



# (12)发明专利

(10)授权公告号 CN 104951280 B

(45)授权公告日 2018. 10. 12

(21)申请号 201510010671.0

(22)申请日 2015.01.09

(65)同一申请的已公布的文献号

申请公布号 CN 104951280 A

(43)申请公布日 2015.09.30

(30)优先权数据

2014-063125 2014.03.26 JP

(73)专利权人 株式会社巨晶片

地址 日本大阪府大阪市

(72)发明人 野本祥平 水野雄介

(74)专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 秦琳 徐红燕

(51)Int.Cl.

G06F 9/38(2006.01)

G06F 15/80(2006.01)

(56)对比文件

CN 101324839 A, 2008.12.17,

US 2014/0032624 A1, 2014.01.30,

US 2014/0052960 A1, 2014.02.20,

审查员 邢白灵

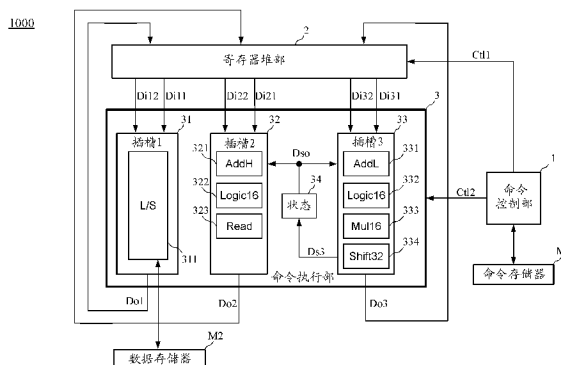
权利要求书2页 说明书16页 附图13页

## (54)发明名称

VLIW处理器

## (57)摘要

本发明涉及VLIW处理器。实现一种能够抑制电路规模的增大并且即使在包括像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算的情况下也能够高效率地执行处理的VLIW处理器。VLIW处理器(1000)具备命令控制部(1)、寄存器堆部(2)、以及命令执行部(3)。命令执行部(3)具有多个插槽,在第2插槽(32)与第3插槽(33)之间设置有用在两插槽间交接N比特数据的状态寄存器(34)。通过在该状态寄存器(34)中储存从第3插槽输出的数据并进行利用,从而能够抑制电路规模的增大并且即使是像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算也能够高效率地进行处理。



1. 一种VLIW处理器,具备:

寄存器堆部,包括多个寄存器;以及

命令执行部,包括第1插槽和能储存N比特的量的数据的状态寄存器,其中,N为自然数,  
所述第1插槽包括:

$N \times 2$ 比特的输入端口,用于输入来自所述寄存器堆部的输出数据;

N比特的第1输出端口,用于向所述寄存器堆部输出数据;

N比特的第2输出端口,用于向所述状态寄存器输出数据;以及

第1插槽用第1扩展运算单元,通过对N比特数据进行运算处理,从而取得 $2 \times N$ 比特数据的输出数据,

将由所述第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的所述输出数据中的N比特的量的数据作为第1数据从所述第1输出端口输出到所述寄存器堆部,

将由所述第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的所述输出数据中的除所述第1数据以外的N比特的量的数据作为第2数据输出到所述状态寄存器,

所述命令执行部还具备第2插槽,所述第2插槽包括:

$N \times 2$ 比特的输入端口,用于输入来自所述寄存器堆部的输出数据;

N比特的输出端口,用于向所述寄存器堆部输出数据;以及

第2插槽用第1运算单元,对N比特数据进行处理的运算,

所述第1插槽输入从所述第1插槽输出到所述寄存器堆部并由所述寄存器堆部保持的所述第1数据,

所述状态寄存器将从所述第1插槽输出的所述第2数据输出到所述第2插槽。

2. 根据权利要求1所述的VLIW处理器,其中,

所述第1插槽还包括N比特的输入端口,其输入从所述第1插槽输出到所述状态寄存器并由所述状态寄存器保持的所述第2数据。

3. 根据权利要求1所述的VLIW处理器,其中,

所述第2插槽还包括读出N比特数据的读出部,

所述第2插槽的所述读出部读出从所述状态寄存器输出的数据,将读出的数据输出到所述寄存器堆部。

4. 根据权利要求1或3所述的VLIW处理器,其中,

所述第2插槽还包括第2插槽用选择部,选择从所述寄存器堆部输出的数据和从所述状态寄存器输出的数据的任一方数据,使所选择的数据输入到所述第2插槽用第1运算单元。

5. 根据权利要求4所述的VLIW处理器,其中,

所述第1插槽还包括对N比特数据进行处理的运算的第1插槽用第1运算单元,

所述第1插槽用第1扩展运算单元是对2个N比特数据进行乘法运算处理的第1插槽用乘法运算单元,

所述第1数据是所述第1插槽用乘法运算单元通过乘法运算处理取得的 $2 \times N$ 比特数据中的下位N比特的数据,

所述第2数据是所述第1插槽用乘法运算单元通过乘法运算处理取得的 $2 \times N$ 比特数据中的上位N比特的数据,

所述第1插槽用第1运算单元是对2个N比特数据进行加减法运算处理的第1插槽用加减

法运算单元，

所述第2插槽用第1运算单元是对2个N比特数据进行加减法运算处理的第2插槽用加减法运算单元。

6. 根据权利要求1或3的任一项所述的VLIW处理器，其中，  
所述状态寄存器将保持的N比特数据输出到所述第1插槽，  
所述第1插槽还包括：

第1插槽用第2扩展运算单元，通过对N比特数据进行运算处理，从而取得 $2 \times N$ 比特数据的输出数据；

第1插槽用第1运算单元，对N比特数据进行处理的运算；以及

第1插槽用选择部，选择从所述寄存器堆部输出的数据和从所述状态寄存器输出的数据的任一方数据，使所选择的数据输入到所述第1插槽用第2扩展运算单元。

7. 根据权利要求6所述的VLIW处理器，其中，

所述第1插槽用第2扩展运算单元是对2个N比特数据进行位移运算而取得 $2 \times N$ 比特数据的输出数据的第1插槽用位移运算单元。

8. 根据权利要求7所述的VLIW处理器，其中，

所述第1插槽用位移运算单元将位移运算的处理对象的数据的上位N比特数据作为从所述第1插槽用选择部输出的数据来进行位移运算。

9. 根据权利要求5所述的VLIW处理器，其中，

所述寄存器堆部具有储存通过所述第1插槽用加减法运算单元的加减法运算处理取得的进位标记的区域，

所述第2插槽用加减法运算单元使用在所述寄存器堆部中储存的所述进位标记来对2个N比特数据进行加减法运算处理。

10. 根据权利要求5所述的VLIW处理器，其中，

所述第2插槽用加减法运算单元将加减法运算处理的处理对象的一方的N比特数据作为从所述第2插槽用选择部输出的数据来进行加减法运算处理。

11. 根据权利要求1至3的任一项所述的VLIW处理器，其中，

所述命令执行部还包括第3插槽，所述第3插槽包括：

$N \times 2$ 比特的输入端口，用于输入来自所述寄存器堆部的输出数据；

N比特的输出端口，用于向所述寄存器堆部输出数据；以及

加载/存储单元，进行从规定的存储器取得数据的加载处理和将数据储存在所述规定的存储器中的存储处理的至少一方处理。

12. 根据权利要求1至3的任一项所述的VLIW处理器，其中，

所述第1数据是由所述第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的所述输出数据中的下位N比特的量的数据，

所述第2数据是由所述第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的所述输出数据中的上位N比特的量的数据。

## VLIW处理器

### 技术领域

[0001] 本发明涉及一种采用VLIW(Very long instruction word:超长指令字)架构(architecture)的处理器(VLIW处理器)。

### 背景技术

[0002] 为了对像图像数据那样的大容量数据高效率地进行运算处理,正在开发多种多样的处理器技术。

[0003] 例如,在专利文献1中有如下这样的图像处理处理器的公开,所述图像处理处理器是图形处理用的图像处理处理器,所述图像处理处理器具备:加载/存储单元、多个运算单元、以及用于在多个运算单元之间使某个运算单元的运算结果输入到另一个运算单元的切换通道(channel)。

[0004] 可是,在以往的处理器的中,在执行图像处理、图像识别处理的情况下,主要执行以8比特或16比特为单位的运算。近年来,伴随着图像处理、图像识别处理的高度化/复杂化,在用于执行图像处理、图像识别处理的处理器中,执行32比特单位的运算的比例不断大幅增加。

[0005] 与此伴随地,要求不仅能应对以8比特或16比特为单位的运算、还能应对以32比特为单位的运算的处理器(VLIW处理器)。

[0006] 现有技术文献

[0007] 专利文献

[0008] 专利文献1:日本特开2003-216943号公报。

[0009] 发明要解决的课题

[0010] 例如,在使用专利文献1的技术来构成不仅能应对以16比特为单位的运算、还能应对以32比特为单位的运算的处理器(图13所示)的情况下,考虑如图13所示的结构。

[0011] 图13是使用现有技术来构成的也能应对以32比特为单位的运算的处理器900的概略结构图。

[0012] 如图13所示,处理器900具备命令控制部91、切换通道92、命令执行部93、命令存储器M91、以及数据存储器M92。

[0013] 命令控制部91从命令存储器M91读出命令(进行命令取出处理),对读出的命令进行解码(进行命令解码处理)。然后,命令控制部91基于命令解码处理的结果来进行对切换通道92以及命令执行部93的控制。

[0014] 命令执行部93为了在1个周期(1个时钟周期)中同时执行多个命令而具备能在1个周期中同时进行运算的多个命令插槽(slot)。如图13所示,命令执行部93具有3个插槽,即,第1插槽931、第2插槽932、以及第3插槽933。

[0015] 第1插槽931具有对数据存储器M92进行数据的加载/存储的加载/存储单元。

[0016] 第2插槽932具有进行32比特运算的加法运算单元(在图13中用“Add32”示出的单元)、进行16比特运算的逻辑运算单元(在图13中用“Logic16”示出的单元)、进行32比特运

算的位移运算单元(在图13中用“Shift32”示出的单元)。

[0017] 第3插槽933具有进行16比特运算的加法运算单元(在图13中用“Add16”示出的单元)、进行16比特运算的逻辑运算单元(在图13中用“Logic16”示出的单元)、进行16比特运算的乘法运算单元(在图13中用“Mul16”示出的单元)。

[0018] 命令存储器M91是存储用于在处理器900中执行运算的命令等的存储装置。

[0019] 数据存储器M92是能够存储用于在处理器900中执行运算的数据等的存储装置。

[0020] 从图13可知,在处理器900中,在第2插槽932设置有进行32比特运算的单元,因此,需要从切换通道92向第2插槽932发送2个的量的32比特的数据。例如,在用加法运算单元Add16进行32比特数据的加法运算的情况下,需要2个32比特数据。因此,在处理器900中,如图13所示,在切换通道92与第2插槽932之间需要4个(数据通路Di92~Di95)16比特数据的转发用通路。也就是说,在处理器900中,在切换通道92与第2插槽932之间需要64比特的量的数据通路。

[0021] 此外,在第2插槽932中执行32比特运算的情况下,其输出结果为32比特数据,因此,为了将该输出结果从第2插槽932发送到切换通道92,需要32比特的量的数据通路。在图13中,由于利用16比特的数据转发用的数据通路Do92、Do93从第2插槽932向切换通道92进行发送,所以确保了32比特的量的数据通路。

[0022] 此外,在处理器900中,在第3插槽933设置有输出结果为32比特的进行16比特数据的乘法运算的乘法运算单元Mul16,因此,为了将该输出结果从第3插槽933发送到切换通道92,需要32比特的量的数据通路。在图13中,由于利用16比特的数据转发用的数据通路Do94、Do95从第3插槽933向切换通道92进行发送,所以确保了32比特的量的数据通路。

[0023] 像这样,在使用现有技术来构成也能应对以32比特为单位的运算的处理器900的情况下,必须设置在切换通道92与命令执行部93之间的输入/输出端口数变多,导致电路规模的增大。此外,虽然也考虑将切换通道92例如置换为通用寄存器堆(register file)的结构,但是,在该情况下,必须设置在通用寄存器堆与命令执行部93之间的输入/输出端口数也会变多,导致电路规模的增大。

## 发明内容

[0024] 因此,鉴于上述问题,本发明的目的在于,实现一种能够抑制电路规模的增大并且即使在包括像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算的情况下也能够高效率地执行处理的VLIW处理器。

[0025] 用于解决课题的方案

[0026] 为了解决上述课题,第1发明是具备寄存器堆部和命令执行部的VLIW处理器。

[0027] 命令执行部包括第1插槽和能够储存N比特的量(N:自然数)的数据的状态寄存器(state register)。

[0028] 第1插槽包括:用于输入来自寄存器堆部的输出数据的 $N \times 2$ 比特(N:自然数)的输入端口;用于向寄存器堆部输出数据的N比特的第1输出端口;以及用于向状态寄存器输出数据的N比特的第2输出端口。此外,第1插槽包括第1插槽用第1扩展运算单元,所述第1插槽用第1扩展运算单元通过对N比特数据进行运算处理,从而取得 $2 \times N$ 比特数据的输出数据。

[0029] 而且,第1插槽将由第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的输出数据

中的N比特的量的数据作为第1数据从第1输出端口输出到寄存器堆部,将由第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的输出数据中的除第1数据以外的N比特的量的数据作为第2数据输出到所述状态寄存器。

[0030] 在该VLIW处理器中,第1插槽将由第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据作为划分成各N比特的2个数据输出到寄存器堆部和状态寄存器。也就是说,在该VLIW处理器中,即使在执行处理对象为 $2 \times N$ 比特数据的 $2 \times N$ 比特运算的情况下,该 $2 \times N$ 比特数据中的N比特数据也能够由状态寄存器保持,例如,在下一个周期(时钟周期)中由规定的运算单元进行利用。

