



(12)发明专利

(10)授权公告号 CN 104702294 B

(45)授权公告日 2018.04.27

(21)申请号 201510136860.2

(22)申请日 2015.03.26

(65)同一申请的已公布的文献号

申请公布号 CN 104702294 A

(43)申请公布日 2015.06.10

(73)专利权人 北京思朗科技有限责任公司

地址 102412 北京市房山区阎村镇阎富路1
号11号楼4层402

(72)发明人 吴军宁 赵旭莹 吴义如 董佳佳
王晓琴

(74)专利代理机构 中科专利商标代理有限责任
公司 11021

代理人 宋焰琴

(51)Int.Cl.

H03M 13/27(2006.01)

(56)对比文件

CN 103873073 A,2014.06.18,

CN 101667451 A,2010.03.10,

CN 101237240 A,2008.08.06,

CN 101789845 A,2010.07.28,

CN 102201818 A,2011.09.28,

CN 101882933 A,2010.11.10,

CN 103905066 A,2014.07.02,

US 6956909 B2,2005.10.18,

JP 2000196471 A,2000.07.14,

审查员 梁旭姣

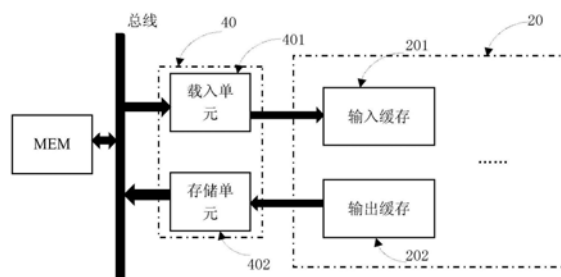
权利要求书1页 说明书5页 附图4页

(54)发明名称

一种Turbo译码器的位宽非对称仿存接口

(57)摘要

本发明提出一种通用的可配置的Turbo译码器接口逻辑电路,包括输入输出接口和缓存区,输入输出接口包括输入数据载入单元和输出数据存储单元,缓存区包括输入缓存和输出缓存,输入数据载入单元用于将待译码数据以滑窗为单位从一个存储器中读入到所述输入缓存中;输出数据存储单元用于在译码完成之后将输出缓存中的硬比特信息写入到存储器中。本发明的输入输出位宽可以根据CPU的数据总线位宽进行配置和修改,且根据不同的译码并行度P,可通过修改有限状态机的状态跳转条件来完成。本发明支持LTE协议中所规定的所有188种不同长度的码块,解决了CPU和译码器数据总线位宽不对称造成存储器读写问题,可广泛应用于通信基站以及移动终端设备中。



1. 一种Turbo译码器接口逻辑电路,包括输入输出接口(40)和缓存区(20),所述输入输出接口(40)包括输入数据载入单元(401)和输出数据存储单元(402),所述缓存区(20)包括输入缓存(201)和输出缓存(202),其中,

所述输出缓存(202)用于缓存硬比特信息;

所述输入数据载入单元(401)用于将待译码数据以滑窗为单位从一个存储器中读入到所述输入缓存(201)中;

所述输出数据存储单元(402)用于在译码完成之后将输出缓存(202)中的硬比特信息写入到所述存储器中;

其中,所述输入缓存(201)的写入端口的数据总线位宽与CPU数据总线位宽CPU_D_WIDTH一致,所述输入缓存(201)的读出端口的数据位宽则固定为8比特;

所述输出缓存(202)的写入端口的数据位宽固定为8比特,所述输出缓存(202)的读出端口的数据位宽与CPU总线位宽CPU_D_WIDTH一致。

2. 如权利要求1所述的Turbo译码器接口逻辑电路,其特征在于,所述输入缓存(201)的大小由译码算法采用的最大滑窗长度MAX_SW_LEN决定,所述输出缓存(202)的大小由LTE协议规定的最大码块长度MAX_CB_LEN决定。

