一种余数系统基扩展算法及VLSI实现

马 上，汪陈浩，胡剑浩

(电子科技大学通信抗干扰技术国家级重点实验室 成都 611731)

【摘要】基扩展是余数系统(RNS)在数字信号处理(DSP)系统中应用的关键问题之一。该文提出了一种新型基扩展算法，实现基为 $\{2^n-1,2^n,2^n+1\}$ 的余数系统到基为 $\{2^{2n}-1,2^{2n},2^{2n}+1\}$ 的余数系统的动态范围扩展。给出其VLSI实现结构，并基于 $\{2^n-1,2^n,2^n+1\}$ 的特性对该结构进行了优化，使该实现结构仅由普通二进制加法器和模加法器构成。基于单位门模型和ASIC的性能对比分析结果表明，在实现相同动态范围扩展时，该算法具有良好的VLSI实现性能。

关 键 词 基扩展； 数字信号处理； 余数系统； 超大规模集成电路

中图分类号 TP33 **文献标志码** A doi:10.3969/j.issn.1001-0548.2015.02.008

A New Base Extension Algorithm and VLSI Implement for Residue Number System

MA Shang, WANG Chen-hao, and HU Jian-hao

(National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China Chengdu 611731)

Abstract The base extension operation for residue number systems (RNS) plays an important role in RNS-based digital signal processing (DSP) systems. In this paper, a new base extension algorithm is proposed which can extend the dynamic range from moduli set $\{2^n-1,2^n,2^n+1\}$ to moduli set $\{2^{2n}-1,2^{2n},2^{2n}+1\}$. In this paper, the very large scale integrated (VLSI) circuits implement of the proposed algorithm is also presented, with the properties of moduli set $\{2^n-1,2^n,2^n+1\}$, the implement is composed of binary adders and modular adders; The analysis result based on unit-gate model and ASIC (application specific integrated circuit) implementation shows that the VLSI implementation of the proposed base extension algorithm exhibits better performances for the same dynamic range extension.

Key words base extension; digital signal processing; residue number system; VLSI circuits

余数系统(residue number system，RNS)是一种非权重数值表征系统，它将传统的较大位宽的乘加运算分解为多个较小位宽的并行通道进行处理，从而降低了复杂度和计算的关键路径，可获得高速、低功耗的VLSI(very large scale integrated circuits)实现性能。因此，近年来余数系统在乘加密集型的数字信号处理(digital signal processing, DSP)系统中得到了广泛研究[1-3]。然而，在DSP系统的运算过程中，数值的动态范围会随着乘、加等基本运算而增加。这是RNS在DSP系统中应用面临的基本问题之一，如何高效地实现RNS动态范围的扩展对于其在DSP系统中应用有重要意义。

由于余数系统具有非权重的特性，故需要采用特殊的基扩展技术解决该问题。余数系统基扩展方法可以分为两类：第一类保留原余数基，通过增加新的余数基分量来扩大余数系统的动态范围[4-9]；第二类保留原余数系统的通道数量不变，通过增加原余数基的数据位宽来实现动态范围的扩大。

目前关于余数动态范围的扩展主要集中在第一类方法的研究上。文献[4]提出了Szabo-Tanaka算法，该算法利用了混合基转换(mixed radix conversion, MRC)和一个附加修正单元现基扩展，其实质为先做余数系统到二进制系统转换(residue to binary，R2B)，恢复出原数据后再对新余数基求模，算法复杂度较高。文献[5]讨论了两通道余数基 $\{m-1,m+1\}$ 向第三个余数基分量 $\{m\}$ 扩展的方法，同时，该算法可以推广至基为 $\{nm-1,nm+1\}$ 的余数系统基扩展(其中 n 为正整数)。文献[6]在文献[5]的基础上提出了通用的两通道余数系统向三通道余数系统的扩展方法。文献[7]提出了以中国剩余定理(chinese

收稿日期：2013－09－04；修回日期：2015－01－21
基金项目：国家自然科学青年基金(61101033)；特殊环境机器人技术四川省重点实验室开放基金(13zxtk02)
作者简介：马上(1978－)，男，博士，副教授，主要从事电路理论、通信信号基带处理等方面的研究.

