

# 基于 FPGA 的低资源极化码 SC 译码架构研究与实现 \*

曹 蓉<sup>1,2</sup>, 赵德政<sup>2</sup>, 郭 佳<sup>2</sup>, 李家鑫<sup>1</sup>

(1. 华北计算机系统工程研究所, 北京 100083; 2. 中电智能科技有限公司, 北京 100083)

**摘要:** 针对无线传感器网络中对资源消耗及成本敏感的应用场景, 研究并提出了一种基于 FPGA 的低资源极化码连续删除(Successive Cancellation, SC)译码架构。该译码架构采用同级计算单元串行运算, 不同级计算单元并行运算, 不同组译码数据并行处理的方式, 通过减少计算单元(Processing Element, PE)个数、复用寄存器存储资源提升硬件资源利用率, 复用译码延迟提升吞吐率。通过 Xilinx xc7vx330t 综合结果分析, 该译码架构在码长为  $N=128$  时译码最高时钟频率为 220.444 MHz, 吞吐率为 89.86 Mb/s, 与树型 SC 译码架构相比, 计算单元利用率提升了 14.67 倍, 在主要硬件资源指标查找表(Look-Up-Table, LUT)和触发器(Flip-Flop, FF)上分别节省了 74.22% 和 62.1%。

**关键词:** FPGA; 极化码; 低资源; 计算单元; SC 译码架构

中图分类号: TN911

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.200203

中文引用格式: 曹蓉, 赵德政, 郭佳, 等. 基于 FPGA 的低资源极化码 SC 译码架构研究与实现[J]. 电子技术应用, 2020, 46(9): 74-78, 84.

英文引用格式: Cao Rong, Zhao Dezheng, Guo Jia, et al. Research and implementation of low resource polar code SC decoding architecture based on FPGA[J]. Application of Electronic Technique, 2020, 46(9): 74-78, 84.

## Research and implementation of low resource polar code SC decoding architecture based on FPGA

Cao Rong<sup>1,2</sup>, Zhao Dezheng<sup>2</sup>, Guo Jia<sup>2</sup>, Li Jiaxin<sup>1</sup>

(1. National Computer System Engineering Research Institute of China, Beijing 100083, China;

2. Intelligence Technology of CEC Co., Ltd., Beijing 100083, China)

**Abstract:** In view of the application scenarios are sensitive to resource consumption and cost in wireless sensor networks, this paper studies and proposes a low resource polar code successive cancellation(SC) decoding architecture based on FPGA. The decoding architecture adopts the serial operation of the same level computing units, the parallel operation of different levels of computing units, and the parallel processing of different groups of decoding data. By reducing the number of processing elements(PE), reusing register storage resources, the architecture improves the utilization of hardware resources, and reusing decoding delay improves the throughput. According to the analysis of the comprehensive results of Xilinx xc7vx330t, when the code length is  $N=128$ , the architecture's highest decoding clock frequency is 220.444 MHz, and the throughput is 89.86 Mb/s. Compared with the tree SC decoding architecture, its utilization ratio of computing unit is increased by 14.67 times. It saves 74.22% and 62.1% on Look-Up-Table(LUT) and Flip-Flop(FF) of main hardware resource indexes respectively.

**Key words:** FPGA; polar code; low resource; processing elements; SC decoding algorithm

## 0 引言

无线传感器网络是将自动控制技术、传感器技术、无线传输技术、数字信号分析处理技术融合于一体的先进系统, 由大量带有传感器并能够进行无线通信的节点通过自组织的方式构成网络, 各节点之间相互协作对周围环境进行物理感知, 并将处理后的信息返回终端设备<sup>[1]</sup>。无线传感器网络的数据传输差错控制一般采用循环冗余校验(Cyclic Redundancy Check, CRC), 而循环冗余校验

只能校验出某段信息中出现差错, 然后通知发送端重传信息, 当传感器网络传输大量数据, 并且传输距离较远时, 信息差错率明显提升, 通信系统的可靠性大大降低。从网络能耗方面分析, 无线通信模块发送、接收信息的能耗是无线传感器网络中最大的, 重传率的升高必然导致能量消耗加快。文献[2]中提出的利用低密度奇偶校验(Low-Density Parity-Check, LDPC)码的信道编码技术可以大幅度提升数据的可靠性, 从而降低无线传感器的能量消耗, 但是 LDPC 码并没有达到香农极限, 因此无线传感器传输的可靠性还有进一步提升的可能性。

\* 基金项目: 国防基础科研计划资助项目(JCKY2018211C001)