3. 如权利要求2所述的Turbo译码器接口逻辑电路,其特征在于,将译码结果从所述输出缓存(202)读出所需的仿存次数为 $K/\text{CPU_D_WIDTH}$;码块长度K的取值范围为40~6144,其值由LTE物理层协议36.212中5.1.3.2.3表5.1.3-3所定义,共188种。

一种Turbo译码器的位宽非对称仿存接口

技术领域

[0001] 本发明涉及通信技术领域,特别是涉及一种用于通信系统中Turbo译码器的位宽非对称仿存接口。

背景技术

[0002] 随着半导体技术的发展,无线通信技术得到了广泛应用。但是,由于用户数据在传输过程中,会受到各种噪声以及信号的干扰,致使信号出错或者丢失。为了提高信号传输质量,增加无线通信系统的可靠性,通信系统需要具有很强的检错纠错的能力。

[0003] 在长期演进(Long Term Evolution,LTE)系统中,广泛采用的一种纠错码就是Turbo码。Turbo码是由Berrou等人于1993年在卷积码和级联码的基础上首次提出的,其典型特征就是编译码复杂度较高、时延大,但误码性能优异,适合于大数据量的长码块以及对时延要求不高的数据传输。并且,能够很好地满足香农信道编码理论中的随机性条件,在译码的过程采用了迭代译码的方式,能够获得逼近香农限的译码性能。

[0004] 由于译码算法的迭代特性,Turbo码具有出色译码性能的代价,就是较高的计算复杂度和存储资源消耗。因此,在移动终端设备和通信基站中,Turbo译码器大多是由专用集成电路(Application Specific Integrated Circuit,ASIC)来实现,以期减少由于计算部件流水线冲突所带来的译码延迟,提高译码器的吞吐率。

[0005] 通过专用集成电路来实现Turbo译码器的方式,首先要考虑的就是待译码的软比特数据和译码结果在存储设备与译码器之间如何传输。待译码的软比特数据,通常需要较大的存储空间,如LTE协议中所规定的最大码块长度为6144,其待译码的软比特数据就需要18KB大小的存储单元;而这些数据在译码的时候,都需要按译码算法一个或多个同时送入译码部件进行译码。对于译码结果来说,由于是符号判决之后的硬比特数据,一般都采用字节压缩的形式先暂存在输出缓存中,译码结束之后再输出到存储单元。

[0006] 通常地,对于以协处理器方式实现的译码器来说,这里的输入输出工作都是由直接存取存储器(Direct Memory Access,DMA)控制器完成,这也是大多数现有产品的实现方式。主控制器只需要在译码的时候向DMA控制器发送启动指令,DMA控制器就会从指定的地址空间读取待译码数据,送到译码器中;在译码结束之后,DMA控制器会自动将译码结果写入指定的地址空间里,并向主控单元发出译码完成信号。另外的一种较为常见的实现方式,就是大多数公开论文中所采用的,直接在把存储单元和译码器连在一起,输入和输出数据都是直接存取的。

[0007] 图1是LTE系统中广泛采用的Turbo译码器的基本结构框图。其接收的待译码数据包括系统信息Systematic、校验1信息Parity1和校验2信息Parity2。首先,系统信息Systematic和校验1信息以及先验信息A-priori经过译码器1进行译码,获得外信息Extri.1。外信息Extri.1经过交织器完成交织操作,作为译码器2的先验信息A-priori,与交织后的系统信息Systematic和校验2信息Parity2,送入译码器2中,进行译码。获得外信息Extri.2,经过解交织器之后作为译码器1的先验信息;同时,LLR信息经过硬判决单元完

成符号判决。译码完成则输出译码结果；否则，两个译码器迭代进行译码。

[0008] 图2是一种通用的Turbo译码器实现结构框图主要包括三部分：输入输出存储单元10、子译码器20和外信息存储单元30。其中，输入输出存储单元10又包括输入的系统信息、校验1信息和校验2信息等译码数据存储单元，和输出存储单元。子译码器20包括输入缓存、分支度量计算、前/后向度量计算和LLR/外信息计算等功能部件。在译码开始的时候，需要把输入缓存中的数据以滑窗为单位送入输入缓存中进行译码，前半次迭代译码完成，生成的外信息送入外信息存储单元20；在进行后半次译码的时候，将交织后的系统信息和外信息以及校验2信息送入子译码器进行译码，整个译码过程如前所述。

