TECHNISCHE UNIVERSITÄT DRESDEN

Residue Number System und Modulararithmetik

Patrick Russell

Dresden, 18. Juni 2015

Zahlendarstellung

RNS Systeme

Anwendungsbeispiele

Literatur

01 Zahlendarstellung

- Darstellung von Zahlen als Tupel von Resten in Bezug auf eine vorher definierte Menge teilerfremder Moduli

Gewichtete Zahl: $Z = (z_{(k-1)}, z_{(k-2)}, \ldots, z_0)$

Moduli Menge: $(m_{(n-1)}, m_{(n-2)}, \ldots, m_0)$

RNS-Darstellung: $(x_{(n-1)}, x_{(n-2)}, \ldots, x_0)$

$$x_i = Z \bmod m_i = |Z|_{mi}, \quad 0 \leq x_i < m_i$$

Vorzeichenlose RNS Darstellung (5,3,2)

N	Residue to base 5	3	2	N	Residue to base 5	3	2	N	Residue to base 5	3	2
0	0	0	0	10	0	1	0	20	0	2	0
1	1	1	1	11	1	2	1	21	1	0	1
2	2	2	0	12	2	0	0	22	2	1	0
3	3	0	1	13	3	1	1	23	3	2	1
4	4	1	0	14	4	2	0	24	4	0	0
5	0	2	1	15	0	0	1	25	0	1	1
6	1	0	0	16	1	1	0	26	1	2	0
7	2	1	1	17	2	2	1	27	2	0	1
8	3	2	0	18	3	0	0	28	3	1	0
9	4	0	1	19	4	1	1	29	4	2	1

$$R_5 = 13 \, mod \, 5 = 3$$
$$R_3 = 13 \, mod \, 3 = 1$$
$$R_2 = 13 \, mod \, 2 = 1$$

$$\Rightarrow [3,1,1]$$

$$R_5 = 8 \, mod \, 5 = 3$$
$$R_3 = 8 \, mod \, 3 = 2$$
$$R_2 = 8 \, mod \, 2 = 0$$

$$\Rightarrow [3,2,0]$$

Vorzeichenbehaftete RNS Darstellung (5,3,2)

N	Residue to base 5	3	2	N	Residue to base 5	3	2	N	Residue to base 5	3	2
0	0	0	0	10	0	1	0	-10	0	2	0
1	1	1	1	11	1	2	1	-9	1	0	1
2	2	2	0	12	2	0	0	-8	2	1	0
3	3	0	1	13	3	1	1	-7	3	2	1
4	4	1	0	14	4	2	0	-6	4	0	0
5	0	2	1	-15	0	0	1	-5	0	1	1
6	1	0	0	-14	1	1	0	-4	1	2	0
7	2	1	1	-13	2	2	1	-3	2	0	1
8	3	2	0	-12	3	0	0	-2	3	1	0
9	4	0	1	-11	4	1	1	-1	4	2	1

$$R_5 = -13 \, mod \, 5 = 2$$
$$R_3 = -13 \, mod \, 3 = 2$$
$$R_2 = -13 \, mod \, 2 = 1$$

$$\Rightarrow [2,2,1]$$

$$R_5 = -8 \, mod \, 5 = 2$$
$$R_3 = -8 \, mod \, 3 = 1$$
$$R_2 = -8 \, mod \, 2 = 0$$

$$\Rightarrow [2,1,0]$$

Chinese Remainder Theorem (CRT)

- Bei einer gegebenen teilerfremden Menge an Moduli ist die Menge an Resten für jede Zahl X < M einzigartig, wobei:

$$M = \prod_{i=0}^{n} m_i$$

Beweis:

$$\begin{array}{lll} (m_{(n-1)}, m_{(n-2)}, \ldots, m_0) & x_i = y_i & X - Y \; Vielfaches \; von \; m_i \\ X = (x_{(n-1)}, x_{(n-2)}, \ldots, x_0) \Rightarrow & x_i = X \bmod m_i \Rightarrow & X - Y \; Vielfaches \; von \; M \\ Y = (y_{(n-1)}, y_{(n-2)}, \ldots, y_0) & y_i = Y \bmod m_i & X \; oder \; Y \geq M \end{array}$$

Warum RNS?