极化码是目前唯一被理论证实能够达到香农极限<sup>[3]</sup>的一种基于信道极化现象提出的信道编码方式,具有较低的编译码复杂度和高可靠传输性能等优势,被确立为5G eMBB(Enhanced Mobile Broadband)场景下控制信道的编码方式<sup>[4]</sup>。5G的应用场景要求低延时高吞吐率,因此在极化码SC译码器研究中大部分研究者主要在吞吐率上的优化和提升,如文献[5]提出的FFT型SC译码器架构,需要 $N \log_2 N$ ( $N$ 为SC码长)个节点计算单元,每一级需要 $N$ 个寄存器,该译码结构采用自右向左的译码顺序,译码 $N$ 比特信息需要 $2N-2$ 个时钟周期(Clock Cycle, CC),也就是每 $2N-2$ 个时钟周期所有计算节点都只使用了一次,可见在FFT型SC架构译码过程中大部分时间计算节点都是处于空闲状态,造成了大量硬件资源的浪费。树型SC架构译码过程中,当任意级 $k$ 开始计算似然比信息(Likelihood Ratio, LR)时,最多只有 $2^k$ 个节点参与计算,树型SC架构包括 $N-1$ 个PE和 $N-1$ 个寄存器,PE是可以选择执行 $f$ 函数或 $g$ 函数的可配置单元。和FFT架构相比,尽管吞吐率和计算量没有提升,但是树型架构减少了PE和寄存器的数量。线型SC架构寄存器依旧保持树型架构,寄存器和PE仿照树型架构通过资源多路复用的方式相连,尽管线型结构加入了二输入多路复用器,但是PE数量的减少使它显著降低了硬件复杂度,又同时达到了和树型架构一样的吞吐率。以上SC译码器架构虽然能达到较高的吞吐率,但是在码长较长(如 $N=128$ )时,计算单元PE使用的数据较大,消耗硬件资源极大。但是对于无线传感器网络,1 Mb/s的吞吐率就能满足需求,同时无线传感器网络节点多采用电池进行供电,能量有限<sup>[2,6]</sup>,因此在牺牲部分吞吐率的前提下,大幅度降低极化码SC译码器的硬件资源消耗对于极化码在无线传感器网络的应用具有重大意义。

基于以上问题,本文提出了一种基于FPGA的低资源极化码SC译码架构,采用同级计算单元串行运算,不同级计算单元并行运算,不同组译码数据并行处理的硬件架构,通过减少计算单元的个数、复用译码延时和寄存器存储资源,提升硬件利用率,降低硬件资源消耗和硬件成本。通过Xilinx xc7vx330t综合结果分析,该译码架构在码长为128时吞吐率为89.86 Mb/s,主要硬件资源指标LUT和FF分别是690和274。本文研究的低资源的极化码SC译码架构的计算单元PE利用率相比树型SC架构提升了14.67倍;相比文献[7]提出的SC译码架构,在码长 $N=128$ 时,主要硬件资源指标LUT和FF上分别节省了74.22%和62.1%。

## 1 低资源的极化码SC译码器

### 1.1 极化码的传统SC译码架构

SC译码架构的核心思想是通过判断每个传输码字的LR来进行译码,对于信道索引 $i\in$

$\{1, 2, \dots, N\}$ ,计算信息比特 $u_i$ 的估计值 $\hat{u}_i=0$ 或 $\hat{u}_i=1$ 时的 $W_N^{(i)}$ 转移概率大小进行判断,译码按照 $i$ 从 $1\sim N$ 的顺序依次进行。对于任意参数 $(N, K, A, u_{Ac})$ 的极化码,其中参数 $N$ 表示编码后的码字长度, $K$ 表示原始信息比特长度, $A$ 表示传输信息比特的子信道的集合, $A_c$ 表示的是传输冻结比特的子信道的集合, $u_{Ac}$ 用来表示固定比特取值。每个译码比特 $u_i$ 的LR定义如式(1),其中 $y_1^N$ 为接收信号, $\hat{u}_1^{i-1}$ 为输入比特的估计值。

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) = \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | 1)} \quad (1)$$

判决单元按照接收顺序依次进行判决,当 $i \in A_c$ 时,表明该比特是冻结比特,即收发端事先约定的比特,直接判决为 $\hat{u}_i=u_i$ ;当 $i \in A$ 时,表明该比特是承载信息的信息比特,要通过判决函数进行判决,各比特的判决规则如式(2)所示:

$$\hat{u}_i = \begin{cases} u_{Ac}, & i \in A_c \\ 0, & i \in A \text{ and } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1, & i \in A \text{ and } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) < 1 \end{cases} \quad (2)$$

