

利用概率模型对微程序功能的评价

陈道文

(中国科学院自动化研究所)

摘 要

本文重点讨论计算机机器语言级指令执行与微程序操作特性之间的关系，建立微程序操作模型，导出性能评价公式及微指令平均执行时间和微程序处理的平行度两个参数。以一台大规模集成电路微程序计算机为背景，评价流水线控制方式和非流水线控制方式的性能。

计算机性能的一个重要指标是指令执行速度。微程序计算机具有机器语言级和控制存储级两级结构，它的执行时间由两部分构成：

$$T = n(R + D) + \sum_{i=1}^n \sum_{k=1}^{l_i} (t_f + t_p)$$

单位为基本机器周期。其中

n 为所执行的机器语言指令数； R 为主存读出时间； D 为机器语言指令译码时间； l_i 为解释第 i 条机器语言指令所需的微指令数； t_f 为控制存储器读出周期时间； t_p 为第 i 条机器指令对应的微程序第 k 条微指令周期。

从上式可看出，公式右边第一项由机器指令条数决定；第二项为微程序执行时间，可以通过优化微程序，提高微程序执行速度来提高整机性能。本文讨论单处理机、单存储系统、单总线结构微程序计算机的两个重要性能参数，即平均微指令执行时间及微程序处理的平行度，以及提高性能的方法。

