

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁶

G06F 7/50

[12] 发明专利申请公开说明书

[21] 申请号 97198461.1

[43]公开日 1999 年 10 月 20 日

[11]公开号 CN 1232561A

[22]申请日 97.7.4 [21]申请号 97198461.1

[30]优先权

[32]96.10.2 [33]GB [31]9620526.5

[86]国际申请 PCT/GB97/01812 97.7.4

[87]国际公布 WO98/14864 英 98.4.9

[85]进入国家阶段日期 99.4.1

[71]申请人 ARM 有限公司

地址 英国剑桥郡

[72]发明人 刘鉴伟

[74]专利代理机构 中国专利代理(香港)有限公司

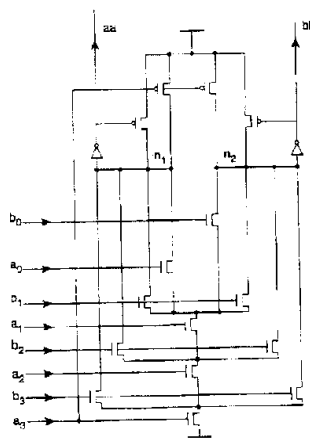
代理人 王 勇 王 岳

权利要求书 3 页 说明书 13 页 附图页数 14 页

[54]发明名称 数字加法器电路

[57]摘要

二进制加法器电路包括用两个信号值(V,W)编码进位生成控制信号的进位 计算电路,使得 $V=W=0$ 表示进位消除, $V=W=1$ 表示进位生成及 $V \neq W$ 表示进位传播。进位计算电路可用静态或动态 CMOS 逻辑实现。



ISSN 1008-4274

权 利 要 求 书

1. 一种用于相加第一二进制数与第二二进制数的加法器电路, 所述加法器电路包括:

5 进位计算电路, 用于计算表示所述第一二进制数与所述第二二进制数的对应位块及对所述块的输入进位值之和的进位生成控制信号, 所述进位生成控制信号包括各可具有值 P 或 Q 之一的两个信号 V 与 W, 所述进位生成控制信号按照下述关系从所述和中编码进位结果:

10 $V = W = P$ 表示进位消除, 从而所述进位结果为与所述输入进位值无关的零;

$V \neq W$ 表示进位传播, 从而所述进位结果等于所述输入进位值; 以及

$V = W = Q$ 表示进位生成, 从而所述进位结果为与所述输入进位值无关的 1。

15 2. 权利要求 1 中所要求的加法器电路, 其中 $P = 0$ 及 $Q = 1$ 。

3. 权利要求 1 与 2 中任何一项中所要求的加法器电路, 其中所述进位计算电路响应包括所述第一二进制数与所述第二二进制数的两个对应位对之一及两个前面计算的进位生成控制信号的两对输入信号 (a_1, b_1) 与 (a_2, b_2) , 以及 V 与 W 给出如下:

20 $V = a_2 \cdot b_2 + (a_2 + b_2) \cdot a_1$; 以及

$W = a_2 \cdot b_2 + (a_2 + b_2) \cdot b_1$ 。

4. 权利要求 1 与 2 中任何一项中所要求的加法器电路, 其中所述进位计算电路响应包括所述第一二进制数与所述第二二进制数的三个对应位对之一及三个前面计算的进位生成控制信号的三对输入信号 (a_1, b_1) 、 (a_2, b_2) 与 (a_3, b_3) , 以及 V 与 W 给出如下:

25 $V = a_3 \cdot b_3 + (a_3 + b_3) (a_2 \cdot b_2 + (a_2 + b_2) \cdot a_1)$; 以及

$W = a_3 \cdot b_3 + (a_3 + b_3) (a_2 \cdot b_2 + (a_2 + b_2) \cdot b_1)$ 。

5. 权利要求 1 与 2 中任何一项中所要求的加法器电路, 其中所述进位计算电路响应包括所述第一二进制数与所述第二二进制数的四个对应位对之一及四个前面计算的进位生成控制信号的四对输入信号 (a_1, b_1) 、 (a_2, b_2) 、 (a_3, b_3) 及 (a_4, b_4) , 以及 V 与 W 给出如下:



$$V = a_4 \cdot b_4 + (a_4 + b_4)(a_3 \cdot b_3 + (a_3 + b_3)(a_2 \cdot b_2 + (a_2 + b_2) \cdot a_1));$$

以及

$$W = a_4 \cdot b_4 + (a_4 + b_4)(a_3 \cdot b_3 + (a_3 + b_3)(a_2 \cdot b_2 + (a_2 + b_2) \cdot b_1)).$$

6. 权利要求 1 与 2 中任何一项中所要求的加法器电路, 其中所述
5 进位计算电路响应包括所述第一二进制数与所述第二二进制数的 N 个
对应位对之一及 N 个前面计算的进位生成控制信号的 N 对输入信号
(a_1, b_1)、(a_2, b_2)、 $\dots$ (a_N, b_N), 以及 V 与 W 按照下述关系给
出:

$$V = f_V^N(a_N, b_N, \dots a_1, b_1); \text{ 及}$$

10 $W = f_W^N(a_N, b_N, \dots a_1, b_1); \text{ 其中}$

$$f_V^1 = a_1;$$

$$f_W^1 = b_1; \text{ 而对于 } 2 \text{ 与 } N \text{ 之间的 } i;$$

$$f_V^i(a_i, b_i, \dots a_1, b_1) = a_i \cdot b_i + (a_i + b_i) \cdot f_V^{i-1}; \text{ 以及}$$

$$f_W^i(a_i, b_i, \dots a_1, b_1) = a_i \cdot b_i + (a_i + b_i) \cdot f_W^{i-1}.$$

15 7. 前面的权利要求中任何一项中所要求的加法器电路, 其中在并
行的前置结构中使用多个所述进位计算电路从所述第一二进制数与
所述第二二进制数计算全组进位位。

8. 前面的权利要求中任何一项中所要求的加法器, 其中所述进位
计算电路由多个静态 CMOS 逻辑门构成。

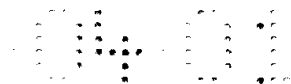
20 9. 权利要求 1 至 7 中任何一项中所要求的加法器, 其中所述进位
计算电路由多个动态 CMOS 逻辑门构成。

10. 前面的权利要求中任何一项中所要求的加法器电路, 包括:

25 进位二进制数确定电路, 响应所述第一二进制数与所述第二二进
制数确定由所述第一二进制数与所述第二二进制数之和的进位位组
成的进位二进制数, 所述进位二进制数确定电路具有进行串行操作来
确定所述进位二进制数的多个电路级, 各电路级用于部分地解决所述
进位二进制数, 及至少一个电路级包含至少所述进位位计算电路之
一, 所述进位位计算电路生成在所述电路级之间传递的作为对下一电
路级的输入信号的进位控制生成信号; 以及

30 组合逻辑电路, 响应所述第一二进制数、所述第二二进制数及所
述进位二进制数的各自的对应位以生成结果二进制数的对应位。

11. 权利要求 10 中所要求的加法器电路, 其中对于所述第一二进



制数、所述第二二进制数及所述进位二进制数的对应位 A、B 及 C，所述组合逻辑电路执行运算 $((A \text{ XOR } B) \text{ XOR } C)$ 。

12. 前面的权利要求中任何一项中所要求的加法器电路，其中对于进位计算电路，所述进位控制生成信号为之必须表示进位消除或进位生成之一， $V = W$ ，及可利用单一信号线来传输所述进位控制生成信号。

13. 前面的权利要求中任何一项中所要求的加法器电路，其中所述进位计算电路具有一个进位输入及生成一个进位输出，使得当 $V \neq W$ 时，以 1 作为进位输入则 V 等于进位输出，而以 0 作为进位输入则 W 等于进位输出。

14. 权利要求 13 中所要求的加法器电路，其中所述进位计算电路与进位选择电路组合来生成最终结果数。

15. 一种包含前面的权利要求中任何一项中所要求的加法器电路的集成电路微处理器。

16. 一种操作用于相加第一二进制数及第二二进制数的加法器电路的方法，所述方法包括下述步骤：

计算表示所述第一二进制数与所述第二二进制数的对应位块及对所述块的输入进位值之和的进位生成控制信号，所述进位生成控制信号包括各能具有值 P 或 Q 之一的两个信号 V 与 W，所述进位生成控制信号按照下述关系从所述和中编码进位结果：

$V = W = P$ 表示进位消除，从而所述进位结果为与所述输入进位值无关的零；

$V \neq W$ 表示进位传播，从而所述进位结果等于所述输入进位值；以及

$V = W = Q$ 表示进位生成，从而所述进位结果为与所述输入进位值无关的 1。

说明书

数字加法器电路

5 本发明涉及数据处理领域。更具体地，本发明涉及数据处理系统内使用的数字加法器电路。

加法是频繁地在数据处理系统内执行的最重要的算术运算之一。生产高速加法器电路的问题是结果的高位取决于来自低位的进位值。这一后果使加法运算势必相对地慢。数据处理系统内的永恒目标在于运算应当尽可能地快，并为此目的已历经多年在设计与研制能快速运算的加法器电路上花费了可观的精力。

在以高速运算的努力中已提出过这样的技术，其中两个输入运算数之和的进位是与在这两个输入运算数上所执行的异或运算分开计算的，而在最终阶段将结果组合以生成和。

15 这一技术的实例为 R P Brent 与 H T Kung 的论文“并行加法器的常规配置” IEEE Trans, Comput, 卷 31, 260 至 264 页, 1982 年 3 月, 中所描述的 Brent 与 Kung 加法器。

