Investigation of "A Signed-Digit Architecture for Residue to Binary Transformation", Simulation and Literature survey

James Wiebe

Wed, Dec 8, 2004

Introduction

There has been interest in Residue Number Systems (RNS) as a basis for computational hardware since the 1950s [16], due to the inherent properties of RNS such as parallelism, modularity and carry-free operations.

Unfortunately, other arithmetic operations like division, comparison and sign detection are hard [4]. For this reason, it is necessary to convert the results of RNS calculations back to a weighted number system such as binary.

Available Methods

- Chinese Remainder Theorem
- Mixed-Radix Conversion
- Core function [5]

Presentation Structure

- Concept overview
- Theorems
- Converters – two-modulus and three-modulus
- Simulation
- Performance Analysis
- Literature Survey
- Conclusions

3

Concept overview

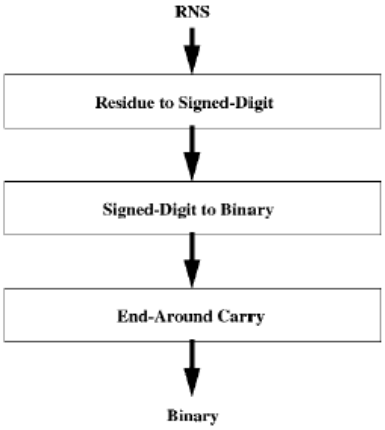
- The moduli set is introduced

$$S^k = \left\{2^m - 1, 2^{2^0 m} + 1, 2^{2^1 m} + 1, 2^{2^2 m} + 1, \ldots, 2^{2^k m} + 1\right\}$$

- This moduli set has the useful property that the multiplicative inverses used in the CRT will be powers of 2.
- Modulo operations can be replaced by input bit positioning in hardware and an end-around carry operation.
- The trade-off is that the conjugate moduli that give this benefit result in subtraction operations being required and signed digits are used here.

4

Three-stage R-B conversion



5

Theorems

the k+2 members of S^k are relatively prime.

$\gcd(2^{2^j m}+1,\ 2^{2^k m}+1) = 1$, for all k>j.

$\left|N_i^{-1}\right|_{m_i}$ is always a power of two.

- Shown for the first modulus
- Shown for the rest, of the form

$$m_i = 2^{2^{i-2} m} + 1$$

6

Notation

Let $\left|X\right|_m$ be "X modulo m"

let $\left|X^{-1}\right|_m$ be the multiplicative inverse of $\left|X\right|_m$

let $M = \prod m_i$

let $N_i = M/m_i$

let gcd(x, y) be the greatest common divisor of x and y.

Theorem: the k+2 members of S^k are relatively prime.

Proof: First it is shown that gcd($2^m - 1$, $2^{2^j m} + 1$) = 1. Since $\left|2^m\right|_{2^m-1} = 1$ (it is one more than the modulus), it is implied that $\left|2^{2^j m}\right|_{2^m-1} = \left|\left(2^m\right)^{2^j}\right|_{2^m-1} = 1$, because each modulus of 1 multiplied together 2^j times will still equal 1. As a result, $\left|2^{2^j m} + 1\right|_{2^m-1} = 2$. If there is a divisor $x \neq 1$ where $\left|2^m - 1\right|_x = \left|2^{2^j m} + 1\right|_x = 0$, then $\left|2\right|_x = 0$, which means that x=2. However, this is a contradiction, since all moduli are odd numbers.

2. It is shown that gcd($2^{2^j m}+1$, $2^{2^k m}+1$) = 1, for all k>j. Since $\left|2^{2^j m}\right|_{2^{2^j m}+1} = \left|-1\right|_{2^{2^j m}+1}$ (it is one less than the modulus), it is implied that $\left|2^{2^k m}\right|_{2^{2^j m}+1}$ = (via exponent algebra) $\left|\left(2^{2^k m}\right)^{2^{k-j}}\right|_{2^{2^j m}+1} = \left|(-1)^{2^{k-j}}\right|_{2^{2^j m}+1}$ (just multiplied 2^{k-j} times); and = 1. As a result, $\left|2^{2^k m}+1\right|_{2^{2^j m}+1}$ = 2, since the inner term is one more than that which had modulus 1. Similarly, if $\left|2^{2^j m}+1\right|_x = \left|2^{2^k m}+1\right|_x = 0$, then $\left|2\right|_x = 0$, which is a contradiction. QED.
Note that, thanks to the conjugate moduli, the range of the moduli set S^k is [0,M), where $M = (2^m-1)(2^{2^0 m}+1)(2^{2^1 m}+1)(2^{2^2 m}+1)\ldots(2^{2^k m}+1) = (2^{2^{k+1} m}-1)$ (equ. 2), since $(2^m-1)(2^{2^0 m}+1) = (2^{2^1 m}-1)$ (i.e., if k=0, we have 1 as a power of two after multiplying the 0-order conjugate modulus).

$\left|N_i^{-1}\right|_{m_i}$ is always a power of two.

