文章编号：2095-2163(2022)04-0035-06 中图分类号：TN911.22 文献标志码：A

Turbo 码中基 4 并行 QPP 交织器算法研究

史宜巧[1]，李 丛[2]
（1 江苏电子信息职业学院 智能制造学院，江苏 淮安 223003；2 南京理工大学 泰州科技学院，江苏 泰州 225300）

摘 要：为了最小化相邻比特之间的相关性，在 Turbo 编码过程中引入交织器。QPP 交织器作为无竞争交织器，结构灵活，性能优良，然而计算过程比较复杂，增加了硬件电路的设计难度。为了解决这个问题，本文提出了一种简化的 4 路并行基 4 交织器算法。将交织地址的计算简化为递推计算，并进一步推导了 4 路并行基 4 条件下交织器设计，设计了一种单入多出（Single Input Multiple Output，SIMO）交织器。最后对所设计的交织器进行 FPGA 实现，仿真结果表明该交织器每个时钟可以并行输出 8 个正确交织地址，提升了 Turbo 译码性能。

关键词：交织器；QPP；并行；基 4；单入多出

Research on radix-4 parallel QPP interleaver algorithm in Turbo codes

SHI Yiqiao[1], LI Cong[2]
(1 School of Intelligent Manufacturing , Jiangsu Vocational College of Electronics and Information Technology, Huai′an Jiangsu 223003, China; 2 Taizhou Institute, Nanjing University of Science and Technology, Taizhou Jiangsu 225300, China)

【Abstract】 In order to minimize the correlation between adjacent bits, the interleaver is introduced in the process of the Turbo encoding. As a competition-free interleaver, the QPP interleaver has flexible structure and excellent performance. However, the calculation process is relatively complicated, which increases the design difficulty of the hardware circuit. To address this issue, this paper proposes a simplified four-way parallel radix-4 QPP interleaver. The calculation of the interleaving address is simplified to recursive calculation, and the calculation method of the interleaving address under the four-way parallel radix-4 condition is deduced, to design a single input multiple output (SIMO) interleaver. Finally, the proposed interleaver is implemented on FPGA. The simulated results show the interleaver can output 8 correct interleaving addresses in parallel in one clock, which improves the parallel Turbo decoding performance.

【Key words】 interleaver; QPP; parallel; radix-4; SIMO

0 引 言

信道编码是提高信道可靠性的重要理论和方法，Turbo 编码就是其中之一。Turbo 编码应用了随机编译码条件，将卷积编码与随机交织器结合在一起，采用软输出迭代译码来逼近最大似然译码，从而获得了接近 Shannon 理论极限的译码性能[1]。

Turbo 由分量码和交织器级联而成，因此，分量码和交织器设计的优劣直接影响着 Turbo 码性能[2-5]。其中，交织器主要功能是减小相邻比特之间的相关性，针对其研究主要从交织算法以及交织结构两个角度进行。文献[6]提出一种利用内存映射矩阵进行地址交织的方法，将每一个译码器的输出 i 填充到标记为 $M(i)$ 的存储器中，而这其中的映射关系由映射矩阵 $\boldsymbol{M}$ 决定，但是矩阵 $\boldsymbol{M}$ 需要使用退火算法逐级填充，求解过程较为复杂；文献[7]基于通用处理器（GPP）架构的实时信号处理技术，利用单指令多数据（SIMD）技术一条指令可以处理多个数据的特点，从指令级进行优化，提出一种高度并行的交织器，为 DSP 信号处理提供了良好的借鉴。为了消除并行交织译码过程中可能带来的地址冲突，引入了二次置换多项式交织算法[8]（QPP），该交织器具有 2 个特点，一是从 N 个并行 SISO 计算出来的 N 个外信息在解交织后，总能够被存入到 N 个不同的存储器中；二是该交织器是 N 个并行 SISO 所需要访问的 N 个交织地址总是指向 N 个存储器的同一个位置。文献[9]提出一种基于置换模式的优化交织器方案，该方案利用一个统一的交织硬件电路来计算所有并行的交织地址。其交织电路的设计虽然较优，但由于需要保存不同配置下的初始交