- „Carry-freie“ Addition, Subtraktion und Multiplikation durch parallele Abarbeitung einzelner Ziffern
- Reduzierung der Laufzeit der Operation auf die Laufzeit des langsamsten Modulus

Binär (7-Bit)	RNS (7,5,3,2)	
0101101_2	$[3,0,0,1]$	$[011,000,00,1]$
$+0110000_2$	$+[6,3,0,0]$	$+[110,011,00,0]$
1011101	$[2,3,0,1]$	$[010,011,00,1]$

Schwächen

- Keine einfachen Lösungen für Divisionen, Vergleiche, Überlauferkennung und Vorzeichenerkennung
- Hoher zeitlicher und räumlicher Overhead durch Konverter und modular rechnende Einheiten

RNS (7,5,3,2)

$$\begin{array}{r} [3,0,0,1] \\ +[6,3,0,0] \\ \hline [2,3,0,1] \end{array}$$

Negativ? Überlauf?

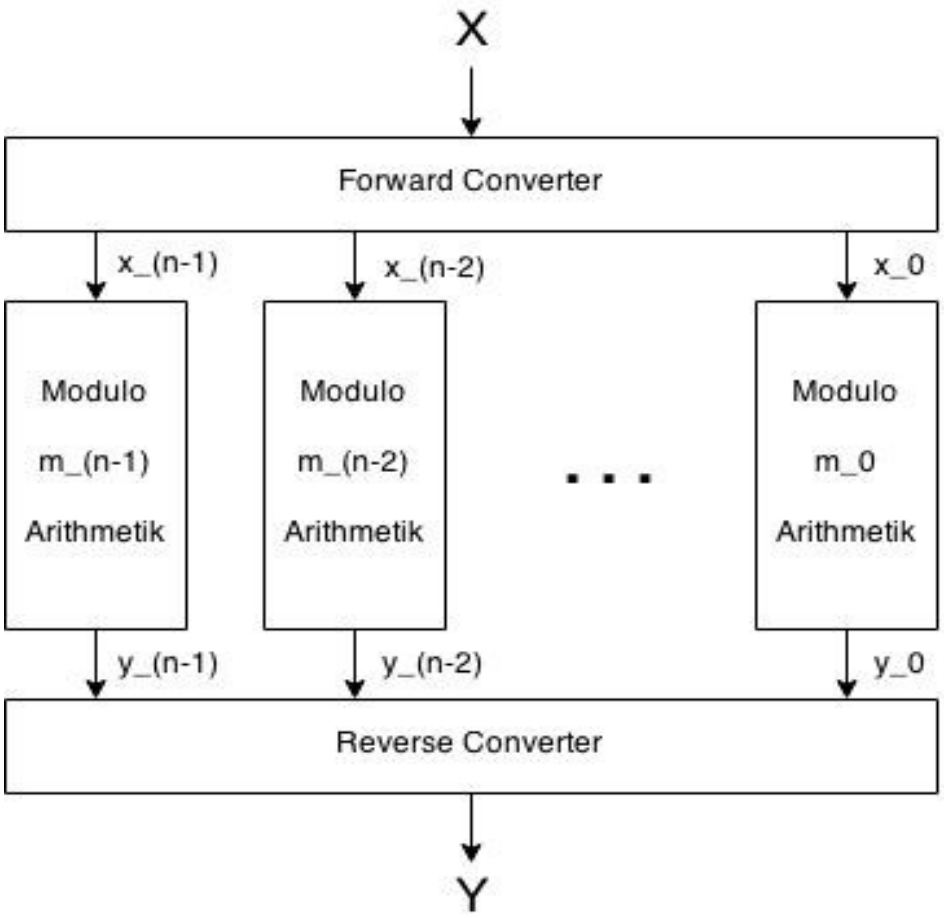
Zahlendarstellung

RNS Systeme

Anwendungsbeispiele

Literatur

02 RNS Systeme



Forward Converter

- Konvertierung in RNS Darstellung
- Lösung durch Zerlegen der Eingabezahlen
- Parallele oder serielle Implementierung

$$|X|_m = \left| \sum_{j=0}^{n-1} |x_j 2^j|_m \right|_m = \left| \sum_{j=0}^{k-1} |2^{jp} B_j|_m \right|_m$$