For the first modulus, $N_1 = M/m_1$
is equal to: $N_1 = (2^{2^0 m}+1)(2^{2^1 m}+1)(2^{2^2 m}+1)\ldots(2^{2^k m}+1)$ (equ. 4)

Due to the fact that $\left|2^{jm}\right|_{2^m-1} = 1$, each factor of N_1 modulo m_1 is equal to two ($\left|2^{jm}\right|_{2^m-1} = \left|\left(2^m\right)^j\right|_{2^m-1} = \left|(1)^j\right|_{2^m-1}$=1; each term has an additional 1 added). As a result: $\left|N_1\right|_{m_1} = \left|2.2.2...2\right|_{2^m-1} = \left|2^{k+1}\right|_{2^m-1}$ (equ. 5).
Therefore, $\left|N_1^{-1}\right|_{m_1} = 2^{m-(k+1)}$ (equ. 6). This can be verified by writing $\left|N_1 \times \left|N_1^{-1}\right|_{m_1}\right|_{m_1} = \left|2^m\right|_{2^m-1} = 1$, from equ. 5.

For the rest of the moduli of the form $m_i = 2^{2^{i-2}m} + 1$ $2 \le i \le k+2$

$|N_i|_{m_j} = \left|(2^m - 1)(2^{2^0 m} + 1)(2^{2^1 m} + 1)..(2^{2^{i-3} m} + 1)(2^{2^{i-1} m} + 1)..(2^{2^k m} + 1)\right|_{m_j}$ (equ. 7) – (note the $(2^{2^{i-2} m} + 1)$ term was divided out).

If the binomials up to and including the i-3 term are multiplied together,

$|N_i|_{m_j} = \left|(2^{2^{i-2} m} - 1)(2^{2^{i-1} m} + 1)(2^{2^i m} + 1)..(2^{2^k m} + 1)\right|_{m_j}$ (equ. 8)

The first term of this can be reduced to:

$\left|(2^{2^{i-2} m} - 1)\right|_{m_i} = \left|2^{2^{i-2} m}\right|_{m_i} - 1\Big|_{m_i} = |(-1) - 1|_{m_j} = |-2|_{m_i}$, since $m_i = (2^{2^i m} + 1)$. (equ. 9).

The range of the rest of the terms in (8) is $i-1 \le j \le k$ and for the rest of the terms we can write

$\left|2^{2^j m} - 1\right|_{m_i} = \left|\left(2^{2^{j-2} m}\right)^{2^{j-(i-2)}} + 1\right|_{m_i} = \left|(-1)^{2^{j-(i-2)}} + 1\right|_{m_j} = |2|_{m_j}$ (equ. 10)

which is obtained by noticing that j>i-2, given the range under consideration.

Substituting (9) and (10) into (8) gives:

$|N_i|_{m_j} = |(-2)(2)(2)..(2)|_{m_i} = \left|(-2)2^{k-(i-2)}\right|_{m_j}$ given the range of twos is exclusive of i-2 of the first term in (8) (which is -2).

Thus $|N_i|_{m_j} = \left|-2^{k-i+3}\right|_{m_j}$ (equ. 11).

Therefore, for all m_i where $2 \le i \le k+2$, the multiplicative inverses are calculated as $\left|N_i^{-1}\right|_{m_i} = 2^{2^{i-2} m-(k-i+3)}$ (equ. 12).

This can be verified by writing $\left|N_i \times \left|N_i^{-1}\right|_{m_1}\right|_{m_1} = \left|-2^{2^{i-2} m}\right|_{m_j}$ where $m_i = 2^{2^{i-2} m} + 1$. For example, if i=2 at the beginning of this range for the rest of the moduli, $\left|-2^m\right|_{2^m+1}$, the modulus of this is (-8) mod 9 = 1, since -8 = 9(-1) + 1, where -1 is the quotient and +1 is the remainder. See Koren [4], pg 260 section 11.1.

Equations (6) and (12) show that the multiplicative inverses for this moduli set are always powers of two. This allows simplification in the CRT equation where multiplications can be replaced by simple shift-left operations.

Also, if i=2 in (12), $\left|N_2^{-1}\right|_{m_2} = 2^{m-(k-2+3)} = 2^{m-(k+1)} = \left|N_1^{-1}\right|_{m_1}$ from equ. (6) (this is equ. 13).

Note that the result of equ (2) shows that the final modulo-M operation can be replaced by an end-around carry operation, because bits in the next group of $2^{k+1}m$ bits are units in the modulus and the dividend can be partitioned by $2^{k+1}m$ bits when the modulus is $(2^{2^{k+1}m}-1)$; see Koren [4] pg 271 example 11.11.

13

Two-moduli converter

When k=0, we have the S^0 moduli set = $\{2^m-1, 2^m+1\}$. This results in $M=2^{2m}-1$ and a number range of $[0, 2^{2m}-1)$, ie $[0, 2^{2m}-2]$.

The CRT in (3) reduces to:

$$\left|X\right|_M = \left|N_1\left|N_1^{-1}\right|_{m_1} x_1 + N_2\left|N_2^{-1}\right|_{m_2} x_2\right|_M \quad \text{(equ. 14)}$$

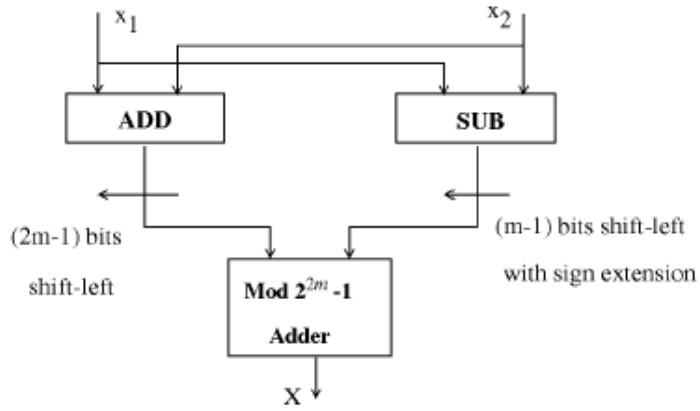
Substituting k=0 in (13), the multiplicative inverses $\left|N_1^{-1}\right|_{m_1}$ and $\left|N_2^{-1}\right|_{m_2}$ are both 2^{m-1}.