基金项目：江苏省“333 工程”科研资助项目（BRA2019304）。

作者简介：史宜巧（1975-），女，硕士，副教授，主要研究方向：计算机控制、自动化技术；李 丛（1984-），男，硕士，副教授，主要研究方向：智能信息处理。

通讯作者：史宜巧 Email：syq@ jsei.edu.cn

收稿日期：2022-01-08

织模式,因而总体的硬件复杂度较高。文献[10]针对实际译码过程中,为了满足高阶蝶形运算需求,设计了一种基 4 蝶形交织器模型,通过奇偶地址分模块并行计算实现,然而该模型中相邻地址计算存在依赖,地址计算过程中递推关系复杂。

为了更好地发挥交织器在 Turbo 译码中的作用,简化硬件实现,本文提出一种基 4 并行 QPP 交织器算法。文章内容安排如下:首先介绍 QPP 交织器原理,并通过递推简化交织地址在并行计算情况下存在的求余等复杂运算;然后进一步推导公式解除相邻交织地址计算的依赖关系,从而提出本文的 QPP 基 4 交织器;最后对本文提出的算法使用 Verilog 语言设计实现,并借助 FPGA 仿真工具与 Matlab 仿真结果对比,结果表明本文算法用于 FPGA 实现交织输出结果完全正确,并且通过相同工艺下的与其他方案对比,显示本文设计综合面积减小 50%左右,证明硬件开销更小。

1 QPP 交织器原理

3GPP 在 LTE 标准中采用了二次置换多项式(QPP)交织器作为 Turbo 码的内交织器,通过二次多项式推导计算交织地址,最终转换为递推计算。

1.1 QPP 交织器递推运算

QPP 交织器中,输出的下标 i 和输入下标 $\Pi(i)$ 的关系满足如下二次方程式:

$$\Pi(i) = (f_1 \cdot i + f_2 \cdot i^2) \bmod K \tag{1}$$

其中,f_1 和 f_2 取决于块的大小 K,在文献[11]中,3GPP 一共规定了 188 种不同长度的 K。

文献[12]对 $\Pi(i)$ 求解做进一步推导,如下:

$$\Pi(i) = (\Pi(i-1) + g(i-1)) \bmod K \tag{2}$$

其中,$g(i-1) = (f_1 + f_2 + 2(i-1)f_2) \bmod K$,很容易得知 $\Pi(i)$ 可以根据 $\Pi(i-1)$ 和 $g(i-1)$ 递推获得。

1.2 并行 QPP 交织器设计

考虑到高速 Turbo 码并行译码设计的需要,交织器也需要并行设计,可以将 K 均分为 N 个子块,一般 $N = \{1,2,3,4\}$,以 $N = 4$ 为例,则每个子块长度 $W = 4$,分块后交织过程如图 1 所示。

由文献[13]可知,QPP 交织器是一个无竞争交织器,即:

$$\Pi(i+tW) \bmod W = (f_1(i+tW) + f_2(i+tW)^2) \bmod W = \Pi(i) \bmod W \tag{3}$$

其中,$t = \{0,1,2,3\}$,表示块编号。先假设 $t = 0$,由于交织分块进行,可以重新表示为:

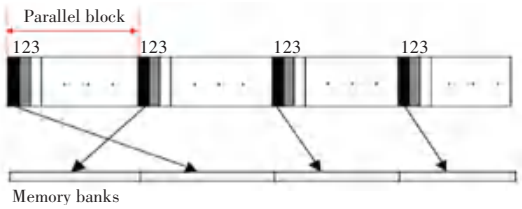

图 1 交织器并行计算示例

Fig. 1 Example of interleaver parallel computation

$$\Pi(i) = \Pi'(i) + q_{\Pi}(i)W \tag{4}$$

其中,$\Pi'(i) = \Pi(i) \bmod W$,表示块内偏移量;$q_{\Pi}(i) = \lfloor \frac{\Pi(i)}{W} \rfloor$ 表示块索引;$\lfloor\ \rfloor$ 表示向下取整。

由式(3)可知,第 i 时刻并行输出的 4 个交织地址块内偏移量 $\Pi'(i)$ 是一致的,因此每次并行计算出 4 个交织地址,实际上只需要计算出一个偏移量 $q_{\Pi}(i)$。