remainder theorem, CRT)为基础并结合冗余基实现第一类基扩展的通用方法，但计算中仍包含完整的R2B转换。文献[8]提出一种基于改进中国剩余定理来实现第一类基扩展的方法，该算法首先改进了中国剩余定理，并利用查找表(look-up table，LUT)实现基扩展，再基于改进后的CRT同时可以实现缩放操作，该算法不需要冗余基。文献[9]提出了利用文献[8]中基扩展算法实现缩放，并认为比文献[4,7]提出的算法更加有效。

第一类基扩展方法的研究均采用余数系统后向转换算法为理论基础。在其实现中通常需要余数系统到二进制系统转换和二进制到余数系统转换(binary to residue，B2R)过程，算法复杂度太高。第二类余数基扩展方法则保留了原余数基的原有特点，仅增加各通道的位宽，保留了原余数基运算通道的电路结构，且不需要考虑扩展基与原始基的互质条件。另一方面，在DSP应用中如乘法级联为特征的离散付氏变换、离散余弦变换和离散小波变换等，需要方便高效地将余数通道的数据位宽扩大一倍。因此，第二类基扩展具有重要的应用价值，目前对这一方法的研究还较少。

针对目前研究最多且最为深入的一组基$\{2^n-1, 2^n, 2^n+1\}$，本文提出了一种基为$\{2^n-1, 2^n, 2^n+1\}$的余数系统扩展为基为$\{2^{2n}-1, 2^{2n}, 2^{2n}+1\}$余数系统的算法及其VLSI实现结构。首先，根据在扩展基中$2^{2n}-1$通道恰好为原始基中2^n-1通道与2^n+1通道之积的特点，利用MRC完成根据原始基中这两个通道数值扩展出扩展基中$2^{2n}-1$通道余数值的计算；其次，根据扩展基中$2^{2n}-1$通道的余数值和原始基中2^n通道的余数值利用同余的特性，完成对扩展基其余两路余数值的计算。本文提出的这种基扩展方法经过优化设计的基扩展电路VLSI实现结构仅需要简单的模加运算即可完成，不需要采用大容量的LUT处理，因此电路结构简单。采用基于单位门模型的性能分析方法，其性能对比表明在实现相同位宽的基扩展时，本文提出的算法和电路结构具有良好的面积和时延特性。

1 背景知识

1.1 余数系统与混合基转换

余数系统由一组两两互质的余数基$\{m_1, m_2, \cdots, m_L\}$定义，对于$0 \leqslant X < M$的整数在该余数系统中可表示为$(x_1, x_2, \cdots, x_L)$，其中，$x_i$为$X$模$m_i$的余数，记为$x_i = \langle X \rangle_{m_i}$ $(1 \leqslant i \leqslant L)$，$M = \prod_{i=1}^{L} m_i$为该余数系统的动态范围。中国剩余定理是余数系统的基本理论之一，是解决如大小比较、R2B转换、数值缩放、基扩展等一系列关键问题的理论基础。

在CRT中，主要问题是模M运算，当动态范围较大时，降低RNS的并行特性。而余数系统有与之相对应的混合基系统，它们具有相同的动态范围，于是有混合基转换(mixed radix conversion, MRC)。

令RNS余数基为$\{m_L, m_{L-1}, \cdots, m_2, m_1\}$，混合基系统的基与之相同，整数$X$的RNS表示为$(x_L, x_{L-1}, \cdots, x_2, x_1)$，MRS表示为$(z_L, z_{L-1}, \cdots, z_2, z_1)$，MRC为由$x_i$求解$z_i$的过程，有：

$$\begin{cases} z_1 = x_1 \\ z_2 = \left\langle \left\langle m_1^{-1} \right\rangle_{m_2} (x_2 - z_1) \right\rangle_{m_2} \\ z_3 = \left\langle \left\langle (m_2 m_1)^{-1} \right\rangle_{m_3} (x_3 - (z_2 m_1 + z_1)) \right\rangle_{m_3} \\ \quad \vdots \\ z_L = \left\langle \begin{matrix} \left\langle (m_{L-1} \cdots m_2 m_1)^{-1} \right\rangle_{m_L} \times \\ (x_{L-1} - (z_{L-1} m_{L-2} \cdots m_1 + \cdots + z_2 m_1 + z_1)) \end{matrix} \right\rangle_{m_n} \end{cases} \tag{1}$$