[0009] 在实际工作和工程实践中，需要根据设计需求，把Turbo译码器作为与算术逻辑单元(Arithmetic Logical Unit, ALU)和乘累加计算单元(Multiply Accumulate, MAC)一样的功能部件放在中央处理单元(Central Processing Unit, CPU)内部，待译码数据和译码结果都是通过CPU的数据总线进行读写访问的。由于不同CPU实现方案，其数据总线的位宽不尽相同，而译码器的输入输出数据总线的位宽却是固定的，且二者之间呈现出一种不对称关系。因此就需要设计一种位宽非对称的仿存接口逻辑电路，完成待译码数据和译码结果的仿存操作。

发明内容

[0010] (一)要解决的技术问题

[0011] 本发明旨在解决Turbo译码器在实现时输入输出仿存接口与CPU数据总线位宽非对称造成读写时序复杂的问题，提出一种高效、通用的译码器输入输出接口电路设计方案。

[0012] (二)技术方案

[0013] 本发明提出一种Turbo译码器接口逻辑电路，包括输入输出接口和缓存区，所述输入输出接口包括输入数据载入单元和输出数据存储单元，所述缓存区包括输入缓存和输出缓存，其中，所述输出缓存用于缓存硬判决比特；所述输入数据载入单元用于将待译码数据以滑窗为单位从一个存储器中读入到所述输入缓存中；所述输出数据存储单元用于在译码完成之后将输出缓存中的硬比特信息写入到所述存储器中。

[0014] 根据本发明的具体实施方式，所述输入缓存的大小由译码算法采用的最大滑窗长度MAX_SW_LEN决定，所述输出缓存的大小由LTE协议规定的最大码块长度MAX_CB_LEN决定。

[0015] 根据本发明的具体实施方式，所述输入缓存的写入端口的数据总线位宽与CPU数据总线位宽CPU_D_WIDTH一致，所述输入缓存201的读出端口的数据位宽则固定为8比特。

[0016] 根据本发明的具体实施方式，所述输出缓存的写入端口的数据位宽固定为8比特，所述输出缓存的读出端口的数据位宽与CPU总线位宽CPU_D_WIDTH一致。

[0017] 根据本发明的具体实施方式，将译码结果从所述输出缓存读出所需的仿存次数为 $K/\text{CPU_D_WIDTH}$ ；码块长度K的取值范围为40~6144，其值由LTE物理层协议36.212中5.1.3.2.3表5.1.3-3所定义，共188种。

[0018] (三)有益效果

[0019] 本发明通过设计一个通用的接口逻辑电路来完成Turbo译码器待译码数据的读入和读出操作，解决了CPU和译码器总线位宽不对称的问题，具体来说具有以下有益之处：

[0020] 1) 接口逻辑电路的输入输出位宽可以根据CPU的数据总线位宽进行配置和修改。

- [0021] 2) 根据不同的译码并行度P,可通过修改有限状态机的状态跳转条件来完成。
- [0022] 3) 支持LTE协议中所规定的所有188种不同长度的码块。

附图说明

- [0023] 图1是应用在LTE系统中的Turbo译码器的基本结构图；
- [0024] 图2是通用的Turbo译码器实现框图；
- [0025] 图3是根据本发明的Turbo译码器接口逻辑电路的一种具体实施例结构图；
- [0026] 图4是根据本发明的Turbo译码器输入缓存结构框图；
- [0027] 图5是根据本发明的一种具体实施例中输入缓存地址分配；
- [0028] 图6是根据本发明的一种具体实施例中接口逻辑电路有限状态机跳转示意图；
- [0029] 图7是根据本发明的一种P路并行译码器接口逻辑电路结构示意图；
- [0030] 图8是根据本发明的一种具体实施例中输出接口逻辑实现框图。