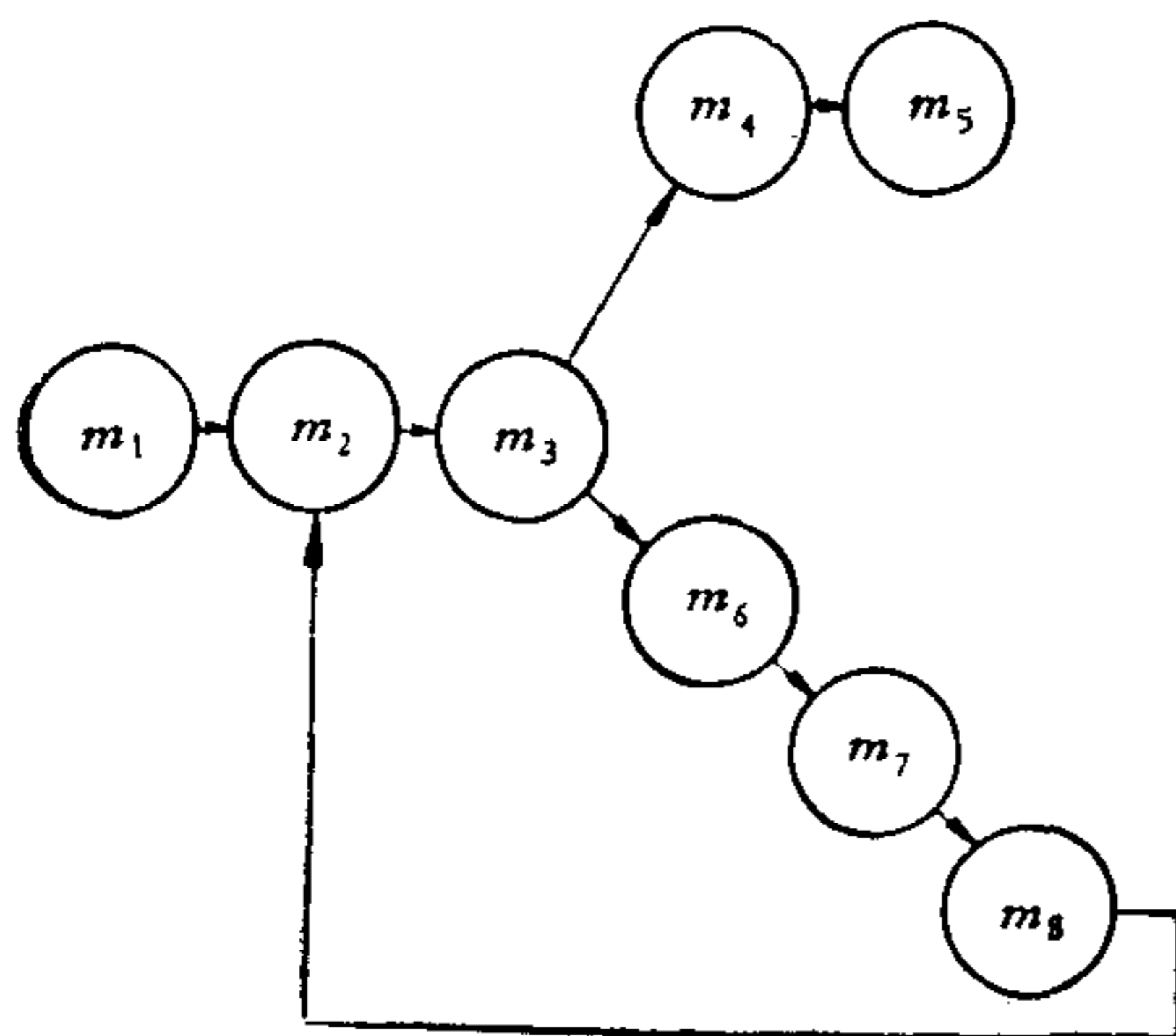


图1 操作模型

一、微程序操作模型及微指令平均执行时间

1. 微程序操作模型

为了推导微程序功能参数，建立微程序操作模型，表示微指令执行次序以及不同微操作的执行，还表示出微程序与机器语言指令之间关系，在图1中用一个节点表示一条微指令，连线箭头表示执行次序以及转移、分支情况。关键点是一个节点需要一个执行周期时间，其中 $r_1 = m_1 m_2 m_3 m_4 m_5$ ， $r_2 = m_1 m_2 m_3 (m_6 m_7 m_8 m_2 m_3)^j m_4 m_5$ 。 r_1 和 r_2 是时序上互斥的微程序。 j 是循环次数，重复执行其中一部分微程序。 m_1, \dots 为微指令。

一个执行周期时间，其中 $r_1 = m_1 m_2 m_3 m_4 m_5$ ， $r_2 = m_1 m_2 m_3 (m_6 m_7 m_8 m_2 m_3)^j m_4 m_5$ 。 r_1 和 r_2 是时序上互斥的微程序。 j 是循环次数，重复执行其中一部分微程序。 m_1, \dots 为微指令。

我们知道在机器语言级实现程序 P 是指令的集合，

$$P = \{I_1, I_2, \dots, I_n\}$$

I_i 为机器语言指令作为程序的元素。根据单处理机和单存储系统的假定， P 事件 I_i 是完全互斥的。在控制存储级，

$$M_i = \{r_1 r_2 \dots r_i \dots r_q\},$$

$$r_i = \{m_1 m_2 \dots m_j\},$$

M 事件出现也是完全互斥的

$$I_1 \cap I_2 \dots \cap I_n = \Phi$$

$$m_1 \cap m_2 \dots \cap m_j = \Phi$$

由于 P 事件出现而引起控制存储级 M 事件，在某一时间点上，假定研究某 M 事件 m_j ，已知概率 $0 < P(I_i) \leq 1$ 且 $\sum_{i=1}^n P(I_i) = 1$ ，则

$$P(m_j) = \sum_{i=1}^n P(m_j | I_i) P(I_i). \quad (1)$$

其中 $P(m_j | I_i)$ 为条件概率。表示微指令 m_j 在执行机器指令 I_i 时出现的概率，可以由图1操作模型计算出。公式(1)表明由不同机器指令 I_i 可能引起的微指令 m_j 分布在全部 n 条指令中的概率，可以通过操作模型求得。因而操作模型建立了机器语言指令执行与微程序操作之间的关系。

2. 微指令平均执行时间

根据微程序操作模型，计算微指令平均执行时间，引入特性参数 t_i, l_i 。 t_i 表示某一子集 r_i 的周期数， l_i 表示子集 r_i 包含的微指令条数，即 $m_1 m_2 \dots$ ，亦是 r_i 的长度，这里必须强调的是 $m_1 m_2$ 可以是一个同步周期时间或是一个异步周期时间，有限集 M 的时间参数集

$$T = \{t_1 t_2 \dots t_i \dots t_q\},$$

其中 t_i 为对应于 r_i 的时间参数，其概率为 $P(t_i) = P(r_i)$ 而

$$\sum_{i=1}^q P(t_i) t_i = T_E$$

T_E 表示在全部 M 集上的每个子集的平均执行时间，为了推导出每条微指令的平均执行时间，考虑 r_i 的长度 l_i ，比值 $\frac{t_i}{l_i} = t_{im}$ 表示 r_i 的每条微指令平均时间间隔，同样有

$$P\left(\frac{t_i}{l_i}\right) = P(r_i)$$

而

$$\sum_{i=1}^q P\left(\frac{t_i}{l_i}\right) t_{im} = \sum_{i=1}^q P(r_i) \frac{t_i}{l_i} = T_M$$