1.2 基扩展定义

基扩展定义为：由基为$\{m_1, m_2, \cdots, m_J\}$的余数系统整数$(x_1^m, x_2^m, \cdots, x_J^m)$求解其在基为$\{n_1, n_2, \cdots, n_I\}$的余数系统中的表示$(x_1^n, x_2^n, \cdots, x_I^n)$过程。若$I > J$且$m_1 = n_1, m_2 = n_2, \cdots, m_L = n_L$，那么为第一类基扩展；若$I = J$且余数基$\{m_1, m_2, \cdots, m_J\}$与余数基$\{n_1, n_2, \cdots, n_I\}$的相应通道具有相同的形式，则为第二类基扩展。本文的基扩展属于第二类基扩展。

2 余数通道数据位增加的基扩展算法及其VLSI实现结构

令整数X在原始余数系统$\{m_1, m_2, m_3\} = \{2^n-1, 2^n, 2^n+1\}$和扩展目标余数系统$\{m_1', m_2', m_3'\} = \{2^{2n}-1, 2^{2n}, 2^{2n}+1\}$中的表示分别为$(x_1, x_2, x_3)$和$(x_1', x_2', x_3')$，下面将基于MRC原理并结合该余数基的特点详细介绍由(x_1, x_2, x_3)求解(x_1', x_2', x_3')的低复杂度算法及其VLSI实现结构。

2.1 $2^{2n}-1$通道扩展

$2^{2n}-1$通道具有一定的特殊性，它实际上为原始余数基中m_1和m_3分量的乘积，即$2^{2n}-1 = (2^n-1)(2^n+1)$。因此，该通道的扩展可以认为是两通道余数系统$\{2^n-1, 2^n+1\}$的R2B转换[10]。采用

MRC，有：

$$\begin{cases} z_1 = x_3 \\ z_2 = \left\langle \left\langle m_3^{-1} \right\rangle_{m_1} (x_1 - z_1) \right\rangle_{m_1} \end{cases} \tag{2}$$

可知 $\left\langle m_3^{-1} \right\rangle_{m_1} = 2^{n-1}$，于是有：

$$x_1 = z_2(2^n + 1) + z_1 = \left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}(2^n + 1) + x_3 \tag{3}$$

对于式(3)，可进行适当优化，以简化VLSI实现结构。对于 $\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}$，有：

$$\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1} = \left\langle 2^{n-1} \left\langle x_1 - x_3 \right\rangle_{2^n-1} \right\rangle_{2^n-1} \tag{4}$$

由端回进位的性质可知，令 $\left\langle x_1 - x_3 \right\rangle_{2^n-1}$ 的二进制表达为 $y_{n-1}y_{n-2}\cdots y_1 y_0$，则式(4)的结果为：$y_0 y_{n-1}y_{n-2}\cdots y_1$ [11]。

对于 $\left\langle x_1 - x_3 \right\rangle_{2^n-1}$，由于有 $0 \leqslant x_1 \leqslant 2^n - 2$，$0 \leqslant x_3 \leqslant 2^n$，故有：

$$\left\langle x_1 - x_3 \right\rangle_{2^n-1} = \left\langle x_1 - \left\langle x_3 \right\rangle_{2^n-1} \right\rangle_{2^n-1} \tag{5}$$

模 $2^{2n}-1$ 的减法可以等效为将减数各位取反后的模 $2^{2n}-1$ 加法[11]。故式(5)中的模减法可以转化为模加法：

$$\left\langle x_1 - \left\langle x_3 \right\rangle_{2^n-1} \right\rangle_{2^n-1} = \left\langle x_1 + \overline{\left\langle x_3 \right\rangle_{2^n-1}} \right\rangle_{2^n-1} \tag{6}$$

