[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 15/163 G06F 17/60
//G06F155: 00

[12]发明专利说明书

[21] ZL 专利号 95107288.9

[45]授权公告日 2003 年 1 月 1 日

[11]授权公告号 CN 1097784C

[22]申请日 1995.7.4 [21]申请号 95107288.9

[30]优先权

[32]1994.7.4 [33]JP [31]152508/94

[73]专利权人 任天堂株式会社

地址 日本京都府

[72]发明人 高桥豊文 田中登志雄 寺川英明

[56]参考文献

JP3 -292558A 1991.12.24 G06F13/18、G06F15/16

JP55 -82366A 1980. 6.21 G06F15/16

JP63 -132368A 1988. 6. 4 G06F15/16

JP63 -76048A 1988. 4. 6 G06F15/16

审查员 张 蕾

[74]专利代理机构 上海专利商标事务所

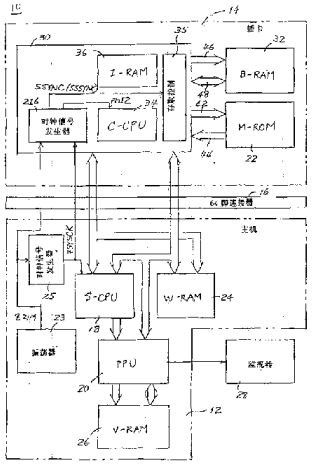
代理人 赵国华

权利要求书 5 页 说明书 25 页 附图 31 页

[54]发明名称 协处理器系统以及带辅助运算功能的外部存储装置

[57]摘要

游戏系统 10 包括游戏机主机 12 的 S - CPU18 和插卡 14 的 C - CPU34,两 CPU 由相同 CPU 内核构成,并具有相同存储器映像。插卡 14 的 M - ROM22 的存取时间比 C - CPU34 的周期时间短,C - CPU34 的周期时间比 S - CPU18 的周期时间短。允许 S - CPU18 对 M - ROM22 存取的时间由信号 SSYNC 设定成比 M - ROM22 的存取时间长但比 S - CPU18 的周期时间短。因而信号 SSYNC 允许 C - CPU34 在宽余时间内存取 M - ROM22,两 CPU 在原有存储容量的情况下实际上同时以两相存取。



ISSN1008-4274

1. 一种协处理器系统，包括：

根据程序在第一周期时间内进行运算处理的第一处理器(18)；

与第一处理器的结构相同，并根据程序在第二周期时间内进行运算处理的第二处理器(34)；

将作为第一周期时间基准的第一时钟信号提供给第一处理器的第一时钟信号供给装置(23，25)；以及

第二处理器可存取和高速读出，并存储第一处理器和第二处理器直接存取的程序的程序存储器(22)，

其特征在于，

所述第二周期时间比所述第一周期时间短，

所述协处理器系统包括：

将作为所述第二周期时间基准的第二时钟信号提供给所述第二处理器的第二时钟信号供给装置(23，216)，以及

存取控制装置(35，54，60，52)，在比所述程序存储器存取时间长但比所述第一周期时间短的第一期间允许所述第一处理器对所述程序存储器直接存取，而在所述第一周期时间中除所述第一期间以外的第二期间则允许所述第二处理器对所述程序存储器直接存取。

2. 如权利要求 1 所述的协处理器系统，其特征在于，所述存取控制装置包括：信号输出装置(216，60)，该装置在所述第一期间有所述第一处理器的存取请求(SMHIT)时，输出第一电平信号，而所述第一期间没有所述第一处理器的存取请求时，则输出第二电平信号；响应所述第一电平信号，将来自所述第一处理器的地址提供给所述程序存储器的地址输入(MA)的地址切换装置(52)。

3. 如权利要求 2 所述的协处理器系统，其特征在于，所述地址切换装置响应所述第二电平信号，将来自所述第二处理器的地址提供给所述程序存储器的所述地址输入(MA)。

4. 如权利要求 1 至 3 中任一项所述的协处理器系统，其特征在于，包括：

将从所述程序存储器读出的程序数据提供给所述第一处理器的数据总线(SD)的第一数据锁存器；以及

向所述第一数据锁存器提供与所述第一时钟信号相对应规定的第一锁存信号(SSYNC)的第一锁存信号供给装置(216)。

5. 如权利要求 4 所述的协处理器系统，其特征在于，所述第一锁存信号的后缘处于比所述第一时钟信号后缘超前的相位。

6. 一种带辅助运算功能的外部存储装置，可脱卸自如地装在信息处理装置上，该信息处理装置包括：根据程序在第一周期时间内进行运算处理的第一处理器(18)；将作为所述第一周期时间基准的第一时钟信号提供给所述第一处理器的第一时钟信号供给装置(23，25)，其特征在于，带辅助运算功能的外部存储装置包括：

与所述第一处理器的结构相同，并根据程序在比所述第一周期时间短的第二周期时间内进行运算处理的第二处理器(34)；

将作为所述第二周期时间基准的第二时钟信号提供给所述第二处理器的第二时钟信号供给装置(23，216)；

所述第二处理器可存取和高速读出，并存储第一处理器和所述第二处理器直接存取的程序的程序存储器(22)；以及

存取控制装置(35，54，60，52)，在比所述程序存储器存取时间长但比所述第一周期时间短的第一期间允许所述第一处理器对所述程序存储器直接存取，而在所述第一周期时间中除所述第一

期间以外的第二期间则允许所述第二处理器对所述程序存储器直接存取。

7. 如权利要求 6 所述的带辅助运算功能的外部存储装置，其特征在于，所述存取控制装置包括：信号输出装置(216，60)，在所述第一期间有所述第一处理器的存取请求时输出第一电平信号，而所述第一期间没有所述第一处理器的存取请求时则输出第二电平信号；响应所述第一电平信号将来自所述第一处理器的地址提供给所述程序存储器的地址输入(MA)的地址切换装置(52)。

8. 如权利要求 7 所述的带辅助运算功能的外部存储装置，其特征在于，所述地址切换装置响应所述第二电平信号，将来自所述第二处理器的地址提供给所述程序存储器的所述地址输入(MA)。

9. 如权利要求 6 至 8 中任一项所述的带辅助运算功能的外部存储装置，其特征在于，包括：

将从所述程序存储器读出的程序数据提供给所述第一处理器数据总线的第一数据锁存器(66)；以及

向所述第一数据锁存器提供与所述第一时钟信号相对应规定的第一锁存信号(SSYNC)的第一锁存信号供给装置(216)。

10. 如权利要求 9 所述的带辅助运算功能的外部存储装置，其特征在于，所述第一锁存信号的后缘处于比所述第一时钟信号后缘超前的相位。

11. 如权利要求 6 所述的带辅助运算功能的外部存储装置，其特征在于，

所述存取控制装置包括一起始地址寄存器，所述第二处理器开

始执行所述程序存储器所存储的程序时用于指定其起始地址，

所述程序存储器使所述第一处理器执行的第一程序与所述第二处理器执行的第二程序混合存储，

所述外部存储装置还包括：

中断处理装置，该中断处理装置从所述起始地址寄存器所指定的地址开始执行所述第二程序，在第二程序执行当中，所述第一处理器需要存取时，通过中断第二程序的执行，并且将中断的地址加载到起始地址寄存器，在第一处理器存取结束之后，再从起始地址寄存器所指定的地址开始，执行第二程序。

12. 如权利要求 6 所述的带辅助运算功能的外部存储装置，其特征在于，

所述第二处理器包含有临时保存所述第二程序执行得到的处理结果的内部寄存器，

所述外部存储装置还包括：

用于保留所述内部寄存器所保存的结果的保留存储器，

所述第二处理器执行所述第二程序当中，执行新的第二程序时，将所述内部寄存器的保存结果保留在所述保留存储器中，所述新的第二程序执行结束以后，将所述保留存储器所保留的结果送回所述内部寄存器，从而可以再度执行所述第二程序。

13. 一种使信息处理装置高速工作的方法，该信息处理装置包括一具有第一周期时间、对程序存储器进行存取并按照该程序动作的第一处理器，其特征在于，

设有与所述第一处理器具有程序互换性，并具有比所述第一周期时间短的第二周期时间的第二处理器；

所述程序存储器采用存取时间小于或等于所述第二周期时间的程序存储器；

所述第一处理器为存取所述程序存储器而确保时间长于或等于

第一周期时间；

使所述第二处理器可以在所述第一处理器存取所述程序存储器剩余的时间内存取所述程序存储器；

由此，使得所述第一处理器和所述第二处理器实际上可以同时存取所述程序存储器，使所述信息处理装置高速化。

协处理器系统以及带辅助运算功能的外部存储装置

技术领域

本发明涉及一种协处理器系统以及带辅助运算功能的外部存储装置，尤其涉及这样一种协处理器系统和带辅助运算功能的外部存储装置，在外部存储装置的存储器中存储有程序，该外部存储装置可灵活脱卸地装在诸如游戏机、图像处理装置之类信息处理装置中，外部存储装置中所设的处理器与信息处理装置中所设的处理器协同动作。

背景技术

例如，1994 年 3 月 22 日将专利申请公开的特开平 6-79093 号公报揭示了一例这种协处理器系统。该现有技术简单而言，如图 1 所示，信息处理装置主机的 CPU1 与称为“马里奥芯片”(マリオチップ)的另一 CPU2 存取 ROM3 的程序，CPU1 按照程序执行信息处理。该现有技术中，用以使 CPU1 动作的程序由 ROM3 传送至 RAM4 加以存储(下行加载)。然后，CPU1 通过直接存取传送至 RAM4 的程序进行处理，来间接处理 ROM3 的程序。另一方面，马里奥芯片 2 在不向 RAM4 传送期间(即 CPU1 不存取 ROM3 期间)，对 ROM3 进行直接存取，执行程序。

此外，图 2 所示的并行处理装置也可以作为一种协处理器系统，但图 2 所示系统的总线控制器 5 是分别切换使第 1 CPU6a 和第 2 CPU6b 与 ROM7 连接，藉此使所需的程序从 ROM7 下行加载至第 1 RAM8a 或第 2 RAM8b 的。而且，CPU6a 和 CPU6b 由时钟信号发生器 9 提供共同的时钟信号。CPU6a 或 CPU6b 又分别根据相同时钟

信号同时(或同步)地执行相应 RAM8a 或 RAM8b 所存储的程序。具体来说，图 2 系统的两个 CPU6a 和 CPU6b 不是在相同的机器工作周期期间直接存取 ROM7，执行 ROM7 程序，而是先将程序传送至对应的 RAM8a、8b 一下，再在 RAM8a、8b 上直接处理(存取)，因而两个 CPU6a、6b 不是在相同动作周期期间直接存取 ROM7。

图 1 现有技术中，马里奥芯片 2 动作时，ROM3 就被马里奥芯片 2 所占用，这样，要使 CPU1 与马里奥芯片 2 同时动作，就需要 RAM4 这种另外的存储器。因而，在程序编制方面有以下诸多制约。(1)必须根据 CPU1 和马里奥芯片 2 各自的总体结构，分别为它们准备程序。(2)ROM3 内装入 CPU1 用的程序时，需要如上所述对它下行加载，因而必须把该程序写入到一系列 ROM 地址。(3)RAM4 的容量不足时，无法加载 CPU1 所需的全部程序，必须将它分割，对每个分割程序块进行加载，在(3)情况下，必须预先将转移、子程序写入每一分割程序块，即便是相同的子程序也需设置于每一分割程序块，因而 ROM 容量浪费较多。

图 2 现有技术，除了上述图 1 现有技术的问题(3)以外，还需要 RAM8a、RAM8b，因而器件个数增多，系统整体成本高。而且，为 RAM8a、RAM8b 低成本的目的而采用存储容量小的 RAM 的话，就不得不将 ROM7 的程序数据每次分作 RAM 最大存储容量频繁进行传送，因而 CPU6a、6b 的程序处理频繁地被中断，数据传送的等待时间增大，使用户在该期间处于等待状态，妨碍程序的高速处理。

发明内容

为此，本发明的主要目的在于提供一种对程序无制约，可以在整体上高速化的协处理器系统。

本发明另一目的在于提供一种对程序无制约，可以从整体上使信息处理装置的动作高速化的带辅助运算功能的外部存储装置。

本发明再一目的在于提供这样一种协处理器系统和带辅助运算功能的外部存储装置，即总体结构共同的两个处理器可共用程序

存储器，两个处理器通过非同步且直接地执行程序存储器的程序，不必采用转送程序用的 RAM，便能够以低成本构成作高速处理。

本发明第一方面的协处理器系统，包括：

根据程序在第一周期时间内进行运算处理的第一处理器(18)；

与第一处理器的结构相同，并根据程序在第二周期时间内进行运算处理的第二处理器(34)；

将作为第一周期时间基准的第一时钟信号提供给第一处理器的第一时钟信号供给装置(23，25)；以及

第二处理器可存取和高速读出，并存储第一处理器和第二处理器直接存取的程序的程序存储器(22)，

其特征在于，

所述第二周期时间比所述第一周期时间短，

所述协处理器系统包括：

将作为所述第二周期时间基准的第二时钟信号提供给所述第二处理器的第二时钟信号供给装置(23，216)，以及

存取控制装置(35，54，60，52)，在比所述程序存储器存取时间长但比所述第一周期时间短的第一期间允许所述第一处理器对所述程序存储器直接存取，而在所述第一周期时间中除所述第一期间以外的第二期间则允许所述第二处理器对所述程序存储器直接存取。

本发明第二方面的带辅助运算功能的外部存储装置，可脱卸自如地装在信息处理装置上，该信息处理装置包括：根据程序在第一周期时间内进行运算处理的第一处理器(18)；将作为所述第一周期时间基准的第一时钟信号提供给所述第一处理器的第一时钟信号供给装置(23，25)，其特征在于，带辅助运算功能的外部存储装置包括：

与所述第一处理器的结构相同，并根据程序在比所述第一周期时间短的第二周期时间内进行运算处理的第二处理器(34)；

将作为所述第二周期时间基准的第二时钟信号提供给所述第二

处理器的第二时钟信号供给装置(23，216)；

所述第二处理器可存取和高速读出，并存储第一处理器和所述第二处理器直接存取的程序的程序存储器(22)；以及

存取控制装置(35，54，60，52)，在比所述程序存储器存取时间长但比所述第一周期时间短的第一期间允许所述第一处理器对所述程序存储器直接存取，而在所述第一周期时间中除所述第一期间以外的第二期间则允许所述第二处理器对所述程序存储器直接存取。

例如，信号 SSYNC 规定第一期间和第二期间。具体来说，信号 SSYNC 为高电平时为第一期间，而低电平时为第二期间。第一期间当中，第一处理器直接存取程序存储器，按照其程序数据执行运算处理。第二期间当中，第二处理器直接存取程序存储器，按照其程序数据执行运算处理。由于第一处理器和第二处理器总体结构相同，具有程序互换性，所以通过两处理器都直接存取程序存储器，可以执行运算处理。

按照本发明，就象是一个 CPU(处理器)在进行串行处理那样编制出的程序，只要稍加修改，便可以变为两个 CPU(处理器)进行并行处理用的程序。而且，不对程序下行加载，所以没必要将程序写入一系列的 ROM 地址。也没有必要将相同子程序存储到每一分割程序块，这样就不会造成 ROM 容量的浪费。所以在程序编制方面没有制约。

按照本发明，用于下行加载程序的 RAM 也不再需要，所以系统整体的成本变低。此外，在先行发售的信息处理装置不具有下行加载程序用的 RAM 的情况下，采用现有技术不能提升处理能力，但采用本发明带辅助运算功能的外部存储装置的话，就可以以比较低价格使系统整体的信息处理能力得到升级。

本发明上述目的以及其它目的和特征可以从参照附图进行的下述说明当中进一步清楚。

附图说明

图 1 是简略示意一例现有技术的框图。

图 2 是简略示意另一侧现有技术的框图。

图 3 是示意本发明一实施例的框图。

图 4 是示意图 3 实施例从 S-CPU 看到的存储器映像的说明图。

图 5 是示意图 3 实施例从 C-CPU 看到的存储器映像的说明图。

图 6 是示意图 3 实施例的协处理器的框图。

图 7 是详示图 3 实施例协处理器一部分的框图。

图 8 是详示图 3 实施例协处理器另一部分的框图。

图 9 是详示图 7 实施例 M-ROM 时序器的框图。

图 10 是示意按图 9 实施例对 M-ROM 存取的状态说明图。

图 11 是详示图 7 实施例 B-RAM 时序器(I-RAM 时序器)的框图。

图 12 是示意按图 11 实施例对 B-RAM 存取的状态说明图。

图 13 是示意按图 11 实施例对 I-RAM 存取的状态说明图。

图 14 是示意图 7 实施例 SCPUIF 的框图。

图 15 是详示图 7 实施例 SCPUIF 一部分的框图。

图 16 是详示图 7 实施例 SCPUIF 另一部分的框图。

图 17 是详示图 7 实施例时钟信号发生器的框图。

图 18 是示意一例图 17 实施例时钟信号发生器所产生的各个定时信号或时钟信号的说明图。

图 19 是示意另一例图 17 实施例时钟信号发生器所产生的各个定时信号或时钟信号的说明图。

图 20 是示意实施例中 S-CPU 起动 C-CPU 时的动作的流程图。

图 21 是示意实施例中 S-CPU 停止 C-CPU 时的动作的流程图。

图 22 是示意实施例中 S-CPU 对 C-CPU 进行 NMI 中断时的动作的流程图。

图 23 是示意实施例中 S-CPU 对 C-CPU 进行 IRQ 中断时的动作的流程图。

图 24 是示意实施例中 C-CPU 修改 S-CPU 的 NMI 向量时的动作

的流程图。

图 25 是示意实施例中 C-CPU 对 S-CPU 进行 IRQ 中断时的动作的流程图。

图 26 是示意实施例中加速模式动作的流程图。

图 27 是示意按照图 26 动作的系统作分布处理的状态说明图。

图 28 是示意实施例中混合处理模式动作的流程图。

图 29 是示意按照图 28 动作的系统作分布处理的状态说明图。

图 30 是示意实施例中替代处理模式动作的流程图。

图 31 是示意按照图 30 动作的系统作分布处理的状态说明图。

具体实施方式

以下揭示的实施例中，本发明用于游戏系统。但需要预先说明的是，本发明不只用于游戏系统，也可以用于图像处理系统和其他相同的信息处理系统。

图 3 是本发明一实施例的协处理器系统框图。图 3 所示的作为协处理器系统一例的游戏系统 10，包括主机 12，和作为例如由 64 脚连接器 16 脱卸自如装在该主机 12 上的带辅助运算功能的外部存储装置一例的插卡 14。主机 12 包括作为主机一侧处理器的 S-CPU18，和 PPU(图像处理单元)。S-CPU18 按照插卡 14 中含的 M-ROM22 中预先存储的程序动作。S-CPU18 上连接工作 RAM(W-RAM24)。主机 12 还包含振荡器 23，该振荡器 23 输出的约 21MHz 的基本时钟信号 Z21M 提供给时钟信号发生器 25。时钟信号发生器 25 包括分频器，由该分频器对基本时钟信号 Z21M 进行八分频，得到约 2.68MHz 的系统时钟信号 SYSCK，提供给 S-CPU18。PPU20 在 S-CPU18 的控制之下，与 V-RAM26 协同动作执行图像修理，向监视器 28 提供视频信号，从而在监视器 28 的屏幕上显示游戏图像。另外，主机 12 的 S-CPU18、PPU20 的一般动作其本身已经公知，而且与本发明无直接关系，因此这里省略详细说明。