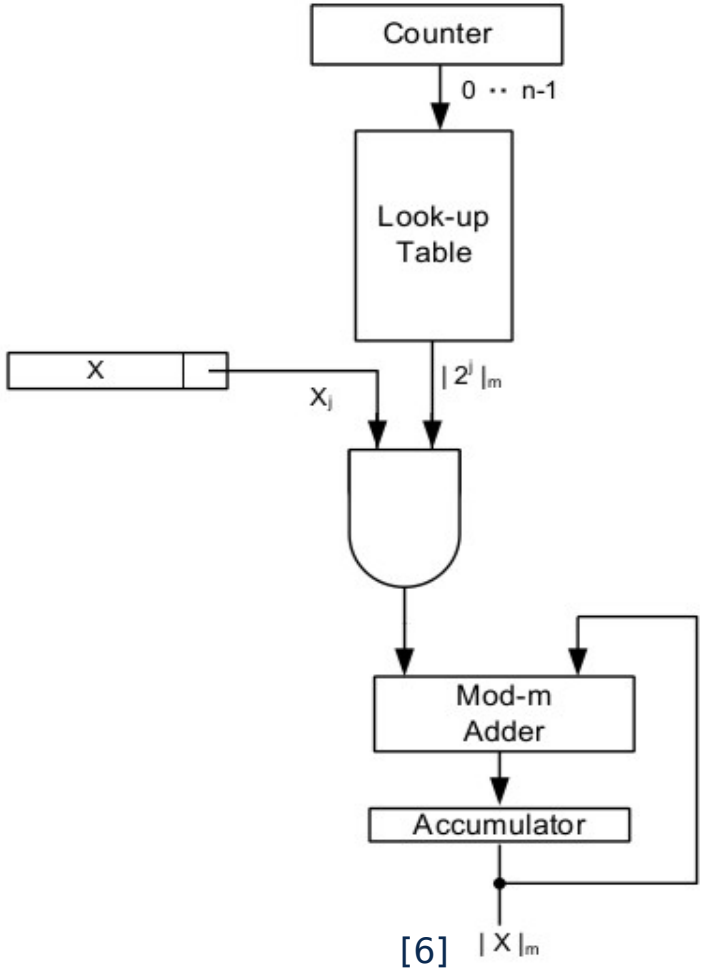
1	1	0	1	0	1
5					0

$$= \left| |1|_m + |4|_m + |16|_m + |32|_m \right|_m$$

1	1	0	1	0	1

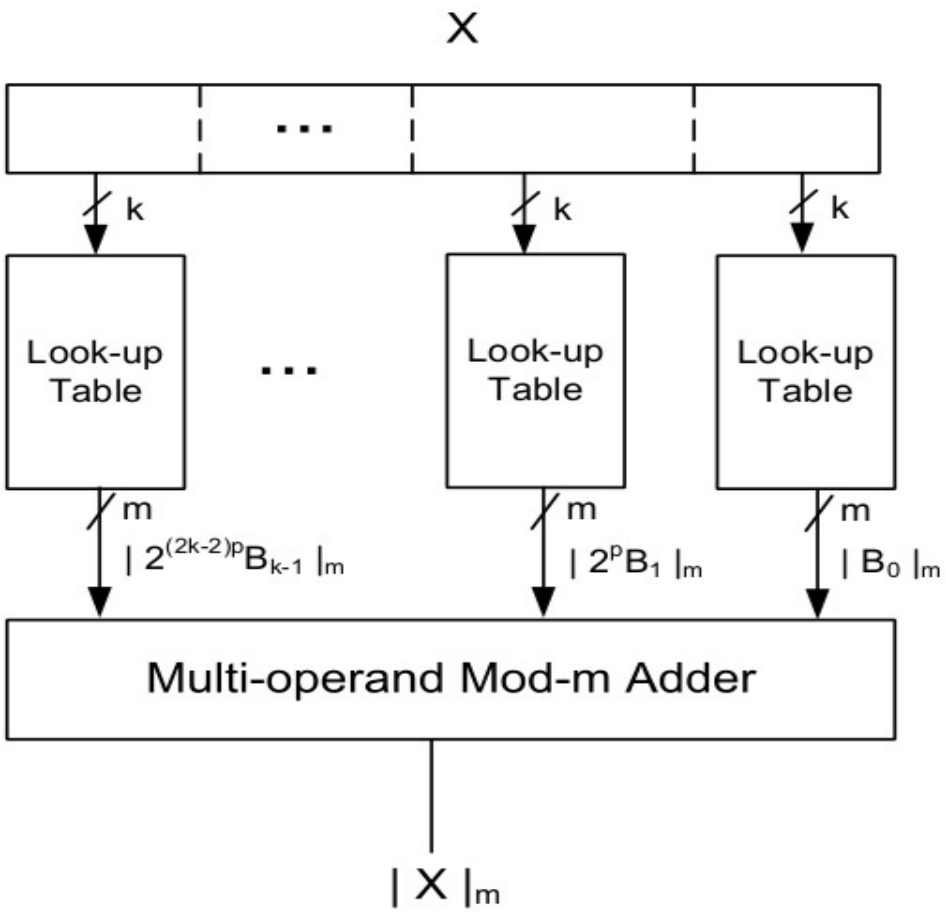
$$= \left| |2^0 * 5|_m + |2^4 * 6|_m \right|_m$$