虽然这种进位计算方案改进了性能，但进位计算需要多层逻辑来执行。这一逻辑消耗电路面积与功率并且限制了能达到的最终性能。

20 从一个方面看，本发明提供了用于相加第一二进制数与第二二进制数的加法器电路，所述加法器电路包括：

进位计算电路，用于计算表示所述第一二进制数与所述第二二进制数的一块对应位及对所述块的输入进位值之和的进位生成控制信号，所述进位生成控制信号包括各可具有 P 或 Q 值的两个信号 V 与 W，所述进位生成控制信号按下述关系从所述和中编码进位结果：

25 $V = W = P$ 表示进位消除，从而所述进位结果为与所述输入进位值无关的零；

$V \neq W$ 表示进位传播，从而所述进位结果等于所述输入进位值；以及

30 $V = W = Q$ 表示进位生成，从而所述进位结果为与所述输入进位值无关的一。

本发明提供能以较少电路元件与较少逻辑层得出更快更小及更节省功率的电路生成进位生成控制信号的编码。应理解两个信号 V 与

W 能取的值 P 与 Q 可具有各种绝对值及可以以正或反形式。然而，在有利地简单的较佳实施例中，值 P 与 Q 在电压干线上取系统的二进制值，其中 $P = 0$ 及 $Q = 1$ 。

5 执行上述编码的进位计算电路通常在整个加法器电路中出现许多次。取决于特定进位计算电路在加法器电路中的位置，它作为其输入接收来自第一二进制数与第二二进制数的位或，或者来自电路中的上游的包含前面计算的进位生成控制信号的位。更具体地，在本发明的较佳实施例中，所述进位计算电路响应包含所述第一二进制数与所述第二二进制数的两个相应的位对之一或两个前面计算的进位生成控制信号的两对输入信号 (a_1, b_1) 与 (a_2, b_2) ，并且 V 与 W 由下式给出：

$$V = a_2 \cdot b_2 + (a_2 + b_2) \cdot a_1; \text{ 及}$$

$$W = a_2 \cdot b_2 + (a_2 + b_2) \cdot b_1; \text{ 及。}$$

15 在进位计算电路能在一次运算中解决的输入信号对的数目与达到该计算的必要电路的规模、复杂性及实际性之间存在着要解决的平衡。更具体地，如果进位计算电路能解决更多的输入信号对，则整个加法器电路中需要较少的进位计算电路层。然而，随着对进位计算电路的输入信号对的数目的提高，该进位计算电路所执行的判定的复杂性随着电路规模及其在集成电路的有限电压与串联损耗限制内的操作能力的增加而增加。

20 由于上述原因，已发现响应二、三或四对输入信号之一的进位计算电路最好。然而，应理解，一般地说进位计算电路可响应 N 对输入信号（其中 N 为大于 1 的整数）。

25 虽然可将上述进位计算电路用在多种多样的范围中，但特别适用于这样的系统，其中在并行的前置结构中利用多个所述进位计算电路来计算来自所述第一二进制数及所述第二二进制数的整进位位组。

进位计算电路所执行的编码是非常适合于在多个静态 CMOS 逻辑门中高效实现的编码。

30 作为替代，如果为了缩小电路面积而希望用动态集成电路，则编码也适用于多个动态 CMOS 逻辑门的实施例。

可在其中采用上述技术的较佳加法器电路结构包括：

进位二进制数确定电路，其响应所述第一二进制数及所述第二二

进制数确定由所述第一二进制数及所述第二二进制数之和的进位所构成的进位二进制数，所述进位二进制数确定电路具有串行操作来确定所述进位二进制数的多个电路级，各电路级用于部分地解决所述进位二进制数及至少一个电路级包含至少一个所述进位位计算电路，后者生成在所述电路级之间传递的作为对下一电路级的输入信号的进位控制生成信号；以及

组合逻辑电路，其响应所述第一二进制数、所述第二二进制数及所述进位二进制数的各自的对应位以生成结果二进制数的对应位。

在上述电路中，可作出进位二进制数的有利的快的确定。一旦确定了这一进位二进制数，便能在实施例中分别为所述第一二进制数、所述第二二进制数及所述进位二进制数的对应位 A、B 与 C 确定最终的和数，所述组合逻辑电路执行运算 $((A \text{ XOR } B) \text{ XOR } C)$ 。

有助于缩小所需总电路面积的进一步改进为其中对于进位计算电路（所述进位控制生成信号必须为之表示进位消除或进位生成之一， $V = W$ ）及可用单一信号线来传输所述进位控制生成信号。

本发明的加法器电路特别适用于在集成电路微处理器内使用。

从另一方面看，本发明提供了操作用于相加第一二进制数与第二二进制数的加法器电路的方法，所述方法包括下述步骤：

计算表示所述第一二进制数与所述第二二进制数的一个对应位块及对所述块的输入进位值之和的进位生成控制信号，所述进位生成控制信号包括各能具有值 P 或 Q 的两个信号 V 与 W，所述进位生成控制信号按下述关系从所述和编码进位结果：

$V = W = P$ 表示进位消除，从而所述进位结果为与所述输入进位值无关的零；

$V \neq W$ 表示进位传播，从而所述进位结果等于所述输入进位值；以及

$V = W = Q$ 表示进位生成，从而所述进位结果为与所述输入进位值无关的 1。

下面参照附图只是以示例方式描述本发明的实施例，附图中：

图 1 示出两输入对进位计算电路；

图 2 示出包含图 1 的进位计算电路的 16 位进位二进制数计算系统；

图 4 示出包含图 3 的进位计算电路的 9 位进位二进制数计算系统；

图 5 示出图 1 的电路的静态 CMOS 实施例；

5 图 6 示出图 3 的电路的静态 CMOS 实施例；

图 7 示出用来计算 32 位进位二进制数的最高位的四输入对进位计算电路之间的互连；

图 8 对应于图 7，除了计算的是次最高位之外；

图 9 示出 4 输入对进位计算电路；

10 图 10 示出使用 4 输入对进位计算电路的 16 位加法器；

图 11 和 12 示出 4 输入对进位计算电路的 CMOS 电路实现；

图 13 示出传统的加法器电路；以及

图 14 示出包含 4 输入对进位计算电路的加法器电路。

15 首先考虑进位仲裁的一般概念。在一般情况中，进位 C_{i+1} 是通过相加两个 1 位二进制数 a_i 与 b_i 计算的。存在着两种用 a_i 与 b_i 之值定义的一般情况。当两个运算数位相等时出现存在一个输出进位请求的第一种情况。如果两个输入为 1，出现 1 进位请求，而如果两个输入为 0 则出现 0 进位请求。当运算数位具有不同值时，出现不存在输出进位请求的第二种情况。见表 1，其中字母 u 表示无输出进位请求。

a_i, b_i	C_{i+1}
0 0	0
1 1	1
0 1	u
1 0	u

表 1

20

一个输入对 (a_i, b_i) 可以作出也可以不作出进位请求。如果采用两个输入对 (a_i, b_i) 及 (a_j, b_j) ，可同时出现两个进位请求。因此，必须仲裁这两个进位请求。注意 i 与 j 涉及计算中的两个邻接的位（在第一级上）或位块（在以后的级上），从而如果我们在涉及前面所仲裁的三位的块的进位请求之间进行仲裁，则 $i=j+3$ 。图 1 示出 2 输入对进位仲裁器（进位计算电路）。输入对 (a_i, b_i) 能作出不可屏蔽的进位请求（不可屏蔽的含义是输出进位 C_{i+1} 必须总是确认这一请求）。
25

输入对 (a_j, b_j) 能作出可屏蔽的进位请求（可屏蔽的含义是不可屏蔽的进位请求可以屏蔽这一请求）。只有在没有来自输入对 (a_i, b_i) 的不可屏蔽的进位请求时，输出进位 C_{i+1} 才确认来自输入对 (a_j, b_j) 的可屏蔽进位请求。这示出在表 2 中。

a_i, b_i	a_j, b_j	C_{i+1}
0 0	- -	0
1 1	- -	1
01 (或 1 0)	0 0	0
01 (或 1 0)	1 1	1
01 (或 1 0)	01 (或 1 0)	u

表 2

输出进位 C_{i+1} 能用两条线 (V_i, W_i) 编码，如表 3 所示。这两条线上的信号构成进位生成控制信号。下式满足表 2 与 3

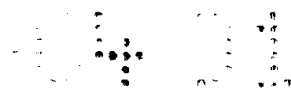
$$\begin{aligned} v_i &= a_i b_i + (a_i + b_i) a_j \\ w_i &= a_i b_i + (a_i + b_i) b_j \end{aligned} \tag{1}$$

C_{i+1}	v_i, w_i
0	0 0
1	1 1
u	0 1
u	1 0

表 3

图 2 中示出利用 2 输入对进位仲裁器的 16 位快速进位计算，图中示出该方案是规则的。图 2 中各节点为 2 输入位进位仲裁器。根据其输入，考虑各节点“表决”要传递给电路中下一级的结果，节点能指示一个进位生成 $(1, 1)$ （赞成）、进位消除 $(0, 0)$ （反对）、或进位传播 $(0, 1)$ 或 $(1, 0)$ （弃权）。在最低级上，这一仲裁是在输入运算数的位间作出的，而在较高级上则在前面确定的仲裁结果之间作出。

图 2 的系统与上面提及的 Brent 与 Kung 提出的方案不同。本方案中不需要 Brent 与 Kung 加法器中 g_i （生成信号）与 P_i （传播信号）



所需的计算逻辑。这导致需要较少的逻辑层并因而高速生成进位。

在 Brent 与 Kung 方案中可将从输入对 (a_i, b_i) 与 (a_j, b_j) 生成的两对 (g_i, p_i) 与 (g_j, p_j) 看作新输入对。新输入对 (g_i, p_i) 在 g_j 与 p_j 都为 0 时作出 0 进位请求, 在 g_i 为 1 时作出 1 进位请求, 而在 p_i 为 1 时无进位请求。注意 g_i 与 p_i 是相互排斥的, 如表 4 中所示。

g_i, p_i	g_j, p_j	c_{i+j}
0 0	- -	0
1 0	- -	1
0 1	0 0	0
0 1	1 0	1
0 1	0 1	u

表 4

输出进位 c_{i+j} 可利用两条线 (v_i, w_i) 编码, 如表 5 中所示。下式满足表 4 与 5。

$$\begin{aligned} v_i &= g_i + p_i g_j \\ w_i &= p_i p_j \end{aligned} \quad (2)$$

c_{i+j}	v_i, w_i
0	0 0
1	1 0
u	0 1

表 5

上面的式 (2) 是著名的 Brent 与 Kung 加法器的关键概念。除了理解如何生成与传播进位以外, 进位生成 g_i 与进位传播 p_i 的逻辑计算是浪费的。