插卡 14 除上述 M-ROM22 以外，还包括协处理器 30 和 B-RAM32。

协处理器 30 则包括：用来达到辅助运算功能的 CPU 内核 C-CPU34，存取控制电路 35，I-RAM36，以及时钟信号发生器 216。B-RAM32 是由例如锂电池等作为后备支持的 RAM，具有例如 4M 位的存储容量。I-RAM36 由例如 16K 位的 SRAM 构成。B-RAM32 和 I-RAM36 均起到 S-CPU18 和和 C-CPU34 的外部 RAM 的作用。另外，M-ROM22 由例如最大达 64M 位的掩膜 ROM 构成，该 ROM 中存储有游戏程序。M-ROM22 的存取时间有一例是字长 16 位时约 120 纳秒(字长 8 位时约 60 纳秒)。

S-CPU18 和 C-CPU34 采用的是与例如西方设计公司(WDC)制造的“65C816”这种相同的 CPU 内核。但本实施例中，C-CPU34 的工作频率是 10.73MHz，S-CPU18 的工作频率是 2.68MHz(或 3.58MHz)。换言之，S-CPU18 和 C-CPU34 采用相同内核，所以以相同的结构动作。具体来说，具有程序互换性，不过就工作速度而言，C-CPU34 的工作速度比起 S-CPU18 工作速度最高达 4 倍。8 位 CPU 的 S-CPU18 其周期时间在 2.68MHz 时约为 372 纳秒，8 位 CPU 的 C-CPU34 其周期时间在 10.37MHz 时约为 93 纳秒。另一方面，M-ROM22 的存取时间如上所述，16 位字长时约为 120 纳秒，因而 M-ROM22 的存取时间按 8 位换算比 C-CPU34 的周期时间短。因此，以下实施例中利用这类时间差，把足以存取 M-ROM22 并由此读出程序数据的例如 186 纳秒分配给 S-CPU18，也同样分配给 C-CPU34 186 纳秒，从而使得 S-CPU18 与 C-CPU34 实际上能够同时对 M-ROM22 存取。而且，分配给 C-CPU34 的 186 纳秒是 C-CPU34 的周期时间的 2 倍，所以 C-CPU34 在 186 秒期间动作两次。

另外，实施例的游戏系统 10 中，S-CPU18、W-RAM24 和协处理器 30(即 C-CPU34)用地址总线 38 连接，S-CPU18、PPU20、W-RAM24 和协处理器 30(即 C-CPU34)用数据总线 40 连接。协处理器 30(即 C-CPU34)与 M-ROM22 用地址总线 42 和数据总线 44 连接，协处理器 30(即 C-CPU34)与 B-RAM32 用地址总线 46 和数据总线 48 连接。地址总线 38 为 24 位，数据总线 40 为 8 位，地址总线 42 为 23 位，

数据总线 44 为 16 位，地址总线 46 为 19 位，数据总线 48 为 8 位。而且，S-CPU18 与 C-CPU34 是通过相同存取控制电路 35 存取 M-ROM22 或 B-RAM32 的。所以，对于 S-CPU18 和 C-CPU34 来说，如图 4 和图 5 所示，除了只有 S-CPU18 能够存取的 W-RAM24 的地址空间外，都分配相同的例如 24 位地址空间。如上所述，各 CPU 虽为 8 位，但利用存储排切换技术，可有 24 位地址空间。具体来说，图 4 示出的是从 S-CPU18 看到的存储器映像，而图 5 示出的是从 C-CPU34 看到的存储器映像，两个处理器的存储变换是相同的。但 I-RAM36 的地址变换，S-CPU18 与 C-CPU34 多少有些不同。本实施例中设法使 S-CPU18 在“**3000h—**37FFh”上可以存取 I-RAM36，而 C-CPU34 则在“**0000h—**07FFh”和“**3000h—**37FFh”上可以存取 I-RAM36。另外，“**”是指“00h—3Fh”存储排和“80h—BFh”存储排。

另外，S-CPU18 对各存储器 22、32 和 36 的存取与 C-CPU34 对各存储器 22、32 和 36 的存取都是由接收来自时钟信号发生器 216 的信号 SSYNC(SSSYNC)的存取控制电路 35 控制的。该存取控制电路 35 包括后面述及的 M-ROM 时序器 54、B-RAM 时序器 98 和 I-RAM 时序器，它们的动作后面叙述。

图 6—图 8 是协处理器 30 的框图。协处理器 30 包括与 C-CPU34 相连的地址译码器 50。地址译码器 50 对 S-CPU18 和 C-CPU34 两者的地址译码。所以，地址译码器 50 接收 S-CPU18 的地址数据 SA 和 C-CPU34 的地址数据 CA。地址译码器 50 的输出如后面所述提供作为种种寄存器的起动信号或时钟信号。另外，地址译码器 50 还从 C-CPU34 接收该 C-CPU34 从 M-ROM22 读出程序数据用的信号 CRWB(“B”是反相，“”的意思，以下同)。

M-ROMSMMC52 进行 M-ROM22(图 3)存储排切换的同时，切换 S-CPU18 和 C-CPU34 的存取。为此，M-ROMSMMC52 接收 S-CPU18 的地址数据 SA 和 C-CPU34 的地址数据 CA，以及 S-CPU18 的数据 SD 和 C-CPU34 的数据 CD，同时还接收 M-ROM 时序器 54 来的信号 TMSA。

图3存取控制电路35所含的M-ROM时序器54如图9详示的那样，包含有地址译码器56和58。地址译码器56对S-CPU18的地址数据SA译码，当地址数据SA命中图4中示出的M-ROM22的预定存取地址时，输出高电平信号SMHIT。地址译码器58对C-CPU34的地址数据CA译码，当地址数据CA命中图5中示出的M-ROM22的预定存取地址时，输出高电平信号CMHIT。信号SMHIT提供给“与”门60的输入端，对该“与”门60的另一输入端，提供为S-CPU18存取M-ROM22而确定的定时信号SSYNC(后面图18或图19中述及)。该“与”门60的输出作为信号TMSA送给上述M-ROMSMMC52。因而，在M-ROMSMMC52中，该信号TMSA为高电平时S-CPU18的地址SA便有效，低电压时C-CPU18的地址SA便有效。这样便可将有效的地址SA或CA提供给M-ROM22的地址输入MA(图7)。

地址译码器56的输出还提供给“与”门62的一输入端，对“与”门62的另一输入端提供的是S-CPU18读出M-ROM22时为低电平的信号SCPURDB。因而，“与”门62的输出信号SMRDOUT在S-CPU18应读出M-ROM22时便成为高电平。信号SMRDOUT可从M-ROM22读出，送至锁存/缓冲器64(图7)，用于将数据输出端MD所输出的程序数据输入S-CPU18的数据总线SD。图7的锁存器/缓冲器64包含与数据输出端MD连接的锁存器66，前述信号SSYNC提供作为该锁存器66的锁存信号。锁存器66的高8位和低8位分别提供给三态缓冲器68主70。“与”门72和74的输出分别提供作为三态缓冲器68和70的控制输入。对“与”门72和74的一输入端分别提供上述信号SMRDOUT，对“与”门72的另一输入端则提供S-CPU18地址SA的最低有效位SA0，对“与”门74的另一端输入则提供最低有效位SA0的反相信号。因而，最低有效位SA0为低电平时便从三态缓冲器70输出M-ROM22数据的低8位MD0—MD7，最低有效位SA0为高电平时便从三态缓冲器68输出M-ROM22数据的高8位MD8-MD15后，分别输入S-CPU18的数据总线SD。

图9中，上述信号CMHIT提供给“与非”门76的一输入端，

对该“与非”门 76 的另一输入端则提供上述“与”门 60 输出的信号 TMSA。“与非”门电路 176 的输出为信号 RDY，该信号 RDY 起着用以停止 C-CPU34 的信号的作用。具体来说，信号 RDY 为低电平时 C-CPU34 便处于待机状态。因而，C-CPU34 和 S-CPU18 都存取 M-ROM22 时，信号 RDY 变为低电平，作为信号 CRDY 提供给 C-CPU34。C-CPU34 相应地便处于待机状态。由此，S-CPU18 与 C-CPU34 同时存取 M-ROM22 或 B-RAM32 时，C-CPU34 处于待机状态，S-CPU18 的存取优先。而且，S-CPU18 的存取结束后，C-CPU34 的待机状态便解除，C-CPU34 对 M-ROM22 或 B-RAM32 进行存取。因此，S-CPU18 和 C-CPU34 对 M-ROM22 或 B-RAM32 按非同步的两相进行存取是可行的，S-CPU18 和 C-CPU34 可以同时执行一个存储器(即 M-ROM22)所存储的程序。

而且，“与”门 60 输出的信号 TMSA 的反相信号，地址译码器 58 输出的信号 CMHIT，和 C-CPU34 应读出 M-ROM22 时为高电平的信号 CRWB 分别提供给“与”门 78 的 3 个输入端。因而，S-CPU18 不存取 M-ROM22 时，该“与”门 78 就响应信号 CMHIT，在 C-CPU34 应从 M-ROM22 读出数据时输出高电平的信号 CMRDOUT。信号 CMRDOUT 送至锁存/缓冲器 80(图 7)，用以将 M-ROM22 输出的程序数据输入 C-CPU34 的数据总线 CD。图 7 的锁存/缓冲器 80 包含有与 M-ROM22 的数据输出端 MD 相连接的锁存器 82，信号 CSYNC(后面述及)提供作为该锁存器 82 的锁存信号。锁存器 82 的高 8 位和低 8 位分别提供给三态缓冲器 84 和 86。三态缓冲器 84 和 86 的控制输入分别由“与”门 88 和 90 的输出提供。对“与”门 88 和 90 的一输入端分别提供上述信号 CMRDOUT，对“与”门 88 的另一输入端提供 C-CPU34 地址 CA 的最低有效位 CA0，对“与”门 90 另一输入端提供最低有效位 CA0 的反相信号。因而，最低有效位 CA0 低电平时从三态缓冲器 86 输出 M-ROM22 数据的低 8 位 MD0—MD7，最低有效位 CA0 为高电平时从三态缓冲器 84 输出 M-ROM22 数据的高 8 位 MD8—MD15 后，分别输入 C-CPU34 的数据总线 CD。

前述信号 CSYNC 是表示 C-CPU34 应从 M-ROM22 读出数据，并将读出的数据锁存至锁存器 82 的定时的信号，可由“与”门 92 输出。具体来说，将信号 CRWB、信号 CMHIT、地址 CA 的最低有效位 CA0 的反相信号，和时钟信号 PH12 提供给“与”门 92 的 4 个输入端。时钟信号 PH12 如图 18 或图 19 所示，具有用于 C-CPU34 的频率，例如为 10.73MHz 的时钟信号。因而，信号 CSYNC 当 C-CPU34 存取 M-ROM22 时变为高电平输出，而当低电平时锁存器 82 就锁存由 M-ROM22 数据总线 MD 提供的 16 位程序数据。

由图 4 和图 5 的存储器映像可知，S-CPU18 和 C-CPU34 共享 M-ROM22 的整个区域，不论何时都可以非同步地存取 M-ROM22。这时对存取 M-ROM22 进行仲裁的是 M-ROM 时序器 54。若图 9 的 M-ROM 时序器 54 的“与”门 60 有信号 TMSA 输出，图 8 的 M-ROMSMMC52 就使来自 S-CPU18 的地址 SA 有效。因而，此时如果 C-CPU34 命中 M-ROM22 地址的话，“与非”门 76 就有信号 RDY 输出，所以如图 10(B)或(C)所示，在信号 TMSA 高电平期间，C-CPU34 便等着对 M-ROM22 进行存取。因此，C-CPU34 在信号 TMSA 变为低电平，从而信号 RDY 变为高电平时，就可以存取 M-ROM22。这样，S-CPU18 和 C-CPU34 就可以如图 10(C)所示，按照非同步的两相(两种状态)存取 M-ROM22。

另外，在如图 10(A)所示，只有 C-CPU34 存取 M-ROM22 时，C-CPU34 的工作频率根据时钟信号 PH12，是 10.73MHz。但在如图 10(B)或(C)所示 S-CPU18 和 C-CPU34 一起按两相存取 M-ROM22 时，S-CPU18 存取 M-ROM22 期间 C-CPU34 处于等待状态，所以 C-CPU34 的工作频率为 5.37MHz。可 S-CPU18 的工作频率按照系统时钟信号 SYSCK，通常设定为 2.68MHz。

图 8 所示的 B-RAMSMMC94 和 I-RAMSMMC96 都具有与先前说明的 M-ROMSMMC52 相同的功能。具体来说，B-RAMSMMC94 对 B-RAM32(图 3)的存储排进行切换的同时，对 S-CPU18 和 C-CPU34 也进行切换。为此，B-RAMSMMC94 不仅接 S-CPU18 的地址数据 SA 和 C-CPU34 的

地址数据 CA，以及 S-CPU18 的数据 SD 和 C-CPU34 的数据 CD，还接收来自 B-RAM 时序器 98 的信号 TBSA。

I-RAMSMMC96 对 I-RAM36(图 3)的存储排进行切换的同时，对 S-CPU18 和 C-CPU34 也进行切换。为此，I-RAMSMMC96 不仅接收 S-CPU18 的地址数据 SA 和 C-CPU34 的地址数据 CA，以及 S-CPU18 的数据 SD 和 C-CPU34 的数据 CD，还接收来自 I-RAM 时序器 10 的信号 TISA。

图 3 存取控制电路 35 所含的 B-RAM 时序器 98 与 I-RAM 时序器 100 具有相同的电路构成，如图 11 详示的那样，包含有地址译码器 102 和 104。地址译码器 102 对 S-CPU18 的地址数据 SA 译码，若地址数据 SA 命中 B-RAM32 的预定存取地址，就输出高电平信号 SBHIT。地址译码器 104 对 C-CPU34 的地址数据 CA 译码，若地址数据 CA 命中 B-RAM32 的预定存取地址，就输出高电平信号 CBHIT。信号 SBHIT 提供给“与”门 106 的一输入端，对该“与”门 106 的另一输入端则提供先前说明的定时信号 SSYNC，该信号用于确定 S-CPU18 可以存取 B-RAM32 的时间。该“与”门 106 的输出作为信号 TBSA 送至 B-RAMSMMC94。因而，该信号 TBSA 高电平时，就为 S-CPU18 存取 B-RAM32 的时间。具体来说，信号 TBSA 高电平时，B-RAMSMMC94 就将 S-CPU18 输出的地址 SA 送给 B-RAM32 的地址输入端 BA，而低电平时则将 C-CPU34 输出的地址 CA 送给地址输入端 BA。

地址译码器 102 的输出还提供给“与”门 108 的一输入端，对“与”门 108 的另一输入端则提供 S-CPU18 读出 B-RAM32 时为低电平的信号 SCPURDB。因而，“与”门 108 的输出信号 SBRDOUT，在 S-CPU18 应读出 B-RAM32 时为高电平，在向 B-RAM32 写入数据时为低电平。信号 SBRDOUT 提供给锁存/缓冲器 110(图 8)，用于将 B-RAM32 的数据端子 BD 输出的数据输入 S-CPU18 的数据总线 SD。图 8 的锁存/缓冲器 110 包含有与数据端子 BD 连接的锁存器 112，前述信号 SSYNC 提供作为该锁存/缓冲器 112 的锁存信号。

锁存器 112 接收 B-RAM32 的 8 位输出数据，响应信号 SSYNC 将它锁存。锁存器 112 的输出送给三态缓冲器 114。而且，B-RAM32 的 8 位输出数据照原样送给三态缓冲器 116。B-RAM 时序器 98 输出的信号 SBRDOUT 和信号 CBRDOUT(后面述及)分别提供作为三态缓冲器 114 和 116 的控制输入。因而，信号 SBRDOUT 高电平时便由三态缓冲器 114 输出 B-RAM32 的输出数据。同样，信号 CBRDOUT 高电平时便由三态缓冲器 116 输出 B-RAM32 的输出数据，再分别输入 S-CPU18 的数据总线 SD 和 C-CPU34 的数据总线 CD。

图 11 的上述信号 CMHIT 提供给“与非”门 118 的一输入端，对该“与非”门 118 的另一输入端则提供来自“与”门 106 的信号 TBSA。“与非”门 106 的输出为 RDY，该信号 RDY 起着用以停止 C-CPU34 的信号的作用。具体来说，信号 RDY 低电平时 C-CPU34 处于待机状态。因此，C-CPU34 和 S-CPU18 都存取 B-RAM32 时，信号 RDY 便变为低电平，作为信号 CRDY 提供给 C-CPU34。C-CPU34 相应地变为待机状态。

而且，“与”门 106 输出的信号 TBSA 的反相信号，地址译码器 104 输出的信号 CBHIT，和 C-CPU34 写入时为低电平、读出时为高电平的信号 CRWB 分别送给“与”门 120 的三个输入端。因而，当 S-CPU18 不存取 B-RAM32 时，该“与”门 120 便响应信号 CBHIT，输出高电平的信号 CBRDOUT。信号 CBRDOUT 如前所述提供给三态缓冲器 116 作为控制信号。

如先前说明过的那样，I-RAM 时序器 100 与 B-RAM 时存器 98 也相同，因而，这里省略了 I-RAM 时序器 100 的详细说明。不过，图 11 中利用括号图示了图 11 电路用作 I-RAM 时序器 100 时所接收或输出的信号。

图 3 存取控制电路 35 所包含的 I-RAM 时序器 100 输出的信号 SIRDOUT 提供给锁存/缓冲器 122(图 8)，用以将来自 I-RAM36 的数据输入 S-CPU18 的数据总线 SD。图 8 锁存/缓冲器 122 包含有接收 I-RAM30 读出数据 IRD 的锁存器 124，前述信号 SSYNC 提供作为该

锁存器 124 的锁存信号。锁存器 124 接收 I-RAM36 的 8 位输出数据，响应信号 SSYNC 对它进行锁存。锁存器 124 的输出提供给三态缓冲器 126。而且，I-RAM36 的 8 位输出数据照原样提供给三态缓冲器 128。I-RAM 时序器 100 输出的信号 SIRDOUT 和信号 CIRDOUT 分别提供作为三态缓冲器 126 和 128 的控制输入。因而，信号 SIRDOUT 高电平时便由三态缓冲器 126 输出 I-RAM36 的输出数据。同样，信号 CIRDOUT 高电平时便由三态缓冲器 128 输出 I-RAM36 的输出数据，再分别输入 S-CPU18 的数据总线 SD 和 C-CPU34 的数据总线 CD。

