



(12) 发明专利申请

(10) 申请公布号 CN 105049057 A

(43) 申请公布日 2015. 11. 11

(21) 申请号 201510504485. 2

(22) 申请日 2015. 08. 17

(71) 申请人 中国航天科技集团公司第九研究院
第七七一研究所

地址 710068 陕西省西安市太白南路 198 号

(72) 发明人 赵坤鹏 陈庆宇 吴龙胜 马徐瀚

(74) 专利代理机构 西安通大专利代理有限责任
公司 61200

代理人 李宏德

(51) Int. Cl.

H03M 13/09(2006. 01)

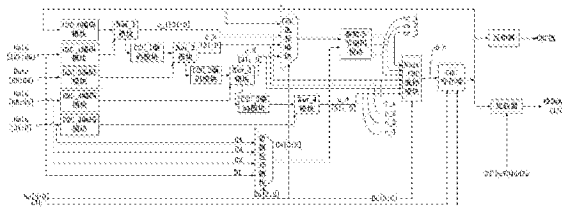
权利要求书4页 说明书12页 附图3页

(54) 发明名称

一种面向 128 位并行输入的 CRC-32 校验电路

(57) 摘要

本发明提供了一种面向 128 位并行输入的 CRC-32 校验电路,通过基于预处理矩阵的硬件电路对 4 组 32 位并行数据分别进行编码,对 CRC 寄存器中的 CRC 值进行多级编码操作。将得到的各级编码结果分别与 4 组并行数据的预处理结果进行异或运算,在整数字节编码模块中对数据和各级 CRC 值编码结果进行进一步编码运算得到非整字的编码结果,最终通过 next_crc 选择模块选择出最终的编码结果,通过反相器得到 CRC 值,通过比较器得到校验结果。在不增加硬件成本、不降低整体系统性能的基础上,提供了高效率的 CRC 编解码电路,大大提高了实时处理数据的速度和数据传输效率,能够实现对 128 位以内的 8*N 位并行数据的 CRC 编码校验。



1. 一种面向 128 位并行输入的 CRC-32 校验电路,其特征在於,包括整数字节编码模块,反相器,比较器,分别以字节有效信号 be 为数据选择输出信号的 CRC 选择模块, NEXT_CRC 选择模块, CRC 寄存器和数据选择模块;以及用于对 $8*N$ 位并行数据从高位到低位均分为四组后的输入数据对应进行预处理编码的四个 D2C 编码模块,其中, $N = 1, 2, \dots, 16$;用于对 CRC 寄存器中的 CRC 值进行四级编码的四级 C2C 编码模块;用于将四组预处理结果和各级编码结果对应进行异或运算的四级 Xor 模块;

CRC 选择模块用于对 CRC 寄存器中的 CRC 值和前三级 Xor 模块的输出进行选择输出,作为整数字节编码模块的初始 CRC 值;

数据选择模块用于对四组输入数据进行选择输出 32 位并行数据,作为整数字节编码模块的输入值;

整数字节编码模块用于将初始 CRC 值对输入值进行校验,然后输出三个分别关于高 8 位、高 16 位和高 24 位的 CRC 校验结果;

NEXT_CRC 选择模块采用多路选择器,输入端分别连接整数字节编码模块的三路校验结果和四级 Xor 模块的异或运算结果,输出端输出选择后得到的 CRC 校验结果,并连接至 CRC 寄存器的输入端;

反相器用于将 CRC 寄存器中的 CRC 值按位取反输出所需的 CRC 编码;

比较器用于将 CRC 寄存器中的 CRC 值与魔数相比,并将校验成功或失败的结果输出。

2. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征在於,四个 D2C 编码模块包括 D2C_1 编码模块, D2C_2 编码模块, D2C_3 编码模块和 D2C_4 编码模块;

四级 C2C 编码模块包括 C2C_0 编码模块, C2C_1 编码模块, C2C_2 编码模块和 C2C_3 编码模块;

四级 Xor 模块包括 Xor_1 模块, Xor_2 模块, Xor_3 模块和 Xor_4 模块;

四个 D2C 编码模块的输出端和四级 C2C 编码模块的输出端分别连接四级 Xor 模块的输入端; Xor_1 模块的输出端连接 C2C_1 编码模块的输入端, Xor_2 模块的输出端连接 C2C_2 编码模块的输入端, Xor_3 模块的输出端连接 C2C_3 编码模块的输入端。

3. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征在於, CRC 寄存器连接有 CRC 时钟信号 clk 和 CRC 复位信号 rst。

4. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征在於,当 $N < 16$ 时,将并行数据低位补 0 达到 128 位后均分为 4 组数据再分别进行预处理。

5. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征在於, C2C 编码模块,将输入的 32 位并行数据按以下编码方式进行编码,其中, + 表示二进制异或运算;

$$\begin{aligned}
c2c[31] &= c11 + c15 + c23 + c24 + c25 + c27 + c28 + c29 + c30 + c31 + c5 + c8 + c9 \\
c2c[30] &= c10 + c14 + c22 + c23 + c24 + c26 + c27 + c28 + c29 + c30 + c4 + c7 + c8 \\
c2c[29] &= c13 + c21 + c22 + c23 + c25 + c26 + c27 + c28 + c29 + c3 + c31 + c6 + c7 + c9 \\
c2c[28] &= c12 + c2 + c20 + c21 + c22 + c24 + c25 + c26 + c27 + c28 + c30 + c5 + c6 + c8 \\
c2c[27] &= c1 + c11 + c19 + c20 + c21 + c23 + c24 + c25 + c26 + c27 + c29 + c4 + c5 + c7 \\
c2c[26] &= c0 + c10 + c18 + c19 + c20 + c22 + c23 + c24 + c25 + c26 + c28 + c3 + c31 + c4 + c6 \\
c2c[25] &= c11 + c15 + c17 + c18 + c19 + c2 + c21 + c22 + c28 + c29 + c3 + c31 + c8 \\
c2c[24] &= c1 + c10 + c14 + c16 + c17 + c18 + c2 + c20 + c21 + c27 + c28 + c30 + c7 \\
c2c[23] &= c0 + c1 + c13 + c15 + c16 + c17 + c19 + c20 + c26 + c27 + c29 + c31 + c6 + c9 \\
c2c[22] &= c0 + c11 + c12 + c14 + c16 + c18 + c19 + c23 + c24 + c26 + c27 + c29 + c31 + c9 \\
c2c[21] &= c10 + c13 + c17 + c18 + c22 + c24 + c26 + c27 + c29 + c31 + c5 + c9 \\
c2c[20] &= c12 + c16 + c17 + c21 + c23 + c25 + c26 + c28 + c30 + c4 + c8 + c9 \\
c2c[19] &= c11 + c15 + c16 + c20 + c22 + c24 + c25 + c27 + c29 + c3 + c7 + c8 \\
c2c[18] &= c10 + c14 + c15 + c19 + c2 + c21 + c23 + c24 + c26 + c28 + c31 + c6 + c7 \\
c2c[17] &= c1 + c13 + c14 + c18 + c20 + c22 + c23 + c25 + c27 + c30 + c31 + c5 + c6 + c9 \\
c2c[16] &= c0 + c12 + c13 + c17 + c19 + c21 + c22 + c24 + c26 + c29 + c30 + c4 + c5 + c8 \\
c2c[15] &= c12 + c15 + c16 + c18 + c20 + c21 + c24 + c27 + c3 + c30 + c4 + c5 + c7 + c8 + c9 \\
c2c[14] &= c11 + c14 + c15 + c17 + c19 + c2 + c20 + c23 + c26 + c29 + c3 + c4 + c6 + c7 + c8 \\
c2c[13] &= c1 + c10 + c13 + c14 + c16 + c18 + c19 + c2 + c22 + c25 + c28 + c3 + c31 + c5 + c6 + c7 \\
c2c[12] &= c0 + c1 + c12 + c13 + c15 + c17 + c18 + c2 + c21 + c24 + c27 + c30 + c31 + c4 + c5 + c6 + c9 \\
c2c[11] &= c0 + c1 + c12 + c14 + c15 + c16 + c17 + c20 + c24 + c25 + c26 + c27 + c28 + c3 + c31 + c4 + c9 \\
c2c[10] &= c0 + c13 + c14 + c16 + c19 + c2 + c26 + c28 + c29 + c3 + c31 + c5 + c9 \\
c2c[9] &= c1 + c11 + c12 + c13 + c18 + c2 + c23 + c24 + c29 + c4 + c5 + c9 \\
c2c[8] &= c0 + c1 + c10 + c11 + c12 + c17 + c22 + c23 + c28 + c3 + c31 + c4 + c8 \\
c2c[7] &= c0 + c10 + c15 + c16 + c2 + c21 + c22 + c23 + c24 + c25 + c28 + c29 + c3 + c5 + c7 + c8 \\
c2c[6] &= c1 + c11 + c14 + c2 + c20 + c21 + c22 + c25 + c29 + c30 + c4 + c5 + c6 + c7 + c8 \\
c2c[5] &= c0 + c1 + c10 + c13 + c19 + c20 + c21 + c24 + c28 + c29 + c3 + c4 + c5 + c6 + c7 \\
c2c[4] &= c0 + c11 + c12 + c15 + c18 + c19 + c2 + c20 + c24 + c25 + c29 + c3 + c30 + c31 + c4 + c6 + c8 \\
c2c[3] &= c1 + c10 + c14 + c15 + c17 + c18 + c19 + c2 + c25 + c27 + c3 + c31 + c7 + c8 + c9 \\
c2c[2] &= c0 + c1 + c13 + c14 + c16 + c17 + c18 + c2 + c24 + c26 + c30 + c31 + c6 + c7 + c8 + c9 \\
c2c[1] &= c0 + c1 + c11 + c12 + c13 + c16 + c17 + c24 + c27 + c28 + c6 + c7 + c9 \\
c2c[0] &= c0 + c10 + c12 + c16 + c24 + c25 + c26 + c28 + c29 + c30 + c31 + c6 + c9
\end{aligned}$$

6. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征在干, D2C 编码模块将输入的 32 位并行数据按以下编码方式进行编码,其中,+表示二进制异或运算;

$$\begin{aligned}
 d2c[31] &= d12 + d14 + d15 + d16 + d2 + d24 + d25 + d26 + d27 + d28 + d30 + d31 + d8 \\
 d2c[30] &= d0 + d13 + d15 + d16 + d17 + d25 + d26 + d27 + d28 + d29 + d3 + d31 + d9 \\
 d2c[29] &= d0 + d1 + d10 + d14 + d16 + d17 + d18 + d24 + d26 + d27 + d28 + d29 + d30 + d4 \\
 d2c[28] &= d1 + d11 + d15 + d17 + d18 + d19 + d2 + d25 + d27 + d28 + d29 + d30 + d31 + d5 \\
 d2c[27] &= d0 + d12 + d16 + d18 + d19 + d2 + d20 + d26 + d28 + d29 + d3 + d30 + d31 + d6 \\
 d2c[26] &= d1 + d13 + d16 + d17 + d19 + d20 + d21 + d24 + d27 + d29 + d3 + d30 + d31 + d4 + d7 \\
 d2c[25] &= d12 + d15 + d17 + d18 + d20 + d21 + d22 + d24 + d26 + d27 + d4 + d5 + d8 \\
 d2c[24] &= d0 + d13 + d18 + d19 + d21 + d22 + d23 + d25 + d27 + d28 + d5 + d6 + d9 \\
 d2c[23] &= d1 + d10 + d14 + d19 + d20 + d22 + d23 + d24 + d26 + d28 + d29 + d6 + d7 + d8 \\
 d2c[22] &= d11 + d12 + d14 + d16 + d20 + d21 + d23 + d24 + d26 + d28 + d29 + d31 + d7 + d9 \\
 d2c[21] &= d10 + d13 + d14 + d17 + d2 + d21 + d22 + d24 + d26 + d28 + d29 + d31 \\
 d2c[20] &= d11 + d14 + d15 + d16 + d18 + d22 + d23 + d25 + d27 + d29 + d3 + d30 \\
 d2c[19] &= d0 + d12 + d15 + d17 + d19 + d23 + d26 + d28 + d30 + d31 + d4 + d8 \\
 d2c[18] &= d0 + d1 + d13 + d16 + d18 + d20 + d24 + d27 + d29 + d31 + d5 + d8 + d9 \\
 d2c[17] &= d1 + d10 + d14 + d16 + d17 + d19 + d2 + d21 + d24 + d25 + d28 + d30 + d6 + d9 \\
 d2c[16] &= d10 + d11 + d15 + d17 + d18 + d2 + d20 + d22 + d25 + d26 + d29 + d3 + d31 + d7 \\
 d2c[15] &= d0 + d11 + d14 + d15 + d18 + d19 + d2 + d21 + d23 + d25 + d28 + d3 + d31 + d4 + d8 \\
 d2c[14] &= d0 + d1 + d12 + d15 + d16 + d19 + d20 + d22 + d26 + d29 + d3 + d4 + d5 + d8 + d9 \\
 d2c[13] &= d0 + d1 + d10 + d13 + d17 + d2 + d20 + d21 + d23 + d24 + d27 + d30 + d4 + d5 + d6 + d9 \\
 d2c[12] &= d1 + d10 + d11 + d14 + d18 + d2 + d21 + d22 + d24 + d25 + d28 + d3 + d31 + d5 + d6 + d7 + d8 \\
 d2c[11] &= d11 + d14 + d19 + d22 + d23 + d24 + d27 + d28 + d29 + d3 + d30 + d31 + d4 + d6 + d7 + d8 + d9 \\
 d2c[10] &= d10 + d14 + d2 + d20 + d23 + d24 + d26 + d27 + d29 + d4 + d5 + d7 + d9 \\
 d2c[9] &= d10 + d11 + d12 + d14 + d16 + d2 + d21 + d26 + d3 + d31 + d5 + d6 \\
 d2c[8] &= d11 + d12 + d13 + d15 + d16 + d17 + d22 + d24 + d27 + d3 + d4 + d6 + d7 \\
 d2c[7] &= d0 + d13 + d15 + d16 + d17 + d18 + d2 + d23 + d26 + d27 + d30 + d31 + d4 + d5 + d7 + d8 \\
 d2c[6] &= d0 + d1 + d12 + d15 + d17 + d18 + d19 + d2 + d25 + d26 + d3 + d30 + d5 + d6 + d9 \\
 d2c[5] &= d0 + d1 + d10 + d13 + d18 + d19 + d2 + d20 + d26 + d27 + d3 + d31 + d4 + d6 + d7 \\
 d2c[4] &= d1 + d11 + d12 + d15 + d19 + d20 + d21 + d24 + d25 + d26 + d3 + d30 + d31 + d4 + d5 + d7 + d8 \\
 d2c[3] &= d0 + d13 + d14 + d15 + d20 + d21 + d22 + d24 + d28 + d30 + d4 + d5 + d6 + d8 + d9 \\
 d2c[2] &= d0 + d1 + d10 + d14 + d15 + d21 + d22 + d23 + d24 + d25 + d29 + d31 + d5 + d6 + d7 + d9 \\
 d2c[1] &= d0 + d1 + d10 + d11 + d12 + d14 + d22 + d23 + d27 + d28 + d31 + d6 + d7 \\
 d2c[0] &= d1 + d11 + d13 + d14 + d23 + d24 + d25 + d26 + d27 + d29 + d30 + d31 + d7
 \end{aligned}$$

7. 根据权利要求 1 所述的一种面向 128 位并行输入的 CRC-32 校验电路,其特征不在于, 整数字节编码模块将输入初始 CRC 值对输入值按以下编码方式进行编码,其中,+ 表示二进制异或运算;

cre_8[31]=c23 + c29 + d2
cre_8[30]=c22 + c28 + c31 + d0 + d3
cre_8[29]=c21 + c27 + c30 + c31 + d0 + d1 + d4
cre_8[28]=c20 + c26 + c29 + c30 + d1 + d2 + d5
cre_8[27]=c19 + c25 + c28 + c29 + c31 + d0 + d2 + d3 + d6
cre_8[26]=c18 + c24 + c27 + c28 + c30 + d1 + d3 + d4 + d7
cre_8[25]=c17 + c26 + c27 + d4 + d5
cre_8[24]=c16 + c25 + c26 + c31 + d0 + d5 + d6
cre_8[23]=c15 + c24 + c25 + c30 + d1 + d6 + d7
cre_8[22]=c14 + c24 + d7
cre_8[21]=c13 + c29 + d2
cre_8[20]=c12 + c28 + d3
cre_8[19]=c11 + c27 + c31 + d0 + d4
cre_8[18]=c10 + c26 + c30 + c31 + d0 + d1 + d5
cre_8[17]=c25 + c29 + c30 + c9 + d1 + d2 + d6
cre_8[16]=c24 + c28 + c29 + c8 + d2 + d3 + d7
cre_8[15]=c27 + c28 + c29 + c31 + c7 + d0 + d2 + d3 + d4
cre_8[14]=c26 + c27 + c28 + c30 + c31 + c6 + d0 + d1 + d3 + d4 + d5
cre_8[13]=c25 + c26 + c27 + c29 + c30 + c31 + c5 + d0 + d1 + d2 + d4 + d5 + d6
cre_8[12]=c24 + c25 + c26 + c28 + c29 + c30 + c4 + d1 + d2 + d3 + d5 + d6 + d7
cre_8[11]=c24 + c25 + c27 + c28 + c3 + d3 + d4 + d6 + d7
cre_8[10]=c2 + c24 + c26 + c27 + c29 + d2 + d4 + d5 + d7
cre_8[9]=c1 + c25 + c26 + c28 + c29 + d2 + d3 + d5 + d6
cre_8[8]=c0 + c24 + c25 + c27 + c28 + d3 + d4 + d6 + d7
cre_8[7]=c24 + c26 + c27 + c29 + c31 + d0 + d2 + d4 + d5 + d7
cre_8[6]=c25 + c26 + c28 + c29 + c30 + c31 + d0 + d1 + d2 + d3 + d5 + d6
cre_8[5]=c24 + c25 + c27 + c28 + c29 + c30 + c31 + d0 + d1 + d2 + d3 + d4 + d6 + d7
cre_8[4]=c24 + c26 + c27 + c28 + c30 + d1 + d3 + d4 + d5 + d7
cre_8[3]=c25 + c26 + c27 + c31 + d0 + d4 + d5 + d6
cre_8[2]=c24 + c25 + c26 + c30 + c31 + d0 + d1 + d5 + d6 + d7
cre_8[1]=c24 + c25 + c30 + c31 + d0 + d1 + d6 + d7
cre_8[0]=c24 + c30 + d1 + d7