图 3 中示出按照本发明的 3 输入对进位仲裁器。输入对 (a_i, b_i) 能作出不可屏蔽的进位请求。输入对 (a_j, b_j) 与 (a_k, b_k) 能同时作出可屏蔽的进位请求。然而, 输入对 (a_j, b_j) 具有比输入对 (a_k, b_k) 高

的优先级。只在没有来自输入对 (a_i, b_i) 的不可屏蔽进位请求及没有来自输入对 (a_j, b_j) 的可屏蔽进位时，输出进位 C_{i+1} 才确认来自输入对 (a_k, b_k) 的可屏蔽进位请求，如表 6 中所示。下式满足表 3 与 6。

$$\begin{aligned} v_i &= a_i b_i + (a_i + b_i)(a_j b_j + (a_j + b_j)a_k) \\ w_i &= a_i b_i + (a_i + b_i)(a_j b_j + (a_j + b_j)b_k) \end{aligned} \quad (3)$$

a_i, b_i	a_j, b_j	a_k, b_k	C_{i+1}
0 0	- -	- -	0
1 1	- -	- -	1
0 1 (或 1 0)	0 0	- -	0
0 1 (或 1 0)	1 1	- -	1
0 1 (或 1 0)	0 1 (或 1 0)	0 0	0
0 1 (或 1 0)	0 1 (或 1 0)	1 1	1
0 1 (或 1 0)	0 1 (或 1 0)	0 1 (或 1 0)	u

表 6

图 4 示出利用 3 输入对进位仲裁器的 9 位进位计算，它只有两层逻辑并因而导致高速生成进位。可在与 $O(\log_3 n)$ 成比例的时间中用 3 输入对进位仲裁器执行 n 位二进制数加法，因此与利用 2 输入对进位仲裁器计算时间为 $O(\log_2 n)$ 的相比，效率更高。

采用类似于 2 或 3 输入对进位仲裁器的方法，能导出带有任何数目输入对的进位仲裁器。然而，通常对带有 4 个以上输入对的进位仲裁器不感兴趣。首先，需要太多的串联晶体管来实现这些仲裁器，它导致低效的 CMOS 设计。第二，仲裁器单元布局变得对数据通路的位片太大。

图 5 示出 2 输入对进位仲裁器的静态 CMOS 实现。注意输出 V_i 与 W_i 为互补的信号。然而，仲裁器相当对称并且在逆逻辑中实现下一级是直观的。通过两个仲裁器的信号自然是正为真的，因此不需要反相器。

图 6 示出 3 输入对进位仲裁器的静态 CMOS 实现。如果采用动态 CMOS 技术，3 或 4 输入对进位仲裁器可能是有利的，在这一情况中可采用图 3 的电路的上拉或下拉部分。通过以类似于图 5 与 6 之间的扩展的方式对称地扩展级，可达到扩展图 6 的 3 输入对电路以产生 4 输

入对或更高的电路。

通过考虑基于 2 输入对仲裁器的 n 位加法器作为实例，可在形式上进行设计验证。令 $a_n, a_{n-1}, \dots, a_1$ 及 $b_n, b_{n-1}, \dots, b_1$ 为不带进位 $C_n, C_{n-1}, \dots, C_1$ 的 n 位二进制数，并令 C_0 为输入进位位。将运算符 $\circ[1]$

5 定义如下：

$$(a, b) \circ (a', b') = (ab + (a+b)a', ab + (a+b)b') \quad (4)$$

定理 11 (定理)：令

$$(V_i, W_i) = (C_i, C_i) \text{ 如果 } i = 1, \text{ 及}$$

$$(V_i, W_i) = (a_i, b_i) \circ (V_{i-1}, W_{i-1}) \text{ 如果 } 2 \leq i \leq n$$

10 其中 $C_1 = a_1 b_1 + (a_1 + b_1) C_0$ 。

$$\text{则 } C_i = V_i = W_i \quad \text{对于 } 1, 2, \dots, n. \quad (5)$$

证：通过对 i 进行归纳来证明该定理。可知式 (5) 对 $i=1$ 成立。

如果 $i > 1$ 且 $C_{i-1} = V_{i-1} = W_{i-1}$ ，则

$$\begin{aligned} (V_i, W_i) &= (a_i, b_i) \circ (V_{i-1}, W_{i-1}) \\ 15 \quad &= (a_i, b_i) \circ (C_{i-1}, C_{i-1}) \\ &= (a_i b_i + (a_i + b_i) C_{i-1}, a_i b_i + (a_i + b_i) C_{i-1}) \\ &= (C_i, C_i) \end{aligned}$$

从而，通过归纳法式 (5) 成立。

20 能证明运算符 $\circ$ 是结合的。因此，能从给定的输入值中以任何次序计算出 V_i 与 W_i 。这提供了用树结构来生成进位的基础。注意运算符 $\circ$ 不是交换的，这蕴含不同输入对的优先级。

图 7 示出为第 32 位生成进位的一部分 32 位加法器设计。在第一与第二行（下起）中采用 4 输入对进位仲裁器，而在第三行中采用 2 输入对仲裁器。进位计算只通过三个逻辑层。

25 图 8 示出生成第 31 位进位的电路部分。对于进位结果的其它位，利用与图 7 与 8 中所示的类似的电路（互连）。一旦确定了进位位（即进位输入与结构得出生成或消除，在这一点上传播是不可能的），则可用单一信号线将该结果传递给更高的级。

30 最后一行为求和电路，它进行操作来异或输入运算数及进位结果。

通过将进位仲裁器的三行扩展成包含位 32，然后在第四行的位置 32 中放置带有来自前一行的位置 0 与 32 的输入的一个两位仲裁器，

便能得到从图 7 与 8 的加法器的进位输出。作为替代, 将 (加法电路的) 第四行中的位置 31 上的电路设置成既生成和又生成进位输出而不是只有异或电路的和输出的全加法器。

在典型的实施例中, 本方案采用 $0.6\mu\text{m}$ CMOS 工艺技术只占用
5 1.85ns 来完成 32 位进位计算。

只要 V_i 与 W_i 相等 (表示已生成进位), 只需要发送单干线信号 (Single-rail signal) 而不是双干线信号 (dual-rail signal)。这导致芯片面积的明显缩小, 尤其是在第三行中, 其中需要更多的空间来容纳从最低位跨到最高位的信号。从而, 得出的加法器是相当紧
10 致的。

作为采用上述技术的实例, 下面考虑带有适度芯片面积的 80 位高速加法器的设计。

通过相加两个 1 位数 a_i 与 b_i 来估算进位 C_i , 如上面表 1 中所示。值 a_i 与 b_i 定义了两种一般情况。第一情况, 当两个运算数位相等时产生进位请求。如果两个输入都是 1 出现 1 进位请求, 而如果两个输入
15 都是 0 则出现 0 进位请求。第二情况, 当运算数位具有不同值时不产生进位请求。字母 u 表示无进位请求。

通过取图 9 中所示的四路进位仲裁器作为示例引入进位仲裁的概念。任何输入对 a_i 与 b_i ($0 \leq i \leq 3$) 都能作出进位请求, 从而两个或更多进位请求可同时出现。因此, 必须仲裁这些进位请求。输入对
20 (a_3, b_3) 能作出不可屏蔽的进位请求, 这里不可屏蔽意味着输出 C 必须总是确认来自输入对 (a_3, b_3) 的进位请求。其它三个输入对 (a_2, b_2)、(a_1, b_1) 及 (a_0, b_0) 能作出可屏蔽的进位请求, 这里可屏蔽意味着输入对 (a_3, b_3) 可以屏蔽来自这三个输入对的进位请求。输入对 (a_2, b_2)
25 比输入对 (a_1, b_1) 及 (a_0, b_0) 具有更高的优先级。输入对 (a_0, b_0) 具有最低优先级。

只有在没有来自输入对 (a_3, b_3) 的不可屏蔽的进位请求时, 输出 C 才确认来自输入对 (a_2, b_2) 的可屏蔽进位请求。只有在没有来自输入对 (a_3, b_3) 的不可屏蔽的进位请求且没有来自输入对 (a_2, b_2) 的可屏蔽的进位请求时, 输出 C 才确认来自输入对 (a_1, b_1) 的可屏蔽的进
30 位请求。只有在没有来自输入对 (a_3, b_3)、(a_2, b_2) 及 (a_1, b_1) 的进

位请求时，输出 C 才确认来自输入对 (a_0, b_0) 的进位请求。表 7 列出实现四路进位仲裁器所需的真值表的概要。

a_3, b_3	a_2, b_2	a_1, b_1	a_0, b_0	C
00	—	—	—	0
11	—	—	—	1
01 或 10	00	—	—	0
01 或 10	11	—	—	1
01 或 10	01 或 10	00	—	0
01 或 10	01 或 10	11	—	1
01 或 10	01 或 10	01 或 10	00	0
01 或 10	01 或 10	01 或 10	11	1
01 或 10	01 或 10	01 或 10	01 或 10	u

表 7

利用相同的方法，能导出具有任何路数的进位仲裁器。通过使用组合进位利用进位计算的树结构的进位仲裁器，便能快速地生成进位。例如，图 10 示出基于进位仲裁的 16 位进位计算。实心点表示进位仲裁器。利用 m 路进位仲裁器的 n 位数加法能在与 $O(\log_m n)$ 成比例的时间中完成。