至 B-RAM32 和 I-RAM36 的写入数据分别由选择器 130 和 132 选择。具体来说，选择器 130 接收来自 S-CPU18 数据总线 SD 的 8 位数据和来自 C-CPU34 数据总线 CD 的 8 位数据。B-RAM 时序器 98 输出的信号 RDY 提供作为选择器 130 的选择信号。也就是说，信号 RDY 低电平时选择器 130 便选择 S-CPU18 来的写入数据，提供给 B-RAM32，信号 RDY 高电平时选择器 130 便选择 C-CPU34 来的写入数据，提供给 B-RAM32。同样，选择器 132 接收来自 S-CPU18 数据总线 SD 的 8 位数据和来自 C-CPU34 数据总线 CD 的 8 位数据。I-RAM 时序器 100 输出的信号 RDY 提供作为选择器 132 的选择信号。也就是说，信号 RDY 低电平时选择器 132 便选择 S-CPU18 来的写入数据，提供给 I-RAM36，信号 RDY 高电平时选择器 132 便选择 C-CPU34 来的写入数据，提供给 I-RAM36。

由图 4 和图 5 的存储器映像可知，S-CPU18 和 C-CPU34 共享 B-RAM32 的整个区域，不论何时都可以非同步地存取 B-RAM32。这时对 B-RAM32 存取进行仲裁的是 B-RAM 时序器 98。若图 11 的 B-RAM 时序器 98 的“与”门 106 有高电平信号 TBSA 输出，图 8 的 B-RAMSSC94 中就使 S-CPU18 来的地址 SA 有效。因此，这时如果 C-CPU34 命中 B-RAM32 地址的话，便从“与非”门 118 输出信号 RDY，所以如图 12(B)和(C)所示，在信号 TMSA 高电平期间，C-CPU34 等着对 B-RAM32 进行存取。因而，信号 RDY 低电平，并且信号 TMSA 也低电平时，

C-CPU34 可以存取 B-RAM32。这样，S-CPU18 和 C-CPU34 就可以完全按两相对 B-RAM32 进行存取。

另外，在如图 12(A)所示只有 C-CPU34 存取 B-RAM32 时，C-CPU34 的工作频率按照时钟信号 PH12 为 5.37MHz。但在如图 12(B)或(C)所示，S-CPU18 和 C-CPU34 一起以两相存取 B-RAM32 时，C-CPU34 的工作频率为 2.68MHz。此外，S-CPU18 的工作频率按照系统时钟信号 SYSCK，通常设定为 2.68MHz。

同样，由图 4 和图 5 的存储器映像可知，S-CPU18 和 C-CPU34 可以存取 I-RAM36 的整个区域，不论何时都可以非同步地存取 I-RAM36。此时对 I-RAM36 存取进行仲裁的是 I-RAM 时序器 100。若图 11 的 I-RAM 时序器 100 的“与”门电路 106 有高电平信号 TISA 输出，图 8 的 I-RAMSMMC96 中就使 S-CPU18 来的地址 SA 有效。因此，这时如果 C-CPU34 命中 I-RAM36 地址的话，就从“与非”门电路 118 输出信号 RDY，所以如图 13(B)所示，在信号 TISA 高电平期间 C-CPU34 等着对 I-RAM36 进行存取。因而，信号 RDY 低电平，并且信号 TISA 也低电平时，C-CPU34 就可以存取 I-RAM36。这样，S-CPU18 和 C-CPU34 就可以完全按两相对 I-RAM36 进行存取。

另外，在如图 13(A)所示只有 C-CPU34 存取 I-RAM36 时，C-CPU34 的工作频率按照时钟信号为 10.73MHz。但在如图 13(B)所示，S-CPU18 与 C-CPU34 一起以两相存取 I-RAM36 时，C-CPU34 的工作频率变为 5.37MHz。不过，S-CPU18 的工作频率按照系统时钟信号 SYSCK，通常设定为 2.68MHz。

回到图 7，SCPUIF134 有时在 S-CPU18 与 C-CPU34 之间互相提供中断 IRQ，有时从 S-CPU18 提供给 C-CPU34 复位信号、中断 NMI，或者有时在 S-CPU18 与 C-CPU34 之间互相提供向量。为此，该 SCPUIF134 接收 S-CPU18 的地址 SA 和数据 SD，以及 C-CPU34 的地址 CA 和数据 CD。

该 SCPUIF134 详细示于图 14—图 16 中。如图 15 所示，它包括：接收 S-CPU18 数据低 8 位 SD0—SD7 的最高有效位 SD7 的 C-CPU

控制寄存器 136，接收高端第三位 SD5 的 C-CPU 控制寄存器 138，接收高端第四位 SD4 的 C—CPU 控制寄存器 140，以及接收低端 4 位 SD3—SD0 的 C-CPU 控制寄存器 142。此外还包括分别接收数据 SD0—SD7 全部各位的译码器 144 和 146，以及向量寄存器 148、150、152、154、156 和 158。后面述及的时钟信号发生器 216 输出的 S-CPU18 用的系统时钟信号 SYSCK 经反相器 160 反相，提供作为全部这些 C-CPU 控制寄存器、译码器以及向量寄存器 136—158 的时钟信号。

图 7 地址译码器 50 输出的，S-CPU18 写入控制 C-CPU34 用的控制信号的定时信号 CCNTW 提供作为 C-CPU 控制寄存器 136—142 的起动信号。图 7 地址译码器 50 输出的，S-CPU18 写入控制中断用的控制信号的定时信号 SIFW 以及 SICW 提供作为译码器 144 和 146 的起动信号。图 7 地址译码器 50 输出的，S-CPU18 写入复位向量用的定时信号 RVLW 和 RVHW 提供作为复位向量寄存器 148 以及 150 的起动信号。图 7 地址译码器 50 输出的，S-CPU18 写入 NMI 向量用的定时信号 CNVLW 和 CNVHW 提供作为 NMI 向量寄存器 152 和 154 的起动信号。此外，图 7 地址译码器 50 输出的，S-CPU18 写入 IRQ 向量的定时信号 CIVLW 和 CIVHW 提供作为 IRQ 向量寄存器 156 和 158 的起动信号。

此外，如图 16 所示，SCPUIF134 还包括：接收来自 C-CPU34 的数据 CD 其低 8 位 CD7—CD0 中最高有效位 CD7 的 S-CPU 控制寄存器 162，以及接收低 8 位 CD7—CD0 中 CD6—CD0 等 7 位的 S-CPU 控制寄存器 164。数据 CD 的低 8 位 CD7—CD0 还提供给两个译码器 160 和 168，以及四个向量寄存器 170、172、174 和 176。时钟信号发生器 216 输出的 C-CPU34 用的时钟信号 PH12 经反相器 178 反相后，提供作为这些寄存器和译码器 162—176 的时钟信号。

图 7 地址译码器 50 输出的，C-CPU34 写入控制 S-CPU18 用的控制信号的定时信号 SCNTW，提供作为 S-CPU18 控制寄存器 162 和 164 的起动信号。图 7 地址译码器 50 输出的，C-CPU34 写入用于

控制 S-CPU34 来的中断的控制信号的定时信号 CIEW 和 CICW，提供作为译码器 166 的起动信号。图 7 地址译码器 50 输出的，C-CPU34 写入 NMI 向量用的定时信号 SNVLW 和 SNVHW，提供作为 NMI 向量寄存器 170 和 172 的起动信号。图 7 地址译码器 50 输出的，C-CPU34 写入 IRQ 向量用的定时信号 SIVLW 和 SIVHW，提供作为 IRQ 向量寄存器 174 和 176 的起动信号。

C—CPU 控制寄存器 136 的输出 Q，同译码器 166 输出的用于许可从 C-CPU34 对 S-CPU18 进行 IRQ 中断的信号 SIRQEN 一起提供给“与”门 180 后，由该“与”门 180 输出一信号 CIRQ，用来由 S-CPU18 向 C-CPU34 加 IRQ 中断。另外，C-CPU 控制寄存器 136 由译码器 168 输出的用来结束 S-CPU18 发出的 IRQ 中断的信号 SIRQCL 复位。C-CPU 控制寄存器 138 的输出 Q 照原样输出作为 S-CPU18 使 C-CPU34 复位用的信号 CRES。D-FF140 的输出 Q，同译码器 166 输出的用于许可从 C-CPU34 对 S-CPU18 进行 NMI 中断的信号 SNMIEN 一起提供给“与”门 182 后，由该“与”门 182 输出一信号 CNMI，用来向 C-CPU34 加 NMI 中断。另外，D-FF140 由译码器 168 输出的用来结束 NMI 中断的信号 SNMICL 复位。C-CPU 控制寄存器 142 输出用于从 S-CPU18 提供给 C-CPU34 的 4 位信号 SMEG3—SMEG0。这 4 位信号是为了表示中断处理的状态而采用的，例如 S-CPU18 向 C-CPU34 加 NMI 中断的同时，在 C—CPU 控制寄存器 142 的 SMEG3—SMEG0 中写入 0，由此使可以知道向 C-CPU34 加了 0 号 NMI 中断。若采用 SMEG3—SMEG0 4 位的话可以表示 0—15 的数字，因而 S-CPU18 可以向 C-CPU34 加 16 种 NMI 中断。同样，IRQ 中断的时候若采用 SMEG3—SMEG0，便可以加 16 种 IRQ 中断。

S-CPU 控制寄存器 162 的输出，同译码器 144 输出的用于许可从 S-CPU18 对 C-CPU34 进行 IRQ 中断的信号 CIRQEN 一起提供给“与”门 184，该“与”门 184 输出一信号 SIRQ，用于由 C-CPU34 向 S-CPU18 加 IRQ 中断。另外，S-CPU 控制寄存器 162 由译码器 144 输出的用于结束来自 C-CPU34 的 IRQ 中断的信号 CIRQCL 来复位。

S-CPU 控制寄存器 164 输出用于由 C-CPU34 提供给 S-CPU18 的 4 位信号 CMEG3—CMEG0。这 4 位信号是用来表示中断处理状态的，例如，C-CPU34 向 S-CPU18 加 IRQ 中断的同时，在 S-CPU 控制寄存器 164 的 CMEG3—CMEG0 中写入 0，由此可知向 S-CPU18 加了 0 号 IRQ 中断。若采用 CMEG3—CMEG0 4 位，可表示 0—15 的数字，因而 C-CPU34 可以向 S-CPU18 加 16 种 IRQ 中断。同样，如后面所述，C-CPU34 修改 S-CPU18 的 NMI 向量时采用 CMEG3—CMEG0 的话，可以对 16 种 NMI 向量进行修改。

另外，“与”门 180 和 182 的输出经过缓冲器 186 和 188，而且 C-CPU 控制寄存器 142 的输出经过缓冲器 190 后，输出作为至 S-CPU18 的数据 SD7—SD0。复位向量寄存器 148 和 150 的输出经过缓冲器 192 和 194 后，作为复位向量数据 RVL7—RVL0 和 RVH7—RVH0，输出至数据 SD7—SD0。NMI 向量寄存器 152 和 154 的输出通过缓冲器 196 和 198 后，作为 NMI 向量数据 CNVL7—CNVL0 和 CNVH7—CNVH0 输出至数据 SD7—SD0。IRQ 向量寄存器 156 和 158 的输出通过缓冲器 200 和 202 后，作为 IRQ 向量数据 CIVL7—CIVL0 和 CIVH7—CIVH0，输出至数据 SD7—SD0。

而且，“与”门 184 的输出通过缓冲器 204，S-CPU18 控制寄存器 164 的输出经过缓冲器 206 后，输出作为至 C-CPU34 的数据 CD7—CD0。NMI 向量寄存器 170 和 172 的输出通过缓冲器 208 和 210 后，作为 NMI 向量数据 SNVL7—SNVL0 和 SNVH7—SNVH0，输出至数据 CD7—CD0。IRQ 向量寄存器 174 和 176 的输出通过缓冲器 212 和 214 后，作为 IRQ 向量数据 SIVL7—SIVL0 和 SIVH7—SIVH0，输出至数据 CD7—CD0。

如图 7 所示的时钟信号发生器 216 接收基本振荡器 23(图 3)传来的频率约为 21MHz 的基本时钟信号，形成 S-CPU18 及 C-CPU34 所需的系统时钟信号 SYSCK 及时钟信号 PHI20 等。参照图 17，时钟信号发生器 216 接收基本时钟信号 Z21M，使其通过缓冲存储器 218，作为图 18 或图 19 所示那样的时钟信号 21M 输出。时钟信号

21M 在提供给 D-FF220 及 222 的时钟信号输入端的同时，经反相器 226 倒相后，提供给 D-FF224 的时钟信号输入端。又将系统复位信号 RES 提供给 D-FF-220—224 的各复位输入端。经缓冲存储器 228，对 D-FF220 的数据输入端提供来自时钟信号发生器 25(图 3)的基本系统时钟信号 ZSYSCK，而 D-FF220 的输出提供给 D-FF-222 的数据输入端。D-FF222 的输出 Q 与 D-FF220 的输出 Q 的倒相信号一起送给“与”门 230。“与”门 230 的输出倒相后送至“与”门 232 的一输入端，另一输入端供以 D-FF224 的输出 Q 的倒相信号。而 D-FF224 的输出 Q 通过缓冲存储器 234、作为图 18 或图 9 所示的那样的时钟信号 PH12 输出。亦即，时钟信号 PH12 是基本时钟脉冲信号 21M 的 2 分频信号。

基本系统时钟信号 ZSYSSK 经缓冲存储器 228，作为图 18 或图 19 所示那样的系统时钟信号 SYSCK 输出。

“与”门 230 的输出供给 D-FF236 的数据输入端。该 D-FF236 的复位输入，与其他 D-FF238、240 及 242 相同，由系统复位信号 RES 提供。而且，作为这些 D-FF236—242 的时钟信号输入，提供时钟信号 PH12。来自缓冲存储器 228 的系统时钟信号 SYSCK 经倒相后输入“与”门 224 及 226 的一输入端，该“与”门 224 及 226 的另一输入端接收 D-FF240 的输出 Q。“与”门 224 的输出供给 D-FF-238 作数据输入。在 S-CPU18 从 M-ROM22 读出程序数据时成为低电平的信号 SCPURDB 以及在 S-CPU18 对 B-RAM32 或 I-RAM36 读出数据时成为低电平，而写入数据时成为高电平的 SCPUWRD 分别倒相，与此同时，D-FF238 的输出 Q 提供给“或”门 248。D-FF236 的输出 Q 与“与”门 246 的输出一起，经“或”门 250，提供给 D-FF240 的输入端。D-FF240 的输出 Q 和“或”门 28 的输出一起提供到“与”门 252 的 K 输入端，“与”门 252 的输出与“或”门 250 的输出一起输入“或”门 254。“或”门 254 的输出提供给 D-FF242 的数据输入端。

D-FF242 的输出 Q 在 S-CPU18 以 3.58MHz 工作时，作为允许 S-

CPU18 存取存储器 22、32 或 36 的信号 SSSYNC 输出，同时 D-FF242 的输出 Q 作为允许 S-CPU18 存取存储器 22、32 或 36 的定时信号 SSYNC 输出。亦即，信号 SSYNC 如图 18 或图 19 所示，具有比系统时钟信号 SYSCK(在图 18 或图 19 以 SYSCK-FAST 或 SYSCK-SLOWS 表示)的后缘略为超前的后缘，信号 SSSYNC 具有与 SSYNC 前缘同步的前缘，且持续相当于基本时钟信号 21M 的两个周期后再出现后缘。

这样由时钟信号发生器 216 形成 S-CPU18 的动作时钟信号 SYSCK 及 C-CPU34 的动作时钟信号 PH12，同时，形成允许或禁止 S-CPU18 及 C-CPU34 存取存储器 22、32 或 36 的信号 SSYNC(或 SSSYNC)。

为了使系统时钟信号 SYSCK 与时钟信号 PH12 对好相位，使用基本时钟信号 21M，如图 18 或图 19 所示，时钟信号 SYSCK 能与该信号 21M 取得同步是在 SYSCK-FAST 或 SYSCK-SLOW 时。S-CPU18 在该时钟信号 SYSCK-FAST 或 SYSCK-SLOW 的后缘读入从存储器 22、32 或 36 读出的数据。因而，在 S-CPU18 存取这些存储器时间，必须超前于该时钟信号 SYSCK 的后缘。于是，在图 17 的时钟信号发生器 216 中，信号 SSYNC 或 SSSYNC 的后缘存在于较时钟脉冲信号 SYSCK 的后缘超前的地方。又，S-CPU18 以 3.58MHz 工作时，如图 19 所示，时钟信号 SYSCK 的相位发生变化，因此，一旦使用信号 SSYNC 作为允许 S-CPU18 存取的信号，有时就会发生与读入数据的前后动作周期相重迭。于是采用如图 19 所示的，高电平持续时间比信号 SSYNC 中的短的信号 SSSYNC。

于是，如前所述，在该信号 SSYNC (SSSYNC)的高电平时间里允许 S-CPU18 存取存储器，在低电平时间里允许 C-CPU34 存取存储器，由此，S-CPU18 及 C-CPU34 可以实质上同时以两相存取各存储器 22、32 或 36。

这里参照图 20—图 25，对 S-CPU18 与 C-CPU34 通过 SCPUIF134 的相互作用加以说明。

S-CPU18 在使 C-CPU34 起动时，如图 20 的步骤 S1 所示，S-CPU18 用数据 SD0—SD7，在复位向量寄存器 148 及 150 设定想要使 C-CPU34 执行的程序的起始地址。然后，在步骤 S2 中，在 C-CPU 控制寄存器 138 用数据位 SD5 设定“1”。相应地、从 C-CPU 控制寄存器 138 输出高电平的 C-CPU 复位信号 CRES，如图 7 所示，该信号送给 C-CPU34，C-CPU34 的复位被解除。而且，C-CPU34 相应地存取 M-ROM22，从复位向量寄存器 148 及 150 指定的 M-ROM22 的起始地址开始执行程序。

如图 21 所示，S-CPU18 使 C-CPU34 停止时，S-CPU18 用数据位 SD5 在 C-CPU 控制寄存器 138 上设定“0”。相应地，从 C-CPU 控制寄存器 138 输出低电平的 C-CPU 复位信号 CRES，如图 7 所示该信号送给 C-CPU34，C-CPU 复位。