Substituting in (14), we have $\left|X\right|_M = \left|(2^m+1)2^{m-1}x_1 + (2^m-1)2^{m-1}x_2\right|_{2^{2m}-1}$ (equ. 15)

which can be rearranged into: $\left|X\right|_M = \left|2^{2m-1}(x_1+x_2) + 2^{m-1}(x_1-x_2)\right|_{2^{2m}-1}$ (equ 16).

14

Hardware realization concept



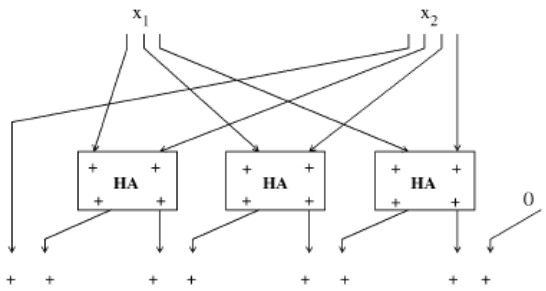
15

If m = 3, the S^0 moduli set = $\{2^3-1, 2^3+1\}$ and $M=2^6-1 = 63$ and a number range of $[0,63) = [0,62]$ results.
The CRT becomes

$$|X|_M = \left|2^5(x_1+x_2)+2^2(x_1-x_2)\right|_{2^6-1} \text{ (equ 17).}$$

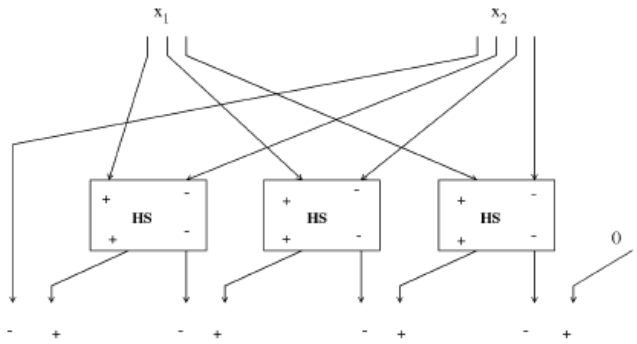
16

Developing the HW realization



The Add block

17



The Subtract block

18

If $x_1 = a_2a_1a_0$ and $x_2 = b_3b_2b_1b_0$ the input bits can be arranged to have already their shifted-left positions of 5 bits shifted left for the sum and 2 bits shifted left for the difference as shown in Fig. 5. Notice the partitioning to the left of the sixth bit, i.e., between bits 5 and 6. The nature of the $2^6 - 1$ modulo process will result in those bits being moved around to the lowest-order position.

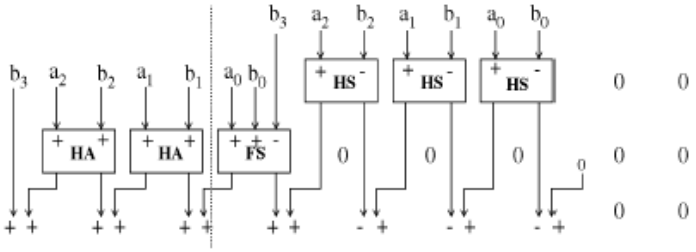


Fig 3.5. Results of the shift-left operation

Two-mod. HW - final

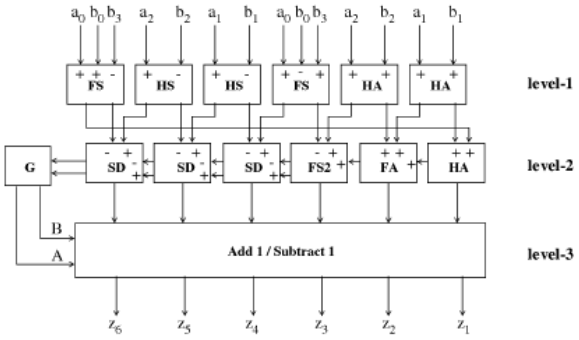


Fig 3.6. The two-moduli converter architecture.

What about without SDs?

