5ns	0.1ns	
由線路支配性能 ←			→ 由互連線支配性能		

(多晶矽互連線近似具有 100 倍於電阻值)

圖 2: 性能與互連線的關係

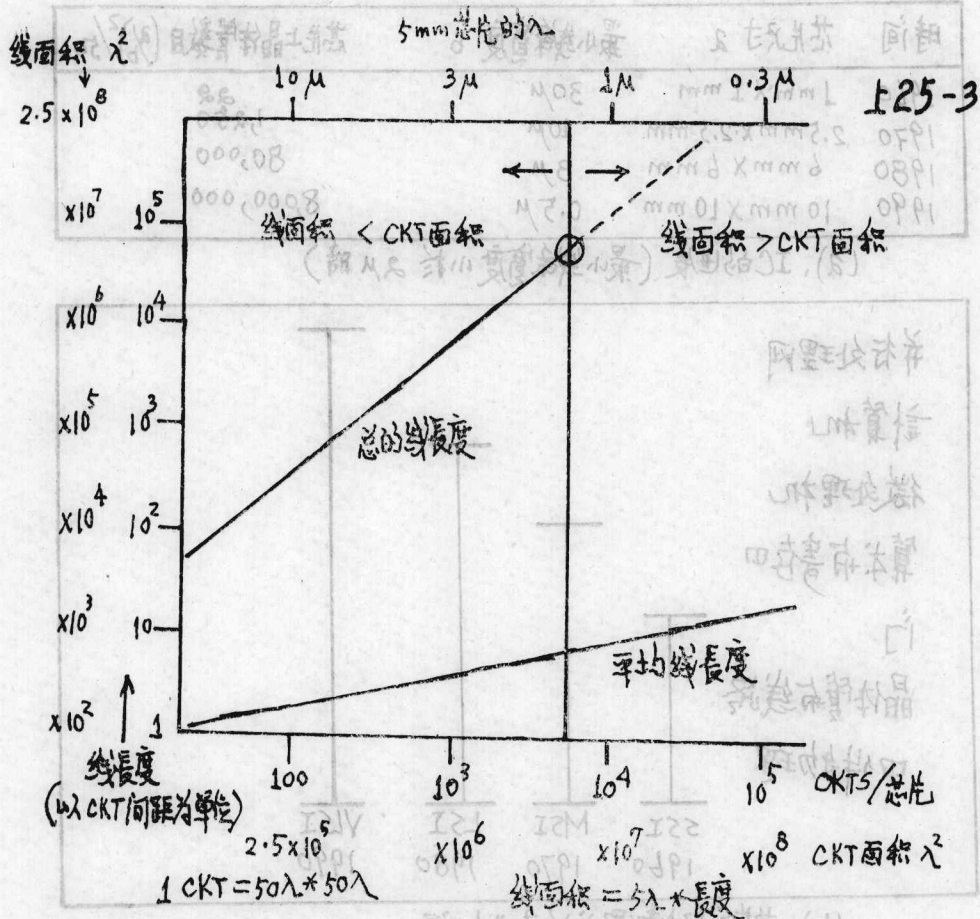


图3: 相对的互连线面积

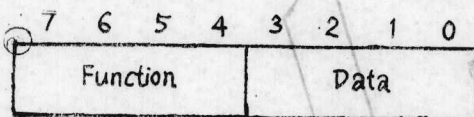
BENCHMARK	RISC I	68000	28002	VAX-11/760	11/70	C/70
	msecs	比 RISC I 快的倍数				
E - String search	.46	2.8	1.6	1.3	0.9	2.2
F - bit fast	.06	4.8	7.2	4.8	6.2	9.2
H - linked list	.10	1.6	2.4	1.2	1.9	2.5
K - bit Matrix	.43	4.0	5.2	3.0	4.0	9.3
I - Quicksort	50.4	4.1	5.2	3.0	3.6	5.8
Ackermann(3,6)	3200	---	2.8	1.6	1.6	---
recursive sort	800	---	5.9	2.3	3.2	1.3
puzzle (subscript)	4700	---	4.2	2.0	1.6	3.4
puzzle (pointer)	3200	4.2	2.3	1.3	2.0	2.1
sed (batch editor)	5100	---	4.4	1.1	1.1	2.6
towers hanoi (18)	6800	---	4.2	1.8	2.3	1.6
Average $\pm$ std. dev.		3.5 $\pm$ 1.8	4.1 $\pm$ 1.6	2.1 $\pm$ 1.1	2.6 $\pm$ 1.5	4.0 $\pm$ 2.6

图4: C Benchmarks: RISC I 的执行时间, 而 RISC I 的性能比上 25-4

BENCHMARK	RISC I	68000	28002	VAX-11/780	11/70
E - String Search	.62	.17	.92	.23	.53
F - Bit Test	1.00	.23	.87	.34	.50
H - Linked List	1.00	.92	.96	.88	.83
K - Bit Matrix	.94	.21	.29	.34	.24
I - Quicksort	.92	.16	.44	.47	---
Average $\pm$ std. dev.	.90 $\pm$.1	.34 $\pm$.3	.46 $\pm$.3	.45 $\pm$.2	.50 $\pm$.2