S-CPU18 对 C-CPU34 执行不可屏蔽中断(NMI)时，如图 22 的步骤 S4 所示，S-CPU18 用数据 SD0—SD7 在 NMI 向量寄存器 152 及 154 设定 M-ROM22 存储 NMI 程序的起始地址。然后，在步骤 S6 中，在 C-CPU 控制寄存器 140 上用数据位 SD4 设定“1”。相应地，从 C-CPU 控制寄存器 140 输出高电平的 C-CPU 中断信号 CNMI，如图 7 所示供给 C-CPU34，使 C-CPU34 中断起动。而 C-CPU34 相应存取 M-ROM22，根据 NMI 向量寄存器 152 及 154 指定的 M-ROM22 的起始地址执行 NMI 中断程序。

S-CPU18 对 C-CPU34 执行 IRQ 中断时，如图 23 的步骤 S7 所示，S-CPU18 用数据 SD0—SD7 在 IRQ 向量寄存器 156 及 158 设定 M-ROM22 存储 IRQ 程序的起始地址。然后，在步骤 S8，在 C-CPU 控制寄存器 136 上用数据位 SD7 设定“1”。相应地、从 C-CPU 控制寄存器 136 输出高电平的 C-CPU 中断信号 CIRQ，如图 7 所示提供给 C-CPU34，使 C-CPU34 中断起动。而且，相应地，C-CPU34 存取 M-ROM22，根据 IRQ 向量寄存器 156 及 158 指定的 M-ROM22 的起始地址执行 IRQ 中断程序。

接着，C-CPU34 改变 S-CPU18 的 NMI 向量时，如图 24 的步骤 S10

所示，C-CPU34 用数据 CD0—CD7，在 NMI 向量寄存器 170 及 172 上设定 M-ROM22 存储 NMI 程序的起始地址。然后，在步骤 S11，在 S-CPU 控制寄存器 162 上用数据位 CD4 设定“1”。相应地，S-CPU18 中断时被读出的 NMI 向量地址从 M-ROM22 内转换到 NMI 向量寄存器 170 与 172。而且，S-CPU18 根据已施加 NMI 中断的状况，存取 M-ROM22，从 NMI 向量寄存器 170 及 172 指定的 M-ROM 的起始地址开始执行 NMI 中断。

然后，C-CPU34 对 S-CPU18 实行 IRQ 中断时，如图 25 的步骤 S12 所示，由数据 CD0—CD7 在 IRQ 向量寄存器 174 及 176 设定 M-ROM22 存储 IRQ 程序的起始地址。然后，在步骤 S13，由数据位 CD7 在 S-CPU 控制寄存器 162 设定“1”，相应地，从 S-CPU 控制寄存器 162 输出高电平的 S-CPU 中断信号 SIRQ，该信号送给 S-CPU18，使 S-CPU18 中断起动。而且，S-CPU18 相应地存取 M-ROM22，根据 IRQ 向量寄存器 174 及 176 指定的 M-ROM22 的起始地址开始执行 IRQ 中断程序。

在实施例的游戏系统 10 中，协处理器 30 的 C-CPU34，主要是在(a)加速模式；(b)混合处理模式；(c)代行处理模式中发挥其作用。

(a)加速模式

在这一模式中，如图 26 所示，C-CPU34 被置于停止状态。然后，在步骤 S101，S-CPU18 用数据 SD0—SD7 在复位向量寄存器 148 及 150 设定想让 C-CPU34 执行的程序的起始地址。然后，在步骤 S102，S-CPU18 用数据位 SD5 在 C-CPU 控制寄存器 138 上设定“1”。相应地，从 C-CPU 控制寄存器 138 输出高电平的 C-CPU 复位信号 CRES，该信号送给 C-CPU34，C-CPU34 的复位被解除。从而，在图 26 的步骤 S201，C-CPU34 存取 M-ROM22，根据指定的 M-ROM22 的起始地址执行程序。

在步骤 S202，C-CPU34 为了将程序执行结果送给 S-CPU18，用数据位 CD7 在 S-CPU 控制寄存器 162 上设定“1”。相应地，从 S-CPU

控制寄存器 162 及“与”门 184 输出高电平的 S-CPU 中断信号 SIRQ，该信号送给 S-CPU18，对 S-CPU18 进行中断。

S-CPU18 执行步骤 S103、S104 及 S105 直到在步骤 S6 检测出上述 IRQ 为止。亦即，S-CPU18，为了能够执行来自 C-CPU34 的 IRQ 中断，在步骤 S103，首先使 INT 起动寄存器 144 起动。相应地，从该寄存器 144 输出信号 CIRQEN，该信号如图 16 所示，送给“与”门 184。因而，通过该“与”门后，如上所述，可以由 C-CPU34 对 S-CPU18 进行 IRQ 中断。然后，在步骤 S104，S-CPU18 移至主机 12 内的 W-RAM24，等待 C-CPU34 来的 IRQ 中断(程序等待)。

在步骤 S106，检测出 C-CPU34 来的 IRQ，接收 C-CPU34 处理的结果后，在步骤 S107，S-CPU18 用数据位 SD5 在 C-CPU 控制寄存器 138 设定“0”。相应地，从 C-CPU 控制寄存器 138 输出低电平的 C-CPU 复位信号 CRES，这信号供给 C-CPU34，使 C-CPU34 复位，处于停止状态。

在这样的加速模式中，如图 27 所示，由 C-CPU34 处理费时间的子程序(例如碰撞判定、3D 座标运算等)，从而 S-CPU18 可以以极高速度执行总体处理。

(b)混合处理模式

在这种情况下，如同在图 28 的步骤 S211，C-CPU34 进行着并行处理(执行别的程序)。然后，如图 28 的步骤 S111 所示，S-CPU18 用数据 SD0—SD7 在 NMI 向量寄存器 152 及 154 设定 M-ROM22 存储 NMI 程序的起始地址。然后，在步骤 S112，在 C-CPU 控制寄存器 140 用数据位 SD4 设定“1”。相应地，从 C-CPU 控制寄存器 140 输出高电平的 C-CPU 中断信号 CNMI，该信号送给 C-CPU34，使 C-CPU34 中断起动。而且，在步骤 S212，C-CPU34 读入中断标志，在步骤 S213 检查是否有来自 S-CPU18 的 NMI 中断。一旦检测到来自 S-CPU18 的 NMI 中断，在步骤 S214、C-CPU34 即执行前面说明过的加速处理。

为了将程序执行结果送到 S-CPU18，在步骤 S215，C-CPU34 借

助于数据位CD7在S-CPU控制寄存器162设定“1”。相应地，从S-CPU控制寄存器162及“与”门184输出高电平的S-CPU中断信号SIRQ，该信号送给S-CPU18，使S-CPU18中断。亦即，S-CPU18根据指定的M-ROM22的起始地址执行IRQ中断程序。

在步骤S116，S-CPU18在检测出上述IRQ之前执行步骤S113、S114及S115。亦即，为了使来自C-CPU34的IRQ中断成为可能，S-CPU18首先在步骤S113使INT起动寄存器144起动。相应地，该寄存器144输出信号CIRQEN、该信号供给图16所示的“与”门184，从而，通过该“与”门184后如上所述，可以由C-CPU34给S-CPU18以IRQ中断。然后，在步骤S114 S-CPU18移至主机12内的W-RAM24，等待来自C-CPU34的IRQ中断(程序等待)。

在步骤S115，S-CPU18读取中断标志，在步骤S116检测出C-CPU34来的IRQ。在步骤S117，S-CPU18返回主程序。又，在步骤S216,C-CPU34恢复为先前S-CPU18来的中断所中止的并行处理(执行别的程序)。

这样的混合处理模式,特别是在并行处理时,协处理器(C-CPU34)有余量，两个处理器S-CPU18及C-CPU34的处理速度想要取得平衡的情况下，以及对于处理频度较高的S-CPU18一侧的子程序(各种运算子程序；旋转、扩大、缩小参数的计算等)是有效的。亦即，即使两CPU并行处理时也可以将C-CPU34作为S-CPU18的加速器起动，因此，两CPU的处理速度可以取得平衡。

(c)代行处理模式

在这一模式，如图30所示，在步骤S221 C-CPU34处于停止状态。然后，在步骤S121，S-CPU18用数据SD0—SD7在复位向量寄存器148及150设定想要让C-CPU34执行的程序的起始地址。然后，在步骤122，S-CPU18在C-CPU控制寄存器138上用数据位SD5设定“1”。C-CPU控制寄存器138响应输出高电平的C-CPU复位信号CRES，该信号送给C-CPU34，C-CPU34的复位被解除。因而，在图30的步骤S222，C-CPU34存取M-ROM22，根据指定的M-ROM22

的起始地址执行程序。

然后,S-CPU18 按步骤 S123 及步骤 S124 执行 DMA。这时,C-CPU34 取代 S-CPU18 执行程序(代行处理)。

为了将程序执行结果送到 C-CPU34，在步骤 S223，C-CPU34 用数据位 CD7 在 S-CPU 控制寄存器 162 设定“1”。相应地，从 S-CPU 控制寄存器 162 及“与”门 184 输出高电平的 S-CPU 中断信号 SIRQ，该信号提供给 S-CPU18，使 S-CPU18 中断。

在步骤 S128，S-CPU18 在检测出上述 IRQ 之前执行步骤 S125、S126 及 S127。亦即，为了使来自 C-CPU34 的 IRQ 中断成为可能，首先，在步骤 S125 S-CPU18 使 INT 起动寄存器 114 起动。相应地，从该寄存器 144 输出信号 CIRQEN，该信号输送给图 16 所示的“与”门 184。因而，通过该“与”门 184 后，如上所示，可以由 C-CPU34 给 S-CPU18 IRQ 中断。然后，在步骤 S126，S-CPU18 移至主机 12 内的 W-RAM24，等待 C-CPU34 来的 IRQ 中断(程序等待)。

在步骤 S127，S-CPU18 读取中断标志，在步骤 S128 检测出来自 C-CPU34 的 IRQ，接收到 C-CPU34 的处理结果后，在步骤 S129，S-CPU18 用数据位 SD5 在 C-CPU 控制寄存器 138 上设定“0”。相应地，从 C-CPU34 控制寄存器 138 输出低电平的 C-CPU 复位信号 CRES，该信号送给 C-CPU34，使 C-CPU34 复位，处于停止状态(步骤 S224)。

在这样的代行处理模式中，如图 31 所示，S-CPU18 在 DMA 时停止，因而，一向串行的处理由 C-CPU34 代行，从而可望大幅度提高总体处理速度。因此，这种代行处理模式对于例如压缩图像的展开、动画等，反复进行与 DMA 平行处理的子程序特别有效。

而且，在上述实施例中，M-ROM22、B-RAM32 及 I-RAM36 等存储器的存取时间，不仅比 S-CPU18，即使与 C-CPU34 的周期时间相比，也是短的。也就是说，存储器的存取速度比 S-CPU18 和 C-CPU34 的动作速度快。但是，不需要所有的存储器都满足这一条件，只要两个 CPU 以及可能为处理器存取的存储器满足这一条件即可。

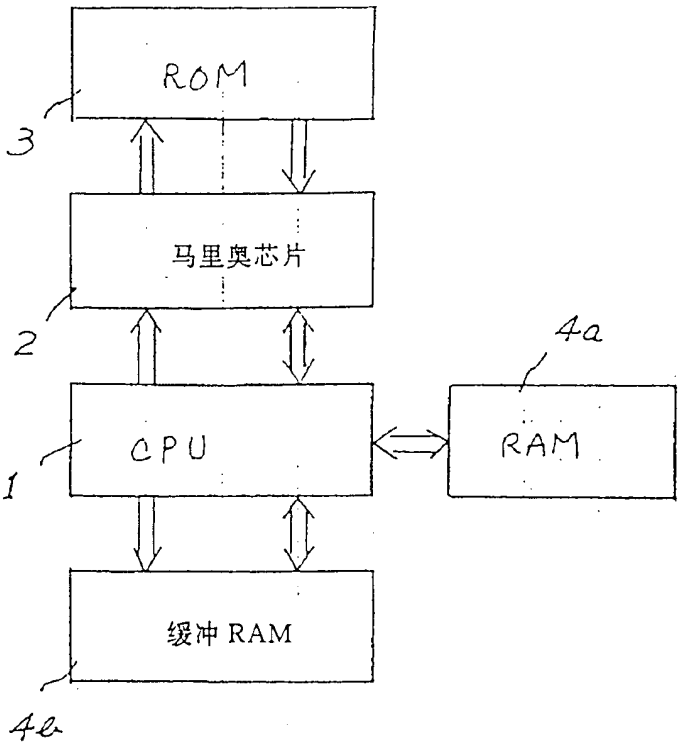


图 1

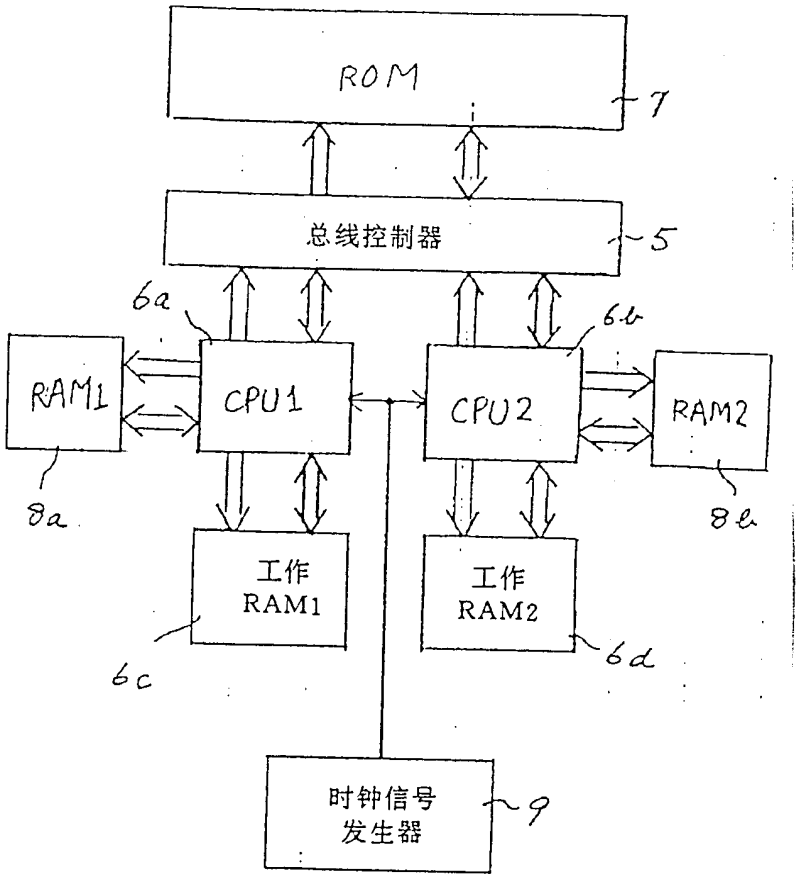

ROM
7
总线控制器
5
6a
6b
RAM1
CPU1
CPU2
RAM2
8a
8b
工作
RAM1
工作
RAM2
6c
6d
时钟信号
发生器
9

图 2

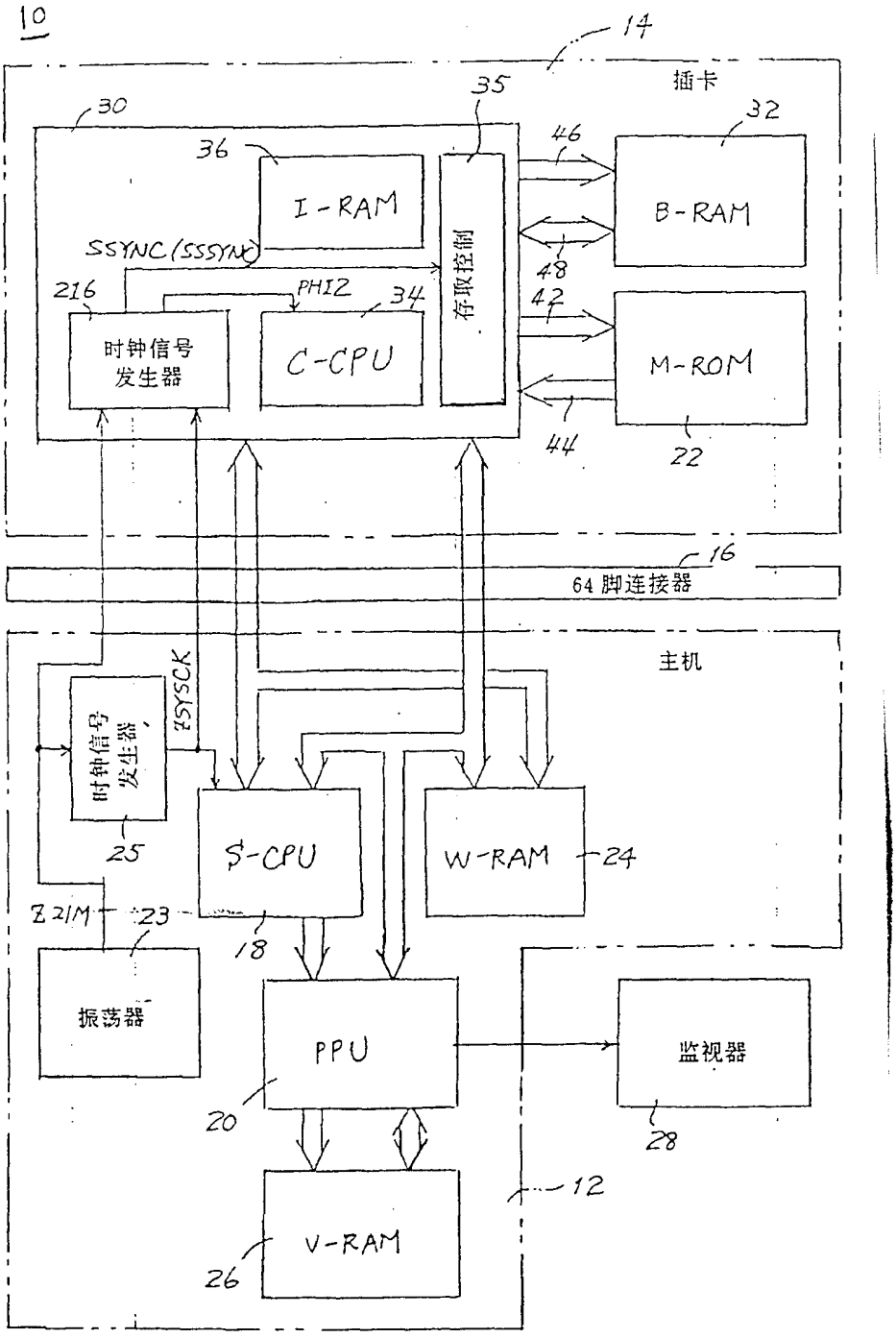

10
14
插卡
35
30
32
36
46
I-RAM
B-RAM
存取控制
SSYNC(SSYN)
48
216
PHI2
34
42
时钟信号
发生器
C-CPU
M-ROM
44
22
16
64 脚连接器
主机
25SYSCK
时钟信号
发生器
25
S-CPU
W-RAM
24
Z21M
23
18
振荡器
PPU
监视器
20
28
12
V-RAM
26