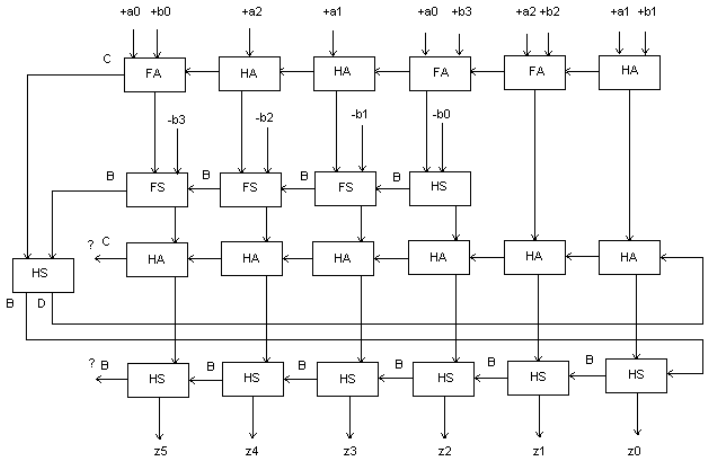
•Attempting an implementation without signed digits might not result in any simplification.
•I have attempted one, using only Full and Half Adders and Subtractors.
•Subtractors take a positive difference and generate a negative borrow, which has to be subtracted from the next stage.

•HS: D = x'y + xy', B = x'y [Mano, Digital Design [35]]
•FS: D = x'y'z + x'yz' + xyz' + xyz, B = x'y + x'z + yz = x'(y + z) + yz, x is the minuend, y is the subtrahend, z is the Borrow in, D is the difference, and B is the borrow out.

•Note that the equations for the difference are the same as those for the sum in an adder, and that the equation for the borrow in the FS is very similar to the equation for the carry in a FA; the only difference being the minuend is complemented.

21

Attempt without SDs



22

Three-Moduli Converter

When k=1, we have the S^1 moduli set = $\{2^m-1, 2^m+1, 2^{2m}+1\}$. This results in $M=2^{4m}-1$ and a number range of $[0, 2^{4m}-1)$, ie $[0, 2^{4m}-2]$.
The CRT in (3) reduces to:

$$|X|_M = \left| N_1 |N_1^{-1}|_{m_1} x_1 + N_2 |N_2^{-1}|_{m_2} x_2 + N_3 |N_3^{-1}|_{m_3} x_3 \right|_M \quad \text{(equ. 18)}$$

Substituting k=1 in (13), the multiplicative inverses $|N_1^{-1}|_{m_1}$ and $|N_2^{-1}|_{m_2}$ are both 2^{m-2}.

Substituting k=1 and i=3 in (12), $|N_3^{-1}|_{m_3} = 2^{2m-1}$.

Substituting in (18), we have $|X|_M =$

$$\left|(2^m+1)(2^{2m}+1)2^{m-2}x_1 + (2^m-1)(2^{2m}+1)2^{m-2}x_2 + (2^m-1)(2^m+1)2^{2m-1}x_3\right|_{2^{4m}-1} =$$

$$\left|(2^m+1)(2^{2m}+1)2^{m-2}x_1 + (2^m-1)(2^{2m}+1)2^{m-2}x_2 + (2^{2m}-1)2^{2m-1}x_3\right|_{2^{4m}-1} \quad \text{(equ. 19)}$$

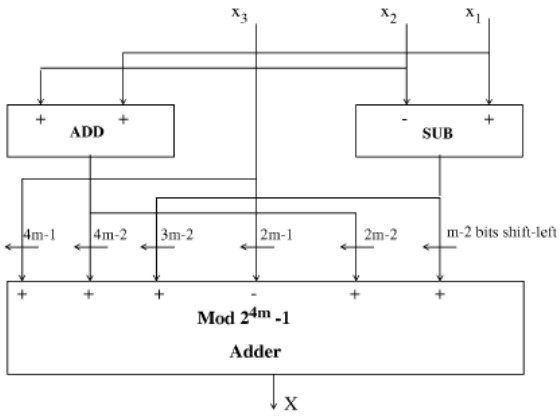
which can be rearranged into: $|X|_M =$

$$\left|(2^{4m-1})x_3 + (2^{4m-2})(x_1+x_2) + (2^{3m-2})(x_1-x_2) - 2^{2m-1}x_3 + (2^{2m-2})(x_1+x_2) + (2^{m-2})(x_1-x_2)\right|_{2^{4m}-1}$$

(equ 20).

23

3-mod. HW realization concept



24

If x1 = a2a1a0, x2 = b3b2b1b0, and x3 = c6c5c4c3c2c1c0, when m=3, Fig. 4.9 shows the shift-left concept.

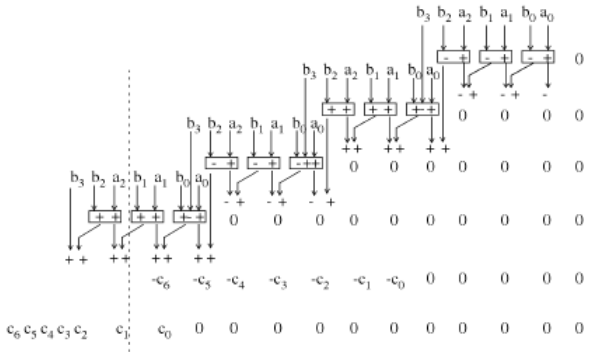


Fig. 4.9. Bits positioned already in their shifted-left positions

25

A four-operand adder/subtractor is used to group the pre-shifted positions of the operand bits as shown in Fig. 4.10.

