[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 12/00

[12]发明专利说明书

[21] ZL 专利号 96106203.7

[45]授权公告日 2002 年 3 月 20 日

[11]授权公告号 CN 1081359C

[22]申请日 1996.5.7 [24]颁证日 2002.3.20
[21]申请号 96106203.7
[30]优先权
[32]1995.6.2 [33]US [31]459,490
[73]专利权人 国际商业机器公司
地址 美国纽约
[72]发明人 戴维·阿诺尔德·雷克
[56]参考文献

US 4128883A	1988. 8.23	G06F12/06
US 4652993A	1987. 3.24	G06F9/18

审查员 朱世菡

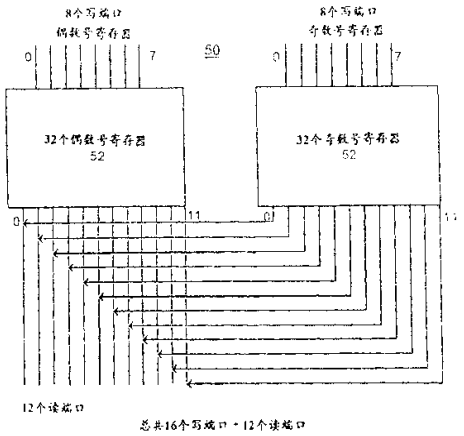
[74]专利代理机构 中国国际贸易促进委员会专利商标事务所
代理人 王以平

权利要求书 3 页 说明书 9 页 附图页数 11 页

[54]发明名称 具有交错式写端口的多端口寄存器堆

[57]摘要

提供一种用于超长指令字(VLIW)和 N 路超标量处理器的改 进的高速寄存器堆。此高速寄存器堆包含选定数目的通用寄存器(GPR)构件的拷贝。GPR 构件至少包含两个交错式寄存器子片。每 个子片包含 N 个数目的写端口和 M 个数目的读端口。子片的写端 口是交错式的并且读端口是非交错式的。



ISSN1008-4274

权 利 要 求 书

1. 一种用于和超长指令字和N路超标量处理器的高速寄存器堆，包括:

一个通用寄存器构件，所述通用寄存器构件包含至少两个交错式寄存器子片，每个所述的子片包含N个数目的写端口和M个数目的读端口，所述子片的写端口是交错式的而读端口是非交错式的；以及

所述通用寄存器构件的一个选定数目的拷贝。

2. 根据权利要求1中所述的高速寄存器堆，其特征在于所述通用寄存器构件包含一个奇数号寄存器子片和一个偶数号寄存器子片，该奇数号寄存器子片包含所述N个数目的奇数号写端口，该偶数号寄存器子片包含所述N个数目的偶数号写端口。

3. 根据权利要求1中所述的高速寄存器堆，其特征在于所述选定数目的拷贝被组合到预定的组中。

4. 根据权利要求3中所述的高速寄存器堆，其特征在于超长指令字处理器包括十六个算术逻辑单元0－15，并且或者第一组八个算术逻辑单元0－7或者第二组八个算术逻辑单元8－15向所述预定组中的预定某组写。

5. 根据权利要求4中所述的高速寄存器堆，其特征在于所述拷贝的所述的选定数目等于4并且所述的第一组八个算术逻辑单元0－7对所述拷贝中的两个进行写，所述第二组八个算术逻辑单元8－15对所述四个拷贝中的另外两个进行写。

6. 根据权利要求 1 中所述的高速寄存器堆，其特征在于定义所述通用寄存器构件的所述寄存器的所述至少两个交错式子片的所述 M 个读端口相互进行“或”，从而为所述通用寄存器构件提供了 M 个读端口。

7. 根据权利要求 1 中所述的高速寄存器堆，其特征在于所述通用寄存器构件包含所述的 M 个数目的读端口并且所述拷贝的所述选定数目等于四，从而提供了总共四倍于 M 的读端口。

8. 根据权利要求 7 中所述的高速寄存器堆，其特征在于所述的读端口数目 M 等于 12，从而提供了总共 48 个读端口。

9. 根据权利要求 1 中所述的高速寄存器堆，其特征在于所述通用寄存器构件包含两倍于所述 N 个数目的写端口。

10. 根据权利要求 1 中所述的高速寄存器堆，其特征在于所述选定数目的拷贝为四个，即拷贝 1、拷贝 2、拷贝 3 和拷贝 4，并且超长指令字处理器包含十六个算术逻辑单元 0－15，而且第一组八个算术逻辑单元 0－7 对拷贝 1 和拷贝 2 进行写，第二组八个算术逻辑单元 8－15 对拷贝 3 和拷贝 4 进行写。

11. 根据权利要求 10 中所述的高速寄存器堆，其特征在于所述四个拷贝即拷贝 1、拷贝 2、拷贝 3 和拷贝 4 与一个高速缓冲存储器相连，用于对所有所述的四个拷贝提供高速缓存负载写。

12. 根据权利要求 1 中所述的高速寄存器堆，其特征在于超长指令字处理器包含 16 个流水线 0－15，并且选定数目的拷贝为四个，即拷贝 1、拷贝 2、拷贝 3 和拷贝 4，所述拷贝 1 和拷贝 2 与所述流水线 0－7 相连，所述拷贝 3 和拷贝 4 与所述流

水线 8－15 相连。

13. 根据权利要求 12 中所述的高速寄存器堆，其特征在于所述拷贝通过八写端口与所述流水线相连。

14. 根据权利要求 1 中所述的高速寄存器堆，其特征在于所述 M 个数目的所述读端口既包含写入端口也包含非写入端口。

说　明　书

具有交错式写端口的多端口寄存器堆

本发明涉及具有多个读写端口的随机读/写存取存储器(*RAM*)或多端口寄存器堆，更具体地涉及包含64个或更多端口的高速寄存器堆。

已知的多重或多端口寄存器堆不适用于超长指令字(*VLIW*)和 *N* 路超标量处理器。*VLIW* 处理器由一系列能同时执行指令的各个部分的算术逻辑或功能单元组成。具有8个，16个或更多流水线功能单元的 *VLIW* 和 *N* 路超标量处理器需要一个中心寄存器堆用于通信，该寄存器堆必须给每个流水线提供三个或四个端口用于可能的64端口或更多的要求。当前的技术恰恰不允许构造有如此多端口的直接物理实现，特别是访问和写时间要求非常严格时。目前大多数物理上可连线的端口可能少于或等于二十(20)。

本发明的主要目的是提供一种用于超长指令字(*VLIW*)和 *N* 路超标量处理器的改进的多端口寄存器堆。

简言之，就是提供一种用于超长指令字(*VLIW*)和 *N* 路超标量处理器的高速寄存器堆。该高速寄存器堆包含选定数目的通用寄存器(*GPR*)构件的拷贝。*GPR* 构件包含至少两个交错式寄存器子片。每个子片包含 *N* 个数目的写端口和 *M* 个数目的读端口。子片的写端口是交错式的并且读端口是非交错式的。