具体实施方式

[0031] 本发明提出了一个通用的可配置的Turbo译码器接口逻辑电路,可以很好地解决CPU和译码器数据总线位宽不对称造成存储器读写问题,该实现方法和装置可以很容易地根据CPU位宽进行相应设计。

[0032] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本发明作进一步的详细说明。

[0033] 图3是根据本发明的Turbo译码器接口逻辑电路的一个具体实施例的结构图,该电路主要包括输入输出接口40和缓存区20,缓存区20包括输入缓存201和输出缓存202,输出缓存202用于缓存硬判决比特。输入输出接口40主要包括输入数据载入单元401和输出数据存储单元402。输入数据载入单元401主要完成将待译码数据以滑窗为单位经由数据总线从存储器(MEM)中读入到输入缓存201中,输出数据存储单元402则在译码完成之后将输出缓冲区分中的硬比特信息经由数据总线写入到存储器(MEM)中。

[0034] 根据本发明的一种具体实施方式,输入缓存201的大小由译码算法采用的最大滑窗长度MAX_SW_LEN决定,输出缓存202的大小由LTE协议规定的最大码块长度MAX_CB_LEN决定。

[0035] 根据本发明的一种具体实施方式,输入缓存201的写入端口,其数据总线位宽与CPU数据总线位宽CPU_D_WIDTH一致,读出端口的数据位宽则固定为8比特。输出缓存201的写入端口,其数据位宽根据字节压缩方案也固定为8比特,读出端口的数据位宽,则同样与CPU总线位宽CPU_D_WIDTH一样。

[0036] 根据本发明的一种具体实施方式,对于多路并行译码器的设计,包括P个输入缓存,译码器的输入接口逻辑电路需要在译码开始阶段将待译码数据写入每个子译码器的输入缓存中。填充完一个子译码器的输入缓存所需的仿存次数为 $\text{MAX_SW_LEN}/\text{CPU_D_WIDTH}$;填充完P个子译码器的输入缓存所需的仿存次数为 $P * \text{MAX_SW_LEN}/\text{CPU_D_WIDTH}$ 。

[0037] 根据本发明的一种具体实施方式,将译码结果从输出缓存读出所需的仿存次数由当前码块长度K决定,为 $K/\text{CPU_D_WIDTH}$ 。码块长度K的取值范围为40-6144,其值由LTE物理层协议36.212中5.1.3.2.3表5.1.3-3所定义,共188种。

[0038] 根据本发明的一种具体实施方式,Turbo译码一个完整迭代被分为交织前和交织后的两个半次迭代,前半次迭代译码需要读入系统信息SYS和校验1信息PAR1,后半次迭代译码需要读入校验2信息PAR2。这些待译码数据的读入过程由一个通用的有限状态机(Finite State Machine,FSM)完成。

[0039] 为了使本实施例更易于理解,现对前面所述的一些参数进行示例赋值:

$$[0040] \quad \begin{cases} MAX_SW_LEN = 128 \\ MAX_CB_LEN = 6144 \\ CPU_D_WIDTH = 512 \end{cases}$$

[0041] 且假设,每个待译码数据占1字节存储空间。

[0042] 则,每个滑窗最多需要 $MAX_SW_LEN/CPU_D_WIDTH=128*8/512=2$ 次仿存操作,才能完成系统信息或者校验信息的写入操作。

[0043] 图4是根据本发明的Turbo译码器输入缓存201的结构框图,根据Turbo译码器实现的硬件流水关系,前后两个滑窗的前向度量计算和后向度量计算时并行完成的,因此需要系统信息Systematic和校验信息Parity均需要两个滑窗缓存SYS_SW0/SYS_SW1和PAR_SW0/PAR_SW1。同时,由于每个滑窗缓存长度为128Byte,每次仿存操作可以写入 $CPU_D_WIDTH/8=64$ Byte大小的数据,因此将滑窗缓存的深度设计为2,写入数据位宽为512比特;而读出数据位宽固定为一个待译码数据的宽度,即8bit。