首先是 $\Pi'(i)$,由于 $K = 4 \cdot W$,那么可得 $(A \bmod K) \bmod W = A \bmod W$,已知求余运算有以下性质[13]:

$$(A+B) \bmod K = (A \bmod K + B \bmod K) \bmod K = \begin{cases} A \bmod K + B \bmod K & \text{if } A \bmod K + B \bmod K < K \\ A \bmod K + B \bmod K - K & \text{if } A \bmod K + B \bmod K \geqslant K \end{cases} \tag{5}$$

可推出 $\prod'(i)$ 的表达式:

$$\Pi'(i) = (\Pi(i-1) \bmod W + g(i-1) \bmod W) \bmod W \tag{6}$$

为简化运算,使硬件实现中不出现求余运算,令 $g'(i) = g(i) \bmod W$,代入式(6),再结合式(5)性质可得:

$$\Pi'(i) = \begin{cases} \Pi'(i-1) + g'(i-1) & \text{if } \Pi'(i-1) + g'(i-1) < W \\ \Pi'(i-1) + g'(i-1) - W & \text{if } \Pi'(i-1) + g'(i-1) \geqslant W \end{cases} \tag{7}$$

进一步简化表达式,引入 $Ind_{\Pi'(i-1)}$ 表示式(6)中判断的结果,即:

$$Ind_{\Pi'(i-1)} = \begin{cases} 0 & \text{if } \Pi'(i-1) + g'(i-1) < W \\ 1 & \text{if } \Pi'(i-1) + g'(i-1) \geqslant W \end{cases} \tag{8}$$

可以看到 $Ind_{\Pi'(i-1)}$ 是仅与下标 $i-1$ 时刻的值有关,同理,后续出现的 Ind 都照此规定。那么式(7)可以写成:

$$\Pi'(i) = g'(i-1) + \Pi'(i-1) - Ind_{\Pi'(i-1)} \times W \tag{9}$$

从式(9)可知,$\Pi'(i)$ 可以根据 $\Pi'(i-1)$ 和 $g'(i-1)$ 递推得到。将 $g'(i) = g(i) \bmod W$ 代入式

(9)可得：

$$g'(i)=(f_1+f_2+2if_2)\bmod W=r_{2f}+g'(i-1)-\underset{g'(i-1)}{Ind}\times W \tag{10}$$

其中，$r_{2f}=(2f_2)\bmod W$。由上式可知，$g'(i)$ 可以通过 $g'(i-1)$ 递推得到。参照 $\Pi'(i)$ 推导过程，同样有：

$$q_{\Pi}(i)=\lfloor\frac{\Pi(i)}{W}\rfloor=q_{\Pi}(i-1)+q_g(i-1)+\lfloor\frac{\Pi'(i-1)+g'(i-1)}{W}\rfloor-\underset{q_{\Pi}(i-1)}{Ind}\times 4 \tag{11}$$

$$q_g(i)=\lfloor\frac{g(i)}{W}\rfloor=q_g(i-1)+q_{2f}+\lfloor\frac{g'(i-1)+r_{2f}}{W}\rfloor-\underset{q_g(i-1)}{Ind}\times 4 \tag{12}$$

其中，$q_{2f}=\lfloor\frac{2f_2}{W}\rfloor$。由于 $\Pi'(i)$、$g'(i)$ 和 r_{2f} 均小于 W，因此，$\lfloor\frac{g'(i)+r_{2f}}{W}\rfloor$ 和 $\lfloor\frac{\Pi'(i)+g'(i)}{W}\rfloor$ 可以简化为：

$$\lfloor\frac{g'(i)+r_{2f}}{W}\rfloor=\begin{cases}1 & \text{if } g'(i)+r_{2f}\geqslant W\\ 0 & \text{if } g'(i)+r_{2f}<W\end{cases} \tag{13}$$

$$\lfloor\frac{\Pi'(i)+g'(i)}{W}\rfloor=\begin{cases}1 & \text{if } g'(i)+\Pi'(i)\geqslant W\\ 0 & \text{if } g'(i)+\Pi'(i)<W\end{cases} \tag{14}$$