根据本发明的高速寄存器堆适用于超长指令字（VLIW）和N路超标量处理器。

从下面图示所说明的本发明最佳实施方式的详细描述中，可以更好地理解本发明的上述目的以及其它目的和优点，其中：

图 1 是一个框图，它说明了一对根据本发明的有二十个端口的通用寄存器堆(*GPR*)物理构件；

图 2 是一个框图，它说明了一个六十四端口寄存器堆，该寄存器堆包含图 1 中二十八端口逻辑 *GPR* 堆逻辑构件的四个拷贝；

图 3 是一个框图，它说明了利用图 1 中 *GPR* 堆构件所进行的一种写端口的划分方案；

图 4 是一个框图，它说明了利用图 1 中 *GPR* 堆构件所进行的本发明的一种替代的简化的写端口划分寄存器堆方案；

图 5 是一个框图，它说明了一个包含图 4 中六十四端口寄存器堆的超长指令字(*VLIW*)处理单元；

图 6 是一个框图，它说明了一个七十二端口寄存器堆。该寄存器堆包含图 1 中二十八端口寄存器堆构件的四个拷贝；

图 7 是一个说明了图 1 中寄存器堆构件中的一个单元以及一个地址解码和算术逻辑单元(*ALU*)加法器的示意图；

图 8 是一个说明了根据图 1 中本发明的 *GPR* 堆构件的一个可变性能读访问的示意图；

图 9 是图 1 中 *GPR* 堆构件位 0 的一个示例性的布局；

图 10 是图 1 中 *GPR* 堆构件位 0 的又一个示例性的布局。它说明了根据本发明的一种替代的读端口布局；

图 11 是一个说明了包含图 1 中 *GPR* 堆构件的处理单元的一个示例性的布局的示意图。

现在参考附图，图 1 说明了根据本发明的一个二十八端口通用

寄存器(*GPR*)堆逻辑构件，一般由参考符 50 标识。根据本发明的特点，交错式写端口，可以看出写端口特别分成了奇数号和偶数号寄存器子片。位宽 32 的 *GPR* 寄存器堆构件 50 产生总共 16 个可用的写端口和 12 个可用的读端口。

GPR 堆逻辑构件 50 包含一对二十端口的 *GPR* 堆物理构件 52。该二十端口 *GPR* 堆物理构件 52 包含一个偶数号 *GPR* 子片 52 和一个奇数号 *GPR* 子片 52 的交错式写端口。这两个 *GPR* 子片 52 提供 12 个读端口，并且其中一个有 32 个偶数号寄存器的子片提供 8 个写端口，另一个有 32 个奇数号寄存器的子片提供 8 个写端口。该 *GPR* 寄存器堆构件 50 包含十六个物理写端口和十二个物理读端口，它们可用于构造一个 64 端口 *GPR* 逻辑寄存器堆，这在图 2 中总体由 60 标识，它具有极佳的性能特性。

现在参考图 2，该六十四(64)端口通用寄存器堆包含图 1 中二十八端口寄存器堆逻辑构件 50 的四个拷贝。随着 *GPR* 堆逻辑构件 50 复制了 4 次，读端口 4 倍地增长可满足总共 48 个读端口的所需数目。对于 64 位宽的寄存器堆 60，64 个寄存器分成两组；一组 32 个偶数号的寄存器和一组 32 个奇数号的寄存器。实用时，此寄存器堆方案 60 允许编译器通过某些寄存器分配组合及重命名算法，来产生奇数号和偶数号目标寄存器操作数目大致相同的指令的基本块。在 *VLIW* 指令严格编码的情况下，每个指令包都固定地赋予或者一个奇数号(奇数包)或者一个偶数号目标操作数寄存器堆端口，从而省去了在指令中对寄存器地址的最后一位进行编码的需要。

由于在目标寄存器中执行写操作之前，与非交错式寄存器堆的 16 路写选择相比，交错式寄存器堆 60 只需 8 路写选择，因而其写

性能提高了。然而复制不可能引起写端口的增加，这是由于所有拷贝接收所有的写以保持有效的拷贝。*GPR* 64 端口寄存器堆 60 还提高了寄存器堆的实际读端口性能，这是由于它将拷贝 1，2，3，4 放在了最接近于它所驱动的流水线数据流处。这种复制策略的一个缺点是目标操作数必须物理上遍历图 2 寄存器堆 60 中 *GPR* 逻辑构件 50 的所有四个拷贝的整个区域。

图 3 是一个框图，它说明了利用这四个 *GPR* 构件 50 的拷贝而进行的一种写端口划分寄存器堆方案，一般标识为 68。每个块 70 包含两个 *GPR* 构件 52，带有进入每个单元的 8 个附加的写路径或端口。在寄存器堆 68 中，任何装载都能同时写所有的拷贝 1，2，3 和 4，并且任何 *ALU* 都能同时写所有拷贝 1、2、3 和 4。由于需要提供 16 个 *ALU* 写端口和 8 个负载写端口所需的 8 个附加的写端口，方案 68 是不现实的。

图 4 说明了一种包含四个块 80 的替代的、简化的写端口划分寄存器堆方案，一般标识为 78。每个块 80 包含两个 1/2 块或子片 52 或一个块 50。在拷贝 1 和 2 中，其中一个子片 52 用于 *ALU* 写管道 0—7 并且另一个子片 52 用于 8 个负载写端口。在拷贝 3 和 4 中，其中一个子片 52 用于 *ALU* 写管道 8—15，并且另一个子片 52 用于 8 个负载写端口。图 4 的划分方案 78 提供了一种实用的解决方法，它总共需要十六个写端口，而不是图 3 中不实用的方案 68 总共所需要的 24 个写端口。一个复杂的编译器能在一个给定周期内把目标写只限定于 4 个拷贝 80 中的两个。对于产生必须写入寄存器堆 78 的目标结果的 *ALU* 操作，编译器设法使一个基本块的所有或大部分工作限于 *VLIW* 处理器的总共十六个流水线槽或管道

中的第一组八条流水线 0—7 或第二组八条流水线 8— 15 中。

寄存器堆 78 的两组拷贝 80 中的每一组只需从十六条管道中一半的管道中接收 *ALU* 目标结果：或者是八条管道 0—7 或者是八条管道 8—15，这使得每个拷贝所需的总的写端口数减少 8 个。附加的读端口并不十分昂贵；但是，由于影响了具有额外输入端口的所有拷贝 1，2，3，4，附加的全局写端口是十分昂贵的。当管道 8—15 需要查看管道 0—7 的结果或反之时，编译器适当地调度移动寄存器操作并适当地调度寄存器引用，使得由此引起的管道阻塞得以完全避免。