SC译码架构的复杂度主要集中在似然比信息LR的计算上,LR计算按长度 $\{N, N/2, N/4, \dots, 1\}$ 逐级进行,直到长度为1停止计算。由图1可以看到, $N=8$ 时,第二级计算 $N/2=4$ 个LR,第一级计算 $N/4=2$ 个LR,第0级计算 $N/8=1$ 个LR,要得到所有的LR值总共需要计算 $N(1+\log N)$ 次,所以SC译码架构的复杂度为 $O(N \log N)$ 。

### 1.2 低资源极化码SC译码架构工作原理

引言中介绍的3种译码架构有着共同点,即每一级有多个PE并行工作,虽然此种方式吞吐率高,但是FPGA资源消耗大,以传统SC译码的蝶形架构为基础,本文提出的低资源的SC译码架构通过大幅减少PE的数量来提升PE利用率,节省FPGA资源。低资源的SC译码架构只需 $\log_2 N$ 个PE,每一级复用同一个PE串行工作,同一时钟周期不同级的PE并行工作,上一组译码还

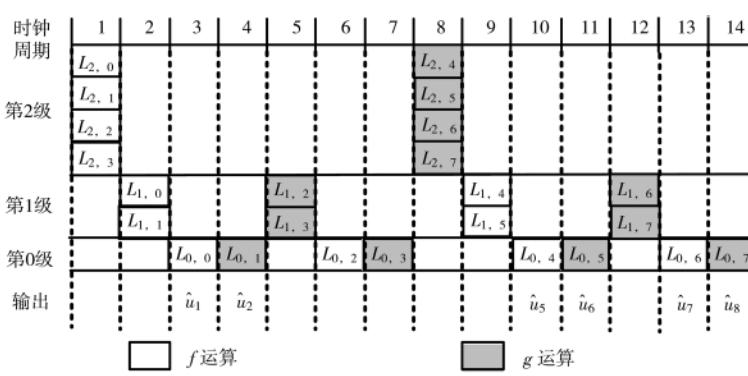


图1  $N=8$ 时传统SC译码架构LR数据流图

未完成就可以开启下一组译码,两组译码流程并行执行,此种方式可以牺牲部分吞吐率节省大量的PE资源。

### 1.2.1 不同级的并行运算

从图2可以看出,码长为8时低资源的SC译码架构共分为3级,按顺序每一级的计算单元分别是PE2、PE1、PE0,第二级的PE2在CC{1,4}串行运算4次,在CC#4(第4个时钟周期)第一级的PE1也开始运算,这就是处于不同级的计算单元可以在同一时钟周期并行运算,最大程度减少PE空闲时间,提高吞吐率。

### 1.2.2 不同组的并行运算

低资源的SC架构连续译出两组数据共需25个时钟周期,在第一组译码数据进行到CC#11的时候启动第二组译码数据运算,不同级的PE2和PE1在同一个时钟周期并行处理两组不同的译码数据,且同级别的寄存器组可以在不同时钟周期给多组译码数据进行复用,可以节省寄存器占用的硬件资源,计算产生的数据需要寄存器组regs存储相对应的值,一个寄存器组拆分为三个不同的寄存器,分别存储f运算结果(f\_reg)、g运算加法结果(gadd\_reg)和g运算减法结果(gsub\_reg),共需要消耗 $(N-1) \times 3$ 个寄存器。

### 1.3 译码周期和译码延迟

低资源的SC架构每一级只有一个PE,同一级PE串行运算,同一时刻的不同级PE并行运算,低资源的SC译码器译码周期 $t_{cc}$ 计算公式如下:

$$t_{cc} = \frac{N}{4} \log_2 N + \frac{5}{4} N - 1 \quad (3)$$

低资源的SC译码器在不同级PE是并行运算,因此更适用于连续译码,从图2可以看出在第一组译码结束时(第15时刻),第二组的PE2和PE1也执行完毕,在第16时刻执行PE0,同时译出第二组的第一个结果,因此第二组不需要额外的译码延时,译码延迟 $t_{late}$ 的计算公式如下:

$$t_{late} = \frac{N}{2} + \log_2 N - 2 \quad (4)$$

由式(3)和式(4)可得,低资源SC架构译码过程实际消耗的时钟周期 $t_{real}$ 的计算公式如下:

$$t_{real} = t_{cc} - t_{late} = \frac{3}{4} N + \log_2 N \left( \frac{N}{4} - 1 \right) + 1 \quad (5)$$