上述求解仅仅是针对 $q_{\Pi}(i)$，而 $q_{\Pi}(i+tW)$ $(t=\{1,2,3\})$ 可以根据 $q_{\Pi}(i)$ 求解，推导公式如下：

$$q_{\Pi}(i+tW)=\lfloor\frac{\Pi(i+tW)\bmod K}{W}\rfloor=q_{\Pi}(i)+((f_1t+2f_2ti+t^2f_2W)\bmod 4)-Ind\times 4=(q_{\Pi}(i+tW)+f_1\bmod 4)\bmod 4 \tag{15}$$

2　基 4 并行 QPP 交织器算法设计

每时刻每个子块仅输出一个地址，因此递推关系也仅存在于相邻 2 个地址间，而实际的 Turbo 译码系统需要面临高阶蝶形运算需求，例如基 4 蝶形译码系统中需要交织器同时输出奇偶地址，如图 2 所示。因此本文考虑设计一种算法，通过初始地址一次性计算得到奇偶地址。

基 4 并行 QPP 交织器每个时刻需要并行输出 $\prod(2n+tW)$，$\prod(2n+1+tW)$ 共 8 个地址，其中 n 表示第 n 个时钟周期，有 $n=0,1,\cdots,(\frac{K}{8}-1)$，$t$ 为块索引，取值 $t\in\{0,1,2,3\}$。交织地址的内存仍然分为 4 块，每个子块每个时钟周期要同时输出奇偶两个地址，即该子块的第 $2n$ 个交织地址与第 $2n+1$ 个交织地址，因此本文设计如图 3 所示算法对奇偶地址同时计算。

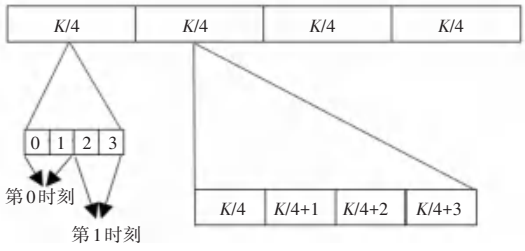

图 2　基 4 并行 QPP 交织器结构示意图

Fig. 2　Schematic diagram of radix-4 parallel QPP interleaver

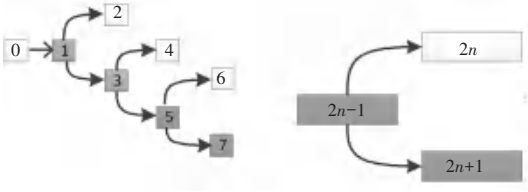

图 3　交织器算法原理图

Fig. 3　Schematic diagram of interleaver algorithm

由图 3 可知，要同时计算出 $2n$ 与 $2n+1$ 两处地址，需要推导两者与 $2n-1$ 处地址的关系。根据上述分析，计算交织地址采用递推的方式，$2n$ 与 $2n+1$ 处的地址实际上是相邻的，因此存在递推关系，以 $g'(i)$ 为例，根据式(10)有：

$$g'(2n)=r_{2f}+g'(2n-1)-\underset{g'(2n-1)}{Ind}\times W \tag{16}$$

$$g'(2n+1)=r_{2f}+g'(2n)-\underset{g'(2n)}{Ind}\times W \tag{17}$$

其中，$r_{2f}+g'(2n-1)\geqslant W$ 时，$\underset{g'(2n-1)}{Ind}=1$，否则 $\underset{g'(2n-1)}{Ind}=0$；$r_{2f}+g'(2n)-\underset{g'(2n)}{Ind}\times W\geqslant W$ 时，$\underset{g'(2n)}{Ind}=1$，否则 $\underset{g'(2n)}{Ind}=0$。

从式(16)和式(17)中可以看出，$g'(2n+1)$ 的计算依赖于 $g'(2n)$，这样奇偶地址无法同时计算。为消除两者之间的依赖关系，对式(15)作进一步递推，将式(16)代入式(17)可得：

$$g'(2n+1)=2r_{2f}+g'(2n-1)-(\underset{g'(2n-1)}{Ind}+\underset{g'(2n)}{Ind})\times W \tag{18}$$

