



(12)发明专利

(10)授权公告号 CN 103873181 B

(45)授权公告日 2017.01.18

(21)申请号 201410136806.3

(22)申请日 2014.04.04

(65)同一申请的已公布的文献号

申请公布号 CN 103873181 A

(43)申请公布日 2014.06.18

(73)专利权人 中国科学院自动化研究所

地址 100190 北京市海淀区中关村东路95号

(72)发明人 吴军宁 王晓琴 郭晓龙 赵旭莹

林啸 张森 郭璟 王伟康

(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.

H04J 13/10(2011.01)

(56)对比文件

CN 102546157 A,2012.07.04,

CN 103378917 A,2013.10.30,

CN 102684871 A,2012.09.19,

US 2008019425 A1,2008.01.24,

US 2009116459 A1,2009.05.07,

审查员 杨文君

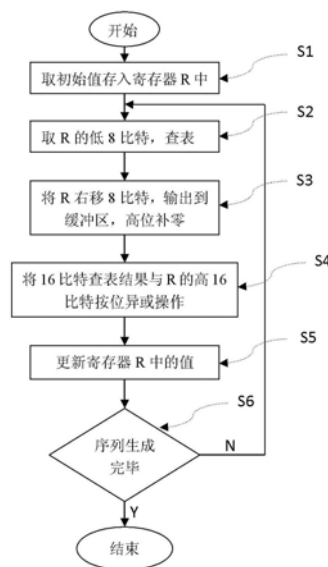
权利要求书1页 说明书6页 附图3页

(54)发明名称

一种LTE系统中伪随机序列并行生成方法

(57)摘要

本发明提出并公开了一种通过查表以及简单的逻辑运算操作,实现8比特并行生成伪随机序列的方法,查找表构造时的计算规则以及查表输出结果与输入数据通过移位、按位异或操作,完成伪随机序列生成过程的一系列操作,同时提出了采用SIMD指令多路并行生成伪随机序列的方法。本发明能够降低伪随机序列的生成时间,实现较快地生成数据序列,提高设计灵活性,可广泛应用于4G基站以及移动设备中。



1. 一种并行生成伪随机序列的方法,其特征在于,包括如下步骤:

步骤S1、根据LTE物理层协议中的规定,将伪随机序列初始状态值 C_{init} 转换成31比特的二进制数据,存入32比特的寄存器(R)中,最高位补零;

步骤S2、取所述寄存器中的低8比特作为查表操作的输入进行查表,得到16比特的查表输出结果;

步骤S3、将所述寄存器右移8比特,移出的8比特作为本次迭代得到伪随机序列存入输出缓冲区,所述寄存器的高8比特补零;

步骤S4、将步骤S2中查表得到的16比特查表输出结果与所述寄存器的高16比特按位异或操作;

步骤S5、使用步骤S4中异或操作后得到的值更新所述寄存器,用于下次迭代。

2. 根据权利要求1所述的并行生成伪随机序列的方法,其特征在于,在步骤S5之后还包括:

