

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 9/00 (2006.01)



[12] 发明专利说明书

专利号 ZL 02816970.0

[45] 授权公告日 2009 年 7 月 22 日

[11] 授权公告号 CN 100517215C

[22] 申请日 2002.8.29 [21] 申请号 02816970.0

[30] 优先权

[32] 2001.8.29 [33] US [31] 60/315,655

[86] 国际申请 PCT/US2002/027695 2002.8.29

[87] 国际公布 WO2003/021426 英 2003.3.13

[85] 进入国家阶段日期 2004.2.27

[73] 专利权人 联发科技股份有限公司

地址 台湾省新竹科学工业园区

[72] 发明人 波尔·R·延森 莫滕·尼尔森

莫恩斯·克里斯蒂安森

[56] 参考文献

US5592173A 1997.1.7

US5560024A 1996.9.24

EP0939495A1 1999.9.1

US6255882B1 2001.7.3

审查员 马雅凡

[74] 专利代理机构 北京三友知识产权代理有限公司

代理人 任默闻

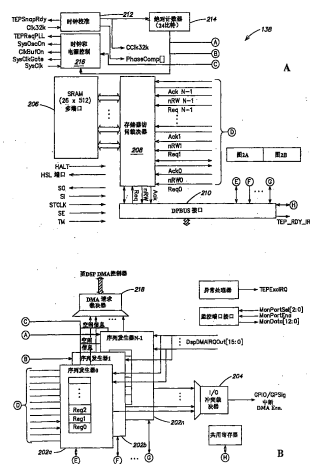
权利要求书 3 页 说明书 26 页 附图 23 页

[54] 发明名称

用于无线系统中定时及事件处理的方法和装置

[57] 摘要

一种用于与不同的无线系统同时操作的数字基带处理器。数字基带处理器包括至少一个主处理器，用于执行第一指令序列中的指令；和定时及事件处理器，由数字信号处理器和微控制器控制，用于执行定时敏感性指令。定时及事件处理器包括：两个或多个指令序列发生器，用于执行定时敏感性指令线程；时基发生器，用于生成定时信号，该定时信号启动多个指令序列发生器中每一个中指令线程的执行，该时基发生器包括时钟校准电路；存储器，用于保持用于该两个或多个指令序列发生器的指令和数据；绝对计数器，用于计数校准后的低频时钟，并生成可编程的定时信号；I/O 冲突裁决器，解决由该两个或多个指令序列发生器产生的输出中的冲突，以及响应冲突而产生异常。



1. 一种数字基带处理器，包括：
至少一个主处理器，用于执行第一指令序列中的指令；和
定时及事件处理器，连接至所述主处理器，用于执行第二指令序列中的定时敏感性指令，
所述定时及事件处理器包括：
两个或多个指令序列发生器，用于执行第二指令序列的线程；
存储器，用于保持用于所述两个或多个指令序列发生器的指令和数据；
时基发生器，用于产生定时信号，该定时信号用于使两个或多个指令序列发生器中每一个中的指令开始执行，其中，所述时基发生器包括时钟校准电路，用于根据相对稳定的高频时钟来校准相对不稳定的低频时钟，并生成校准后的低频时钟；和
绝对计数器，用于计数校准后的低频时钟，并生成可编程的所述定时信号；
I/O 冲突裁决器，用于解决由所述两个或多个指令序列发生器产生的输出中的冲突，以及响应冲突而产生异常。
2. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于启动数字基带处理器的元件之间的时间精确的 DMA 传送的装置。
3. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于能够时间精确地启用数字基带处理器中 DMA 通道的装置。
4. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于产生时间精确的至所述主处理器的中断的装置。

5. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于产生对数字基带处理器其他元件的时间精确的触发的装置。

6. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于产生时间精确的输出信号的装置。

7. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括用于根据高频时钟校准低频时钟，并生成相位补偿信号的装置。

8. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括响应时基发生器的电源控制电路，用于禁止至数字基带处理器中空闲状态的模块的时钟。

9. 如权利要求 1 所述的数字基带处理器，其中所述存储器是多端口的，其中所述定时及事件处理器还包括存储器访问裁决器，用于控制所述两个或多个指令序列发生器对存储器的访问。

10. 如权利要求 1 所述的数字基带处理器，还包括 DMA 控制器，用于处理 DMA 请求，其中所述定时及事件处理器包括 DMA 接口，用于启动时间精确的 DMA 传送。

11. 如权利要求 10 所述的数字基带处理器，其中所述定时及事件处理器还包括 DMA 请求裁决器，用于解决由所述两个或多个指令序列发生器产生的 DMA 请求中的冲突。

12. 如权利要求 1 所述的数字基带处理器，其中所述定时及事件处理器还包括总线接口，用于将所述两个或多个指令序列发生器连接

至处理总线。

13. 一种数字基带处理器，用于与不同无线系统同时操作，包括：
数字信号处理器，用于执行数字信号处理器指令；
微控制器，用于执行微控制器指令；和
定时及事件处理器，由所述数字信号处理器和所述微处理器控制，
用于执行定时敏感性指令，
所述定时及事件处理器包括：
多个指令序列发生器，用于执行定时敏感性指令线程；
存储器，用于保持用于所述多个指令序列发生器的指令和数据；
时基发生器，用于产生定时信号，该定时信号用于使多个指令序列发生器中每一个中的指令开始执行，其中，所述时基发生器包括时钟校准电路，用于根据相对稳定的高频时钟来校准相对不稳定的低频时钟，并生成校准后的低频时钟；和
绝对计数器，用于计数校准后的低频时钟，并生成可编程的所述定时信号；
I/O 冲突裁决器，用于解决由所述两个或多个指令序列发生器产生的输出中的冲突，以及响应冲突而产生异常。

用于无线系统中定时及事件处理的方法和装置

相关申请

本申请以 2001 年 8 月 29 日提交的申请号为 60/315,655 的临时申请为在先申请，并将其全部合并入本发明作为参考。

技术领域

本发明涉及无线通信，特别是无线系统中的定时及事件处理。

背景技术

随着无线通信网络的迅速发展，经常建立新的无线通信标准来代替旧的废弃的标准。然而，经常要花费时间在广泛的物理领域实现基于新标准的新无线网络。因此，人们常希望有既能和新无线网络通信又能和现存无线网络通信的无线终端。此外，随着无线计算机数据网络的迅速发展，人们常希望有能够和这些网络通信的无线终端，以允许用户浏览因特网或者收发电子邮件。此外，和不同的无线系统同时通信是有用的，这样用户可以例如在无线数据网络上查看电子邮件，同时在 2G 无线网络上进行语音电话呼叫。

这样的无线系统常使用不同的时基。例如 2G GSM 网络使用帧具有 4.615 毫秒的持续时间，并被分割为 8 时隙的时基。而 3G WCDMA 网络使用帧具有 10 毫秒的持续时间，并被分割为 15 时隙的时基。无论移动终端是与一个无线系统一起操作还是同时与两个或更多无线系统一起操作，移动终端中的事件必须精确定时并相对于每一个无线系统同步。

此外，为了维持便携性，无线终端一般由电池供电，其中在再充电之间的时间是最大电流（current drawn）的反函数。因为期望允许用户在再充电之间尽可能长时间操作无线终端，所以电源管理是重要的考虑因素。

发明内容

根据本发明的第一方面，提供一种数字基带处理器。该数字基带处理器包括至少一个主处理器，用于执行第一指令序列中的指令；和定时及事件处理器，连接至主处理器，用于执行第二指令序列中的定时敏感性指令。定时及事件处理器包括：两个或多个指令序列发生器，用于执行第二指令序列的线程；存储器，用于保持用于所述两个或多个指令序列发生器的指令和数据；时基发生器，用于产生定时信号，该定时信号用于使两个或多个指令序列发生器中每一个中的指令开始执行，其中，所述时基发生器包括时钟校准电路，用于根据相对稳定的高频时钟来校准相对不稳定的低频时钟，并生成校准后的低频时钟；绝对计数器，用于计数校准后的低频时钟，并生成可编程的所述定时信号；I/O 冲突裁决器，用于解决由所述两个或多个指令序列发生器产生的输出中的冲突，以及响应冲突而产生异常。

根据本发明的又一方面，提供一种数字基带处理器，用于与不同无线系统同时操作。该数字基带处理器包括数字信号处理器，用于执行数字信号处理器指令；微控制器，用于执行微控制器指令；和定时及事件处理器，由数字信号处理器和微处理器控制，用于执行定时敏感性指令。定时及事件处理器包括多个指令序列发生器，用于执行定时敏感性指令线程；存储器，用于保持用于所述多个指令序列发生器的指令和数据；时基发生器，用于产生定时信号，该定时信号用于使两个或多个指令序列发生器中每一个中的指令开始执行，其中，所述时基发生器包括时钟校准电路，用于根据相对稳定的高频时钟来校准相对不稳定的低频时钟，并生成校准后的低频时钟；绝对计数器，用于计数校准后的低频时钟，并生成可编程的所述定时信号；I/O 冲突裁决器，用于解决由所述两个或多个指令序列发生器产生的输出中的冲突，以及响应冲突而产生异常。

根据本发明的又一方面，提供一种用于产生定时信号的方法，该定时信号用于操作具有无线系统时基的无线系统中的无线终端。该方法包括产生校准后的慢时钟；通过计数校准后的慢时钟而产生绝对时间值，以提供统一的时基；和定时无线系统中的事件，这基于独立于

无线系统时基的统一时基的绝对时间值。

根据本发明的又一方面，提供一种用于产生校准后的时钟的方法。该方法包括接收自由运行的快时钟；接收自由运行的慢时钟；修正自由运行慢时钟以提供校准后的慢时钟，该慢时钟与快时钟具有规定的关系；和提供相位补偿信号，该信号代表校准后的慢时钟中的相位误差。

根据本发明的又一方面，提供一种用于产生校准后的时钟的方法。该方法包括接收自由运行的快时钟；接收自由运行的慢时钟；规定快时钟和校准后的慢时钟之间的关系；计数所选择数量的自由运行慢时钟周期中的快时钟周期的数量，以提供比较值；基于快时钟和校准后的慢时钟之间规定的关系和比较值，从自由运行慢时钟中除去周期，以提供校准后的慢时钟；和提供相位补偿信号，该信号代表校准后的慢时钟中的相位误差。

根据本发明的又一方面，提供一种用于在基带处理器中执行 DMA 传送的方法。该方法包括在数字信号处理器内核中执行计算；在定时及事件处理器中生成定时信号；响应来自数字信号处理器的请求以及响应来自定时及事件处理器的定时信号，执行 DMA 传送，以提供定时的 DMA 传送。

根据本发明的又一方面，提供一种用于无线应用的基带处理器。该基带处理器包括数字信号处理器内核，用于执行数字信号计算；定时及事件处理器，连接至所述数字信号处理器内核，用于执行定时敏感性操作，所述定时及事件处理器包括时基发生器，用于产生定时信号，和 DMA 控制电路，用于启动 DMA 请求，以响应来自数字信号处理器内核的指令和来自时基发生器的定时信号；和 DMA 控制器，用于执行 DMA 请求，以提供定时的 DMA 传送。

附图说明

图 1 是根据本发明一实施例的通信处理器的框图；

图 2 是根据本发明一实施例的图 1 通信处理器中的定时及事件处理器的框图；

图 3 是根据本发明一实施例的图 2 定时及事件处理器中的序列发生器的框图；

图 4 是根据本发明一实施例的适用于序列发生器的指令格式的例子。

图 5 是用于说明根据本发明一实施例的定时及事件处理器的功能的图。

图 6 是根据本发明一实施例的直接存储器存取控制器与定时及事件处理器之间接口的框图；

图 7 是用于说明根据本发明一实施例的利用定时及事件处理器进行直接存储器存取的方法的例子框图。

图 8 是根据本发明一实施例的与定时及事件处理器连接的外部总线的框图；

图 9A 是根据本发明一实施例的 N 分时器的例子框图；

图 9B 是表示根据本发明一实施例的由 N 分时器生成的校准后的时钟和相位补偿的示意图；

图 10 是根据本发明一实施例的绝对计数器和可配置的周期触发脉冲发生器的框图；

图 11A 是根据本发明一实施例的快照有限态机器的框图；

图 11B 是根据本发明一实施例的图 11A 中快照有限态机器的操作的流程图；

图 12 是根据本发明一实施例的图 2 中定时及事件处理器中时钟和电源控制模块的框图；

图 13A 是根据本发明一实施例的时钟产生模块的框图；

图 13B 是根据本发明一实施例的时钟分配模块的示意图；

图 14A 是根据本发明一实施例的时钟门控模块的示意图；

图 14B 是用于说明根据本发明一实施例的图 14A 中所示的寄存器的内容的表；

图 15 是用于说明根据本发明一实施例的振荡器断电需求的框图。

具体实施方式

无线终端可以包括无线电单元、数字基带处理器、用户接口、和

电池。基带处理器可以包括用于执行数字处理算法和其他复杂计算的数字信号处理器，和用于执行控制功能和相对简单计算的微控制器。由无线终端中的基带处理器执行的许多任务都需要精确定时。例如，在无线通信网络中，预定无线信道中的动作以规定的精度在指定的时间发生。专用的定时及事件处理器（TEP）可以用来达到这样的定时精度。例如，TEP 可以负责产生定时信号、调度事件、产生至处理器的中断、启动其他模块中的操作、为芯片外电路产生控制信号，例如无线电单元。TEP 可以与数字信号处理器、微控制器和基带控制器的其他元件协同工作，来控制无线终端中的所有定时和事件。

有时会期望同时与若干不同的无线系统进行通信。例如，通信处理器与诸如蓝牙网络的无线数据网络进行通信，以监控用户的新电子邮件，同时监控无线 CDMA 网络的寻呼信道，以发现新的语音电话呼叫。通常，与通信处理器通信的不同的无线系统利用不同的时基。TEP 可以利用共同的基准时钟作为时基来为无线系统安排事件，从而为与通信处理器通信的任何无线系统安排事件。

图 1 表示根据本发明一个实施例的基带通信处理器 100 的框图。图 1 所示的处理器包括两个处理内核。数字信号处理器（DSP）内核 102 可以用来执行通信处理器 100 的数字信号处理功能，比如与单元搜索、信号的相关、信道编码和解码相关的处理。许多其他的信号处理功能可以由 DSP 内核 102 执行。适用于本发明实施例的 DSP 内核的一例在 2000 年 11 月 16 日公开的 PCT 公开号为 WO 00/687783 中公开。但是，应该理解可以使用许多其他类型的数字信号处理器，本发明不限于任何特定的数字信号处理器。微控制器单元（MCU）104 处理内核可以用来执行用于通信处理器 100 的控制码，比如协议堆栈指令的执行。市面上可获得的适用于本发明的 MCU 的一例是 ARM7TDMI 内核，由 Advanced RISC Machine, Ltd. 出售。但是，应该理解可以使用许多其他类型的微控制器，本发明不限于任何特定的微控制器。

通信处理器 100 也包括系统存储器 106。系统存储器 106 可以是静态随机存储器 (SRAM)、或任何其他类型的易失性存储器或非易失性存储器, 比如动态随机存储器 (DRAM)、同步动态随机存储器 (SDRAM) 或铁电随机存储器 (FRAM)。DSP 内核 102 和 MCU104 利用共同的存储映像表。因此, 这些处理器可以共享对系统存储器 106 的访问并可以通过系统存储器 106 互相通信。

图 1 所示的每一个元件可以由单一集成电路和多个集成电路实现。在一些实施例中, 将整个通信处理器 100 制造在单一芯片上。应该理解本发明不限于这方面。

直接存储器存取 (DMA) 控制器 134 和 136 用来促进通信处理器 100 中的数据传送。DMA 控制器 134 和 136 允许在设备与存储器 (例如, 系统存储器 106) 之间直接传送存储内容而不受处理器干涉。可以为设备分配 DMA 通道以允许这些设备请求 DMA 传送。通道配置由 DSP 内核 102 与 MCU 104 确定。尽管两个处理器都可以访问每一个 DMA 通道, 但是可以由 DSP 内核 102 控制一个通道组的通道配置, 由 MCU 内核 104 控制另一个通道组的通道配置。同样, DMA 控制器 134 可以控制由 DSP 内核 102 配置的通道的 DMA 传送, 而 DMA 控制器 136 可以控制由 MCU 104 配置的通道的 DMA 传送。

DSP 内核 102 可以包括级别 1 (L1) 指令高速缓存 144 和 L1 数据高速缓存 146, 用来当访问存储数据时提供较低的等待时间。DSP 内核 102 可以具有两条连接至 L1 数据高速缓存 146 的数据总线、一条连接至 L1 指令高速缓存 144 的指令总线、和一条连接至高速缓存 144 和 146 的 DMA 总线。级别 2 (L2) 存储器 148 可以是由 DSP 内核 102 使用的专用的 SRAM。存储器 148 可以被 DMA 控制器 134 访问。存储器 148 也可以被 MCU 104、DMA 控制器 136、和外部应用处理器接口 (EAPI) 142 访问。

系统总线接口单元 (SBIU) 132 执行总线桥接功能。例如, SBIU 132 可以作为非对称交叉开关, 用来将来自 DSP 内核 102、DMA 控制器 134、MCU 104、DMA 控制器 136、和外部应用处理器接口 142 的请求分配到适当的系统资源, 比如 L1 高速缓存 144、L1 高速缓存 146、L2 存储器 148、和其他系统资源。SBIU 132 允许各种总线之间并行和同时的数据传送。