[0031] 此外,在该VLIW处理器中,在第1插槽中,来自寄存器堆部的输入端口是 $N \times 2$ 比特( $N \times 2$ 比特的量的输入端口),向寄存器堆部的输出端口是N比特(N比特的量的输出端口),因此,与VLIW处理器只执行N比特运算的情况相比,不需要增加输入/输出端口数。

[0032] 像这样,在VLIW处理器中,如上所述,能够在不增加寄存器堆部的输入/输出端口数的情况下支持 $2 \times N$ 比特的运算。

[0033] 因此,在VLIW处理器中,能够抑制电路规模的增大,并且,即使在包括像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算( $N \times 2$ 比特运算)的情况下,也能够高效率地执行处理。

[0034] 第2发明如第1发明,第1插槽输入从第1插槽输出到状态寄存器并由状态寄存器保持的第2数据。

[0035] 由此,在该VLIW处理器中,例如,能够在当前的周期中使在前一个周期中从第1插槽输出的N比特数据从状态寄存器输入到第1插槽以用于第1插槽中的运算。

[0036] 第3发明如第1发明,命令执行部还具备第2插槽。

[0037] 第2插槽包括:用于输入来自寄存器堆部的输出数据的 $N \times 2$ 比特(N:自然数)的输入端口;用于向寄存器堆部输出数据的N比特的输出端口;以及对N比特数据进行处理运算的第2插槽用第1运算单元。

[0038] 而且,第1插槽输入从第1插槽输出到所述寄存器堆部并由寄存器堆部保持的所述第1数据。

[0039] 状态寄存器将从第1插槽输出的第2数据输出到第2插槽。

[0040] 由此,在该VLIW处理器中,例如,能够在当前的周期中使在前一个周期中从第1插槽输出的N比特数据从状态寄存器输入到第2插槽以用于第2插槽中的运算。

[0041] 第4发明如第3发明,第2插槽还包括读出N比特数据的读出部。

[0042] 第2插槽的读出部读出从状态寄存器输出的数据,将读出的数据输出到寄存器堆部。

[0043] 在该VLIW处理器中,能够利用第2插槽的读出部读出在状态寄存器中保持的N比特数据并在规定的定时输出到寄存器堆部,因此,即使存在处理 $2 \times N$ 比特数据的运算,也能够高效率地向寄存器堆部输出规定的数据并在寄存器堆部中保持所需的数据。

[0044] 第5发明如第3或第4发明,第2插槽还包括第2插槽用选择部,所述第2插槽用选择部选择从寄存器堆部输出的数据和从状态寄存器输出的数据的任一方数据,使所选择的数据输入到第2插槽用第1运算单元。

[0045] 由此,在该VLIW处理器中,能够选择从寄存器堆部输出的数据和从状态寄存器输

出的数据的任一方并使其输入到第2插槽用第1运算单元,因此,能够将所需的数据输入到第2插槽用第1运算单元。例如,在第2插槽用第1运算单元是进行加减法运算处理的运算单元的情况下,在将该运算单元用于N比特数据的加减法运算处理的情况下,能够利用第1选择部输入从寄存器堆部输出的2个N比特数据。另一方面,在将上述运算单元用于 $2 \times N$ 比特数据的加减法运算处理的上位N比特的量的加减法运算处理的情况下,只要在状态寄存器中保持 $2 \times N$ 比特数据中的上位N比特的量的数据(例如,由第1插槽的加法运算单元取得的N比特的量的数据),利用第2插槽用选择部向上述运算单元输入状态寄存器的输出,进而,从寄存器堆部输入与该上位N比特的量的数据进行加法运算的 $2 \times N$ 比特数据的上位N比特数据即可。由此,能够在该运算单元中执行 $2 \times N$ 比特数据的加减法运算处理的上位N比特数据的加减法运算处理。

[0046] 第6发明如第5发明,第1插槽还包括对N比特数据进行处理的运算的第1插槽用第1运算单元。

[0047] 第1插槽用第1扩展运算单元是对2个N比特数据进行乘法运算处理的第1插槽用乘法运算单元。

[0048] 第1数据是第1插槽用乘法运算单元通过乘法运算处理取得的 $2 \times N$ 比特数据中的下位N比特的数据。

[0049] 第2数据是第1插槽用乘法运算单元通过乘法运算处理取得的 $2 \times N$ 比特数据中的上位N比特的数据。

[0050] 第1插槽用第1运算单元是对2个N比特数据进行加减法运算处理的第1插槽用加减法运算单元。

[0051] 第2插槽用第1运算单元是对2个N比特数据进行加减法运算处理的第2插槽用加减法运算单元。

[0052] 由此,能够对第1插槽分配进行N比特的乘法运算的单元(乘法运算单元)和进行N比特的加减法运算处理的单元(下位N比特数据的加法运算单元),能够对第2插槽分配进行N比特的加减法运算处理的单元(上位N比特数据的加法运算单元)。

[0053] 第7发明如第3至第6的任一项发明,状态寄存器将保持的N比特数据输出到第1插槽。

[0054] 第1插槽还包括第1插槽用第2扩展运算单元、第1插槽用第1运算单元、以及第1插槽用选择部。

[0055] 第1插槽用第2扩展运算单元通过对N比特数据进行运算处理,从而取得 $2 \times N$ 比特数据的输出数据。

[0056] 第1插槽用第1运算单元对N比特数据进行处理的运算。

[0057] 第1插槽用选择部选择从寄存器堆部输出的数据和从状态寄存器输出的数据的任一方数据,使所选择的数据输入到第1插槽用第2扩展运算单元。

[0058] 由此,在该VLIW处理器中,例如,能够在当前的周期中使在前一个周期中从第1插槽输出的N比特数据从状态寄存器输入到第1插槽的第1插槽用第2扩展运算单元以用于该运算单元中的运算。

[0059] 第8发明如第7发明,第1插槽用第2扩展运算单元是对2个N比特数据进行位移运算而取得 $2 \times N$ 比特数据的输出数据的第1插槽用位移运算单元。

[0060] 由此,在该VLIW处理器中,能够利用第1插槽用位移运算单元例如使用从状态寄存器输出的N比特数据和从寄存器堆部输出的N比特数据来执行位移运算处理。

[0061] 第9发明如第8发明,第1插槽用位移运算单元将位移运算的处理对象的数据的上位N比特数据作为从第1插槽用选择部输出的数据来进行位移运算。

[0062] 由此,在该VLIW处理器中,能够利用第1插槽用位移运算单元例如将从状态寄存器输出的N比特数据作为上位N比特数据并将从寄存器堆部输出的N比特数据作为下位N比特数据来执行位移运算处理。

[0063] 第10发明如第6发明,寄存器堆部具有储存通过第1插槽用加减法运算单元的加减法运算处理取得的进位标记的区域。

[0064] 第2插槽用加减法运算单元使用在寄存器堆部中储存的进位标记来对2个N比特数据进行加减法运算处理。

[0065] 由此,在该VLIW处理器中,例如,第2插槽用加减法运算单元能够使用通过第1插槽的运算单元中的加减法运算处理而生成的进位标记、即储存在寄存器堆部中的进位标记来执行对2个N比特数据的加减法运算处理。

[0066] 第11发明如第6发明,第2插槽用加减法运算单元将加减法运算处理的处理对象的一方的N比特数据作为从第2插槽用选择部输出的数据来进行加减法运算处理。

[0067] 由此,在第2插槽中也能够利用在状态寄存器中保持的数据,能够高效率地执行N比特数据的运算处理(例如,N比特数据的加减法运算处理)和 $2 \times N$ 比特的运算处理(扩展运算处理(例如, $2 \times N$ 比特数据的加减法运算处理))。

[0068] 第12发明如第1至第11的任一项发明,命令执行部还具备第3插槽。

[0069] 第3插槽包括:用于输入来自寄存器堆部的输出数据的 $N \times 2$ 比特(N:自然数)的输入端口;用于向寄存器堆部输出数据的N比特的输出端口;以及进行从规定的存储器取得数据的加载处理和将数据储存在所述规定的存储器中的存储处理的至少一方处理的加载/存储单元。

[0070] 由此,能够实现具备包括加载/存储单元的第3插槽的VLIW处理器。

[0071] 第13发明如第1至第12的任一项发明,第1数据是由第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的输出数据中的下位N比特的量的数据。

[0072] 第2数据是由第1插槽用第1扩展运算单元取得的 $2 \times N$ 比特数据的输出数据中的上位N比特的量的数据。

[0073] 由此,在该VLIW处理器中,能够使第1数据(输出到寄存器堆部的数据)为下位N比特数据并且使第2数据(输出到状态寄存器的数据)为上位N比特数据来高效率地执行运算处理。

[0074] 发明效果

[0075] 根据本发明,能实现一种能够抑制电路规模的增大并且即使在包括像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算的情况下也能够高效率地执行处理的VLIW处理器。

## 附图说明

[0076] 图1是第1实施方式的VLIW处理器1000的概略结构图。

- [0077] 图2是命令执行部3的结构例(一个例子)的概略图。
- [0078] 图3是示出了MulShift处理中的分配给各插槽的运算处理的图、以及明示了在周期2中被激活的VLIW处理器1000的部分的图。
- [0079] 图4是示出了MulShift处理中的分配给各插槽的运算处理的图、以及明示了在周期3中被激活的VLIW处理器1000的部分的图。
- [0080] 图5是示出了MulShift处理中的分配给各插槽的运算处理的图、以及明示了在周期4中被激活的VLIW处理器1000的部分的图。
- [0081] 图6是示出了MulAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期3中被激活的VLIW处理器1000的部分的图。
- [0082] 图7是示出了MulAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期4中被激活的VLIW处理器1000的部分的图。
- [0083] 图8是示出了MulShiftAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期4中被激活的VLIW处理器1000的部分的图。
- [0084] 图9是示出了MulShiftAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期5中被激活的VLIW处理器1000的部分的图。
- [0085] 图10是示出了ShiftAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期2中被激活的VLIW处理器1000的部分的图。
- [0086] 图11是示出了ShiftAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期3中被激活的VLIW处理器1000的部分的图。
- [0087] 图12是示出了ShiftAdd处理中的分配给各插槽的运算处理的图、以及明示了在周期4中被激活的VLIW处理器1000的部分的图。
- [0088] 图13是使用现有技术来构成的也能应对以32比特为单位的运算的处理器900的概略结构图。

## 具体实施方式

- [0089] [第1实施方式]
- [0090] 以下,针对第1实施方式,一边参照附图一边进行说明。
- [0091] <1.1:VLIW处理器的结构>
- [0092] 图1是第1实施方式的VLIW处理器1000的概略结构图。
- [0093] VLIW处理器1000是进行以N比特(N:自然数)为单位的运算和以 $2 \times N$ 比特为单位的运算的VLIW处理器。
- [0094] 再有,以下,为了便于说明,对 $N=16$ 、也就是说VLIW处理器1000是进行以16比特为单位的运算和以32比特为单位的运算的VLIW处理器的情况进行说明。
- [0095] 如图1所示,VLIW处理器1000具备命令控制部1、寄存器堆部2、命令执行部3、命令存储器M1、以及数据存储器M2。
- [0096] 命令控制部1从命令存储器M1读出命令(进行命令取出处理),对读出的命令进行解码(进行命令解码处理)。而且,命令控制部1基于命令解码处理的结果生成用于控制寄存器堆部2的控制信号Ct11,将生成的控制信号Ct11输出到寄存器堆部2。
- [0097] 此外,命令控制部1基于命令解码处理的结果生成用于控制命令执行部3的控制信

号Ct12,将生成的控制信号Ct12输出到命令执行部3。

[0098] 寄存器堆部2具有多个寄存器,基于来自命令控制部1的控制信号Ct11将规定的寄存器的数据输出到命令执行部3的规定的插槽。此外,寄存器堆部2基于控制信号Ct11以将从命令执行部3输出的数据输入到规定的寄存器的方式进行控制。

[0099] 命令执行部3为了在1个周期(1个时钟周期)中执行多个命令而具备能够在1个周期中同时进行运算的多个命令插槽。以下,为了便于说明,对命令插槽数为“3”的情况进行说明。

[0100] 如图1所示,命令执行部3具有3个插槽,即,第1插槽31、第2插槽32、以及第3插槽33。此外,命令执行部3具备状态寄存器34。

[0101] 第1插槽31具备对数据存储器M2进行16比特的数据的加载/存储的加载/存储单元311。

[0102] 第2插槽32具备:进行16比特数据的加法运算处理的加法运算单元321;对16比特数据进行逻辑运算的逻辑运算单元322;以及从状态寄存器34读出16比特数据的状态读出单元323。再有,“加法运算处理”也可以包括减法运算处理。也就是说,“加法运算处理”也可以是加减法运算处理(以下相同)。

[0103] 第3插槽33具备:进行16比特数据的加法运算处理的加法运算单元331;对16比特数据进行逻辑运算的逻辑运算单元332;进行16比特数据的乘法运算的乘法运算单元333;以及对32比特数据进行比特位移处理的位移运算单元334。

[0104] 状态寄存器34输入从第3插槽33输出的16比特数据并存储保持所输入的16比特数据。状态寄存器34将保持的16比特数据输出到第2插槽32和第3插槽33。