Forward Converter – seriell & bitweise



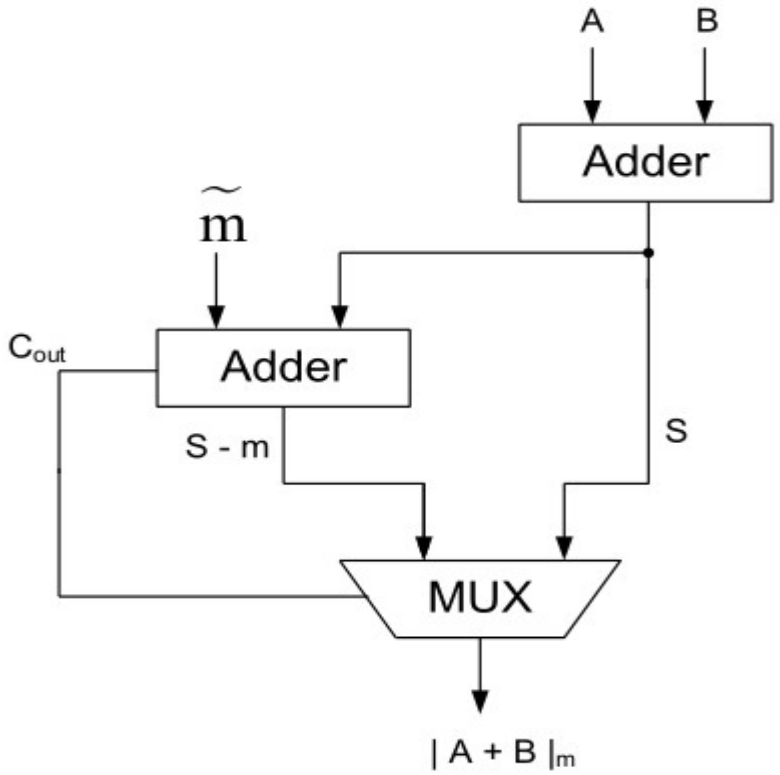
[6]

Forward Converter – parallel & blockweise



[6]

Modulo Addition



[6]