人机接口 (MMI) 模块 150 提供通信处理器 100 的硬件用户接口并可以通过 PBUS 总线 128 访问。MMI 150 模块可以包括通信处理器 100 的通用输入输出 (GPIO) 引脚的接口。这些引脚可以用作各种用途, 包括接口无线电单元和其他外部设备。其他 MMI 模块可以包括显示屏接口、串行端口接口、通用异步收发器接口 (UART)、USB 接口、和包含嵌入了通信处理器 100 的无线终端的唯一序列号的用户标识模块 (SIM)。许多其他接口模块可以包括在 MMI 150 中。

内务处理模块执行各种用于通信处理器 152 的内务处理功能, 并可以通过 PBUS 总线 128 访问。这种功能包括当通信处理器中发生软件死锁时超时并产生复位的监视时钟 (WDT), 用于触发通用定时功能的通用时钟, 和用于管理至 DSP 内核 102 和 MCU 104 的中断的 IRQ 控制器。

无线系统模块 154 提供外部无线系统元件与通信处理器 100 的接口并可以通过 PBUS 总线 128 访问。例如, 无线系统模块 154 可以包括作为至模拟基带芯片的控制串行端口的 CS 端口, 和至频率合成器的接口。

DSP 外围设备与 DSP 内核 102 一起执行各种数字信号处理功能并可以通过 DPBUS 总线 110 访问。DSP 外围设备可以包括, 例如协处理器接口 162、BS 端口 164、旗标 I/O 166、高速记录器 168、加

密引擎 170、和 DSP IRQ 控制器 172。

利用一条或多条总线，数据可以在通信处理器的各种元件之间和通信处理器与芯片外设备之间传送。每条总线都可以是并行或串行总线。每条总线都可以是不定向的或定向的。此外，每条总线都可以包括地址总线、数据总线和控制总线中的任何一个。图 1 所示的通信处理器 100 的总线配置包括多个总线系统。下面大概介绍每一个总线系统的功能。对本领域技术人员而言，可以在本发明的主旨和范围内对图 1 所示的总线配置进行许多变更、修改、和改进。

SYSL2 总线 108 连接在 SBIU 132 与 L2 存储器 148 接口之间。存储器 144 在 MCU 104、系统 DMA 控制器 136、DSP DMA 控制器 134、和 DSP 102 之间共享。DPBUS 总线 110 是 DSP 外围设备总线和各种 DSP 外围设备的接口，外围设备比如可以是基带串行端口的 BS 端口 164、协处理器接口 162、旗标 I/O 166、高速记录器 168、加密引擎 170、和 DSP IRQ 控制器 172。MCU 104、系统 DMA 控制器 136、和 DSP DMA 控制器 134 之间共享对 DPMBUS 总线 110 的访问。DSP 内核 102 也可以通过 SBIU 132 访问 DPMBUS。DPMBUS 总线 112 是 DSP 内核 102 至 PBUS 总线 128、系统存储器 106、和 EBUS 总线 100 的接口。DABUS 总线 114 作为 DSP DMA 控制器至 SBIU 132 的接口。DMABUS 总线 116 是系统 DMA 控制器 136 与 PBUS 总线 128、RBUS 总线 118、和 EBUS 总线 120 上的资源之间的接口。RBUS 总线 118 是至系统存储器 106 的接口。MCU 104、系统 DMA 控制器 136、DSP DMA 控制器 134、和 DSP 内核 102 之间共享对 RBUS 总线 118 的访问。EBUS 总线 120 作为至位于通信处理器 100 外部的闪速存储器和 SRAM 的接口。SBUS 总线 122 是 MCU 104 的主系统总线。EAPI 总线 124 作为从通信处理器 100 外部的应用处理器至通信处理器 100 的资源的接口。EABUS 总线 140 是 EAPI 142 与通信处理器 100 外部的应用处理器之间的接口。应该理解并不是必须提供外部应用处理器。CBUS 总线 126 是至外部协处理器的接口。PBUS 总线 128 是外围设备总线，

将无线系统外围设备 154、内务外围设备 152、和 MMI 外围设备 150 连接至 MCU 104、系统 DMA 控制器 136、DSP DMA 控制器 134、和 DSP 内核 102。

因为对某些总线的访问被多个元件共享，比如 PBUS 总线 128 和 RBUS 总线 118，所以提供总线仲裁器 130a、130b 和 130c 来管理对这些总线的访问。

通信处理器 100 包括定时及事件处理器 (TEP) 138，用来为通信处理器 100 安排事件。这些事件可以包括，例如，I/O 引脚的置位和清除、产生至 DSP 内核 102 和 MCU 104 的中断、初始化 TEP 138 和通信处理器 100 其他模块之间的 DMA 存储内容传送。TEP 138 经由 DPBUS 总线 110 连接至通信处理器 100 的其他模块，也连接至 DSP DMA 控制器 134 和 DSP IRQ 控制器 172。

在 TEP 138 中，不同的无线系统时基被转换为不特定于任何无线系统的统一的时基。利用统一的时基，事件被安排为在绝对时间点触发。TEP 138 产生校准后的慢时钟作为统一时基的基准，通过使用高精度的自由运行快时钟作为校准基准来获得校准后的慢时钟的长期稳定性。通过从自由运行慢时钟中除去时钟脉冲来产生被用作统一时基的时钟的校准后的慢时钟。这会产生相位误差，为了获得精确的定时信号，该误差被补偿。为自由运行慢时钟的每个时钟周期计算相位补偿。相位补偿表示为自由运行快时钟的时钟周期的数量，并与校准后的慢时钟一起使用，来提供精确定时。一个特征是即使切断自由运行快时钟，相位补偿值仍能保持。这些特征在下文详细论述。

图 5 是说明 TEP 138 功能的例子的示意图。TEP 138 可以置位和清除 GPIO 引脚以控制与外部设备的接口。TEP 138 也可以与系统 DMA 控制器 136 和 DSP DMA 控制器 134 通信来启用 DMA 通道。利用专用的 DMA 通道 516，TEP 138 也可以读和写任何存储映像位置，允许

TEP 138 与诸如无线系统 154 的其他模块通信，例如，使用频率合成器接口 154a 来对频率合成器编程。TEP 138 可以与 DSP 和 MCU IRQ 控制器 506 接口来生成每一个处理内核的中断，通过接收来自 TEP 138 的中断，允许处理内核在不工作时进入空闲状态，在必要时退出空闲状态。可以如下所述对 TEP 的所有功能进行精确定时和安排。

图 2 是根据本发明一个实施例的 TEP 结构 138 的例的框图。TEP 138 可以作为通信处理器 100 的定时及计划机构。在无线系统的操作中，所有无线电控制事件在预定的事件发生并需要精确定时。在通信处理器 100 动作之间的某些时间，特别是无线通信应用中，MCU 104 和 DSP 内核 102 都不需要执行任何处理功能，可以进入空闲方式或“休眠”方式。在这种方式中，不需要为处理内核计时，因此允许振荡器断电。通信处理器 100 可以嵌入无线终端，由电池供电。通过在不必要时使处理器空闲或使振荡器断电而节约电能可以延长电池需要再次充电前的时间。然而，在处理内核空闲之前，它们可以指示给 TEP 138 它们需要再次启动的时间。

TEP 138 可以包括多个序列发生器 202a-202n，用来执行被 TEP 138 利用的指令以执行时间特定的动作。TEP 138 也包括存储器 206，该存储器可以是例如静态随机存储器（SRAM）。序列发生器 202a-202n 可以使用存储器 206 来存储编码和数据。存储器存取裁决器 208 处理来自序列发生器 202a-202n 和 DPBUS 总线 110 的存储器存取请求。DPBUS 总线接口模块 210 提供系统时钟与 TEP 138 中 DPBUS 总线时钟域之间的桥接。下文对 DPBUS 总线接口模块 201 进行详细阐述。TEP 138 还包括时钟校准块 212，可以用作 TEP 中统一时基的时钟校准。下文对时钟校准块 212 进行详细论述。TEP 138 也可以包括被序列发生器 202a-202n 用来定时的绝对计数器 214。下文对绝对计数器 214 进行详细论述。TEP 138 可以包括 I/O 冲突裁决器 204，用来解决从序列发生器 202a-202n 接收的冲突信号。下文对 I/O 冲突裁决器 204 进行详细论述。下文做详细论述的时钟和电源控制块 216 用来在可能时

给系统时钟断电。

序列发生器 202a-202n 可以是处理器，比如 RISC 处理器，具有专用的指令系统并可以同时为多个无线系统提供定时。即，序列发生器 202a-202n 可以产生置位和清除 GPIO 引脚的信号、给 DMA 控制器发信号、为 DSP 内核 102 和 MCU 104 产生中断。序列发生器可以用来为每一个被同时支持的无线系统执行指令。为每一个无线系统提供两个或更多序列发生器可以提高性能。例如，在本发明的一个实施例中，可以为每一个同时被支持的系统提供两个序列发生器。在这种配置中，一个序列发生器执行指令的同时另一个序列发生器正在加载指令。应该理解单个序列发生器可以支持多个无线系统。单个序列发生器可以加载与两个不同的无线系统相关的指令。然而，使用单个序列发生器不能实现真正的并行性，因为执行和第一个无线系统相关的指令的时间与执行和第二个无线系统相关的指令的时间重叠。因为这些指令由单个序列发生器顺序执行，所以它们不能同时执行。然而，也应该理解不是必须为每一个无线系统使用两个序列发生器。每个无线系统可以使用一个序列发生器，或者使用三个或更多序列发生器。与无线系统的处理不相关的一个附加的序列发生器可以用来提供普通定时。例如，附加的序列发生器可以用来安排计时事件来更新无线终端的显示屏上的时钟。在一些实施例中，TEP 138 包括为每个同时被支持的无线系统的两个序列发生器和一个附加的序列发生器。

使用多个序列发生器允许通信处理器 100 同时与多个不同的无线系统通信，而不管无线系统使用不同定时的事情。例如，无线终端监视 GSM 网络的寻呼信道，同时从无线 LAN、蓝牙网络、或其他 802.11b 网络接收数据。同样，具备具有多个序列发生器的通信处理器的无线终端，一启动就同时执行 2G GSM 网络和 3G WCDMA 网络的单元搜索。

如上所述，TEP 138 能够置位和清除 GPIO 引脚、启动 DMA 通

道、产生中断、和执行时钟校准。然而，两个或多个序列发生器会声明冲突信号。例如，一个序列发生器声明置位一个特定 I/O 引脚的信号，而同时不同的序列发生器声明要清除同一个引脚的信号。如图 2 所示的 I/O 冲突裁决器 204 处理这种冲突。I/O 冲突裁决器 204 包含解决冲突的规则。例如，一个规则可以是任何清除信号优先于置位信号。会产生异常来通知冲突的软件进程，并发送中断到处理内核。对于中断和 DMA 通道启动，会简单地结合冲突信号，例如使用逻辑或操作。

如图 2 所示的存储器，可以通过与 DPBUS 总线 110 连接的 DPBUS 总线接口 210 由处理器访问。根据本发明的一个实施例，存储器 206 是 26 比特宽，并且是多端口的，以允许同时被序列发生器 202a-202n 和 DPBUS 总线 110 访问。为存储器 206 提供的读和写端口的数量根据 TEP 138 中序列发生器的数量来选择。例如，为每一个序列发生器提供一个读端口和一个写端口。然而，大量的端口消耗芯片空间，并需要更多指令解码器。或者为每个序列发生器提供一个指令解码器。此外，不可能所有的序列发生器将在相同时钟周期访问存储器。因此，存储器 206 的读端口的数量可以根据同时被支持的无线系统的数量来选择。例如，为每个被支持的无线系统提供一个读端口。因为写访问发生的频率少于读访问，所以可以提供少于读端口的写端口。存储器 206 端口的数量可以根据任何标准选择，本发明不限定存储器 206 端口的数量。

如上所述，存储器访问裁决器 208 处理来自序列发生器 202a-202n 和 DPBUS 总线 110 的对存储器 206 的访问请求。存储器访问裁决器 208 也处理冲突，例如在访问请求多于读端口的情况下。存储器访问裁决器 208 例如通过基于循环法区分请求的优先顺序来处理这种情况。在这种循环法中，移回（shift-back）寄存器可以用来确定优先顺序。在一个实施例中，当存储器访问发生冲突时，寄存器移位。在另一个实施例中，在序列发生器的每一次存储器访问时，移回寄存器就移位。然而，应该理解可以使用许多其他解决请求冲突的方法。

DPBUS 总线接口模块 210 在系统时钟和 TEP 138 中 DPMBUS 总线时钟域之间提供桥接。DPMBUS 总线接口模块 210 也处理 DPMBUS 总线 110 与内部 TEP 总线之间的 16/32 比特接口。

图 8 表示根据本发明一个实施例的 DPMBUS 总线接口模块 210 的框图。如上所述，DPMBUS 总线接口模块 210 执行 DPMBUS 总线时钟与系统时钟之间的时钟同步。可以利用用于相互同步的联络信号分别控制每个时钟域。DPS 协议 FSM 802 处理至 DPMBUS 总线的联络信号。TEP 访问 FSM 804 处理至内部 TEP 总线的联络信号。

图 3 是用于说明根据本发明一个实施例的序列发生器 202a 结构的框图。序列发生器 202a 可以是包括读取、解码、和执行阶段的流水线处理器。指令解码器 328 解码从多路复用器 342 接收的指令。指令读取自 TEP 138 的存储器 206。在 DSP 内核 102 和 MCU 104 的控制下，序列发生器指令被加载到 TEP 138 的存储器 206 中。多路复用器 342 将与指令相关的数据送至寄存器 326。寄存器 326 中存储的数据是读或变更操作中被写入的数据或者是从读操作得到的数据。序列发生器 202a 还包括与 DMA 控制器 134 和 DMA 控制器 136 接口的 DMA 控制模块 348。序列发生器 202a 包括多个 DMA 寄存器（例如，302、304、306、308、310），用来配置 DMA 通道。序列发生器 202a 包括时钟预变换（pre-scale）模块 346，用来产生时间脉冲信号以增加 delta 计时器 336。序列发生器控制模块 334 处理序列发生器的全部操作，在下文进行详细论述。

图 4 表示由序列发生器 202a-202n 执行的指令的格式的例子。指令 400 包括用于识别指令类型的 6 比特操作码字段 402。4 比特数据字段 404 包括处理指令需要的数据。例如，用于置位通用输入输出（GPIO）引脚的指令包括用于识别要置位的 GPIO 引脚的数据域。扩展字段 406 作为数据字段 404 的 8 比特扩展可以选择地使用。如果与

指令相关的数据超过 4 比特数据字段，可以使用扩展字段 406 保持溢出的部分。Delta 时间字段 408 用来指示指令执行前的时间延迟。Delta 时间字段 408 指示在前一个指令执行后和当前指令（即指令 400）执行前要等待的时间。

由指令解码器 328 对指令解码后，图 3 所示的 delta 计时器 336 计时在指令的 delta 域中指示的等待时间段。当时间延迟达到了 delta 计时器 336 的时间，执行单元 330 将执行命令。基于 delta 计时器的指令执行允许序列发生器执行随时间变化的功能，即计划在特定时间发生的功能。例如，序列发生器可以向处理内核 102 和 104（图 1）产生定时中断以允许处理器不工作时进入空闲状态，并且产生定时中断以在适当的时间退出这种空闲状态。序列发生器指令可以控制引脚的置位从而控制外部设备，并可以给无线电单元加电或断电。序列发生器指令也可以在特定的时间启动 DMA 通道。

图 3 所示的时钟预变换模块 346 用来产生时间脉冲信号以增加 delta 计时器 336。时钟预变换模块 346 执行系统时钟的时钟分割以产生时间脉冲信号。为了节约电能，期望尽可能地使用低频率，而仍能够为无线系统的操作提供充分的定时精度。因为时钟频率取决于无线系统的定时，时钟预变换模块 346 可以利用 2 至 64 之间的任何预变换值分割系统时钟。预变换值储存在寄存器 314 中。

表 1 给出了根据本发明一个实施例的序列发生器指令系统的例子。

表 1

操作码	指令 (6 比特)	数据字段 (4 比特)	扩展字段 (8 比特)	Delta 时间 (8 比特)	说明
0x00	Clr GPIOA	选择引脚	不使用	Delta 时间	清除所选择的 GPIOA 引脚，与 GPSigA 同步
0x01	Set GPIOA	选择引脚	不使用	Delta 时间	置位所选择的 GPIOA 引脚，同步 到 GPSigA
0x02	Clr GPIOB	选择引脚	不使用	Delta 时间	清除所选择的 GPIOB 引脚
0x03	Set GPIOB	选择引脚	不使用	Delta 时间	置位所选择的 GPIOB 引脚
0x04	Clr GPIOC	选择引脚	不使用	Delta 时间	清除所选择的 GPIOC 引脚
0x05	Set GPIOC	选择引脚	不使用	Delta 时间	置位所选择的 GPIOC 引脚
0x06	Toggle GPSigA	选择信号	不使用	Delta 时间	在所选择的 GPSigA 线产生脉冲
0x07	Toggle GPSigB	选择信号	不使用	Delta 时间	在所选择的 GPSigB 线产生脉冲
0x08	Toggle ARMInt	选择信号	不使用	Delta 时间	在所选择的 MCU 中断线产生脉冲
0x09	Toggle DSPInt	选择信号	不使用	Delta 时间	在所选择的 DSP 中 断线产生脉冲
0x0A	Toggle DMAInt	选择通道	不使用	Delta 时间	在所选择的启动相 关 DMA 通道的 Sys. DMA 启动线产生脉 冲
0x0B	Toggle DSPDMAInt	选择通道	不使用	Delta 时间	在所选择的启动相 关 DMA 通道的 DSP. DMA 启动线产生脉 冲
0x0C	Short Wait	不使用	4 比特不使 用，4 比特 等待	等待	等待<等待> 时间脉冲信号
0x3x	LongWait (2 比特) 绝对时间 (24 比特)				等待到<Abs.Time> 到达加上 3-4 SysClk 周期附加延迟，开 始下一个指令。指

