TRADEOFFS BETWEEN RESIDUE NUMBER SYSTEM AND TRADITIONAL FIR FILTERS

Alberto Nannarelli, Marco Re and Gian Carlo Cardarilli

Department of Electrical Engineering
University of Rome "Tor Vergata"
Rome, 00133 ITALY

ABSTRACT

In this work, a study on the implementation of FIR filters in the Residue Number System (RNS) is carried out. For different configurations, RNS filters are compared with filters realized in the traditional two's complement system (TCS) in terms of delay, area and power dissipation. The resulting implementations show that the RNS filters are smaller and consume less power than the corresponding ones in TCS, when the number of taps is larger than sixteen.

1. INTRODUCTION

The new generation of telecommunication equipment often require the use of high order high-speed low-power Finite Impulse Response (FIR) filters which can be effectively implemented by using the Residue Number System (RNS) arithmetic.

The use of the RNS allows the decomposition of a given dynamic range in slices of smaller range on which the computation can be efficiently implemented in parallel [1], [2]. The drawback presented by the RNS is the overhead introduced by the input and output conversions from binary to RNS and vice versa. This overhead (latency and area) can be reduced by using efficient conversion techniques [3], [4].

Recently, the RNS has gained importance in low power design. In [5] and [6] the power dissipation is reduced by taking advantage of the speed-up due to the parallelism of the RNS structure.

In this work, we explore the design work-space of FIR filters with respect to their structure (direct or transposed form), characteristics (constant or variable coefficients), and number of taps. We compare their implementations in the traditional Two's Complement System (TCS) with RNS implementations and consider speed, area and power dissipation tradeoffs. We emphasize on power dissipation aspects for which RNS seems to be better (lower consumption) than TCS.

This work was partially supported by MURST National Project: Codesign Methods for Low Power Integrated Circuits.

The filters have been implemented in a $0.35\mu m$ library of Standard Cells. Delay, area and power dissipation have been determined with Synopsys tools.

Results show that, for a relatively large number or taps, RNS filters offer smaller area and power consumption than corresponding TCS filters.

2. FIR FILTERS

A FIR filter could be realized in either direct or transposed form (Figure 1). As a starting point in our design, we chose input and coefficients size of 10 bits and assumed that a 20-bit dynamic range is enough to have an error-free filter. However, we also extended our investigation to two's complement truncated FIR filters.

A key point in the design of the RNS filter is the choice of moduli. In order to cover the dynamic range of 20 bits, we chose a set of prime numbers, which simplify the multiplications, as explained later, and a power of two (modular operations are trivial) to boost the range. Based on results of simulations, we chose for our RNS filters the following set of moduli

$$\{3, 5, 7, 11, 17, 64\}$$

which gave us the best delay/area/power tradeoffs.

3. FIR FILTERS IN TRANSPOSED FORM

Because FIR filters in transposed form are modular with respect to the number of taps (i.e. adding extra taps does not alter the filter architecture), we set as the main design constraint the maximum delay (i.e. the critical path) which is the propagation delay between two registers in adjacent taps (see Figure 1.a).

3.1. Transposed FIR Filters in Two's Compl. System

The composing blocks of a FIR filter are multipliers, adders and registers. In the following, we describe the architectures chosen for implementing these composing blocks.

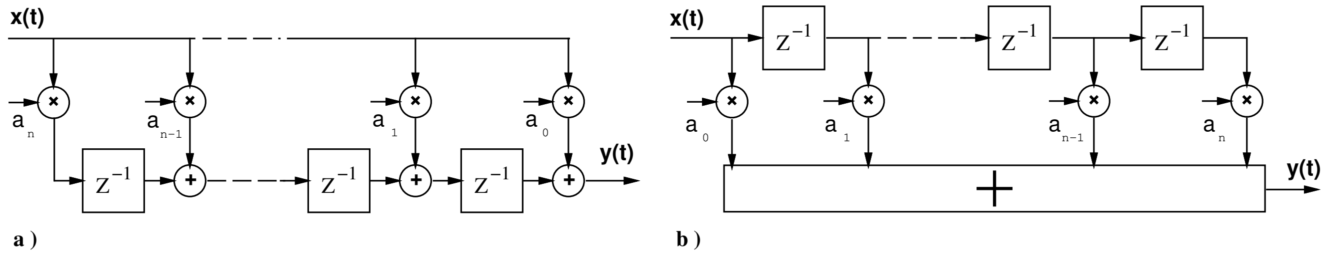

Figure 1: FIR filters in transposed (a) and direct (b) form.