一种面向 128 位并行输入的 CRC-32 校验电路

技术领域

[0001] 本发明专利涉及容错技术领域,具体为一种面向 128 位并行输入的 CRC-32 校验电路。

背景技术

[0002] 循环冗余校验 CRC (Cyclic Redundancy Check) 以较优的误码检错能力和易于硬件实现的优点,广泛应用于各种通信协议中。

[0003] 传统的 CRC 编解码电路基于串行数据输入方式,其实现方式简单,资源占用小,但是其只能处理串行输入的数据,难以实现当前高速并行网络的数据校验。当前,实现并行数据的校验电路多样,主要有:(1) 公式法,根据串行移位电路推导得出当前 CRC 值与前一级 CRC 值和当前并行数据输入的关系。随着位宽的提高,公式法推导关系变得极为复杂,难以实现。(2) 查表法,根据输入数据从 CRC 码表中查找出相应的值进行处理。由于查表法需要大容量的存储单元,其硬件消耗巨大,随着位宽的提高,其代价更高,不利于实现高位宽的实时校验处理。

发明内容

[0004] 针对现有技术中存在的问题,本发明提供一种硬件成本低,实时处理速度快,数据传输效率高的面向 128 位并行输入的 CRC-32 校验电路。

[0005] 本发明是通过以下技术方案来实现:

[0006] 一种面向 128 位并行输入的 CRC-32 校验电路,包括整数字节编码模块,反相器,比较器,分别以字节有效信号 be 为数据选择输出信号的 CRC 选择模块, NEXT_CRC 选择模块, CRC 寄存器和数据选择模块;以及用于对 $8*N$ 位并行数据从高位到低位均分为四组后的输入数据对应进行预处理编码的四个 D2C 编码模块,其中, $N = 1, 2, \dots, 16$;用于对 CRC 寄存器中的 CRC 值进行四级编码的四级 C2C 编码模块;用于将四组预处理结果和各级编码结果对应进行异或运算的四级 Xor 模块;CRC 选择模块用于对 CRC 寄存器中的 CRC 值和前三级 Xor 模块的输出进行选择输出,作为整数字节编码模块的初始 CRC 值;数据选择模块用于对四组输入数据进行选择输出 32 位并行数据,作为整数字节编码模块的输入值;整数字节编码模块用于将初始 CRC 值对输入值进行校验,然后输出三个分别关于高 8 位、高 16 位和高 24 位的 CRC 校验结果;NEXT_CRC 选择模块采用多路选择器,输入端分别连接整数字节编码模块的三路校验结果和四级 Xor 模块的异或运算结果,输出端输出选择后得到的 CRC 校验结果,并连接至 CRC 寄存器的输入端;反相器用于将 CRC 寄存器中的 CRC 值按位取反输出所需的 CRC 编码;比较器用于将 CRC 寄存器中的 CRC 值与魔数相比,并将校验成功或失败的结果输出。

[0007] 优选的,四个 D2C 编码模块包括 D2C_1 编码模块,D2C_2 编码模块,D2C_3 编码模块和 D2C_4 编码模块;四级 C2C 编码模块包括 C2C_0 编码模块,C2C_1 编码模块,C2C_2 编码模块和 C2C_3 编码模块;四级 Xor 模块包括 Xor_1 模块, Xor_2 模块, Xor_3 模块和 Xor_4 模

块；四个 D2C 编码模块的输出端和四级 C2C 编码模块的输出端分别连接四级 Xor 模块的输入端；Xor_1 模块的输出端连接 C2C_1 编码模块的输入端，Xor_2 模块的输出端连接 C2C_2 编码模块的输入端，Xor_3 模块的输出端连接 C2C_3 编码模块的输入端。

[0008] 优选的，CRC 寄存器连接有 CRC 时钟信号 clk 和 CRC 复位信号 rst。

[0009] 优选的，当 $N < 16$ 时，将并行数据低位补 0 达到 128 位后均分为 4 组数据再分别进行预处理。

[0010] 优选的，C2C 编码模块，将输入的 32 位并行数据按以下编码方式进行编码，其中，+ 表示二进制异或运算；

[0011]

$$\begin{aligned}
 c2c[31] &= c11 + c15 + c23 + c24 + c25 + c27 + c28 + c29 + c30 + c31 + c5 + c8 + c9 \\
 c2c[30] &= c10 + c14 + c22 + c23 + c24 + c26 + c27 + c28 + c29 + c30 + c4 + c7 + c8 \\
 c2c[29] &= c13 + c21 + c22 + c23 + c25 + c26 + c27 + c28 + c29 + c3 + c31 + c6 + c7 + c9 \\
 c2c[28] &= c12 + c2 + c20 + c21 + c22 + c24 + c25 + c26 + c27 + c28 + c30 + c5 + c6 + c8 \\
 c2c[27] &= c1 + c11 + c19 + c20 + c21 + c23 + c24 + c25 + c26 + c27 + c29 + c4 + c5 + c7 \\
 c2c[26] &= c0 + c10 + c18 + c19 + c20 + c22 + c23 + c24 + c25 + c26 + c28 + c3 + c31 + c4 + c6 \\
 c2c[25] &= c11 + c15 + c17 + c18 + c19 + c2 + c21 + c22 + c28 + c29 + c3 + c31 + c8 \\
 c2c[24] &= c1 + c10 + c14 + c16 + c17 + c18 + c2 + c20 + c21 + c27 + c28 + c30 + c7 \\
 c2c[23] &= c0 + c1 + c13 + c15 + c16 + c17 + c19 + c20 + c26 + c27 + c29 + c31 + c6 + c9 \\
 c2c[22] &= c0 + c11 + c12 + c14 + c16 + c18 + c19 + c23 + c24 + c26 + c27 + c29 + c31 + c9 \\
 c2c[21] &= c10 + c13 + c17 + c18 + c22 + c24 + c26 + c27 + c29 + c31 + c5 + c9 \\
 c2c[20] &= c12 + c16 + c17 + c21 + c23 + c25 + c26 + c28 + c30 + c4 + c8 + c9 \\
 c2c[19] &= c11 + c15 + c16 + c20 + c22 + c24 + c25 + c27 + c29 + c3 + c7 + c8 \\
 c2c[18] &= c10 + c14 + c15 + c19 + c2 + c21 + c23 + c24 + c26 + c28 + c31 + c6 + c7 \\
 c2c[17] &= c1 + c13 + c14 + c18 + c20 + c22 + c23 + c25 + c27 + c30 + c31 + c5 + c6 + c9 \\
 c2c[16] &= c0 + c12 + c13 + c17 + c19 + c21 + c22 + c24 + c26 + c29 + c30 + c4 + c5 + c8 \\
 c2c[15] &= c12 + c15 + c16 + c18 + c20 + c21 + c24 + c27 + c3 + c30 + c4 + c5 + c7 + c8 + c9 \\
 c2c[14] &= c11 + c14 + c15 + c17 + c19 + c2 + c20 + c23 + c26 + c29 + c3 + c4 + c6 + c7 + c8 \\
 c2c[13] &= c1 + c10 + c13 + c14 + c16 + c18 + c19 + c2 + c22 + c25 + c28 + c3 + c31 + c5 + c6 + c7 \\
 c2c[12] &= c0 + c1 + c12 + c13 + c15 + c17 + c18 + c2 + c21 + c24 + c27 + c30 + c31 + c4 + c5 + c6 + c9 \\
 c2c[11] &= c0 + c1 + c12 + c14 + c15 + c16 + c17 + c20 + c24 + c25 + c26 + c27 + c28 + c3 + c31 + c4 + c9 \\
 c2c[10] &= c0 + c13 + c14 + c16 + c19 + c2 + c26 + c28 + c29 + c3 + c31 + c5 + c9 \\
 c2c[9] &= c1 + c11 + c12 + c13 + c18 + c2 + c23 + c24 + c29 + c4 + c5 + c9 \\
 c2c[8] &= c0 + c1 + c10 + c11 + c12 + c17 + c22 + c23 + c28 + c3 + c31 + c4 + c8 \\
 c2c[7] &= c0 + c10 + c15 + c16 + c2 + c21 + c22 + c23 + c24 + c25 + c28 + c29 + c3 + c5 + c7 + c8 \\
 c2c[6] &= c1 + c11 + c14 + c2 + c20 + c21 + c22 + c25 + c29 + c30 + c4 + c5 + c6 + c7 + c8 \\
 c2c[5] &= c0 + c1 + c10 + c13 + c19 + c20 + c21 + c24 + c28 + c29 + c3 + c4 + c5 + c6 + c7 \\
 c2c[4] &= c0 + c11 + c12 + c15 + c18 + c19 + c2 + c20 + c24 + c25 + c29 + c3 + c30 + c31 + c4 + c6 + c8 \\
 c2c[3] &= c1 + c10 + c14 + c15 + c17 + c18 + c19 + c2 + c25 + c27 + c3 + c31 + c7 + c8 + c9 \\
 c2c[2] &= c0 + c1 + c13 + c14 + c16 + c17 + c18 + c2 + c24 + c26 + c30 + c31 + c6 + c7 + c8 + c9 \\
 c2c[1] &= c0 + c1 + c11 + c12 + c13 + c16 + c17 + c24 + c27 + c28 + c6 + c7 + c9 \\
 c2c[0] &= c0 + c10 + c12 + c16 + c24 + c25 + c26 + c28 + c29 + c30 + c31 + c6 + c9
 \end{aligned}$$