图 3

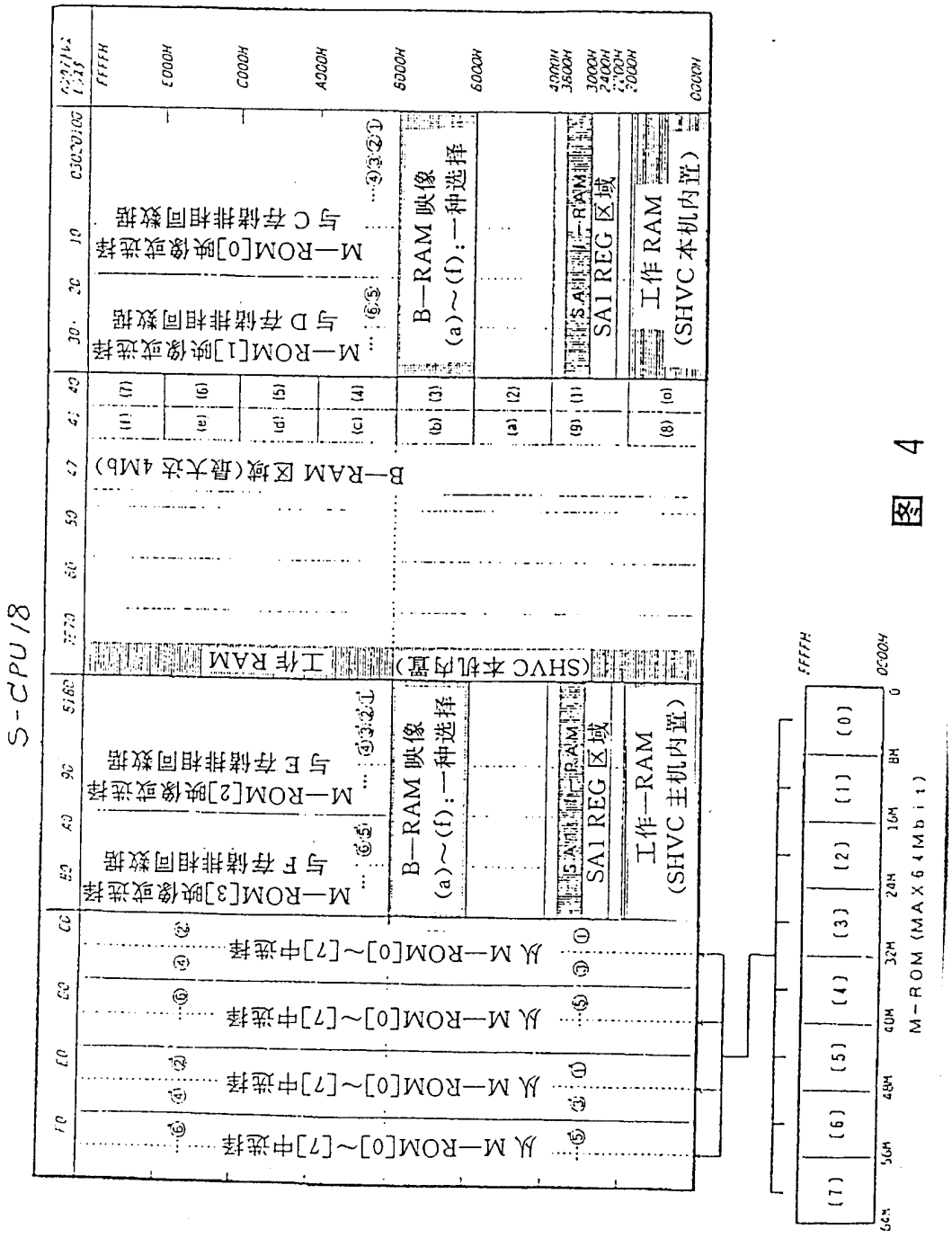

S-CPU18
从 M—ROM[0]～[7]中选择
从 M—ROM[0]～[7]中选择
从 M—ROM[0]～[7]中选择
从 M—ROM[0]～[7]中选择
M—ROM[3]映像或选择
与 F 存储排相同数据
M—ROM[2]映像或选择
与 E 存储排相同数据
M—ROM[1]映像或选择
与 D 存储排相同数据
M—ROM[0]映像或选择
与 C 存储排相同数据
B—RAM 映像
(a)～(f)：一种选择
SA1 REG 区域
工作—RAM
(SHVC 主机内置)
工作 RAM
(SHVC 本机内置)
B—RAM 区域(最大达 4Mb)
M-ROM (MAX64Mbit)

图 4

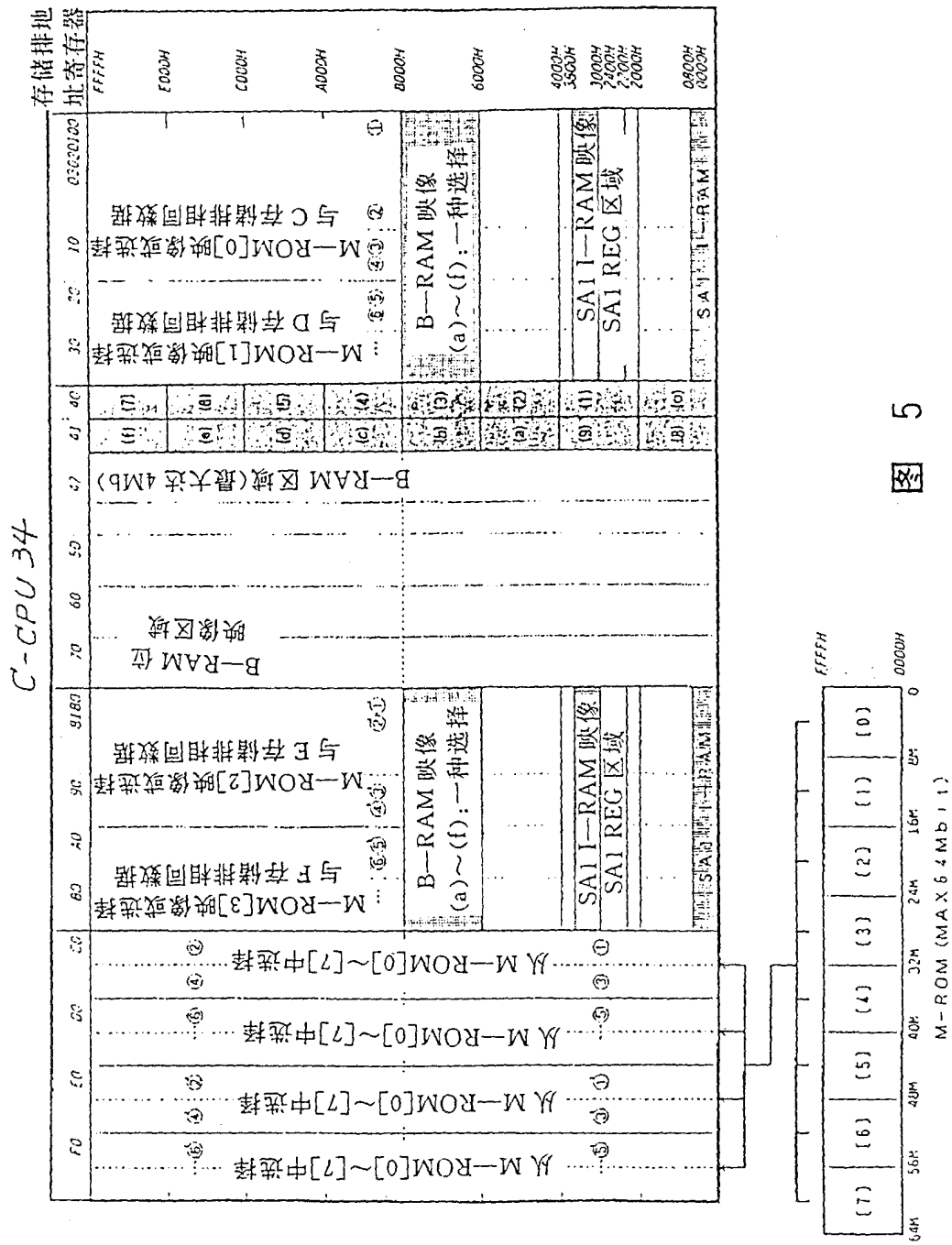

C-CPU 34
存储排地址寄存器
B—RAM 区域(最大达 4Mb)
B—RAM 位映像区域
从 M—ROM[0]～[7]中选择
M—ROM[3]映像或选择
与 F 存储排相同数据
M—ROM[2]映像或选择
与 E 存储排相同数据
M—ROM[1]映像或选择
与 D 存储排相同数据
M—ROM[0]映像或选择
与 C 存储排相同数据
B—RAM 映像
(a)～(f):一种选择
SA1 I—RAM 映像
SA1 REG 区域
SA1 I—RAM
M-ROM (MAX 64Mbit)
FFFFH
0000H

图 5

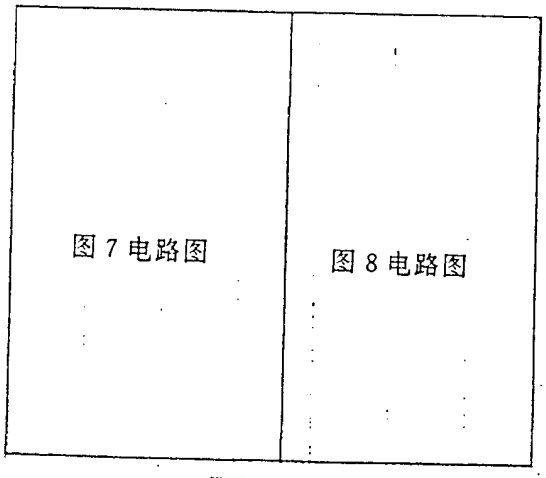


图 6

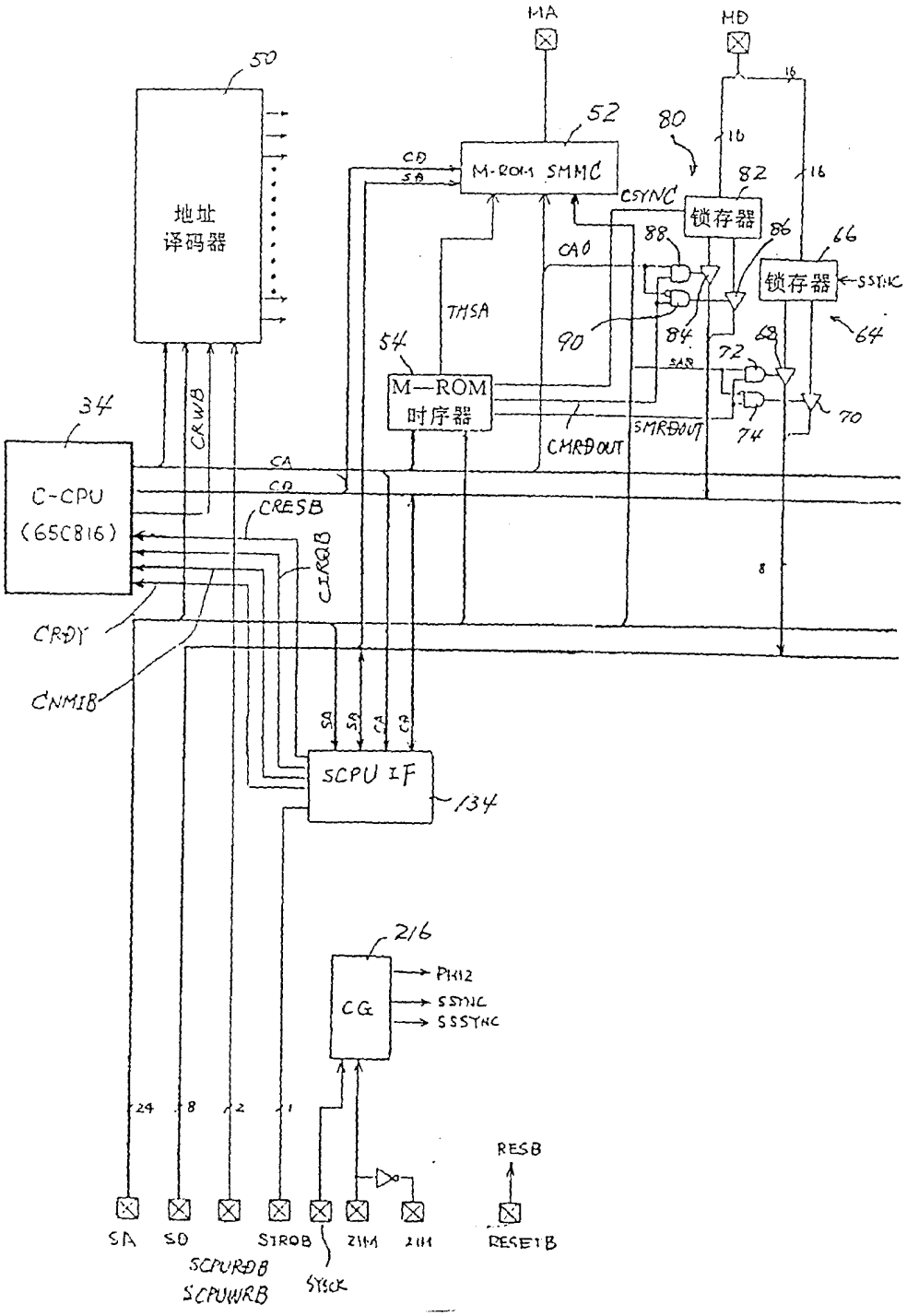

MA
MD
50
地址
译码器
52
M-ROM SMMC
80
82
锁存器
66
锁存器
SSYNC
64
CSYNC
CD
SA
CA0
TMSA
88
86
90
84
72
68
70
74
54
M—ROM
时序器
SMRDOUT
CMRDOUT
34
C-CPU
(65C816)
CRWB
CA
CD
CRESB
CIRQB
CRDY
CNMIB
SA
SD
CA
CD
SCPU IF
134
216
CG
PH12
SSYNC
SSSYNC
24
8
2
1
RESB
SA
SD
SIROB
Z1M4
Z1M
RESETB
SCPURDB
SCPUWRB
SYSCK

图 7

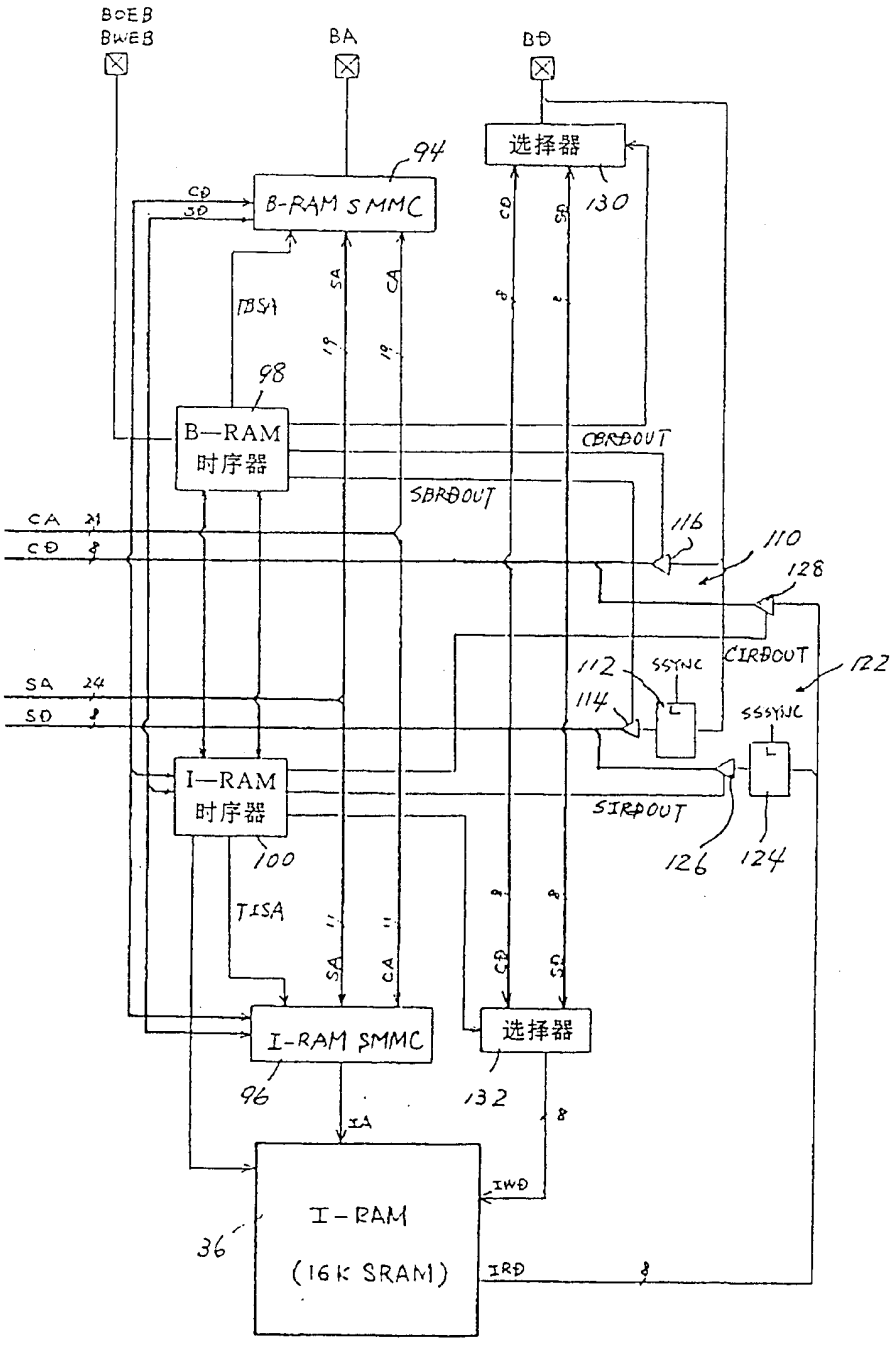

BOEB
BWEB
BA
BD
选择器
94
B-RAM SMMC
130
CD
SD
TBSA
SA
CA
98
B—RAM
时序器
CBRDOUT
SBRDOUT
116
110
128
CA
CD
SA
SD
112
SSYNC
CIRDOUT
122
114
SSSYNC
I—RAM
时序器
SIRDOUT
126
124
100
TISA
I-RAM SMMC
选择器
96
132
IA
IWD
I-RAM
(16K SRAM)
36
IRD