图 5 是一个框图，它说明了一个包含六十四端口 *GPR* 堆 78 的 *VLIW* 处理单元，一般标识为 82。图 5 所示的 *ALU* 目标划分方案的另一个显著的优点是极大地减少了获得任何 *ALU* 结果到任何写端口所需的总线连线的总长度，并降低了负载需求，即从没有写端口划分时的 4 减少到 2。标号 84 的连线代表划分寄存器堆 78 所需的最长的 *ALU* 写总线连线。标号 86 的连线代表未划分的寄存器堆 60 所需的最长的 *ALU* 写总线连线。在划分的 *ALU* 目标寄存器堆 78 中，最坏情况下，一个 *ALU* 只需驱动其自身和一个相邻的寄存器堆拷贝，与之相反，在非划分方案中则要驱动下一个与之相邻的拷贝。这也潜在地允许这些总线连线是双倍粗和双倍宽的连线。更粗和更宽的连线具有较低的电阻，并且还由于与非划分寄存器堆方案相比，总线连线可以只以大约一半的长度工作，从而提供了更好的性能。写端口目标划分显然不用于高速缓存数据负载写端口，其部分原因是由于在商用环境中，性能通常受到可用的负载管道数的限制，而不是 *ALU* 数的限制。这样 *ALU* 操作能够在每个 *ALU* 目标

划分中重复而不会冲突，负载就不能。而且，存在许多负载相关的情况：例如链表，这使得延迟非常重要。对于高速缓存数据负载的情况，由于数据总线已经是带有高电流驱动源的长连线，几乎不存在周期时间/写连线延迟差别。驱动4个寄存器堆拷贝，而不仅仅是2个，使得总线延迟的改变不显著，特别是假定负载路径已经是2个周期的路径时。

图6说明了一个七十二端口寄存器堆，它包含图1中所示的二十八端口*GPR*堆逻辑构件50的四个拷贝，一般标识为90。图6中，整个寄存器堆的布局包含八个基本的20端口*GPR*子片52，所产生的逻辑寄存器堆能够对总共72个逻辑端口进行每周期最多48次读和24次写。这些逻辑端口具有非常快的读访问和写入性能能力。

图7是一个说明了图1中寄存器堆构件52的一个单元的示意图。一个*DCACHE*数据输入施加到一个8路字节校准器100上，而该8路字节 对齐的输出端与一个4路多路选择器(*MUX*)102相连。*MUX* 102与一个2路选择器(*MUX*)104相连。它们共同定义了单元106中的一个8路*MUX*操作。单元106包含一对锁存器，表示为*L*1和*L*2锁存器或一个移位寄存器锁存器。*L*1锁存器与一个8路*MUX* 108相连，该*MUX*的输出端与8路*MUX* 110相连。一般标识为112的*GPR*地址译码器包含一对地址缓冲器114，116，并且地址*L*1锁存器118，120将地址信号施加到*MUX* 108和110上。*C*1时钟输出用于为*L*1和*L*2锁存器118和120设置地址缓冲器114和116。*MUX* 110的输出施加到加法器122上，该加法器与相关联的算术逻辑单元(*ALU*)的*L*2*锁存器124相连。*L*2*

锁存器 124 的输出施加到 4 路 *MUX* 126 上，该 *MUX* 通过一个标号为 128 的全局网路线与 2 路 *MUX* 104 相连。在具有 6 级金属喷镀的互补金属氧化物半导体(*CMOS*)集成电路实现中，全局网路线 128 可以是第 5 级金属并做得较厚，例如 2 倍于标准的线宽。*L*2 锁存器 106 的单元输出施加到 64 路 *MUX* 130 上，该 *MUX* 提供了一条恢复路径。

图 8 说明了本发明的一种可变读端口性能特性。对每个寄存器堆拷贝 50，不是所有的 12 个读端口都有相同的性能要求。8 个读端口要求尽可能快地写入。4 个快速读端口馈给 *ALU* 的右输入端，这样对减法和其他功能必须具有求真和取补的功能。其它的 4 个快速读端口馈给其它的 *ALU* 输入端(不取补)和 *CACHE* 逻辑(机器关键路径)并且也要做得尽可能快。一个 *L*1 锁存器 142 提供 2 个锁存相 — *PHASE* 和＋*PHASE*。*L*1 锁存器相＋*PHASE* 用于驱动多个读端口选择器 144 和 146 以提供快速的 *L*1 写入端口。写入端口在同一个时钟周期内将数据从一个写端口传送到读端口。同样编译器能够保证在任意给定周期内没有 2 个以上的输出读选择器会访问相同的寄存器，从而限制了必须提供的电流，并进一步提高了性能。*L*1 锁存器相－*PHASE* 用于驱动缓冲器 148，该缓冲器与一对选择器 150，152 相连，从而提供了 4 个慢速的 *L*2 读端口，这些读端口是非写入式的，并且没有严格的性能要求，这是由于当它们无论用于存储寄存器数据访问还是加载意外保存恢复寄存器时都没有严格的定时要求。这 4 个慢速读端口可以容忍较长，较细和较不规则的总线连线，这给了设计者额外的自由度。与写端口相连的旁路选择 154 给 *ALU* 管道提供了快速读端口。

图 9 提供了图 1 中 *GPR* 堆逻辑构件 50 位 0 的示例性布局。如布局中间附近所示，64 个 *GPR* 寄存器单元分为 32 个奇数号单元 106 和 32 个偶数号单元 106。与 8:1 多路选择器相邻的列定义了 4 路和 2 路 *MUX* 102，126，并且 104 通过多条写连线与偶数号和奇数号单元 106 相连，这些写连线由标号为 160 的示例性连线指示。八个 *L*1 读端口和四个 *L*2 读端口利用指示单元 106 到读端口的示例性连接的连线进行说明。

参考图 10，图 1 中 *GPR* 堆逻辑构件 50 位 0 的另一个示例性布局说明了根据本发明的一种替代的读端口布局。根据图 8 中所说明和描述的，*GPR* 堆逻辑构件 50 包含八个快速 *L*1 读端口和四个慢速 *L*2 读端口。在写译码，时钟，缓冲和测试以及第一组四个八路写 *MUX* 之间提供了四个快速 *L*1 读端口。在第二组四个八路写 *MUX* 和四个慢速 *L*2 读端口之间提供了四个快速 *L*1 读端口。在八个写端口输入端和第一组四个八路 *MUX* 之间，这些八路 *MUX* 和这些单元之间以及这些单元与四个慢速 *L*2 读端口的其中一个端口之间表示了示例性的连线。