步骤S6、根据所需伪随机序列的长度,控制循环迭代的次数。

3. 根据权利要求1所述的并行生成伪随机序列的方法,其特征在于,由3GPP LTE物理层协议36.211中序列生成公式

$$x_2(n+31) = (x_2(n+3) + x_2(n)) \bmod 2, \text{并}$$

根据下列规则生成用于查表操作的表格,

$$LUT_OUT(n) = 0$$

$$LUT_OUT(n+1) = 0$$

$$LUT_OUT(n+2) = 0$$

$$LUT_OUT(n+3) = 0$$

$$LUT_OUT(n+4) = LUT_IN(n)$$

$$LUT_OUT(n+5) = LUT_IN(n+1)$$

$$LUT_OUT(n+6) = LUT_IN(n+2)$$

$$LUT_OUT(n+7) = LUT_IN(n) \oplus LUT_IN(n+3)$$

$$LUT_OUT(n+8) = LUT_IN(n+1) \oplus LUT_IN(n+4)$$

$$LUT_OUT(n+9) = LUT_IN(n+2) \oplus LUT_IN(n+5)$$

$$LUT_OUT(n+10) = LUT_IN(n+3) \oplus LUT_IN(n+6)$$

$$LUT_OUT(n+11) = LUT_IN(n+4) \oplus LUT_IN(n+7)$$

$$LUT_OUT(n+12) = LUT_IN(n+5)$$

$$LUT_OUT(n+13) = LUT_IN(n+6)$$

$$LUT_OUT(n+14) = LUT_IN(n+7)$$

$$LUT_OUT(n+15) = 0$$

其中,LUT_IN为8比特输入数据,LUT_OUT为16比特输出数据。

一种LTE系统中伪随机序列并行生成方法

技术领域

[0001] 本发明涉及通信技术领域,特别是涉及一种LTE系统中的伪随机序列并行生成方法。

背景技术

[0002] 随着科技的日益发展,无线通信技术得到了广泛应用,用户可以在任何地点、任何时间获取无线通信系统所带来的有效信息。然而,无线通信系统中如果对传输数据不做任何安全保护措施,则第三方用户可以轻易地获取当前用户的数据信息。同时,在实际的数字通信中,信息流在经过编码处理后,可能会出现连续的“0”或连续的“1”,这样破坏了码的平衡,影响到位同步的建立和保持。

[0003] 在长期演进(Long Term Evolution,LTE)系统中,加扰还具有第三层作用。对小区A和小区B,在信道编码和交织后,分别对其传输信号进行加扰。如果没有加扰,用户设备(UE)的解码器不能区分接收到的信号是来自本小区还是来自其他小区,它既可能对本小区的信号进行解码,也可能对其他小区的信号进行解码,使得性能降低。小区专属加扰可以通过不同的扰码对不同小区的信息进行区分,让UE只针对有用信息进行解码,以降低干扰。

[0004] 而加扰或解扰操作均需要将伪随机序列与原始数据进行按位操作,如加扰为将原始数据与伪随机序列按位进行逻辑异或操作,解扰则为根据伪随机序列改变原始软比特信息的符号。这其中尤为重要的就是根据所提供的初始条件,如何快速高效地生成伪随机序列。

[0005] 现有的伪随机序列生成的方法可分为两种:线性反馈移位寄存器(Linear Feedback Shift Register,LFSR)实现和向量空间变换。LFSR实现是伪随机序列生成最直接和最简便的方法,如申请号为20108004784.5、发明名称为“在无线通信系统中生成扰码的装置和方法”的中国发明专利公开说明书中提出了一种使用Gold-like序列的装置和方法,包括两个LFSR,通过彼此不同的两个m阶多项式生成Gold-like序列,即伪随机序列。向量空间变换方法,是根据序列初始状态和后续状态之间的线性关系,通过线性变换得到伪随机序列的方法,如申请号为201180001258.8、发明名称为“用于生成伪随机序列数据的数据序列的生成方法及设备”的中国发明专利公开说明书中提出了一种用于生成伪随机序列数据的方法,其根据数据序列A的递推公式,确定矩阵P,所述矩阵P中的各元素由数据序列A的递推公式中的抽头系数得到;确定待生成的数据项所对应的索引周期 n_0 ,根据所述数据序列A中的已知数据项以及所述矩阵P和索引周期 n_0 ,得到所述待生成的数据项对应的索引周期 n_0 内的数据序列。该方法需要预先确定矩阵P,并且完成矩阵和向量的乘法操作。

[0006] 在实际工作中,发明人发现现有两类技术均存在相应的问题:LFSR实现时是一种串行实现,每次移位只能生成一位序列数据,所需时间较长;而向量空间变换方法,尽管每次计算可以生成多位序列数据,但每次均需确定转换矩阵P,同时矩阵和向量的乘法操作不利于嵌入式移动设备实现。