图 8

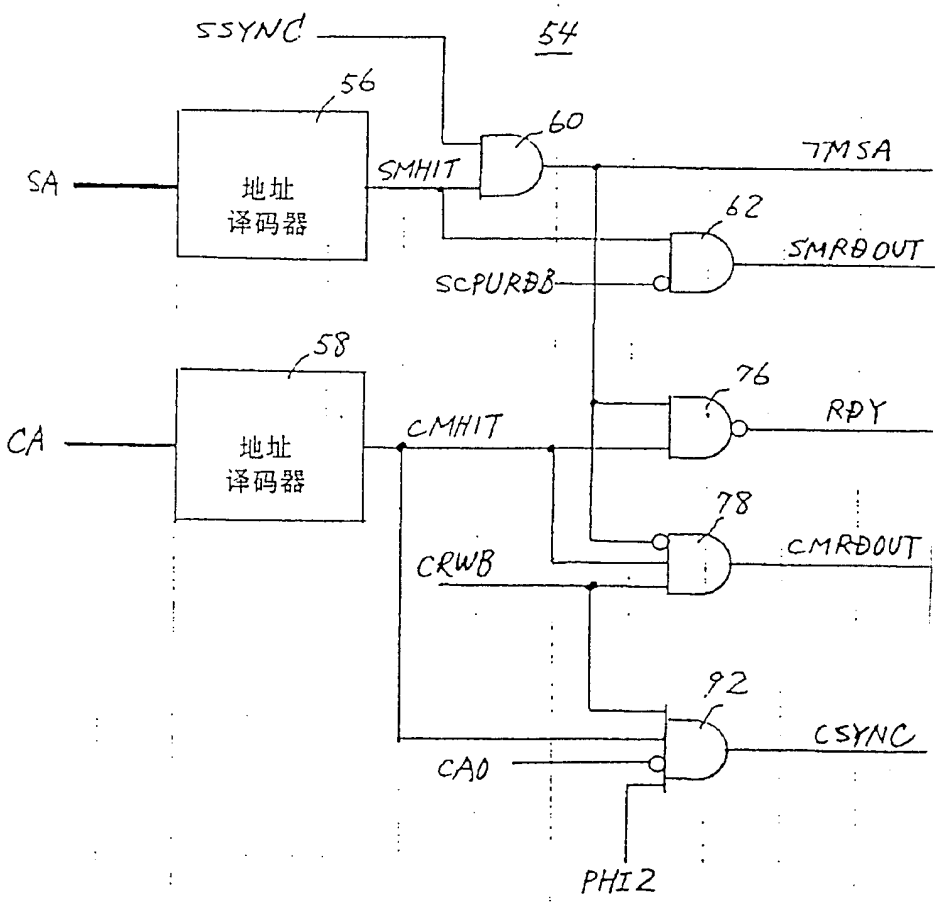

SSYNC
54
56
60
SA
地址
译码器
SMHIT
TMSA
62
SMRDOUT
SCPURDB
58
76
RDY
CA
地址
译码器
CMHIT
78
CMRDOUT
CRWB
92
CSYNC
CA0
PHI2

图 9

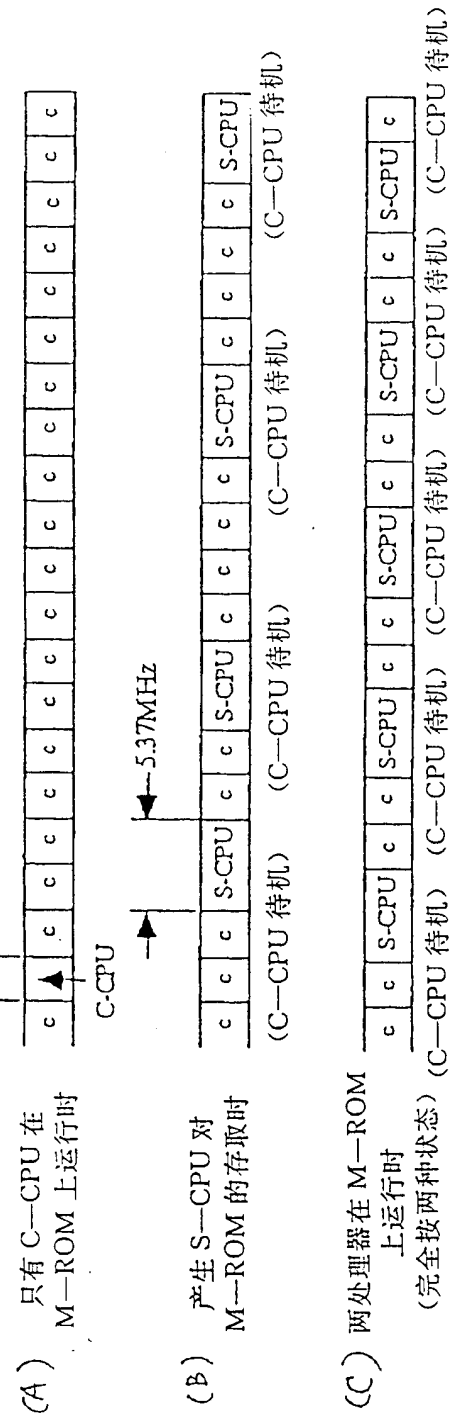

(A) 只有C—CPU在M—ROM上运行时
10.73MHz
C
C-CPU
(B) 产生S—CPU对M—ROM的存取时
5.37MHz
S-CPU
(C—CPU待机)
(C) 两处理器在M—ROM上运行时
(完全按两种状态)

图 10

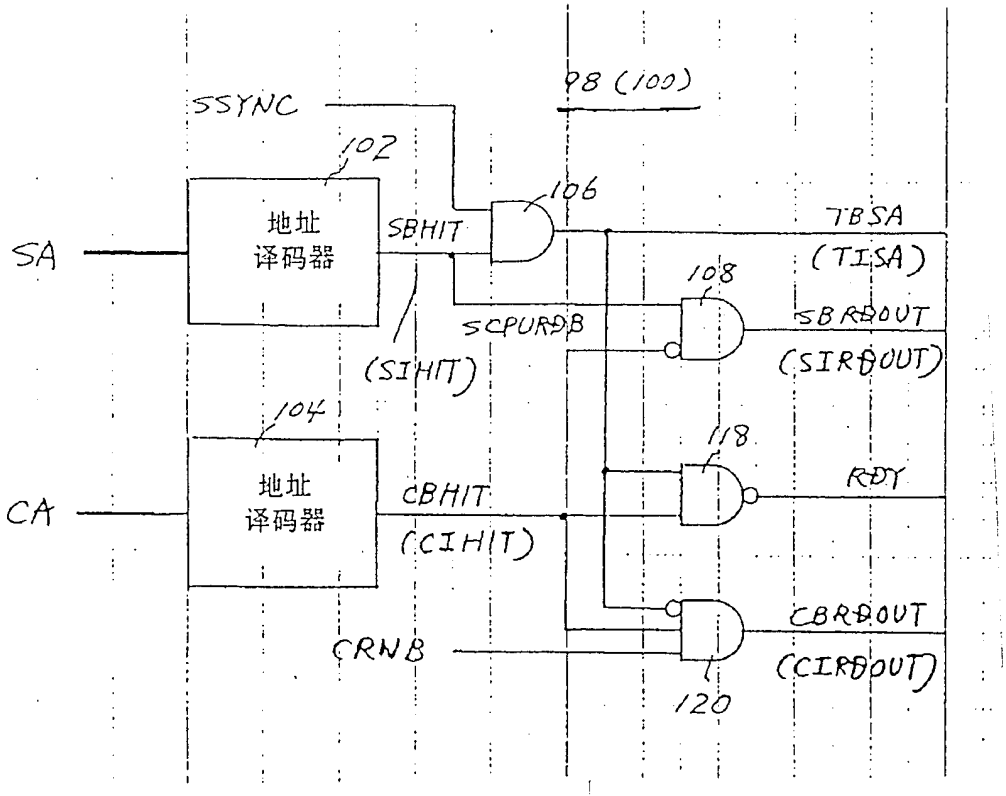

SSYNC
98 (100)
102
106
地址
译码器
SBHIT
SA
TBSA
(TISA)
108
SCPURDB
SBRDOUT
(SIRDOUT)
(SIHIT)
104
118
地址
译码器
CBHIT
(CIHIT)
CA
RDY
CBRDOUT
CRNB
(CIRDOUT)
120

图 11

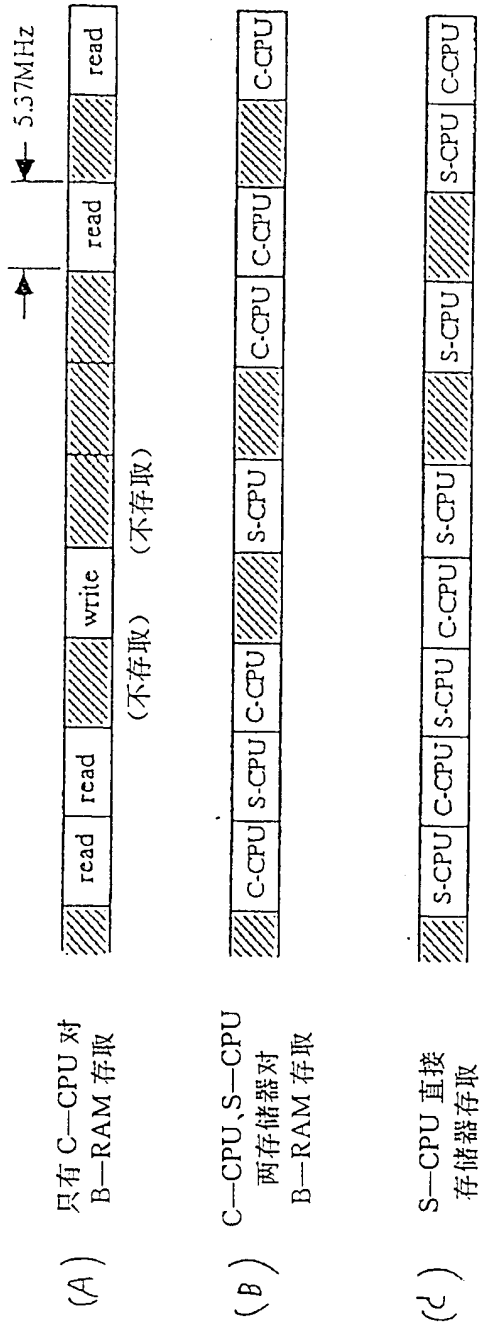

5.37MHz
read
read
write
read
read
(不存取)
(不存取)
C-CPU
S-CPU
C-CPU
S-CPU
C-CPU
C-CPU
C-CPU
C-CPU
S-CPU
C-CPU
S-CPU
C-CPU
S-CPU
S-CPU
S-CPU
S-CPU
(A) 只有 C—CPU 对 B—RAM 存取
(B) C—CPU、S—CPU 两存储器对 B—RAM 存取
(C) S—CPU 直接存储器存取

图 12

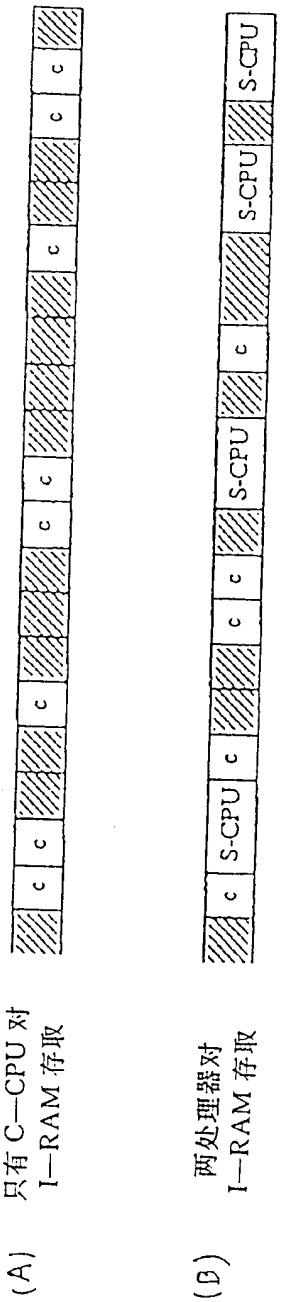


图 13

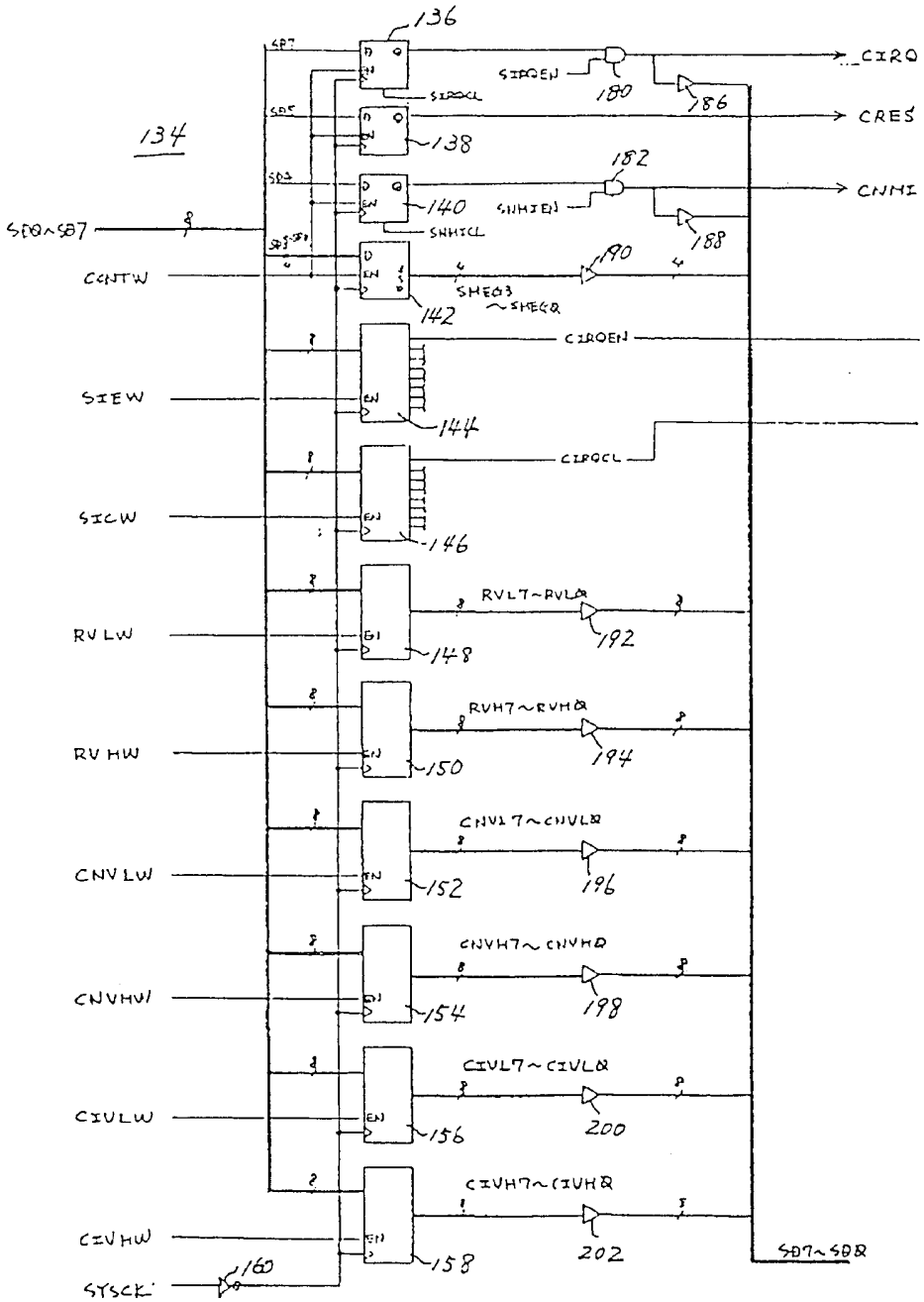

136
SD7
SIRQEN
180
186
CIRQ
SIRQCL
SD5
CRES
138
134
182
SD3
CNNI
140
SNHIEN
SNHICL
188
SD0~SD7
SD1
190
CNTW
SHE03
~SHE00
142
CIRQEN
SIEW
144
CIRQCL
SICW
146
RVL7~RVL0
RVLW
192
148
RVH7~RVH0
RVHW
194
150
CNVL7~CNVL0
CNVLW
196
152
CNVH7~CNVH0
CNVHW
198
154
CIVL7~CIVL0
CIVLW
200
156
CIVH7~CIVH0
CIVHW
202
SD7~SD0
158
SYSCK
160