理论上，各进位仲裁器处理的输入越多，生成进位越快。然而，多于四路的进位仲裁器通常没有实际意义。首先，需要太多串联晶体管来实现这些仲裁器，这导致低效的 CMOS 设计。其次，仲裁器单元布局对于数据通路的位间距很容易变得太大。选择了四路进位仲裁器及它们的动态 CMOS 实现，因为在这一 80 位设计中它们给出好的结果。其它设计，诸如 32 位加法器，最好用 3 路进位仲裁器。

受到在自定时设计中所采用的双轨 (dual rail) 编码的推动，进位请求输出 c 可用两条线 (aa, bb) 编码，如表 8 中所示。式 6 与 7 给出表 7 与 8 定义的表现。

c	aa,bb
0	0 0
1	1 1
u	0 1
u	1 0

表 8

$$\overline{aa} = a_3b_3 + (a_3 + b_3)(a_2b_2 + (a_2 + b_2)(a_1b_1 + (a_1 + b_1)a_0)) \quad (6)$$

$$bb = a_3b_3 + (a_3 + b_3)(a_2b_2 + (a_2 + b_2)(a_1b_1 + (a_1 + b_1)b_0)) \quad (7)$$

图 11 示出按照上式的四路进位仲裁器的直接动态 CMOS 实现。该电路的操作使得在控制握手信号的复位阶段期间输入 a_3 与 b_3 为低时将节点 n1 与 n2 预充电到高，而在计算阶段期间有条件地放电。缓冲器是用来保持驱动强度的。

图 12 给出四路进位仲裁器的修正版本。这里假定每一个输入对 (a_i, b_i) 取三个值 (00)、(11) 与 (10) 之一，并已将 (01) 变换成 (10)。理由是双重的。第一，容易将修改的电路布置进数据通路的位间距中（在本实施例中为 $21.5\mu\text{m}$ ）并且比直接实现快大约 300ps。第二且更重要的是，输出 aa 与 bb 具有新的意义。如果输出 aa 与 bb 具有不同的值，这表示没有来自输入的进位请求，如上所述。然而，我们可以考虑四路进位仲裁器的另一方面。如果将四路进位仲裁器作为 4 位加法的进位生成电路考虑，则可认为输出 aa 与 bb 之一为以零进位输入生成的进位输出，而另一个则是以 1 进位输入生成的。直接实现并不区分哪一个是以零进位输入生成的进位输出及哪一个是以 1 进位输入生成的。修正后的电路正确地给出输出 aa 为以 1 进位输入生成的进位输出而输出 bb 则为以零进位输入生成的进位输出。这导致芯片面积的明显缩小（见下面）并且是本实施例的重要特

征。

然而，修正的实现采用要求将输入从 (01) 转换成 (10)。幸好这不成问题；转换是简单的。每位包括一个 2 输入“与非” (NAND) 门及一个 2 输入“或非” (NOR) 门。为了实现原因，通常无论如何需要这些门来隔离来自输入总线的信号。这里的差别在于用“与非”及“或非”门来替代反相器。如果将两条输入总线设计成使用预充电结构，当将总线预充电到高时，经过“与非”及“或非”门之后的输出自然是低的（在动态实现中所要求的）。此外，这些“与非”及“或非”门能重新用于 ALU 设计中的逻辑运算。

首先考虑高速加法器设计的传统方法。图 13 示出采用进位选择方案的传统加法器设计。将输入分成 d 位的群。每一群需要两个加法器。一个为带有零进位输入的加法器而另一个则带有 1 进位输入。进位发生器负责为所有的群生成边界进位，然后利用多路复用器用它们来选择适当的和。为了平衡进位发生器与群加法器两者的延时，必须作出设计选择来选择适当的群。如果将群加法器做得太长，则群加法器的增加的延时超过进位发生器中的减少的延时。如果将群加法器做得太短，则进位发生器的逻辑深度增加而其延时决定总加法器延时。

图 14 中示出按照本发明的一个实施例的 80 位加法器的框图。将整个加法器看成是（但并不分成）由 5 个 16 位的群构成的。第一行为转换电路，包含 2 输入“与非”及“或非”门。第二与第三行为产生各群内部的进位并具有前面讨论的形式的四路仲裁器。第四行产生带有零进位输入及 1 进位输入的两个中间和。最后一行为选择最终和结果的多路复用器及生成边界进位 C_{16} 、 C_{32} 、 C_{48} 及 C_{64} 的三个进位仲裁器。在进位计算的两行后面已经生成 16 个最低位的进位。很清楚，与传统的进位选择方案比较，已消除了对群加法器的需要。两个中间和是在进位生成树内巧妙地生成的。这导致芯片面积的明显缩小，尤其是将群做得长时，因为群加法器也需要某些用于进位计算的机制。在某种意义上，传统的方法导致太多冗余性。值得指出，如果已知信号 aa 与 bb 相等（说明已生成进位，1 进位或 0 进位请求）只需要发送单轨信号（而不是双轨信号）。

加法器是以 $0.5\mu\text{m}$ 三金属 CMOS 技术设计的。该布局具有规范的结构及均匀的扇入与扇出负载，并从而非常紧致。布局后 HSPICE 模

拟显示加法器占用 3.5ns 来完成 80 位加法。表 9 中概述了这一加法器的特征。

技术	0.5 μ m 三金属 CMOS
电源	3.3Volt
加法时间	3.5ns
有效面积	1720 μ m $\times$ 124 μ m
晶体管数	2546
晶体管密度	12k/mm ²

表 9

5 已研制了一种进位仲裁方案，其中进位是基于进位计算的结合性用若干进位请求的优先化仲裁生成的。所提出的方案不仅由于需要较少的逻辑层而产生高速加法器，并且还提供规范与紧致的布局及均匀的扇入与扇出负载。

10 已设计出及修正了四路进位仲裁器的动态 CMOS 实现。修正版本使用双重含义。如果输出 aa 与 bb 相等，说明已生成了进位。如果它们不同，说明输出 aa 是以 1 进位输入生成的进位输出而输出 bb 则是以零进位输入生成的。

已根据提出的方案描述了异步 80 位加法器设计，它以 0.5 μ m 三金属 CMOS 技术用 3.5ns 来计算 80 位和。有效面积为 1720 μ m $\times$ 124 μ m。加法器的重要特征为在进位生成树内巧妙地生成两个中间和。