图2中 $L_{i,n}$ 表示经过计算单元PE运算后得到的似然信息,其中n表示译码过程中的级索引,i表示在第n级的译码比特索引,起始位置从0开始,例如 $L_{2,0}$ 代表的就是第二级计算单元PE2运算得到的第一个似然信息,以此类推;浅灰色的 $L_{i,n}$ 表示第1组译码的过程数据,深灰色的 $L_{i,n}$ 表示第2组译码的过程数据。

## 2 低资源极化码SC译码硬件架构

低资源的极化码SC译码硬件架构如图3所示,主要由计算单元PE、寄存器组regs、冻结比特ROM、部分和反馈、控制逻辑5个功能模块组成,接下来分别对这5个模块进行详细介绍,其中PE2~PE0表示第2级计算

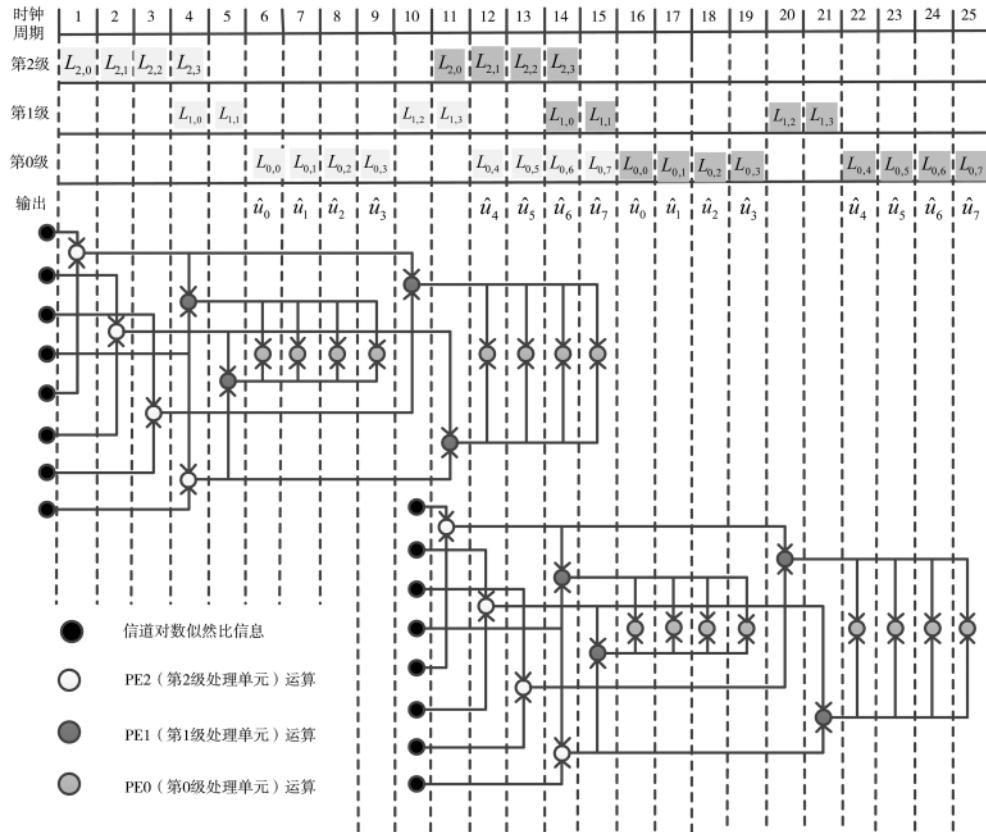


图2  $N=8$  时低资源的极化码SC译码架构连续两组译码运行时刻表

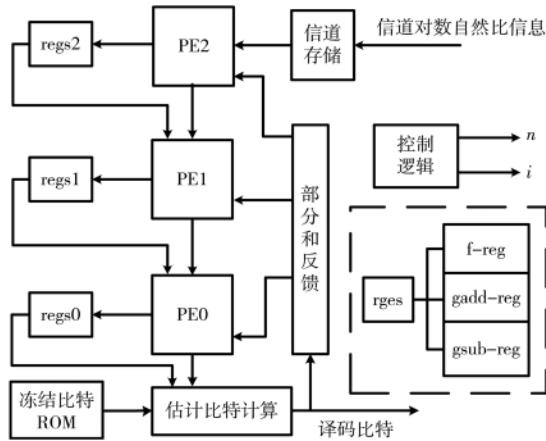


图 3 低资源的极化码 SC 译码硬件架构

单元至第 0 级计算单元,  $regs2 \sim regs0$  表示与相应级的计算单元相对应的存储单元, 控制逻辑中的  $n$  表示译码过程中的级索引,  $i$  表示在第  $n$  级的译码比特索引, 起始位置从 0 开始。