图 11 是一个说明了一个包含 *GPR* 堆 78 的处理单元的示例性布局的示意图。在图 11 中，划分的 *GPR* 堆 78 通常表示在位于寄存器/寄存器(*RR*)管道 0—7 之上和寄存器/存储(*RS*) 管道 0— 5 及相关联的 *DCACHE* 目录(*DDIR*)及段旁顾缓冲器(*SLBs*)0—7 之下的中心位置处。在偶数号和奇数号 *GPR*1/2 拷贝 1(图 1 中的子片 52)之间，以及一个解码器和控制区域之间表明了一条示例性的连线，该控制区域能设置有更粗的连线，例如，两倍于标准的宽度。

虽然参照图中所示的本发明的实施方式详细描述了本发明，但

是这些详述并不是为了限制所附的权利要求中所要求的本发明的范围。

说　明　书　附　图

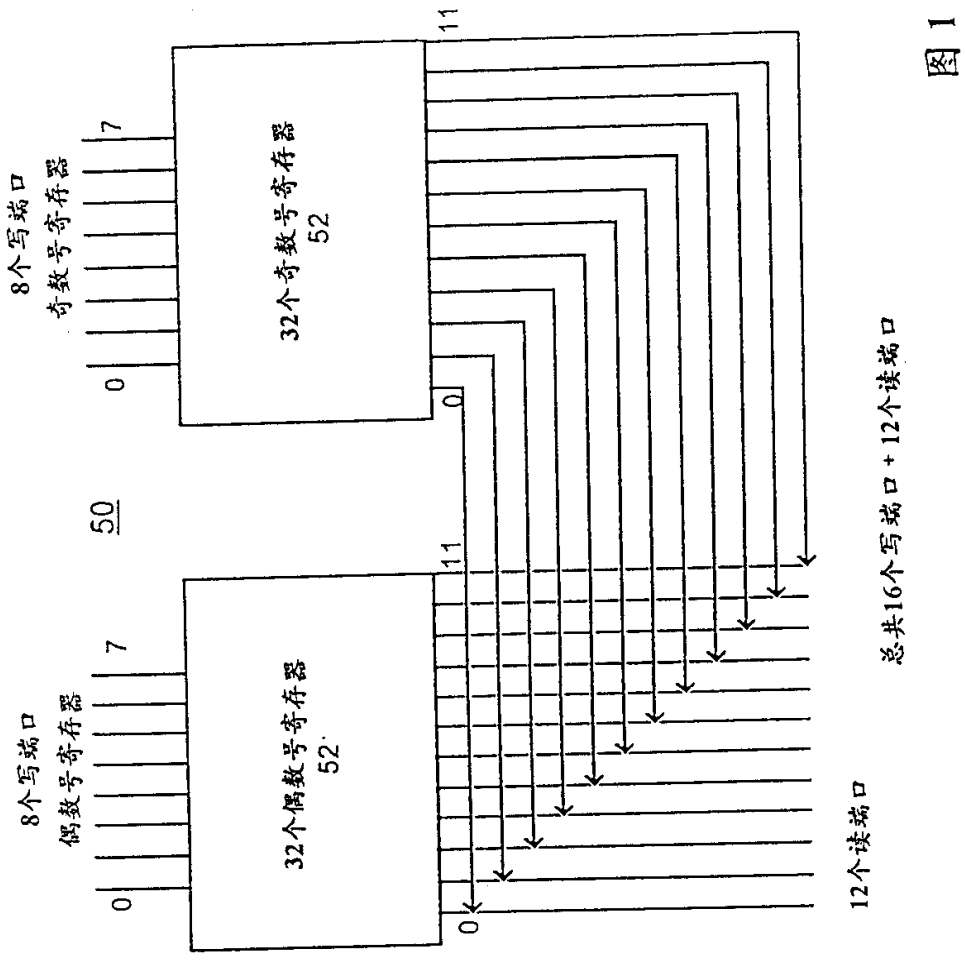

8个写端口
奇数号寄存器
7
0
32个奇数号寄存器
52
11
0
50
8个写端口
偶数号寄存器
7
0
32个偶数号寄存器
52
11
0
12个读端口
总共16个写端口＋12个读端口