图5. HLLESF (HLL Execution Support Factor):

汇编执行时间对HLL执行时间的比 上25-5



指令格式

Zero address instructions

(指令中的 Data 定义一个对计算栈中值的运算)

Add
Subtract
Multiply
Divide
Remainder
Long add
Long subtract
Long multiply
Long divide
Normalise
Difference
Greater than
Equal zero
And
Or
Xor
Shift left
Shift right
Move message
Input message
Output message
Load byte
Store byte
Byte count
Word count
Byte subscript
Word subscript
Check subscript
Extend to word
Check part word
Extend to double
Check word
Read timer
Test error
Reverse
Return
Minimum integer
Initialise
Start process
End process
Alt start
Enable channel
Enable timer
Disable channel
Disable timer
Alt wait

One address instructions

(指令中的 Data 被 Function 看作数值)

Load local
Store local
Load local pointer
Load non-local
Store non-local
Load non-local pointer

Load constant
Add constant
Add to memory

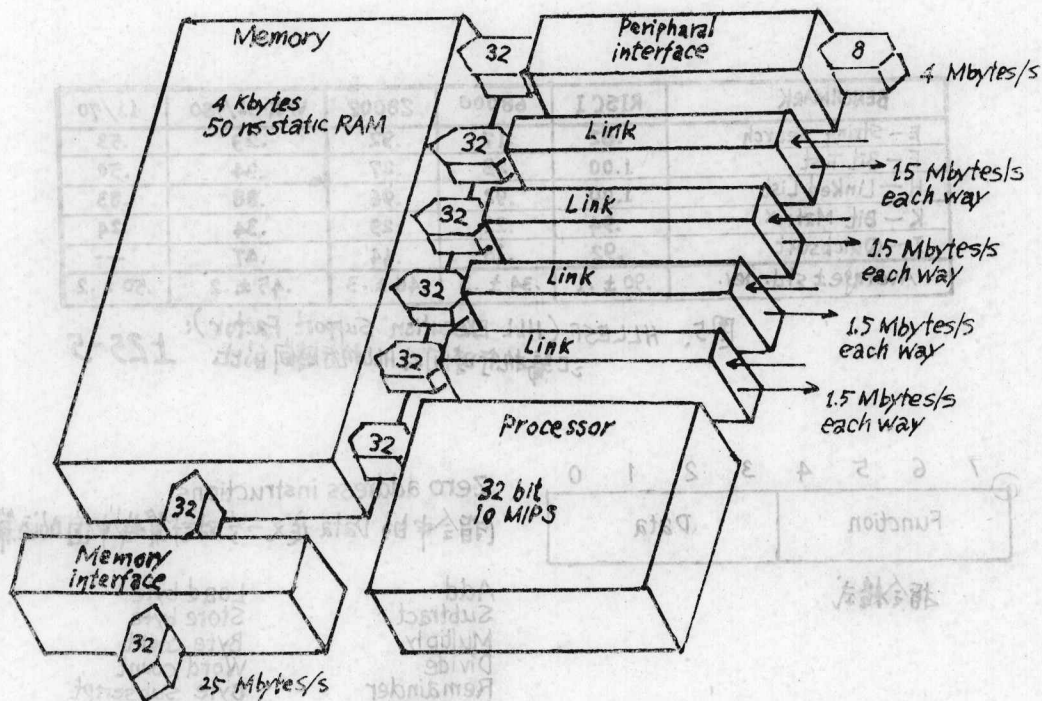
Jump
Conditional jump
Call
Adjust workspace

PREFIX

Complement prefix

(prefix 与 Complement prefix 以统一的方式处理地址指令)
(中的数值,或零地址指令的运算数目。)

图6. Transputer 的指令格式和指令集合 上256



PROCT424 (CHAN Ext.Mem.Outputs, Ext.Mem.Inputs,
Link.Outputs, Link.Inputs,
Peri.Outputs, Peri.Inputs,
Event.Input) =

CHAN Peri.Out[4], Peri.In[4],
Link.Out[4], Link.In[4],
Link.Set[4],
Ext.Mem.Set[4],
Ports.To.Mem[6], Ports.From.Mem[6]:

PAR
Processor (Ports.To.Mem[5], Ports.From.Mem[5],
Ext.Mem.Set,
Link.Out, Link.In,
Link.Set,
Peri.Out, Peri.In,
Event.Input)

PAR n = [0 FOR 4]
Link.Cont (Ports.To.Mem[n], Ports.From.Mem[n],
Link.Outputs[n], Link.Inputs[n],
Link.In[n], Link.Out[n],
Link.Set[n])

Mem.Int.Cont (Ports.To.Mem, Ports.From.Mem,
Ext.Mem.Outputs, Ext.Mem.Inputs,
Ext.Mem.Set)

Peri.Int.Cont (Ports.To.Mem[4], Ports.From.Mem[4],
Peri.Outputs, Peri.Inputs,
Peri.Out, Peri.In)

图7. Transputer 的框图及其 Occam 描述 125-7