### 2.1 计算单元 PE

$f$  运算和  $g$  运算具有相同的输入, 运算规则如式(6)、式(7)所示, 其中  $a$  和  $b$  是输入的信道似然比信息;  $sign(a)$  和  $sign(b)$  是符号函数, 分别输出  $a$  和  $b$  的符号;  $min$  函数为输出二者之中的最小值。

$$f(a, b) \approx sign(a)sign(b)min(|a|, |b|) \quad (6)$$

$$g_{\hat{u}_s}(a, b) = a(-1)^{\hat{u}_s} + b \quad (7)$$

$g$  运算中的  $\hat{u}_s$  为式(2)的判决结果, 若  $\hat{u}_s=0$ , 则:

$$g=g\_add=b+a \quad (8)$$

若  $\hat{u}_s=1$ , 则:

$$g=g\_add=b-a \quad (9)$$

在硬件电路中求解绝对值较为复杂, 可以将  $f$  运算中的绝对值进行简化或者等价替换,  $g$  运算每次都需要计算出  $b+a$  和  $b-a$  的结果, 可以直接取二者运算结果对应的符号进行“异或”操作, 得到  $f$  运算中取绝对值  $|a|$  和  $|b|$  最小值的输出符号,  $f$  节点运算中两个输入数据选择器的输出结果设为  $C_{tmp}$ , 其公式如下:

$$C_{tmp} = \begin{cases} b, & |b| < |a| \\ a, & |b| \geq |a| \end{cases} \quad (10)$$

$f$  节点运算的输出值可以由  $sign(a)$ 、 $sign(b)$  和  $sign(C_{tmp})$  共同决定, 优化后的  $f(a, b)$  运算公式如下所示:

$$f(a, b) = [sign(a)sign(b)sign(C_{tmp})]C_{tmp} \quad (11)$$

PE 计算单元由 3 个异或门、2 个选择器组成,  $a$  和  $b$  为输入, 输出分别有  $f$  运算结果、 $g\_add$  (表示  $g$  运算加法结果)、 $g\_sub$  (表示  $g$  运算减法结果)。PE 计算单元的结构图如图 4 所示。

### 2.2 寄存器组 $regs$

每个寄存器组分为 3 个类型:  $f$  寄存器组 ( $f\_reg$ ) 存储  $f$  运算结果,  $g$  加法寄存器组 ( $gadd\_reg$ ) 存储  $g$  运算加法

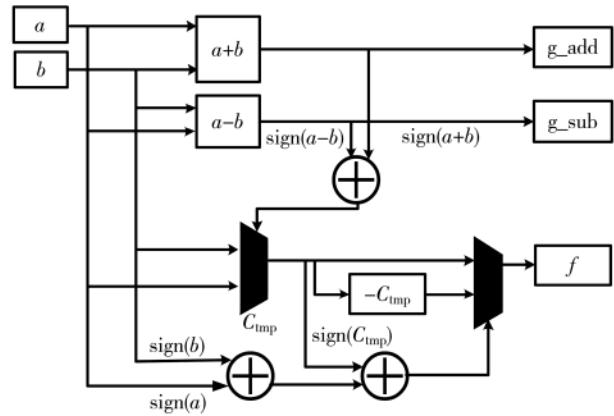


图 4 PE 结构图

结果,  $g$  减法寄存器组 ( $gsub\_reg$ ) 存储  $g$  运算减法结果。以码长  $N=8$  为例,  $regs2$  寄存器组存放的是经过 PE2 计算的 4 组  $f$  运算结果和  $g$  运算加减法结果,  $regs1$  寄存器组存放的是经过 PE1 计算的 2 组  $f$  运算结果和  $g$  运算加减法结果,  $regs0$  寄存器组存放的是经过 PE0 计算的 1 组  $f$  运算结果和  $g$  运算加减法结果。