发明内容

[0007] (一)要解决的技术问题

[0008] 为了克服用于LTE系统中加扰或解扰的伪随机序列串行生成耗时较长和向量空间变换需要频繁计算转换矩阵P以及矩阵向量乘法操作的复杂度较高等不足,二者均不利于嵌入式移动设备的实现,特别是矩阵向量的乘法操作,其实现尤为繁琐。

[0009] (二)技术方案

[0010] 本发明提出了一种通过查表以及简单的逻辑运算操作,实现八位并行生成伪随机序列的方法及装置,降低伪随机序列的生成时间,实现较快地生成数据序列,提高设计灵活性。该方法包括:

[0011] 步骤S2、取一个寄存器R中的低8比特作为查表操作的输入进行查表,得到16比特的查表输出结果;步骤S3、将所述寄存器右移8比特,移出的8比特作为本次迭代得到伪随机序列存入输出缓冲区,所述寄存器的高8比特补零;步骤S4,将步骤S2中查表得到的16比特查表输出结果与所述寄存器的高16比特按位异或操作;步骤S5、使用步骤S4中异或操作后得到的值更新所述寄存器,用于下次迭代。

[0012] 根据本发明的一种具体实施方式,k=8。

[0013] 根据本发明的一种具体实施方式,在步骤S2之前还包括步骤S1、根据LTE物理层协议中的规定,将得到的伪随机序列初始状态值 c_{init} 转换成31比特的二进制数据,存入32比特的所述寄存器中,最高位补零。

[0014] 根据本发明的一种具体实施方式,在步骤S5之后还包括步骤S6、根据所需伪随机序列的长度,控制循环迭代的次数。

[0015] 根据本发明的一种具体实施方式,由3GPP LTE物理层协议36.211中序列生成公式

[0016] $x_2(n+31)=(x_2(n+3)+x_2(n))\bmod 2$

[0017] 根据下列规则生成用于查表操作的表格。

[0018] $LUT_OUT(n)=0$

[0019] $LUT_OUT(n+1)=0$

[0020] $LUT_OUT(n+2)=0$

[0021] $LUT_OUT(n+3)=0$

[0022] $LUT_OUT(n+4)=LUT_IN(n)$

[0023] $LUT_OUT(n+5)=LUT_IN(n+1)$

[0024] $LUT_OUT(n+6)=LUT_IN(n+2)$

[0025] $LUT_OUT(n+7)=LUT_IN(n)\oplus LUT_IN(n+3)$

[0026] $LUT_OUT(n+8)=LUT_IN(n+1)\oplus LUT_IN(n+4)$

[0027] $LUT_OUT(n+9)=LUT_IN(n+2)\oplus LUT_IN(n+5)$

[0028] $LUT_OUT(n+10)=LUT_IN(n+3)\oplus LUT_IN(n+6)$

[0029] $LUT_OUT(n+11)=LUT_IN(n+4)\oplus LUT_IN(n+7)$

[0030] $LUT_OUT(n+12)=LUT_IN(n+5)$

[0031] $LUT_OUT(n+13)=LUT_IN(n+6)$

[0032] $LUT_OUT(n+14)=LUT_IN(n+7)$

[0033] $LUT_OUT(n+15)=0$

[0034] 其中, LUT_IN 为8比特输入数据, LUT_OUT 为16比特输出数据。

[0035] 根据本发明的一种具体实施方式,将伪随机序列的初始值 C_{init} 暂存在32比特寄存器R中,每次取低8比特作为查表操作输入数据,通过查表操作,得到16比特查表输出结果,以便进行后续操作。