式中，当 $0 \leqslant x_3 \leqslant 2^n - 1$ 时，$\left\langle x_3 \right\rangle_{2^n-1} = x_3$；当 $x_3 = 2^n$ 时，$\left\langle x_3 \right\rangle_{2^n-1} = 1$。故在 VLSI 设计中，可以通过 x_3 的最高位对其最低位进行简单的修正求出 $\left\langle x_3 \right\rangle_{2^n-1}$ 值。

由于 $\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}$ 的结果为 n 位，则式(3)中乘法 $\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}(2^n + 1)$ 的 $2n$ 位结果为：高 n 位为 $\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}$，低 n 位也为 $\left\langle 2^{n-1}(x_1 - x_3) \right\rangle_{2^n-1}$。

经优化后的 $2^{2n}-1$ 通道扩展VLSI实现结构如图1所示，可以看出其电路结构非常简单。

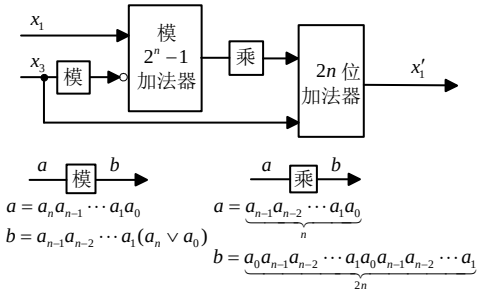

图1　$2^{2n}-1$ 通道扩展VLSI实现结构

2.2　2^{2n} 及 $2^{2n}+1$ 通道扩展

由于 $2^{2n}-1$ 通道的扩展是将原始余数基 m_1 和 m_3 进行后向转换得到，因此经过 $2^{2n}-1$ 通道的扩展，整数 X 可在一个新的余数系统 $\{2^{2n}-1, 2^n\}$ 中表示为 (x_1', x_2)。根据同余定理，必然有：

$$\begin{cases} X = a_1'(2^{2n}-1) + x_1' & \text{(7a)} \\ X = a_2 2^n + x_2 & \text{(7b)} \end{cases}$$

式中，a_1'、a_2 为整数，且 $0 \leqslant a_1' < 2^n$，$0 \leqslant a_2 < 2^{2n}-1$。式(7a)与式(7b)相减，有：

$$a_1'(2^{2n}-1) - a_2 2^n = x_2 - x_1' \tag{8}$$

$$\left\langle -a_1' \right\rangle_{2^n} = \left\langle x_2 - x_1' \right\rangle_{2^n} \tag{9}$$

由于 $0 \leqslant a_1' \leqslant 2^n - 1$，故有：

$$a_1' = 2^n - \left\langle x_2 - x_1' \right\rangle_{2^n} = \left\langle x_1' - x_2 \right\rangle_{2^n} \tag{10}$$

对于基为 $\{2^{2n}-1, 2^{2n}\}$ 的余数系统，已知数 X 在 $2^{2n}-1$ 通道的商 a_1' 和余数 x_1'，设 2^{2n} 通道的商和余数分别为 a_2' 和 x_2'，由同余定理有：

$$\begin{cases} X = a_1'(2^{2n}-1) + x_1' & \text{(11a)} \\ X = a_2' 2^{2n} + x_2' & \text{(11b)} \end{cases}$$

式中，$0 \leqslant a_1' < 2^{2n}$，$0 \leqslant a_2' < 2^{2n}-1$，式(11a)与式(11b)相减，并对 2^{2n} 取模，有：

$$x_2' = \left\langle a_1'(2^{2n}-1) + x_1' - a_2' 2^{2n} \right\rangle_{2^{2n}} = \left\langle x_1' - a_1' \right\rangle_{2^{2n}} \tag{12}$$

对于基为 $\{2^{2n}-1, 2^{2n}+1\}$ 的余数系统，已知数 X 在 $2^{2n}-1$ 通道的商 a_1' 和余数 x_1'，设 $2^{2n}+1$ 通道的商和余数分别为 a_3' 和 x_3'，同理，有：

$$x_3' = \left\langle a_1'(2^{2n}-1) + x_1' - a_3'(2^{2n}+1) \right\rangle_{2^{2n}+1} = \left\langle x_1' - 2a_1' \right\rangle_{2^{2n}+1} \tag{13}$$