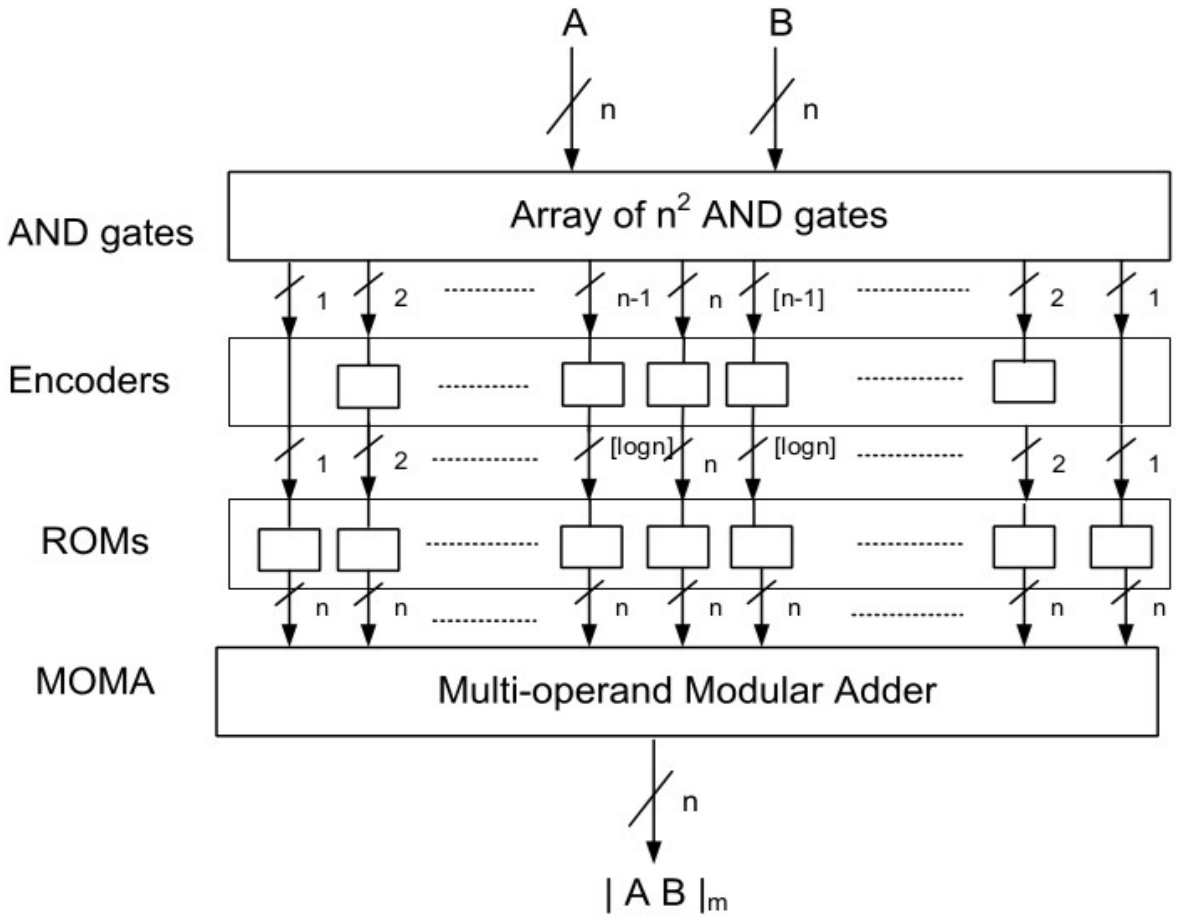
$$|A+B|_m = \begin{cases} A+B & \text{if } A+B<m \\ A+B-m & \text{otherwise} \end{cases}$$

Modulo Multiplikation

- Reduktion des Produkts
- Reduktion der partiellen Produkte
- Look-Up-Tables

$$
\begin{array}{r}
1011 * 1110 \\
\hline
0000 \\
1011 \\
1011 \\
1011 \\
\hline
10011010
\end{array}
$$

Modulo Multiplikation – partielle Reduktion



$$
\begin{array}{r}
1011 * 1110 \\
\hline
0000 \\
1011 \\
1011 \\
1011 \\
\hline
10011010
\end{array}
$$

[6]