图 1

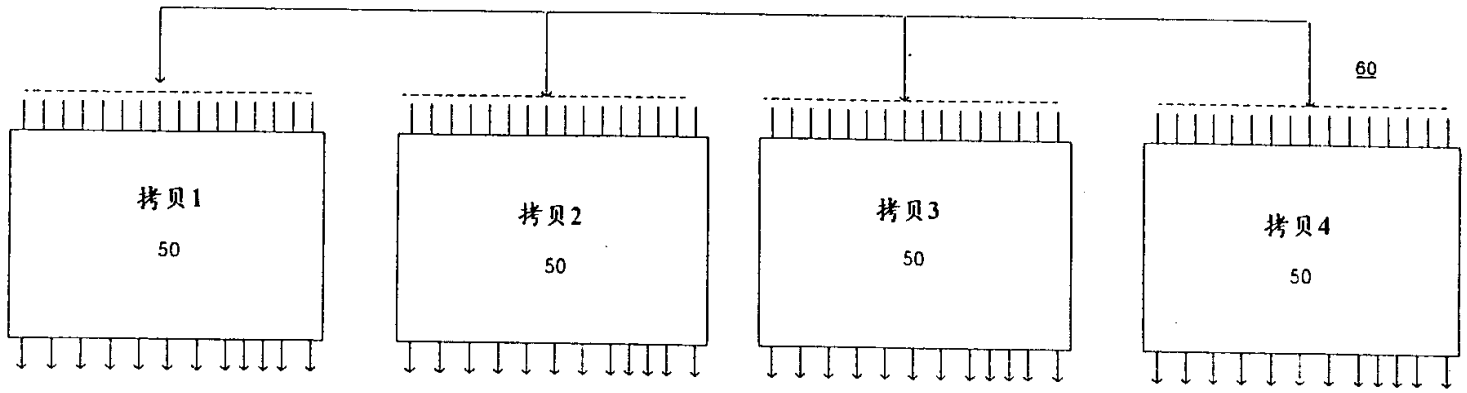


图 2

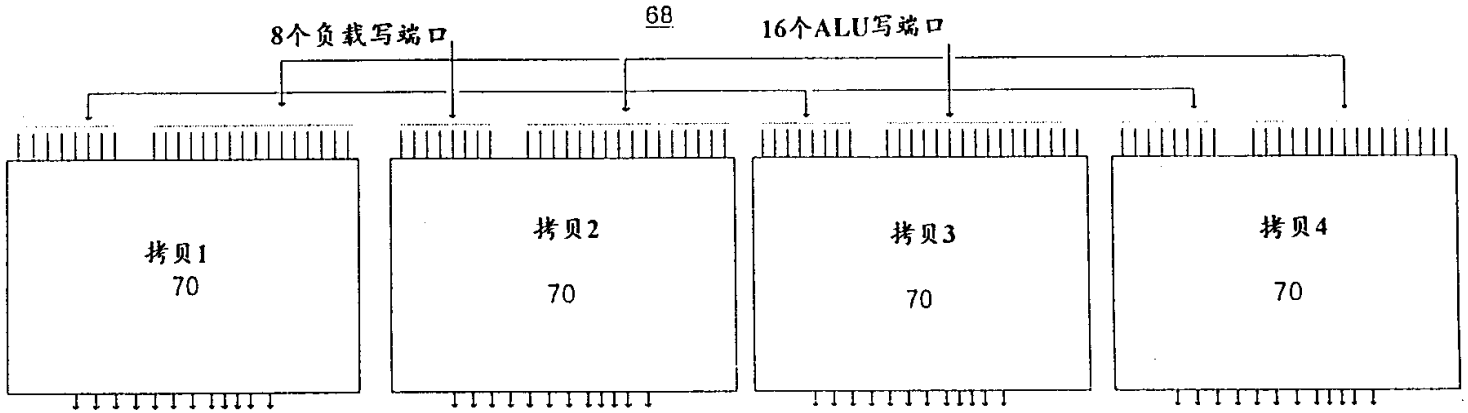


图 3

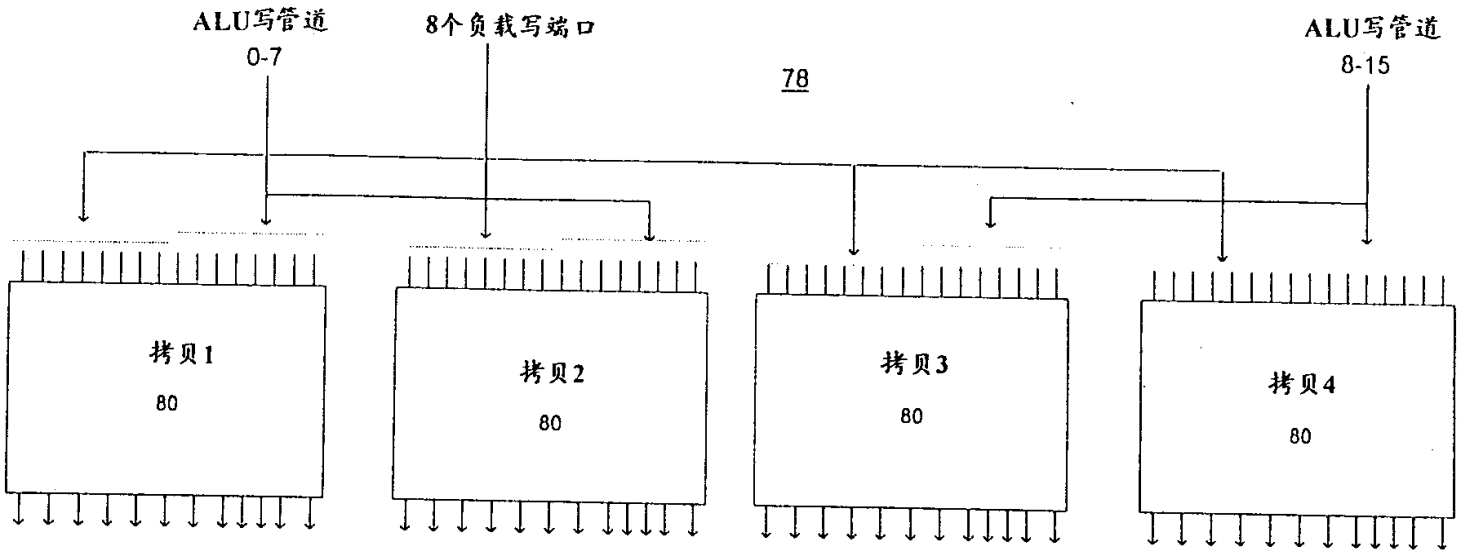


图 4

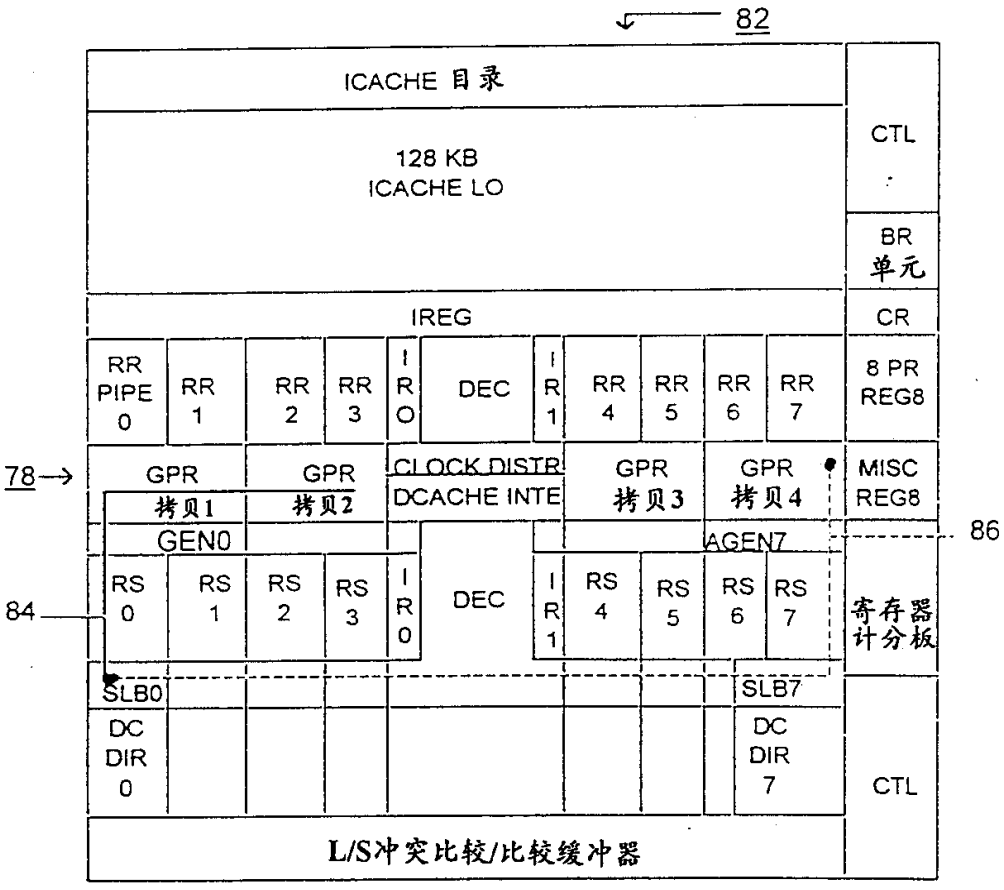

82
ICACHE 目录
128 KB
ICACHE LO
CTL
BR
单元
IREG
CR
RR
PIPE
0
RR
1
RR
2
RR
3
I
R
O
DEC
I
R
1
RR
4
RR
5
RR
6
RR
7
8 PR
REG8
78
GPR
拷贝1
GPR
拷贝2
CLOCK DISTR
DCACHE INTE
GPR
拷贝3
GPR
拷贝4
MISC
REG8
GEN0
AGEN7
86
84
RS
0
RS
1
RS
2
RS
3
I
R
0
DEC
I
R
1
RS
4
RS
5
RS
6
RS
7
寄存器
计分板
SLB0
SLB7
DC
DIR
0
DC
DIR
7
CTL
L/S冲突比较/比较缓冲器