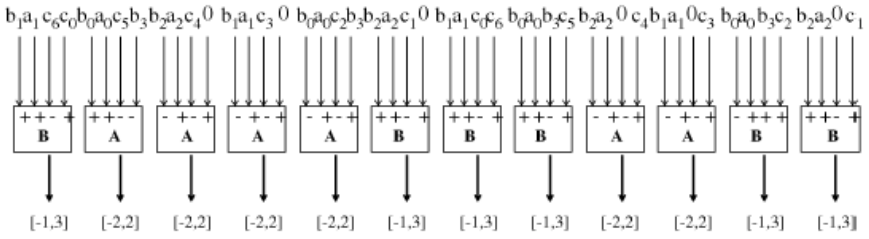
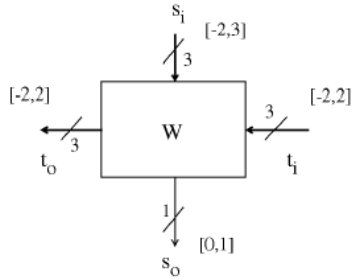


Fig. 4.10. Four-operand addition/subtraction

26

CPA level 2 of 3-mod converter

$s_i + t_i = 2\,t_o + s_o$



27

Simulation

- The adder, subtractor, SD-to-binary and level-3 multiplexing blocks were simulated as C++ objects in Microsoft Visual C++ version 6.0.
- The truth tables and/or equations are given, as applicable

28

Two-mod. simulation
SD HA Truth table

In1	In2	s	c
0	0	0	0
1	0	1	0
-1	0	-1	0
0	1	1	0
-1	1	0	0
1	1	0	1
0	-1	-1	0
1	-1	0	0
-1	-1	0	-1

29

SD FA Truth table

In1	In2	c_{in}	s	c_{out}
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1
0	0	-1	-1	0
1	0	-1	0	0
0	1	-1	0	0
1	1	-1	1	0

30

SD HS Truth table

Subtrahend	Subtractor	Diff	c
0	0	0	0
1	0	1	0
0	1	-1	0
1	1	0	0

31

SD FS Truth table

Subtrahend	Subtractor	c_{in}	Diff	c_{out}
0	0	0	0	0
1	0	0	1	0
0	1	0	-1	0
1	1	0	0	0
0	0	1	1	0
1	0	1	0	1
0	1	1	0	0
1	1	1	1	0

32

SD FS2 Truth table

Subtrahend	"Subtractor"*	c_{in}	Diff	$-c_{out}$	$+c_{out}$
0	0	0	0	0	0
1	0	0	1	0	0
0	1	0	1	0	0
1	1	0	0	0	1
0	-1	0	1	1	0
1	-1	0	0	0	0
0	0	1	1	0	0
1	0	1	0	0	1
0	1	1	0	0	1
1	1	1	1	0	1
0	-1	1	0	0	0
1	-1	1	1	0	0
0	0	-1	1	1	0
1	0	-1	0	0	0
0	1	-1	0	0	0
1	1	-1	1	0	0
0	-1	-1	0	1	0
1	-1	-1	1	1	0

33

SD-B and Carry-Multiplex blocks

- The SD–binary converter is used in two-mod. simulation level 2. The "–in" input has to be able to accept signed digits. It adds them in order to subtract them if they are negative. It adds the rest of the inputs, subtracting only the negative carry-in, which is defined to be positive. The truth table would have 24 rows (3 choices for "-in", 2 for "+in", 2 for "-cin" and 2 for "+cin") and is straightforward, and so is not given here.
- The GA block detects which if any of -cout or +cout results as the end-around carry and signals level 3 whether to add one, subtract one, or do nothing.

34

Simulation of level 1 initial concept (see Fig 3.5)

Using the residues (5,7), which code for 61 in the (2^3-1, 2^3+1) (7, 9) moduli set, the initial concept of level 1 is demonstrated

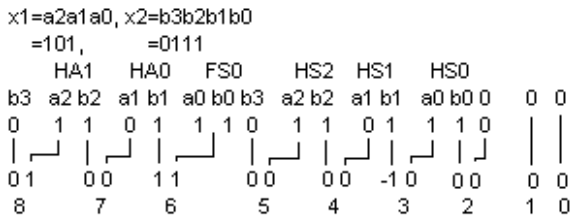


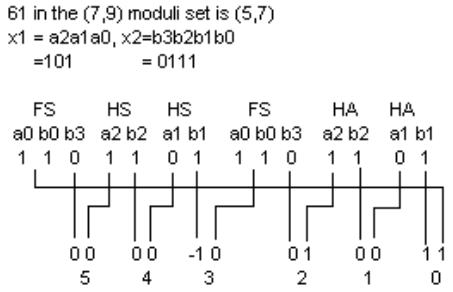
Fig. 5.1. Initial shift-left concept. ([1], Fig. 5)

It can easily be seen that this output equals 256 + 2x64 -8 = 376; $|376|_{63} = 61$

35

Level 1 simulation

Note that the inputs are preshifted, leaving only a possible end-around carry to be applied to level 2.



It can be seen that when the number's carries are added and the results are converted from signed digits, 1,1,1,1,1,0 with a -1 end-around carry will result, which will be 32 + 16 + 8 + 4 + 2 – 1 = 61

36

Two-mod. entire simulation

- This simulation consists of feeding the output from level 1 into the SD, FS2 and adder blocks explained above.
- Level 2 subtracts the negative part of the number from the positive part, making the inputs to the FS2 and SD blocks positive in sense so that negative digits will naturally be subtracted.