说明书附图

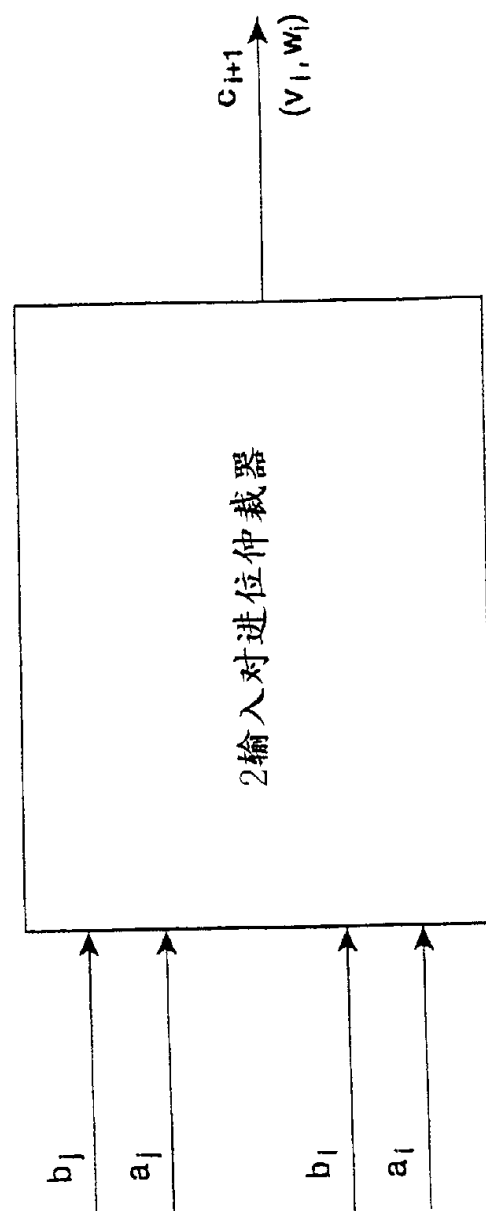


图 1

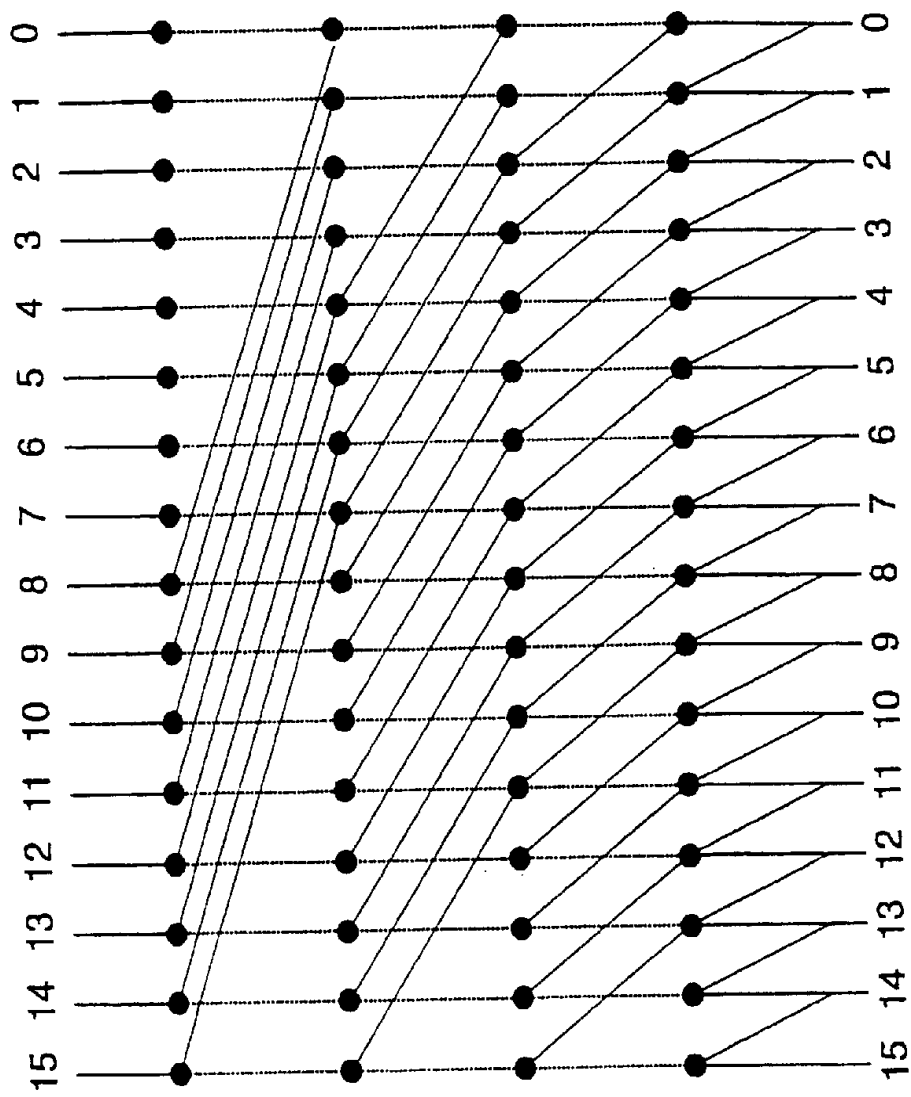


图 2

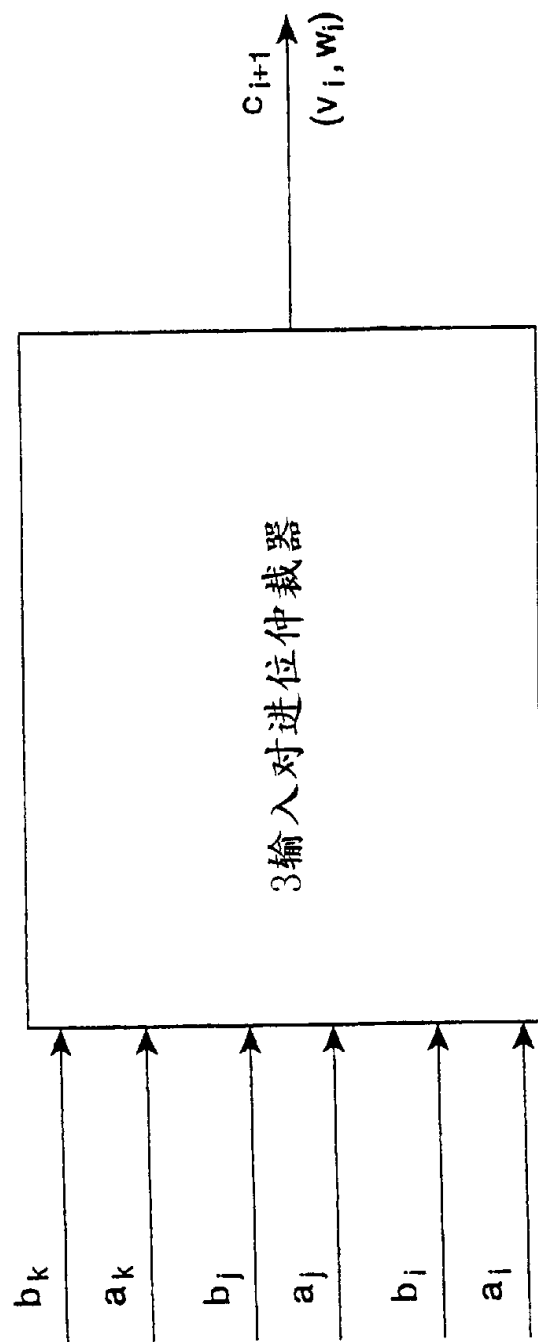


图 3

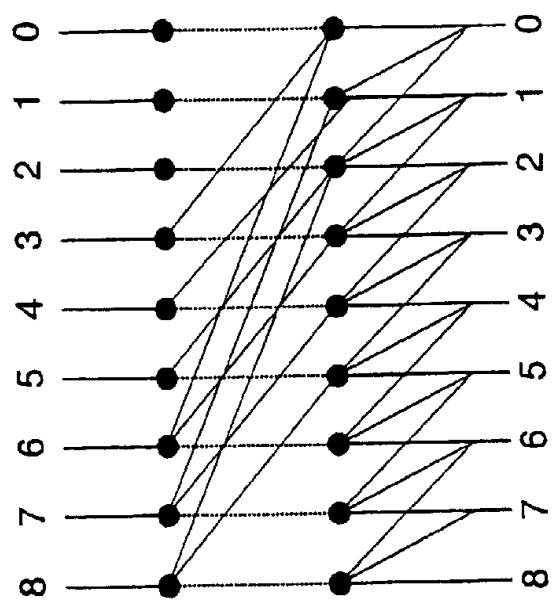