[0105] 在此,使用图2对命令执行部3的结构例(一个例子)进行说明。

[0106] 图2是命令执行部3的结构例(一个例子)的概略图。

[0107] 如图2所示,命令执行部3的第2插槽32具备加法运算单元321、逻辑运算单元322、状态读出单元323、第1切换部324、以及第2切换部325。

[0108] 第1切换部324将从寄存器堆部2输出的16比特数据Di21和从状态寄存器34输出的16比特数据Dso作为输入。第1切换部324按照命令控制部1的指令(控制信号Ct12)将16比特数据Di21和16比特数据Dso的任一方数据输出到加法运算单元321。

[0109] 加法运算单元321将从寄存器堆部2输出的16比特数据Di22和从第1切换部324输出的16比特数据作为输入,对输入的2个16比特数据进行加法运算处理。然后,加法运算单元321将加法运算处理后的数据输出到第2切换部325。

[0110] 此外,加法运算单元321在执行32比特加法运算处理的上位16比特的加法运算处理的情况下将16比特数据Di22、从第1切换部324输出的16比特数据、以及例如第3插槽的加法运算单元331生成并在寄存器堆部2的进位寄存器(未图示)中储存的进位标记(进位比特(1比特数据))作为输入,对输入的3个数据进行加法运算处理。然后,加法运算单元321将加法运算处理后的数据输出到第2切换部325。

[0111] 逻辑运算单元322将从寄存器堆部2输出的2个16比特数据Di21、Di22作为输入,对输入的2个16比特数据进行规定的逻辑运算处理。然后,逻辑运算单元322将逻辑运算处理后的数据输出到第2切换部325。

[0112] 状态读出单元323读出在状态寄存器34中保持的16比特数据。然后,状态读出单元

323将从状态寄存器34读出的数据输出到第2切换部325。

[0113] 第2切换部325将来自加法运算单元321的输出、来自逻辑运算单元322的输出、以及来自状态读出单元323的输出作为输入。然后,第2切换部325按照命令控制部1的指令(控制信号Ct12)将来自加法运算单元321的输出、来自逻辑运算单元322的输出、以及来自状态读出单元323的输出的任一个作为数据Do2输出到寄存器堆部2。再有,第2切换部325在选择来自加法运算单元321的输出来向寄存器堆部2输出的情况下,可以将加法运算的进位标记包含在数据Do2中来向寄存器堆部2输出,此外,也可以使用另外的通路来将加法运算的进位标记输出到寄存器堆部2。

[0114] 如图2所示,命令执行部3的第3插槽33具备加法运算单元331、逻辑运算单元332、乘法运算单元333、位移运算单元334、第3切换部335、第4切换部336、以及第5切换部337。

[0115] 第3切换部335将从寄存器堆部2输出的16比特数据Di32和从状态寄存器34输出的16比特数据Dso作为输入。第3切换部335按照命令控制部1的指令(控制信号Ct12)将16比特数据Di32和16比特数据Dso的任一方数据输出到位移运算单元334。

[0116] 加法运算单元331将从寄存器堆部2输出的2个16比特数据Di31和Di32作为输入,对输入的2个16比特数据进行加法运算处理。然后,加法运算单元331将加法运算处理后的数据输出到第4切换部336。

[0117] 此外,加法运算单元331在执行32比特加法运算处理的下位16比特的加法运算处理的情况下将通过16比特数据Di31与16比特数据Di32的加法运算处理取得的进位标记(进位比特(1比特数据))输出到寄存器堆部2,例如,储存在寄存器堆部2的进位寄存器中(未对输出通路进行图示)。

[0118] 逻辑运算单元332将从寄存器堆部2输出的2个16比特数据Di31和Di32作为输入,对输入的2个16比特数据进行规定的逻辑运算处理。然后,逻辑运算单元332将逻辑运算处理后的数据输出到第4切换部336。

[0119] 乘法运算单元333将从寄存器堆部2输出的2个16比特数据Di31、Di32作为输入,对输入的2个16比特数据进行乘法运算处理。然后,乘法运算单元333将乘法运算处理后的32比特数据中的下位16比特数据输出到第4切换部336并且将上位16比特的数据输出到第5切换部337。

[0120] 位移运算单元334将从寄存器堆部2输出的16比特数据Di31和从第3切换部335输出的16比特数据作为输入,对输入的2个16比特数据进行位移运算处理(32比特位移运算处理)。然后,位移运算单元334将位移运算处理后的32比特数据中的下位16比特数据输出到第4切换部336并且将上位16比特的数据输出到第5切换部337。

[0121] 第4切换部336将来自加法运算单元331的输出、来自逻辑运算单元332的输出、从乘法运算单元333输出的乘法运算结果数据的下位16比特数据、以及从位移运算单元334输出的位移运算结果数据的下位16比特数据作为输入。然后,第4切换部336按照命令控制部1的指令(控制信号Ct12)选择来自上述4个单元的输出的任一个作为数据Do3输出到寄存器堆部2。

[0122] 再有,第4切换部336可以将通过加法运算单元331的加法运算处理取得的进位标记包含在数据Do3中来进行输出,此外,也可以使用另外的通路来将该进位标记输出到寄存器堆部2。

[0123] 第5切换部337将从乘法运算单元333输出的乘法运算结果数据的上位16比特数据以及从位移运算单元334输出的位移运算结果数据的上位16比特数据作为输入。然后,第5切换部337按照命令控制部1的指令(控制信号Ct12)选择来自上述2个单元的输出的任一个作为数据Ds3输出到状态寄存器34。

[0124] 状态寄存器34将从第3插槽33的第5切换部337输出的16比特数据作为输入,对输入的16比特数据进行存储保持。状态寄存器34将保持的16比特数据输出到第2插槽32的状态读出单元323及第1切换部324和第3插槽33的第3切换部335。

[0125] 命令存储器M1是能够存储用于在VLIW处理器1000中执行运算的命令等的存储装置。命令存储器M1是能够由命令控制部1进行访问的存储器。

[0126] 数据存储器M2是能够存储用于在VLIW处理器1000中执行运算的数据等的存储装置。数据存储器M2是能够由命令执行部3的第1插槽31的加载/存储单元311进行访问的存储器。

[0127] 再有,命令存储器M1和数据存储器M2也可以被1个存储器(存储装置)实现。

[0128] <1.2:VLIW处理器的工作>

[0129] 以下,针对像以上那样构成的VLIW处理器1000的工作,一边参照附图一边进行说明。

[0130] 以下,对如下情况进行说明:

[0131] (1)执行乘法运算处理、对乘法运算结果执行位移运算处理的情况(MulShift处理);

[0132] (2)执行乘法运算处理、对乘法运算结果执行加法运算处理的情况(MulAdd处理);

[0133] (3)执行乘法运算处理、对乘法运算结果执行位移运算处理,对位移运算结果执行加法运算处理的情况(MulShiftAdd处理);以及

[0134] (4)执行位移运算处理、对位移运算结果执行加法运算处理的情况(ShiftAdd处理)。

[0135] (1.2.1:MulShift处理)

[0136] 说明在VLIW处理器1000中对2个16比特数据执行乘法运算处理、对作为该乘法运算结果的32比特数据进行32比特位移的情况(将在该情况下的处理称为“MulShift处理”)。

[0137] 图3的左图是按每个周期示出在VLIW处理器1000执行MulShift处理的情况下分配给各插槽(第1插槽31、第2插槽32以及第3插槽33)的运算处理的图。此外,图3的右图是明示了在周期2(Cyc2)中被激活的部分的图。再有,被激活的部分用粗线示出(以下相同)。

[0138] 图4、图5也是与图3相同的图。

[0139] 图4的右图是明示了在周期3(Cyc3)中被激活的部分的图。

[0140] 图5的右图是明示了在周期4(Cyc4)中被激活的部分的图。

[0141] MulShift处理例如相当于对2个16比特数据进行乘法运算并对准固定小数点位置的处理。

[0142] (Cyc0):

[0143] 在周期0(Cyc0)中,命令控制部1通过控制信号Ct12对命令执行部3发布加载/存储命令,第1插槽31的加载/存储单元311基于该加载/存储命令从数据存储器M2读出由第3插

槽33的乘法运算单元333进行的乘法运算处理 (Mu116) 所需的2个16比特数据 (源操作数) 中的1个。然后, 第1插槽31将读出的16比特数据输出到寄存器堆部2。

[0144] 寄存器堆部2将从第1插槽31读出的16比特数据储存在规定的寄存器中。

[0145] (Cyc1):

[0146] 在周期1 (Cyc1) 中, 命令控制部1通过控制信号Ct12对命令执行部3发布加载/存储命令, 第1插槽31的加载/存储单元311基于该加载/存储命令从数据存储器M2读出由第3插槽33的乘法运算单元333进行的乘法运算处理 (Mu116) 所需的2个16比特数据 (源操作数) 中的另1个数据。然后, 第1插槽31将读出的16比特数据输出到寄存器堆部2。

[0147] 寄存器堆部2将从第1插槽31读出的16比特数据储存在规定的寄存器中。

[0148] (Cyc2):

[0149] 在周期2 (Cyc2) 中, 命令控制部1通过控制信号Ct12对命令执行部3发布乘法运算命令, 第3插槽33的乘法运算单元333变为能够执行的状态。此外, 寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期0、1中经由数据通路Di31、Di32将储存在规定的寄存器中的乘法运算处理对象的2个16比特数据输出到第3插槽33。

[0150] 第3插槽33的乘法运算单元333对经由数据通路Di31、Di32输入的2个16比特数据执行乘法运算处理。然后, 乘法运算单元333将乘法运算结果的32比特数据中的上位16比特的数据作为16比特数据Ds3经由第5切换部337输出到状态寄存器34。然后, 状态寄存器34储存从第3插槽33输出的16比特数据Ds3。

[0151] 此外, 乘法运算单元333将乘法运算结果的32比特数据中的下位16比特的数据作为16比特数据Do3经由第4切换部336输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。

[0152] (Cyc3):

[0153] 在周期3 (Cyc3) 中, 命令控制部1通过控制信号Ct12对命令执行部3发布位移运算命令, 第3插槽33的位移运算单元334变为能够执行的状态。此外, 寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期2中经由数据通路Di31将储存在规定的寄存器中的位移运算对象的16比特数据 (乘法运算结果的下位16比特数据) 输出到第3插槽33。

[0154] 位移运算单元334取得从寄存器堆部2经由数据通路Di31输入的16比特数据和从状态寄存器34输出的16比特数据 (乘法运算结果的上位16比特数据)。再有, 此时, 第3切换部335被控制为根据来自命令控制部1的指令选择来自状态寄存器34的输出来向位移运算单元334输出。

[0155] 位移运算单元334对像上述那样输入的由2个16比特数据构成的32比特数据执行位移运算处理。然后, 位移运算单元334将位移运算处理结果的32比特数据中的上位16比特的数据作为16比特数据Ds3经由第5切换部337输出到状态寄存器34。然后, 状态寄存器34储存从第3插槽33输出的16比特数据Ds3。

[0156] 此外, 位移运算单元334将位移运算处理结果的32比特数据中的下位16比特的数据作为16比特数据Do3经由第4切换部336输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。

[0157] (Cyc4):

[0158] 在周期4 (Cyc4) 中, 命令控制部1通过控制信号Ct12对命令执行部3发布读出命令,

第2插槽32的状态读出单元323变为能够执行的状态。状态读出单元323将周期3中的位移运算结果的上位16比特数据作为16比特数据Do2经由第2切换部325输出到寄存器堆部2。再有,此时,第2切换部325被控制为根据控制信号Ct12选择来自状态读出单元323的输出来进行输出。

[0159] 寄存器堆部2将所接收的上述16比特数据Do2储存在规定的寄存器中。

[0160] 此外,与上述同样地执行对另外的数据的乘法运算处理(由第3插槽的乘法运算单元333进行的乘法运算处理)。再有,如图5的右图所示,成为该乘法运算处理的处理对象的2个16比特数据在周期2、3中由第1插槽31的加载/存储单元311从数据存储器M2读出并储存在寄存器堆部2的规定的寄存器中。

[0161] 根据以上,在VLIW处理器1000中,能够对2个16比特数据执行乘法运算处理,能够执行对作为该乘法运算结果的32比特数据进行32比特位移的处理(MulShift处理)。

[0162] (1.2.2:MulAdd处理)

[0163] 接着,使用图6、图7来说明在VLIW处理器1000中对2个16比特数据执行乘法运算处理、将作为该乘法运算结果的32比特数据与其它32比特数据进行加法运算的情况(将在该情况下的处理称为“MulAdd处理”)。作为该MulAdd处理的一个例子,有2个16比特数据的乘法运算结果的累积加法运算处理。这样的处理多用在图像处理、图像识别处理等中。

[0164] 图6的左图是按每个周期示出在VLIW处理器1000执行MulAdd处理的情况下分配给各插槽(第1插槽31、第2插槽32以及第3插槽33)的运算处理的图。此外,图6的右图是明示了在周期3(Cyc3)中被激活的部分的图。

[0165] 图7是与图6相同的图。

[0166] 图7的右图是明示了在周期4(Cyc4)中被激活的部分的图。

[0167] (Cyc0~2):

[0168] 因为周期0~2(Cyc0~2)的处理与“1.2.1:MulShift处理”中的周期0~2的处理相同,所以省略详细的说明。

[0169] (Cyc3):

[0170] 在周期3(Cyc3)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算命令(AddL命令),第3插槽33的加法运算单元331变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期2中经由数据通路Di31将储存在规定的寄存器中的乘法运算结果的下位16比特数据输出到第3插槽33。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12经由数据通路Di32将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的下位16比特数据输出到第3插槽33。

[0171] 第3插槽33的加法运算单元331对经由数据通路Di31、Di32输入的2个16比特数据执行加法运算处理。然后,加法运算单元331将加法运算结果的16比特数据作为16比特数据Do3输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。再有,通过加法运算单元331中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0172] (Cyc4):