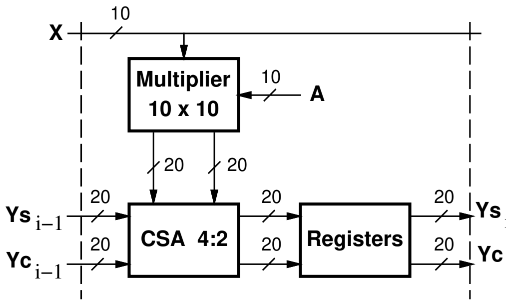

Figure 2: Tap structure for the direct TCS FIR filter.

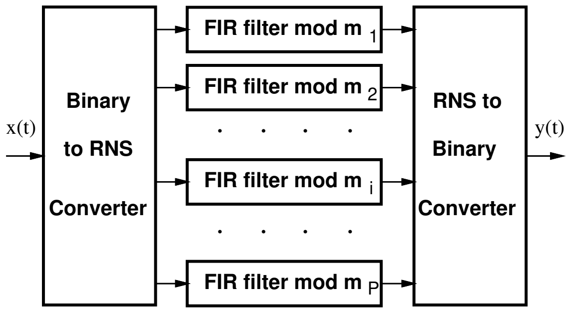

Figure 3: RNS FIR filter.

For the implementation of multipliers with the traditional binary system (TCS), we chose to keep the product in carry-save (CS) format in order to speed-up the operation, and delayed the assimilation of the CS representation to the last stage of the filter. For the FIR filter in transposed form (Figure 1.a), in each tap we need to add the CS representation of the product to the value stored in the register (previous tap). Again, to avoid the propagation of the carry, we can store the CS representation. For this reason, we need to implement the addition with an array of 4:2 carry-save adders (CSA), as shown in Figure 2.

The CS representation is finally converted into the two's complement representation by a carry-propagate adder (realized with a carry-look-ahead scheme) in the last stage of the filter.

Figure 2 shows the implementation of the tap of a filter with programmable coefficients. For filters with constant coefficients (not programmable) the implementation scheme is the same with the only difference in the multiplier which is simplified.

For both, variable and constant coefficients, implementations the critical path is:

$$t_{MULT} + t_{CSA4:2} + t_{REG} \ .$$

The filter latency is given by the number of taps N plus one extra cycle needed to compute y from its CS representation ($y = y_s + y_c$). The expression for the area as a function of the number of taps is:

$$A_{TAP} \cdot N + A_{CPA}$$

where A_{CPA} is the area of the final carry-propagate adder. For the power dissipation the expression is similar to that of the area:

$$P_{TAP} \cdot N + P_{CPA}$$

with the limitation that the term P_{TAP} strongly depends on the switching activity and consequently on the value of the coefficients.

The results are summarized in Table 1 with their actual values.

3.2. Transposed FIR Filters in Residue Number System

As already mentioned, a RNS filter can be decomposed, as shown in Figure 3, into P filters of smaller dynamic range (P is the number of moduli) working in parallel.

In each tap, a modular multiplier is needed and because of the complexity of modular multiplication, we used the isomorphism technique to implement the product of residues for prime moduli (except 3) [7]. By using isomorphisms, the product of the two residues is transformed into the sum of their indices which are obtained by an isomorphic transformation (see [8] for implementation detail).

Table 1: FIR FILTERS IN TRANSPOSED FORM: SUMMARY OF RESULTS

Filter with	TCS Cycle $[ns]$	TCS Latency (cycles)	TCS Area (NAND2 equiv.)	TCS Power (*) $[mW]$	RNS Cycle $[ns]$	RNS Latency (cycles)	RNS Area (NAND2 equiv.)	RNS Power (*) $[mW]$
var. coeff.	5.0	$N+1$	$1250N+230$	$13.5N+14.9$	5.0	$N+3$	$745N+2910$	$6.9N+51.0$
const. coeff.	4.9	$N+1$	$844N+230$	$8.6N+14.9$	4.3	$N+3$	$500N+2910$	$4.2N+51.0$
	TCS truncated							
var. coeff.	5.0	$N+1$	$860N+120$	$9.2N+7.1$				
const. coeff.	4.7	$N+1$	$565N+120$	$5.8N+7.1$				

(*) Power at 100MHz assuming random input activity.