				令隐含实现校准相位补偿
0x10	If...Skip Next	地址 (12 比特)	比特	在内部 TEP 存储器的<地址>测试<比特>, 如果置位, 下一个指令将被具有相同执行延迟的 NOP 替代
0x11	ClrBit	地址 (12 比特)	比特	清除<地址>中的<比特>
0x12	SetBit	地址 (12 比特)	比特	置位<地址>中的<比特>
0x15	Shift	地址 (12 比特)	不使用	将<地址>的内容左移一比特
0x0D	Wait	触发选择, 比特 6: DSP DMA ch0 比特 5-0: GPSigB[5:0] 6 比特不使用	不使用	等待直到发生所选择的触发 GPSigB[5:0]或 DSP DMA 通道 0 IRQOut
0x14	NOP	不使用		无操作
0x0E	Jump	绝对地址	不使用	标准分支, 跳转到<绝对地址>
0x0F	Die	不使用		停止执行并将序列发生器硬件置为空闲
0x18	WriteRegF	Reg.File 地址	写数据 (16 比特)	将给定数据立即写入序列发生器的内部寄存器
0x13	DataMoveE	不使用		启动 DMA 传送, sing 存储在 RegFile 中的 DMA 设置

有时候必须同时置位或清除两个或更多 I/O 引脚。尽管序列发生器指令系统可以提供置位或清除 I/O 引脚的指令, 但这些指令顺次扩展, 不是同时的。同时置位或清除两个或更多 I/O 引脚, 置位和清除指令同步到特定的信号。例如, 如果需要同时置位引脚 GPIOA 和引脚 GPIOB, 序列发生器会将这些信号同步到 GPSigA。然后, 序列发生器首先执行置位引脚 GPIOA 的指令, 接着是置位引脚 GPIOB 的指令。直到执行了切换 GPSigA 指令后, 这些引脚才被真正置位, 该指

令使得两个引脚同时被置位。

当序列发生器执行 LongWait 指令时使用 LongWait 比较模块 340。当给定数量的时间内序列发生器不执行序列发生器指令的时候，就执行 LongWait 指令。为了节省电能，LongWait 指令允许系统时钟断电并允许序列发生器使用慢时钟来定时。

LongWait 比较模块 340 将 LongWait 指令所指示的等待时间与绝对计数器 214（图 2）的值进行比较，下文进行详细论述。等待时间是 24 比特的值，于是需要使用 8 比特 delta 时间字段、8 比特扩展字段、4 比特数据字段、和 6 比特操作码的 4 比特。LongWait 比较模块 340 接收来自 24 位绝对计数器 214 的输入，并将该值与 LongWait 指令的 24 比特等待时间进行比较。当值相匹配时，序列发生器执行接下来的指令。LongWait 比较模块 340 也输出空闲信息，TEP 138 的时钟和电源控制块 216 利用该信息来确定序列发生器是否正在执行 LongWait 指令，从而当所有序列发生器在空闲状态时给系统时钟断电。

如果振荡器已经关闭，PreAbs32 寄存器 338 用来确定给振荡器加电的时间。PreAbs32 寄存器指示给振荡器加电的绝对时间点，允许振荡器在当前正在执行的 LongWait 指令结束和下一个指令开始执行以前有充分的时间来稳定。

序列发生器控制模块 334 控制程序流程并处理序列发生器 202a-202n 的中断。序列发生器控制模块 334 根据程序计数寄存器 322 中的内容来请求来自存储器的指令。程序计数寄存器 322 保存下一个将被执行的指令的地址。序列发生器控制模块 334 通过线路 344 接收来自中断选择器 332 的中断，该中断选择器 332 从多个中断源中选择最高优先次序的中断请求。当接收到中断，寄存器 316 中的中断允许位被置位，并且中断向量的地址被加载到寄存器 318。序列发生器跳转至

寄存器 318 中中断向量的地址并从那里继续执行。

当序列发生器接收到硬复位或执行终止指令，序列发生器进入空闲状态。软复位被用来指示序列发生器读取第一条指令并开始执行指令。当序列发生器接收到软复位或中断，它将继续正常执行。如果序列发生器接收到软复位，序列发生器要跳转至并开始执行的地址保存在寄存器 320 中。如果序列发生器接收到中断，序列发生器要跳转至并开始执行的中断向量的地址保存在寄存器 318 中。

DMA 寄存器 302、304、306、308 和 310 被序列发生器用来存储 DMA 通道配置信息。例如，这些寄存器存储源地址、目标地址、和若干要传送的字节。DMA 控制模块 348 连接 DSPDMA 控制器 134（图 1）和系统 DMA 控制器 136 来初始化 DMA 传送。

图 6 表示 TEP 138 和 DMA 控制器 134 之间接口的例子。一个 DMA 通道由 TEP 138 专用而不能被其他资源利用。例如，尽管任何 DMA 通道都能使用，分配通道 0 被 TEP 使用。通过执行指令，序列发生器 202a-202n 利用固定的通道来启动 DMA 传送，例如表 1 的序列发生器指令系统所示的 DataMoveE 指令。DataMoveE 指令从 DMA 寄存器 302、304、306、308、和 310 中得到 DMA 通道配置信息，并将该信息复制到 DMA 控制器 134 的内部 RAM 604。若干序列发生器会同时请求访问专用的 DMA 通道。请求裁决器有限态机器（FSM）218 处理这些同时的请求。例如，请求裁决器 FSM 218 使用循环优先权方法来允许序列发生器 202a-202n 有权使用 DMA 通道。当访问被允许后，DMA 寄存器的值被复制到 DMA 控制器的存储器 604，请求裁决器 FSM 218 置位用于启动专用 DMA 通道的通道启动标志。当 DMA 传送结束后，DMA 控制器 134 向请求裁决器 FSM 218 返回一个中断。此外，请求裁决器 FSM 218 在使用中会声明一个 DRReqSysClk 标志（未表示），以确保在 DMA 传送中系统时钟不会被断电。

图 7 示意地表示如以上关于图 6 的说明的 TEP 初始化 DMA 传送的例子。首先，TEP 138 利用专用通道向 DMA 控制器 134 发送通道配置信息，并向 DMA 控制器 134 发送通道启动信息。DMA 控制器 134 执行数据传送并向 TEP 138 发送中断以表示数据传送的结束。

图 2 所示的时钟校准单元 212 用来校准通信处理器 100 的慢时钟。通信处理器 100 从例如 13MHz 的系统时钟和例如 32kHz 的自由运行慢时钟接收时钟信号。如系统时钟那样的高频时钟用来给通信处理器 100 的处理内核计时，而慢时钟可以在处理内核处于空闲状态，并且不需要由高频时钟计时的时候用作计时控制，以节省电能。TEP 138 可以从慢时钟得到系统计时。由 TEP 138 处理的计时事件基于慢时钟的时间和与慢时钟周期相关的 δ 时间（在系统时钟周期中计数）。当不需要时，系统振荡器可以由通信处理器 100 的任何模块断电，当接下来预定的操作需要时，依靠慢时钟来给系统时钟加电。系统振荡器的断电在下文进行详细论述。

慢时钟没有高频系统时钟精确，并且对温度变化更敏感。因此，要校准慢时钟以保证期望的精确程度。可以利用系统时钟或通过无线电接收的无线系统的计时来校准慢时钟。如果利用系统时钟校准，在所选择数量的慢时钟周期之上计算系统时钟周期的数量。如果利用来自无线系统的计时来校准，在所选择数量的慢时钟周期之上计算无线系统时钟周期（通过无线电接收）的数量。

利用频率合成器或 VCO 校准慢时钟将消耗功率，为了节电，通过从自由运行慢时钟中除去时钟周期以提供校准后的慢时钟来校准慢时钟。即，选择比自由运行慢时钟的预期频率低的频率（例如，在 32kHz 慢时钟的情况下选择 31kHz），并且通过从自由运行慢时钟信号中除去时钟脉冲来生成校准后的时钟信号。通过 N 分时器来调整自由运行慢时钟，它定期地从自由运行慢时钟中除去时钟周期。从自由运行慢时钟中除去时钟周期的时间段取决于规定的分数、系数值，和比较慢时

钟和系统时钟而获得信息。例如，如果除去时钟周期的时间段是 9 个慢时钟周期，则每 9 个未校准的慢时钟周期会生成 8 个校准后的慢时钟周期。

然而，从自由运行慢时钟周期中除去时钟周期会将相位误差引入校准后的慢时钟。这种相位误差是由于校准后的慢时钟不是真正的周期的而引起的。例如，假定 40 kHz 校准后的慢时钟是从自由运行 50 kHz 时钟中生成。50 kHz 时钟每 20 μ 秒具有一个上升沿。即，50 kHz 时钟在 20 μ 秒、40 μ 秒、60 μ 秒、80 μ 秒、100 μ 秒、120 μ 秒等具有上升沿。40 kHz 校准后的慢时钟是通过定期除去周期而生成。因此，校准后的慢时钟将在 20 μ 秒、40 μ 秒、60 μ 秒、100 μ 秒、120 μ 秒等具有上升沿。校准后的慢时钟达到 40 kHz 的平均数，即每秒 40,000 时钟上升沿，但是与真正的 40 kHz 时钟不同相。真正的 40 kHz 时钟每 25 μ 秒具有一个时钟上升沿。例如，真正的 40 kHz 时钟将在 25 μ 秒、50 μ 秒、75 μ 秒、100 μ 秒、125 μ 秒等具有上升沿。因此，校准后的 40 kHz 时钟的上升沿与真正的 40 kHz 时钟的上升沿发生在不同的时间，由于校准后的慢时钟与相同频率的真正的时钟之间的相位差异而采用相位补偿，下文将阐述。

图 9A 和 9B 分别表示根据本发明的一个实施例，带相位补偿的 N 分时器的实现和用于说明相位补偿的时序图。分数增加寄存器 902 存储快时钟周期与自由运行 32 kHz 时钟周期的比率，并作为加法器 904 的一个输入。相位补偿寄存器 906 是用于累加加法器 904 的输出的累加器并作为模数操作器 912 的一个输入。模数寄存器 908 存储的值通过比较器 914 与相位补偿寄存器 906 中的上 10 位进行比较。比较器 914 作为与门 910 的输入并控制自由运行 32 kHz 时钟是否通过门 910。模数寄存器 908 是模数操作器 912 的第二个输入。模数操作器 912 计算模量，当比较器被置位时，该模量被用来增加相位补偿寄存器 906。

当寄存器 906 中的值达到模数寄存器 908 中的值，比较器 914 的

输出被置位，从而禁止门 910 的输出。可以看出，在每一个自由运行 32 kHz 时钟周期，相位补偿的数量（即，寄存器 906 的值）被累积并线性增长。当累加器达到模数寄存器 908，自由运行 32 kHz 时钟周期输入被禁止，直到下一个时钟周期。然后相位补偿累加器由经模数操作器 912 计算的模量回绕（wrap around）。

如上所述，除去时钟脉冲将相位误差引入校准后的时钟信号。相位误差的产生是由于图 9B 所示的已除去脉冲的校准后的慢时钟 930 具有与同频率的自由运行时钟不同时间发生的时钟沿。如图 9B 所示的波形 932，相位误差伴随每一个慢时钟周期增加直到除去一个脉冲，然后归零。如果不进行相位补偿，这种相位误差将在无线系统中导致定时误差。通过利用校准后的慢时钟和代表相位误差的相位补偿信号，可以利用校准后的慢时钟获得精确的定时。

因此，当使用校准后的时钟信号驱动绝对计数器 214 的时候，通过利用相位补偿寄存器 906 中计算的相位补偿来补偿校准后的时钟信号中的相位误差。参考上文利用 50kHz 自由运行慢时钟和 40kHz 校准后的慢时钟而论述的例子，假定一个事件预定在 40kHz 时钟信号的第三个上升沿发生。如上所述，在真正的 40kHz 时钟中，第三个上升沿发生在 75 μ 秒。而在校准后的 40kHz 时钟中，第三个上升沿发生在 60 μ 秒。因此，校准后的慢时钟与真正的 40kHz 时钟不同相 15 μ 秒。当校准后的慢时钟在 60 μ 秒到达第三个上升沿时，在系统时钟周期中计算的 15 μ 秒的进一步延迟在预定时间执行前被增加。如此，序列发生器补偿校准后的时钟信号的调整后的频率。

有时，由于例如温度的迅速变化，不能得到慢时钟的充分的稳定性。然而，仍然需要生成校准后的慢时钟信号，来驱动绝对计数器并为 LongWait 指令的执行定时。在这种情况下，可以使用系统时钟的频分。例如，分时器可以将系统时钟分割为校准后的慢时钟。

图 2 所示的绝对计数器 214 由校准后的慢时钟计时。图 10 更详细表示了绝对计数器 214。在一个实施例中，绝对计数器 214 是 24 比特计数器，并且由正在执行 LongWait 指令的序列发生器 202a-202n 使用，来确定等待时间段何时期满。例如，序列发生器 202a-202n 将绝对计数器 214 的值与 LongWait 指令的等待时间段进行比较，来确定等待时间段何时期满。

提供两个周期性触发脉冲发生器 1002 和 1004，可以用作各种用途，比如触发中断或者触发快照。快照是慢时钟相对于系统时钟的测量，或者慢时钟相对于通过无线电接收的无线系统的计时的测量，它们用作慢时钟的校准。快照包括计算给定的慢时钟周期数量中系统时钟周期的数量。

图 11A 和 11B 分别表示用于获得快照的框图和状态转移图。快照由多个不同的输入启动。例如，基于任何序列发生器的寄存器文件中的 SeqCtrl 寄存器 312（图 3）的置位比特，快照可以由绝对计数器的两个周期性的触发脉冲中的任何一个启动，或者由运行在两个处理内核的任何一个中的软件启动。

当启动快照时，校准信号被声明以防止系统时钟断电。接下来，快照 FSM 1108 进入准备状态 1103，在该状态中它等待接收 SysClkOk 信号 1110，表明系统时钟振荡器没有断电。当接收到 SysClkOk 信号 1110，快照 FSM 1108 进入快照状态 1105，在该状态中，在若干慢时钟周期时间段计算系统时钟周期的数量。在 TCLR 寄存器 1112 中规定慢时钟周期的数量，该数量可由软件配置。在 TCLR 寄存器 1112 中规定的慢时钟周期的数量已经被慢时钟周期计数器 1114 计数之后，产生中断并且快照 FSM 1108 进入回读状态 1107。当快照 FSM 1108 在回读状态 1107 时，系统时钟周期计数器 1116 可由处理内核经由 DPBUS 总线接口 210 读取，以更新慢时钟的校准所需要的任何计数器。读计数器 1116 之后，快照 FSM 1108 返回空闲状态 1101。

图 2 表示了时钟和电源控制块 216。当 TEP 138 的某些模块不使用时，为了省电可以关闭至这些模块的时钟信号。如果一个或多个序列发生器正在执行 LongWait 指令并且没有其他模块需要使用系统时钟，时钟和电源控制块 216 确定 LongWait 指令的持续时间是否充分以允许系统时钟断电。

图 12 是时钟和电源控制块 216 的框图。许多 TEP 模块需要使用系统时钟，比如存储器存取裁决器 208 和 DMA 请求裁决器 218，它们可以分别通过信号 1206 和 1208 向时钟和电源控制块 216 指示需要系统时钟。外部源提供外部信号 ReqSysClk 1210 来向 TEP 指示一个或多个外部模块需要使用系统时钟。关于何时提供 ReqSysClk 信号的确定在下文详细论述。时钟校准模块在执行慢时钟校准时声明校准信号 1212，并请求系统时钟维持激活。

每个序列发生器 202a-202n 通过置位复位触发器（SRFF）1218 向时钟和电源控制块 216 指示需要系统时钟。SRFF 1218 的 Q 输出是 TEPReqSysClk 信号 1216。不需要系统时钟的每个序列发生器通过与门 1222 声明 KillSysOsc 信号。当没有序列发生器需要系统时钟时，SRFF 1218 进入复位状态，不声明信号 1216。如果有任何序列发生器需要使用系统时钟，它通过或门 1224 声明一个重启系统振荡器信号。作为响应，SRFF 1218 进入置位状态并声明信号 1216。考虑下一条指令的执行时间和振荡器必要的预热时间，PreAbs32 寄存器 338 用来存储振荡器保持断电的最近的时间。如果当前时间小于它的 PreAbs32 寄存器 338 中的时间，序列发生器声明 KillSysOsc 信号。如果当前时间等于它的 PreAbs32 寄存器 338 中的时间，序列发生器声明 RestartSysOsc 信号。当前时间等于 LongWait 指令的期满时间时，应该稳定系统时钟。