### 2.3 冻结比特 ROM

极化码是基于信道极化的编码方式, 当码长足够大时, 极化子信道的信道容量会出现极化现象, 一部分信道容量趋于 0, 另一部分信道容量趋于 1, 随着码长增加, 极化子信道的对称容量越来越集中于 0 和 1, 也就是极化现象越明显。可以推测当码长无限长, 信道完全极化时, 信息传输速率就能达到香农极限。极化码的思想就是要在对称容量趋于 1 的  $K$  个信道传输信息比特, 信息位在此信道可以实现无差错传输; 在对称容量趋于 0 的  $(N-K)$  个信道传输冻结比特, 这部分信道可靠性低, 传输信息出错概率大。接收端事先约定的规则将冻结比特均取“0”, 低资源的 SC 架构将冻结比特按照译码顺序存储在一个大小为 1 KB 的 ROM 中, 译码时将译码计数器作为 ROM 的读地址读取 ROM。当读取结果为“0”时, 将译码结果置“0”; 当读取结果为“1”时, 再根据 PE0 得到译码结果。

### 2.4 部分和反馈

在整个译码过程中, 特定的部分和反馈信息是 PE 运算的一部分。部分和反馈由  $\hat{S}_{n,i}$  表示, 其中  $n$  代表译码过程中的级索引,  $i$  表示在第  $n$  级的译码比特索引。由图 3 可知, 不同级的 PE 需要不同的部分和  $\hat{S}_{n,i}$ , 根据图 2 的译码数据流图, 可以得到  $n \in \{0, 2\}$  每一级的部分和反馈, 其中  $\hat{u}_0 \sim \hat{u}_7$  为信息比特  $u_0 \sim u_7$  的估计值。

当  $n=2$  时, 部分和反馈如下:

$$\hat{S}_{2,0} = \hat{u}_0 \oplus \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3$$

$$\hat{S}_{2,1} = \hat{u}_1 \oplus \hat{u}_3$$

$$\hat{S}_{2,2} = \hat{u}_2 \oplus \hat{u}_3$$

$$\hat{S}_{2,3} = \hat{u}_3$$

当  $n=1$  时, 部分和反馈如下:

$$\hat{S}_{1,0} = \hat{u}_0 \oplus \hat{u}_1$$

$$\hat{S}_{1,1} = \hat{u}_1$$

$$\hat{S}_{1,4} = \hat{u}_4 \oplus \hat{u}_5$$

$$\hat{S}_{1,5} = \hat{u}_5$$

当  $n=0$  时, 部分和反馈如下:

$$\hat{S}_{0,0} = \hat{u}_0$$

$$\hat{S}_{0,2} = \hat{u}_2$$

$$\hat{S}_{0,4} = \hat{u}_4$$

$$\hat{S}_{0,6} = \hat{u}_6$$

$N=8$  的改进 SC 架构在执行译码过程中, 每一级需要 4 个部分和反馈给相应级的 PE, 根据不同的部分和反馈选择 PE 是进行  $g$  运算的加法输出还是减法输出。

## 2.5 控制逻辑

控制单元负责每个 CC 开始的译码操作, 以码长  $N=8$  为例, 在 CC#1(第一个时钟周期)开始计算位于 stage2 级 LLR0 和 LLR4 的  $f$  运算和  $g$  运算, 在 CC#2 开始计算位于 stage2 级 LLR1 和 LLR5 的  $f$  运算和  $g$  运算, 在 CC#3 开始计算位于 stage2 级 LLR2 和 LLR6 的  $f$  运算和  $g$  运算, 在 CC#4 开始计算位于 stage2 级 LLR3 和 LLR7 的  $f$  运算和  $g$  运算。由于 stage1 级计算单元 PE1 需要的输入在 CC#1 和 CC#3 已经计算完成, 因此在 CC#4 可以同步计算出 stage1 级的  $f$  运算和  $g$  运算, 而不需要等到 stage2 级全部计算完成。CC#5 开始计算位于 stage1 级 LLR1 和 LLR3 的  $f$  运算和  $g$  运算, CC#6 开始计算位于 stage0 级 LLR0 和 LLR1 的  $f$  运算和  $g$  运算, 同时译出  $\hat{u}_0$ , CC#7 根据  $\hat{u}_0$  的结果选择输出  $g$  运算加法结果还是减法结果, 从而判决出  $\hat{u}_1$ 。依此类推, 计算控制单元调度每一个时钟周期开始的译码操作, 使译码过程有条不紊地进行下去。

## 2.6 不同 SC 译码架构对比

本文分析了 FFT 型、树型、线型的 SC 架构, 以蝶形译码架构为基础, 改进得到的低资源极化码 SC 译码架构只需牺牲少量的吞吐量, 就可显著降低硬件资源消耗, 提升计算单元的利用率。此译码结构适用于多种码长, 译码结构的通用性更高。