In case of filters with constant coefficients, the multiplication for a constant can be easily performed by a look-up table implemented with synthesized logic.

The addition is also a modular operation and a correction step is needed if the result of the binary addition exceeds the modulo (see [8] for implementation detail).

As already mentioned, the RNS FIR filter is completed by an input and an output conversion block.

The critical path for RNS filters in transposed form is:

$$t_{modMULT} + t_{modADD} + t_{REG}$$

and the latency is $N + 3$ cycles, because 1 cycle for input binary-RNS conversion and 2 cycles for output RNS-binary conversion are needed. The expressions for area and power are similar to those of the corresponding TCS filters with the substitution of the converters in place of the CPA. The actual values are reported in Table 1. Moreover, in Table 2 the values per tap for different moduli paths are shown.

Table 2: DELAY, AREA AND POWER DISSIPATION PER TAP IN RNS FIR (TRANSPOSED FORM).

mod	variable coeff. Delay $[ns]$	variable coeff. Area	variable coeff. Power $[mW]$	constant coeff. Delay $[ns]$	constant coeff. Area	constant coeff. Power $[mW]$
3	1.7	37	0.25	1.1	24	0.14
5	3.4	77	0.62	2.8	65	0.41
7	3.8	93	0.84	1.9	50	0.41
11	5.0	175	1.75	3.2	98	0.96
17	5.0	175	1.76	4.3	148	1.24
64	3.2	188	1.66	2.5	116	0.98

Area as number of NAND2.
Power at 100MHz assuming random input activity.

3.3. Transposed Truncated FIR Filters

Sometimes accuracy is traded with performance by implementing filters with truncated dynamic range. Truncation or rounding schemes are very tricky to be implemented in RNS, and for this reason we limited our investigation to TCS filters. We have tried different truncation schemes, but we report here only the case in which the results of multiplications are truncated after 10 bits. Table 1 shows the values for the truncated implementation of the filters with variable and constant coefficients.

3.4. Transposed FIR Filters: Summary

The table shows that the RNS filter has a higher latency, due to the conversions, but it can be clocked at the same rate of the TCS filters and sustain the same throughput. Moreover for error-free filters (first two rows) the RNS filter is better than the corresponding TCS for more than 8 taps. Even if we compare the truncated TCS filters with error-free RNS filters, the RNS filters show smaller area and power dissipation when the number of taps is larger than 40.

Furthermore, in RNS filters by lowering the supply voltage for moduli not in the critical path (moduli 3, 5, 7, and 64, according to Table 2) the power dissipation can be further reduced without affecting the overall performance [8].

4. FIR FILTERS IN DIRECT FORM

For the FIR filter in direct form (Figure 1.b), it is convenient to implement the addition with a Wallace's tree to speed-up the operations. However, the use of the tree makes the design of the filter not modular in the number of taps and the delay of the structure changes with its size.

4.1. Direct FIR Filters in Two's Complement System

By opting for the CS representation of the products, for a N-tap filter we have to add $2 \times N$ addends. The resulting expressions for the critical path is

$$t_{mult} + t_{CSA4:2} \cdot k + t_{set-up} \qquad \text{with } k = \lceil \log_2 N \rceil$$

assuming the tree realized with 4:2 CSAs and a register (REG) placed at the output of the tree before the final addition (CS to two's complement). The expression for the area is

$$A_{TAP}N + A_{CSA4:2}(2^{\lceil log_2 N \rceil} - 1) + A_{REG} + A_{CPA}$$

Table 3: FIR FILTERS IN DIRECT FORM: SUMMARY OF RESULTS

Filter with	TCS Cr. Path $[ns]$	TCS Area (NAND2 equiv.)	TCS Power (*) $[mW]$	RNS Cr. Path $[ns]$	RNS Area (NAND2 equiv.)	RNS Power (*) $[mW]$
var. coeff.	$3.5 + 1.7k$	$1069N + 71$	81.4	$5.1 + 1.7(k-1)$	$691N + 3045$	80.6
const. coeff.	$3.5 + 1.7k$	$649N + 71$	68.8	$3.5 + 1.7(k-1)$	$462N + 3045$	73.5
	TCS truncated					
var. coeff.	$3.5 + 1.7k$	$766N + 56$	70.1			
const. coeff.	$3.5 + 1.7k$	$493N + 56$	51.7			