图 4

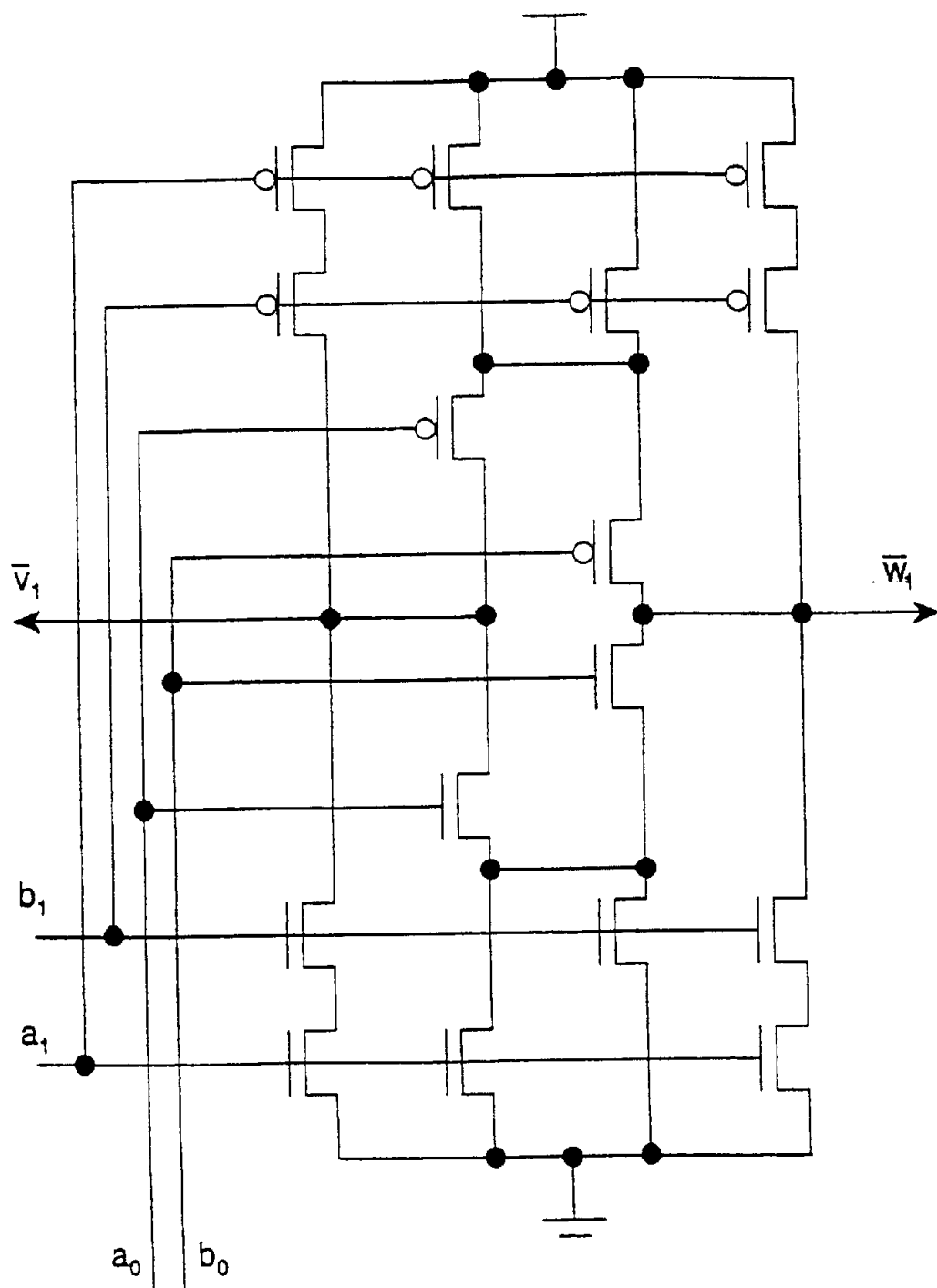


图 5

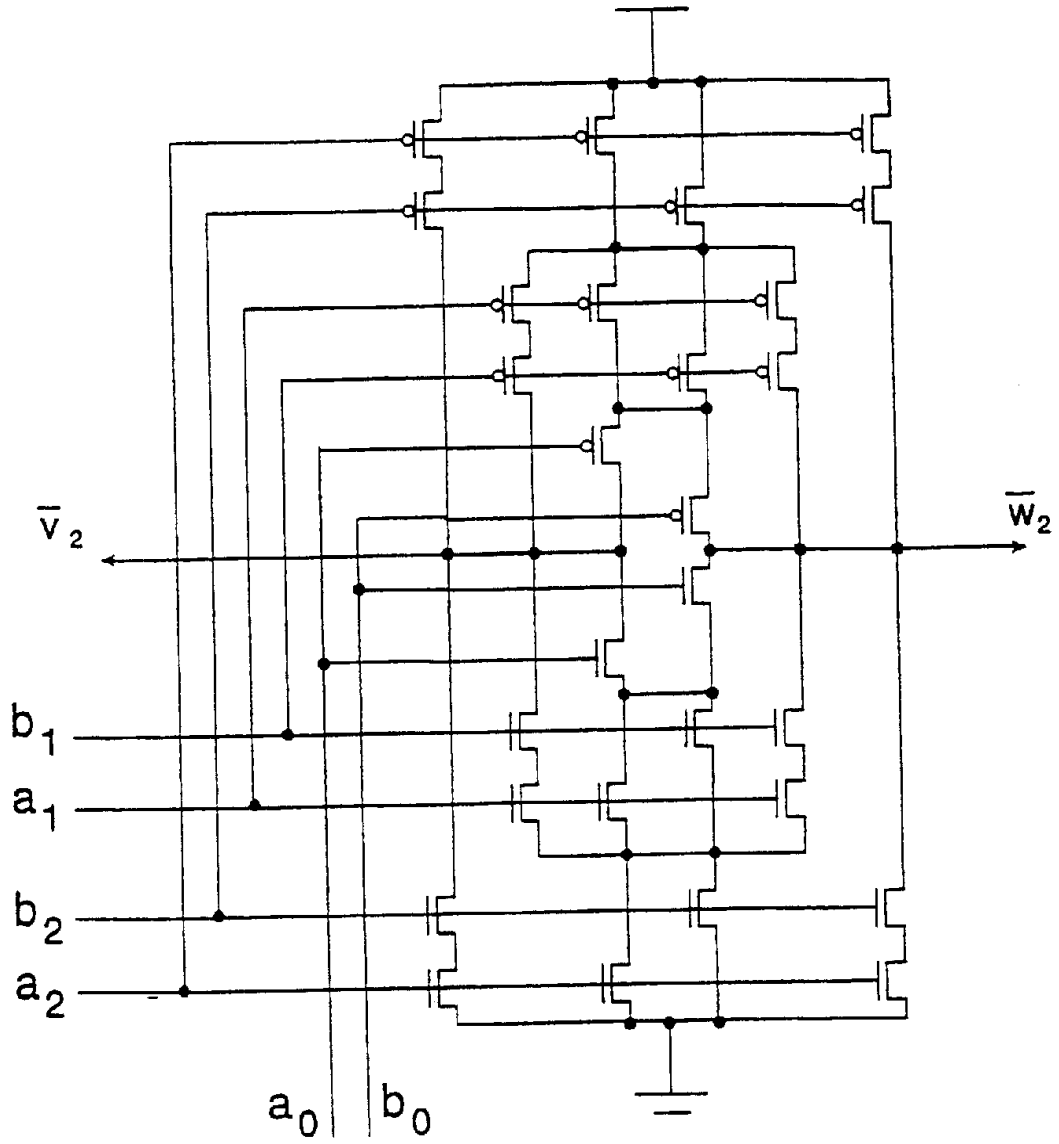


图 6

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