图 5

图 6

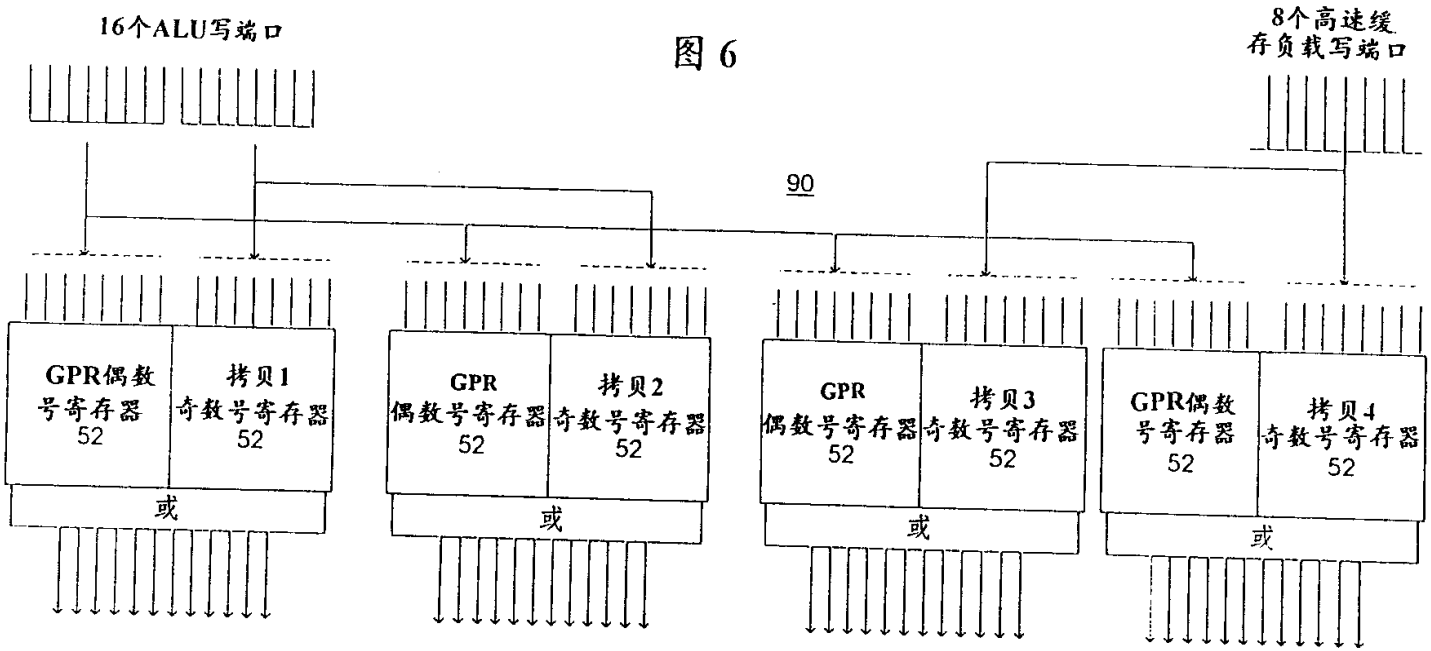

16个ALU写端口
8个高速缓存负载写端口
90
GPR偶数号寄存器 52
拷贝1 奇数号寄存器 52
或
GPR 偶数号寄存器 52
拷贝2 奇数号寄存器 52
或
GPR 偶数号寄存器 52
拷贝3 奇数号寄存器 52
或
GPR偶数号寄存器 52
拷贝4 奇数号寄存器 52
或