图 15

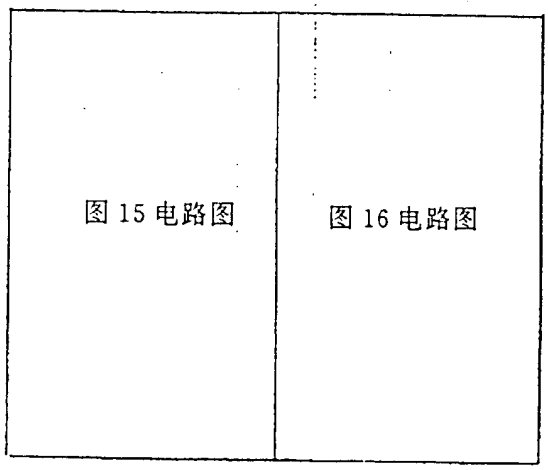


图 14

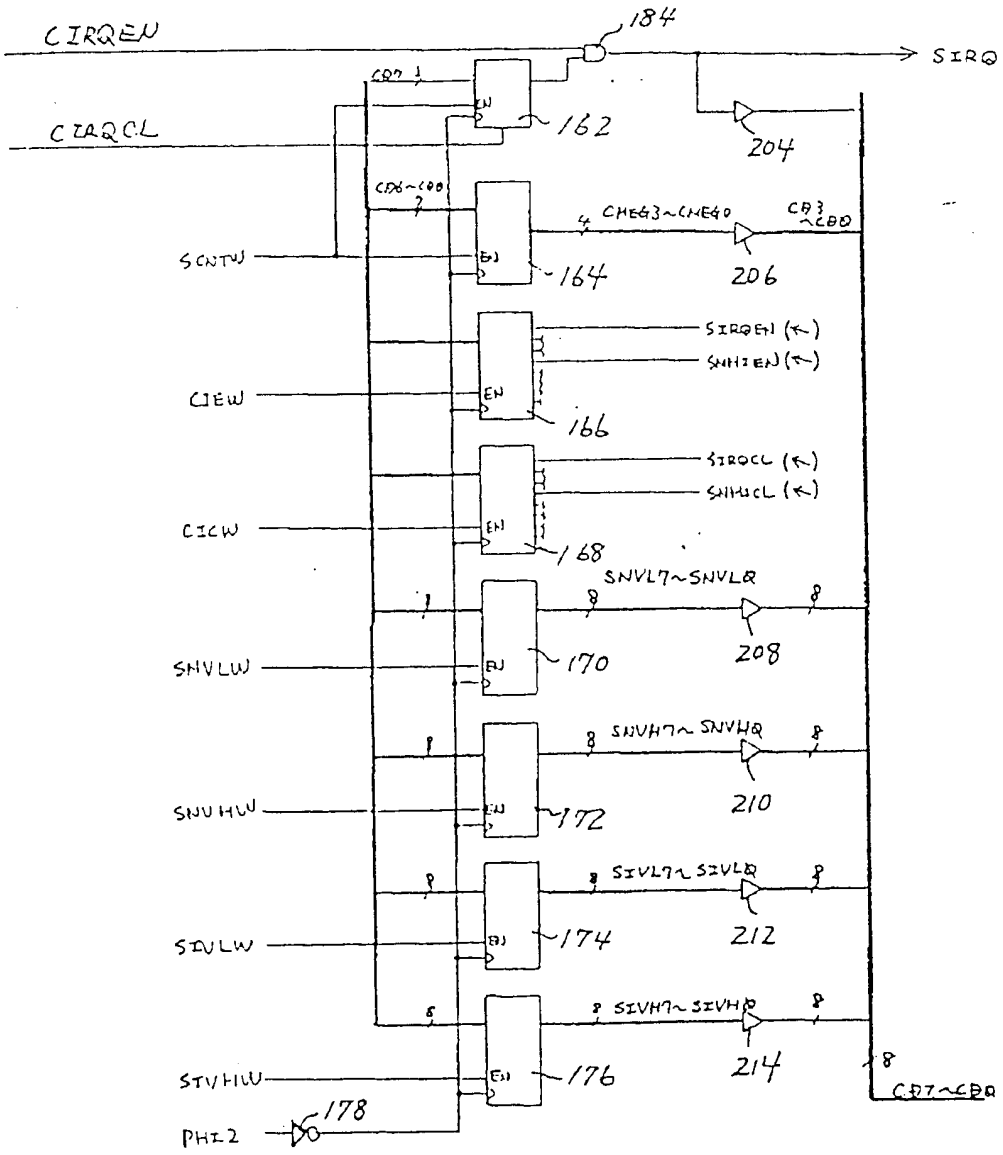

CIRQEN
CIRQCL
SCNTW
CIEW
CICW
SNVLW
SNVHW
SIVLW
SIVHW
PHI2
SIRQ
CHEG3~CHEG0
SIRQEN (←)
SNVIEN (←)
SIRQCL (←)
SNVICL (←)
SNVL7~SNVL0
SNVH7~SNVH0
SIVL7~SIVL0
SIVH7~SIVH0
CB7~CB0
162
164
166
168
170
172
174
176
178
184
204
206
208
210
212
214

图 16

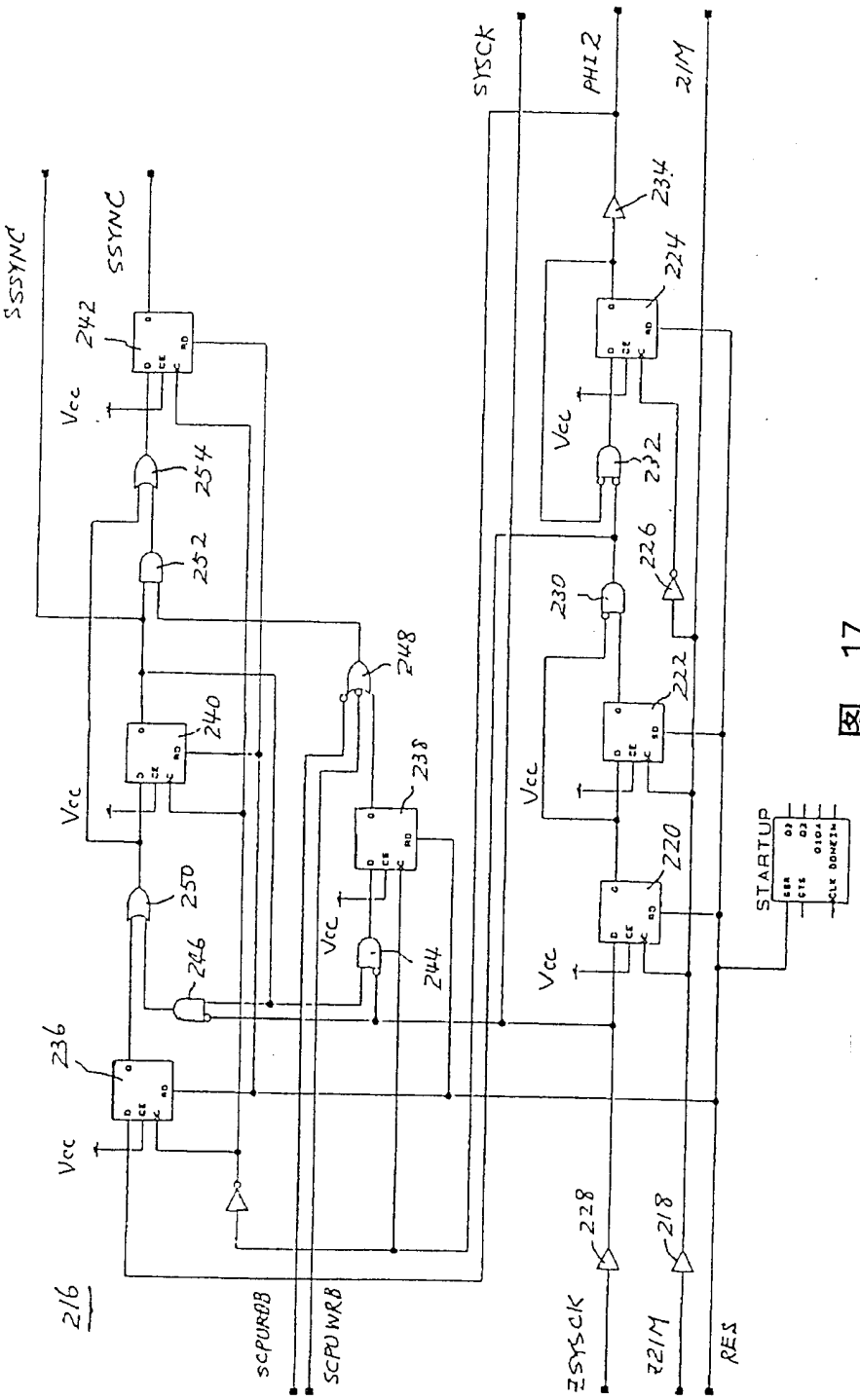

SSYNC
SSYNC
242
Vcc
254
252
240
248
238
Vcc
250
246
Vcc
244
236
Vcc
216
SCPURQB
SCPUWRB
SYSCK
PHI2
21M
234
224
Vcc
232
226
230
222
Vcc
220
Vcc
STARTUP
228
218
ZSYSCK
Z21M
RES

图 17

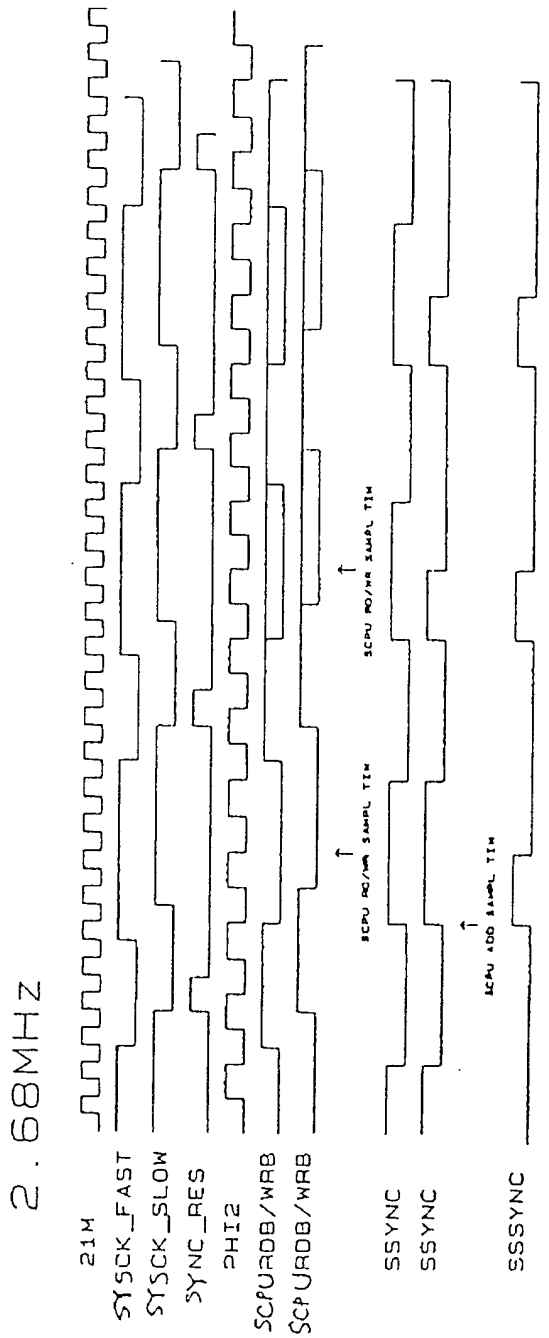

2.68MHz
21M
SYSCK_FAST
SYSCK_SLOW
SYNC_RES
PHI2
SCPURDB/WRB
SCPURDB/WRB
SSYNC
SSYNC
SSSYNC

图 18

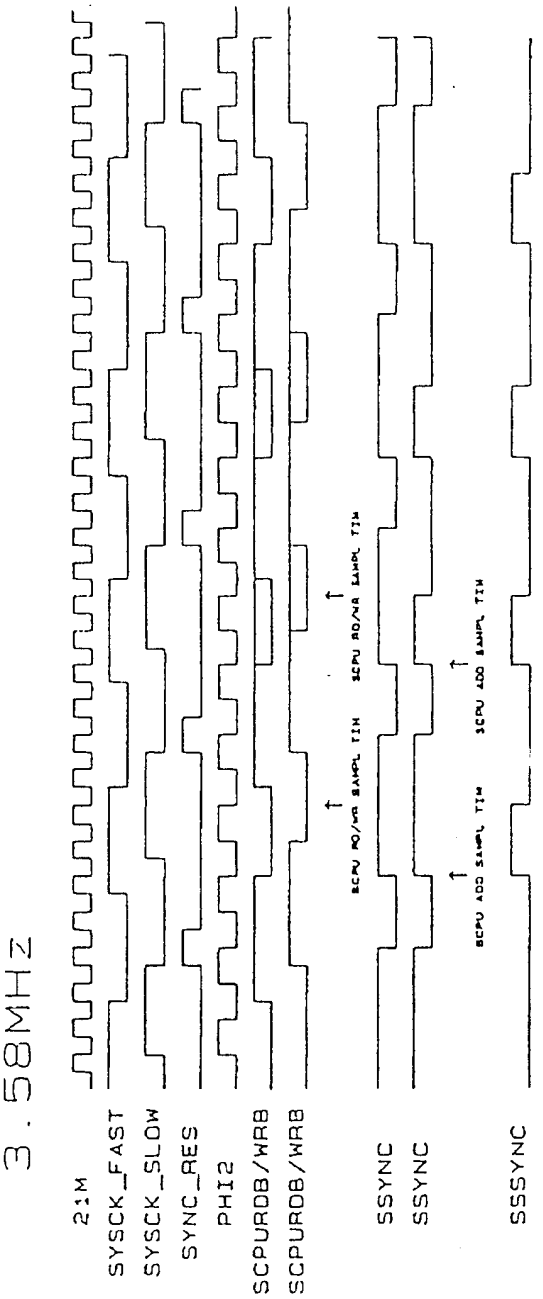

3.58MHz
21M
SYSCK_FAST
SYSCK_SLOW
SYNC_RES
PHI2
SCPURDB/WRB
SCPUROB/WRB
SSYNC
SSYNC
SSSYNC

图 19

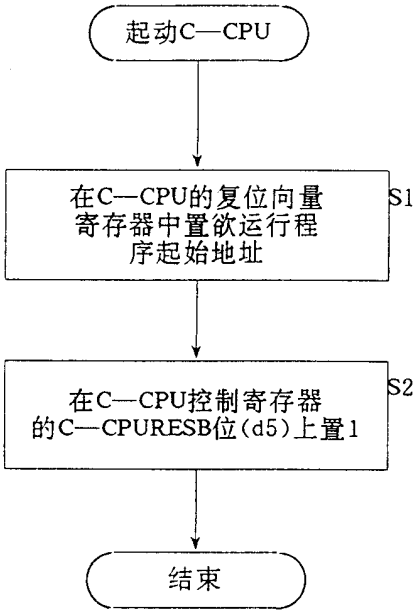

起动C—CPU
在C—CPU的复位向量寄存器中置欲运行程序起始地址
S1
在C—CPU控制寄存器的C—CPURESB位(d5)上置1
S2
结束

图 20

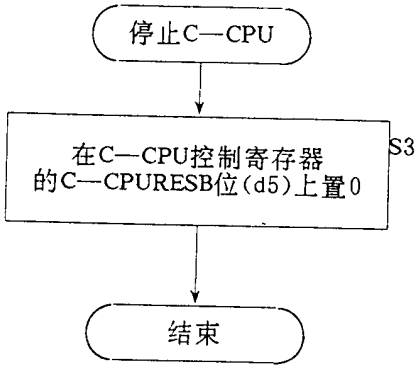


图 21

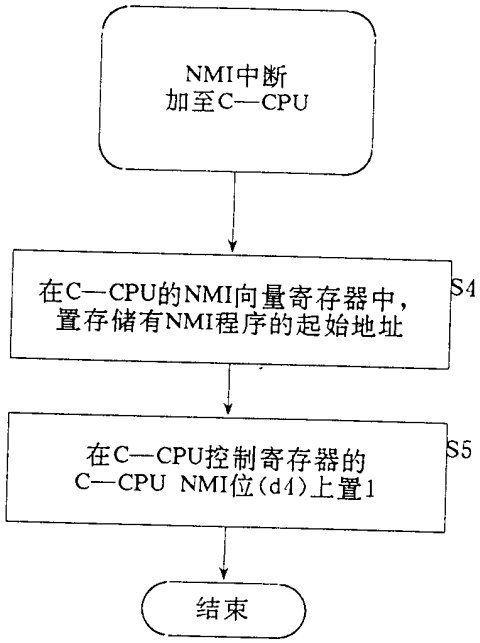

NMI中断
加至C—CPU
在C—CPU的NMI向量寄存器中，
置存储有NMI程序的起始地址
S4
在C—CPU控制寄存器的
C—CPU NMI位(d4)上置1
S5
结束

图 22

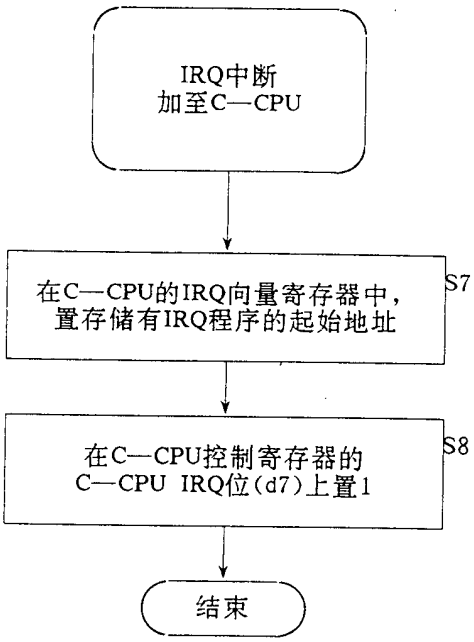

IRQ中断
加至C—CPU
在C—CPU的IRQ向量寄存器中，
置存储有IRQ程序的起始地址
S7
在C—CPU控制寄存器的
C—CPU IRQ位(d7)上置1
S8
结束

图 23

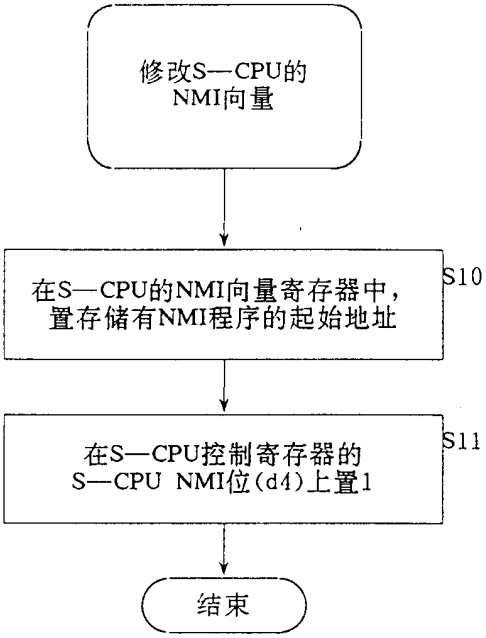

修改S—CPU的
NMI向量
在S—CPU的NMI向量寄存器中，
置存储有NMI程序的起始地址
S10
在S—CPU控制寄存器的
S—CPU NMI位(d4)上置1
S11
结束

图 24

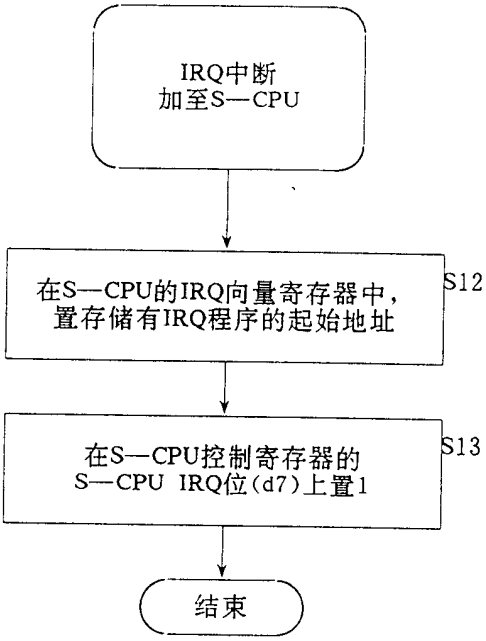

IRQ中断
加至S—CPU
在S—CPU的IRQ向量寄存器中，
置存储有IRQ程序的起始地址
S12
在S—CPU控制寄存器的
S—CPU IRQ位(d7)上置1
S13
结束