[0173] 在周期4(Cyc4)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算命令(AddH命令),第2插槽32的加法运算单元321变为能够执行的状态。此外,寄存器堆部2

基于来自命令控制部1的控制信号Ct12经由数据通路Di22将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的上位16比特数据输出到第2插槽32。

[0174] 此外,状态寄存器34经由第1切换部324将在周期2中储存的乘法运算结果(由乘法运算单元333得到的乘法运算结果的上位16比特数据)输出到加法运算单元321。此时,第1切换部324被控制为根据控制信号Ct12向加法运算单元321输出状态寄存器34的输出。

[0175] 第2插槽32的加法运算单元321对经由数据通路Di22输入的16比特数据、从第1切换部324输出的16比特数据(由乘法运算单元333得到的乘法运算结果的上位16比特数据)、以及通过加法运算单元331中的加法运算处理取得的进位比特(未对进位比特的输入通路进行图示)执行加法运算处理。然后,加法运算单元321将加法运算结果的16比特数据作为16比特数据Do2经由第2切换部325输出到寄存器堆部2。再有,此时,第2切换部325被控制为根据控制信号Ct12向寄存器堆部2输出加法运算单元321的输出。

[0176] 寄存器堆部2将所接收的该16比特数据Do2储存在规定的寄存器中。再有,通过加法运算单元321中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0177] 此外,在周期4中,命令控制部1通过控制信号Ct12对命令执行部3发布乘法运算命令,第3插槽33的乘法运算单元333变为能够执行的状态。然后,执行与周期2的第3插槽33的乘法运算单元333的处理相同的处理。也就是说,与上述同样地执行对另外的数据的乘法运算处理(由第3插槽的乘法运算单元333进行的乘法运算处理)。再有,该乘法运算处理的处理对象在周期2、3中由第1插槽31的加载/存储单元311从数据存储器M2读出并储存在寄存器堆部2的规定的寄存器中。

[0178] 根据以上,在VLIW处理器1000中,能够对2个16比特数据执行乘法运算处理,能够执行将作为该乘法运算结果的32比特数据与其它32比特数据进行加法运算的处理(MulAdd处理)。

[0179] (1.2.3:MulShiftAdd处理)

[0180] 接着,使用图8、图9来说明在VLIW处理器1000中对2个16比特数据执行乘法运算处理、对作为该乘法运算结果的32比特数据进行位移运算、将该位移运算结果与其它32比特数据进行加法运算的情况(将在该情况下的处理称为“MulShiftAdd处理”)。作为该MulShiftAdd处理的一个例子,有对2个16比特数据的乘法运算结果数据在对准了固定小数点位置之后进行累积加法运算的处理。这样的处理多用在图像处理、图像识别处理等中。

[0181] 图8的左图是按每个周期示出在VLIW处理器1000执行MulShiftAdd处理的情况下分配给各插槽(第1插槽31、第2插槽32以及第3插槽33)的运算处理的图。此外,图8的右图是明示了在周期4(Cyc4)中被激活的部分的图。

[0182] 图9是与图8相同的图。

[0183] 图9的右图是明示了在周期5(Cyc5)中被激活的部分的图。

[0184] (Cyc0~3):

[0185] 因为周期0~3(Cyc0~3)的处理与“1.2.1:MulShift处理”中的周期0~3的处理相同,所以省略详细的说明。

[0186] (Cyc4):

[0187] 在周期4(Cyc4)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算

命令(AddL命令),第3插槽33的加法运算单元331变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期3中经由数据通路Di31将储存在规定的寄存器中的位移运算处理结果的下位16比特数据输出到第3插槽33。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12经由数据通路Di32将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的下位16比特数据输出到第3插槽33。

[0188] 第3插槽33的加法运算单元331对经由数据通路Di31、Di32输入的2个16比特数据执行加法运算处理。然后,加法运算单元331将加法运算结果的16比特数据作为16比特数据Do3输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。再有,通过加法运算单元331中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0189] (Cyc5):

[0190] 在周期5(Cyc5)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算命令(AddH命令),第2插槽32的加法运算单元321变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12经由数据通路Di22将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的上位16比特数据输出到第2插槽32。

[0191] 此外,状态寄存器34经由第1切换部324将在周期3中储存的位移运算处理结果(由位移运算单元334得到的位移运算处理结果的上位16比特数据)输出到加法运算单元321。此时,第1切换部324被控制为根据控制信号Ct12向加法运算单元321输出状态寄存器34的输出。

[0192] 第2插槽32的加法运算单元321对经由数据通路Di22输入的16比特数据、从第1切换部324输出的16比特数据(由位移运算单元334得到的位移运算处理结果的上位16比特数据)、以及通过加法运算单元331中的加法运算处理取得的进位比特(未对进位比特的输入通路进行图示)执行加法运算处理。然后,加法运算单元321将加法运算结果的16比特数据作为16比特数据Do2经由第2切换部325输出到寄存器堆部2。再有,此时,第2切换部325被控制为根据控制信号Ct12向寄存器堆部2输出加法运算单元321的输出。

[0193] 寄存器堆部2将所接收的该16比特数据Do2储存在规定的寄存器中。再有,通过加法运算单元321中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0194] 此外,在周期5中,命令控制部1通过控制信号Ct12对命令执行部3发布乘法运算命令,第3插槽33的乘法运算单元333变为能够执行的状态。然后,执行与周期2的第3插槽33的乘法运算单元333的处理相同的处理。也就是说,与上述同样地执行对另外的数据的乘法运算处理(由第3插槽的乘法运算单元333进行的乘法运算处理)。再有,该乘法运算处理的处理对象在周期3、4中由第1插槽31的加载/存储单元311从数据存储器M2读出并储存在寄存器堆部2的规定的寄存器中。

[0195] 根据以上,在VLIW处理器1000中,能够对2个16比特数据执行乘法运算处理,能够对作为该乘法运算结果的32比特数据进行位移运算,能够执行将该位移运算结果与其它32比特数据进行加法运算的处理(MulShiftAdd处理)。

[0196] (1.2.4:ShiftAdd处理)

[0197] 接着,使用图10~12来说明在VLIW处理器1000中对32比特数据执行位移运算处

理、对作为该位移运算处理结果的32比特数据和其它32比特数据进行加法运算的情况(将在该情况下的处理称为“ShiftAdd处理”)。作为该ShiftAdd处理的一个例子,有对准了固定小数点位置后的累积加法运算处理。这样的处理多用在图像处理、图像识别处理等中。

[0198] 图10的左图是按每个周期示出在VLIW处理器1000执行ShiftAdd处理的情况下分配给各插槽(第1插槽31、第2插槽32以及第3插槽33)的运算处理的图。此外,图10的右图是明示了在周期2(Cyc2)中被激活的部分的图。

[0199] 图11、图12是与图10相同的图。

[0200] 图11的右图是明示了在周期3(Cyc3)中被激活的部分的图。

[0201] 图12的右图是明示了在周期4(Cyc4)中被激活的部分的图。

[0202] (Cyc0~1):

[0203] 因为周期0~1(Cyc0~1)的处理与“1.2.1:MulShift处理”中的周期0~1的处理相同,所以省略详细的说明。

[0204] (Cyc2):

[0205] 在周期2(Cyc2)中,命令控制部1通过控制信号Ct12对命令执行部3发布位移运算命令(Shift32命令),第3插槽33的位移运算单元334变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期0、1中将储存在规定的寄存器中的位移运算处理对象的32比特数据的下位16比特数据经由数据通路Di31输出到第3插槽33并且将该数据的上位16比特数据经由数据通路Di32输出到第3插槽33。

[0206] 第3插槽33的位移运算单元334对经由数据通路Di31输入的16比特数据和经由数据通路Di32及第3切换部335输入的16比特数据的2个16比特数据执行位移运算处理。然后,位移运算单元334将位移运算处理结果的32比特数据的上位16比特数据作为16比特数据Ds3经由第5切换部337输出到状态寄存器34。

[0207] 状态寄存器34储存所接收的16比特数据Ds3。

[0208] 此外,位移运算单元334将位移运算处理结果的32比特数据的下位16比特数据作为16比特数据Do3经由第4切换部336输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。

[0209] (Cyc3):

[0210] 在周期3(Cyc3)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算命令(AddL命令),第3插槽33的加法运算单元331变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12在周期2中经由数据通路Di31将储存在规定的寄存器中的位移运算处理结果的下位16比特数据输出到第3插槽33。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12经由数据通路Di32将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的下位16比特数据输出到第3插槽33。

[0211] 第3插槽33的加法运算单元331对经由数据通路Di31、Di32输入的2个16比特数据执行加法运算处理。然后,加法运算单元331将加法运算结果的16比特数据作为16比特数据Do3输出到寄存器堆部2。寄存器堆部2将所接收的该16比特数据Do3储存在规定的寄存器中。再有,通过加法运算单元331中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0212] (Cyc4):

[0213] 在周期4(Cyc4)中,命令控制部1通过控制信号Ct12对命令执行部3发布加法运算命令(AddH命令),第2插槽32的加法运算单元321变为能够执行的状态。此外,寄存器堆部2基于来自命令控制部1的控制信号Ct12经由数据通路Di22将进行加法运算的32比特数据(例如,累积加法运算值(32比特数据))的上位16比特数据输出到第2插槽32。

[0214] 此外,状态寄存器34经由第1切换部324将在周期2中储存的位移运算处理结果(由位移运算单元334得到的位移运算处理结果的上位16比特数据)输出到加法运算单元321。此时,第1切换部324被控制为根据控制信号Ct12向加法运算单元321输出状态寄存器34的输出。

[0215] 第2插槽32的加法运算单元321对经由数据通路Di22输入的16比特数据和从第1切换部324输出的16比特数据(由位移运算单元334得到的位移运算处理结果的上位16比特数据)执行加法运算处理。然后,加法运算单元321将加法运算结果的16比特数据作为16比特数据Do2经由第2切换部325输出到寄存器堆部2。再有,此时,第2切换部325被控制为根据控制信号Ct12向寄存器堆部2输出加法运算单元321的输出。

[0216] 寄存器堆部2将所接收的该16比特数据Do2储存在规定的寄存器中。再有,通过加法运算单元321中的加法运算处理取得的进位比特也被输出到寄存器堆部2并储存在规定的寄存器(例如,进位寄存器)中。再有,对进位比特的数据通路省略了图示。

[0217] 此外,在周期4中,命令控制部1通过控制信号Ct12对命令执行部3发布位移运算命令(Shift32命令),第3插槽33的位移运算单元334变为能够执行的状态。然后,执行与周期2的第3插槽33的位移运算单元334的处理相同的处理。也就是说,与上述同样地执行对另外的数据的位移运算处理。再有,该位移运算处理的处理对象在周期2、3中由第1插槽31的加载/存储单元311从数据存储单元M2读出并储存在寄存器堆部2的规定的寄存器中。

[0218] 根据以上,在VLIW处理器1000中,能够对32比特数据执行位移运算处理,能够执行对作为该位移运算处理结果的32比特数据和其它32比特数据进行加法运算的处理(ShiftAdd处理)。

[0219] 像以上那样,在VLIW处理器1000中,即使在执行处理对象为32比特数据的32比特运算的情况下,该32比特数据的上位16比特数据也能够被状态寄存器34保持并且在下一个周期中被规定的运算单元利用。

[0220] 此外,如上所述,在VLIW处理器1000中,状态寄存器34能够对从第3插槽33输出的32比特数据的上位16比特数据进行储存保持,将储存的数据输出到第2插槽32或第3插槽33。因此,在VLIW处理器1000中,不需要从第3插槽33向寄存器堆部2输出32比特数据,此外,也不需要从寄存器堆部2向第3插槽33输出32比特数据。也就是说,在VLIW处理器1000中,保证了在寄存器堆部2与命令执行部3之间仅输入/输出16比特数据,因此,与仅执行16比特运算的情况相比不需要增加寄存器堆部2与命令执行部3之间的输入/输出端口。

[0221] 即,在VLIW处理器1000中,如上所述,能够在不增加寄存器堆部2的输入/输出端口数的情况下一边将N比特运算(N:自然数)分配给各插槽一边支持 $2 \times N$ 比特的运算。

[0222] 像这样,在VLIW处理器1000中,能够抑制电路规模的增大,并且,即使在包括像多用于图像处理、图像识别处理等中的命令流那样进行比特扩展的运算(在上述中为32比特运算)的情况下,也能够高效率地执行处理。

[0223] [其它实施方式]

[0224] 虽然在上述实施方式中说明了将从第3插槽33输出的32比特数据的上位16比特数据输出到状态寄存器34并且将下位16比特数据输出到寄存器堆部2的情况,但是不限于于此。在VLIW处理器1000中,例如,也可以将从第3插槽33输出的32比特数据的下位16比特数据输出到状态寄存器34并且将上位16比特数据输出到寄存器堆部2。

[0225] 此外,虽然在上述实施方式中说明了在32比特加法运算处理中使用第2插槽32的加法运算单元321和第3插槽33的加法运算单元331的情况,但是不限于于此。例如,也可以使第2插槽32的加法运算单元321和第3插槽33的加法运算单元331分别作为执行16比特数据的加法运算处理的运算处理单元来发挥作用。由此,在VLIW处理器1000中,也能够同一周期中将16比特数据的加法运算处理分配给第2插槽32和第3插槽33。