T_M 为微指令平均执行时间，是与 M 集 r_i 出现概率有关，分布在全部机器语言指令频率上。

二、微程序处理的平行度

水平微程序主要特点是能够同时利用机内多个资源，并行执行多个微操作。在一台微程序计算机内，计算资源是属于基本操作部件，接受微操作控制。按性质不同可以把它

们分成两大类即存储资源和数据通路资源,存储资源如存储器、寄存器、指示器及暂存器等等.数据通路资源是数据流通的计算资源,像数据总线、指令总线、全加器、求补器等都属于这种类型的资源.全部微程序计算资源集合记作 R ,存储资源集合记作 S ,数据通路资源集合记作 H ,有 $R = S \cup H$.

通常完成一个微操作是将被操作数据从一个存储部件经由数据通路传送到另一个存储部件,如果两个或更多的微操作企图同时传送数据,经过相同的数据通路或送数据到同一目标存储资源时,就不可避免地产生了资源竞争,产生资源竞争的两个微操作称为互斥,反之称为相容.因此,必须研究微操作数据通路和目标存储资源利用情况,以检测微程序资源竞争及互斥微操作.

建立资源分配矩阵如图 2,矩阵中每一个非零项 R_{ij} 指出第 i 个微操作对应第 j 个资源,且第 i 个微操作存在不影响第 $i+1$ 个微操作的出现.由图 2 可知 R_{22} 与 R_{32} 需要共享目标资源 R_2 ,产生资源竞争,而 R_{22} , R_{44} 不存在资源竞争,故 μOP_2 与 μOP_4 可以兼容.必须注意到,操作的同时性不仅仅取决于硬件资源结构,也反映了控制字形式,例如两个微操作同时需要常数,而在只有一个微常数字段情况下,两个微操作常数必须是相同的,否则就不能放在同一条微指令里.