加电序列发生器 1226 接收来自或门 1220 的输入信号，该信号向

TEP 指示是否有任何内部或外部模块需要使用系统时钟。如果声明了该信号，加电序列发生器 1226 通过声明 SysOscOn 信号 1236 来给系统时钟振荡器加电。时钟 pad 加电寄存器 (CPPUR) 1228 存储时钟 pad 缓冲器的设置时间，振荡器预热寄存器 OWUR 1230 存储振荡器的预热时间。当从或门 1220 到加电序列发生器 1226 的输入信号被声明时，FSM 1234 从零启动 10 比特计数器 1232 并声明 SysOscOn 信号 1236，从而给系统时钟振荡器加电。当计数器 1232 达到了 OWUR 1230 中规定的时间，ClkBufOn 信号 1238 被声明，来启动时钟 pad 缓冲器。当计数器 1232 等于 OWUR 1230 中规定的时间和 CPPUR 1228 中规定的时间相加之和时，SysClkGate 信号 1240 被声明，来表明系统时钟振荡器输出有效并且启用与门 1242。与门 1242 禁止系统时钟振荡器输出直到振荡器有充分的时间稳定。在到达了 OWUR 1230 中规定的振荡器预热时间与 CPPUR 1228 中存储的时钟 pad 加电延迟时间相加的时间之后，振荡器稳定。当到达该时间时，SysClkGate 信号 1240 启用与门 1242，并且允许来自振荡器的时钟信号从门通过。

如上所述，时钟和电源控制块 216 接收来自 TEP 138 外部资源的 ReqSysclk 信号。该信号表明是否有任何 TEP 138 外部的模块需要使用系统振荡器，比如 DSP 内核 102 和 MCU 104。图 13A 表示时钟信号如何在通信处理器 100 中产生。电源 1300 为系统振荡器 1301 供电。可以如上所述，电源由来自 TEP 的 SysOscOn 信号来控制。该信号被用来控制振荡器 1301 加电或断电。振荡器输出被输入到 pad 缓冲放大器 1303。缓冲放大器 1303 通过来自 TEP 的控制信号加电和断电。缓冲放大器 1303 的时钟信号输出被输入到与门 1305。门 1305 的第二个输入是来自 TEP 138 的 SysClkGate 信号，该信号允许在振荡器预热期间禁止振荡器的输出。

来自门 1305 的时钟信号输出被输入到锁相环 (PLL) 1307，它将时钟信号增加到适于为 DSP 内核 102 计时的频率。在 DSP 内核空闲的状态下，不需要使用 PLL 1307 增加时钟信号，并且从门 1305 输

出的时钟信号不提供给 PLL 1307。多路复用器 1309 或者选择来自 PLL 1307 的被增加的时钟信号，或者选择门 1305 的输出。如图 13B 所示，从多路复用器 1309 的输出可以生成许多时钟信号。首先，DCLK 时钟作为 PLL 1307 的输出而生成。DCLK 时钟用来为 DSP 内核 102 计时。Not-gated DCLK (nGDCLK) 时钟 1319 输入到与门 1311。当 DSP 内核不需要 DCLK 时钟时，利用与门 1311 将其禁止 (gate off)。然后，利用分频器 1321 分割 DCLK 时钟而生成 DSCLK 时钟。分频器 1321 是软件可编程的，用 1 或 2 来除 DCLK 时钟。DSCLK 时钟用来为 DSP 子系统计时，包括 DSP 外围设备和 DSP DMA 控制器 134。当不需要 DSCLK 时钟时，利用与门 1313 将其禁止。Not-gated DSCLK (nGDSCLK) 时钟 1323 提供给可编程分时器 1325，它利用 1 至 8 之间的数字分割输入信号来生成 BCLK 时钟。BCLK 时钟用来驱动通信处理器 100 的总线。当不需要 BCLK 时钟时，利用与门 1315 将其禁止。MCLK 时钟与 BCLK 时钟的频率相同，用来为 MCU 104 定时。当不需要 MCLK 时钟时，利用与门 1317 将其禁止。

图 14A 和 14B 表示在不需要时如何禁止图 13B 中的时钟信号。如图 14A 所示，PLL 1307 增加振荡器输出以生成时钟信号。分时钟器 1419 与图 13B 中的分时器 1321 和 1325 执行相同的操作。分时器 1419 的时钟信号输出是 ngDCLK 信号、nGDSCLK 信号、和 nGDBCLK 信号。每一个时钟信号直接至多路复用器 1309a-1309c 之一，然后至适当的与门 1311-1315。

寄存器 1405 是 MCU 休眠时钟需求寄存器 (MSCRR)。如图 14B 所知，MSCRR 寄存器 1405 表明当 MCU 休眠或处于空闲模式时需要哪些时钟。同样，MCU 激活时钟需求寄存器 (MACRR) 1407 存储关于 MCU 激活时需要哪些时钟的信息。由 MCU 104 生成的 MCU 激活信号 1427 被多路复用器 1423 用来确定输出 MSCRR 寄存器 1405 或是 MACRR 寄存器 1407 的内容。当置位 MSCRR 寄存器 1405 中的 PLL 旁路位时，允许当 MCU 104 休眠时绕过 PLL 1307。因为 MCU 104 在

休眠时不需要定时，所以 PLL 1307 不需要增加振荡器至高频来驱动 MCU 104。因此，绕过 PLL 1307 可以节省电能。此外，在不需要高处理速度的情况下，DSP 内核 102 和 MCU 104 可以以未经过 PLL 累加而输入到通信处理器的系统时钟运行。

与 MCU 104 同样，也提供了两个 DSP 寄存器：DSP 休眠时钟需求寄存器 (DSCRR) 1401 和 DSP 激活时钟需求寄存器 (DACRR) 1403。寄存器 1401 和 1403 分别表明当 DSP 内核 102 休眠和当 DSP 内核 102 激活时各需要哪些时钟。由 DSP 内核 102 生成的 DSP 激活信号被多路复用器 1421 用来确定输出 DSCRR 寄存器 1401 或是 DACRR 寄存器 1403 的内容。或门 1409、1411、和 1413 结合 MCU 需求寄存器 1405、1407 和 DSP 需求寄存器 1401、1403 的输出。与门 1415、1311、1313、1315、和 1317 用来启动或禁止与寄存器 1401、1403、1405、和 1407 的内容相应的时钟信号。

除了当不需要某些时钟信号时将其禁止以节电以外，当通信处理器 100 没有模块需要时钟时，系统时钟振荡器断电从而不生成系统时钟信号。图 15 表示如何给振荡器断电。DSP 内核 102 和 MCU 内核 104 更新寄存器 1503，表明处理内核是否需要时钟信号以及是否有任何外围设备需要时钟信号。时钟控制模块 1501 监视寄存器以确定通信处理器 100 是否有任何模块需要时钟信号。如果不需要任何时钟信号，时钟控制模块 1501 向 TEP 138 发送 SysClkReq 信号。如上文参照图 12 的论述，TEP 138 决定是否给系统振荡器断电。按照这样，为了省电，当需要时给系统振荡器加电，不需要时断电。