[0036] 根据本发明的一种具体实施方式,将寄存器R右移8比特,且高位补零,移出的8比特数据,即作为伪随机序列输出。

[0037] 根据本发明的一种具体实施方式,将得到的16比特查找表输出结果,与寄存器R的高16比特做按位异或操作,结果存回寄存器R,以便进行下次循环操作。

[0038] 根据本发明的一种具体实施方式,根据所需生成伪随机序列长度L,控制循环过程执行次数,完成伪随机序列的生成。

[0039] 根据本发明的一种具体实施方式,将寄存器R的位宽设定为32比特,但是在具体实现时,可以充分利用单指令多数据流(Single Instruction Multiple Data,SIMD)的并行特性,扩展到64比特、128比特、256比特等等,同时完成多个伪随机序列的生成过程。

[0040] 根据本发明的一种具体实施方式,可以在伪随机序列生成过程中,结合申请号为201180001258.8、发明名称为“用于生成伪随机序列数据的数据序列的生成方法及设备”的中国发明专利公开说明书中提出了一种用于生成伪随机序列数据的方法,通过转换矩阵,将伪随机序列初始值($x_{30}, x_{29}, \dots, x_3, x_2, x_1, x_0$)转换到1600个周期之后的状态,即($x_{1630}, x_{1629}, \dots, x_{1603}, x_{1602}, x_{1601}, x_{1600}$),这样可以有效减少伪随机序列生成过程中冗余计算带来的时间消耗,提高系统的工作效率。

[0041] (三)有益效果

[0042] 本发明通过查表操作以及移位、异或等简单的逻辑计算,完成伪随机序列的生成过程,具有以下有益之处:

[0043] 本发明查表操作实现简单,且表项存储仅需 256×16 比特,易于嵌入式处理器实现。

[0044] 本发明仅通过一次移位、异或操作就可以同时输出8比特伪随机序列,可有效降低伪随机序列生成的复杂度提高生成效率。

[0045] 对于目前全面支持SIMD指令的数字信号处理器(Digital Signal Processor,DSP)来说,本发明可以很方便地通过线程级并行方案,进一步提高移动设备的处理能力。

附图说明

[0046] 图1是根据本发明的一种具体实施方式生成伪随机序列的流程图;

[0047] 图2是根据本发明提出的计算规则构造查找表示意图;

[0048] 图3是根据本发明的一种具体实施方式查表操作示意图;

[0049] 图4是根据本发明的一种具体实施方式异或并更新寄存器R示意图;

[0050] 图5是根据本发明提出的线程级并行方案示意图。

具体实施方式

[0051] 本发明提出了一种快速高效计算用于加扰或者解扰操作的伪随机序列的生成方

法,该方法仅仅包括查表以及移位、异或等简单操作,利于嵌入式移动处理器实现,缩短伪随机序列生成时间。

[0052] 该方法主要针对于快速、高效、并行地生成LTE物理层协议中用于加扰或者解扰操作的伪随机序列,但是又不局限于此。从本发明中所述的方法可以看出,对于无线通信协议中加扰操作所需的伪随机序列其生成方法类似,只是伪随机序列的初始状态以及计算规则有差别而已,本方法可以很容易的予以实现。

[0053] 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。

[0054] 图1是根据本发明的一种具体实施方式生成伪随机序列的计算流程图。从附图中可以看出,伪随机序列的生成是一个循环迭代多次的过程,其循环体主要包括四个步骤:步骤S2取寄存器R中的低8比特作为查表操作的输入进行查表,得到16比特的查表输出结果;步骤S3将寄存器R右移8比特,移出的8比特作为本次迭代得到伪随机序列存入输出缓冲区,寄存器R的高8比特补零;步骤S4将步骤S2中查表得到的16比特查表输出结果与寄存器R的高16比特按位异或;步骤S5使用步骤S4中异或操作后得到的值更新寄存器R,用于下次迭代。

[0055] 整个计算流程还包括步骤S1和步骤S6。步骤S1主要完成,根据LTE物理层协议中的规定,将得到的伪随机序列初始状态值 C_{init} 转换成31比特的二进制数据,存入32比特的寄存器R中,最高位补零;步骤S6主要完成,根据所需伪随机序列的长度,控制循环迭代的次数。

