



[12] 发明专利申请公开说明书

[21]申请号 93114719.0

[51]Int.Cl⁵

H04L 29/04

[43]公开日 1994 年 8 月 31 日

[22]申请日 93.11.25

[30]优先权

[32]92.11.28[33]GB[31]9224974.7

[71]申请人 莫托罗拉有限公司

地址 英国贝辛斯托克

[72]发明人 梅易克·卡帕迪阿 格瑞汉姆·琼森
巴瑞·M·肯

[74]专利代理机构 中国国际贸易促进委员会专利商
标事务所

代理人 杨国旭

H04Q 11/04

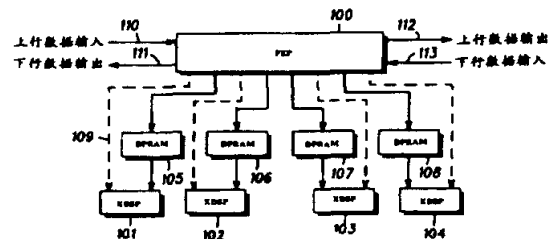
说明书页数:

附图页数:

[54]发明名称 译码器

[57]摘要

本发明的译码器，包含一个用于输入和输出编码话音数据的第一接口，用于输入和输出解码话音数据的第二接口；至少一个对至少 2 个话音数据信道上的话音数据进行译码操作的译码器处理器；以及与第一、第二接口和至少一个译码器处理器连接的、用以控制所述接口和至少一个译码器处理器之间的数据流的输入/输出处理器。这个输入/输出处理器控制至少一个译码器处理器，使它在给定时间对一个选定的信道实行译码操作。



权 利 要 求 书

1. 对安置在多个话音数据信道上的话音数据进行译码操作的一个译码器,其特征在于它包含:

一个用于输入和输出编码话音数据的第一接口(10);

一个用于输入和输出解码话音数据的第二接口(11);

至少一个对设置在至少 2 个话音数据信道上的话音数据进行译码操作的译码器处理器(101—104);

一个输入/输出处理器(100),该处理器连接到第一、第二接口和至少一个译码器处理器,用于控制第一、第二接口和至少一个译码器处理器之间的数据流,从而控制此至少一个的译码器处理器在给定时间为一个选定的信道实行译码操作。

2. 根据权利要求 1 的一个译码器,它包含与至少一个译码器处理器连接的存储器装置(105—108),用来存储此至少一个的译码器处理器和输入/输出处理器之间的缓冲数据。

3. 根据权利要求 1 和 2 的一个译码器,它还包含中断联接装置(109),用于由输入/输出处理器去中断至少一个译码器处理的操作。

4. 根据以上权利要求的任一条的一个译码器,其中译码器处理器设置成用来对安置在至少 2 个信道上的数据执行编码和解码

操作。

5. 根据上述权利要求任一条的一个译码器,它包括多个译码器处理器,其中输入/输出处理器设置成用来循环地控制多个译码器处理器的译码任务。

6. 根据以上权利要求的任一条的一个译码器,它包含多个译码器处理器,用来在至少 8 个信道上在 *CEPT* 标准话音数据和 *GSM* 标准话音数据之间进行译码。

7. 根据以上权利要求的任一条的一个译码器,其中的至少一个译码器处理器是一个相对高速的处理器而输入/输出处理器则是相对低速的处理器。

8. 在多个话音数据信道上进行编码和解码操作的一个译码器,其特征在于它包含:

一个第一处理器(100),用来控制信道的输入和输出数据流;

多个其它的处理器(101—104),被连接到第一处理器,用来接收从第一处理器发出的数据和中断以及将数据传给前端处理器;还有

存储器装置(105—108),分别与所述多个其它处理器中各相应处理器连接,用来存放第一处理器和所述多个其它处理器之间的缓冲数据,因而所述多个其它处理器中的每一个处理器均由第一处理器控制,对一个以上信道实行编码和/或解码操作。

说 明 书

译码器

本发明是有关执行译码操作的译码器,亦即对在多个话音数据信道上的 话音数据进行编码和解码操作的译码器。本发明特别用于 *CEPT* 标准话音数据和 *GSM* 标准话音数据间的译码。