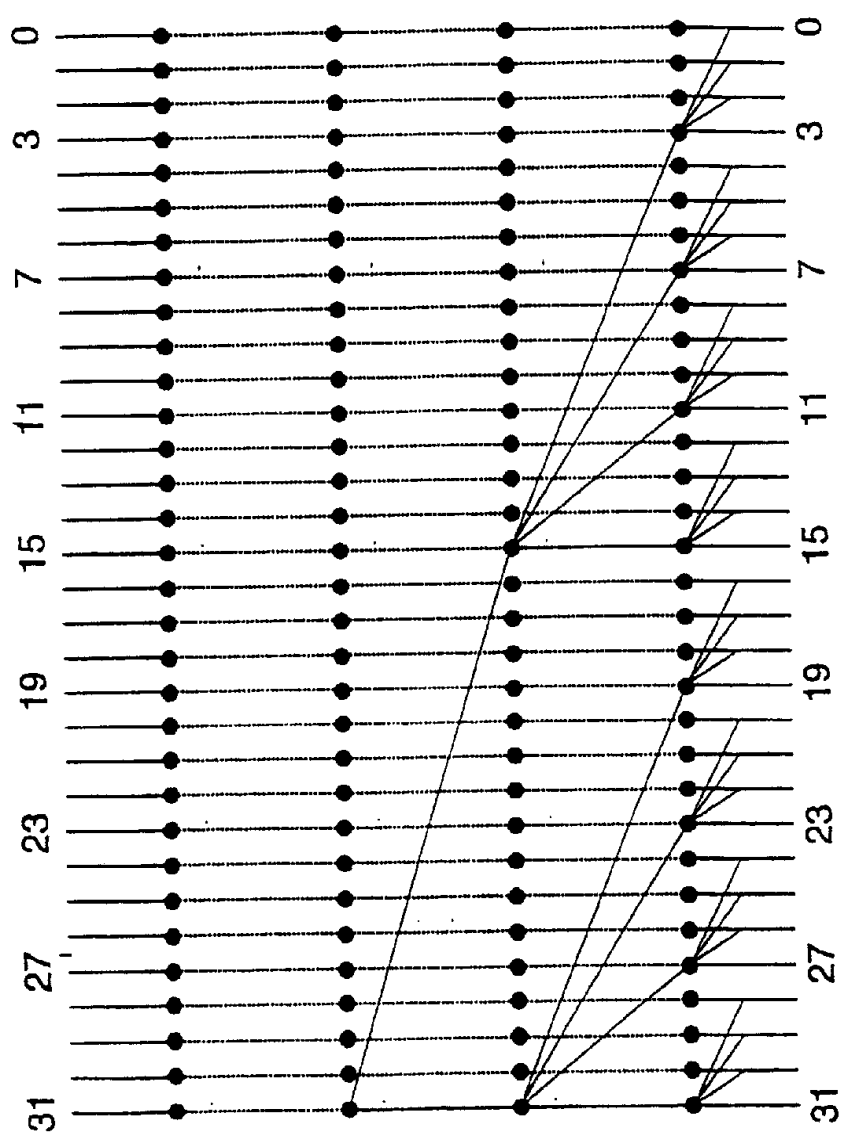


图 7

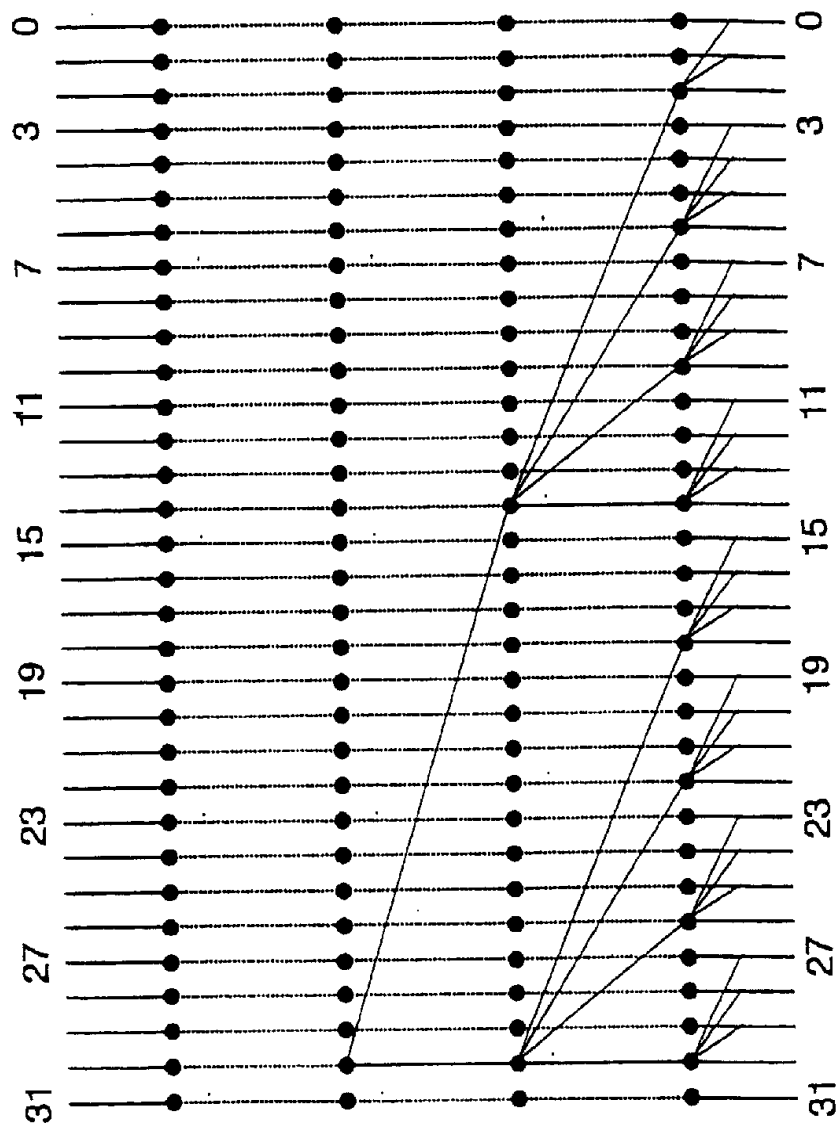


图 8

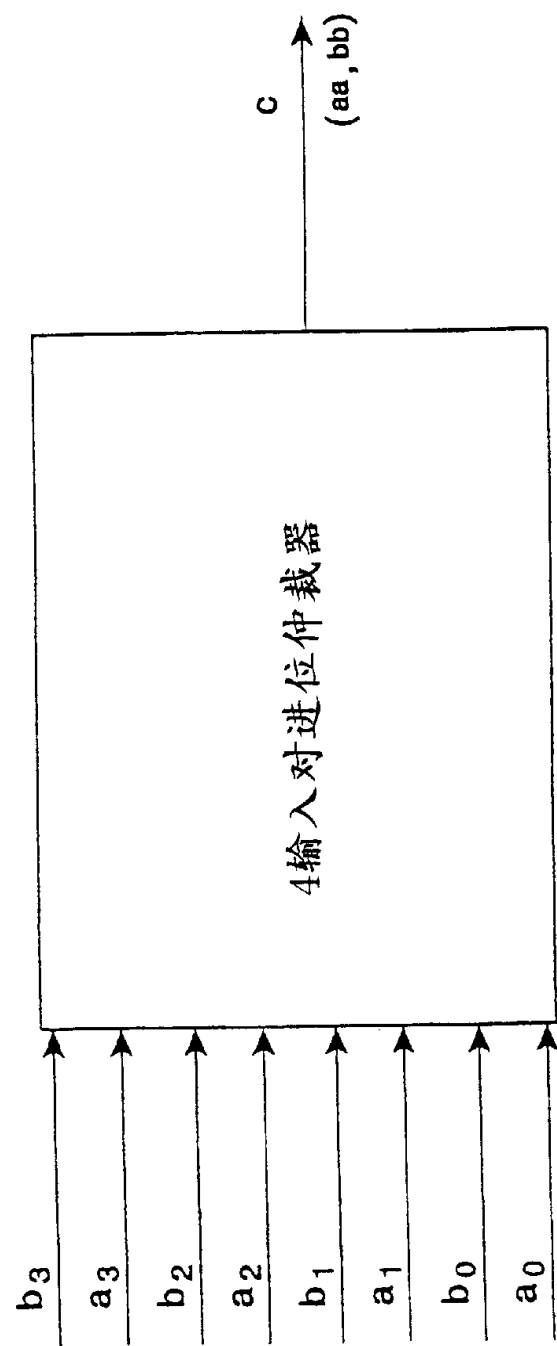


图 9

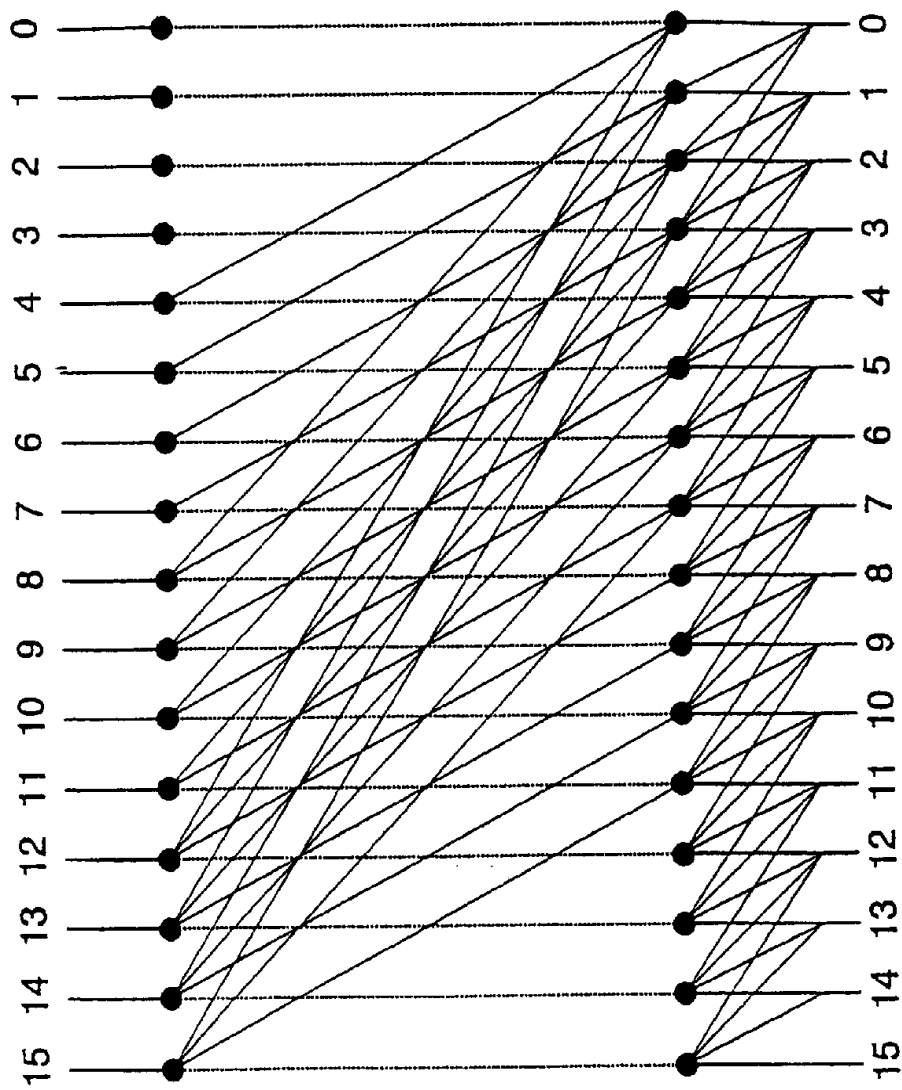


图 10