同样，对 $\underset{g'(2n)}{Ind}$ 的判决条件，也作进一步递推，可以得到：

$$\underset{g'(2n)}{Ind}=\begin{cases}1 & 2r_{2f}+g'(2n-1)-\underset{g'(2n-1)}{Ind}\times W\geqslant W\\ 0 & 2r_{2f}+g'(2n-1)-\underset{g'(2n-1)}{Ind}\times W<W\end{cases} \tag{19}$$

通过上述推导，$g'(2n+1)$ 的计算不需要 $g'(2n)$，两者可以同时求解。与 $g'(2n)$ 和 $g'(2n+1)$ 的推导同理，$\prod'(2n)$ 和 $\prod'(2n+1)$ 的求解如下：

$$\Pi'(2n) = g'(2n-1) + \Pi'(2n-1) - Ind_{\Pi'(2n-1)} \times W \tag{20}$$

$$\Pi'(2n+1) = r_{2f} + 2g'(2n-1) + \Pi'(2n-1) - (Ind_{g'(2n-1)} + Ind_{\Pi'(2n-1)} + Ind_{\Pi'(2n)}) \times W \tag{21}$$

$\prod'(2n)$ 与 $\prod'(2n+1)$ 的计算过程可以用如下伪代码表示，算法的核心在于将每一步判断转化为 *Ind* 值的计算，从而将判断结果保留下来以供后续计算使用。

算法：计算 $\Pi'(2n)$ 与 $\Pi'(2n+1)$

for $n = 1:\frac{K}{8} - 1$

(1) if$(r_{2f} + g'(2n-1) < W)$ $Ind_{g'(2n-1)} = 0$

(2) else $Ind_{g'(2n-1)} = 1$ endif

(3) $g'(2n) = r_{2f} + g'(2n-1) - Ind_{g'(2n-1)} \times W$

(4) if$(2r_{2f} + g'(2n-1) - Ind_{g'(2n-1)} \times W < W)$ $Ind_{g'(2n)} = 0$

(5) else $Ind_{g'(2n)} = 1$ endif

(6) $g'(2n+1) = 2r_{2f} + g'(2n-1) - (Ind_{g'(2n)} + Ind_{g'(2n-1)}) \times W$

(7) if$(g'(2n-1) + \Pi'(2n-1) < W)$ $Ind_{\Pi'(2n-1)} = 0$

(8) else $Ind_{\Pi'(2n-1)} = 1$ endif

(9) if $(r_{2f} + 2g'(2n-1) + \Pi'(2n-1) - (Ind_{g'(2n-1)} + Ind_{\Pi'(2n-1)}) \times W < W)$ $Ind_{\Pi'(2n)} = 0$

(10) else $Ind_{\Pi'(2n)} = 1$ endif

(11) $\Pi'(2n) = g'(2n-1) + \Pi'(2n-1) - Ind_{\Pi'(2n-1)} \times W$

(12) $\Pi'(2n+1) = r_{2f} + 2g'(2n-1) + \Pi'(2n-1) - (Ind_{g'(2n-1)} + Ind_{\Pi'(2n-1)} + Ind_{\Pi'(2n)}) \times W$

endfor

程序代码中，(1)~(5)行执行计算 $Ind_{g'(2n-1)}$ 与 $Ind_{g'(2n)}$，可以看出两者可以同时进行。同理，(6)~(12)行是对 $Ind_{\Pi'(2n-1)}$ 与 $Ind_{\Pi'(2n)}$ 的计算。在完成对这些关键变量的计算之后，就可以根据(11)、(12)一次性计算出 $\prod'(2n)$ 与 $\prod'(2n+1)$。

$q_g(2n)$ 与 $q_g(2n+1)$ 以及 $q_\Pi(2n)$ 与 $q_\Pi(2n+1)$ 的计算类似，即通过往后递推一步，解除 n 时刻计算 $2n$ 与 $2n+1$ 两处地址所需变量之间的依赖关系，保证奇偶地址可以同时输出。