图 25

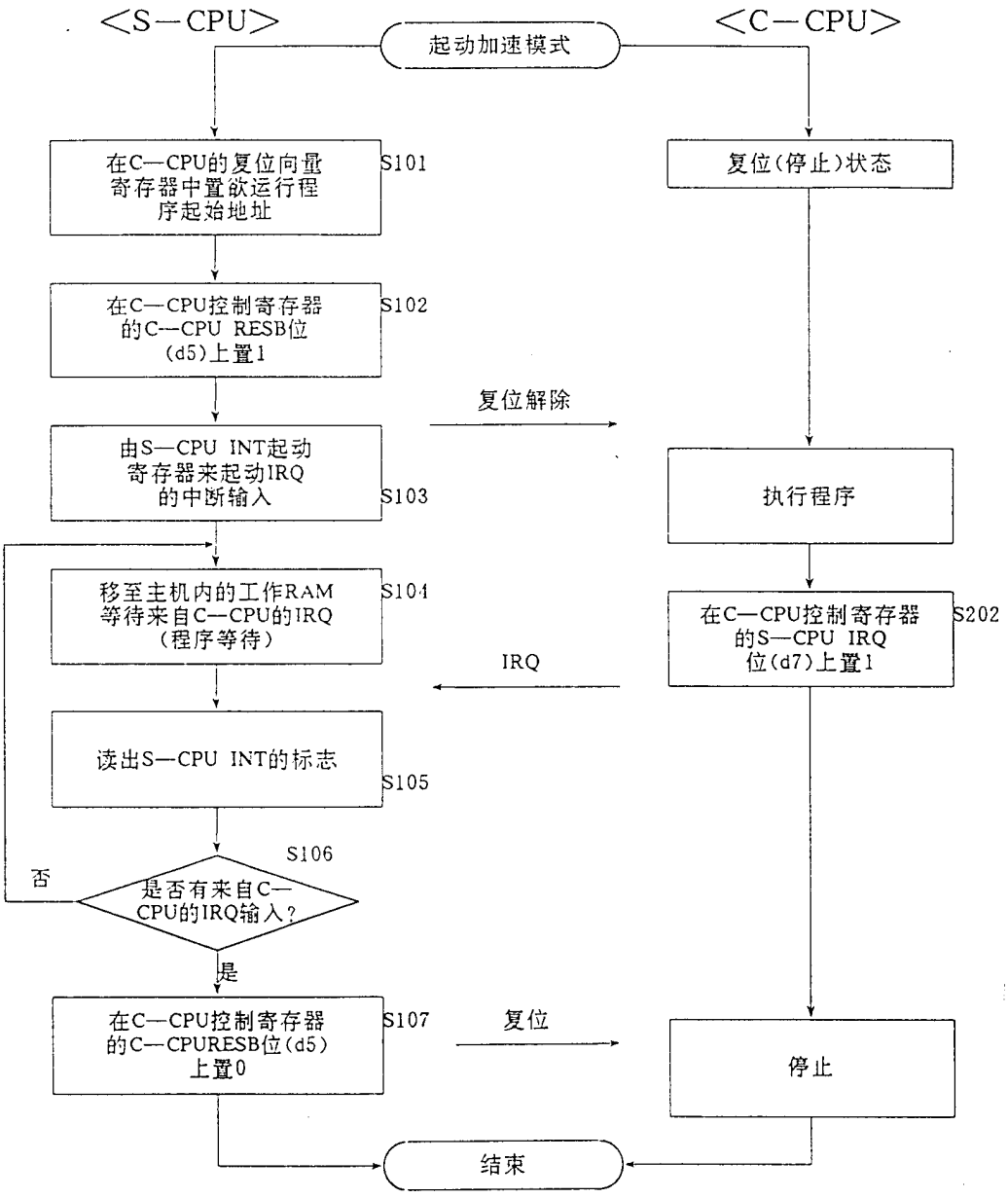

<S—CPU>
起动加速模式
<C—CPU>
在C—CPU的复位向量寄存器中置欲运行程序起始地址
S101
复位(停止)状态
在C—CPU控制寄存器的C—CPU RESB位(d5)上置1
S102
复位解除
由S—CPU INT起动寄存器来起动IRQ的中断输入
S103
执行程序
移至主机内的工作RAM等待来自C—CPU的IRQ(程序等待)
S104
在C—CPU控制寄存器的S—CPU IRQ位(d7)上置1
S202
IRQ
读出S—CPU INT的标志
S105
S106
否
是否有来自C—CPU的IRQ输入?
是
在C—CPU控制寄存器的C—CPURESB位(d5)上置0
S107
复位
停止
结束

图 26

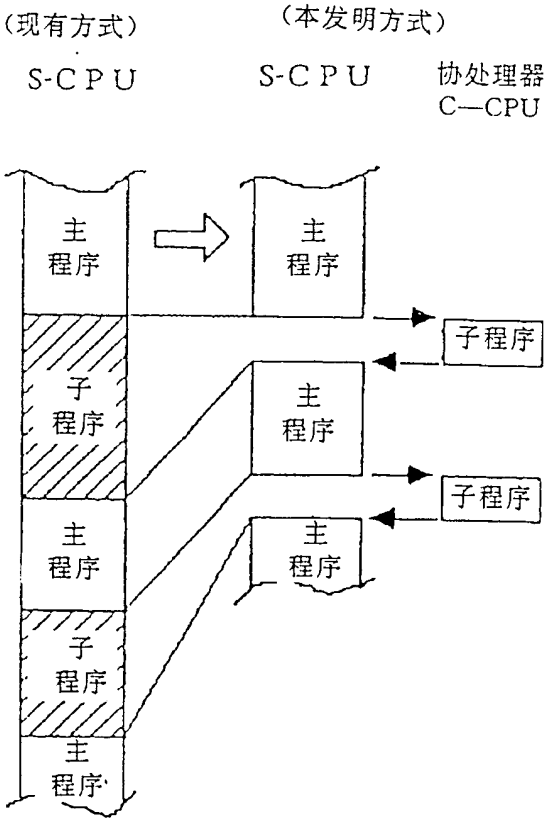

（现有方式）
（本发明方式）
S-CPU
S-CPU
协处理器
C—CPU
主
程序
子
程序
主
程序
子
程序
主
程序
子
程序
主
程序
主
程序
子程序
主
程序
子程序

图 27

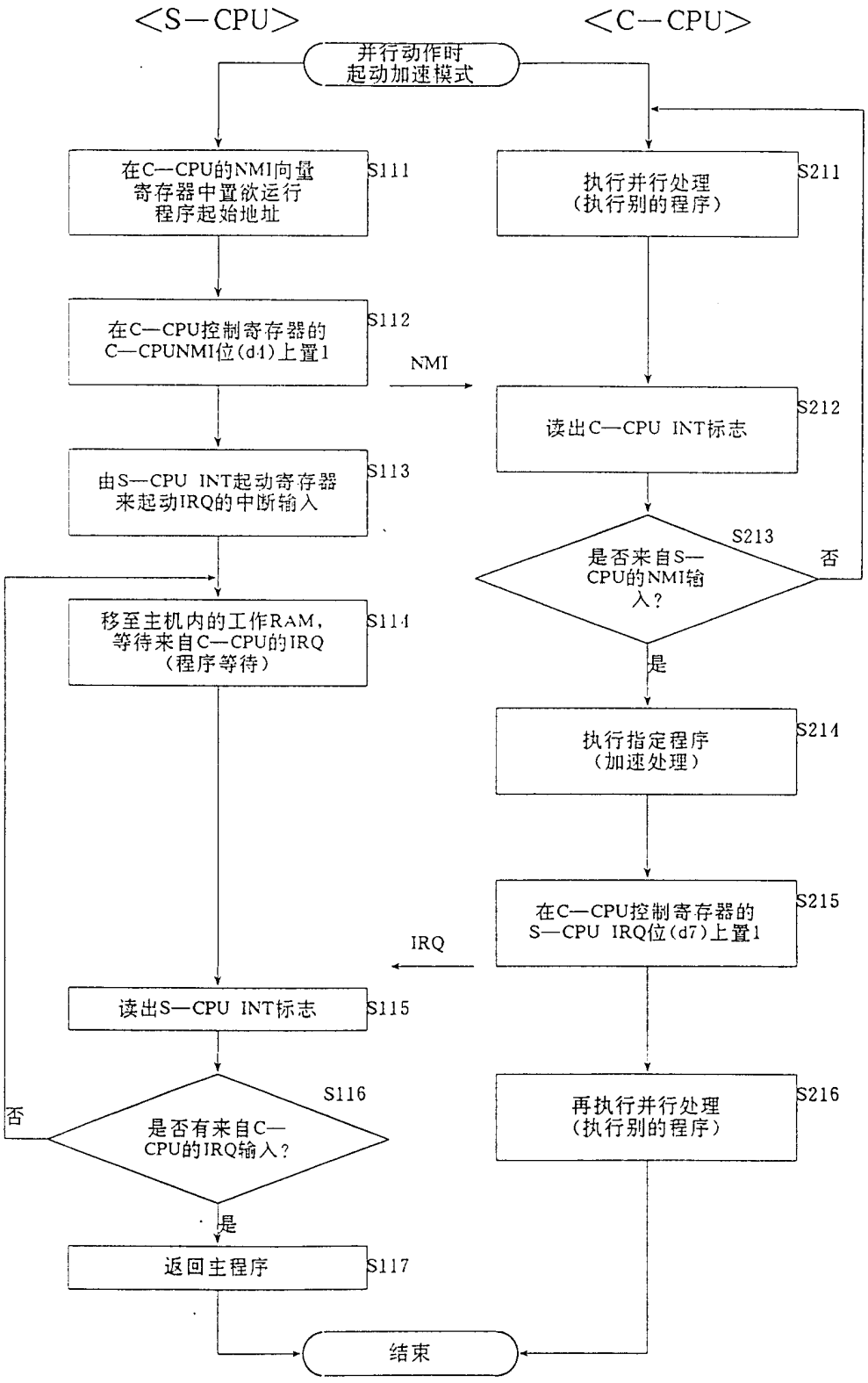

<S—CPU>
<C—CPU>
并行动作时
起动加速模式
在C—CPU的NMI向量
寄存器中置欲运行
程序起始地址
S111
在C—CPU控制寄存器的
C—CPUNMI位(d1)上置1
S112
NMI
由S—CPU INT起动寄存器
来起动IRQ的中断输入
S113
移至主机内的工作RAM，
等待来自C—CPU的IRQ
(程序等待)
S114
读出S—CPU INT标志
S115
是否有来自C—
CPU的IRQ输入？
S116
否
是
返回主程序
S117
结束
执行并行处理
(执行别的程序)
S211
读出C—CPU INT标志
S212
是否来自S—
CPU的NMI输
入？
S213
否
是
执行指定程序
(加速处理)
S214
在C—CPU控制寄存器的
S—CPU IRQ位(d7)上置1
S215
IRQ
再执行并行处理
(执行别的程序)
S216

图 28

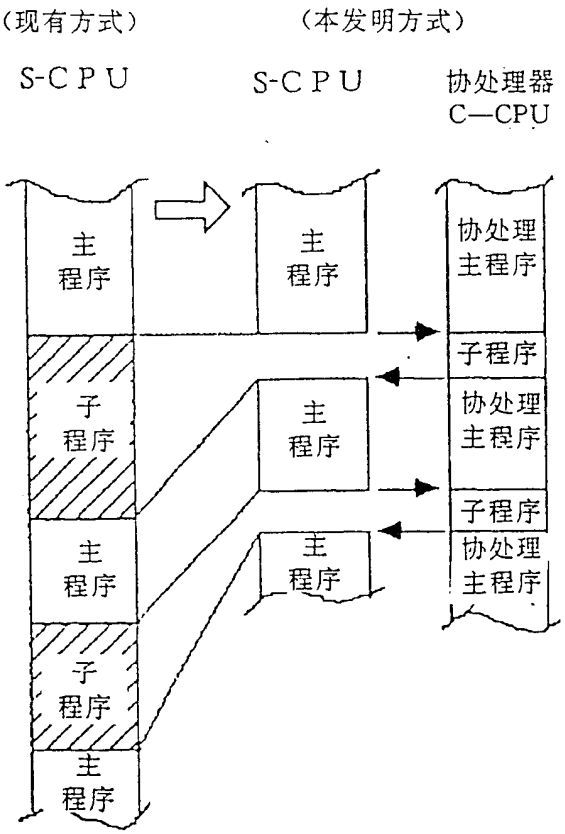

（现有方式）
（本发明方式）
S-C P U
S-C P U
协处理器
C—CPU
主
程序
子
程序
主
程序
子
程序
主
程序
主
程序
协处理
主程序
子程序
协处理
主程序
子程序
协处理
主程序

图 29

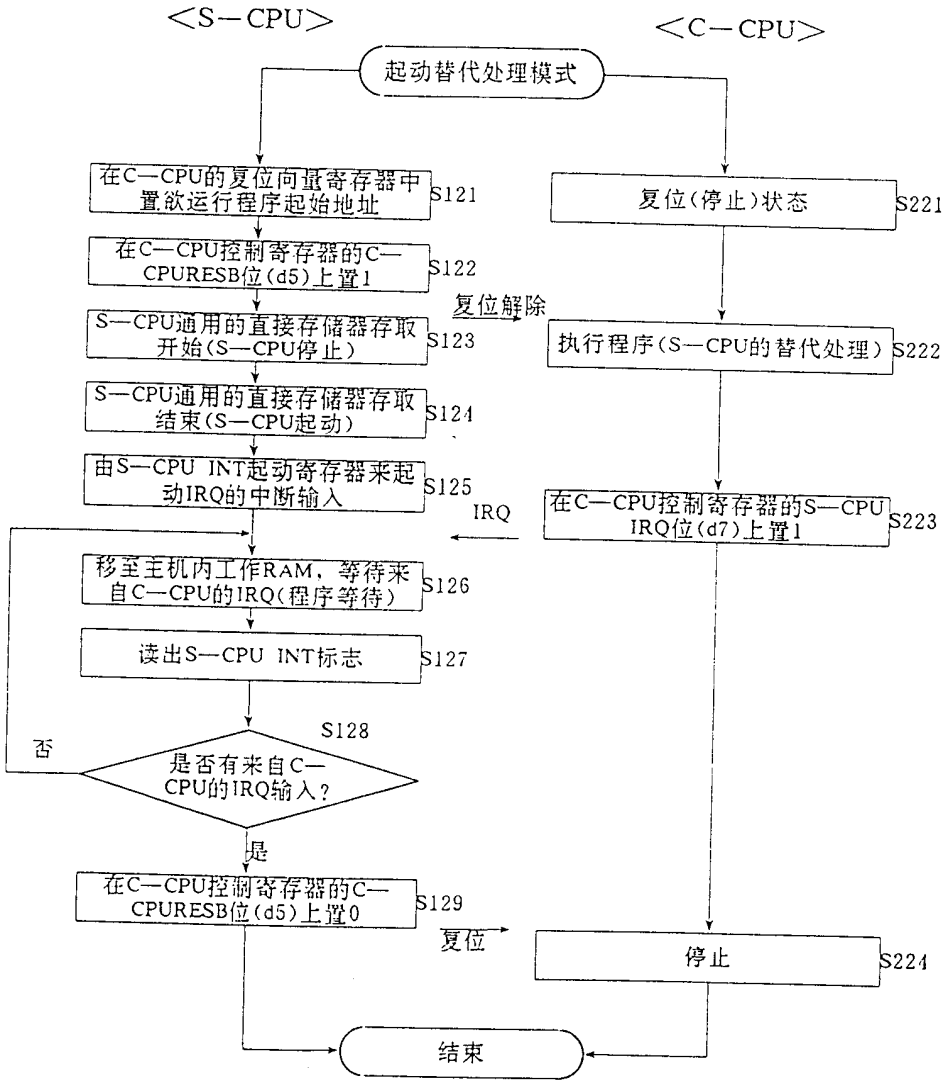

<S—CPU>
<C—CPU>
起动替代处理模式
在C—CPU的复位向量寄存器中置欲运行程序起始地址
S121
在C—CPU控制寄存器的C—CPURESB位(d5)上置1
S122
复位解除
S—CPU通用的直接存储器存取开始(S—CPU停止)
S123
S—CPU通用的直接存储器存取结束(S—CPU起动)
S124
由S—CPU INT起动寄存器来起动IRQ的中断输入
S125
IRQ
移至主机内工作RAM，等待来自C—CPU的IRQ(程序等待)
S126
读出S—CPU INT标志
S127
S128
否
是否有来自C—CPU的IRQ输入？
是
在C—CPU控制寄存器的C—CPURESB位(d5)上置0
S129
复位
结束
复位(停止)状态
S221
执行程序(S—CPU的替代处理)
S222
在C—CPU控制寄存器的S—CPU IRQ位(d7)上置1
S223
停止
S224

图 30

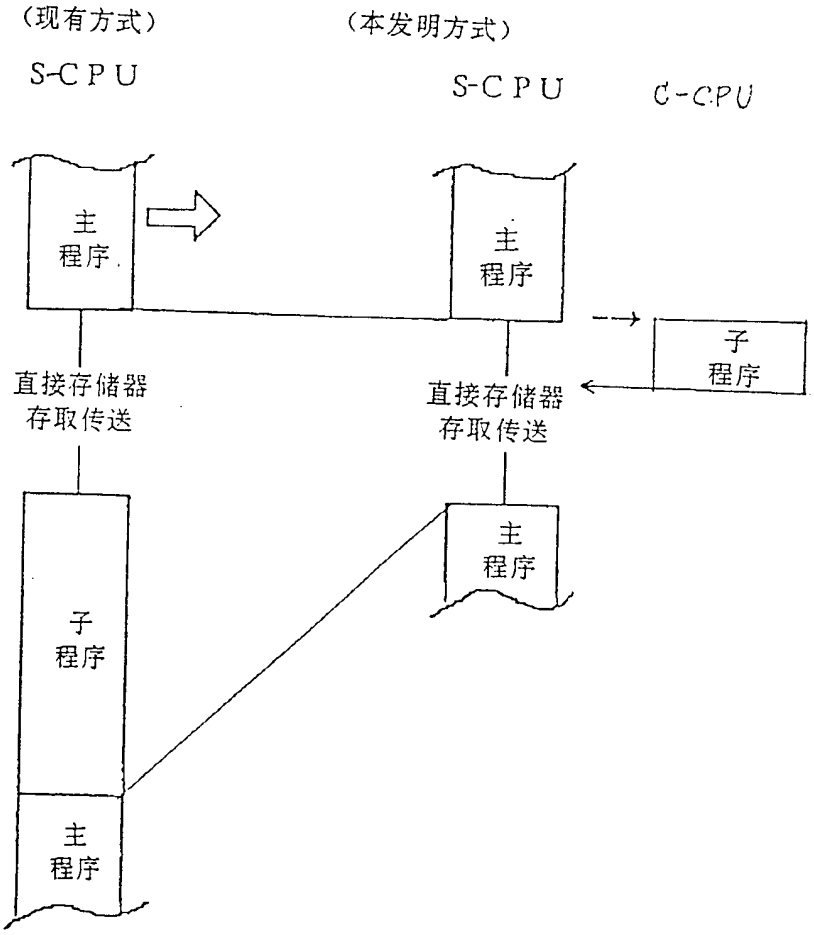

（现有方式）
（本发明方式）
S-CPU
S-CPU
C-CPU
主
程序
主
程序
子
程序
直接存储器
存取传送
直接存储器
存取传送
子
程序
主
程序
主
程序

图 31