[0226] 虽然在上述实施方式中说明了在VLIW处理器1000中命令执行部3的插槽数为“3”的情况,但是不限于于此,命令执行部3也可以具有其它数量的插槽。

[0227] 此外,也可以对上述实施方式的一部分或全部进行组合。

[0228] 此外,上述实施方式中的处理方法的执行顺序不一定被限制于上述实施方式的记载,能够在不脱离发明的主旨的范围内调换执行顺序。

[0229] 再有,本发明的具体的结构不限于前述的实施方式,在不脱离发明的主旨的范围内能够进行各种变更和修正。

[0230] 附图标记の説明

[0231] 1000 VLIW处理器

[0232] 1 命令控制部

[0233] 2 寄存器堆部

[0234] 3 命令执行部

[0235] 31 第1插槽

[0236] 32 第2插槽

[0237] 33 第3插槽

[0238] 311 加载/存储单元

[0239] 321、331 加法运算单元

[0240] 322、332 逻辑运算单元

[0241] 323 状态读出单元

[0242] 333 乘法运算单元

[0243] 334 位移运算单元

[0244] 324 第1切换部

[0245] 325 第2切换部

[0246] 335 第3切换部

[0247] 336 第4切换部

[0248] 34 状态寄存器。

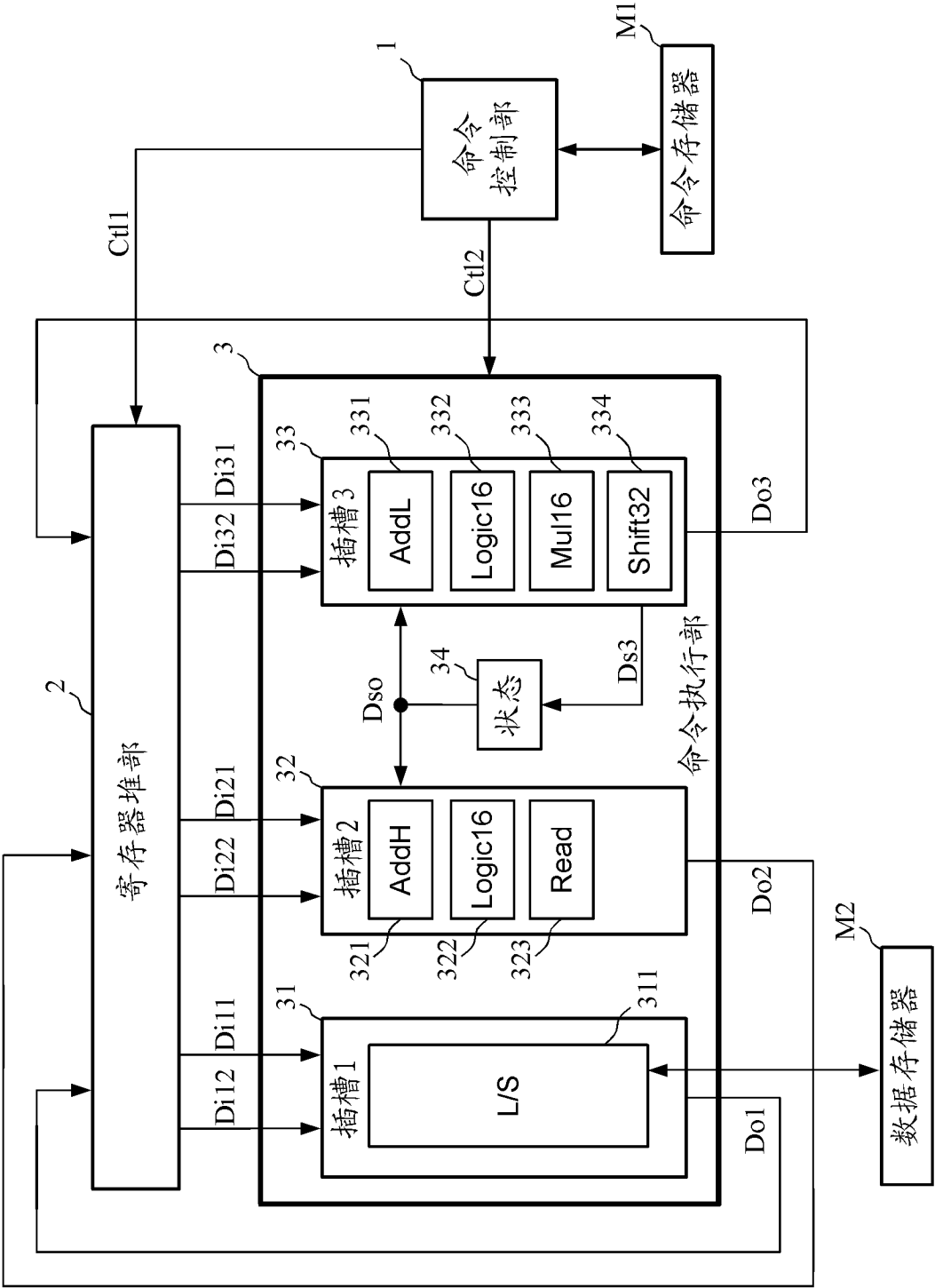


图 1

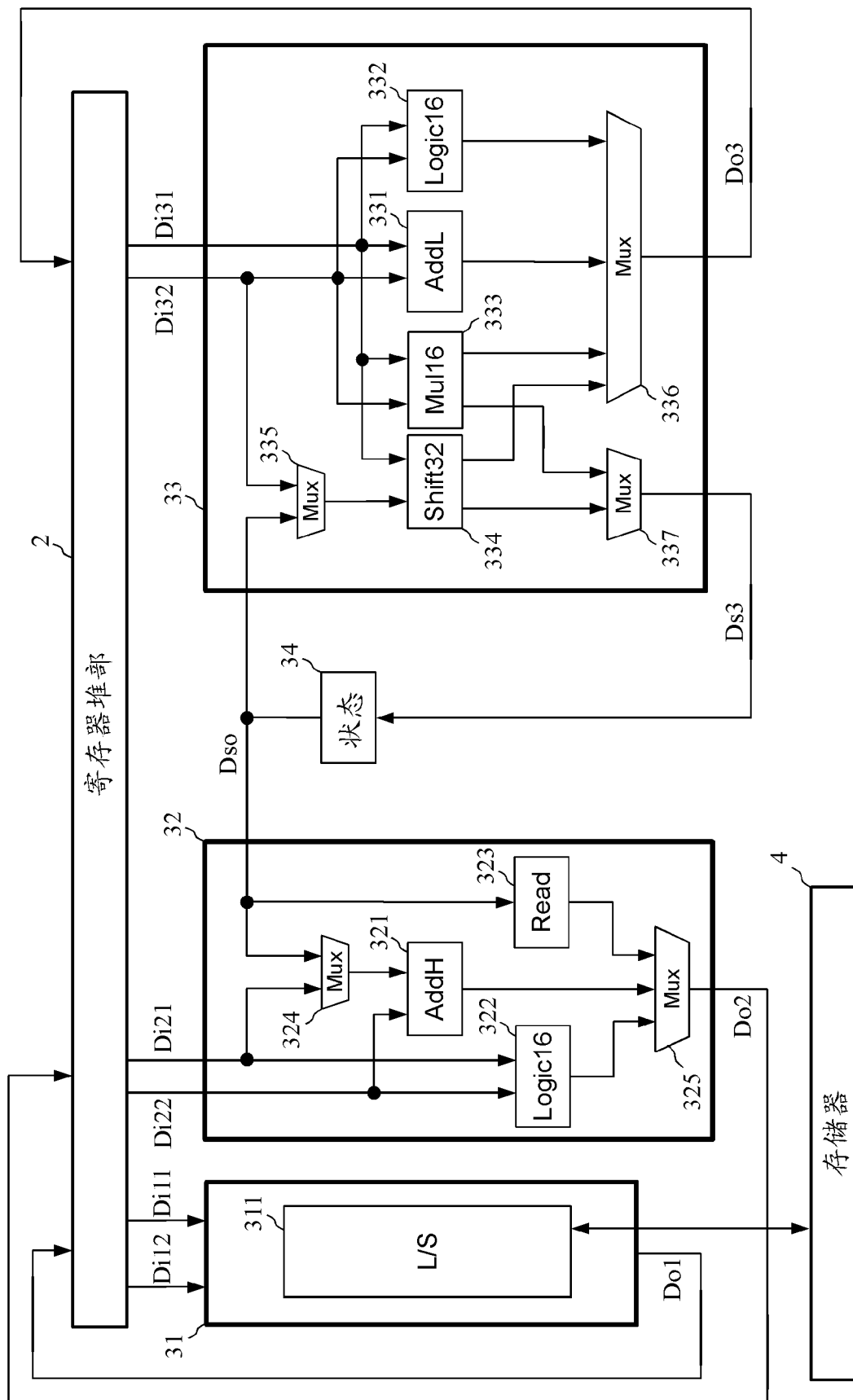
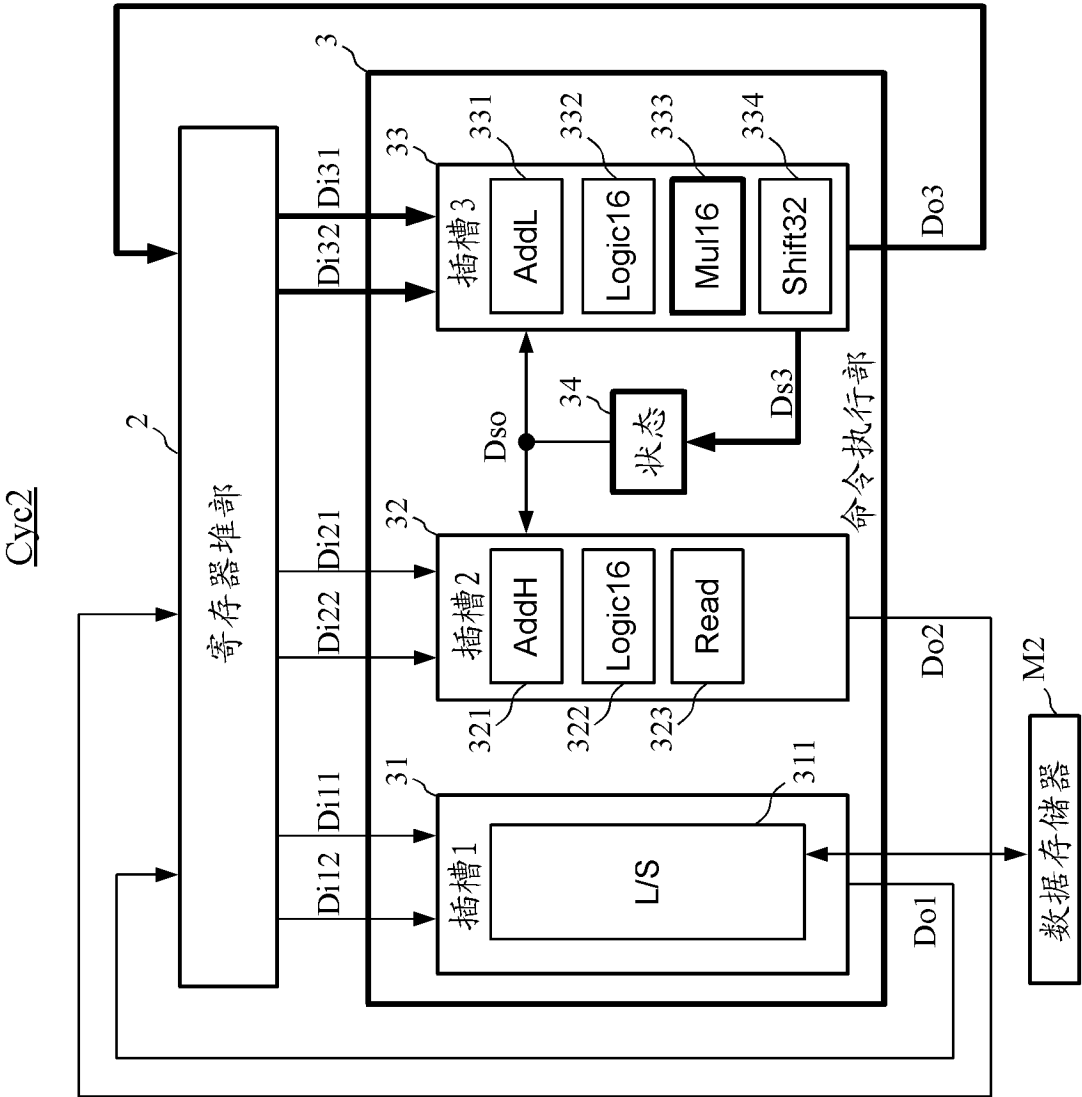