图 1 表示摩托罗拉(*Motorola*)*GSM* 蜂窝式无线基站的一个译码器电路板,此电路板支持 30 个业务信道,在蜂窝式无线通信系统一侧上的一个时分多路通道和在电话交换机一侧上的一个 2.048Mbit/s 串行 *CEPT* 链路之间提供接口,*TDM* 接口 10 和 *CEPT* 接口 11 分别为在蜂窝式无线通信系统一侧上和交换机一侧上编码和解码的话音提供输入和输出。每一个业务信道使用一个 40MIP/80MHz 的标号为 *XCDR1—XCDR30* 的译码器来实现。在图左边的 *TDM* 通道 12 载有压缩话音信息 *GSM*(对于一个数据呼叫来说,为额定传输率的数据),图右边的 *CEPT* 链路 13 载有未压缩语言信息(对于一个数据呼叫来说,为额定传输率的数据)。*TDM* 到 *CEPT* 的方向称做上行链路,由 *CEPT* 到 *TDM* 则称做下行链路。

图 2 表示图 1 中译码器之一(如 *XCDR1*)的结构的简要的第一层分解。圆圈 20、21、22 和 23 代表四种功能,它们支持单个业务信道要求的译码,一个 *SIO* 中断 20 执行 *GSM08.60* 中所规定的功

能并由每隔 125 微秒的中断信号启动,上行链路圆圈 22 代表执行在 GSM06.10 中规定的 GSM 语言解码所需的所求处理和按 GSM06.31 规定的上行链路 GSM 不连续传输功能,它在数据呼叫时也执行如同 GSM08.20 规定的传输率适配的功能,下行链路圆圈 21 代表执行在 GSM06.10 中规定的 GSM 语言编码所需的所有处理和按 GSM06.31 规定的下行链路 GSM 不连续传输功能,它在数据呼叫时,也执行如同 GSM08.20 规定的传输率适配的功能,核心程序圆圈 23 包含按 GSM06.10, GSM08.60、GSM03.05 和 GSM06.31 中所规定的顺序和对于数据呼叫按 GSM08.20、GSM08.60 和 GSM03.05 所规定的顺序来调度上行链路和下行链路任务的例行程序。这个核心程序以 SIO20 收到的每 125 微秒中断为基准安排时间,并配置了一个 2—1 的下行链路—上行链路时间分片机制。这样它提供二个相继的 125 微秒时隙给下行链路处理之后提供一个时隙给上行链路处理。

其缺点是 SIO20 减少了留给另三个功能圆圈的时间总量使得每个业务信道都要求一个 DSP,这要求一块独立的译码器电路板上要有 30 个 DSP 来支持 30 个信道,因此,使设备非常昂贵。

2—1 时间分片机制有二个主要缺点,第一,一个子信道可以得到执行任务的机会,尽管这个子信道并没有要完成的任务。第二,如果一个子信道的任务在分配给它的时隙終了前完成,余下的时间没有其它任务可执行,成为空闲被浪费了。固定的时间分割器禁止一

个子信道任务在另一个子信道时间片周期内执行。

这就需要有一个改进的译码器。

按照本发明所提供的译码器在多个话音数据信道上实行话音数据的译码操作,它包含一个输入输出编码话音数据的第一接口;一个输入输出解码话音数据的第二接口;至少一个译码器处理器,用以对安排在至二个话音数据信道上的话音数据实行译码操作;一个与第一、第二接口以及至少一个译码器处理器连接的输入/输出处理器,用来控制第一、第二接口和至少一个译码器处理器之间的数据流。从而输入/输出处理器控制至少一个译码器处理器,以使在给定的时间和在选定的信道上实行译码操作。

按照本发明的一个译码器提供了一种更为高效的结构,在这种结构中可能有多个(例如四个)译码器处理器,但至少一个译码器处理器由输入/输出处理器控制,为一个以上的信道实行译码操作。

发现可容易地用 5 个 *DSP* 为 8 个信道实行译码操作,这表示所要求的处理器数减少了 $3/8$ 。使用此新的优良结构,图 1 中的 30 个 *DSP* 可代之以 20 个 *DSP*。

本说明中所引用的缩略词意义如下:

CEPT 邮电通信欧洲委员会

DSP 数据信号处理器

FEP 前端处理器

GSM 分组专用移动通信,亦即泛欧数字蜂窝式标准。

TDM 时分多路

SIO 串行输入/输出

XCDR 译码器

XDSP 译码器数字信号处理器

在附图中：

图 1 表示一个应用先有技术的译码器电路板的方框图；

图 2 表示图 1 中单个译码器处理器(*XCDR DSP*)处理功能的数据流的示意图；

图 3 表示按照本发明的译码器电路的方框图；

图 4 表示数据流示意图,它解释图 3 的 *FEP* 和 *DSP* 的最佳运行情况；

图 5 表示图 4 中 *SIO* 中断程序的优选运行的一个流程图；以及

图 6 表示图 3 每个译码器处理器(*XDSP*)的优选运行的数据流示意图；

参照图 3,所示的译码电路包含一个前端处理器(*FEP*)100 和四个译码器数据信号处理器(*DSP*) 101、102、103 与 104,在 *FEP*100 和每个 *XDSP* 之间是双端口随机存取存储器(*DPRAM*)105、106、107 与 108,分别与 *XDSP*101、102、103、104 相连,在这个方框图中,粗实线表示数据的流,而破折线则表示 *FEP*100 对 *XDSP*101—104 的控制。

全部四个 *XDSP*100—104 均为 40MIP/80MHz 的 *DSP*, 诸如

摩托罗拉的 56000 系列,每个 *DPRAM*105—108 之容量是 $2K \times 8$ 比特。

*FEP*100 接到一个 *TDM* 接口(诸如图 1 的接口 10)和一个 *CEPT* 接口(诸如图 1 的接口 11)。*FEP*100 为全部受支撑的 8 个信道执行上行链路和下行链路的输入与输出。破折线 109 代表沿所示方向 发出中断,即是从 *FEP*100 到每个 *XDSP*101—104。图示的其余的交互作用包括与 *DPRAM*105—108 的交互作用结合图 4 进行说明。

参照图 4,图中示出了图 3 的 *DPRAM*105—108 而所示的圆圈 200 代表由 *FEP*100 执行的那些功能。这些都将在后面结合图 5,予以更详细说明。

在图 4 中,所示的数据流线 110 和 111 表示与图 3 中相同的数据流线相应的下行数据输出和上行数据输入,而所示的数据流线 112 和 113 则表示 与图 3 中相同的数据流线相应的下行数据输入和上行数据输出。

SIO 中断程序 200 为使用 4 个 *XDSP*101—104 的 8 个业务信道实行串行输入和输出。在输入数据块能进行处理时,*SIO* 中断程序 200 仅中断处理器 101—104。在下行链路方向上,通过 *FEP* 和 *DPRAM* 之间的数据块是 160 字节的未经压缩的语言,在存储器和 *FEP* 及 *DSP* 处理器之间的另一些数据块是可变的,与先前的装置相反,在那里每逢 *SIO* 中断就执行一次中断程序。现在每个 *XD-*

SP101—104 各为二个业务信道执行上行链路,下行链路和核心的任务。

FEP SIO200 有二个主要功能要完成,它在每次中断时必须把 *TDM/CEPT* 数据转换到图 3 的合适的 *DPRAM* 或转换来自后者的 *TDM/CEP* 数据,同时它必须支持 8 个信道(与先前技术中仅支持一个信道相反),*FEP100* 完成这些功能是借助于图 5 所示的内外二个嵌套的流程环。

参照图 5,图中示出一个连续串行输入输出中断例行子程序,它组成图 4 中的数据流圆圈 200。

图 5 中从程序 500 开始,程序等待一个来自 *TDM* 接口 10 或来自 *CEPF* 接口 11 的中断,一旦收到一个中断(流程 501),程序就进到流程 502,计数器 *N* 置 1。此计数器表示信道号数,程序现在为 8 个信道依次从第 1 个到第 8 个开始译码,流程 502 以后,调用上行链路输入任务(流程 503),也就是从上行链路数据输入线 110 读入数据,假如流程 504 判定有足够的数据来进行完整数据块的译码,流程 505 中就生成一个中断,它(在 $N=1$ 或 2 时)被送到 *XDSP101*,*GSM08.60* 提供在流程 505 中生成一个中断之前所需数据块的大小的说明,在 $N=3$ 或 4 时,中断被送到 *XDSP102* 等等一直到 $N=5$ 到 8。如果数据不足,程序从流程 504 进到流程 506,调用下行链路输入任务,也就是从流线 113 读入数据。如果处理需要的数据是足够的(流程 507),即 160 字节的未压缩语言,则再次发出一个中断到