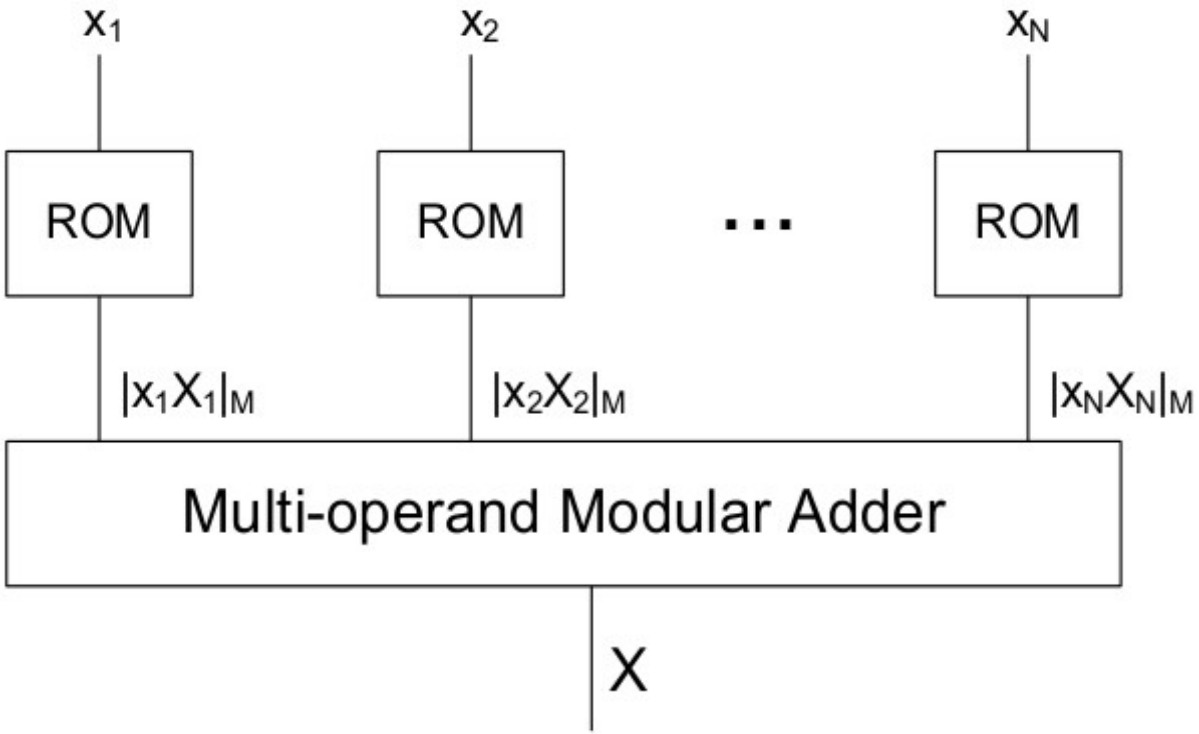
Reverse Converter

- Konvertierung von RNS in konventionelle Darstellung
- Lösung durch Ausnutzen des Chinese-Remainder-Theorem

$$X = \left| \sum_{i=0}^{n-1} |w_i * x_i|_M \right|_M$$

$$\begin{aligned} X &= (x_{(n-1)}, x_{(n-2)}, \ldots, x_0) \\ &= |(x_{(n-1)}, 0, \ldots, 0) + (0, x_{(n-2)}, 0, \ldots, 0) + \ldots + (0, \ldots, 0, x_0)|_M \\ &= ||x_{(n-1)} * (1,0,\ldots,0)|_M + |x_{(n-2)} * (0,1,\ldots,0)|_M + \ldots + |x_0 * (0,\ldots,0,1)|_M|_M \end{aligned}$$

Reverse Converter (CRT)



[6]

Wahl der Moduli Menge

- Teilerfremde Moduli zur Vermeidung von Redundanz
- Effiziente binäre Darstellung der Ziffern

  13: Effizienz 13/16, 17: Effizienz: 17/32
- Ausbalanciert: geringe Differenzen zwischen einzelner

  Moduli verhindern eine Dominanz des größten Modulus
- Vereinfachung der Implementierung

Moduli Mengen

Moduli Set	Moduli n.	RC		Critical Channel	Modular Adders		Modular Multipliers	
		Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n-1, 2^n, 2^n+1\}$ [4]	3	$16n+8$	$\mathbf{31n+13}$	(2^n+1)	$8n+11$	$38n+18$	$16n+12$	$24n^2+7n+15$
$\{2^{n-1}-1, 2^n-1, 2^n\}$ [5]	3	$24n-2$	$54n-45$	(2^n-1)	$\mathbf{8n}$	$\mathbf{21n-7}$	$\mathbf{16n-7}$	$\mathbf{24n^2-35n+12}$
$\{2^n-1, 2^n, 2^{n+1}-1\}$ [6]	3	$\mathbf{8n+30}$	$110n+159$	$(2^{n+1}-1)$	$8n+8$	$21n+7$	$16n+9$	$24n^2-27n+4$

[5]