SC 译码结构中研究较多的是树型架构, 由表 1 可知, 在  $N=128$  时, 树型 SC 架构需要  $N-1=127$  个 PE, 两次译码输入间隔为  $2N-2=254$ , 若只使用一个 PE, 则两次译码输入之间的间隔为  $N\log_2 N=896$ , 则 PE 的利用率计算如下:

$$\frac{1 \times 896}{12 \times 254} = 0.0278 \quad (12)$$

由上式可知, 树型 SC 架构译码延时短, 但是计算单

表 1 不同 SC 架构对比

	FFT 型	树型	线型	低资源型
PE/个	$N\log_2 N$	$N-1$	$\frac{N}{2}$	$\log_2 N$
regs/个	$N\log_2 N$	$N-1$	$N-1$	$(N-1) \times 3$
CC/个	$2N-2$	$2N-2$	$2N-2$	$\frac{3}{4}N + \log_2 N \left(\frac{N}{4}-1\right) + 1$

元利用率不足 2.8%。

在  $N=128$  时, 低资源的 SC 架构需要  $\log_2 N=7$  个 PE, 两次译码输入间隔为  $t_{\text{real}}=314$ , 则 PE 的利用率计算如下:

$$\frac{1 \times 896}{7 \times 314} = 0.408 \quad (13)$$

与树型架构的 PE 利用率相比, 低资源的 SC 架构的 PE 利用率提升了 14.67 倍, 且只增加了少量的译码延时。

## 3 译码仿真结果与分析

为了验证低资源的 SC 译码器功能的正确性, 本文使用 Xilinx 公司的 xc7vx330t 主芯片, 在 ISE 开发环境中进行编译, 并使用 ModelSim 仿真。仿真测试数据配置如下:

SC 码长  $N=8$ , 信息比特数  $K=4$ , 码率  $R=0.5$ , 信息比特集合  $=\{4, 6, 7, 8\}$ , 冻结比特全取 0, 4 个信息比特为 (1111), 则  $u_1^8=(00010111)$ 。经过信道编码和加噪后, 得到接收对数似然比为  $L_1^8=(-2, -2, -4, 1, -6, 6, 16, 3)$ 。

从图 5 译码仿真结果可以看到,  $u_{\text{out}}$  信号在  $u_{\text{valid}}$  信号为高电平时输出有效, 输出结果与  $u_1^8=(00010111)$  一致, 验证译码无误。译码从 255 ns 开始, 405 ns 结束, 码长为 8 的改进 SC 译码架构总耗时 150 ns。由图 3 可以看出, 低资源的 SC 架构适用于连续译码, 在第一组译码还没有完成时就可以开启第二组译码, 多组译码数据还能复用译码延迟, 提升吞吐率。吞吐率  $Th$  的计算公式如下:

$$Th = \frac{N \times f_{\text{max}}}{t_{\text{real}}} \quad (14)$$

式中,  $N$  为编码长度,  $f_{\text{max}}$  是最大时钟频率,  $t_{\text{real}}$  为实际译码消耗的时钟周期数。

对比表 2 和表 3 可以看出, 在主要硬件资源指标 LUT 和 FF 上分别节省了 74.22% 和 62.1%, 低资源的 SC 译码架构具有明显优势。

## 4 结论

本文以工业特殊环境下的无线传感器网络的低成本、低资源为研究点, 提出了一种基于 FPGA 的低资源的极化码 SC 译码器架构, 通过减少计算单元的个数, 同级计算单元串行运算, 不同级计算单元并行运算, 不同组译码数据并行处理的硬件架构, 极大地减少了硬件资源消耗, 降低了硬件成本。通过 Xilinx xc7vx330t 综合结果分析, 该译码架构在码长为 128 时吞吐率为 89.86 Mb/s, 主要硬件资源指标 LUT 和 FF 分别是 690 个和 274 个。

(下转第 84 页)

设计与应用[J].工程建设与设,2017(15):222-224,227.

[17] 刘建平,刘凤英.数控切割管理系统的研发[J].电脑知识与技术,2010,6(19):5357-5358.

[18] FURNESS A. Traceability of GMOs the new EU regulations[Z]. 2004.

[19] 金凯,杨睿哲,杨兆鑫,等.区块链在供应链管理上的应用[J].情报工程,2018,4(3):29-38.

[20] 张延华,杨兆鑫,杨睿哲,等.基于区块链的农产品溯源系统[J].情报工程,2018,4(3):4-13.