以上阐述了本发明的各种实施例，本领域的技术人员将做各种改进和变更。因此，本发明的范围不限于所说明的特定实施例。本发明的范围只由权利要求和与它们等价的部分来限定。

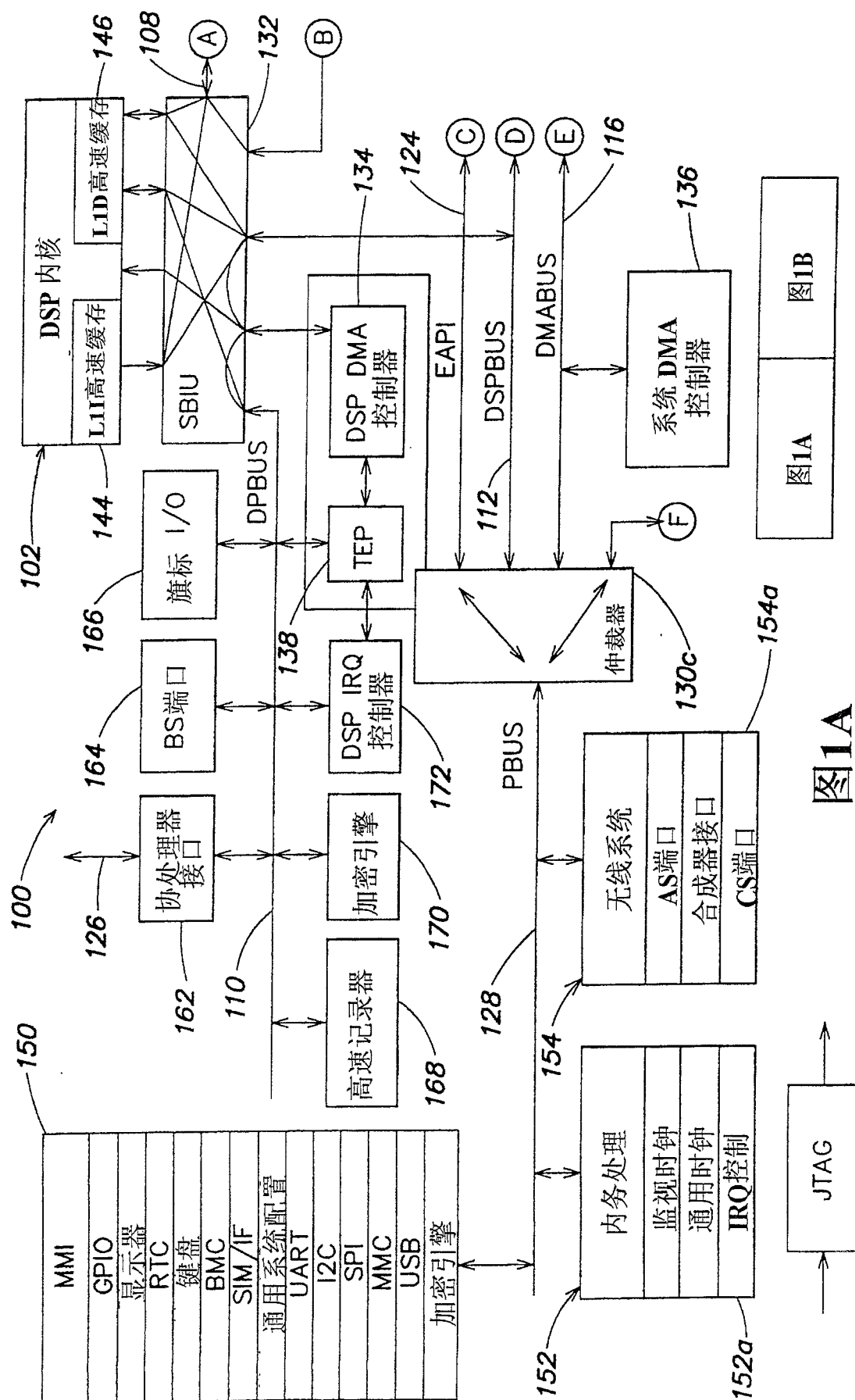


图1A

JTAG

图1B

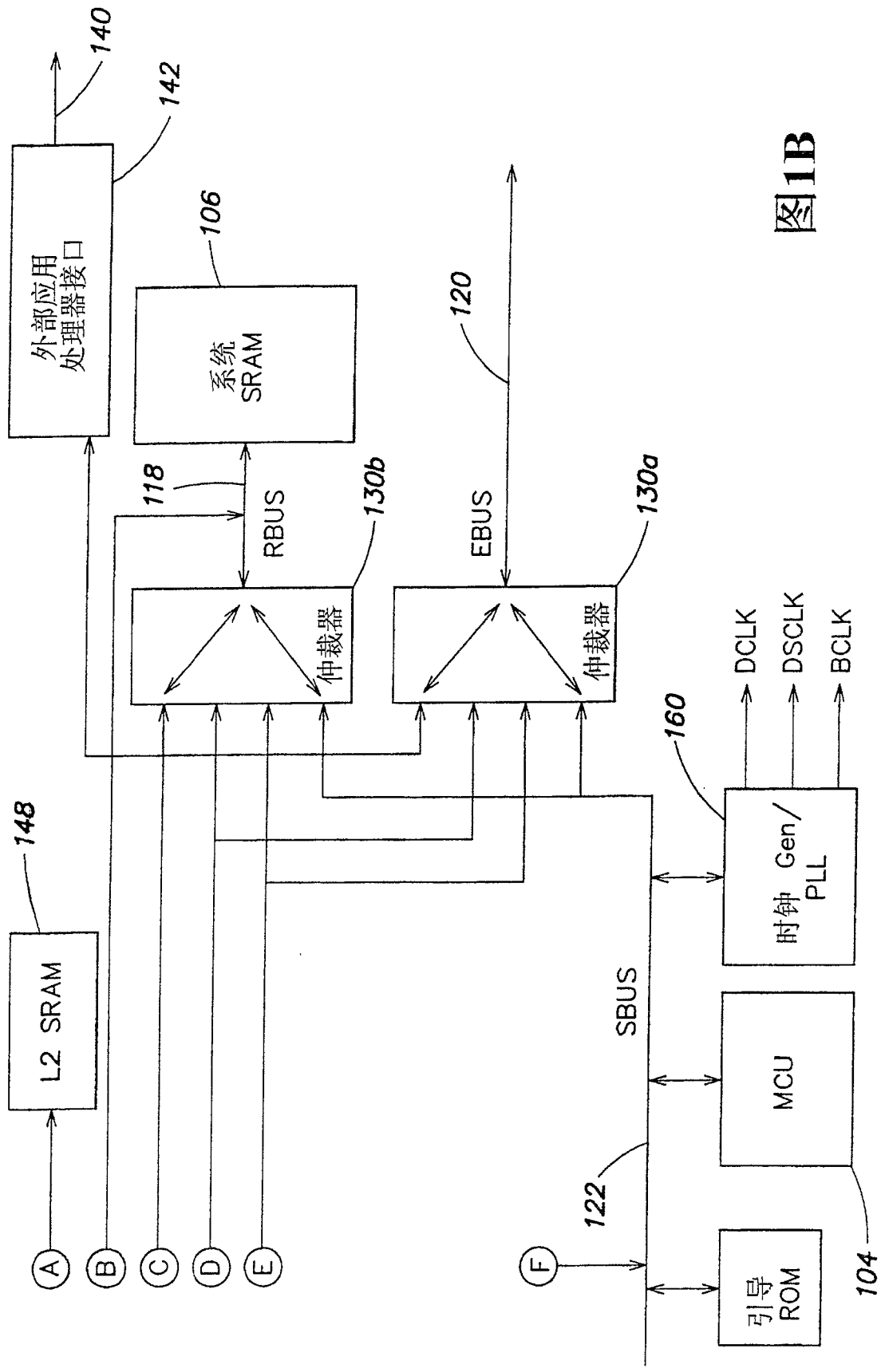
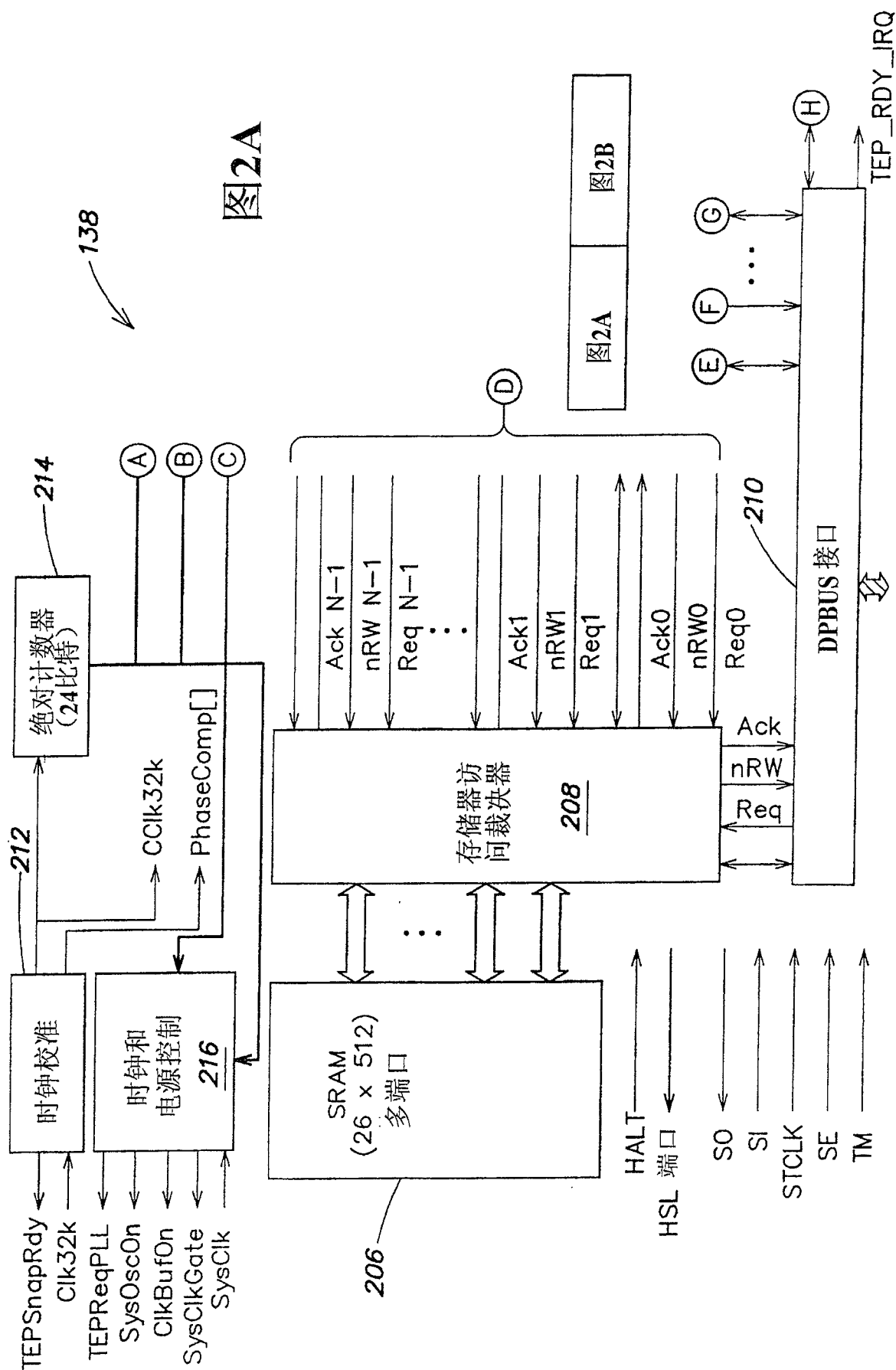
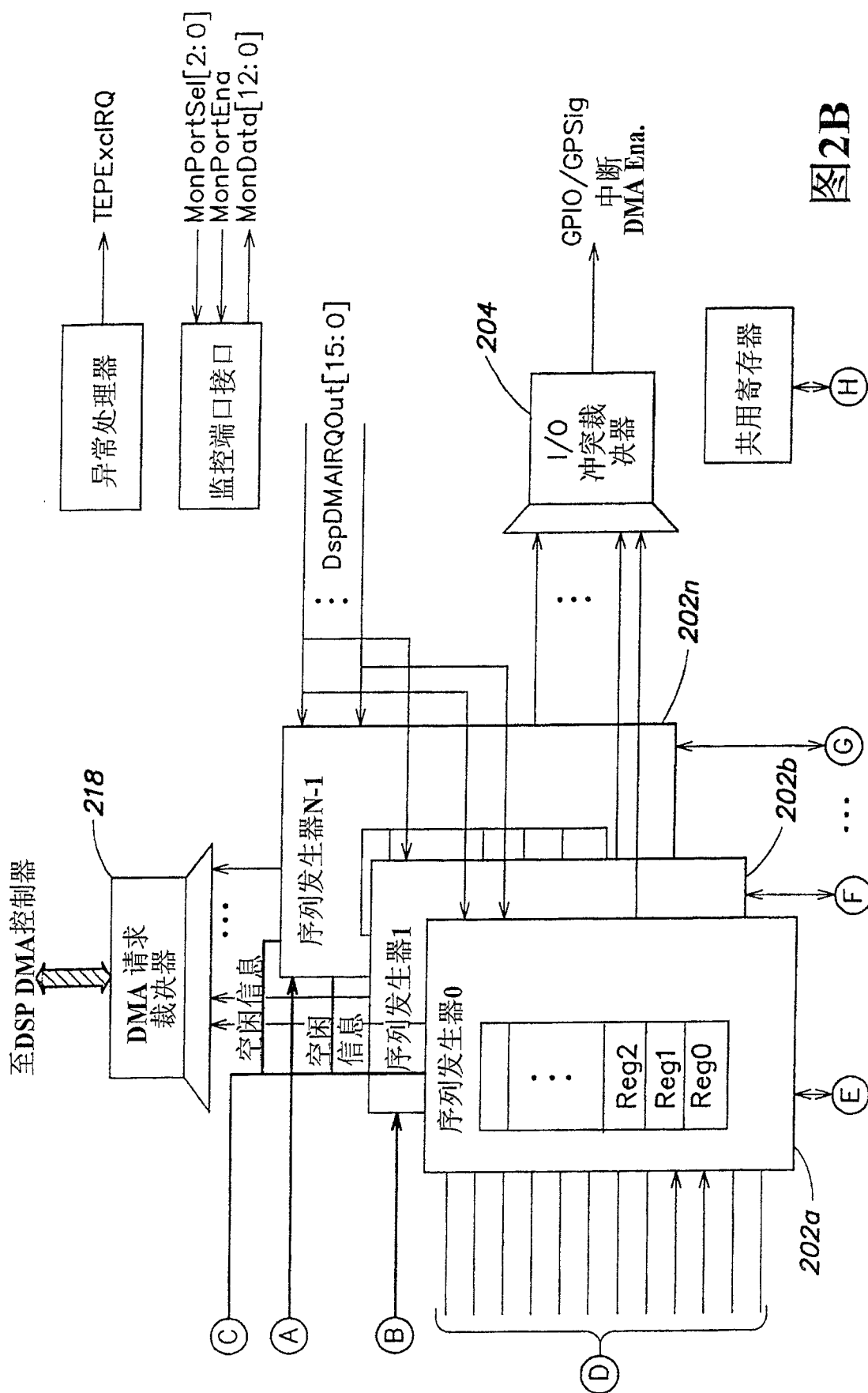
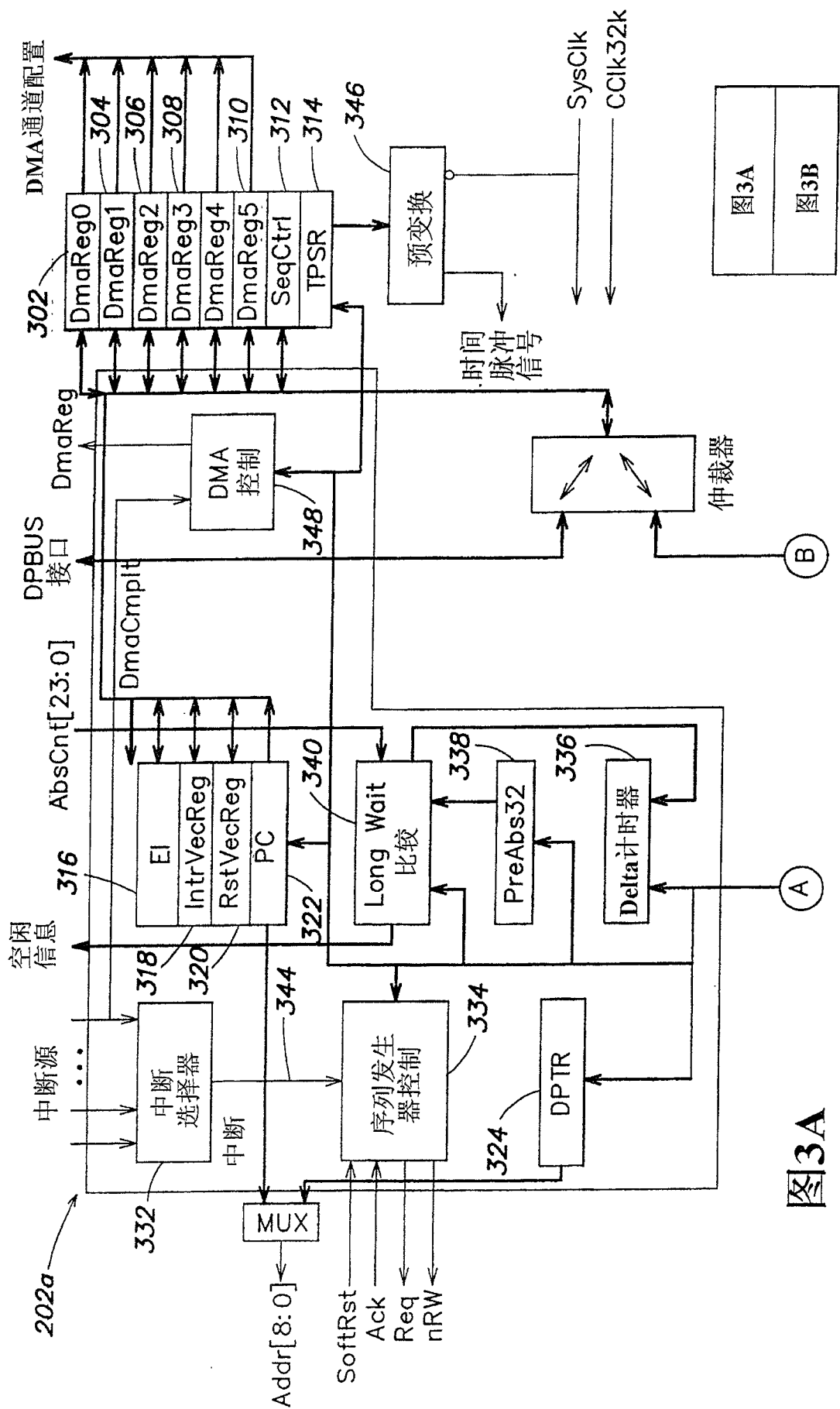