以上是第 0 子块的 $q_\Pi(2n)$ 和 $q_\Pi(2n+1)$ 表达式，剩下 $q_\Pi(2n+tW)$ 和 $q_\Pi(2n+1+tW)$（其中，$t \in \{1,2,3\}$），可以根据式(15)，由第 0 子块的 $q_\Pi(2n)$ 和 $q_\Pi(2n+1)$ 递推得到。最后，代入式(22)求得交织地址，即：

$$\Pi(i+tW) = \Pi'(i) + q_\Pi(i+tW) \times W \tag{22}$$

可以看出通过本文设计，能够实现利用单个输入计算输出多个地址，即单输入多输出(SIMO)。

3 FPGA 设计与仿真分析

为了证明本文设计可行性，对以上提出的算法使用 FPGA 实现，验证仿真结果，并与已有方案进行对比，实现语言采用 Verilog。

3.1 FPGA 设计与硬件复杂度

交织器的顶层电路如图 4 所示，主要包括查找表模块与交织模块两部分，其中后者主要为前者计算交织地址提供输入。这是因为交织地址的计算是递推过程，需要提供初始值与必要的参数[14-15]。

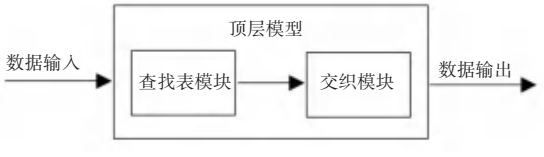

图 4 顶层模块图

Fig. 4 Top module diagram

其中，交织模块的内部实现如图 5 所示，可以看到整个模块都被简化成了判决单元与一些计算单元，而根据第 2 节代码可以知道这些计算单元只包括加减运算，因此综合出来的电路非常简单。另外可以看出上一轮计算得到的 $g'(i)$、$\Pi'(i)$ 以及 $q_g(i)$ 都要作为返回值参与下一轮计算。

将所设计的交织器的硬件复杂度与已有的技术方案进行对比，在 SMIC40 nm 工艺下对本文设计进行综合，并与文献[9]和文献[16]所提出的方案进行了对比，见表 1。表 1 中，文献[9]采用 radix-4，每个时钟通过 8 个 SISO 并行的方式输出 8 个地址。文献[16]采用 radix-2，最高 4 个 SISO 同时运行，也

就是每个时钟输出 4 个地址，并且硬件实现包含了除法器。而本文的设计通过 4 个 SIMO 并行运行每个时钟输出 8 个地址。本文设计的综合面积仅有 0.032 nm，通过归一化面积比可以看出，本文方案相对于另外 2 种方案硬件开销很小。

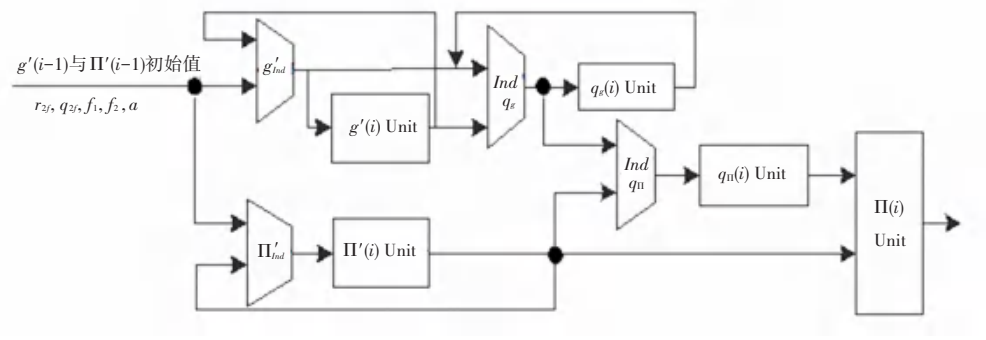

图 5　交织模块实现

Fig. 5　Interleaver module

表 1　硬件开销对比

Tab. 1　Hardware overhead comparison

方案来源	实现方式	并行单元数量	面积/nm	归一化面积比
文献[9]	radix−4	8	0.040 0	2.25
文献[16]	radix−2	4	0.065 2	2.04
本文	radix−4	4	0.032 0	1

3.2　仿真分析

图 6 为交织地址计算模块的输出结果。图 6 中，ttia 与 ttib 分别代表 $\prod'(2n)$ 与 $\prod'(2n+1)$，而 qttiab1 与 qttiab2 代表 $q_\Pi(2n+tW)$ 与 $q_\Pi(2n+1+tW)$，两者都是 8 bit 信号，每 2 bit 代表一个 $q_\Pi(i)$。