由上所述，2^{2n} 及 $2^{2n}+1$ 通道的扩展需要 3 个模减运算：模 2^n 减法、模 2^{2n} 减法和模 $2^{2n}+1$ 减法。由于目前对模减法没有专门深入的研究，通常将其转化为模加法进行设计。

对式(10)，有 $\left\langle x_1' - x_2 \right\rangle_{2^n} = \left\langle \left\langle x_1' \right\rangle_{2^n} + \left\langle 2^n - x_2 \right\rangle_{2^n} \right\rangle_{2^n}$，令 x_2 的二进制表达为 $x_{n-1}^2 x_{n-2}^2 \cdots x_1^2 x_0^2$，则 $\left\langle 2^n - x_2 \right\rangle_{2^n}$ 为 $10\underbrace{\cdots 00}_{n} + 1\overline{x_{n-1}^2}\,\overline{x_{n-2}^2} \cdots \overline{x_1^2}\,\overline{x_0^2} + 1$，取低 n 位，结果为 $\overline{x_{n-1}^2}\,\overline{x_{n-2}^2} \cdots \overline{x_1^2}\,\overline{x_0^2} + 1$ 低 n 位，即 $\left\langle 2^n - x_2 \right\rangle_{2^n} = \left\langle \overline{x_2} + 1 \right\rangle_{2^n}$。因此，有 $\left\langle x_1' - x_2 \right\rangle_{2^n} = \left\langle \overline{x_2} + \left\langle x_1' \right\rangle_{2^n} + 1 \right\rangle_{2^n}$。同理，对于式(13)，$\left\langle x_1' - 2a_1' \right\rangle_{2^{2n}+1} = \left\langle x_1' + \overline{2a_1'} + 2 \right\rangle_{2^{2n}+1}$。