[0044] 图5是根据本发明的一种具体实施例中输入缓存的地址分配表。输入数据载入单元401将数据写入到输入缓存的时候,需要指明写入地址,保证整个译码过程的正确执行。

[0045] 图6是根据本发明的一种具体实施例中接口逻辑电路输入数据载入单元401的有限状态机的跳转示意图,该实施例的状态机包括IDLE状态50、SYS状态60、PAR1状态70、PAR2状态80和WAIT状态90等5个状态。其工作流程为:IDLE状态50是输入数据载入单元401不工作的空闲状态,当子译码器20需要读入一个滑窗的数据进行译码的时候,有限状态机根据当前处于前半次迭代还是处于后半次迭代选择跳转到SYS状态60或者PAR2状态80。这是由于根据本实施例的设计,在前半次迭代的时候需要连续读入系统信息Systematic和校验1信息Parity;而在后半次迭代的时候,只需要读入校验2信息Parity2,所需要的交织后的系统信息Systematic则缓存在子译码器20内部。

[0046] 当有限状态机处在SYS状态60的时候,需要根据当前滑窗的长度判断下一个周期是停留在SYS状态60还是跳转到PAR1状态70,这就是前述的最多需要两个仿存操作才能填满一个滑窗缓存。当采用P个子译码器并行译码的时候,则需要根据子译码器的个数P选择在SYS状态60停留的周期数。

[0047] 当有限状态机处在PAR1状态70和PAR2状态80的时候,其跳转条件类似,工作过程相同,只是需要根据校验1信息和校验2信息在存储单元中的存储位置提供不同的读地址。

[0048] 图7是根据本发明的一种P路并行译码器接口逻辑电路结构示意图,其中包括子译码器簇A0。如前所述,当存在P个子译码器并行译码的时候,输入数据载入单元401就需要在译码开始的时候,以滑窗为单位将待译码数据送入每个子译码器20内部的滑窗缓存中。输入数据载入单元401通过选通信号控制将数据写入到某一个子译码器20,根据图5所示的存储器地址分配表,写入到相应的滑窗缓存中。

[0049] 图8是根据本发明的一种具体实施例中输出接口逻辑实现框图。我们知道,LTE协

议中规定了码块长度K的取值范围是40–6144。存入到图3所示的输出缓存202中的数据是以字节为单位的,而当CPU_D_WIDTH的位宽为512比特时,需要根据地址生成器计算得到的地址将数据首先从输出缓存202中读出,然后经过数据压缩合并单元将数据合并到512比特位宽的数据总线接口中。如前所述,但存在P个子译码器并行译码的时候,地址生成器就需要根据子块的长度 $SB_LEN=K/P$,自动切换所访问的输出缓存,并产生相应的地址。对于数据压缩合并单元,需要根据CPU地址总线位宽CPU_D_WIDTH、码块长度K来判断何时将压缩合并的数据送到总线上。对于本具体实施例,其判断准则为

[0050] $\begin{cases} 1. \text{读入数据个数等于 } 64(\text{即 CPU_D_WIDTH}/8) \\ 2. \text{读入数据个数等于码块长度 } K \end{cases}。$

[0051] 以上所述的具体实施例,对本发明的目的、技术方案和有益效果进行了进一步详细说明,应理解的是,以上所述仅为本发明的具体实施例而已,并不用于限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