37

Three-moduli simulation

- The A and B four-input adder/subtractors are very simple; complexity lies in the bit-positioning of the inputs in order to achieve the required moduli.
- A simply subtracts the bit data on the subtracting inputs from that on the adding inputs, and B differs only in having only one subtracting input.
- CPA: W: $s_i + t_i = 2t_o + s_o$

  $s_o = \text{abs}(s_i+t_i) \bmod 2$, and $t_o = (s_i + t_i - s_o)/2$

38

Simulation of removal of CPA

- The carry propagates can be eliminated, or at least reduced, by parallel calculation of the level 1 output data from blocks A and B, eliminating the block W, which is a type of CPA.
- A conditional-sum simulation is presented.
- Has complexities of its own, due to 5 different carry possibilities in the range [-2,2]
- the s_i plus the possible carries can be easily generated from $s_i + 2$ to $s_i - 2$, the potential s_o are the absolute value of those modulo 2 and the potential t_o are those minus the potential s_o, the result divided by 2.

39

Performance Analysis

Two-Moduli converter

Block	Delay
HA	Δ
HS	Δ
FA	2Δ
FS	2Δ
FS2	2Δ
SD	3Δ
G	Δ

Table 6.1 Gate delays

40

Two-mod. converter performance

- Note that the SD block has to perform two half-additions in parallel, and then a full-subtraction, yielding three gate-delays.
- Level 1 takes 2?, level 2 takes ?(G), 3m?(SDs), 2?(FS2), (m-2)2? (FAs) plus 1? for the one HA and then 2m2? (max) for the final end-around carry FAs of level 3
- This totals (9m+2)?.
- For the moduli set of {5, 17}, which provides the dynamic range of [0, 85), the FA-based RNS to binary converter reported by Stouraitis [2] has a delay of 51?. Using $m = 4$, the two-moduli set of $\{2^4 - 1, 2^4 + 1\}$ provides the dynamic range of [0, 255). Referring to Table 6.2, for $m = 4$, the two-moduli converter here imposes an operational delay of 38?, giving less delay for a greater dynamic range.

41

Three-mod. converter performance

- Referring to Fig. 4.10, the blocks A and B have the same complexity and delay as of SD. The block W of Fig. 4.11 is a three-bit redundant adder introducing a delay of 5?. Here, the block G has a delay of 2?. The delays of the three conversion levels are:
- $D_1 = D_A = D_B = 3?$
- $D_2 = D_G + 4mD_W = 20m+2?$
- $D_3 + 4mD_{FA} = 8m?$

Resulting in a total delay of:

- $D = D_1+D_2+D_3 = (28m+5)?$

42

Literature survey

- From about the time period of P. and Y. [1] (~1997)

Name (et. al.)	Reference	Year	Moduli set	Delay claimed
Stouraitis	[2]	1992	$\{m_1, m_2) = (5,17)$	51?
Gallaher	[3]	1997	$(2^k\text{-}1, 2^k, 2^k+1)$	13 FAs for k=32
Hiasat	[6]	1998	$(2^k, 2^k\text{-}1, 2^{k-1}\text{-}1)$	-of 1 (2k-1)-bit bin. adder
Skavantos	[7]	1998	$m_1 = 2^i$, ? $_{rest} = 2^a - 1$	5FA + 1 mod $(2^{4n}\text{-}1)$ CPA
Yuke Wang	[9]	1998	$(2^n\text{-}1, 2^n, 2^n+1)$	$t_{FA} + t_{inv} + 2t_{CPA2n}$
Yuke Wang	[12]	2002	$(2^n\text{-}1, 2^n, 2^n + 1)$	$t_{inv} + t_{MUX} + t_{FA} + 2t_{CPA(2n)}$
Cardarilli	[13]	1998	$(n\text{-}2^k\ n+2^k)$	Not reported
Wei Wang	[19]	1999	Same as Y&P [1]	(8m+6)?
Bhardwaj	[20]	1999	$(2^n\text{-}1, 2^n, 2^n+1, 2^{n+1}+1)$	10n+13 FA
Wei Wang	[21]	1999	$(2^k, 2^k\text{-}1, 2^{k-1}\text{-}1)$	$(3k+2)t_{FA}$
Wei Wang	[22],[23]	2000 -2002	$(2^k\ 2^k - 1, 2^{k-1} - 1)$	Critical path: $(3k + 2)\ t_{FA}$

Literature survey pg. 1 of 2

43

Name (et. al.)	Reference	Year	Moduli set	Delay claimed
Conway	[24]	1999	$(2^n\text{-}1, 2^n, 2^n+1)$	$t_{OR} + t_{FA} + t_{CPA(n)} + t_{ANDOR} + 2t_{MUX}$
Cao	[31]	2003	$(2^n\text{–}1, 2^n, 2^n+1, 2^{n+1}\text{-}1)$	HS: $(5n + l + 5)t_{FA}$ CE: $(11n + l + 5)t_{FA}$
Cao	[32]	2003	$(2^n\text{–}1, 2^n, 2^n+1, 2^{2n}\text{–}1)$	HS: $2t_{MUX} + t_{INV} + 2t_{NAND} + (2n+3)*t_{FA}$ CE: $t_{MUX} + t_{INV} + 2t_{NAND} + (8n+3)*t_{FA}$
Cao	[33]	2004	$(2^n\text{–}1, 2^n, 2^n+1, 2^{n+1}\text{-}1, 2^{n-1}\text{–}1)$	20-40ns for 20-150 bit ranges.
Lindstrom	[34]	2003	$(2^n - 1, 2^n, 2^n + 1)$	16 bits: 3.04 32 bits: 4.63 64 bits: 6.93ns

Literature survey pg. 2 of 2

44

Conclusions

- The moduli set

$$S^k = \{2^m - 1,\ 2^{2^0 m} + 1,\ 2^{2^1 m} + 1\ 2^{2^2 m} + 1,\ \ldots,\ 2^{2^k m} + 1\}$$

was introduced.