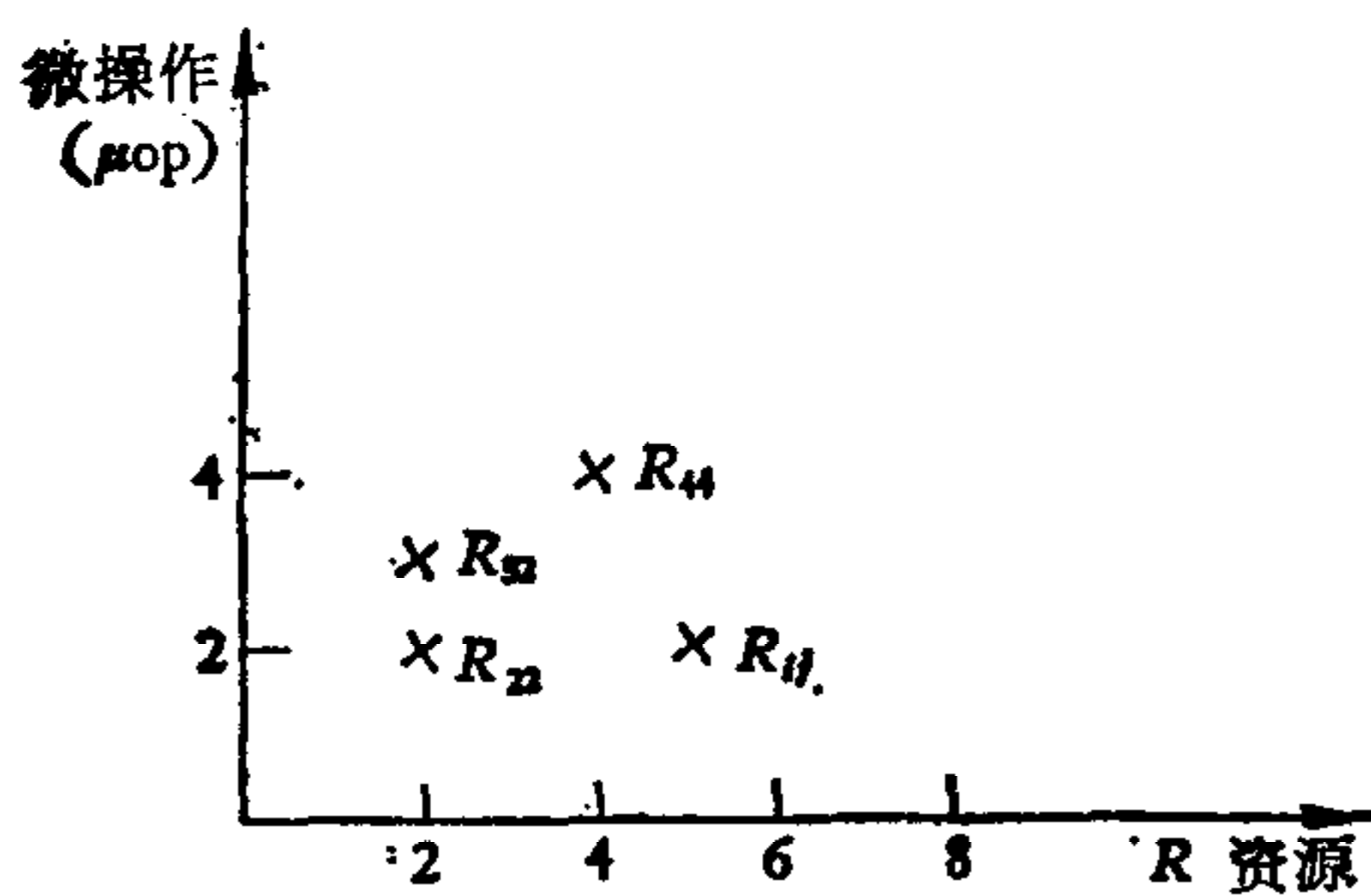


图 2 资源分配矩阵

问题的另一方面是:每条微指令平均使用多少个计算资源?是接近最大可用资源数呢?还是远远小于最大可用资源数?这可以用微指令的平行度参数来评价.

假定微控制字包含 R 个控制字段,在直接控制结构中,每个字段是单独一位控制一个存储资源或选择一个数据通路,在最小编码结构中,每个字段为 b_i 位,可控制 $2b_i$ 个互斥资源.不论哪种控制方式,如果无时间冲突, R 个操作可同时进行,即可能执行的平行度为 R .

回顾第一节已引入的特性参数 t_i , l_i , 现再引入 q_i , q_i 表示 r_i 执行的全部微操作数,则 $C_p^i = q_i/t_i$ 说明 M 事件 r_i 的每个周期的平均平行度,则 M 集的每个周期的平均平行度为

$$C_p = \sum_{i=1}^q P(r_i) C_p^i = \sum_{i=1}^q P(r_i) \cdot \frac{q_i}{t_i}$$

而用平行度利用系数 $PA = C_p/R$ 说明微控制字结构的有效利用率.

$$PA = \frac{C_p}{R} = \frac{\sum_{i=1}^q P(r_i) \frac{q_i}{t_i}}{R}$$

由此看出,平均平行度也是建立在全部 M 事件分布区间之中.

三、微程序性能的改善方法

上面讨论了水平微程序的两个特性参数,即平均微指令执行时间 T_M 和平均平行度

C_p . 为了改善微程序功能,再进一步分析子集 r_i 的执行时间,一般由两部分时间构成,取微指令时间 t_f 和微指令执行时间 t_p ,即 $t_i = t_f + t_p$,利用微程序流水线技术,使取指令与执行微指令时间重叠^[5],则 $t_i = t_p$. 相应平均微指令执行时间(流水线技术):

$$T_{M0} = \sum_{i=1}^q P(r_i) \frac{t_p}{l_i}$$

平均平行度(流水线技术):

$$C_{p0} = \sum_{i=1}^q P(r_i) \frac{q_i}{t_p}$$

由于采用流水线重叠技术使微程序功能得到改善,速度提高 T_M/T_{M0} 倍,平行度利用系数增长 $PA_0/PA = C_{p0}/C_p$ 倍.