[0012] 优选的，D2C 编码模块将输入的 32 位并行数据按以下编码方式进行编码，其中，+ 表示二进制异或运算；

[0013]

$$\begin{aligned}
 d2c[31] &= d12 + d14 + d15 + d16 + d2 + d24 + d25 + d26 + d27 + d28 + d30 + d31 + d8 \\
 d2c[30] &= d0 + d13 + d15 + d16 + d17 + d25 + d26 + d27 + d28 + d29 + d3 + d31 + d9 \\
 d2c[29] &= d0 + d1 + d10 + d14 + d16 + d17 + d18 + d24 + d26 + d27 + d28 + d29 + d30 + d4 \\
 d2c[28] &= d1 + d11 + d15 + d17 + d18 + d19 + d2 + d25 + d27 + d28 + d29 + d30 + d31 + d5 \\
 d2c[27] &= d0 + d12 + d16 + d18 + d19 + d2 + d20 + d26 + d28 + d29 + d3 + d30 + d31 + d6 \\
 d2c[26] &= d1 + d13 + d16 + d17 + d19 + d20 + d21 + d24 + d27 + d29 + d3 + d30 + d31 + d4 + d7 \\
 d2c[25] &= d12 + d15 + d17 + d18 + d20 + d21 + d22 + d24 + d26 + d27 + d4 + d5 + d8 \\
 d2c[24] &= d0 + d13 + d18 + d19 + d21 + d22 + d23 + d25 + d27 + d28 + d5 + d6 + d9 \\
 d2c[23] &= d1 + d10 + d14 + d19 + d20 + d22 + d23 + d24 + d26 + d28 + d29 + d6 + d7 + d8 \\
 d2c[22] &= d11 + d12 + d14 + d16 + d20 + d21 + d23 + d24 + d26 + d28 + d29 + d31 + d7 + d9 \\
 d2c[21] &= d10 + d13 + d14 + d17 + d2 + d21 + d22 + d24 + d26 + d28 + d29 + d31 \\
 d2c[20] &= d11 + d14 + d15 + d16 + d18 + d22 + d23 + d25 + d27 + d29 + d3 + d30 \\
 d2c[19] &= d0 + d12 + d15 + d17 + d19 + d23 + d26 + d28 + d30 + d31 + d4 + d8 \\
 d2c[18] &= d0 + d1 + d13 + d16 + d18 + d20 + d24 + d27 + d29 + d31 + d5 + d8 + d9 \\
 d2c[17] &= d1 + d10 + d14 + d16 + d17 + d19 + d2 + d21 + d24 + d25 + d28 + d30 + d6 + d9 \\
 d2c[16] &= d10 + d11 + d15 + d17 + d18 + d2 + d20 + d22 + d25 + d26 + d29 + d3 + d31 + d7 \\
 d2c[15] &= d0 + d11 + d14 + d15 + d18 + d19 + d2 + d21 + d23 + d25 + d28 + d3 + d31 + d4 + d8 \\
 d2c[14] &= d0 + d1 + d12 + d15 + d16 + d19 + d20 + d22 + d26 + d29 + d3 + d4 + d5 + d8 + d9 \\
 d2c[13] &= d0 + d1 + d10 + d13 + d17 + d2 + d20 + d21 + d23 + d24 + d27 + d30 + d4 + d5 + d6 + d9 \\
 d2c[12] &= d1 + d10 + d11 + d14 + d18 + d2 + d21 + d22 + d24 + d25 + d28 + d3 + d31 + d5 + d6 + d7 + d8 \\
 d2c[11] &= d11 + d14 + d19 + d22 + d23 + d24 + d27 + d28 + d29 + d3 + d30 + d31 + d4 + d6 + d7 + d8 + d9 \\
 d2c[10] &= d10 + d14 + d2 + d20 + d23 + d24 + d26 + d27 + d29 + d4 + d5 + d7 + d9 \\
 d2c[9] &= d10 + d11 + d12 + d14 + d16 + d2 + d21 + d26 + d3 + d31 + d5 + d6 \\
 d2c[8] &= d11 + d12 + d13 + d15 + d16 + d17 + d22 + d24 + d27 + d3 + d4 + d6 + d7 \\
 d2c[7] &= d0 + d13 + d15 + d16 + d17 + d18 + d2 + d23 + d26 + d27 + d30 + d31 + d4 + d5 + d7 + d8 \\
 d2c[6] &= d0 + d1 + d12 + d15 + d17 + d18 + d19 + d2 + d25 + d26 + d3 + d30 + d5 + d6 + d9 \\
 d2c[5] &= d0 + d1 + d10 + d13 + d18 + d19 + d2 + d20 + d26 + d27 + d3 + d31 + d4 + d6 + d7 \\
 d2c[4] &= d1 + d11 + d12 + d15 + d19 + d20 + d21 + d24 + d25 + d26 + d3 + d30 + d31 + d4 + d5 + d7 + d8 \\
 d2c[3] &= d0 + d13 + d14 + d15 + d20 + d21 + d22 + d24 + d28 + d30 + d4 + d5 + d6 + d8 + d9 \\
 d2c[2] &= d0 + d1 + d10 + d14 + d15 + d21 + d22 + d23 + d24 + d25 + d29 + d31 + d5 + d6 + d7 + d9 \\
 d2c[1] &= d0 + d1 + d10 + d11 + d12 + d14 + d22 + d23 + d27 + d28 + d31 + d6 + d7 \\
 d2c[0] &= d1 + d11 + d13 + d14 + d23 + d24 + d25 + d26 + d27 + d29 + d30 + d31 + d7
 \end{aligned}$$

[0014] 优选的, 整数字节编码模块将输入初始 CRC 值对输入值按以下编码方式进行编码, 其中, + 表示二进制异或运算;

[0015]


```

crc_8[31]=c23 + c29 + d2
crc_8[30]=c22 + c28 + c31 + d0 + d3
crc_8[29]=c21 + c27 + c30 + c31 + d0 + d1 + d4
crc_8[28]=c20 + c26 + c29 + c30 + d1 + d2 + d5
crc_8[27]=c19 + c25 + c28 + c29 + c31 + d0 + d2 + d3 + d6
crc_8[26]=c18 + c24 + c27 + c28 + c30 + d1 + d3 + d4 + d7
crc_8[25]=c17 + c26 + c27 + d4 + d5
crc_8[24]=c16 + c25 + c26 + c31 + d0 + d5 + d6
crc_8[23]=c15 + c24 + c25 + c30 + d1 + d6 + d7
crc_8[22]=c14 + c24 + d7
crc_8[21]=c13 + c29 + d2
crc_8[20]=c12 + c28 + d3
crc_8[19]=c11 + c27 + c31 + d0 + d4
crc_8[18]=c10 + c26 + c30 + c31 + d0 + d1 + d5
crc_8[17]=c25 + c29 + c30 + c9 + d1 + d2 + d6
crc_8[16]=c24 + c28 + c29 + c8 + d2 + d3 + d7
crc_8[15]=c27 + c28 + c29 + c31 + c7 + d0 + d2 + d3 + d4
crc_8[14]=c26 + c27 + c28 + c30 + c31 + c6 + d0 + d1 + d3 + d4 + d5
crc_8[13]=c25 + c26 + c27 + c29 + c30 + c31 + c5 + d0 + d1 + d2 + d4 + d5 + d6
crc_8[12]=c24 + c25 + c26 + c28 + c29 + c30 + c4 + d1 + d2 + d3 + d5 + d6 + d7
crc_8[11]=c24 + c25 + c27 + c28 + c3 + d3 + d4 + d6 + d7
crc_8[10]=c2 + c24 + c26 + c27 + c29 + d2 + d4 + d5 + d7
crc_8[9]=c1 + c25 + c26 + c28 + c29 + d2 + d3 + d5 + d6
crc_8[8]=c0 + c24 + c25 + c27 + c28 + d3 + d4 + d6 + d7
crc_8[7]=c24 + c26 + c27 + c29 + c31 + d0 + d2 + d4 + d5 + d7
crc_8[6]=c25 + c26 + c28 + c29 + c30 + c31 + d0 + d1 + d2 + d3 + d5 + d6
crc_8[5]=c24 + c25 + c27 + c28 + c29 + c30 + c31 + d0 + d1 + d2 + d3 + d4 + d6 + d7
crc_8[4]=c24 + c26 + c27 + c28 + c30 + d1 + d3 + d4 + d5 + d7
crc_8[3]=c25 + c26 + c27 + c31 + d0 + d4 + d5 + d6
crc_8[2]=c24 + c25 + c26 + c30 + c31 + d0 + d1 + d5 + d6 + d7
crc_8[1]=c24 + c25 + c30 + c31 + d0 + d1 + d6 + d7
crc_8[0]=c24 + c30 + d1 + d7

```

[0016] 与现有技术相比,本发明具有以下有益的技术效果:

[0017] 本发明提供的面向 128 位并行输入的 CRC-32 校验电路,通过基于预处理矩阵的硬件电路对 4 组 32 位并行数据分别进行编码,对 CRC 寄存器中的 CRC 值进行多级编码操作。将得到的各级编码结果分别与 4 组并行数据的预处理结果进行异或运算,在整数字节编码模块中对数据和各级 CRC 值编码结果进行进一步编码运算得到非整字的编码结果,最终通过 next_crc 选择模块选择出最终的编码结果,通过反相器得到 CRC 值,通过比较器得到校验结果。与现有的 128 位并行数据校验电路相比,在基本不增加硬件成本、不降低整体系统性能的基础上,提供了一种高效率的 CRC 编解码电路,大大提高了实时处理数据的速度,提高了数据的传输效率。能够实现对 8*N 位(如 8 位、16 位、24 位、32 位...96 位、128 位)并

行数据的 CRC 编码校验。

附图说明

- [0018] 图 1 为本发明的编码校验电路示意图。
[0019] 图 2 为基于本发明编码校验电路的编码时序图。
[0020] 图 3 为基于本发明编码校验电路的解码时序图。
[0021] 图 4 为基于 synopsys 公司的 8 位并行数据校验电路的功能时序图。

具体实施方式

[0022] 下面结合附图对本发明做进一步的详细说明,所述是对本发明的解释而不是限定。

[0023] 本发明提出了一种面向 128 位并行输入的 CRC-32 校验电路,该电路将并行的 128 位数据分为 4 组 32 位并行数据,通过基于预处理矩阵的硬件电路对 4 组 32 位并行数据分别进行编码。与此同时,该电路对 CRC 寄存器中的 CRC 值进行多级编码操作,将得到的各级编码结果分别与 4 组并行数据的预处理结果进行异或运算,分别得到关于并行 32、64、94、128 位数据的校验结果。之后,以 CRC 寄存器中的 CRC 值和并行 32、64、96 位数据的校验结果作为 CRC 选择模块的输入,根据字节有效信号的高两位进行选择;同时将 4 组 32 位并行数据作为数据选择模块的输入,根据字节有效信号的高两位进行选择,选择出一组 32 位并行数据。将 CRC 选择模块的选择结果作为整数字节编码模块的初始 CRC 值,将数据选择模块的选择结果作为整数字节编码模块的数据输入,分别对 32 位并行数据的高 8 位、高 16 位、高 24 位数据输入进行编码得到三个分别关于高 8 位、高 16 位、高 24 位(所述高 8、16、24 位是针对整数字节编码模块的 32 位输入)的 CRC 校验结果。最终根据字节有效信号选择出最终的 CRC 校验结果。

[0024] 本发明专利是通过以下技术方案来实现:

[0025] 一种面向 128 位并行输入的 CRC-32 校验电路,该电路主要包括:

[0026] CRC 寄存器,寄存 32 位 CRC 值,输入来自 Next_crc 多路选择器,CRC 时钟信号 clk, CRC 复位信号 rst,字节有效信号 be[3:0],输出为寄存器所寄存的 CRC 值。其中字节有效信号 be[3:0] 跟数据输入信号 data[127:0] 相关,具体关系为:be[3:0] = 4'b0000,表示数据输入 data[127:120] 有效;be[3:0] = 4'b0001,表示数据输入 data[127:112] 有效;... be[3:0] = 4'b1111,表示数据输入 data[127:0] 有效。

[0027] C2C_0 编码模块,将来自 CRC 寄存器中的 32 位 CRC 值,按照以下编码方式进行编码(+ 表示二进制异或运算)

[0028]

```

c2c[31]=c11+c15+c23+c24+c25+c27+c28+c29+c30+c31+c5+c8+c9
c2c[30]=c10+c14+c22+c23+c24+c26+c27+c28+c29+c30+c4+c7+c8
c2c[29]=c13+c21+c22+c23+c25+c26+c27+c28+c29+c3+c31+c6+c7+c9
c2c[28]=c12+c2+c20+c21+c22+c24+c25+c26+c27+c28+c30+c5+c6+c8
c2c[27]=c1+c11+c19+c20+c21+c23+c24+c25+c26+c27+c29+c4+c5+c7
c2c[26]=c0+c10+c18+c19+c20+c22+c23+c24+c25+c26+c28+c3+c31+c4+c6
c2c[25]=c11+c15+c17+c18+c19+c2+c21+c22+c28+c29+c3+c31+c8
c2c[24]=c1+c10+c14+c16+c17+c18+c2+c20+c21+c27+c28+c30+c7
c2c[23]=c0+c1+c13+c15+c16+c17+c19+c20+c26+c27+c29+c31+c6+c9
c2c[22]=c0+c11+c12+c14+c16+c18+c19+c23+c24+c26+c27+c29+c31+c9
c2c[21]=c10+c13+c17+c18+c22+c24+c26+c27+c29+c31+c5+c9
c2c[20]=c12+c16+c17+c21+c23+c25+c26+c28+c30+c4+c8+c9
c2c[19]=c11+c15+c16+c20+c22+c24+c25+c27+c29+c3+c7+c8
c2c[18]=c10+c14+c15+c19+c2+c21+c23+c24+c26+c28+c31+c6+c7
c2c[17]=c1+c13+c14+c18+c20+c22+c23+c25+c27+c30+c31+c5+c6+c9
c2c[16]=c0+c12+c13+c17+c19+c21+c22+c24+c26+c29+c30+c4+c5+c8
c2c[15]=c12+c15+c16+c18+c20+c21+c24+c27+c3+c30+c4+c5+c7+c8+c9
c2c[14]=c11+c14+c15+c17+c19+c2+c20+c23+c26+c29+c3+c4+c6+c7+c8
c2c[13]=c1+c10+c13+c14+c16+c18+c19+c2+c22+c25+c28+c3+c31+c5+c6+c7
c2c[12]=c0+c1+c12+c13+c15+c17+c18+c2+c21+c24+c27+c30+c31+c4+c5+c6+c9
c2c[11]=c0+c1+c12+c14+c15+c16+c17+c20+c24+c25+c26+c27+c28+c3+c31+c4+c9
c2c[10]=c0+c13+c14+c16+c19+c2+c26+c28+c29+c3+c31+c5+c9
c2c[9]=c1+c11+c12+c13+c18+c2+c23+c24+c29+c4+c5+c9
c2c[8]=c0+c1+c10+c11+c12+c17+c22+c23+c28+c3+c31+c4+c8
c2c[7]=c0+c10+c15+c16+c2+c21+c22+c23+c24+c25+c28+c29+c3+c5+c7+c8
c2c[6]=c1+c11+c14+c2+c20+c21+c22+c25+c29+c30+c4+c5+c6+c7+c8
c2c[5]=c0+c1+c10+c13+c19+c20+c21+c24+c28+c29+c3+c4+c5+c6+c7
c2c[4]=c0+c11+c12+c15+c18+c19+c2+c20+c24+c25+c29+c3+c30+c31+c4+c6+c8
c2c[3]=c1+c10+c14+c15+c17+c18+c19+c2+c25+c27+c3+c31+c7+c8+c9
c2c[2]=c0+c1+c13+c14+c16+c17+c18+c2+c24+c26+c30+c31+c6+c7+c8+c9
c2c[1]=c0+c1+c11+c12+c13+c16+c17+c24+c27+c28+c6+c7+c9
c2c[0]=c0+c10+c12+c16+c24+c25+c26+c28+c29+c30+c31+c6+c9

```

[0029] 并将 32 位结果输出至 Xor_1 模块。

[0030] D2C_1 编码模块, 将来自外部 128 位并行数据 data[127:0] 中的高 32 位 (data[127:96]) 按照以下编码方式进行编码 (+ 表示二进制异或运算)。