Moduli Set	Moduli n.	RC		Critical Channel	Modular Adders		Modular Multipliers	
		Delay	*Complexity*		*Delay*	*Complexity*	*Delay*	*Complexity*
$\{2^n-1, 2^n, 2^{2n+1}-1\}$ [7]	3	$40n+20$	$69n+20$	$(2^{2n+1}-1)$	$16n+8$	$\mathbf{38n+25}$	$32n+9$	$48n^2+9n+4$
$\{2^n-1, 2^n+1, 2^{2n}+1\}$ [8]	3	$\mathbf{32n+8}$	$\mathbf{62n+8}$	$(2^{2n}+1)$	$16n+11$	$58n+36$	$32n+12$	$48n^2+62n+15$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ [10-I]	4	$46n+28$	$15n+5+7/2(n^2-3n-4)$	(2^n+1)	$\mathbf{8n+11}$	$45n+25$	$\mathbf{16n+12}$	$\mathbf{32n^2+19n+19}$
$\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [10-II]	4	$48n+62$	$7n^2+102n+108$	$(2^{n+1}+1)$	$8n+19$	$55n+53$	$16n+28$	$32n^2+45n+60$

[5]

Zahlendarstellung

RNS Systeme

Anwendungsbeispiele

Literatur

04 Anwendungsbeispiele

- Anwendungungen dessen vorherrschenden Operationen Additionen, Subtraktionen und Muliplikationen sind
- High-Speed und Low-Power Systeme
- Fehlererkennung und -korrektur

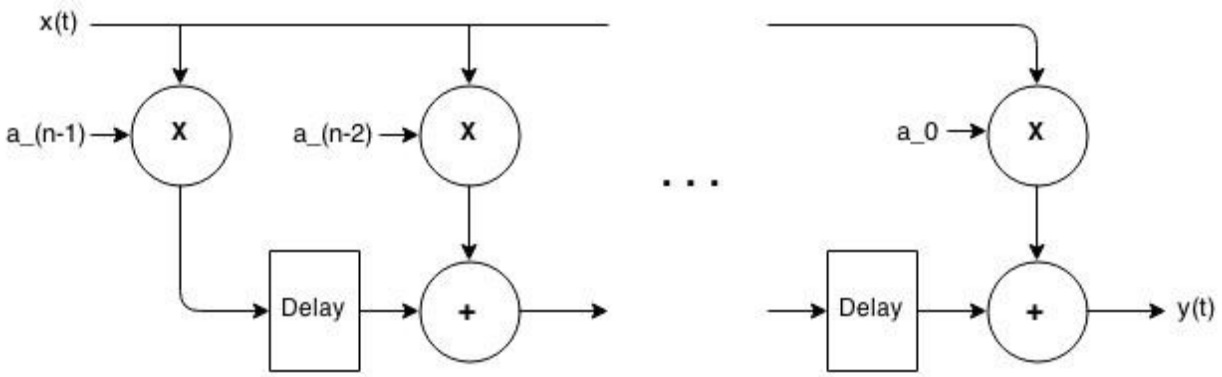
RNS Speedup (32-Bit Ripple-Carry-Adder)

	Measured Delay (ns)	Measured Area ($10^{-3}mm^2$)
32-bit adder	28.67	31.179
Mod 2^{11} adder	8.4	9.949
Mod $2^{11}+1$ adder	10.77	20.206
Mod $2^{11}-1$ adder	9.83	18.12
Reverse Converter	11.5	96.892
Mod $2^{11}+1$ forward conv.	13.8	30.544
Mod $2^{11}-1$ forward conv.	10.4	26.758

[3]