[0056] 图2是根据本发明提出的计算规则,构造查找表的示意图。其中LUT_IN为8比特的查找表输入数据,其取值范围为0000_0000b~1111_1111b,即0~255,共256中可能的输入,也就是说查找表包含256个表项。LUT_OUT为16比特的查找表输出结果,对于任意一种情况的输入数据,根据计算规则,分别计算LUT_OUT中的每一位,得到对应于该输入的查找表表项。如对于输入数据为0101_1101b的情况,LUT_OUT中的每一位输出为:

[0057] $LUT_OUT(n)=0$

[0058] $LUT_OUT(n+1)=0$

[0059] $LUT_OUT(n+2)=0$

[0060] $LUT_OUT(n+3)=0$

[0061] $LUT_OUT(n+4)=1$

[0062] $LUT_OUT(n+5)=1\oplus 0=1$

[0063] $LUT_OUT(n+6)=1\oplus 0\oplus 1=0$

[0064] $LUT_OUT(n+7)=1\oplus 0\oplus 1\oplus 1=1$

[0065] $LUT_OUT(n+8)=0\oplus 1\oplus 1\oplus 1=1$

[0066] $LUT_OUT(n+9)=1\oplus 1\oplus 1\oplus 0=1$

[0067] $LUT_OUT(n+10)=1\oplus 1\oplus 0\oplus 1=1$

[0068] $LUT_OUT(n+11)=1\oplus 0\oplus 1\oplus 0=0$

[0069] $LUT_OUT(n+12)=0\oplus 1\oplus 0=1$

[0070] $LUT_OUT(n+13)=1\oplus 0=1$

[0071] $LUT_OUT(n+14)=0$

[0072] $LUT_OUT(n+15)=0$

[0073] 即 $LUT_OUT_{0101_1101b}=0011_0111_1011_0000b$ 。可以根据此规则,生成所有的256项表项,用于查表操作。需要说明的是,该计算规则对应于LTE物理层协议36.211中序列生成公式

[0074] $x_2(n+31)=(x_2(n+3)+x_2(n+2)+x_2(n+1)+x_2(n))\bmod 2$

[0075] 而对于协议中规定的序列生成公式

[0076] $x_1(n+31)=(x_1(n+3)+x_1(n))\bmod 2$

[0077] 同样地,可以得到构造查找表的计算规则,即

[0078] $LUT_OUT(n)=0$

[0079] $LUT_OUT(n+1)=0$

[0080] $LUT_OUT(n+2)=0$

[0081] $LUT_OUT(n+3)=0$

[0082] $LUT_OUT(n+4)=LUT_IN(n)$

[0083] $LUT_OUT(n+5)=LUT_IN(n+1)$

[0084] $LUT_OUT(n+6)=LUT_IN(n+2)$

[0085] $LUT_OUT(n+7)=LUT_IN(n)\oplus LUT_IN(n+3)$

[0086] $LUT_OUT(n+8)=LUT_IN(n+1)\oplus LUT_IN(n+4)$

[0087] $LUT_OUT(n+9)=LUT_IN(n+2)\oplus LUT_IN(n+5)$

[0088] $LUT_OUT(n+10)=LUT_IN(n+3)\oplus LUT_IN(n+6)$

[0089] $LUT_OUT(n+11)=LUT_IN(n+4)\oplus LUT_IN(n+7)$

[0090] $LUT_OUT(n+12)=LUT_IN(n+5)$

[0091] $LUT_OUT(n+13)=LUT_IN(n+6)$

[0092] $LUT_OUT(n+14)=LUT_IN(n+7)$

[0093] $LUT_OUT(n+15)=0$

[0094] 根据此计算规则,我们可以构造用于生成序列 $x_1(n)$ 的查找表。同样地,对于输入数据0101_1101b的情况, LUT_OUT 中的每一位输出为:

[0095] $LUT_OUT(n)=0$

[0096] $LUT_OUT(n+1)=0$

[0097] $LUT_OUT(n+2)=0$

[0098] $LUT_OUT(n+3)=0$

[0099] $LUT_OUT(n+4)=1$

[0100] $LUT_OUT(n+5)=0=0$

[0101] $LUT_OUT(n+6)=1=1$

[0102] $LUT_OUT(n+7)=1\oplus 1=0$

[0103] $LUT_OUT(n+8)=0\oplus 1=1$

[0104] $LUT_OUT(n+9)=1\oplus 0=1$

[0105] $LUT_OUT(n+10)=1\oplus 1=0$

[0106] $LUT_OUT(n+11)=1\oplus 0=1$

[0107] $LUT_OUT(n+12)=0=0$

[0108] $LUT_OUT(n+13)=1=1$

[0109] $LUT_OUT(n+14)=0$

[0110] $LUT_OUT(n+15)=0$

[0111] 即 $LUT_OUT_{0101_1101b}=0010_1011_0101_0000b$ 。同样地,可以根据此规则,生成所有的256项表项,用于查表操作。

[0112] 图3是根据本发明的一种具体实施方式的查表操作示意图。根据输入的8比特数据,进行查表。由于查找表存储时按从0000_0000b~1111_1111b依次存储,查找表基地址加上地址偏移量,即输入数据,就得到对应于该输入数据的查找表结果表项。实现极为方便。

[0113] 图4是根据本发明的一种具体实施方式将所得16比特查表结果与寄存器R的高16比特按位异或并更新寄存器R的示意图。

[0114] 图5是根据本发明采用SIMD指令完成线程级并行计算伪随机序列的模块示意图。

[0115] SIMD是一种采用一个控制器来控制多个处理器,同时对一组数据(又称“数据矢量”)中的每一个分别执行相同的操作从而实现空间上的并行性的技术。例如Intel的MMX或者SSE,以及支持SIMD指令的TMS320C6455和TMS320C6670等嵌入式移动处理器。

[0116] 图5给出了根据本发明的一种采用128位宽的SIMD指令并行生成伪随机序列的示意框图。如前所述,该并行扩展,不局限于具体的并行度。

[0117] 从图5中可以看出,其计算过程与图1中所示相同。首先同时读入4个伪随机序列的初始状态值R0、R1、R2和R3,按序存入位宽为128比特的寄存器R中,每个初始值占每块的低31个比特。第二步就是同时取R0、R1、R2和R3中的低8比特,输入到LUT0、LUT1、LUT2和LUT3模块,进行查表操作,得到4个16比特的查表结果。第三步,通过移位操作,将R寄存器中的4个低8比特数据输出到结果缓冲区,并在高位进行补零操作,将第二步得到的16比特查表结果进行按位异或。第四步更新寄存器R的值,以便进行下次循环。

[0118] 通过SIMD指令,如前所述,可以同时计算4路伪随机序列,这对于LTE物理层的实现是非常有益的。根据LTE物理层协议规定,各个物理信道的加扰或者解扰操作都需要生成伪随机序列,且下行参考信号如小区特定参考信号、定位参考信号等也需要生成相对应的伪随机序列。如果采用本发明所提出的并行实现方法,可以通过SIMD指令,同时并行生成多个伪随机序列,减少LTE物理层的时间消耗,提高基站或者移动设备的传输速率。