图1B





2B



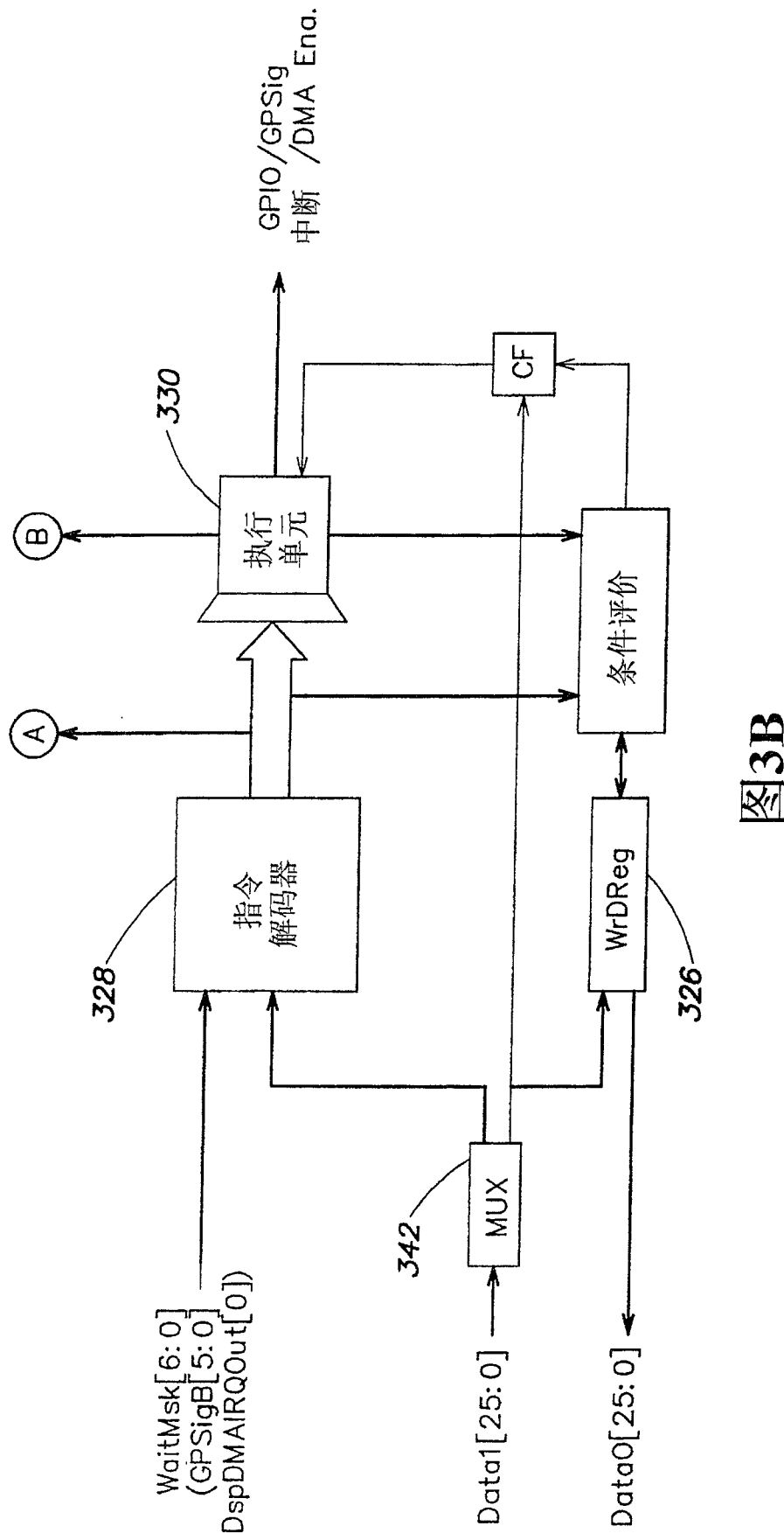


图3B

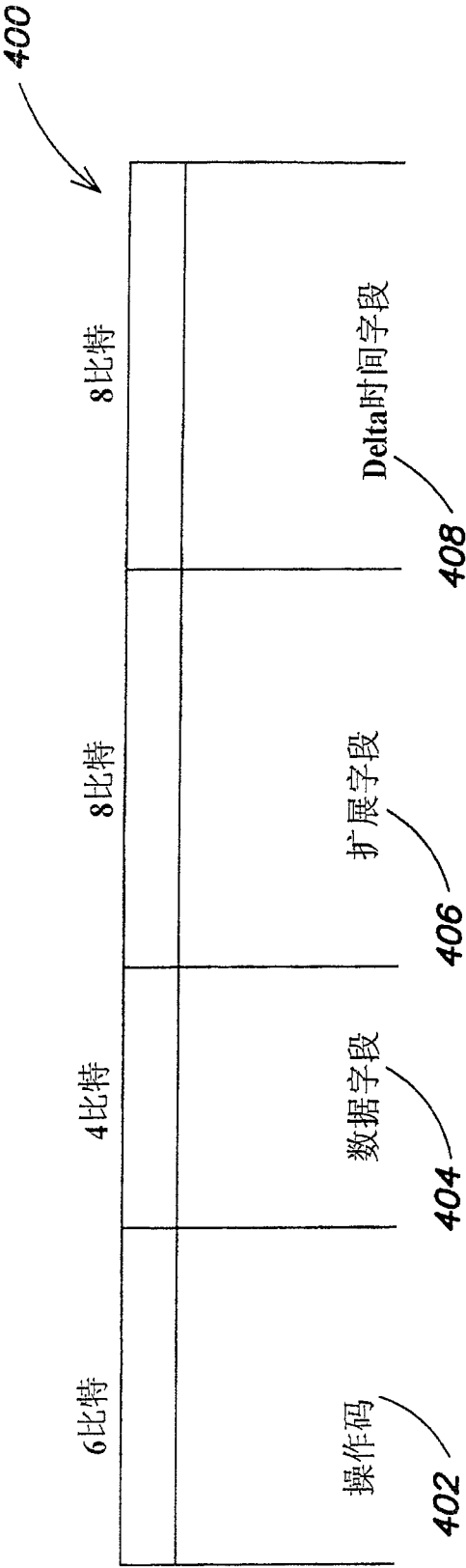


图4

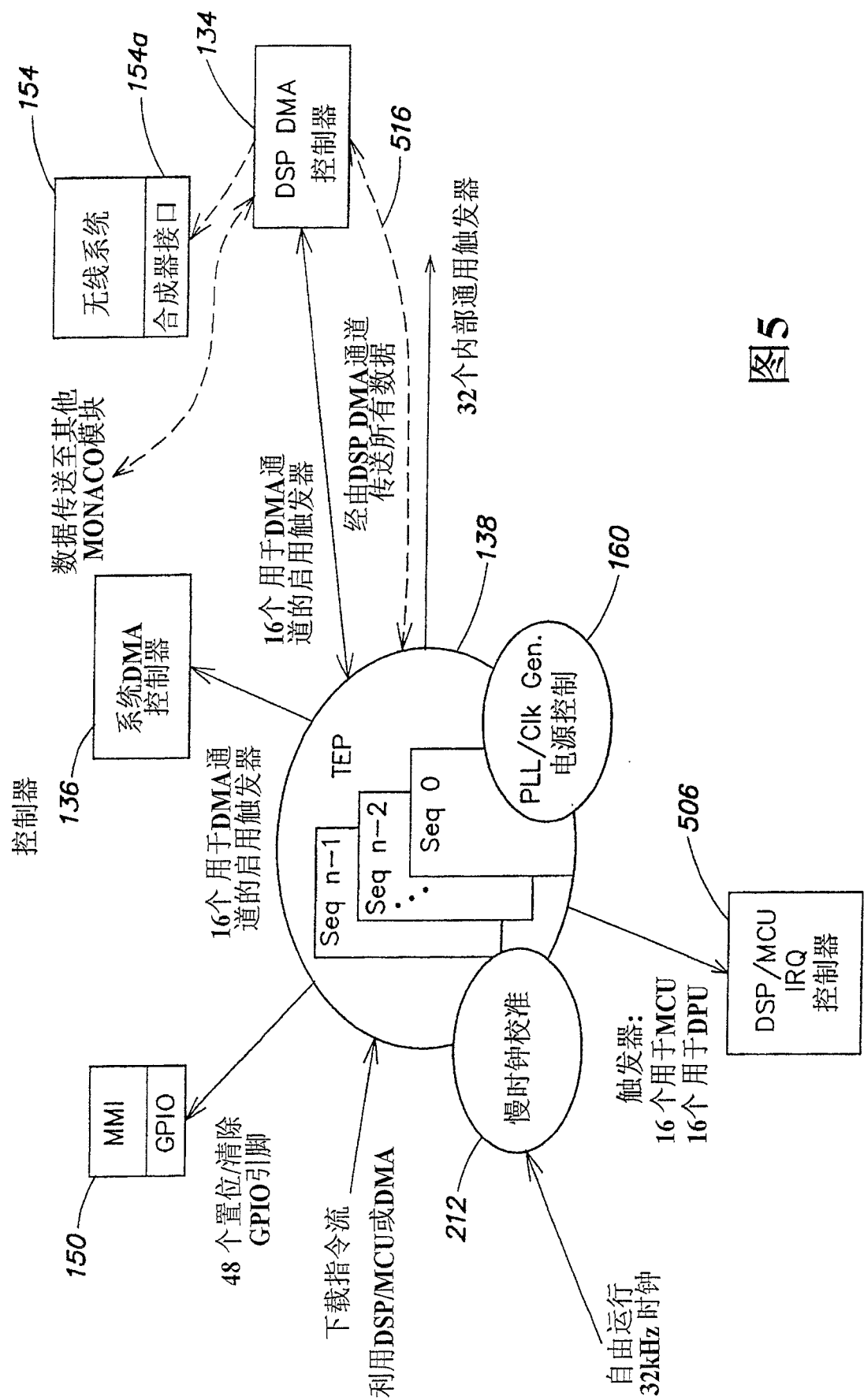


图5

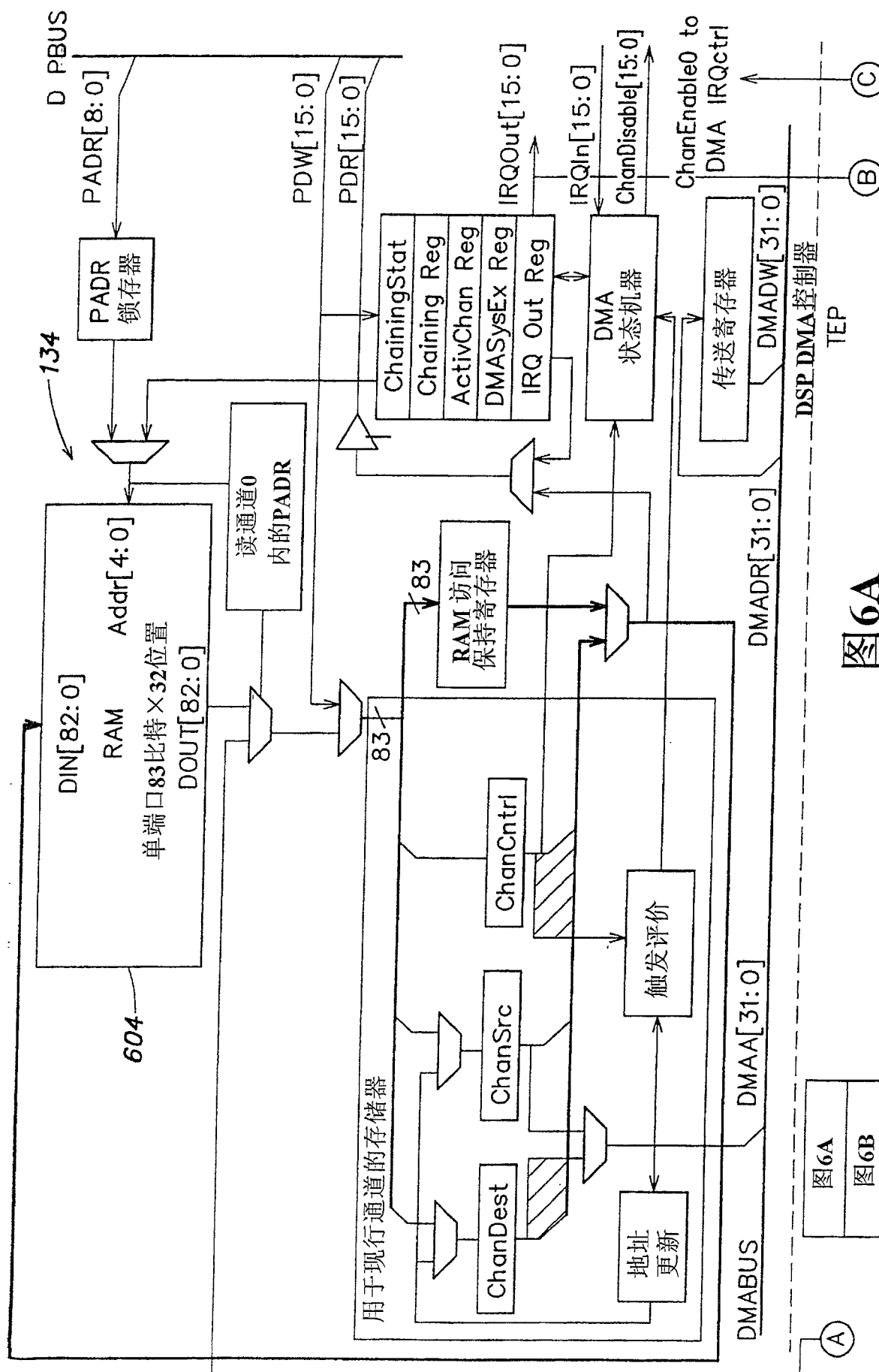


图6A