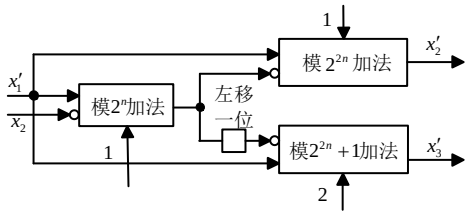

图2　2^{2n} 及 $2^{2n}+1$ 通道扩展VLSI实现结构

由此，3 个模减法器均转化为模加法器，具体

实现电路如图 2 所示，其电路结构非常简单。

由 2.1 节和 2.2 节的分析可知，本文的扩展算法中，需先扩展出 $2^{2n}-1$ 通道的值，然后扩展出其他两路的值，其整体实现框图如图 3 所示。

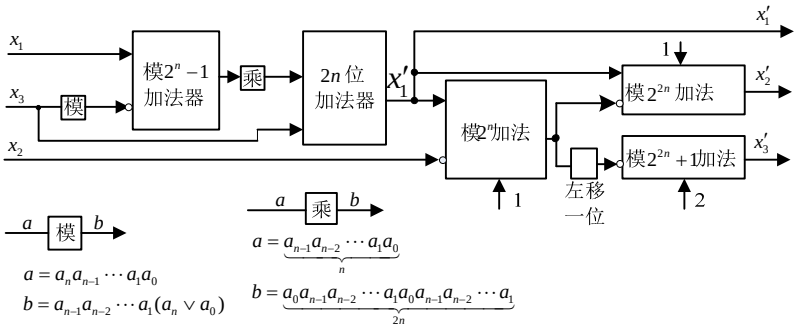

图3 整体设计 VLSI 实现结构

3 性能分析与对比

3.1 性能分析

由图2可知，本文提出的扩展算法需要一个 x_3 输入预处理模块，一个模 2^n-1 加法器，一个 $2n$ 位加法器，一个模 2^n 加法器，一个模 2^{2n} 加法器，一个模 $2^{2n}+1$ 加法器。另外需要一个乘 2^n+1 模块和一个数据左移模块，但是它们不消耗任何逻辑资源，且没有时延，故性能分析中不包含它们。模 2^n 加法器就是普通 n 位加法器，而模 2^n-1 和模 2^n+1 形式加法器是目前基于二进制最简单的模加法器，文献[12-14]都对此做过详细的研究。二进制加法是一个典型的前缀问题，并行前缀运算是加法器设计中被广泛采用的方法[13-14]。故本文中加法器均采用并行前缀方式进行构建，并且均采用SK前缀树。为了避免集成电路工艺和EDA工具对性能评估的影响，本文采用单位门模型进行算法的性能分析，在单位门模型中，两输入的“与”(AND)门、“或”(OR)门、“与非”(NAND)门和“或非”(NOR)门等为单位门，其面积为 A_g，时延为 τ_g；而对于两输入“异或”(XOR)和“同或”(XNOR)这类复杂的门为单位门的两倍，其面积和时延分别为 $2A_g$ 和 $2\tau_g$。根据文献[13]，模 2^n-1 加法器的面积为 $(1.5n\log n+8n)A_g$，时延为 $(2\log n+7)\tau_g$；根据文献[14]，采用CSA加端回进位设计的模 2^n+1 加法器的面积为 $[1.5(n-1)\times \log(n-1)+12(n-1)]A_g$，时延为 $[2\log(n-1)+10]\tau_g$；采用并行前缀方式构建模 2^n 加法器和 n 比特加法器，其面积和时延相同，分别为 $(1.5n\log n+5n)A_g$ 和 $(2\log n+4)\tau_g$；根据文献[12]，对于基于并行前缀的模加法器，有为常数项的第三输入，若输入为1，则仅需要改变模加法器中部分逻辑结构，不改变实现面积及时延；若输入为非1，则采用简化的进位保留加法器(simple carry saved adder，SCSA)将3输入变成2输入形式，此时需要多一级数据预处理，故有面积增加 $3nA_g$，时延增加 $2\tau_g$。x_3 输入预处理模块需要一个“或”(OR)门，故面积和时延分别增加 A_g 和 τ_g。

综上所述，本文提出的基扩展需要的面积和关键时延分别为：

$$A_{\text{proposed}} = [1.5(2n-1)\log(2n-1) + 9n\log n + 66n - 11]A_g \tag{14}$$

$$\tau_{\text{proposed}} = [2\log(2n-1) + 6\log n + 27]\tau_g \tag{15}$$

根据本文提出的基扩展算法，若原始余数系统动态范围为 $3n$ bits，则扩展后动态范围为 $6n$ bits。那么扩展每比特位宽所消耗资源为：$A_{\text{per}} = A_{\text{proposed}}/3n$，$\tau_{\text{per}} = \tau_{\text{proposed}}/3n$。

3.2 性能对比

对于Szabo-Tanaka算法，文献[7]及文献[8]均为第一类基扩展，而本文提出的算法为第二类基扩展。限定原始余数基均为 $\{2^n-1, 2^n, 2^n+1\}$，并设其扩展出的余数基分量为 $2^{3n}-1$，那么可在扩展相同的 $3n$ 比特位宽条件下进行性能对比。

表1给出了本文算法、Szabo-Tanaka算法，文献[7]及文献[8]提出的算法在扩展 $3n$ 比特位宽的条件下的硬件消耗、时延性能对比(基于单位门模型)。

在集成电路实现中，查找表中每比特存储单元所需CMOS管为6个[15]，而相对应的单位门模型中简单门同样需要6个CMOS管，故可大致认为查找表中每比特存储单元的面积消耗对应于单位门模型的面积为 A_g；同时可设查找表的平均查找时延为 τ_{LUT}。据此，可将查找表的硬件、时延性能转换为单位门模型进行分析。

表1　硬件性能及扩展基对比

算法	面积(A_g)	时延(τ_g)	扩展动态范围
Szabo-Tanaka算法	$2^{n+1}n+3(2^n+1)(n+1)+9n\log 3n+$ $3(n-1)\log(n-1)+1.5n\log n+77n-24$	$4\log(n-1)+2\log(3n-1)+27+3\tau_{\mathrm{LUT}}$	$2^{3n}-1$
文献[7]	$4.5(n-1)\log(n-1)+2^{n+2}(n-1)+3\times 2^n(3n-1)+$ $4.5(3n-1)\log(3n-1)+96n-48$	$6\log(3n-1)+21+\tau_{\mathrm{LUT}}$	$2^{3n}-1$
文献[8]	$9n\times 2^{3n+3}+2^{3n}\times 3n+2^{6n+1}\times 3n$	$3\tau_{\mathrm{LUT}}$	$2^{3n}-1$
本文算法	$1.5(2n-1)\log(2n-1)+9n\log n+66n-11$	$2\log(2n-1)+6\log n+27$	$2^{6n}-2^{3n}-2^{2n}+2^n$