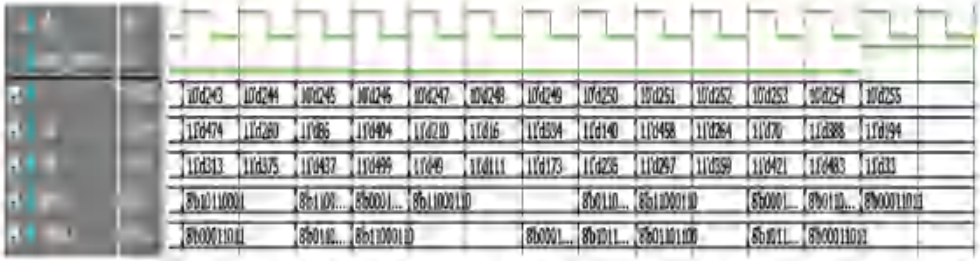

图 6　交织地址计算模块输出

Fig. 6　Interleaved address calculation module output

在交织模块的输出结果基础上，根据式(4)可以计算最终的交织地址，并且将包含正确交织地址 Matlab 仿真结果读入，仿真结果如图 7 所示。与 FPGA 仿真结果进行比对，如果有错误地址输出，则令计数加 1。从图 7 可以看出，每个时钟输出了 8 路地址，并且没有发生错误，说明本设计可以高效、正确地实现地址交织功能。

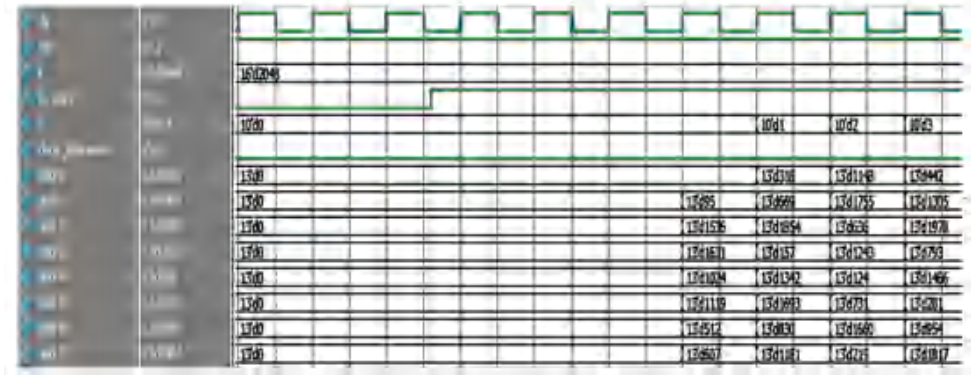

图 7　仿真结果

Fig. 7　The simulation results

4 结束语

本文提出了一种硬件优化的基 4 四路并行 QPP 交织器,针对并行计算场景,简化地址递推计算,并消除相邻地址计算的依赖关系,最终可以每个时钟并行输出 8 路奇偶地址,不仅降低了硬件实现复杂度,也大大提高了地址计算的效率。仿真结果表明本设计硬件开销较小,能够正确地输出交织地址,充分证明了本设计具有可行性。需要指明的是,为了方便 QPP 交织多项式的递推计算,本文在分块并行译码时,并行块数 M 必须满足 $M = 2^n$, 后续可以做进一步优化,将这种并行译码下的递推关系一般化,以方便该并行交织器更好地满足不同场景需求。

参考文献

[1] BERROU C, GLAVIEUX A, THITIMAJSHIMA P. Near Shannon limit error-correcting coding and decoding: Turbo-codes [C]// Proceedings of ICC '93 - IEEE International Conference on Communications. Geneva, Swizerland: IEEE, 1993: 1064-1070.

[2] NIMBALKER A, BLANKENSHIP T K, CLASSON B, et a1. Contention-free interleavers for high-throughput Turbo decoding [J]. IEEE Transactions on Communications, 2008, 56(8): 1258-1267.

[3] SUN Yang, CAVALLARO J R. Efficient hardware implementation of a highly-parallel 3GPP-LTE/LTE-advance turbo decoder [J]. Integration the VLSI Journal, 2011, 44 (4): 305-315.