[0031]

```

d2c[31]=d12+d14+d15+d16+d2+d24+d25+d26+d27+d28+d30+d31+d8
d2c[30]=d0+d13+d15+d16+d17+d25+d26+d27+d28+d29+d3+d31+d9
d2c[29]=d0+d1+d10+d14+d16+d17+d18+d24+d26+d27+d28+d29+d30+d4
d2c[28]=d1+d11+d15+d17+d18+d19+d2+d25+d27+d28+d29+d30+d31+d5
d2c[27]=d0+d12+d16+d18+d19+d2+d20+d26+d28+d29+d3+d30+d31+d6
d2c[26]=d1+d13+d16+d17+d19+d20+d21+d24+d27+d29+d3+d30+d31+d4+d7
d2c[25]=d12+d15+d17+d18+d20+d21+d22+d24+d26+d27+d4+d5+d8
d2c[24]=d0+d13+d18+d19+d21+d22+d23+d25+d27+d28+d5+d6+d9
d2c[23]=d1+d10+d14+d19+d20+d22+d23+d24+d26+d28+d29+d6+d7+d8
d2c[22]=d11+d12+d14+d16+d20+d21+d23+d24+d26+d28+d29+d31+d7+d9
d2c[21]=d10+d13+d14+d17+d2+d21+d22+d24+d26+d28+d29+d31
d2c[20]=d11+d14+d15+d16+d18+d22+d23+d25+d27+d29+d3+d30
d2c[19]=d0+d12+d15+d17+d19+d23+d26+d28+d30+d31+d4+d8
d2c[18]=d0+d1+d13+d16+d18+d20+d24+d27+d29+d31+d5+d8+d9
d2c[17]=d1+d10+d14+d16+d17+d19+d2+d21+d24+d25+d28+d30+d6+d9
d2c[16]=d10+d11+d15+d17+d18+d2+d20+d22+d25+d26+d29+d3+d31+d7
d2c[15]=d0+d11+d14+d15+d18+d19+d2+d21+d23+d25+d28+d3+d31+d4+d8
d2c[14]=d0+d1+d12+d15+d16+d19+d20+d22+d26+d29+d3+d4+d5+d8+d9
d2c[13]=d0+d1+d10+d13+d17+d2+d20+d21+d23+d24+d27+d30+d4+d5+d6+d9
d2c[12]=d1+d10+d11+d14+d18+d2+d21+d22+d24+d25+d28+d3+d31+d5+d6+d7+d8
d2c[11]=d11+d14+d19+d22+d23+d24+d27+d28+d29+d3+d30+d31+d4+d6+d7+d8+d9
d2c[10]=d10+d14+d2+d20+d23+d24+d26+d27+d29+d4+d5+d7+d9
d2c[9]=d10+d11+d12+d14+d16+d2+d21+d26+d3+d31+d5+d6
d2c[8]=d11+d12+d13+d15+d16+d17+d22+d24+d27+d3+d4+d6+d7
d2c[7]=d0+d13+d15+d16+d17+d18+d2+d23+d26+d27+d30+d31+d4+d5+d7+d8
d2c[6]=d0+d1+d12+d15+d17+d18+d19+d2+d25+d26+d3+d30+d5+d6+d9
d2c[5]=d0+d1+d10+d13+d18+d19+d2+d20+d26+d27+d3+d31+d4+d6+d7
d2c[4]=d1+d11+d12+d15+d19+d20+d21+d24+d25+d26+d3+d30+d31+d4+d5+d7+d8
d2c[3]=d0+d13+d14+d15+d20+d21+d22+d24+d28+d30+d4+d5+d6+d8+d9
d2c[2]=d0+d1+d10+d14+d15+d21+d22+d23+d24+d25+d29+d31+d5+d6+d7+d9
d2c[1]=d0+d1+d10+d11+d12+d14+d22+d23+d27+d28+d31+d6+d7
d2c[0]=d1+d11+d13+d14+d23+d24+d25+d26+d27+d29+d30+d31+d7

```

[0032] 将得到的 32 位预处理结果输出至 Xor_1 模块。

[0033] D2C_2 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的 95 至 64 位数据 (data[95:64]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_2 模块。

[0034] D2C_3 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的 63 至 32 位数据 (data[63:32]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_3 模块。

[0035] D2C_4 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的低 32 位数据 (即 data[31:0]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_4 模块。

[0036] Xor_1 模块,其输入分别来自 C2C_0 编码模块和 D2C_1 级编码模块,将两组 32 位数据按位进行异或,得到结果 c_1[31:0],c_1[31:0] 为并行 32 位数据 (即并行高 32 位 data[127:96]) 的 CRC 校验值。其输出至 CRC 选择模块、C2C_1 编码模块和 next_crc 选择模块。

[0037] C2C_1 编码模块,编码方式同 C2C_0 编码模块,输入来自 Xor_1 模块,将数据按规则进行编码,得到的结果输出至 Xor_2 模块。

[0038] Xor_2 模块,其输入分别来自 C2C_1 编码模块和 D2C_2 编码模块,得到结果 c_2[31:0],c_2[31:0] 为并行 64 位数据 (即并行高 64 位 data[127:64]) 的 CRC 校验值,输出至 CRC 选择模块、C2C_2 级编码模块和 next_crc 选择模块。

[0039] C2C_2 编码模块, 编码方式同 C2C_0 编码模块, 输入来自 Xor_2 模块, 将数据按规则进行编码, 得到的结果输出至 Xor_3 模块。

[0040] Xor_3, 其输入分别来自 C2C_2 编码模块和 D2C_3 编码模块, 得到结果 $c_3[31:0]$, $c_3[31:0]$ 为并行 96 位数据 (即并行高 96 位 $data[127:32]$) 的 CRC 校验值, 输出至 CRC 选择模块、C2C_3 编码模块和 next_crc 选择模块。

[0041] C2C_3 编码模块, 编码方式同 C2C_0 编码模块, 输入来自 Xor_3 模块, 将数据按规则进行编码, 得到的结果输出至 Xor_4 模块。

[0042] Xor_4 模块, 其输入分别来自 C2C_3 编码模块和 D2C_4 编码模块, 得到结果 $c_4[31:0]$, $c_4[31:0]$ 为并行 128 位数据的 CRC 校验值, 输出至 CRC 选择模块和 next_crc 选择模块。

[0043] CRC 选择模块, 输入分别来自 CRC 寄存器、Xor_1 模块的 $c_1[31:0]$ 、Xor_2 模块的 $c_2[31:0]$ 、Xor_3 模块的 $c_3[31:0]$ 、be 接口信号高两位 ($be[3:2]$), 根据 be 接口信号的高两位进行选择。输出 32 位数据至 CRC_8 模块。be 信号为字节有效信号, CRC 选择模块的选择规则为: $be[3:2]$ 为 0x00 时, 输出 CRC 寄存器的值, $be[3:2]$ 为 0x01 时, 输出 $c_1[31:0]$, $be[3:2]$ 为 0x10 时, 输出 $c_2[31:0]$, $be[3:2]$ 为 0x11 时, 输出 $c_3[31:0]$ 。

[0044] 数据选择模块, 输入分别来自 4 组数据输入 D1($data[127:96]$)、D2($data[95:64]$)、D3($data[63:32]$)、D4($data[31:0]$) 和 be 接口信号高两位 ($be[3:2]$), 根据 be 接口信号的高两位进行选择, 输出 32 位数据至 CRC_8 模块。数据选择模块的选择规则为: $be[3:2]$ 为 0x00 时, 输出 D1, $be[3:2]$ 为 0x01 时, 输出 D2, $be[3:2]$ 为 0x10 时, 输出 D3, $be[3:2]$ 为 0x11 时, 输出 D4。

[0045] 整数字节编码模块, 其输入来自 CRC 选择模块和数据选择模块, 将 crc 选择模块的值作为初始 crc 值, 将数据选择模块输出的 32 位值, 取其高 24 位, 并分为 3 组, 按规则进行编码, 得到三个 CRC 编码结果, 并输出至 next_crc 选择模块。按照以下编码方式 (+ 表示二进制异或运算)。

[0046]

```

crc_8[31]=c23+c29+d2
crc_8[30]=c22+c28+c31+d0+d3
crc_8[29]=c21+c27+c30+c31+d0+d1+d4
crc_8[28]=c20+c26+c29+c30+d1+d2+d5
crc_8[27]=c19+c25+c28+c29+c31+d0+d2+d3+d6
crc_8[26]=c18+c24+c27+c28+c30+d1+d3+d4+d7
crc_8[25]=c17+c26+c27+d4+d5
crc_8[24]=c16+c25+c26+c31+d0+d5+d6
crc_8[23]=c15+c24+c25+c30+d1+d6+d7
crc_8[22]=c14+c24+d7
crc_8[21]=c13+c29+d2
crc_8[20]=c12+c28+d3
crc_8[19]=c11+c27+c31+d0+d4
crc_8[18]=c10+c26+c30+c31+d0+d1+d5
crc_8[17]=c25+c29+c30+c9+d1+d2+d6
crc_8[16]=c24+c28+c29+c8+d2+d3+d7
crc_8[15]=c27+c28+c29+c31+c7+d0+d2+d3+d4
crc_8[14]=c26+c27+c28+c30+c31+c6+d0+d1+d3+d4+d5
crc_8[13]=c25+c26+c27+c29+c30+c31+c5+d0+d1+d2+d4+d5+d6
crc_8[12]=c24+c25+c26+c28+c29+c30+c4+d1+d2+d3+d5+d6+d7
crc_8[11]=c24+c25+c27+c28+c3+d3+d4+d6+d7
crc_8[10]=c2+c24+c26+c27+c29+d2+d4+d5+d7
crc_8[9]=c1+c25+c26+c28+c29+d2+d3+d5+d6
crc_8[8]=c0+c24+c25+c27+c28+d3+d4+d6+d7
crc_8[7]=c24+c26+c27+c29+c31+d0+d2+d4+d5+d7
crc_8[6]=c25+c26+c28+c29+c30+c31+d0+d1+d2+d3+d5+d6
crc_8[5]=c24+c25+c27+c28+c29+c30+c31+d0+d1+d2+d3+d4+d6+d7
crc_8[4]=c24+c26+c27+c28+c30+d1+d3+d4+d5+d7
crc_8[3]=c25+c26+c27+c31+d0+d4+d5+d6
crc_8[2]=c24+c25+c26+c30+c31+d0+d1+d5+d6+d7
crc_8[1]=c24+c25+c30+c31+d0+d1+d6+d7
crc_8[0]=c24+c30+d1+d7