由图 1,如果 m_1 为访问主存的微指令,为一个异步周期相当于 t_B 个同步周期. 设系统以单总线方式连接处理机、主存和其它外部设备,则当 CPU 访问主存时,需要先取得总线使用权. 假定只考虑高速通道以插入方式占有单总线,插入异步周期相当于 t_{sl} 个同步周期,一次占有总线为 t_B 个同步周期. 则 CPU 等待单总线空闲所需平均等待周期

$$T_W = \frac{t_B}{t_{sl}} \cdot \frac{t_B}{2}$$

则 m_1 微指令所需执行周期数

$$t_1 = t_B + T_W = t_B + \frac{t_B^2}{2t_{sl}}$$

在同步执行时序中,一般 $\beta t_p = [t_1] + 1, [t_1]$ 为整数 t_1 . 而异步执行时序完全由应答关系决定,所以周期较短,从而提高了微指令功能.

一般微程序会由于转移带来速度损失,而这里讨论的水平微程序转移策略,由前条微指令建立转移条件,而本条微指令检测判断,决定后继微指令,因而一般讲转移并不影响微程序速度.

四、评价一台中规模集成电路微程序计算机的功能

利用上面操作模型及计算公式,现对一台中规模集成电路多功能微程序计算机的性能进行定量评价. 为了简化讨论,这里只讨论 4 条定点运算指令加、减、乘、除.

该机为水平型编码控制字,每个控制字的控制字段数 $R = 9$. 采用流水线控制方式,不计取微指令时间,微指令同步周期时间为 400 ns. 机器语言指令字长 16 位. 总线传送一次数据时间 $1.2 \mu s$,插入周期时间为 $6 \mu s$.

指令集合 P ,

$$P = \{I_1 I_2 I_3 I_4\} = \{I_+ I_- I_\times I_\div\}$$

假定四类指令出现概率为

$$P(P) = \{P(I_+), P(I_-), P(I_\times), P(I_\div)\} \\ = \{0.4, 0.3, 0.2, 0.1\}$$