- For this type of moduli, it was shown that the multiplicative inverses are powers of two.
- Therefore all the modulus operations can be replaced, not only by shift-left operations, but by strategic bit positioning, and only a final end-around carry.
- Residue to binary converter architectures based on the CRT, with respect to S^0 and S^1 (two and three) moduli sets, were developed.
- The conversion from R-B was performed in three stages, with SDs appearing to be very helpful in reducing hardware complexity.
- Simulation of both two and three-mod. converters was done, which establishes correctness of concept.
- Simulation of parallel computation of level 3 in the three-mod. converter was done, showing how the final W (CPA) blocks might be eliminated.
- An overview of the literature was presented, showing that the delays of these converters are O(log(n)) and that it is very difficult to make improvements.

45

References (1)

[1] **A signed-digit architecture for residue to binary transformation**, *Pourbigharaz, F., Yassine, H.M.;* Computers, IEEE Transactions on, Volume: 46, Issue: 10, Oct. 1997 Pages: 1146 – 1150

[2] **Efficient convertors for residue and quadratic-residue number systems** *Stouraitis, T.;* Circuits, Devices and Systems, IEE Proceedings G , Volume: 139 , Issue: 6 , Dec. 1992 Pages: 626–634

[3] **The digit parallel method for fast RNS to weighted number system conversion for specific moduli (2k-1,2k,2k +1)** *Gallaher, D.; Petry, F.E.; Srinivasan, P.;* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] , Volume: 44 , Issue: 1 , Jan. 1997 Pages: 53–57

[4] **Computer Arithmetic Algorithms, 2nd Edition,** *Koren, Israel;* A.K. Peters Limited, Natick, MA, 2002. ISBN 156881-160-8

[5] **Scaled and unscaled residue number system to binary conversion techniques using the core function**, Burgess, N.; Computer Arithmetic, 1997. Proceedings., 13th IEEE Symposium on, 6-9 July 1997 Pages: 250–257

[6] **Residue-to-binary arithmetic converter for the moduli set (2k , 2k-1, 2k-1-1)** *Hiasat, A.A.; Abdel-Aty-Zohdy, S.H.;*Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] , Volume: 45 , Issue: 2 , Feb. 1998 Pages: 204-209

[7] **An efficient residue to weighted converter for a new residue number system** *Skavantzos, A.;* VLSI, 1998. Proceedings of the 8th Great Lakes Symposium on , 19-21 Feb. 1998 Pages:185 – 191

46

References (2)

[8] **Comments on "The digit parallel method for fast RNS to weighted number system conversion for specific moduli (2k-1, 2k, 2k+1)"**, *Mohan, P.V.A.;* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] , Volume: 47 , Issue: 9, Sept. 2000, Pages:972 – 974

[9] **Residue to binary number converters for (2n-1, 2n , 2n+1)** *Yuke Wang; Xiaoyu Song; Aboulhamid, M.;* VLSI, 1998. Proceedings of the 8th Great Lakes Symposium on , 19-21 Feb. 1998 Pages:174 – 178

[10] **Residue to binary number converters for three moduli set**, *Yuke Wang; Swamy, M.N.S.; Ahmad, M.O.;* Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on , Volume: 5 , 31 May-3 June 1998 Pages:217 - 220 vol.5

[11] **Residue-to-binary number converters for three moduli sets**, *Yoke Wang; Swamy, M.N.S.; Abmad, M.O.;* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on], Volume:46, Issue: 2, Feb. 1999, Pages:180 - 183

[12] **Adder based residue to binary number converters for (2n-1, 2n, 2n+1)**, *Wang, Y.; Song, X.; Aboulhamid, M.; Shen, H.;* Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] , Volume: 50 , Issue: 7 , July 2002 Pages:1772 - 1779

47

References (3)

[13] **RNS-to-binary conversion for efficient VLSI implementation**, *Cardarilli, G.C.; Re, M.; Lojacono, R.;* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] , Volume: 45 , Issue: 6 , June 1998 Pages:667 – 669

[14] **Breaking the 2n-bit carry propagation barrier in residue to binary conversion for the [2n-1, 2n, 2n+1] modula set** *Bhardwaj, M.; Premkumar, A.B.; Srikanthan, T.;* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] , Volume: 45 , Issue: 9 , Sept. 1998 Pages:998 – 1002

[15] **Comments on "Breaking the 2n-bit carry-propagation barrier in residue to binary conversion for the [2n-1, 2n, 2 n+1] moduli set" and author's reply** *Mohan, P.V.A.; Premkumar, A.B.; Bhardwaj, M.;* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] , Volume: 48 , Issue: 8 , Aug. 2001 Pages:1031

[16] **New Chinese remainder theorems** *Yuke Wang;* Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on, Volume: 1 , 1-4 Nov. 1998 Pages:165 - 171 vol.1

48

References (4)