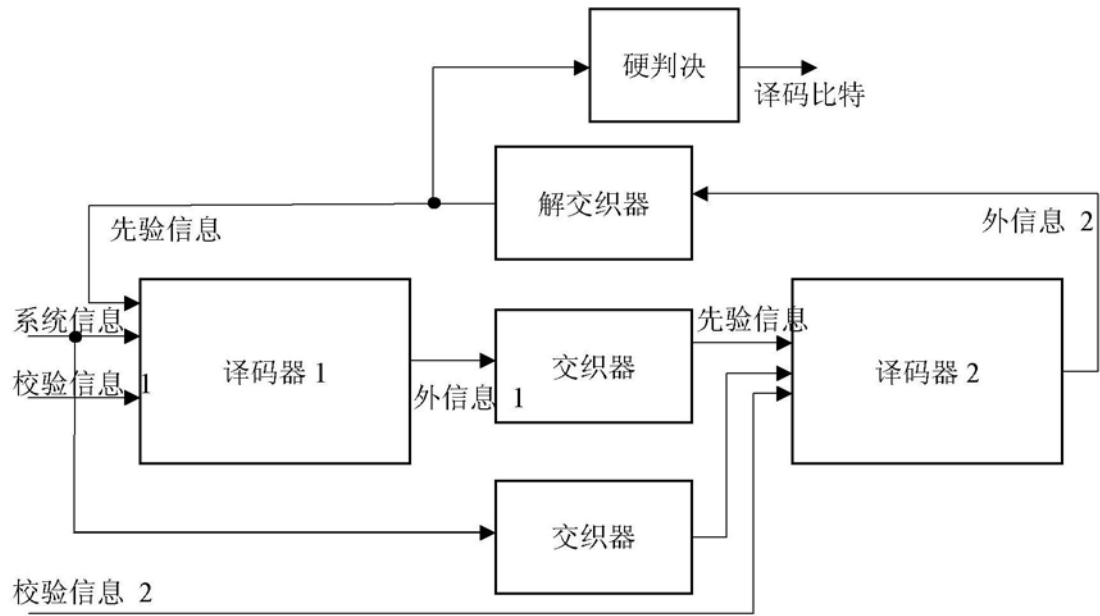


图1

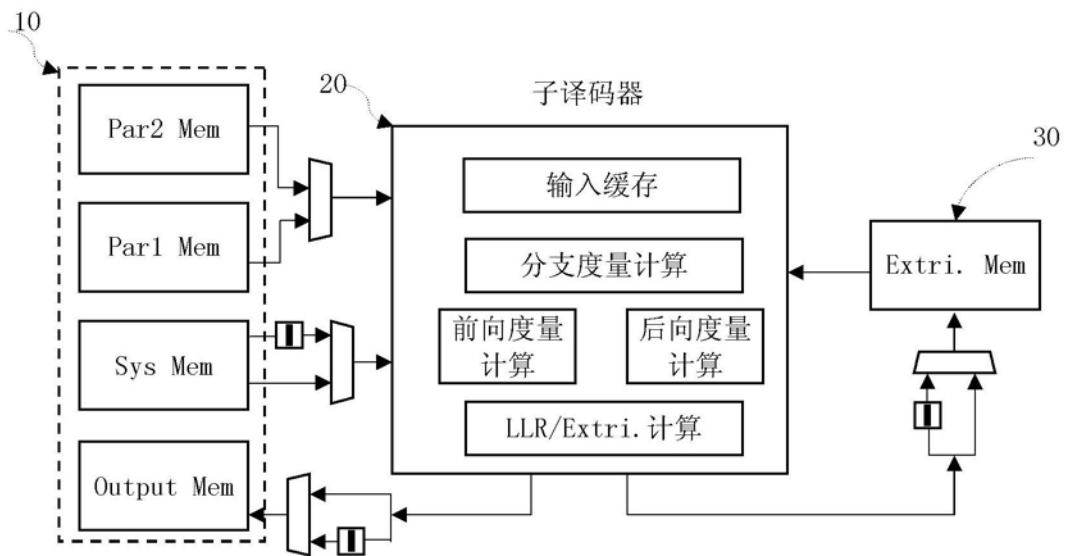


图2

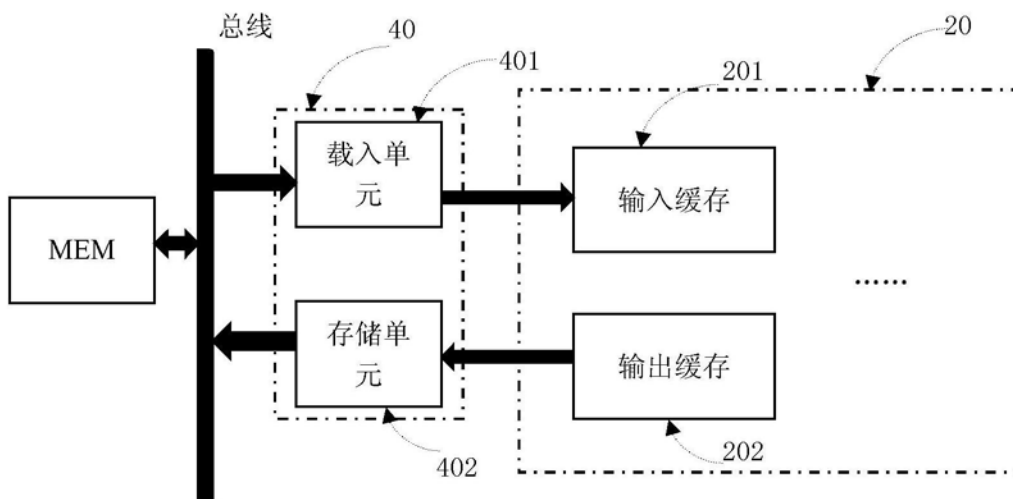


图3