图 7

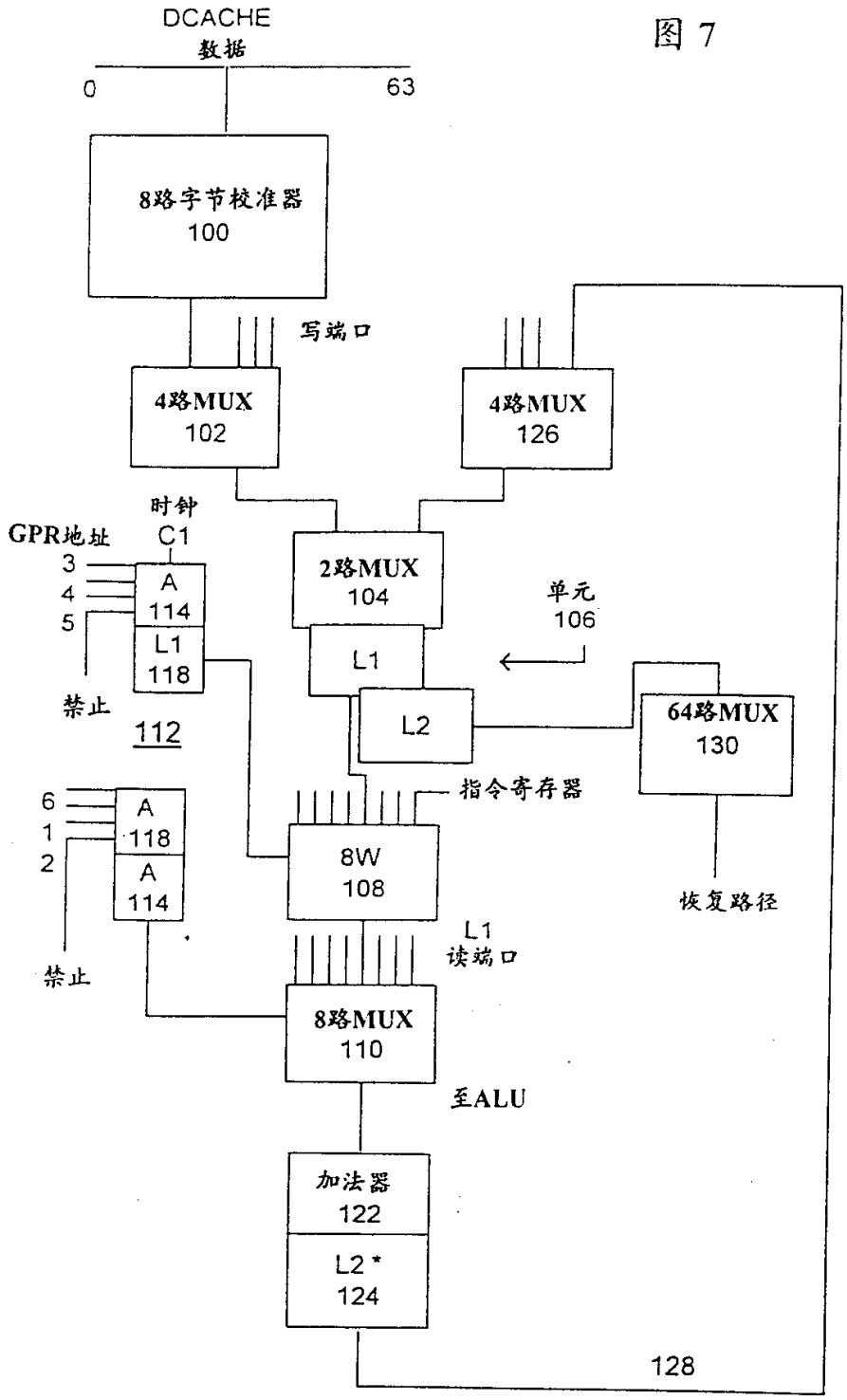

DCACHE
数据
0
63
8路字节校准器
100
写端口
4路MUX
102
4路MUX
126
时钟
C1
GPR地址
3
4
5
A
114
L1
118
禁止
112
2路MUX
104
单元
106
L1
L2
64路MUX
130
6
1
2
A
118
A
114
禁止
指令寄存器
8W
108
恢复路径
L1
读端口
8路MUX
110
至ALU
加法器
122
L2 *
124
128

图 8

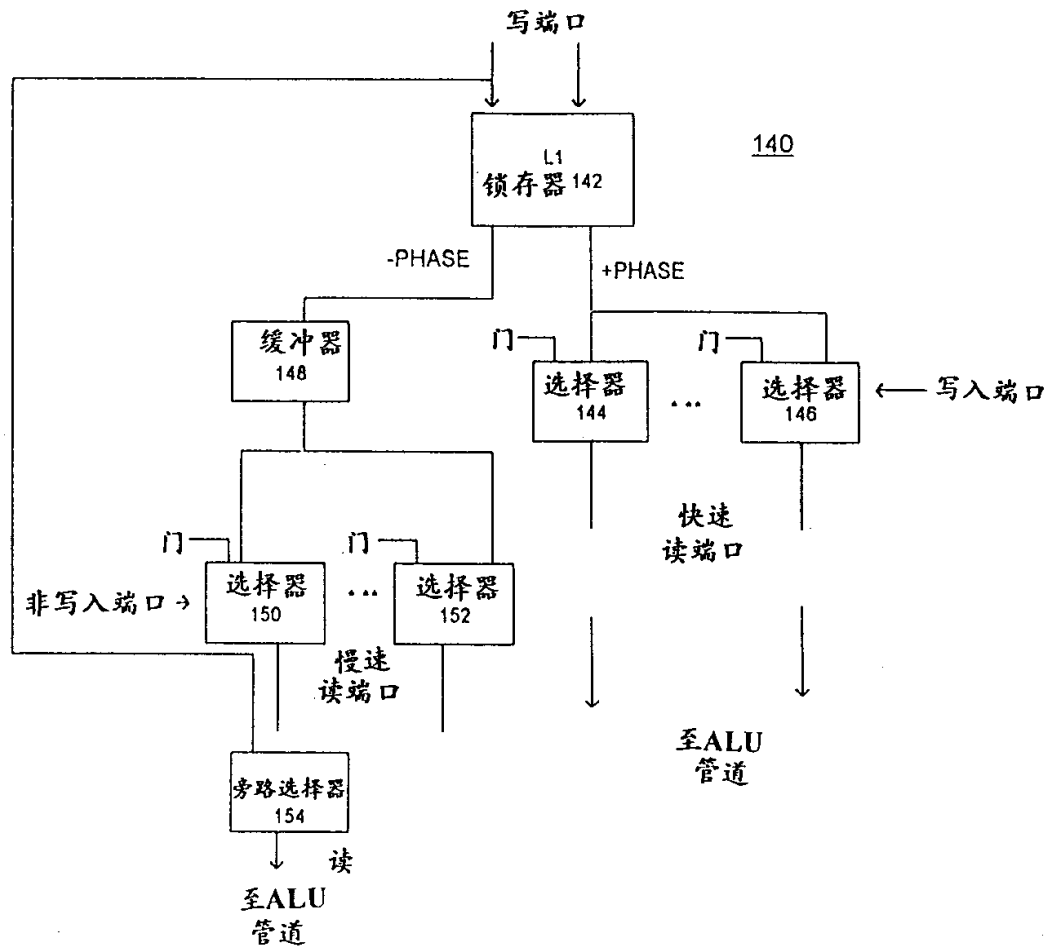

写端口
L1
锁存器 142
140
-PHASE
+PHASE
缓冲器
148
门
选择器
144
...
门
选择器
146
写入端口
快速
读端口
门
门
非写入端口
选择器
150
...
选择器
152
慢速
读端口
至ALU
管道
旁路选择器
154
读
至ALU
管道