XDSP101(流程 508) , 同样,程序通过流程 509 到 512,其中上行链路数据在流线 112 上输出而下行链路数据则在流线 111 上输出。最后程序进到流程 520,其中计数器 N 加 1,准备处理信道 2,程序从流程 521 循环到 503,对信道 2 重复流程 503 到 512,生成的中断再次送到 *XDSP101*。

当计数器再加 1 时,流程 503 到 512 输入数据到 *DPRAM106* 并从 *DPRAM106* 输出数据,生成中断送到 *XDSP102*。同样,在 $N=4$ 或 5 时,数据读入 *DPRAM107* 或从它输出,并生成中断送到 *XDSP103*,而在 $N=7$ 或 8 时数据向 *DPRAM108* 读入或从它输出,且生成的中断送到 *XDSP104*。

最后,在 $N=8$ 时,程序回到流程 500,等待下一个中断,从而重复整个程序。

在这种方式中,数据块建立在各存储器 105—108 中,当执行译码器任务所需的数据充足时,这个任务才在相应的译码器 *XDSP101*—104 中被启动。同样,在当译码器已经完成其任务并且将一帧完整的数据块送到存储器 105—108 之一时,这些数据才可以在流线 111 或 112 上输出。

现在来看图 6,这是一个全面的流程意图,用来说明译码器 *XDSP101* 的操作。这个译码器为信道 1 和信道 2 执行话音编码(下行链路)和解码(上行链路)操作,图中“1”和“2”代表相应的信道,图 6 中的核心程序 600 控制全部任务管理,接收图 5 的来自流程

505、508 的中断,四个处理圆圈 601—604 分别代表信道 1 和 2 的上行链路处理以及信道 1 和 2 的下行链路处理。圆圈 601 包含一个如同在现有技术中的话音解码程序,而圆圈 603 则包含一个如同现有技术中的话音编码程序。圆圈 602 和 604 包含与圆圈 601 和 601 相同的程序,不过是在代表不同信道中的不同数据块进行操作。

SIO 中断程序 200 安排上行链路和下行链路任务的顺序,与 *GSM06.10*、*GSM03.05*、*GSM06.31* 以及 *GSM08.60* 中所列相同。

在上述安排中,没有固定的时分片进程以阻止一个子信道的任务在另外的子信道的时间分片周期中执行,这是由于,没有设置上行链路和下行链路的时间片。要在 *SIO* 中断程序 200 判定是否有足够数据可用于安排来自每一个子信道的下一个任务。对于不需要更多的数据来安排任一给定子信道的下一个任务的情况与对于在同一时间来自二个或更多子信道的一些任务已经就绪等待执行的情况,用同样的方法来处理。核心程序被设计成一次只允许执行一子信道的一个任务,然后由这个核心程序决定序列中的下一个子信道是否有任务要执行,并加以执行。图 2 中先有技术的核心程序 23 由监视指针来决定等待执行的子信道任务数,这些指针由 *SIO* 保持,它指出上行链路与下行链路输入数据的帧位置,但在新设计中,核心程序包含一个中断处理例行程序,它的责任是使用来自前端处理器的中断中的信息以及已对当前帧和子信道执行的任务数的了解,计算出尚

未对每个子信道的当前帧执行的任务数。

模拟结果已表明,在此情况下有可能使用 5 个数字信号处理器(*DSP*)来支持 8 个业务信道的译码,而不是用 8 个 *DSP*。从模拟中也可推测没有可能用一单个 *DSP* 来支持一个以上信道的译码。

本装置具有的好处是可少使用一些昂贵的数字信号处理器。可用运行速度较慢的处理器(费用少)作前端处理器 100 而采用速度较快的一些处理器(费用多些)作译码器 *DSP*101—104,这可以更节省费用。另外,因此而富余的处理能力还可用于更复杂的译码器处理器,诸如即将出现的 *GSM* 半速率编码译码器,还能利用。