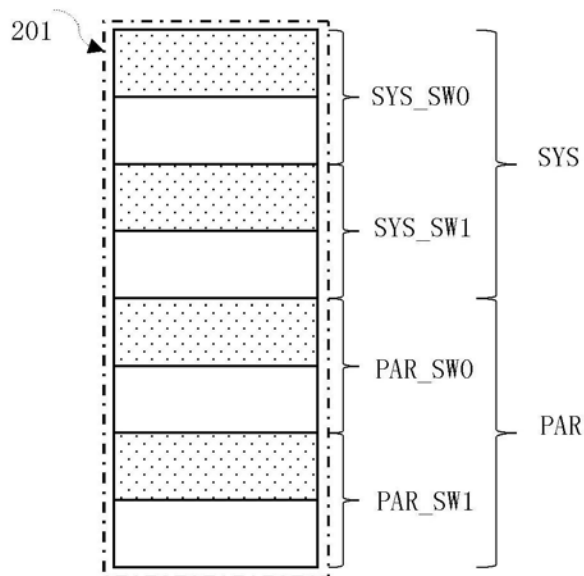


图4

存储器名称	存储器大小	存储器地址分配
SYS_SW0	128Byte	000-001
SYS_SW1	128Byte	010-011
PAR_SW0	128Byte	100-101
PAR_SW1	128Byte	110-111