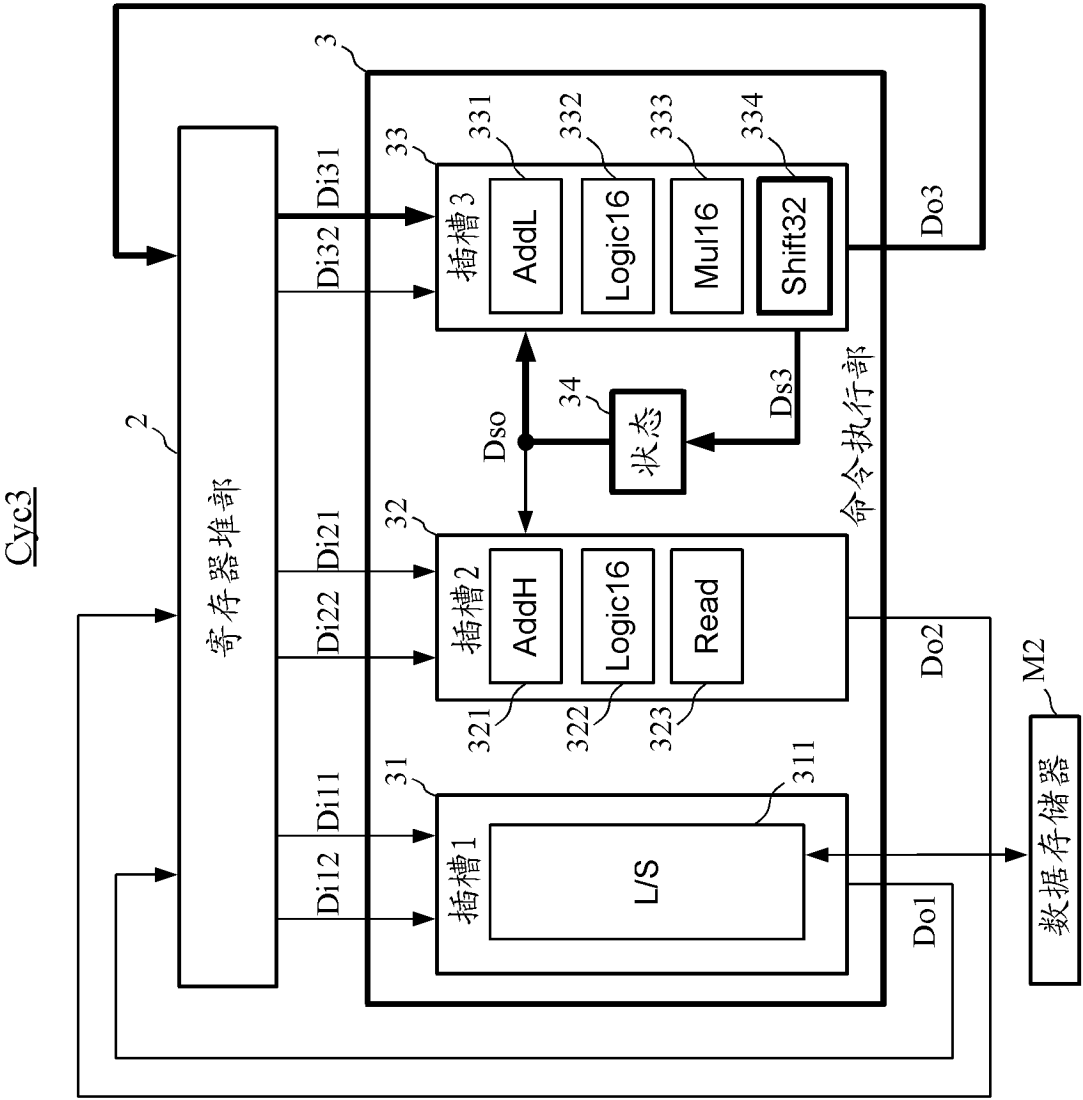
图 2



MulShift处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 | L/S |      | Mul16   |
| Cyc3 | L/S |      | Shift32 |
| Cyc4 | L/S | Read | Mul16   |
| Cyc5 | L/S |      | Shift32 |
| Cyc6 |     | Read | Mul16   |
| Cyc7 |     |      | Shift32 |
| Cyc8 |     | Read |         |

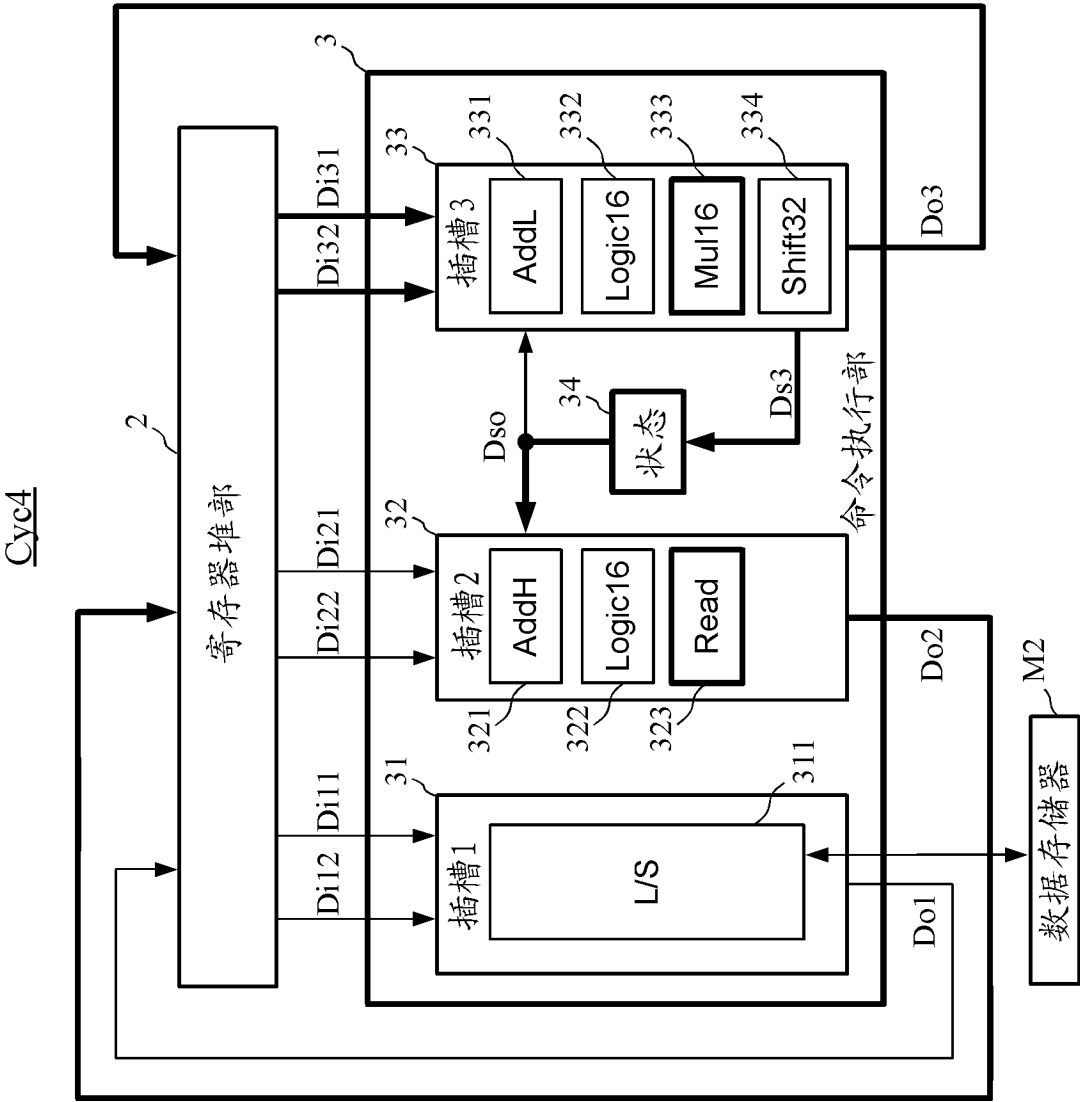
图 3



MulShift处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 | L/S |      | Mul16   |
| Cyc3 | L/S |      | Shift32 |
| Cyc4 | L/S | Read | Mul16   |
| Cyc5 | L/S |      | Shift32 |
| Cyc6 |     | Read | Mul16   |
| Cyc7 |     |      | Shift32 |
| Cyc8 |     | Read |         |

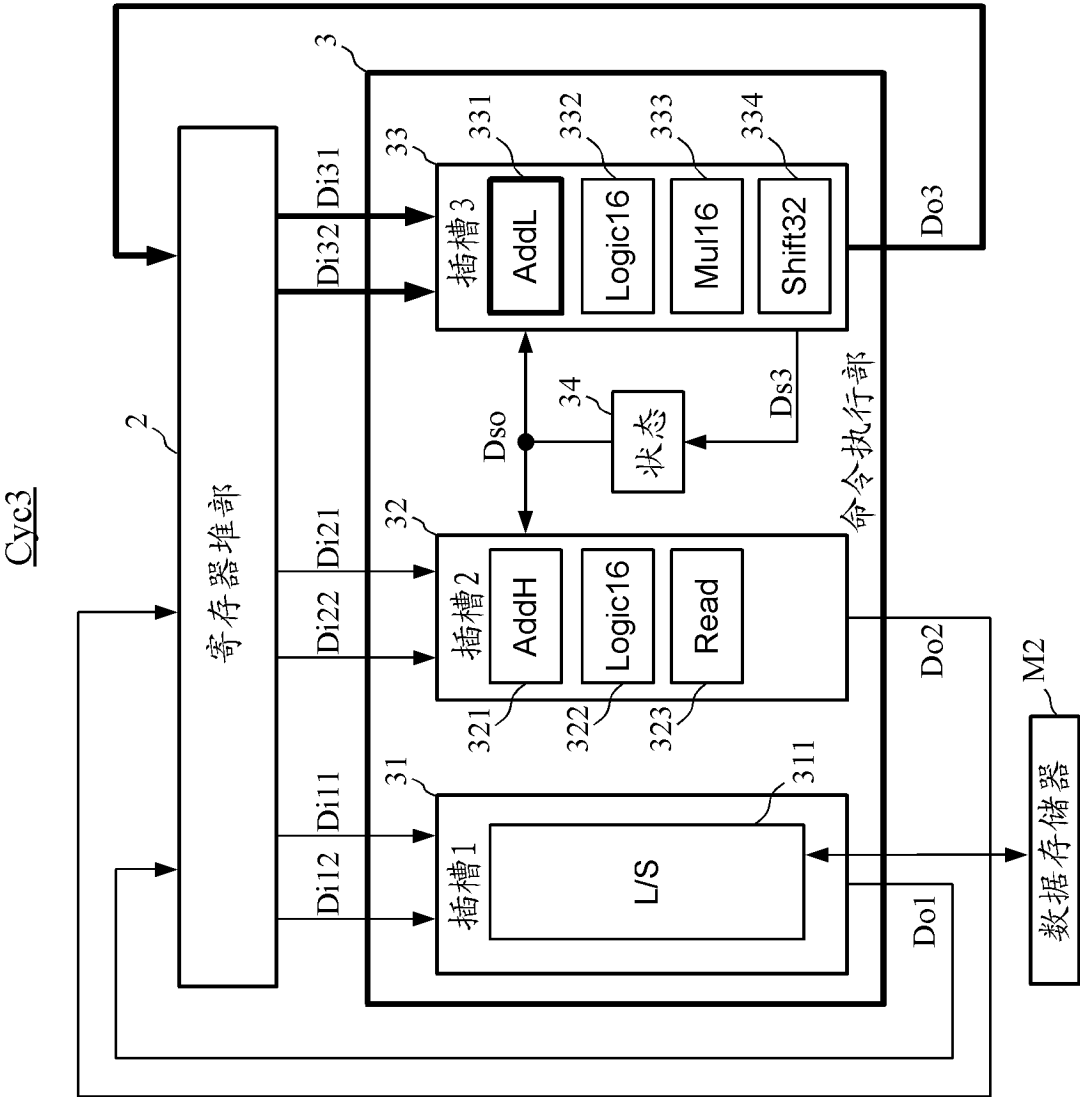
图 4



MulShift处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 | L/S |      | Mul16   |
| Cyc3 | L/S |      | Shift32 |
| Cyc4 | L/S | Read | Mul16   |
| Cyc5 | L/S |      | Shift32 |
| Cyc6 |     | Read | Mul16   |
| Cyc7 |     |      | Shift32 |
| Cyc8 |     | Read |         |