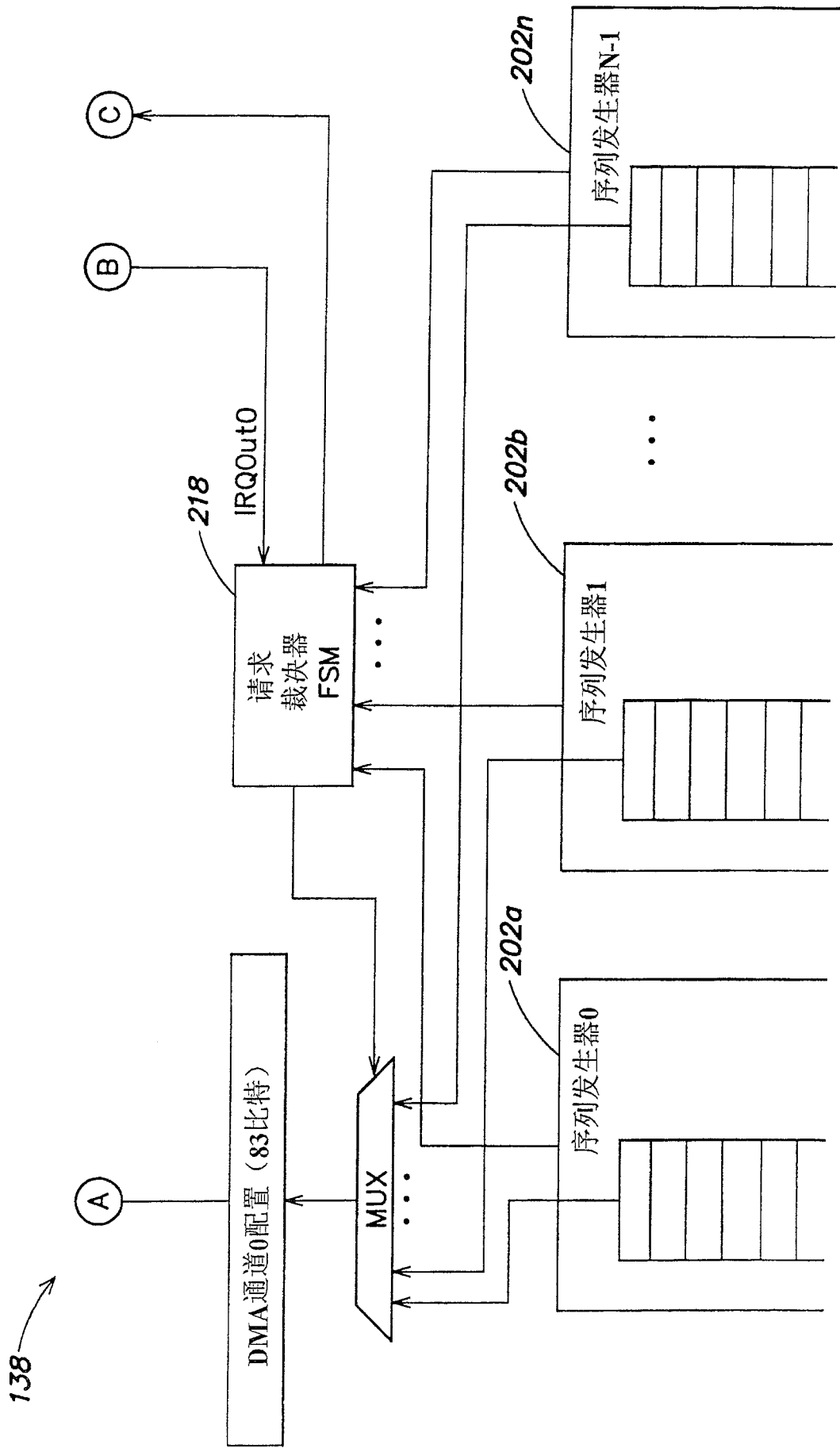


图6B

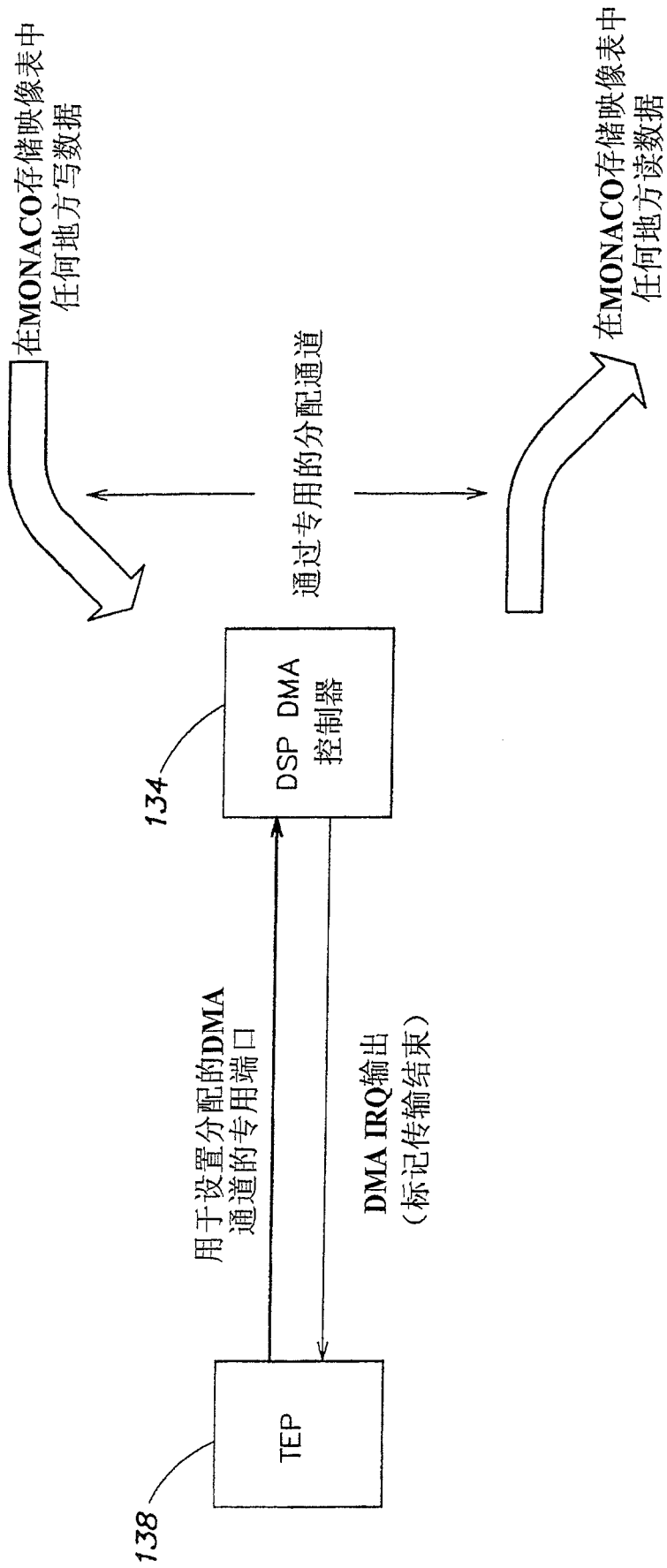


图7

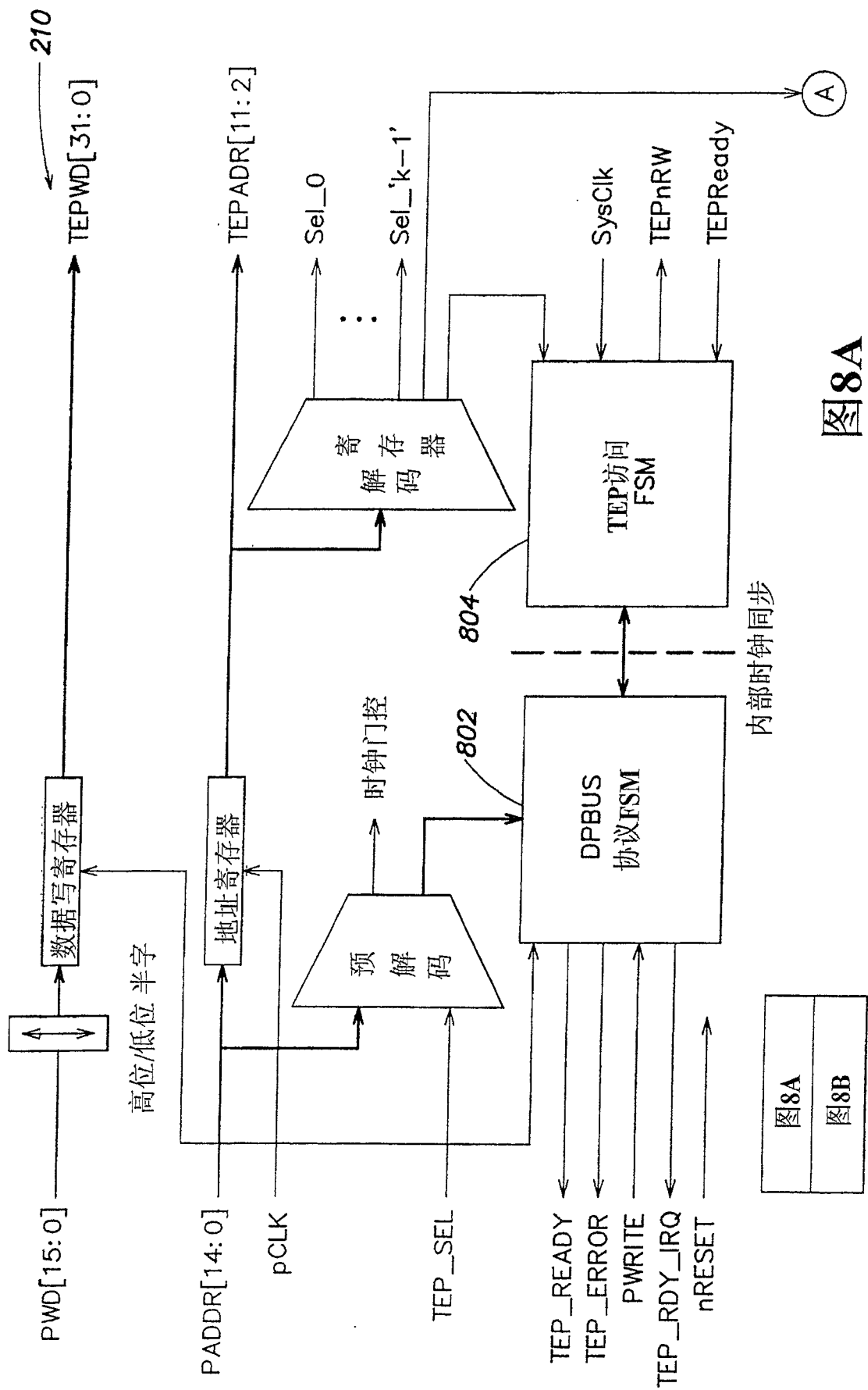


图8A

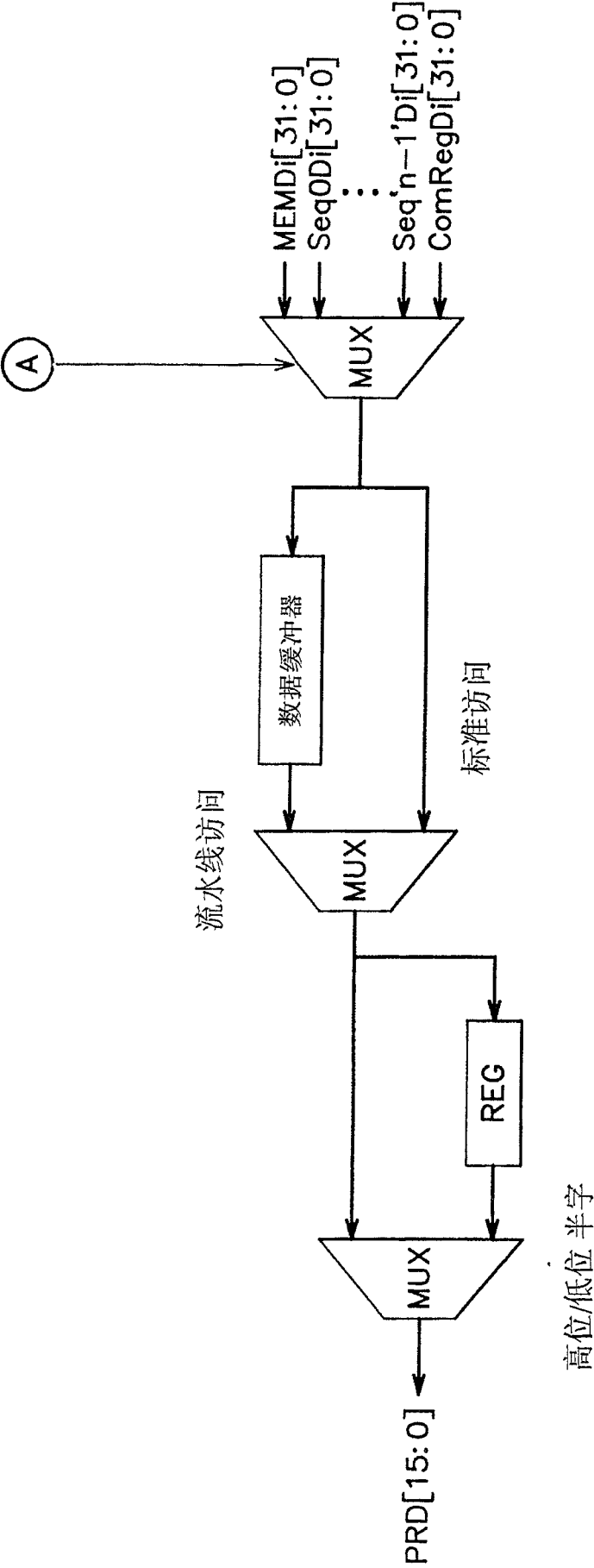
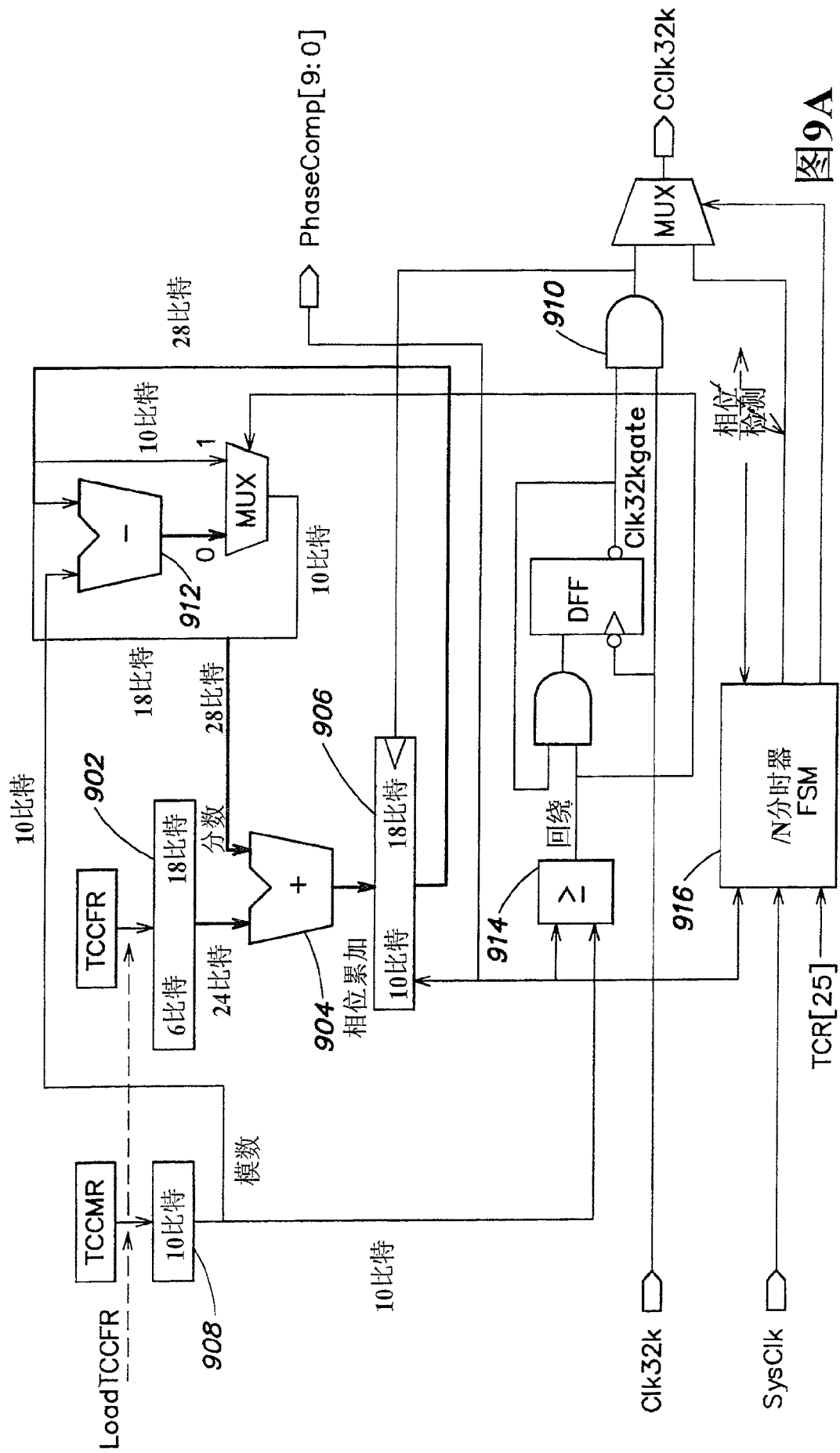