[4] SHRESTHA R, PAILY R P. High-throughput turbo decoder with parallel architecture for LTE wireless communication standards[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2014, 61(9): 2699-2710.

[5] YOO I, KIM B, PARK I C. Tail-overlapped SISO decoding for high-througput LTE-advanced turbo decoders [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2014, 61 (9): 2711-2720.

[6] 黄跃斌,陈 赟,曾晓洋. LTE 中灵活并行无冲突 Turbo 码交织器的实现[J]. 复旦学报(自然科学版),2013,52(03):334-338.

[7] WEI Hu, LIN Tao, LIN Zhenghui. Parallel processing architecture of H. 264 adaptive deblocking filters [J]. Journal of Zhejiang University (Science A: An International Applied Physics & Engineering Journal), 2009, 10(8): 1160-1168.

[8] 丘选锋,赵宏宇. Turbo 码并行无冲突交织器设计[J]. 通信技术,2013,46(08):5-7.

[9] WANG Jian, ZHANG Kangli, KRöLL H, et al. Design of QPP interleavers for the parallel turbo decoding architecture[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2016, 63 (2): 288-299.

[10] 姚彦斌,周一青,林江南,等. 硬件友好的 3GPP-LTE Turbo 交织器设计[J]. 高技术通讯,2017,27(01):20-26.

[11] DOBIN R, PELEG M, GINOSAR R. Parallel VLSI architecture for MAP turbo decoder [C]//IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Lisbon, Portugal: IEEE, 2002: 384-388.

[12] Valbonne. Multiplexing and channel coding [S]. Valbonne, France: 3GPP Organization, 2009.

[13] DENG Jingli, PENG Yuexing, ZHAO Hui. Distance spectrum calculation method for double binary turbo codes [C]//2017 International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom). Vietnam: IEEE, 2017: 98-102.

[14] 王视环. LTE 中 Turbo 码内部交织器的研究[J]. 南京邮电大学学报(自然科学版),2010,30(04):90-94.

[15] 时述有,吉彦军. 基于 FPGA 的高速 TURBO 码编译码器硬件实现方法[J]. 辽东学院学报(自然科学版),2020,27(02):88-94.

[16] KIM B, YOO I, PARK I C. Low-complexity parallel QPP interleaver based on permutation patterns [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2013, 60(3): 162-166.

(上接第 34 页)

空域信息,本文提出了一种芯片外观缺陷特征增强的方法。即在光度立体技术获得的反照图的基础上,进行 Gabor 变换。实验表明,本方法能有效提升芯片外观局部微小缺陷特征清晰度。通过本方法对芯片外观缺陷的预处理,结合三维重建以及深度学习技术,预期能设计出一种比现有基于二维图像处理技术检测精度更高的芯片外观缺陷检测方案。

参考文献

[1] 巢渊.基于机器视觉的半导体芯片表面缺陷在线检测关键技术研究[D]. 南京:东南大学,2017.

[2] 陈恺. 集成电路芯片表面缺陷视觉检测关键技术研究[D]. 南京:东南大学,2016.

[3] 梁丽秀,裴玖玲,孙少杰,等. 自然光成像条件下植物根系图像的增强方法研究[J]. 科技创新与应用,2021,11(21): 83-86, 89.

[4] 梁淑芬,陈琛,冯跃,等. 基于一种局部图像增强和改进分水岭的舌体分割算法[J]. 现代电子技术,2021,44(16): 138-144.

[5] WOODHAM R J. Photometric method for determining surface orientation from multiple images[J]. Optical Engineering, 1980, 19(1): 139-144.

[6] DAUGMAN J G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical fifilters [J]. Journal of the Optical Society of America, 1985, 2(7): 1160-1169.

[7] 温悦欣. 基于机器视觉的 PCB 表面缺陷检测方法研究与系统实现 [D]. 成都:电子科技大学,2020.

[8] WANG Yu, WANG Mingquan, ZHANG Zhijie. Microfocus X-ray printed circuit board inspection system[J]. Optik-International Journal for Light and Electron Optics, 2014, 125(17): 4929-4931.

[9] 夏成蹊. 基于图像处理的集成电路芯片表面缺陷检测算法研究 [D]. 贵州:贵州大学,2019.