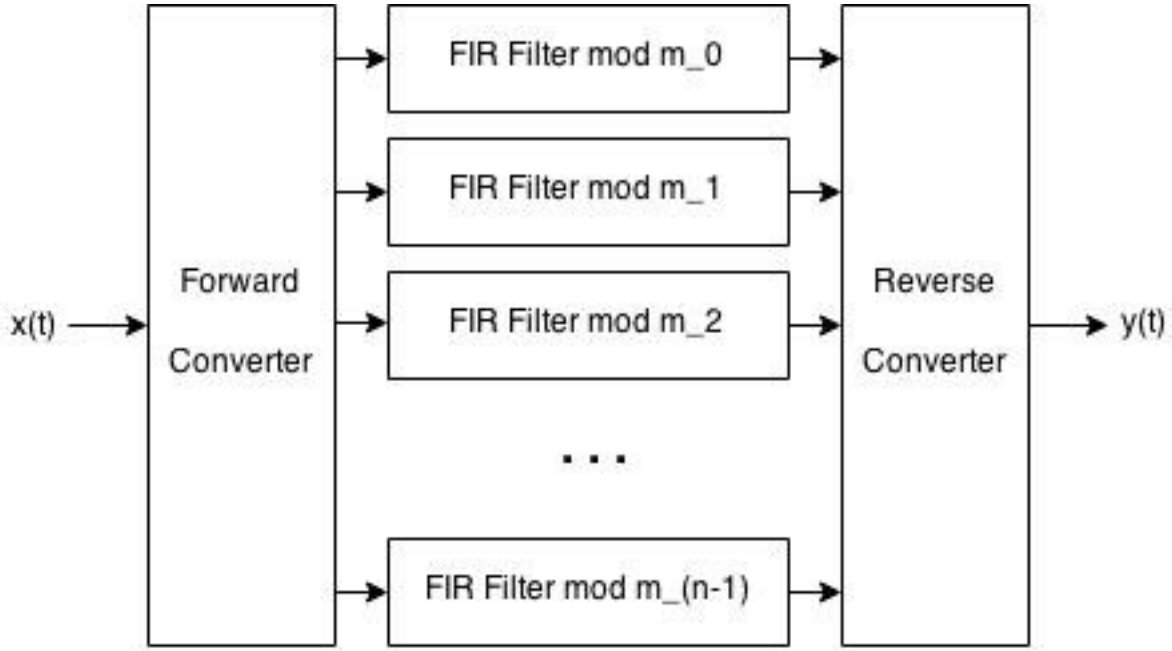
=> Speedup von 28.67/10.77 = 2.66

FIR-Filter



$$y(n) = \sum_{k=0}^{n-1} a_k * x(n-k)$$

RNS FIR-Filter



Low-Power RNS FIR-Filter

Filter	Cycle [ns]	Latency (cycles)	Area (gate equiv.)	Power [mW]
Trad.	5.0	N + 1	230 + 1250N	14.9 + 13.5N
RNS	5.0	N + 3	2910 + 745N	51.0 + 6.9N

[1]

Modulus	Delay [ns]	Area (gate equiv.)	Power [mW]
3	1.7	2016	0.25
5	3.4	4176	0.62
7	3.8	5004	0.84
11	5.0	9450	1.75
17	5.0	9450	1.76
64	3.2	10134	1.66

[1]

- Kleinere Betriebsspannung
- Dual-Voltage Technik
- Speedup = 1
  kein Performance-verlust

One-Hot RNS

- Darstellung einzelner Ziffern im One-Hot-Code

	Wertebereich	Bitbreite (OHC)	#FF (OHC)
Binaercode (8 Bit)	[0,256)	256	256
RNS (5,7,8)	[0,280)	[5,7,8]	5+7+8 = 20

- Einfache und schnelle Implementierung der Arithmetik
- Änderungen reduziert auf Invertieren von maximal zwei Bits

Redundant RNS

- Zusätzliche redundante Moduli ermöglichen die Erkennung und eventuelle Korrektur von fehlerhaften Ziffern
- Unabhängige und ungewichtete Ziffern

Dezimal	RRNS (7,5,3,2)
15	$[3,0,0,1]$
$+\ 3$	$+[6,3,0,1]$
18	$[2,3,1,0] \quad = \quad 58 \geq 30 \quad => Fehler$

Literatur I

[1] G.C. Cardarilli, A. Nannarelli and M. Re: „Reducing power dissipation in FIR filters using the residue number system“, Proceedings of the 43rd IEEE Midwest Symposium on Circuits and System, p. 320-323, Lansing, USA, Aug. 2000

[2] Harvey L. Garner: „The Residue Number System“, IRE Transactions on Electronic Computers, p. 140-147, June 1959

[3] M. Bhardwaj and A. Balaram: „Low power signal processing architectures using residue arithmetic“, Proc. Of the 1998 IEEE International Conference on Acoustics, p. 3017-3020, Seattle, USA, May 1998

[4] Lie-Liang Yang and L. Hanzo: „Redundant residue number system based error correction codes“, Vehicular Technology Conference, p. 1472-1476, Atlantic City, USA, Oct. 2001

Literatur II

[5] D. Younes and P. Steffan: „A comparative study on different moduli sets in residue number system“, International Conference on Computer Systems and Industrial Informatics, p. 1-6, Sharjah, ARE, Dec. 2012

[6] A. Omondi and B. Premkumar: Residue Number Systems, Theory and Implementation. Imperial College Press, 2007