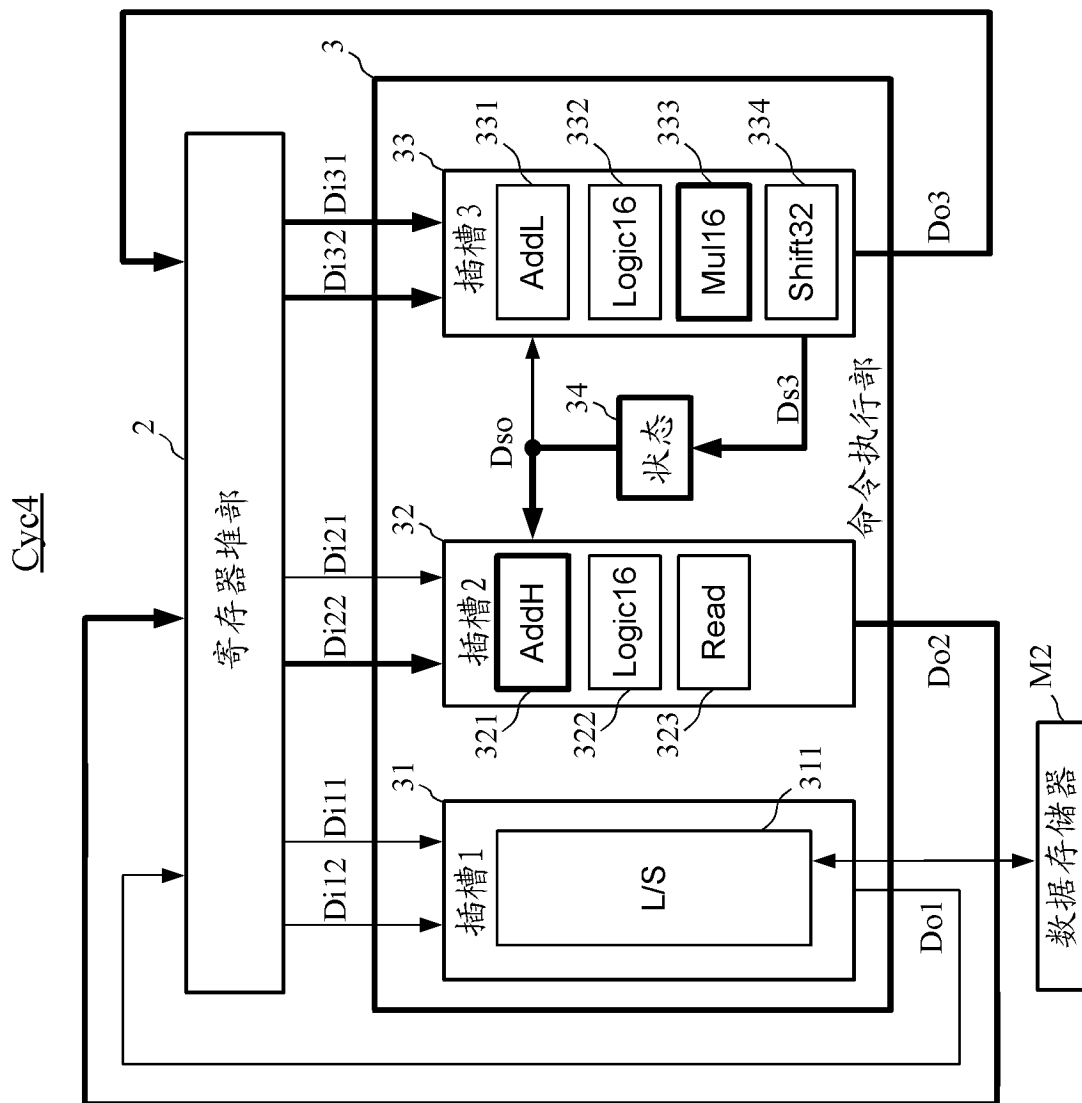
图 5



MulAdd处理

|      | 插槽1 | 插槽2  | 插槽3   |
|------|-----|------|-------|
| Cyc0 | L/S |      |       |
| Cyc1 | L/S |      |       |
| Cyc2 | L/S |      | Mul16 |
| Cyc3 | L/S |      | AddL  |
| Cyc4 | L/S | AddH | Mul16 |
| Cyc5 | L/S |      | AddL  |
| Cyc6 |     | AddH | Mul16 |
| Cyc7 |     |      | AddL  |
| Cyc8 |     | AddH |       |

图 6



|      | 插槽1 | 插槽2  | 插槽3   |
|------|-----|------|-------|
| Cyc0 | L/S |      |       |
| Cyc1 | L/S |      |       |
| Cyc2 | L/S |      | Mul16 |
| Cyc3 | L/S |      | AddL  |
| Cyc4 | L/S | AddH | Mul16 |
| Cyc5 | L/S |      | AddL  |
| Cyc6 |     | AddH | Mul16 |
| Cyc7 |     |      | AddL  |
| Cyc8 |     | AddH |       |

图 7

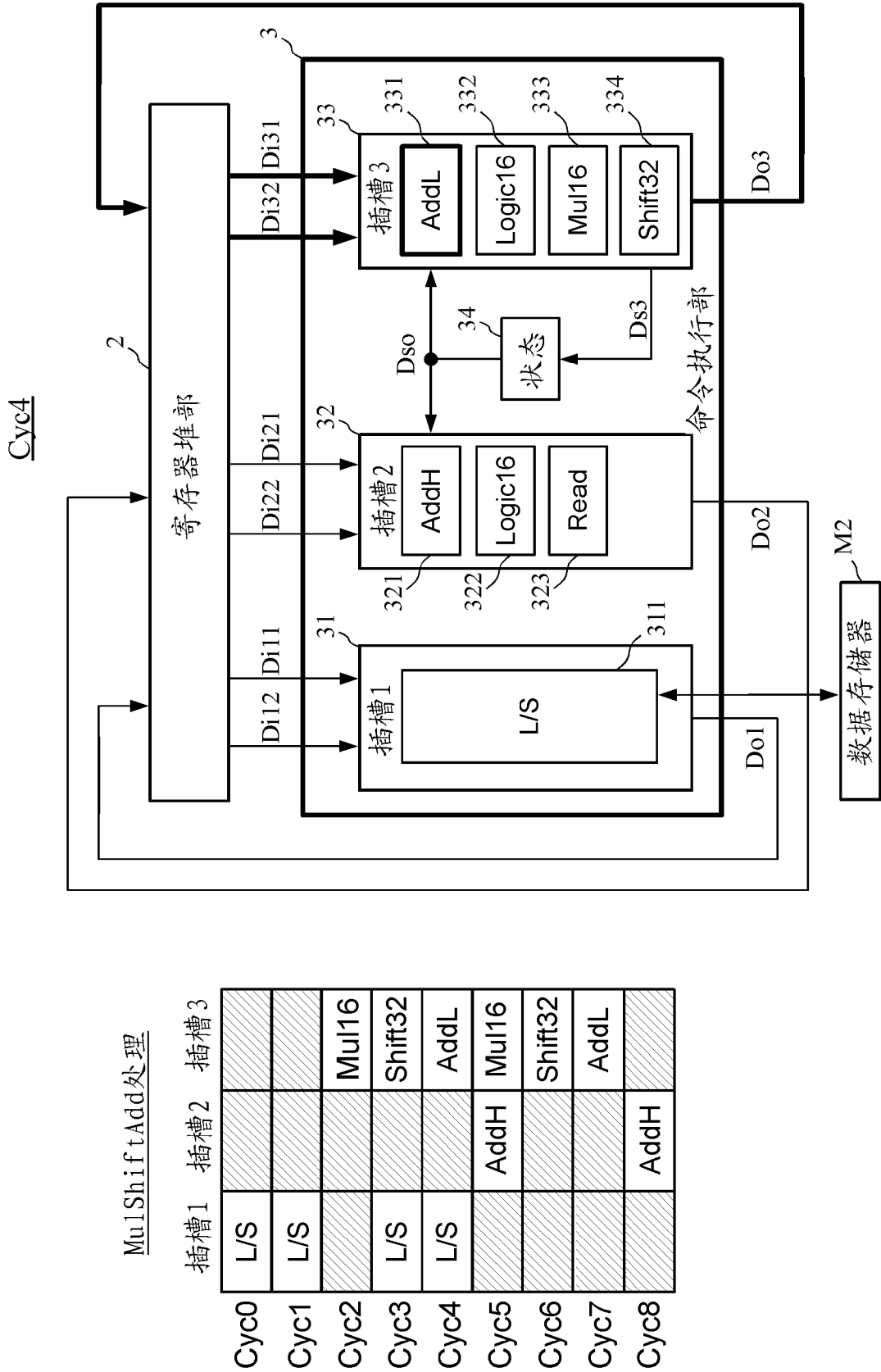
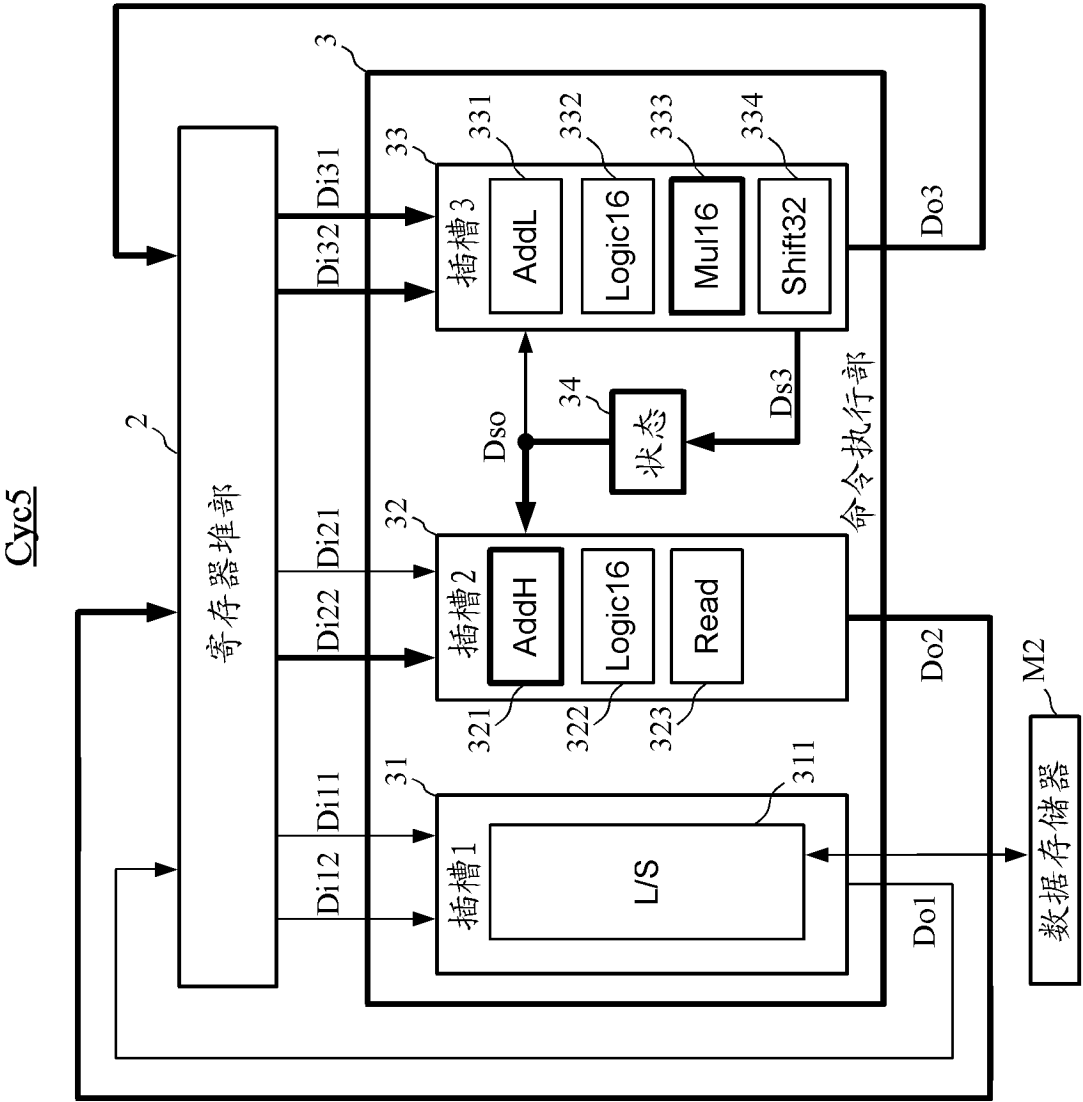


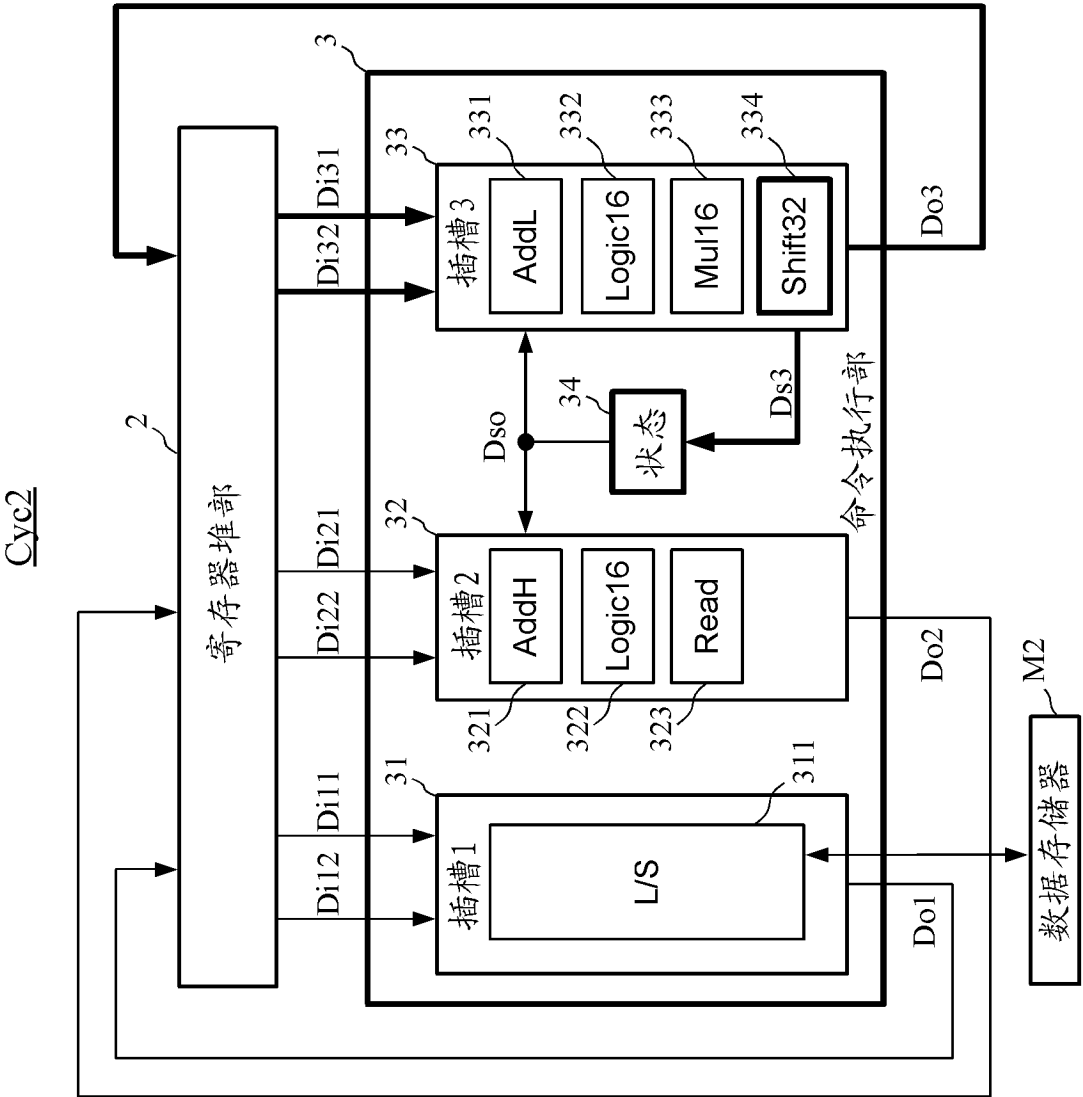
图 8



MulShiftAdd处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 |     |      | Mul16   |
| Cyc3 | L/S |      | Shift32 |
| Cyc4 | L/S |      | AddL    |
| Cyc5 |     | AddH | Mul16   |
| Cyc6 |     |      | Shift32 |
| Cyc7 |     |      | AddL    |
| Cyc8 |     | AddH |         |