```

[0047] next_crc 选择模块,其输入来自整数字节编码模块、Xor_1 模块、Xor_2 模块、Xor_3 模块、Xor_4 模块、be 接口信号。根据 be 信号选择下一级的 crc 编码结果,be 信号跟数据输入的位宽有关,选择得到的结果即当前并行输入数据所校验的结果。其结果输出至 crc 寄存器。如图 1 所示,整数字节编码模块的结果为 cs_1,cs_2,cs_3,Xor_1 模块输出为 c_1、Xor_2 模块输出为 c_2、Xor_3 模块输出为 c_3、Xor_4 模块输出为 c_4。next_crc 选择模块的选择规则为:判断 be[1:0] 是否为 2' b11,如果是则判断 be[3:2],be[3:2] 为 2' b00,输出 c_1;be[3:2] 为 2' b01,输出 c_2;be[3:2] 为 2' b10,输出结果 c_3;be[3:2] 为 2' b11,输出结果 c_4;如果 be[1:0] 不为 2' b11,则判断 be[1:0],be[1:0] 为 2' b00,输出 cs_1;be[1:0] 为 2' b01,输出 cs_2;be[1:0] 为 2' b10,输出 cs_3;

[0048] 反相器,将来自 CRC 寄存器中的 CRC 值按位取反,最终得到所需求的 crc 编码。

[0049] 比较器,将来自 CRC 寄存器中的 CRC 值与魔数 (magic number, 在 CRC-32 的校验结果中为定值,在本发明中其值为 32' hc704dd7b) 相比,将结果输出至 CRC_match 接口信号,信号为 1 则说明校验成功,为 0 则说明校验失败。

[0050] 具体的,如图 1 所示,一种面向 128 位并行输入的 CRC-32 校验电路,接口信号主要有:数据信号 data[127:0]、字节使能信号 be[3:0]、时钟 clk、复位信号 rst、CRC 输出结果、校验匹配结果 CRC_match。主要模块有: CRC 寄存器、C2C_0 编码模块、C2C_1 编码模块、C2C_2 编码模块、C2C_3 编码模块、D2C_1 编码模块、D2C_2 编码模块、D2C_3 编码模块、D2C_4 编码模块、异或模块、多路选择模块、比较器、反相器。

[0051] CRC 寄存器,寄存 32 位 CRC 值,输入来自 Next_crc 多路选择器,CRC 时钟信号 clk,CRC 复位信号 rst,字节有效信号 be[3:0],输出为寄存器所寄存的 CRC 值。其中字节有效信号 be[3:0] 跟数据输入信号 data[127:0] 相关,具体关系为:be[3:0] = 4'b0000,表示数据输入 data[127:120] 有效;be[3:0] = 4'b0001,表示数据输入 data[127:112] 有效;... be[3:0] = 4'b1111,表示数据输入 data[127:0] 有效。

[0052] CRC 寄存器中的 CRC 值作为下一次数据输入的前一级 CRC 值,在对每一次数据进行校验之前,将 CRC 寄存器内的值进行初始化操作,将 32 位寄存器的值赋 0xFFFFFFFF,当数据有效信号有效时(即为 1 时),当时钟上升沿到来时,将计算的结果作为输入更新 CRC 寄存器。

[0053] C2C_0 编码模块,将来自 CRC 寄存器中的 32 位 CRC 值,按照其编码方式进行二进制异或运算的编码;并将 32 位结果输出至 Xor_1 模块。

[0054] 如上述 C2C_0 编码模块,按照其编码方式对 CRC 寄存器中的 32 位 CRC 值进行编码,此次编码作为对前一级 CRC 值的预处理操作。将得到的编码结果输出至 Xor_1 模块与 D2C_1 编码模块所得结果进行异或。

[0055] D2C_1 编码模块,将来自外部 128 位并行数据 data[127:0] 中的高 32 位(data[127:96]) 按照其编码方式进行二进制异或运算的编码。

[0056] 将得到的 32 位预处理结果输出至 Xor_1 模块。

[0057] 如图 1 所示,D2C_1 编码模块对 data[127:96] 进行预处理,与 C2C_0 编码模块同时进行,将得到结果输出至 Xor_1 模块与 C2C_0 编码模块所得结果进行异或。

[0058] D2C_2 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的 95 至 64 位数据(data[95:64]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_2 模块。

[0059] 如图 1 所示,D2C_2 编码模块对 data[95:64] 进行预处理。

[0060] D2C_3 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的 63 至 32 位数据(data[63:32]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_3 模块。

[0061] 如图 1 所示,D2C_3 编码模块对 data[63:32] 进行预处理。

[0062] D2C_4 编码模块,其编码方式同 D2C_1 编码模块,将来自外部 128 位并行数据中的低 32 位数据(即 data[31:0]) 进行编码,并将得到的 32 位预处理结果输出至 Xor_4 模块。

[0063] 如图 1 所示,D2C_4 编码模块对 data[31:0] 进行预处理

[0064] Xor_1 模块,其输入分别来自 C2C_0 编码模块和 D2C_1 级编码模块,将两组 32 位数据按位进行异或,得到结果 c_1[31:0],c_1[31:0] 为并行 32 位数据(即并行高 32 位 data[127:96]) 的 CRC 校验值。其输出至 CRC 选择模块、C2C_1 编码模块和 next_crc 选择模块。

[0065] 如图 1 所示,Xor_1 模块所得结果为 c_1[31:0],c_1[31:0] 作为对 data[127:96] 进行校验的结果输出至 C2C_1 编码模块,作为 C2C_1 编码模块的 CRC 处置已

进行下一级的编码运算。

[0066] C2C_1 编码模块, 编码方式同 C2C_0 编码模块, 输入来自 Xor_1 模块, 将数据按规则进行编码, 得到的结果输出至 Xor_2 模块。

[0067] 将 $c_1[31:0]$ 作为前一级的 CRC 值进行预处理运算。

[0068] Xor_2 模块, 其输入分别来自 C2C_1 编码模块和 D2C_2 编码模块, 得到结果 $c_2[31:0]$, $c_2[31:0]$ 为并行 64 位数据 (即并行高 64 位 $data[127:64]$) 的 CRC 校验值, 输出至 CRC 选择模块、C2C_2 级编码模块和 next_crc 选择模块。

[0069] 如图 1 所示, Xor_2 模块所得到结果为 $c_2[31:0]$, $c_2[31:0]$ 作为对 $data[127:64]$ 进行校验的结果输出至 C2C_2 编码模块, 作为 C2C_2 编码模块的 CRC 初值进行下一级的编码运算。

[0070] C2C_2 编码模块, 编码方式同 C2C_0 编码模块, 输入来自 Xor_2 模块, 将数据按规则进行编码, 得到的结果输出至 Xor_3 模块。

[0071] 将 $c_2[31:0]$ 作为前一级的 CRC 值进行预处理运算。

[0072] Xor_3, 其输入分别来自 C2C_2 编码模块和 D2C_3 编码模块, 得到结果 $c_3[31:0]$, $c_3[31:0]$ 为并行 96 位数据 (即并行高 96 位 $data[127:32]$) 的 CRC 校验值, 输出至 CRC 选择模块、C2C_3 编码模块和 next_crc 选择模块。

[0073] 如图 1 所示, Xor_3 模块所得到结果为 $c_3[31:0]$, $c_3[31:0]$ 作为对 $data[127:32]$ 进行校验的结果输出至 C2C_3 编码模块, 作为 C2C_3 编码模块的 CRC 初值进行下一级的编码运算。

[0074] C2C_3 编码模块, 编码方式同 C2C_0 编码模块, 输入来自 Xor_3 模块, 将数据按规则进行编码, 得到的结果输出至 Xor_4 模块。

[0075] 将 $c_3[31:0]$ 作为前一级的 CRC 值进行预处理运算。

[0076] Xor_4 模块, 其输入分别来自 C2C_3 编码模块和 D2C_4 编码模块, 得到结果 $c_4[31:0]$, $c_4[31:0]$ 为并行 128 位数据的 CRC 校验值, 输出至 CRC 选择模块和 next_crc 选择模块。

[0077] 如图 1 所示, Xor_4 模块所得到结果为 $c_4[31:0]$, $c_3[31:0]$ 作为对 $data[127:0]$ 进行校验的结果。

[0078] CRC 选择模块, 输入分别来自 CRC 寄存器、Xor_1 模块的 $c_1[31:0]$ 、Xor_2 模块的 $c_2[31:0]$ 、Xor_3 模块的 $c_3[31:0]$ 、be 接口信号高两位 ($be[3:2]$), 根据 be 接口信号的高两位进行选择。输出 32 位数据至 CRC_8 模块。be 信号为字节有效信号, CRC 选择模块的选择规则为: $be[3:2]$ 为 0x00 时, 输出 CRC 寄存器的值, $be[3:2]$ 为 0x01 时, 输出 $c_1[31:0]$, $be[3:2]$ 为 0x10 时, 输出 $c_2[31:0]$, $be[3:2]$ 为 0x11 时, 输出 $c_3[31:0]$ 。

[0079] 如图 1 所示, 输入分别为: CRC 寄存器中的数值、 $c_1[31:0]$ 、 $c_2[31:0]$ 、 $c_3[31:0]$ 。根据 be 信号选择输出相应结果。