一个在技术上训练有素者,很容易对此作出种种修改,例如给 *XDSP*101—104 之一分配二个信道,如能采用足够容量的 *DSP*,则可给每个 *DSP* 分配 3 个或 4 个信道。或者对某些特定的 *DSP* 信道分配可以是动态的,使得任一可用的 *DSP* 能在一给定时间执行任何所要求的任务,只要一个特定信道的所有必要信道参数对于分配到这个信道上的那个 *DSP* 来说是合适的。这样的配置将具有一定的耐久性,即使一个 *DSP* 发生故障也不会妨碍一个特定信道的使用。

还有一种配置方式是:选定的 *DSP*(例如 *XDSP*101 和 *XDSP*102)可设置成完成编码操作,而选定的另外一些 *DSP*(例如 *XDSP*s103 和 *XDSP*s104)可设置成完成解码操作。

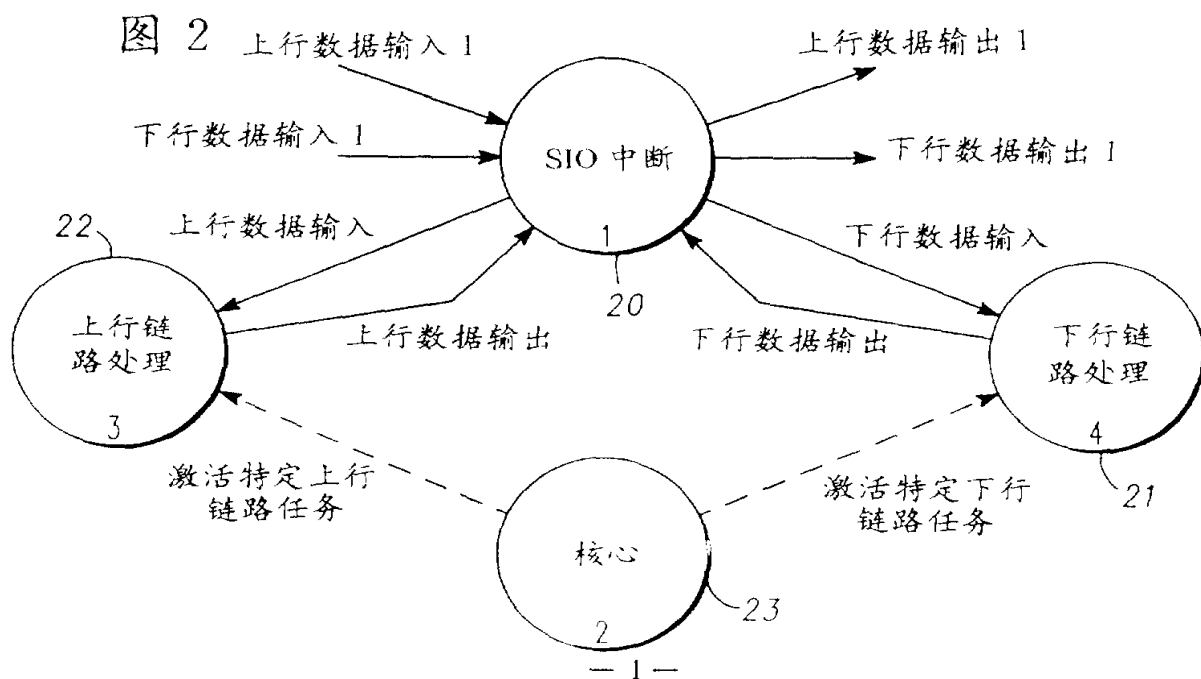
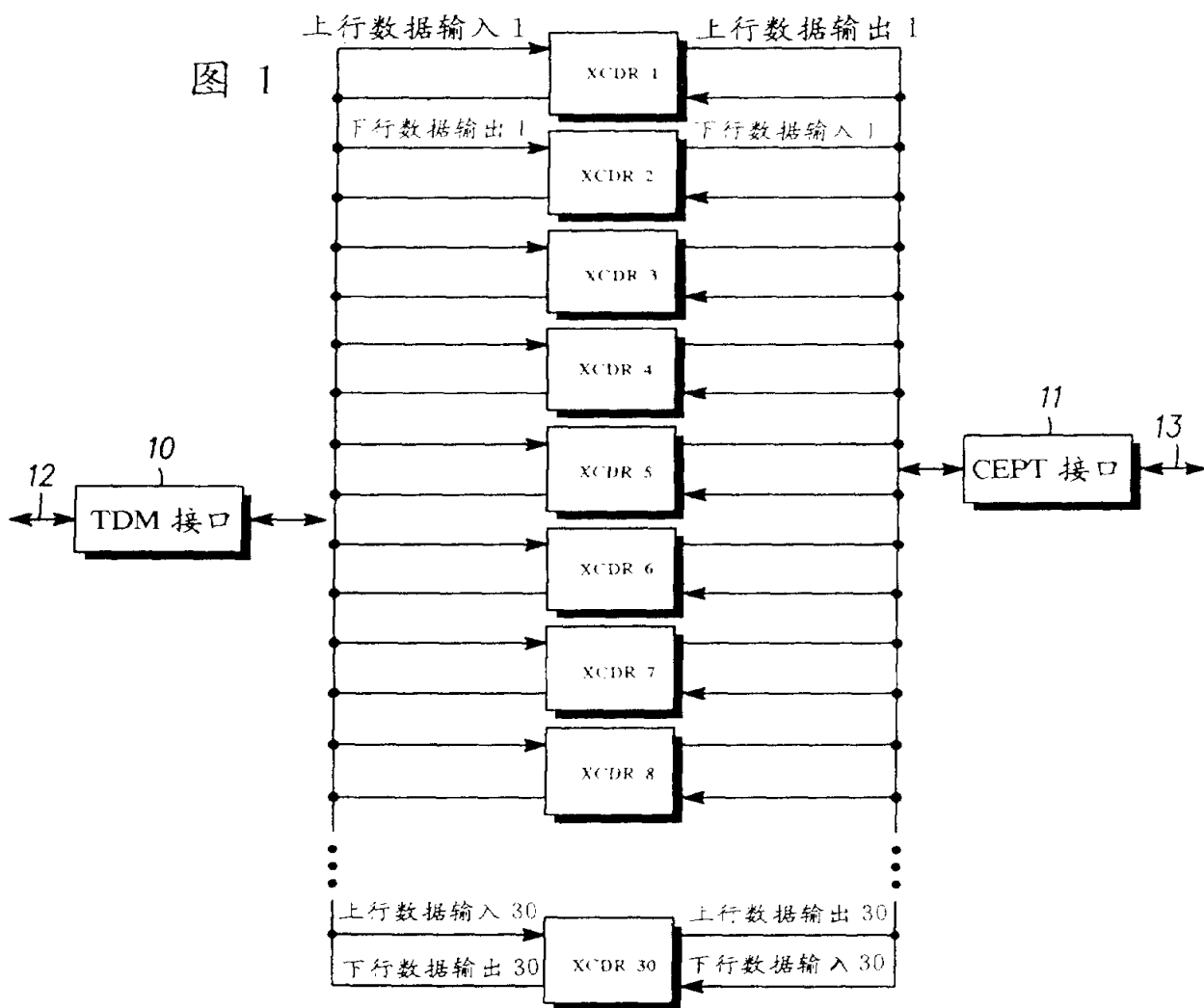