图 9

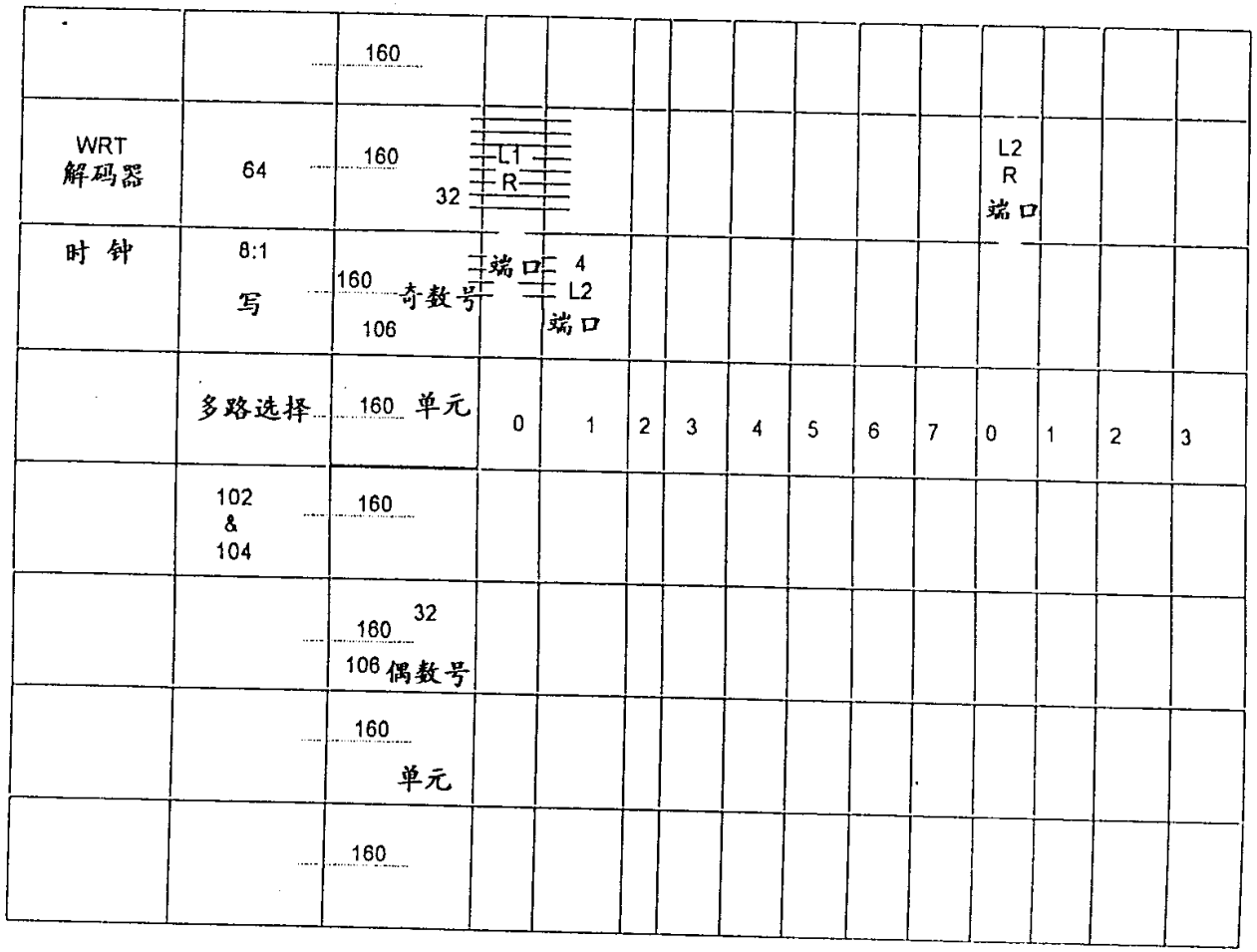
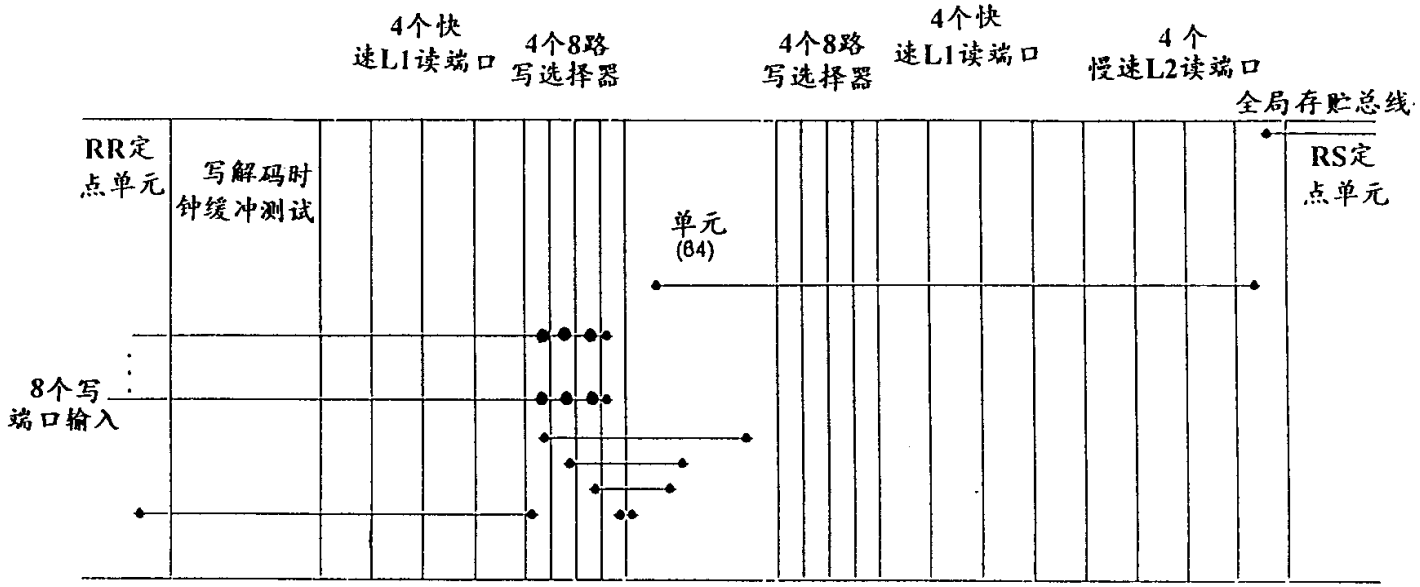

160
WRT
解码器
64
160
32
L1
R
L2
R
端口
时 钟
8:1
写
160
106
奇数号
端口
4
L2
端口
多路选择
160
单元
0
1
2
3
4
5
6
7
0
1
2
3
102
&
104
160
32
160
106
偶数号
160
单元
160

图 10


4个快
速L1读端口
4个8路
写选择器
4个8路
写选择器
4个快
速L1读端口
4 个
慢速L2读端口
全局存贮总线
RR定
点单元
写解码时
钟缓冲测试
单元
(84)
RS定
点单元
8个写
端口输入

I L0 DIR	DDIR SLB 0	1	2	3	CTL	4	5	6	DDIR SLB 7
L0 ICACHE	RS 管道 0			RS 管道 3	IR DEC CTL	RS 管道 4			RS 管道 5
78 ⟶	GPR 偶数号 1/2 拷贝1	GPR 奇数号 1/2 拷贝1	GPR 偶数号 1/2 拷贝2	GPR 奇数号 1/2 拷贝2	DBUS R/DR	GPR 偶数号 1/2 拷贝3	GPR 奇数号 1/2 拷贝3	GPR 偶数号 1/2 拷贝4	GPR 奇数号 1/2 拷贝4
	RR 管道 0				IR DEC				RR 管道 7
					CTL				

图 11