(收稿日期:2019-12-05)

## 作者简介:

邱凯(1994-),男,硕士研究生,主要研究方向:区块链技术。

司鹏搏(1983-),男,博士,教授,主要研究方向:无线通信网络。

曾涛(1972-),通信作者,男,硕士,高级工程师,主要研究方向:工业化建造技术,E-mail:783772328@qq.com。

(上接第 78 页)

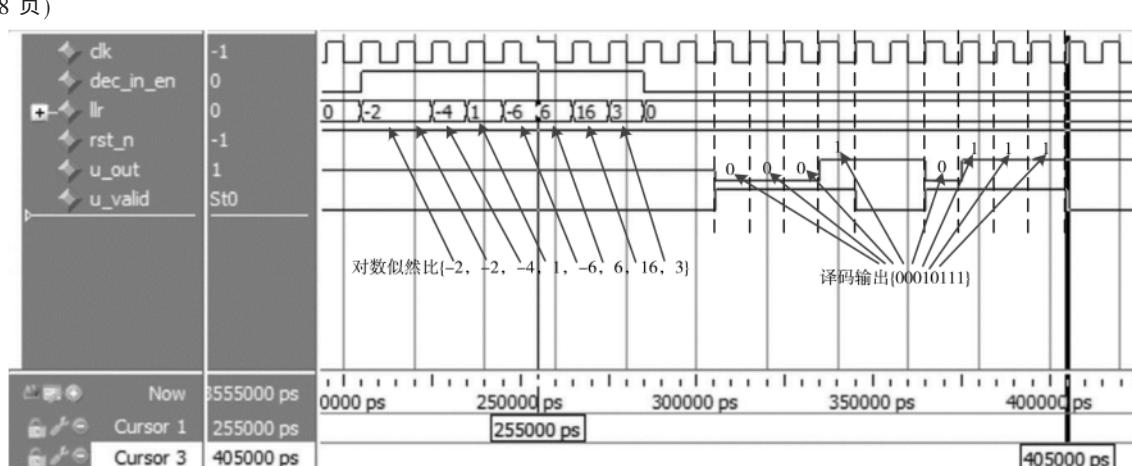


图 5 低资源的 SC 译码仿真图

表 2 低资源 SC 译码架构多种码长的硬件消耗

资源	$N=8$	$N=128$	$N=1\,024$
LUT/个	191	690	2 072
FF/个	95	274	640
IO/个	22	22	22
BUFG/个	1	1	1
$f/\text{MHz}$	259.804	220.444	144.47
吞吐率 $\text{Th}(\text{Mb/s})$	207.84	89.86	44.57

表 3  $N=128$  时传统 SC 译码器的硬件消耗<sup>[7]</sup>

资源	使用量
LUT/个	7 429
FF/个	2 102
IO/个	1 156
BUFG/个	1

本文研究的低资源的极化码 SC 译码架构的计算单元 PE 利用率相比树型 SC 架构提升了 14.67 倍;相比文献[7]提出的 SC 译码架构,在码长  $N=128$  时,在主要硬件资源指标 LUT 和 FF 上分别节省了 74.22% 和 62.1%。

## 参考文献

[1] 詹伟,李琳.无线传感器网络信道编码技术与研究[J].计算机与数字工程,2009(7):11-12.

[2] 谢东福,王琳,陈平平.低资源多边类型 LDPC 码译码器的 FPGA 实现[J].应用科学学报,2010,28(6):633-637.

[3] SHANNO C E. A mathematical theory of communication[J]. Bell Labs Technical Journal, 1984, 5(4):3-55.

[4] ARIKAN E. Channel polarization:a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels[J]. IEEE Transaction on Information Theory, 2009, 55(7):3051-3073.

[5] LEROUX C, TAL I, VARDY A, et al. Hardware architectures for successive cancellation decoding of polar codes[J]. 2011 IEEE Conference on Acoustics, Speech and Signal Processing, 2011.

[6] 许永富.应用于无线传感器网络的 LDPC 码研究与实现[D]. 哈尔滨:哈尔滨工业大学,2010.

[7] 夏阁淞,葛万成.极化码译码架构的改进与 FPGA 实现[J].通信技术,2019,52(11):2611-2616.

(收稿日期:2020-03-19)

## 作者简介:

曹蓉(1992-),女,硕士研究生,主要研究方向:嵌入式、信息与信号处理。

赵德政(1985-),男,博士,高级工程师,主要研究方向:工控安全、工业控制。

郭佳(1988-),男,硕士,助理工程师,主要研究方向:嵌入式、工控安全。

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所