图 3

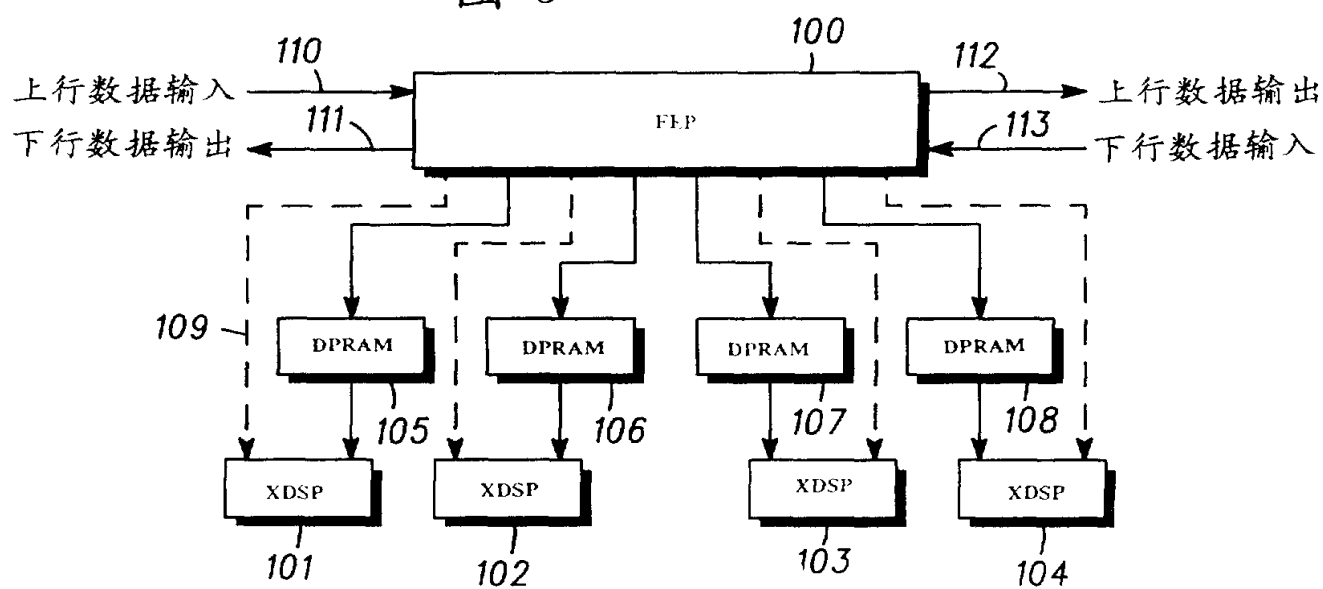


图 4

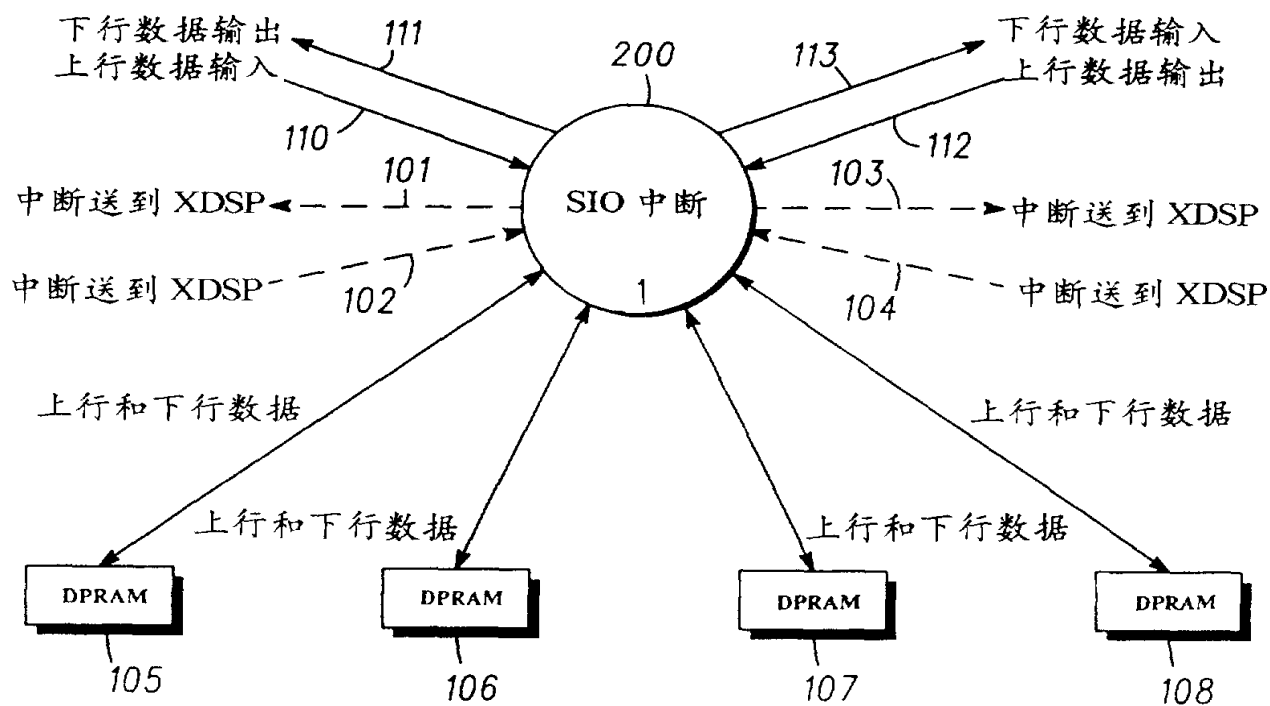