(*) Power at 100MHz assuming random input activity for 8-tap filters

Assuming that N is close to a power of 2, the area is linear with respect to N:

$$(A_{TAP} + A_{CSA4:2})N - A_{CSA4:2} + A_{REG} + A_{CPA}.$$

For the power, we obtain a similar expression as a function of N. However, because the switching activity in the tree is strongly dependent on its size (due to glitches), the values obtained by the formula are very coarse. For this reason, we only show the power dissipation for a 8-tap direct FIR filter in the following.

The results are summarized in Table 3 with their actual values.

4.2. Direct FIR Filters in Residue Number System

In RNS direct FIR filters a N-input Wallace's tree, followed by a block to compute the modulo (described in [4]), is needed. The expressions for critical path and area are the following:

$$t_{mult} + t_{CSA4:2} \cdot (k-1) + t_{mod} + t_{set-up}$$

$$A_{TAP}N + A_{CSA4:2} \cdot (N/2 - 1) + A_{mod} + A_{REG} + A_{conv}$$

with the same assumptions as for the TCS filter. The tree in RNS is one level shallower, however, the extra modulo operation (*mod*) impacts critical path and area (and power as well).

4.3. Direct FIR Filters: Summary

Table 3 summarizes the results for the different implementations. The critical path depends on the number of levels in the tree (and consequently, on the number of taps). We can introduce pipeline latches to speed-up operations, but we increase area and power. For the error-free with variable coefficients filters (first row in Table 3) the critical path is almost the same, but the RNS is smaller and consumes less power for $N > 8$. For the error-free with constant coefficients filters (second row), the RNS is faster (about 1.7 ns less), and its area start to be smaller for more than 16-tap filters.

5. CONCLUSIONS

The results presented here, show that for high-order FIR filters the RNS implementations gives the same throughput as filters implemented in the traditional two's complement system (TCS), but have smaller area and power consumption. Even truncated TCS FIR filters are not performing better than error-free RNS for filters with a large number of taps. Moreover, our investigation showed that FIR in transposed form gives a better performance at expenses of a larger area and power dissipation.

6. REFERENCES

[1] M.A. Sodestrand, W.K. Jenkins, G. A. Jullien, and F. J. Taylor, *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, New York: IEEE Press, 1986.

[2] M.A. Soderstrand and K.Al Marayati, "VLSI implementation of very high-order FIR filters," *IEEE International Symposium on Circuits and Systems (ISCAS'95)*, vol. Vol. 2, pp. 1436–1439, 1995.

[3] G. Cardarilli, M. Re, and R. Lojacono, "A residue to binary conversion algorithm for signed numbers," *European Conference on Circuit Theory and Design (ECCTD'97)*, vol. Vol. 3, pp. 1456–1459, 1997.

[4] G. Cardarilli, M. Re, R. Lojacono, and G. Ferri, "A new efficient architecture for binary to RNS conversion," *Proc. of European Conference on Circuit Theory and Design (ECCTD'99)*, vol. Vol. 2, pp. 1151–1154, 1999.

[5] M. Bhardwaj and A. Balaram, "Low power signal processing architectures using residue arithmetic," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ASSP'98)*, vol. Vol. 5, pp. 3017–3020, 1998.

[6] W.L. Freking and K.K. Parhi, "Low-power digital filters using residue arithmetic," *Thirty-First Asilomar Conference on Signals, Systems and Computers*, vol. Vol. 1, pp. 739–743, 1998.

[7] I.M. Vinogradov, *An Introduction to the Theory of Numbers*, New York: Pergamon Press, 1955.

[8] G. Cardarilli, A. Nannarelli, and M. Re, "Reducing power dissipation in FIR filters using the residue number system," *to appear in Proc. of the 43rd IEEE Midwest Symposium on Circuits and Systems*, Aug. 2000, Available at http://dspvlsi.uniroma2.it/pubs/mwscas00/.