[0119] 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。

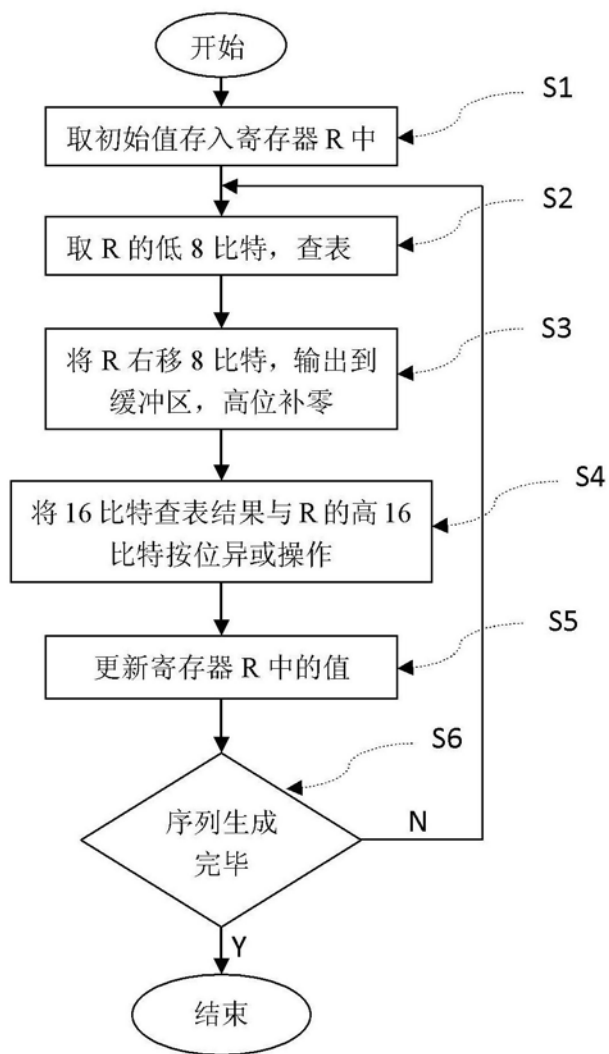


图1

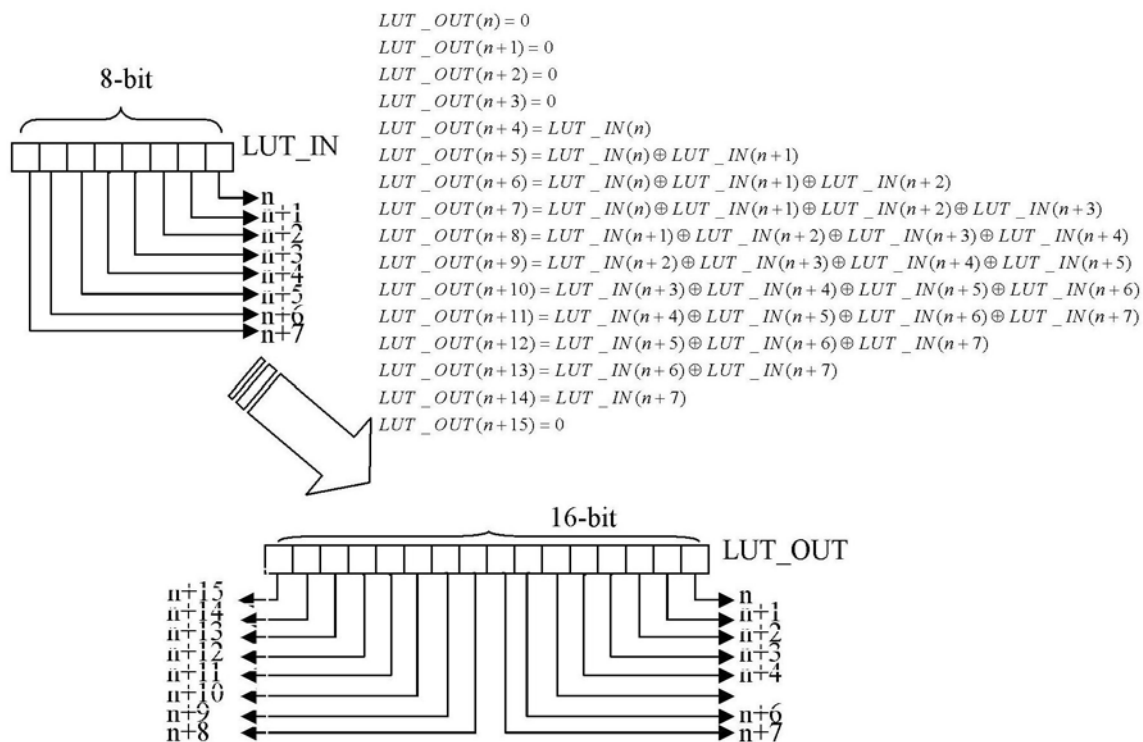