[0080] 数据选择模块, 输入分别来自 4 组数据输入 D1($data[127:96]$)、D2($data[95:64]$)、D3($data[63:32]$)、D4($data[31:0]$) 和 be 接口信号高两位 ($be[3:2]$), 根据 be 接口信号的高两位进行选择, 输出 32 位数据至 CRC_8 模块。数据选择模块的选择规则为: $be[3:2]$ 为 0x00 时, 输出 D1, $be[3:2]$ 为 0x01 时, 输出 D2, $be[3:2]$ 为 0x10 时, 输出 D3, $be[3:2]$ 为 0x11 时, 输出 D4。

[0081] 整数字节编码模块,其输入来自 CRC 选择模块和数据选择模块,将 crc 选择模块的值作为初始 crc 值,将数据选择模块输出的 32 位值,取其高 24 位,并分为 3 组,按规则进行编码,得到三个 CRC 编码结果,并输出至 next_crc 选择模块。按照其编码方式进行二进制异或运算。

[0082] 整数字节编码模块,采用并行 8 位数据的编码规则进行编码,将并行 8 位数据编码电路进行三级级联,32 位数据输入中的高 24 位分为 3 组 8 位并行数据依次作为级联电路中的三级数据输入。级联电路所得到的三级 CRC 编码结果分别如图 1 所表示的 cs_1, cs_2, cs_3, cs_1 为整数字节编码模块中高 8 位数据的校验结果, cs_2 为整数字节编码模块中高 16 位数据的校验结果, cs_3 为整数字节编码模块中高 24 位数据的校验结果。

[0083] next_crc 选择模块,其输入来自整数字节编码模块、Xor_1 模块、Xor_2 模块、Xor_3 模块、Xor_4 模块、be 接口信号。根据 be 信号选择下一级的 crc 编码结果, be 信号跟数据输入的位宽有关,选择得到的结果即当前并行输入数据所校验的结果。其结果输出至 crc 寄存器。如图 1 所示,整数字节编码模块的结果为 cs_1, cs_2, cs_3, Xor_1 模块输出为 c_1, Xor_2 模块输出为 c_2, Xor_3 模块输出为 c_3, Xor_4 模块输出为 c_4。next_crc 选择模块的选择规则为:判断 be[1:0] 是否为 2' b11,如果是则判断 be[3:2], be[3:2] 为 2' b00,输出 c_1 ;be[3:2] 为 2' b01,输出 c_2 ;be[3:2] 为 2' b10,输出结果 c_3 ;be[3:2] 为 2' b11,输出结果 c_4 ;如果 be[1:0] 不为 2' b11,则判断 be[1:0], be[1:0] 为 2' b00,输出 cs_1 ;be[1:0] 为 2' b01,输出 cs_2 ;be[1:0] 为 2' b10,输出 cs_3 ;

[0084] 当 be[1:0] 为 0x11 时,说明数据输入为字的倍数,然后根据 be[3:2] 进行选择输出结果,next_crc 选择模块选择的结果作为最终的 CRC 校验结果,由时钟触发寄存至 CRC 寄存器中。

[0085] 反相器,将来自 CRC 寄存器中的 CRC 值按位取反,最终得到所需求的 crc 编码。

[0086] 比较器,将来自 CRC 寄存器中的 CRC 值与魔数 (magic number, 在 CRC-32 的校验结果中为定值,在本发明中其值为 32' hc704dd7b) 相比,将结果输出至 CRC_match 接口信号,信号为 1 则说明校验成功,为 0 则说明校验失败。

[0087] 图 2 和图 3 为本发明编码校验电路的功能时序图,图 2 数据输入分为两个时钟周期,第一个时钟周期为并行 128 位,第二个时钟周期为并行 80 位;图 3 数据输入基于图 2 的校验结果,将 CRC 校验结果附加到图 2 中输入数据后面,其输入分为两个时钟周期,第一个时钟周期为并行 128 位,第二个时钟周期为并行 112 位;是对本发明有益效果的进一步说明。图 4 为基于 synopsys 公司的 8 位并行数据校验电路的功能时序图,其输入与图 2 的输入相同,分为 26 个时钟周期,每个时钟并行输入 8 位,结合图 2 和图 4 可知,本发明所公开电路的编码结果正确。

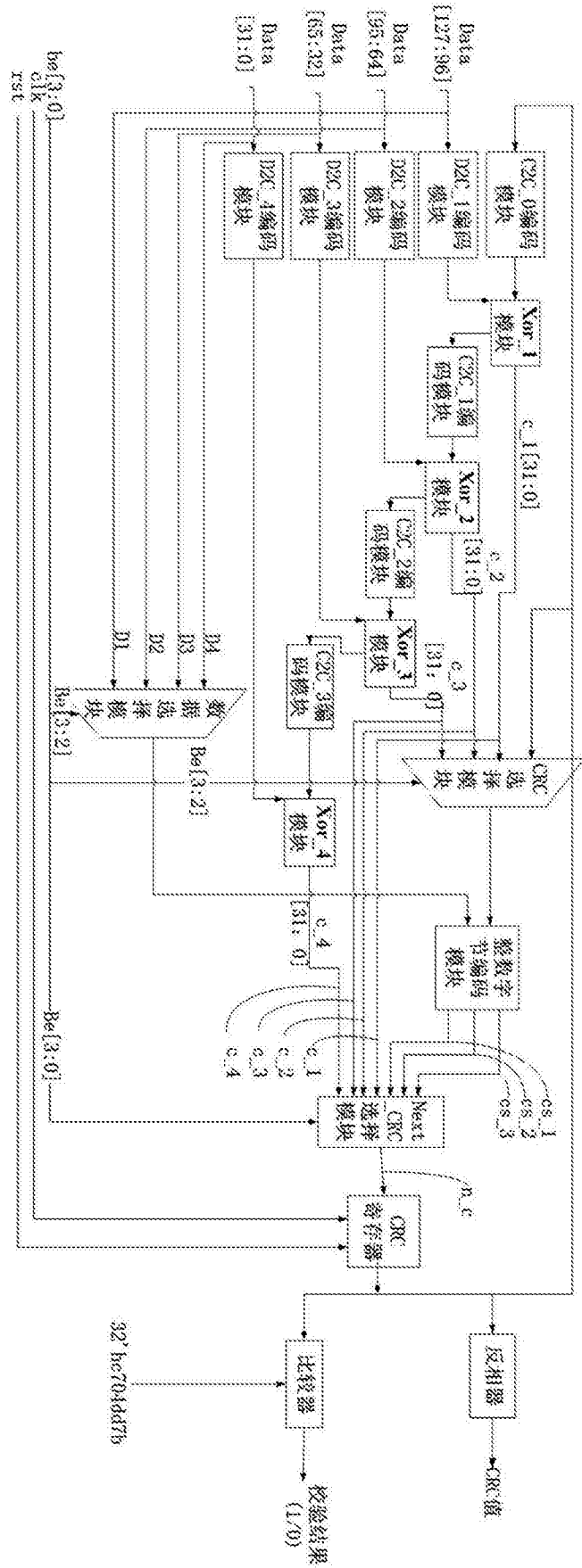


图 1

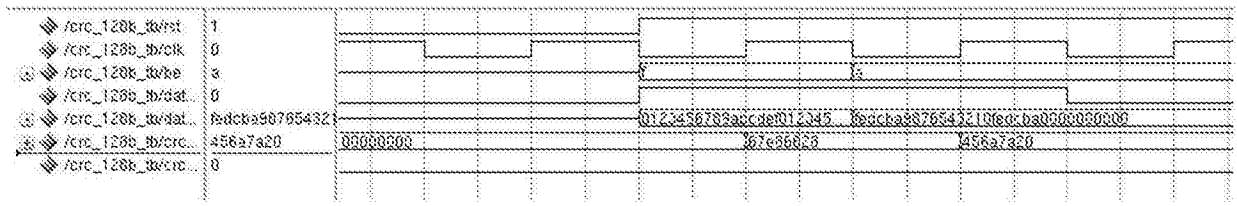


图 2

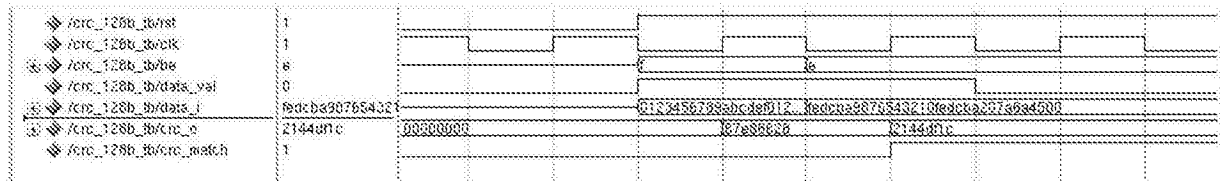


图 3

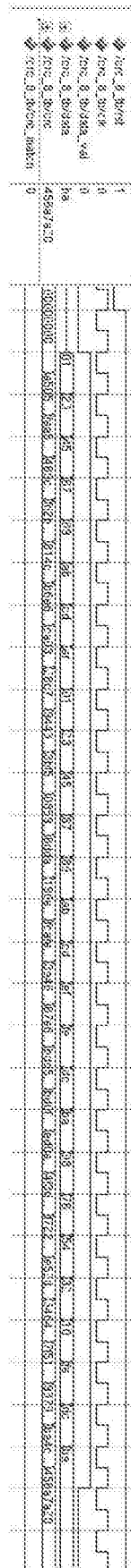


图 4