[17] **Efficient RNS to binary conversion using high-radix SRT division** *Burgess, N.;* Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on , Volume: 2 , 1-4 Nov. 1998 Pages:1240 - 1243 vol.2

[18] **Implementation issues of the two-level residue number system with pairs of conjugate moduli** *Skavantzos, A.; Abdallah, M.;* Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] , Volume: 47 , Issue: 3 , March 1999 Pages:826 – 838

[19] **A parallel residue-to-binary converter** *Wei Wang; Swamy, M.N.S.; Ahmad, M.O.; Wang, Y.;* Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on , Volume: 3 , 15-19 March 1999 Pages:1541 - 1544 vol.3

[20] **A reverse converter for the 4-moduli superset {2n-1, 2 n, 2n+1, 2n+1+1}** *Bhardwaj, M.; Srikanthan, T.; Clarke, C.T.;* Computer Arithmetic, 1999. Proceedings. 14th IEEE Symposium on , 14-16 April 1999

[21] **A high-speed residue-to-binary converter and a scheme for its VLSI implementation** *Wei Wang; Swamy, M.N.S.; Ahmad, M.O.; Yuke Wang;* Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on , Volume: 6 , 30 May-2 June 1999 Pages:330 - 333 vol.6

49

References (5)

[22] **A note on "A high-speed residue-to-binary converter for three-moduli (2k 2k - 1, 2k-1 - 1) RNS and a scheme for its VLSI implementation"** *Wei Wang; Swamy, M.N.S.; Ahmad, M.O.; Yuke Wang;* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] , Volume: 49 , Issue: 3 , March 2002 Pages:230 – 230

[23] **A high-speed residue-to-binary converter for three-moduli (2k , 2k-1, 2k-1-1) RNS and a scheme for its VLSI implementation** *Wei Wang; Swamy, M.N.S.; Ahmad, M.O.; Yuke Wang;* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on], Volume: 47, Issue: 12 , Dec. 2000 Pages:1576 – 1581

[24] **Fast converter for 3 moduli RNS using new property of CRT** *Conway, R.; Nelson, J.;* Computers, IEEE Transactions on , Volume: 48 , Issue: 8 , Aug. 1999 Pages:852 – 860

[25] **A systolic architecture for high-performance scaled residue to binary conversion** *Cardarilli, G.C.; Re, M.; Lojacono, R.; Ferri, G.;* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] , Volume: 47 , Issue: 10 , Oct. 2000 Pages:1523 – 1526

50

References (6)

[26] **Comments on "Residue-to-binary converters based on new Chinese remainder theorems"**, *Mohan, P.V.A.*; Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] , Volume: 47 , Issue: 12 , Dec. 2000 Pages:1541

[27] **Residue-to-binary converters based on new Chinese remainder theorems**, *Wang, Y.*, *IEEE* Trans. Circuits Syst. II, vol. 47, pp. 197–205, 2000.

[28] **A fully parallel mixed-radix-conversion algorithm for residue number applications**, *Huang, C.*, IEEE Trans. Comput., vol. C-32, pp. 398–402, 1983.

[29] **SRT Division Architectures and Implementations,** David L. Harris, Stuart F. Oberman, and Mark A. Horowitz {harrisd, oberman, horowitz}g@leland.stanford.edu Computer Systems Laboratory, Stanford University, Stanford, CA 94305, 1997. http://klabs.org/DEI/Arithmetic/division/dh_arith_97.pdf

[30] **Efficient RNS-to-binary conversion using high-radix SRT division** Burgess, N.; Computers and Digital Techniques, IEE Proceedings- , Volume: 148 , Issue: 1 , Jan 2001 Pages:49 – 52

[31] **New efficient residue-to-binary converters for 4-moduli set {2/sup n/ - 1, 2/sup n/, 2/sup n/ + 1, 2/sup n+1/ - 1}** *Bin Cao; Chip-Hong Chang; Srikanthan, T.;* Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on , Volume: 4 , 25-28 May 2003 Pages:IV-536 - IV-539 vol.4

References (7)

[32] **An efficient reverse converter for the 4-moduli set {2/sup n/ - 1, 2/sup n/, 2/sup n/ + 1, 2/sup 2n/ + 1} based on the new Chinese remainder theorem** *Bin Cao; Chip-Hong Chang; Srikanthan, T.;* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] , Volume: 50 , Issue: 10 , Oct. 2003 Pages:1296 – 1303

[33] **Design of residue-to-binary converter for a new 5-moduli superset residue number system**, *Cao, B.; Srikanthan, T.; Chip-Hong Chang;* Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on, Volume: 2 , 23-26 May 2004, Pages:II - 841-4 Vol.2

[34] **Arithmetic Circuits Combining Residue and Signed-Digit Representations**, *Lindström, Anders, Nordseth, Michael, Bengtsson, Lars, and Omondi, Amos*. Proceedings of the Eighth Asia-Pacific Computer Systems Architecture Conference (ACSAC 2003), Aizu-Wakamatsu City, Japan, Sept. 2003. Springer-Verlag, Lecture Notes in Computer Science (LNCS) Vol. 2823. http://www.etek.chalmers.se/groups/arithdb/files/doc/ACSAC-paper.pdf

[35] **Digital Design**, *Mano, M. Morris*, Prentice-Hall Inc., NJ 1984. ISBN 0-13-212333-9

Thank you!

53