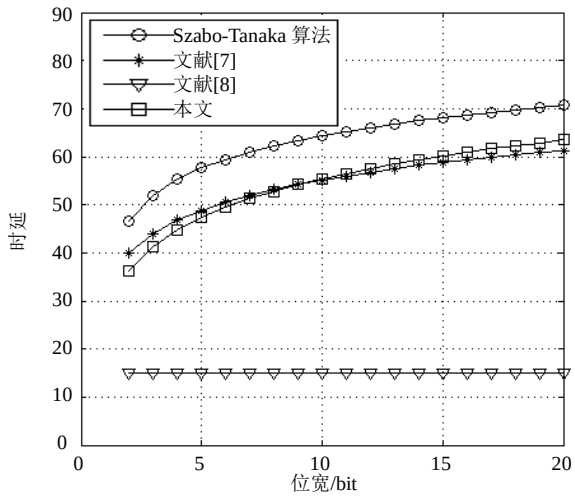

图 4　基于单位门模型的时延对比

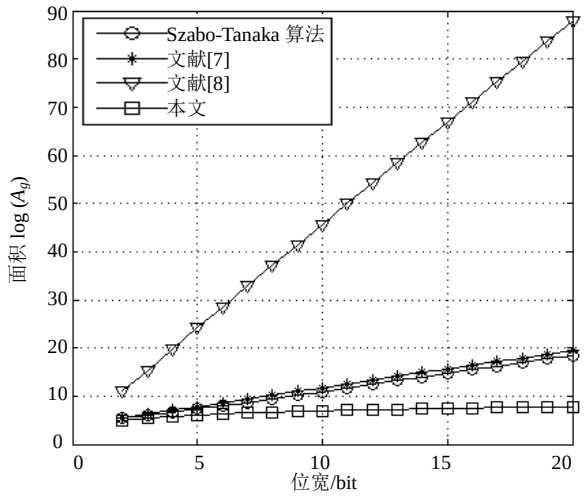

图 5　基于单位门模型的面积对比

图4图5分别给出了随着 n 增加，各算法基于单位门模型分析的时延和面积对比。可以看出本文算法在实现上具有最优的面积性能，其面积消耗随 n 的增加，基本为线性增加。其余3种算法由于使用查找表，实现面积随 n 的增加均为指数增加，随着 n 的增加，文献[8]所需的面积增长最快，文献[7]与Szabo-Tanaka算法所需的面积较为接近。而在时延上，文献[8]最优，文本算法在位宽10 bits以下，比文献[7]更优，位宽再增加，则比文献[7]略高，但总体与其相差不大，Szabo-Tanaka算法时延性能最差。

由于Szabo-Tanaka算法为较经典的第一类基扩展算法，其他第一类基扩展算法大多基于该思想进行设计，因此其对第一类基扩展算法具有较好的代表性。为了进一步进行性能分析和对比，基于VHDL语言对本文所提出的基扩展算法和经典的Szabo-Tanaka算法进行设计，其中所需的模 2^n-1 加法器的设计采用了端回进位法，模 2^n+1 加法器设计采用了消“1”法。然后利用Synopsys公司的Design Compiler对这些设计进行面向ASIC的综合，综合中采用了DC的Class库，工艺为SMIC 130 nm，电压设置为1.08 V，温度为125 ℃。DC实现结果如表2所示。

表2　ASIC综合结果

位宽n/bit	参数	本文	Szabo-Tanaka算法
4	面积/μm^2	3 517	3 332
	时延/ns	5.299 15	5.698 41
6	面积/μm^2	7 823	4 345
	时延/ns	5.598 68	5.794 61
8	面积/μm^2	10 972	7 452
	时延/ns	5.799 76	5.899 36
10	面积/μm^2	16 108	14 502
	时延/ns	5.898 12	6.887 85