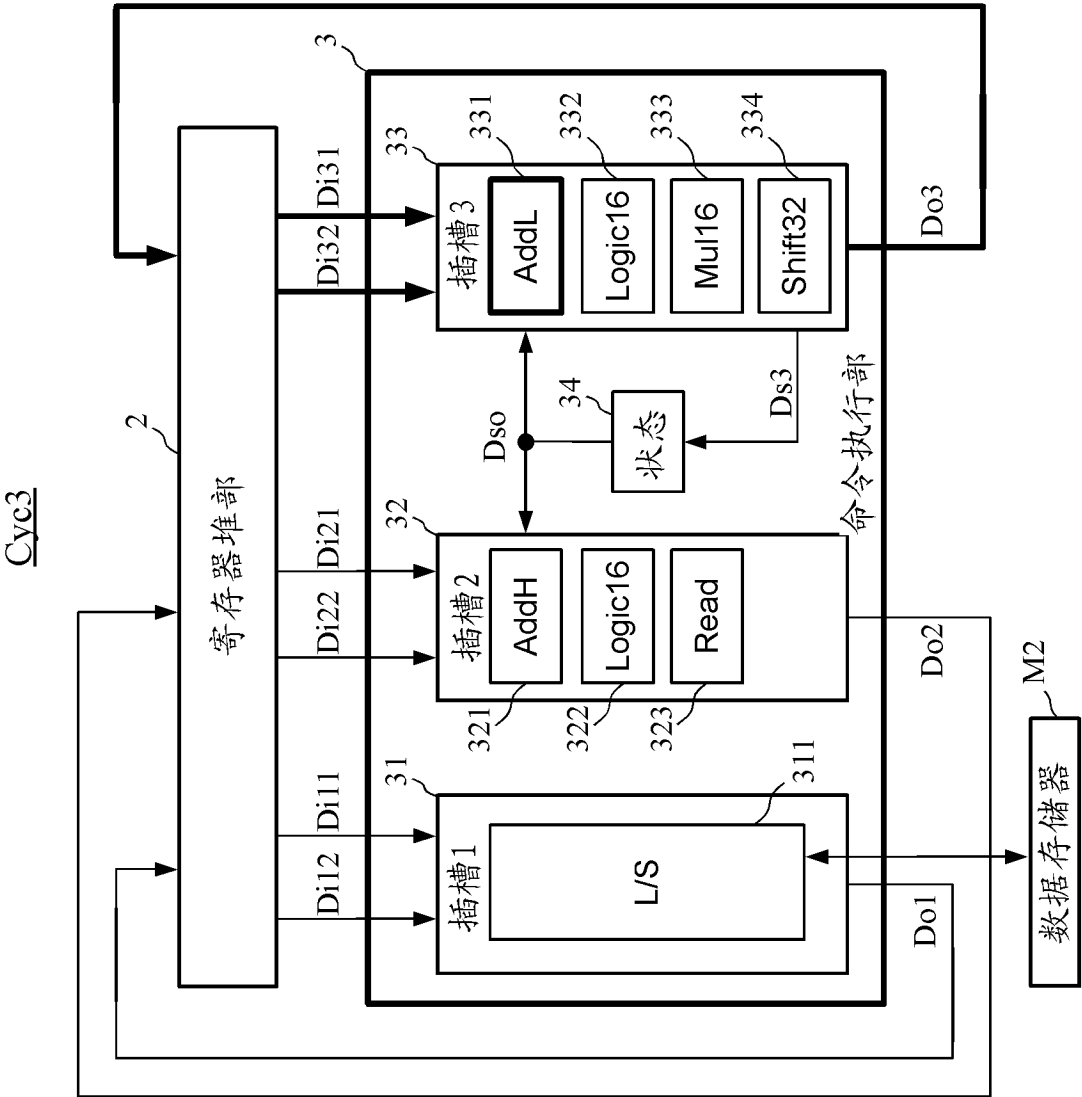
图 9



ShiftAdd处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 | L/S |      | Shift32 |
| Cyc3 | L/S |      | AddL    |
| Cyc4 | L/S | AddH | Shift32 |
| Cyc5 | L/S |      | AddL    |
| Cyc6 |     | AddH | Shift32 |

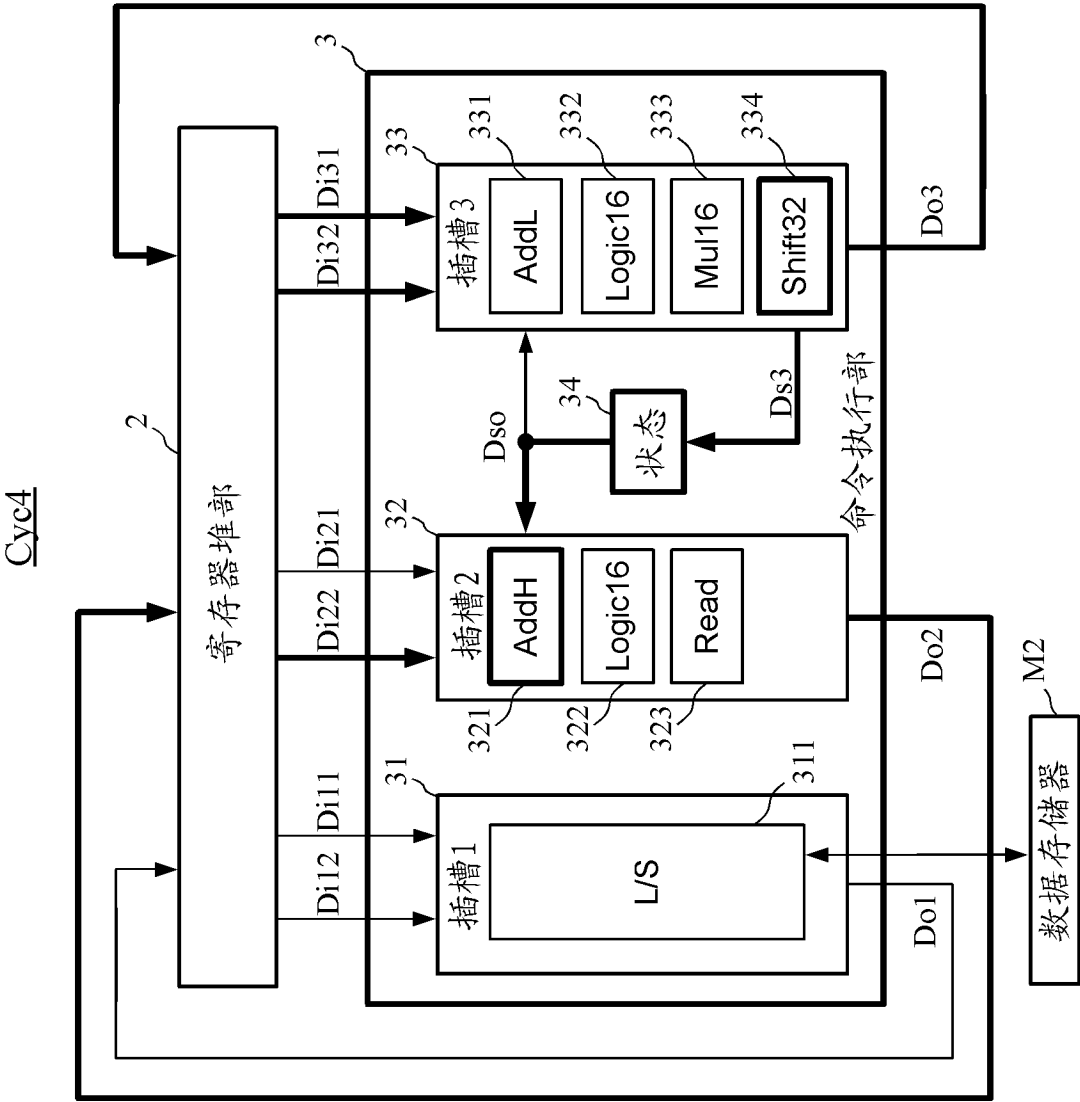
图 10



ShiftAdd处理

|      | 插槽1 | 插槽2  | 插槽3     |
|------|-----|------|---------|
| Cyc0 | L/S |      |         |
| Cyc1 | L/S |      |         |
| Cyc2 | L/S |      | Shift32 |
| Cyc3 | L/S |      | AddL    |
| Cyc4 | L/S | AddH | Shift32 |
| Cyc5 | L/S |      | AddL    |
| Cyc6 |     | AddH | Shift32 |

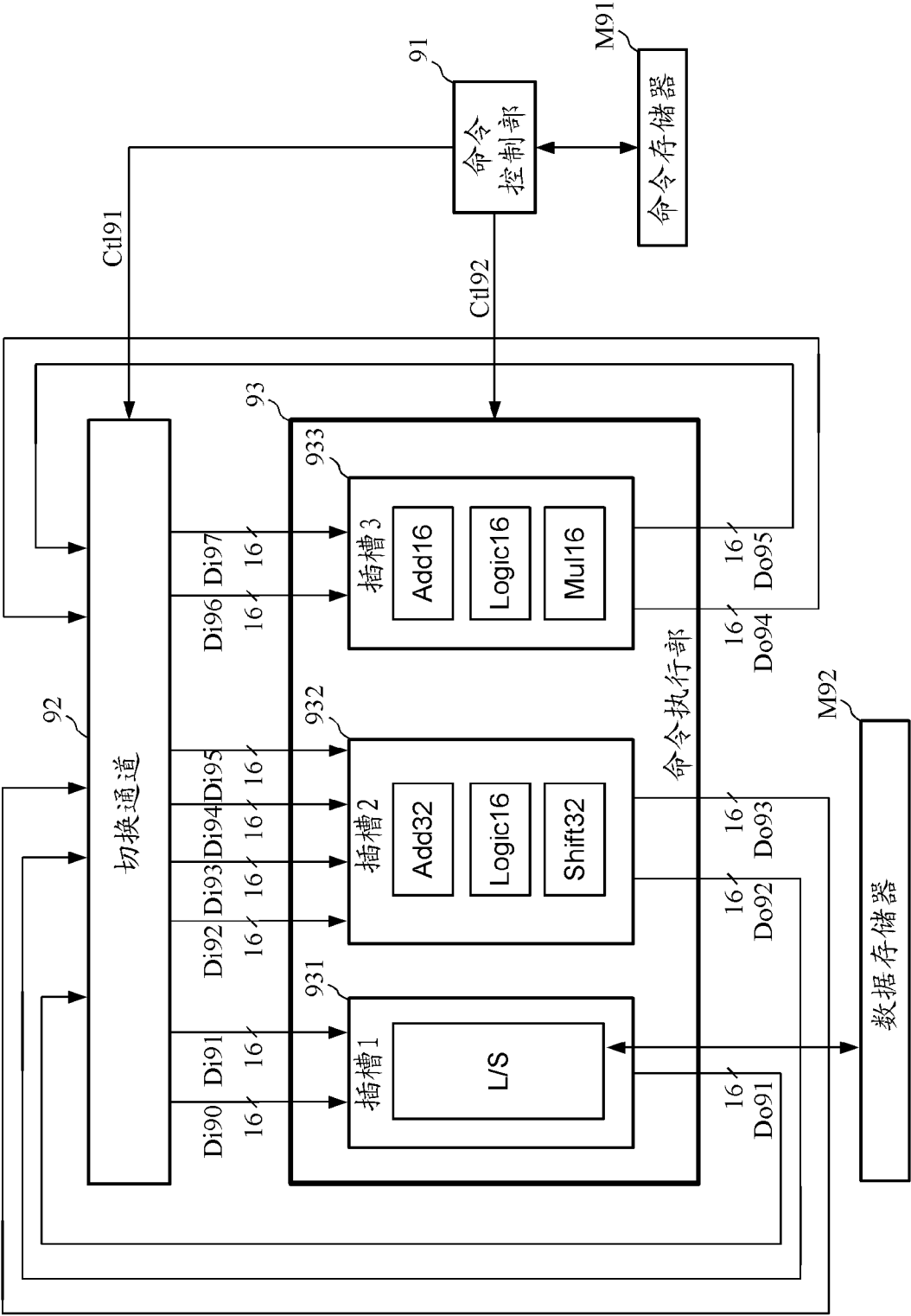
图 11



ShiftAdd处理

|      | 插槽1 | 插槽2     | 插槽3     |
|------|-----|---------|---------|
| Cyc0 | L/S |         |         |
| Cyc1 | L/S |         |         |
| Cyc2 | L/S | Shift32 |         |
| Cyc3 | L/S | AddL    |         |
| Cyc4 | L/S | AddH    | Shift32 |
| Cyc5 | L/S | AddL    |         |
| Cyc6 |     | AddH    | Shift32 |

图 12



900

图 13