图5

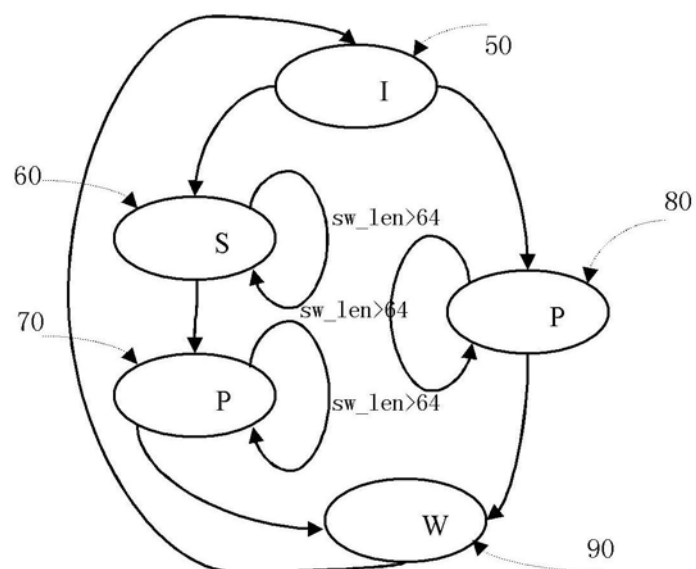


图6

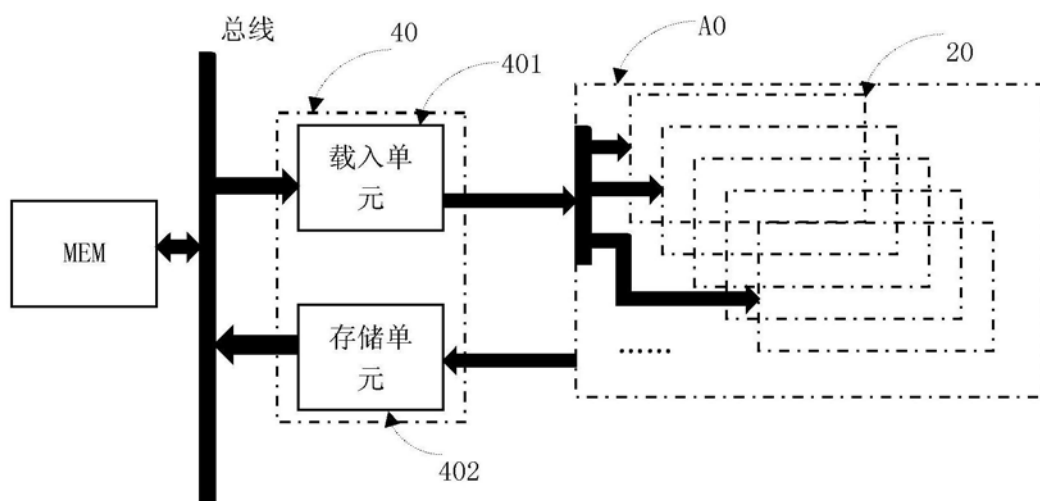


图7

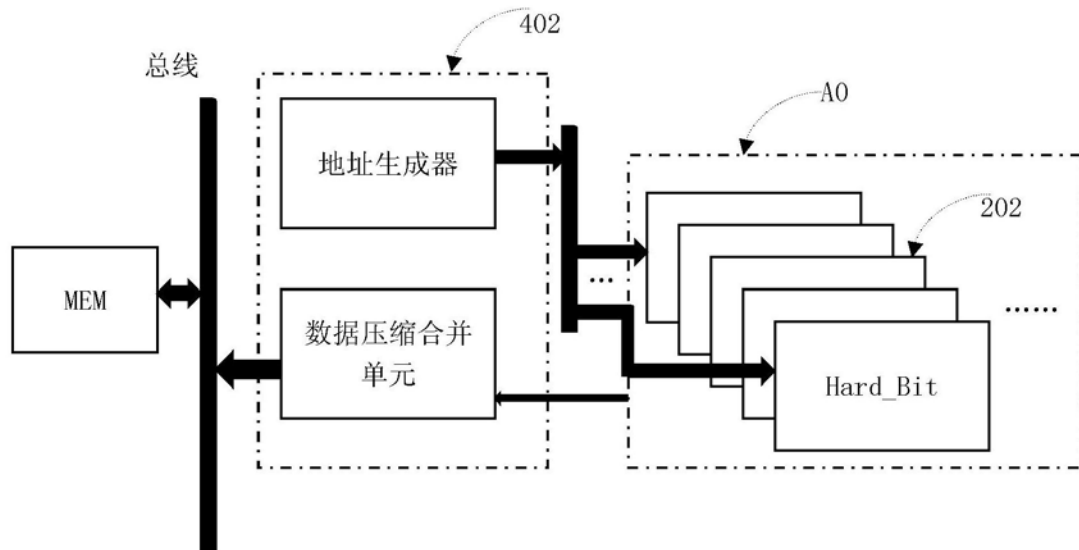


图8