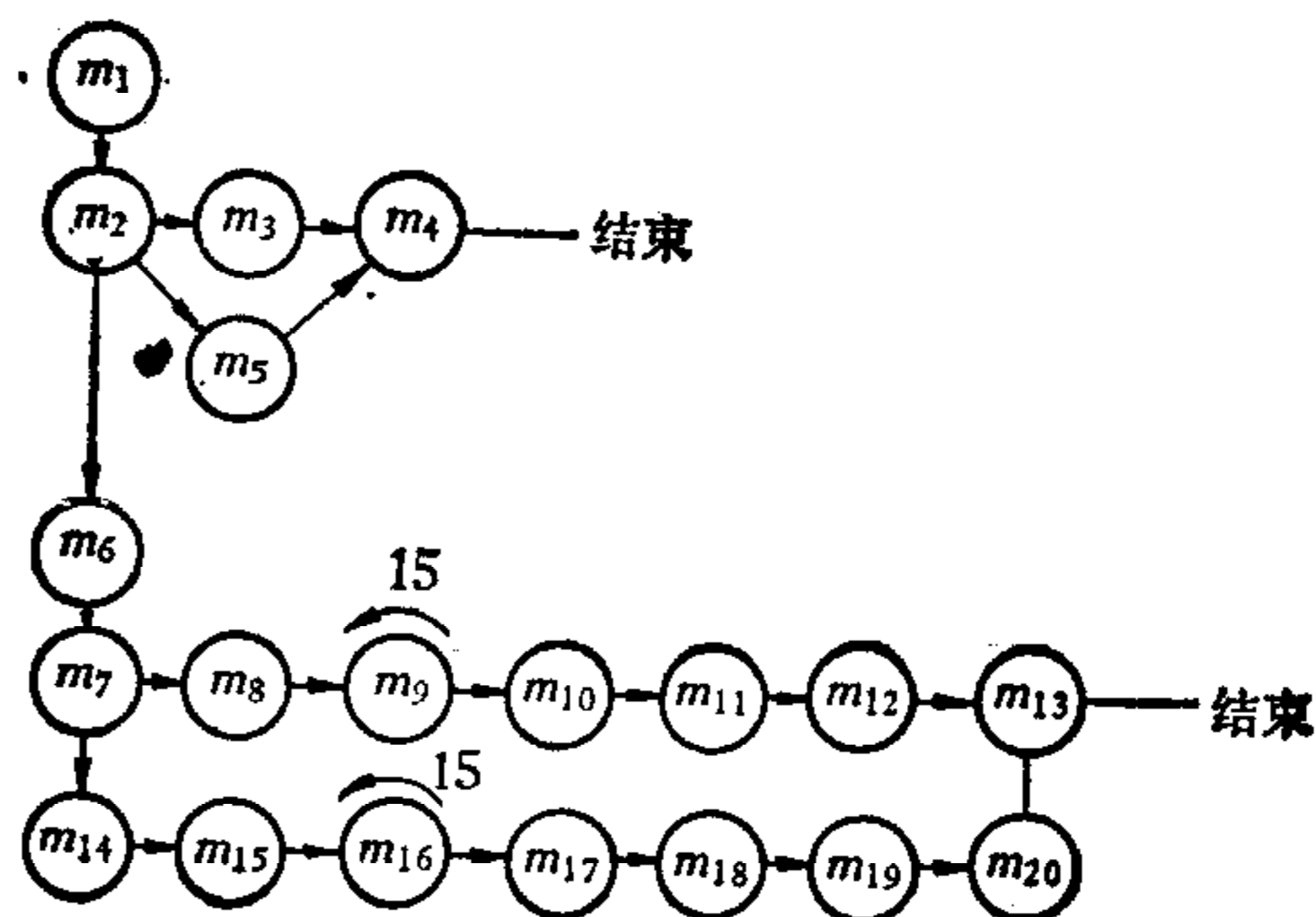


图 3 微程序模型

利用图 3 的模型写出四个微程序

$$r_1 = \{m_1 m_2 m_3 m_4\},$$

$$r_2 = \{m_1 m_2 m_5 m_4\},$$

$$r_3 = \{m_1 m_2 m_6 m_7 m_8 (m_9)^{15} m_{10} m_{11} m_{12} m_{13}\},$$

$$r_4 = \{m_1 m_2 m_6 m_7 m_{14} m_{15} (m_{16})^{15} m_{17} m_{18} m_{19} m_{20} m_{13}\}.$$

其中 m_1 为取机器语言指令的微指令, m_2 为取源操作数的微指令, $m_1 m_2$ 是公用微指令.

表 1 列出每条微指令对应的周期数(指同步周期数)及使用资源数.

表 1

微指令名称		m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	$\overset{15}{\curvearrowright} m_9$	m_{10}
资源数 q_i		9	8	8	6	8	6	8	6	8	7
周期数	流水线	$t_1 = 3.3$	1	1	1	1	1	1	1	1	1
	非流水线	$t'_1 = 4.2$	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4

续表 1

微指令名称		m_{11}	m_{12}	m_{13}	m_{14}	$\overset{15}{\curvearrowright} m_{15}$	m_{16}	m_{17}	m_{18}	m_{19}	m_{20}
资源数 q_i		3	6	6	6	5	8	7	7	2	6
周期数	流水线	1	1	1	1	1	1	1	1	1	1
	非流水线	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4	7.5/4

表中每条微指令, 同步时序占用一个基本执行周期 t_p , 异步时序则占用 t_1 个基本执行周期. 节点 m_9 和 m_{16} 最大 15 次循环.

先计算异步周期 t_1 , 已知该机占用一次总线为 $1.2 \mu s$, 所以 $t_B = 3t_p$. 插入周期时间 $6 \mu s$, 所以 $t_{sl} = 5t_B$ 则对于流水线方式有:

$$t_1 = t_B + \frac{t_B^2}{2t_{sl}} = 3.3t_p.$$

对于非流水线方式, 由于取微指令而增加的周期数为 $3.5/4 t_p^{[5]}$, 所以非流水线方式基本周期为 $7.5/4 t_p$, 而异步周期

$$t'_1 = t_1 + \frac{3.5}{4} t_p = 4.2t_p.$$

利用上式计算出 r 子集参数见表 2(流水线).

表 2

参数 子集	t_i 总周期数	l_i 微指令数	q_i 微操作数	$P(r_i)$ 概率
r_1	6.3	4	31	0.4
r_2	6.3	4	31	0.3
r_3	26.3	24	179	0.2
r_4	28.3	26	190	0.1

计算流水线方式功能参数

1) 平均微指令执行时间 T_{M0}

$$T_{M0} = \sum_{i=1}^4 P(r_i) \frac{t_i}{l_i} = 0.4 \left(\frac{6.3}{4} \right) + 0.3 \left(\frac{6.3}{4} \right) + 0.2 \left(\frac{26.3}{24} \right) + 0.1 \left(\frac{28.3}{26} \right) \\ = 1.43(\text{周期}) = 572(\text{ns}).$$

2) 平均微指令资源 C_{p0}

$$C_{p0} = \sum_{i=1}^4 P(r_i) \frac{q_i}{t_i} \\ = 0.4 \left(\frac{31}{6.3} \right) + 0.3 \left(\frac{31}{6.3} \right) + 0.2 \left(\frac{179}{26.3} \right) + 0.1 \left(\frac{190}{28.3} \right) \\ = 5.48(\text{微操作/周期}).$$

3) 平行度 PA_0

$$PA_0 = \frac{C_{p0}}{R} = \frac{5.48}{9} = 61\%.$$

非流水线 r 子集参数见表 3.