由表2可见，本文提出的基扩展方法在时延方面占有较大优势，但面积与理论分析较为不同，分析原因是由于Szabo-Tanaka算法中的模加法器形式较为统一，因此综合软件采用了较多的复用，从而使得实际综合面积比理论分析面积小。

4　结　论

本文提出了一种基为 $\{2^n-1,2^n,2^n+1\}$ 的余数系统向基为 $\{2^{2n}-1,2^{2n},2^{2n}+1\}$ 的余数系统扩展的方法。首先，利用MRC恢复出 $\{2^n-1,2^n+1\}$ 的值，即扩展基中 $2^{2n}-1$ 路的余数值，然后利用该余数值和原始基中 2^n 路的值计算出扩展基中另外两路的余数值，且算法中的所有计算仅需要加法运算和模加运算。在扩展过程中充分结合了余数基特点进行优化。其VLSI实现结构简单，仅需要普通加法器和模加法器即可实现，避免了传统基扩展中使用的LUT。性能分析及对比结果表明，若采用并行前缀方式构

建实现结构中的加法器，该算法具有最优的面积消耗性能和较为优良的时延性能。

参 考 文 献

[1] CONWAY R, NELSON J. Improved RNS FIR filter architectures[J]. IEEE Transactions on Circuit and Systems II, 2004, 51(1): 26-28.

[2] MADHUKUMAR A S, CHIN F. Enhanced architecture for residue number system-based CDMA for high-rate data transmission[J]. IEEE Transactions on Wireless Communications, 2004, 3(5): 1363-1368.

[3] MA Shang, HU Jian-hao, LING Xiang, et al. The applications of RNS in SDR systems[C]//2008 International Workshop on Software Radio Technology (SRT2008). Beijing: [s.n.], 2008: 49-54.

[4] SZABO N S, TANAKA R I. Residue arithmetic and its applications to computer technology[M]. New York: McGraw-Hill, 1967.

[5] O'KEEFE K H, WRIGHT J L. Remarks on base extension for modular arithmetic[J]. IEEE Trans Comput, 1973, 22: 833-835.

[6] O'KEEFE K H. A note on fast base extension for residue number systems with three moduli[J]. IEEE Trans Comput, 1975, 24: 1132-1133.

[7] SHENOY A P, KUMARESAN R. Fast base extension using a redundant modulus in RNS[J]. IEEE Trans Comput, 1989, 38: 292-296.

[8] BARSI F, PINOTTI M C. Fast base extension and precise scaling in RNS for look-up table implementations[J]. IEEE Trans Signal Processing, 1995, 43: 2427-2430

[9] LAI Yu-feng, KONG Yi-nan. An implementation of a scaler in the residue number system[C]//International Symposium on Communications and Information Technologies. [S.l.]: [s.n.], 2012: 529-532

[10] WANG Yu-ke. New Chinese remainder theorems[C]// Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems & Computers. Pacific Grove: [s.n.], 1998, 1: 165-171.

[11] MA Shang, HU Jian-hao, ZHANG Lin, et al. An efficient RNS parity checker for moduli set $\{2^n-1, 2^n+1, 2^{2n}+1\}$ and its applications[J]. Science in China Series F: Information Sciences, 2008, 51(10): 1563-1571.

[12] MA Shang, HU Jian-hao, WANG Chen-hao. A novel modul 2^n-2^k-1 adder for residue number system[J]. IEEE Transactions on Circuits and Systems-I, 2013, 60(11): 2962-2972.

[13] PATEL R A, BOUSSAKTA S. Fast parallel-prefix architectures for modulo 2^n-1 addition with a single representation of zero[J]. IEEE Trans on Computers, 2007, 56(11): 1484-1492.

[14] EFSTATHIOU C, VERGOS H T, NIKOLOS D. Fast parallel-prefix modulo 2^n+1 adders[J]. IEEE Trans on Computers, 2004, 53(9): 1211-1216.

[15] WESTE H E, HARRIS D M. CMOS VLSI Design[M]. 2nd ed. [S.l.]: Addison Wesley, 2005.

编 辑 税 红