图 5

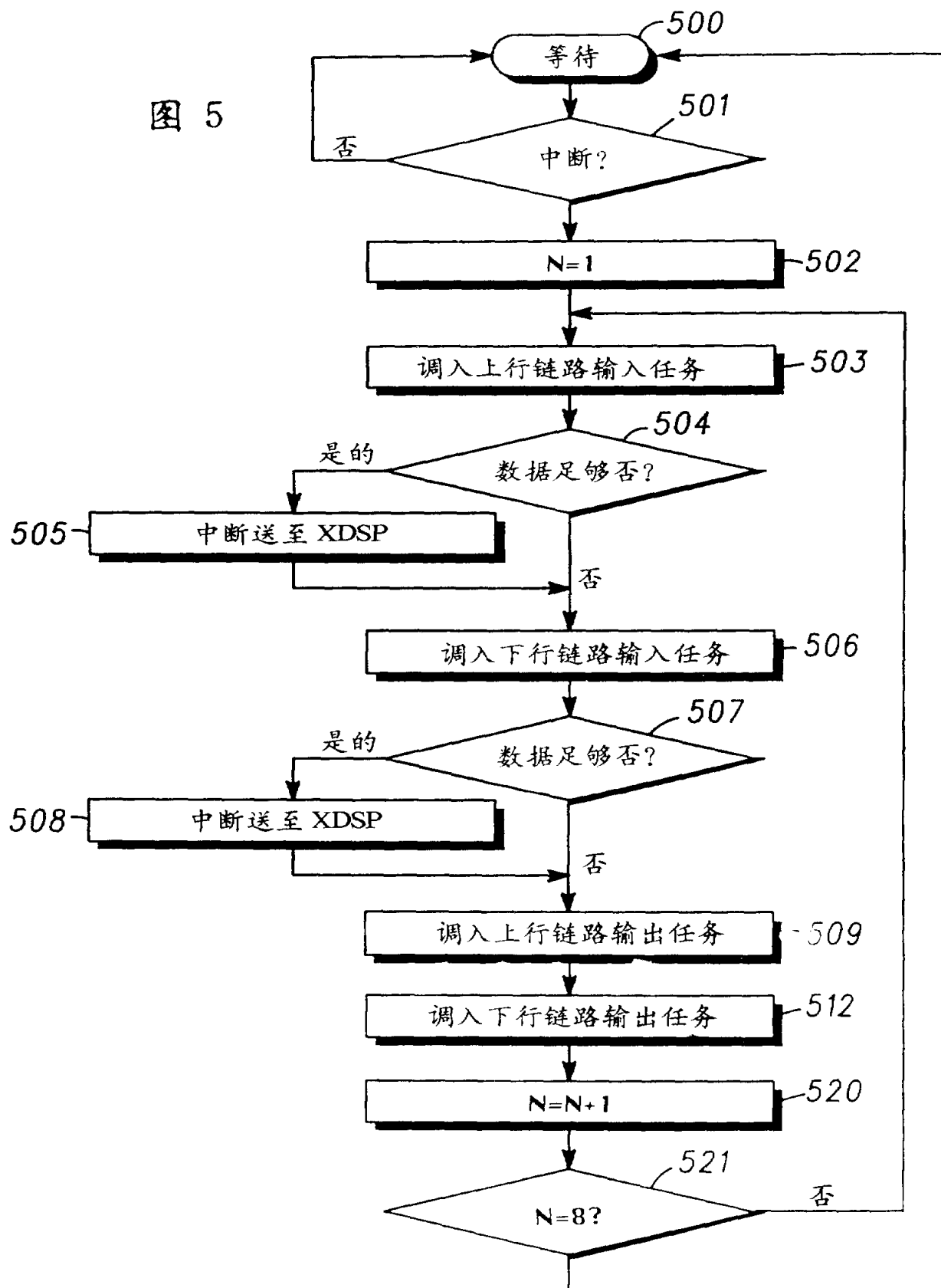


图 6