表 3

子集 \ 参数	t_i 总周期数	l_i 微指令数	q_i 微操作数	$P(r_i)$ 概率
r_1	9.8	4	31	0.4
r_2	9.8	4	31	0.3
r_3	47.3	24	179	0.2
r_4	51.0	26	190	0.1

计算非流水线功能参数

1) 平均微指令执行时间 T_M

$$T_M = \sum_{i=1}^4 P(r_i) \frac{t_i}{l_i} \\ = 0.4 \left(\frac{9.8}{4} \right) + 0.3 \left(\frac{9.8}{4} \right) + 0.2 \left(\frac{47.3}{24} \right) + 0.1 \left(\frac{51.0}{26} \right) \\ = 2.30(\text{周期}) = 920(\text{ns}).$$

2) 平均微指令资源 C_p

$$C_p = \sum_{i=1}^4 P(r_i) \frac{q_i}{t_i} = 0.4 \left(\frac{31}{9.8} \right) + 0.3 \left(\frac{31}{9.8} \right) \\ + 0.2 \left(\frac{179}{47.3} \right) + 0.1 \left(\frac{190}{51.0} \right) = 3.35(\text{微操作/周期}).$$

3) 平行度 PA

$$PA = \frac{C_p}{R} = \frac{3.35}{9} = 37.2\%.$$

所以流水线方式平均资源利用率较非流水线方式提高 $PA_0/PA = 5.48/3.35 \doteq 1.64$

倍;平均每条微指令速度提高 $T_M/T_{M0} = 2.3/1.43 \approx 1.64$ 倍. 显然该机在使用流水线技术后在性能上有较大提高.

五、结 束 语

以上论述了如何通过资源分配矩阵调度分配资源,采用并行技术提高系统 PA 值,减少微程序执行时间.因此利用概率模型推导出的性能评价公式可以用于指导微程序设计.微控制字的安排,应对于出现频率较高的公共微程序力求取得较佳性能,而对出现频率较低微程序可以适当放低要求.由于动态微程序重新定义体系结构的功能评价与微程序处理过程都有一个共同功能参数即执行速度,因此本文的方法亦适用于评价重新定义系统结构的动态微程序中解释性微程序的功能.

参 考 文 献

- [1] Rauscher T. G., Agrawala A. K., Dynamic Problem—Oriented Redefinition of Computer Architecture via Microprogramming, *IEEE Trans. on Computer*, C-27 (1978), No. 11.
- [2] Foster C. C., Gonter R. H., Risemen E. M., Measure of Op-Code Utilization, *IEEE Trans. on Computer*, C-20 (1971), 582—584.
- [3] Tarter J., Dasgupta S., A Probabilistic Model for the Evaluation of Microprogram Performance, The Seventh Annual Workshop on Microprogramming, (1974).
- [4] Tauchiys M., Gonzales M. J., An Approach to Optimization of Horizontal Microprograms, The Seventh Annual Workshop on Microprogramming, (1974).
- [5] 陈道文, 一台中规模集成电路计算机的微程序设计, 电子计算机动态, (1980.5).
- [6] Courtois B., On Balancing Hardware Firmware for Designing a Fault-Tolerant Computers Series, The 8th Annual Workshop on Microprogramming, (1975).

EVALUATING MICROPROGRAM PERFORMANCE WITH A PROBABILITY MODEL

CHEN DAOWEN

(Institute of Automation, Academia Sinica)

ABSTRACT

This paper mainly discusses the relationship between the instruction execution at machine language level and the operation behaviour at microprogram level of a computer. Presenting a microprogram model, formulae of performance evaluation and performance parameters, mean microinstruction execution time and the parallism of microprogram processing are derived. With the background of our microprogram computer made of medium scale integrated (MSI), performance parameters of both pipe line and non-pipe line organization are given.