图8B



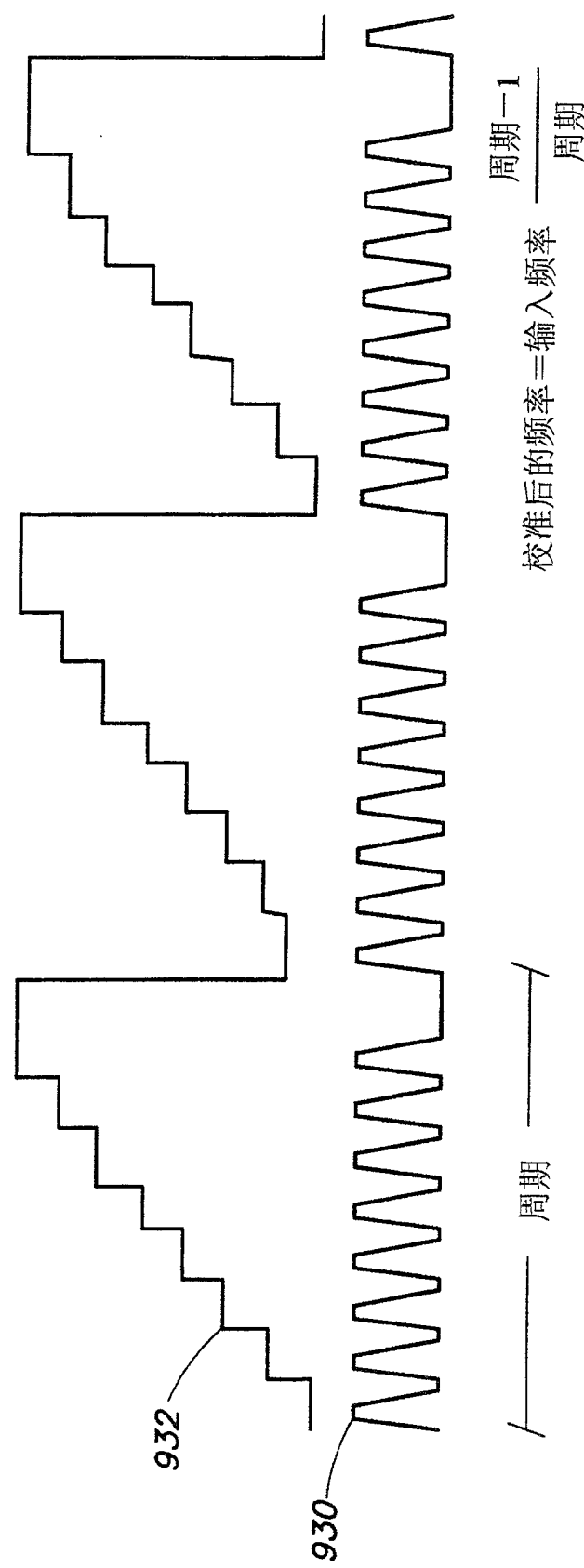


图9B

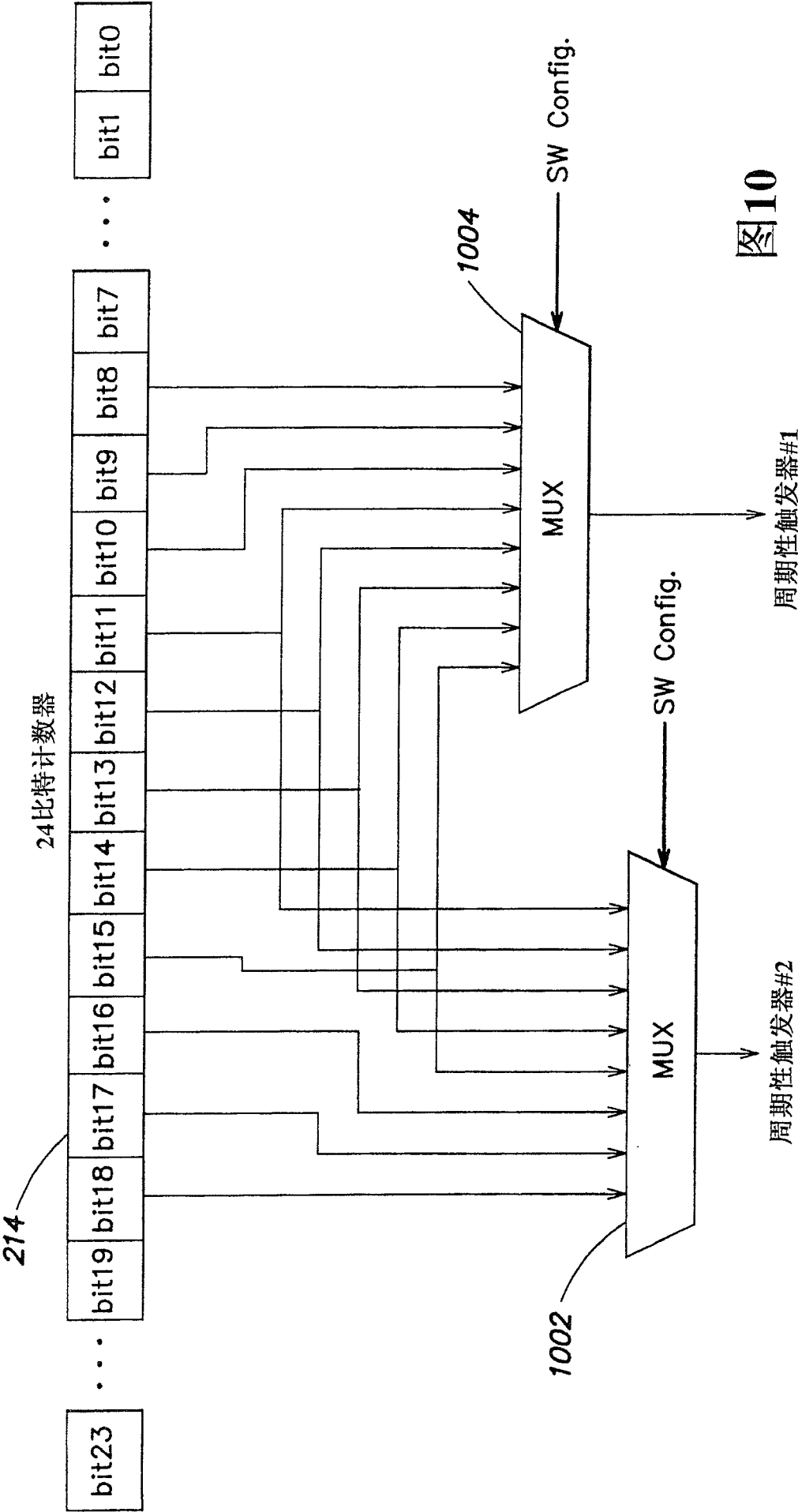


图10

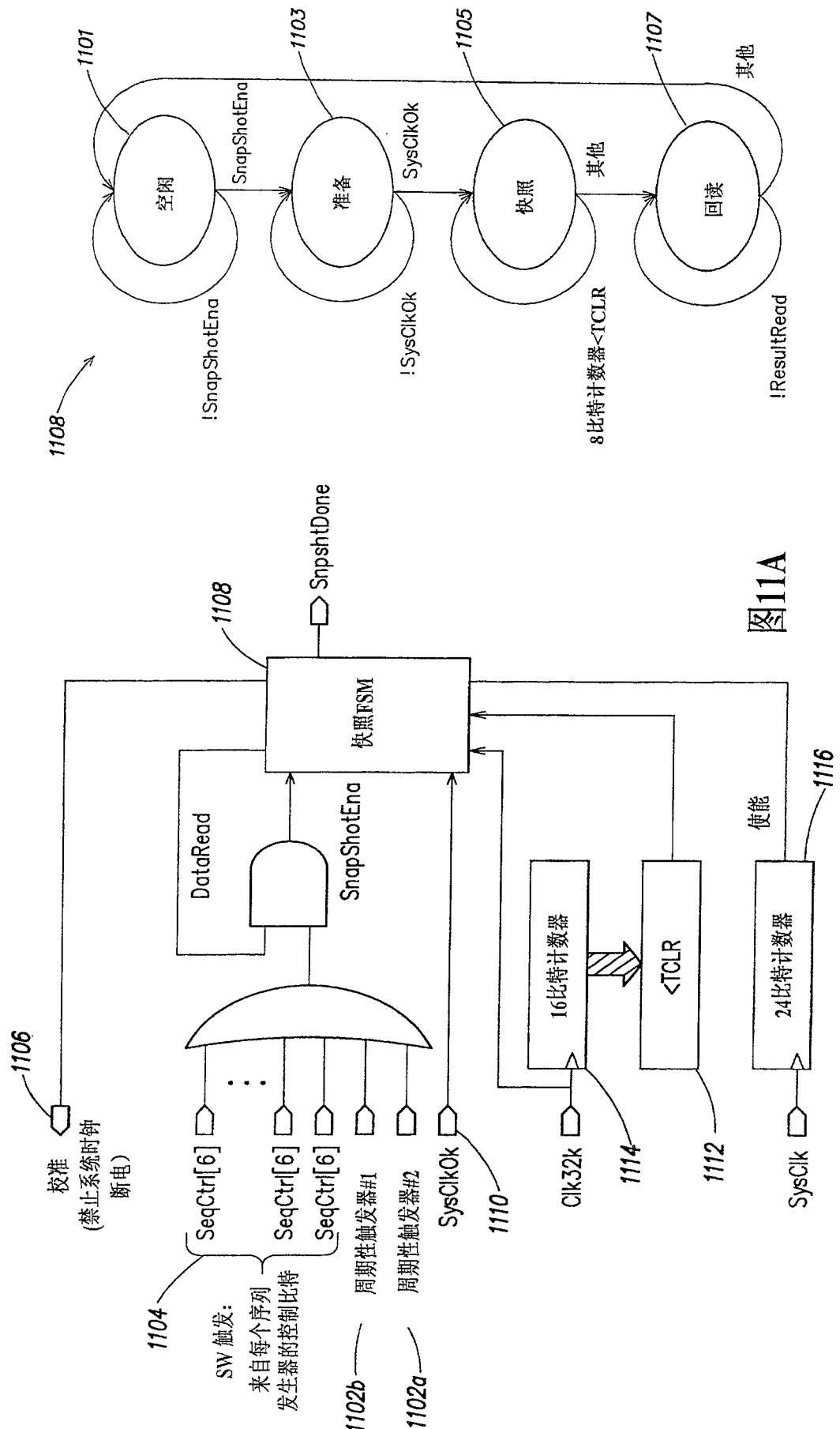


图11B

图11A

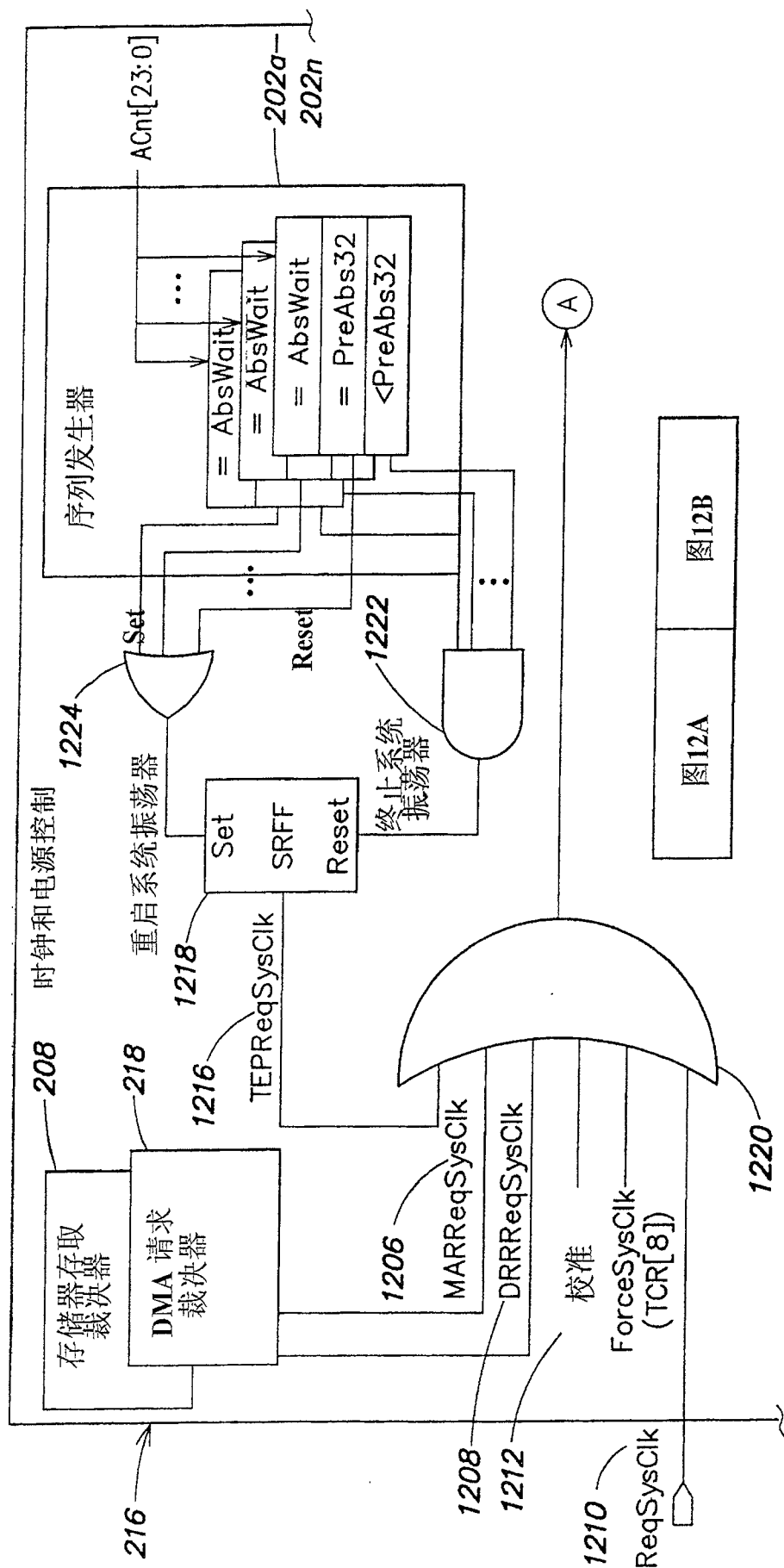


图12A

图12B

图12A

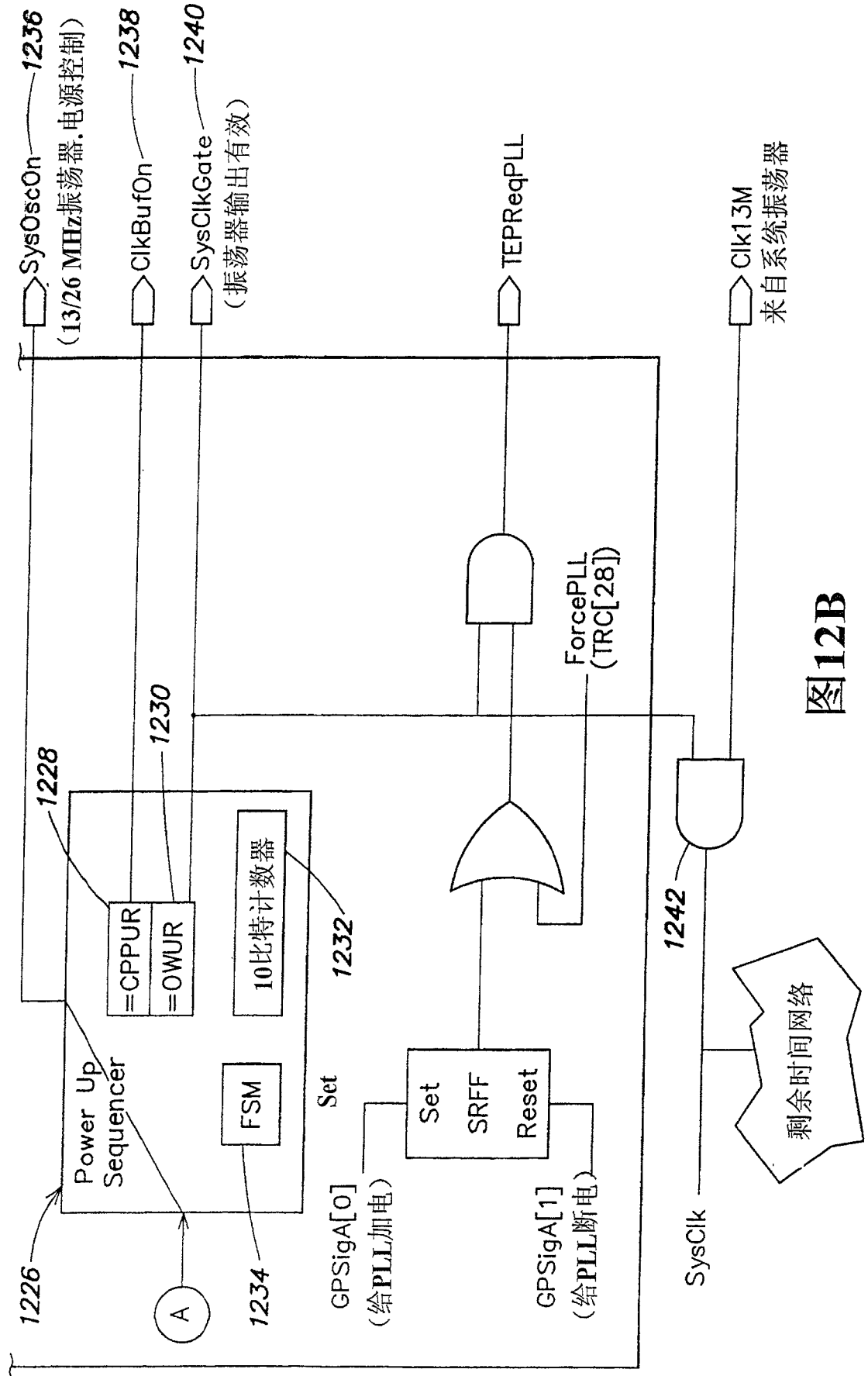


图12B

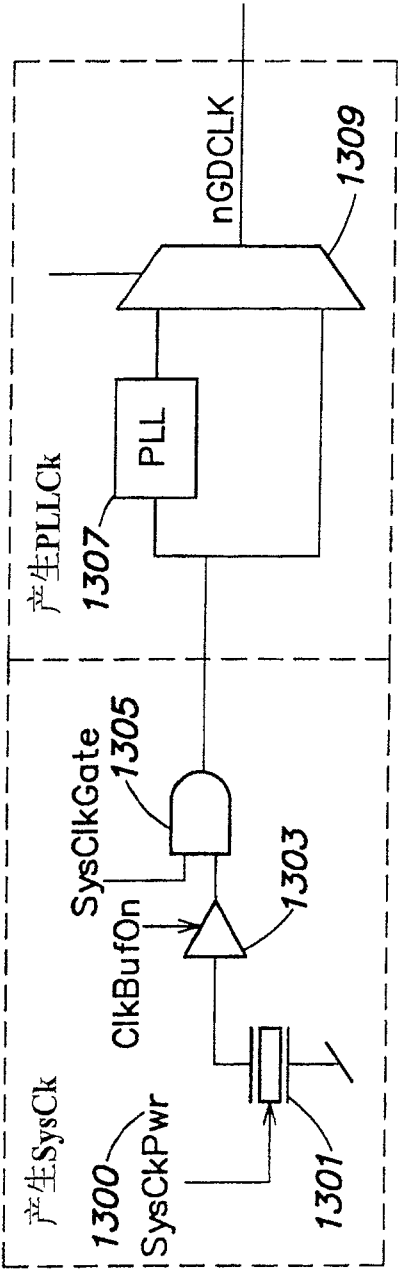


图13A

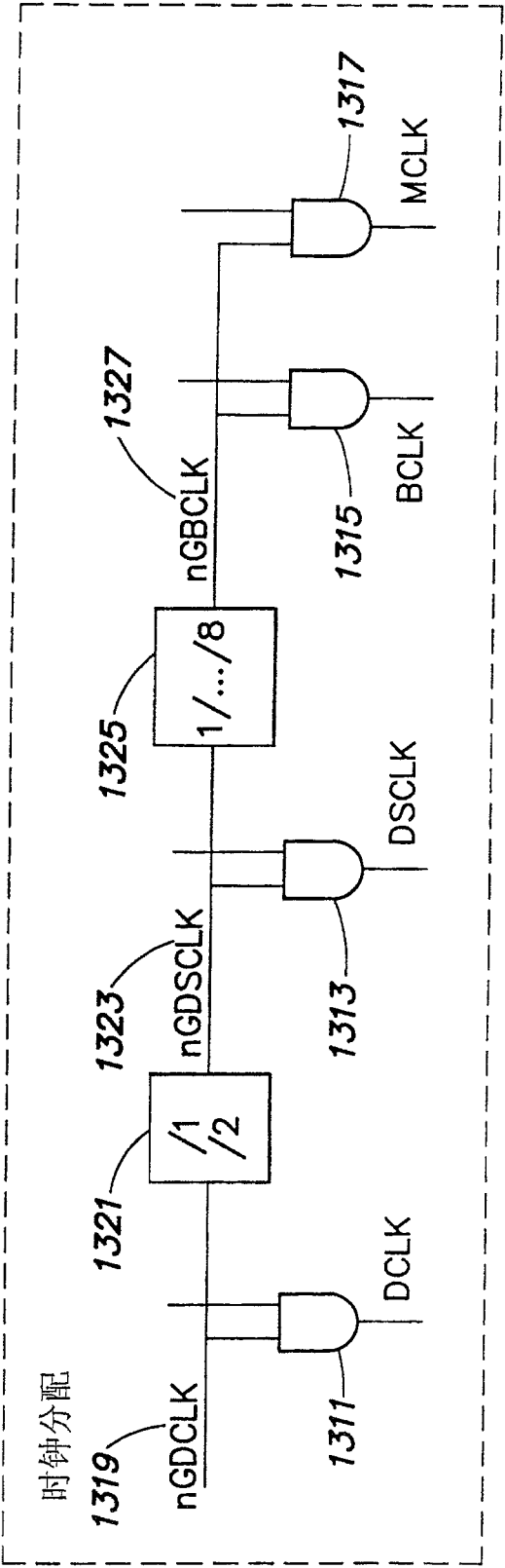


图13B

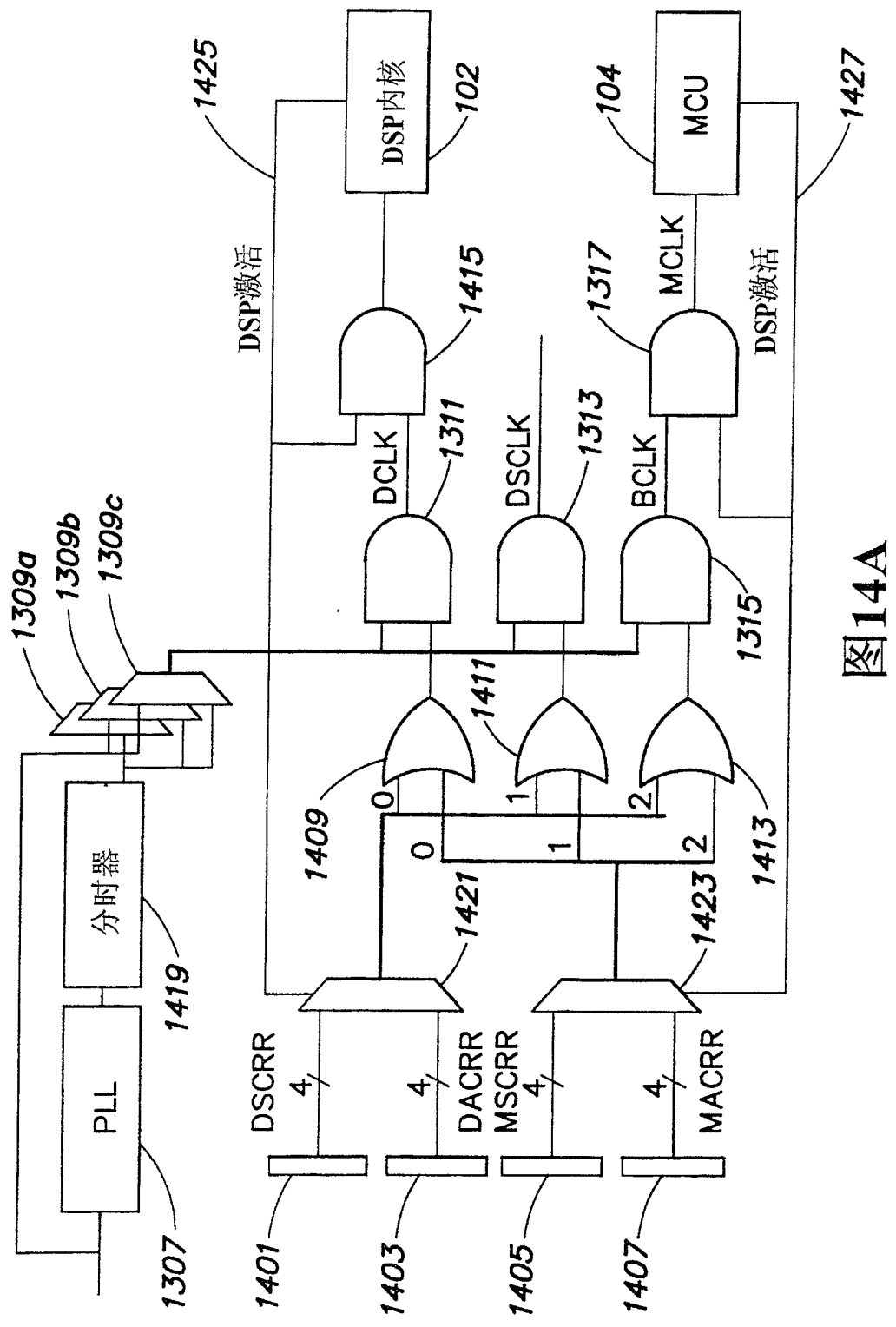


图14A

MCU休眠时钟需求寄存器 (MSCRR)

- [0] -DCLK, 1 = 在MCU休眠期间必须是激活的
- [1] -DSCLK, 1 = 在MCU休眠期间必须是激活的
- [2] -BCLK, 1 = 在MCU休眠期间必须是激活的
- [3] - 允许在休眠期间绕过PLL

MCU激活时钟需求寄存器 (MACRR)

- [0] -DCLK 1 = 在MCU激活期间是激活的
- [1] -DSCLK 1 = 在MCU激活期间必须是激活的
- [2] -BCLK 该比特在该寄存器中强制为1
- [3] - 允许绕过PLL。该比特在该寄存器中强制为0

DSP休眠时钟需求寄存器 (DSCRR)

- [0] -DCLK, 1 = 在DSP休眠期间必须是激活的
- [1] -DSCLK, 1 = 在DSP休眠期间必须是激活的
- [2] -BCLK, 1 = 在DSP休眠期间必须是激活的
- [3] - 允许在休眠期间绕过PLL

DSP激活时钟需求寄存器 (DACRR)

- [0] -DCLK 该比特在该寄存器中强制为1
- [1] -DSCLK 该比特在该寄存器中强制为1
- [2] -BCLK 该比特在该寄存器中强制为1
- [3] - 允许绕过PLL。该比特在该寄存器中强制为0

图14B

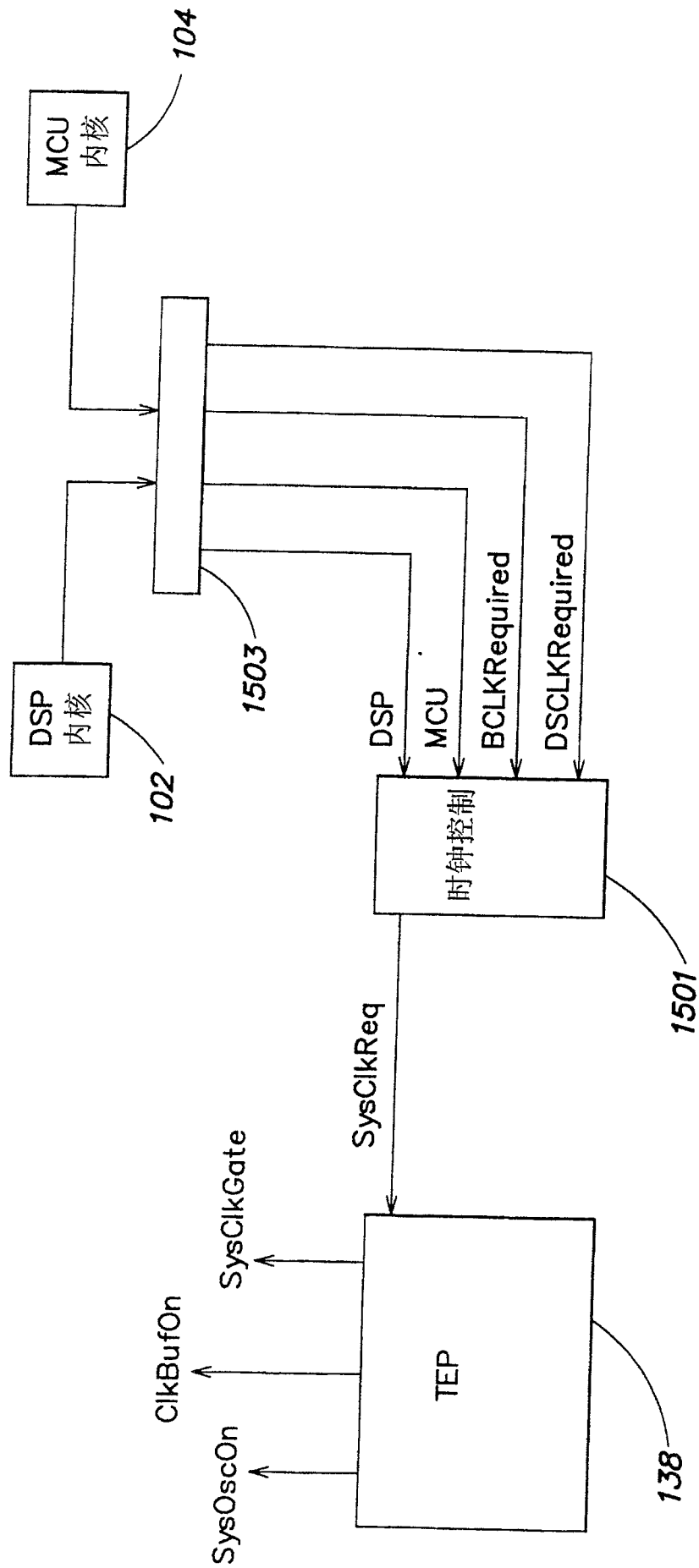


图15