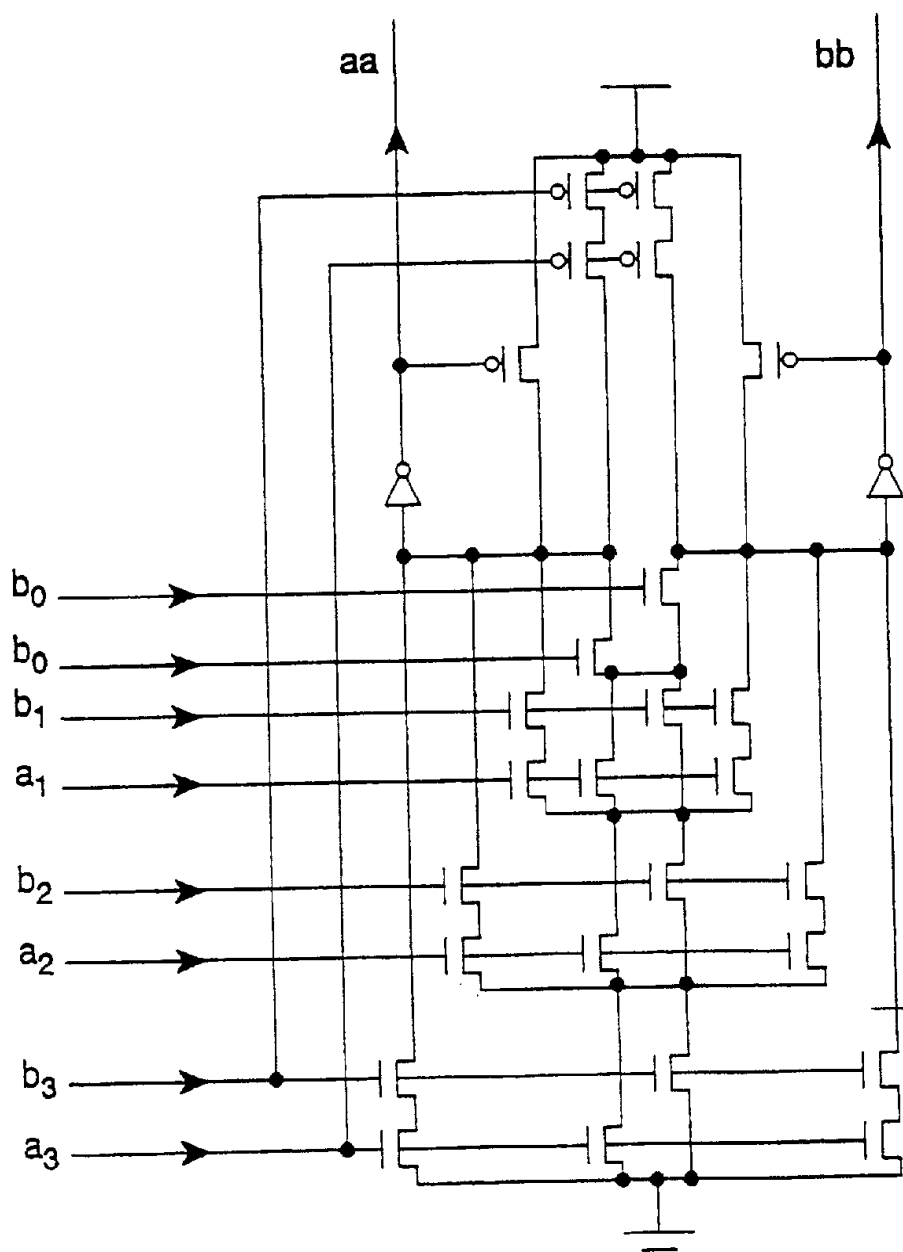


图 11

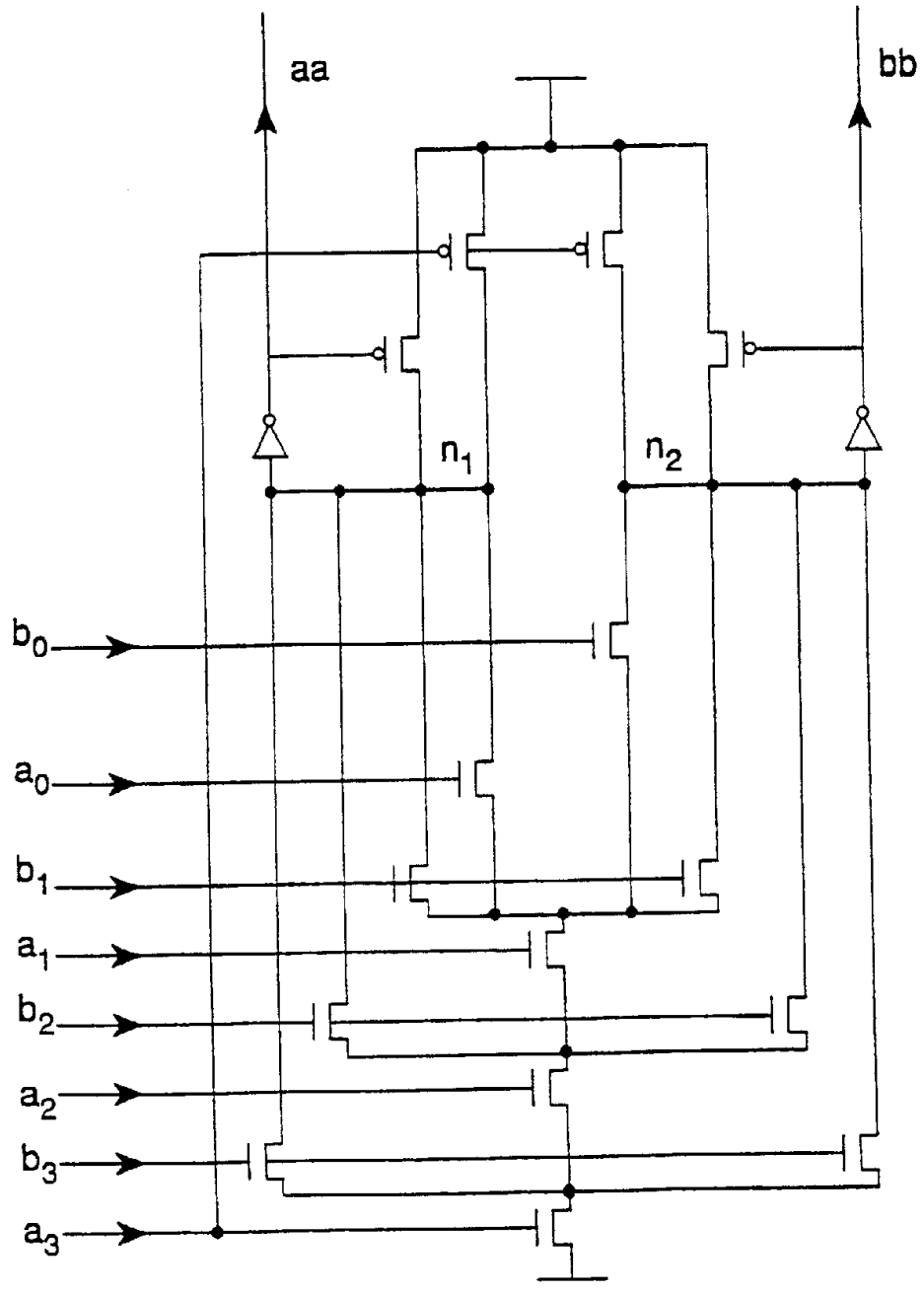


图 12

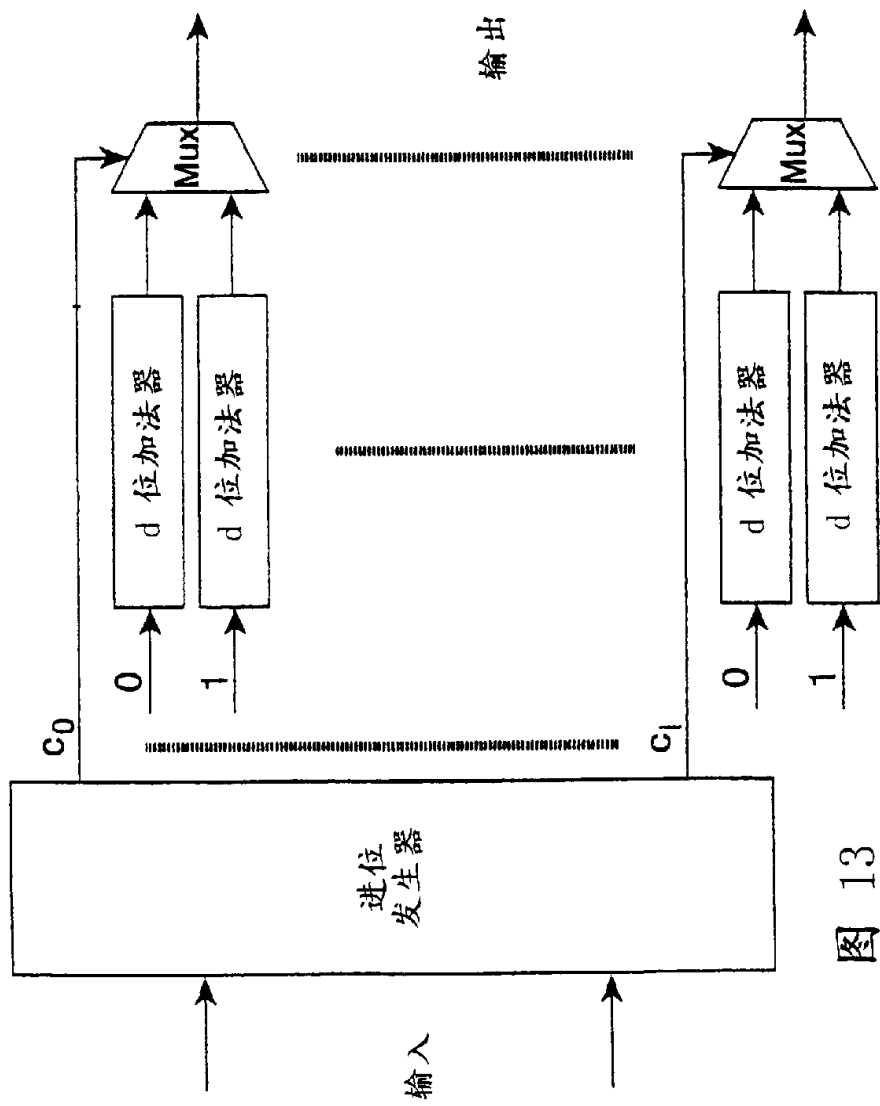


图 13

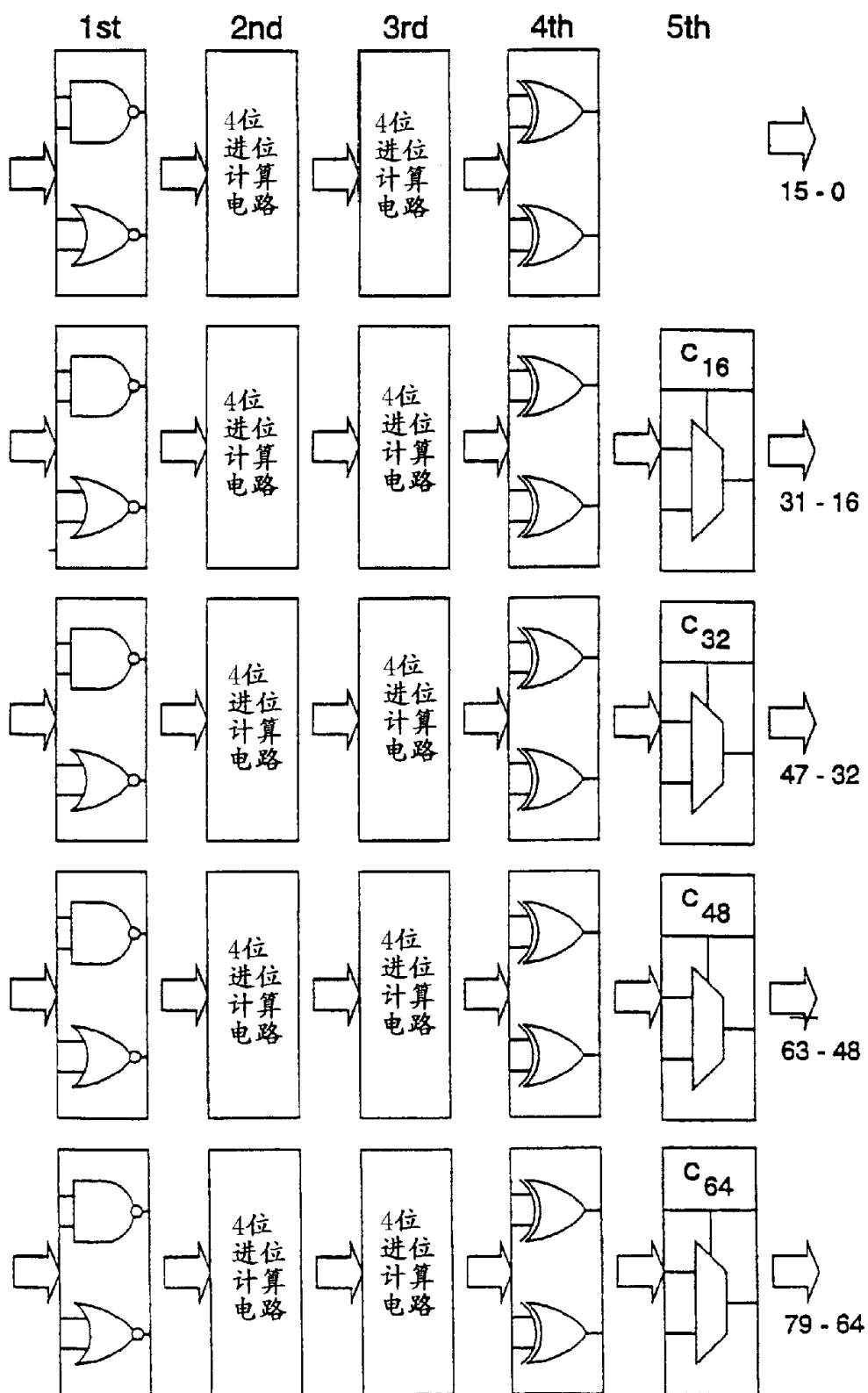


图 14