图2

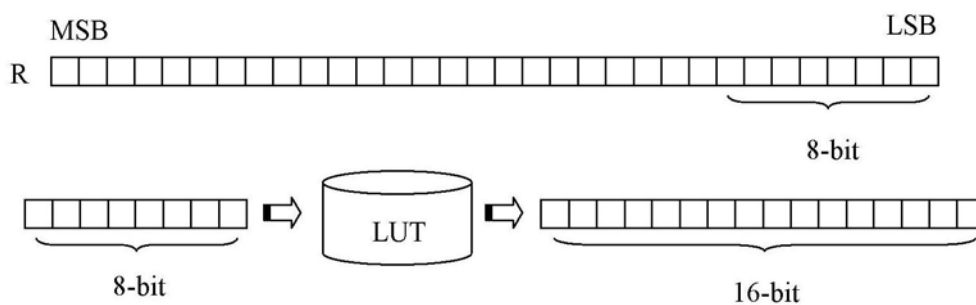


图3

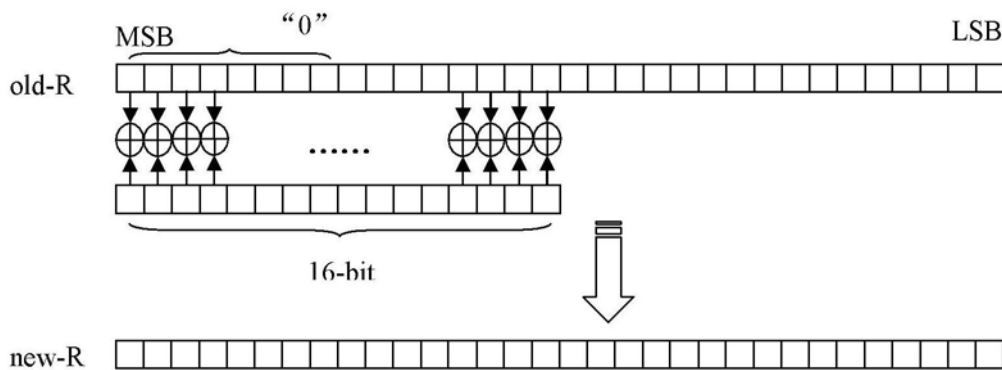


图4

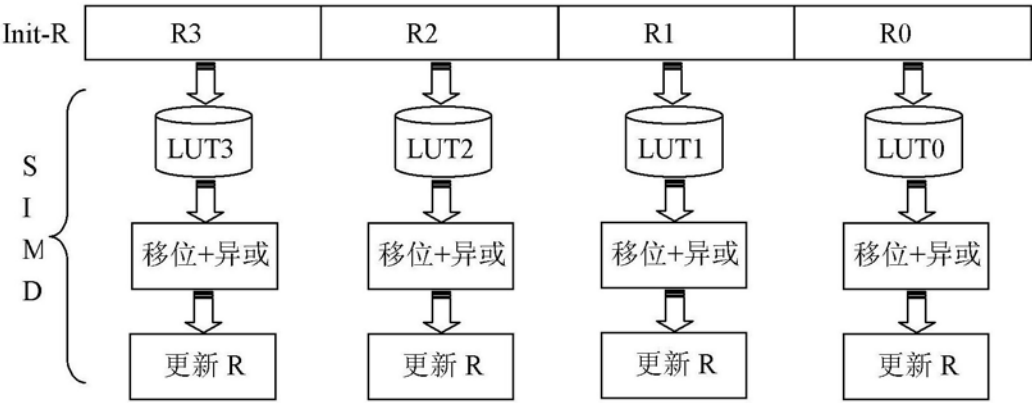


图5