

# A High Speed RNS Processor with TSC Code Error Detection

Jimson Mathew and Damu Radhakrishnan

Division of Computing Systems, SAS, Nanyang Technological University  
Nanyang Ave., Singapore 639798, email: asdrkrishnan@ntu.edu.sg

**Abstract** Several Digital Signal Processing (DSP) structures based on Residue number system (RNS) have been proposed in the technical literature. Most of them employ a look up table approach, which is space consuming and lacks flexibility and programmability. In this regard, we propose a memoryless high-speed error detecting processor architecture using 1-out-of-n code for representing the residue numbers. With the use of 1-out-of-n code, error detection is achieved without any redundant moduli. The proposed design exhibits VLSI efficient layout, operand independent delay and low power consumption.

## 1. Introduction

Even though RNS provides very fast and reliable arithmetic due to its inherent parallel processing and error correction capabilities, its use is severely limited to a few applications due to the complexities involved in its conversion circuitry. An RNS is defined by a set of relatively prime integers  $m_1, m_2, \dots, m_r$  called the moduli of the number system. Such a system provides unique representation of numbers from 0 to  $M-1$ , where  $M = \prod_{i=1}^r m_i$ . Each integer  $X$  is represented by an  $r$  tuple  $(x_1, x_2, \dots, x_r)$ , where each residue  $x_i = X \bmod m_i$ , defined as the least remainder when  $X$  is divided by the moduli  $m_i$ . In RNS, arithmetic operations on large integers are done by converting them into smaller residues and performing the operations independently and all in parallel, thereby speeding up the whole operation. In present systems an analog signal is first converted into binary and then a binary-to-residue converter is used to generate the residues. The residue results produced at the end of processing are finally reconverted back to binary. These conversions from binary-to-residue (forward conversion) and from residue-to-binary (reverse conversion) are complex operations for a general moduli set. Two reverse conversion methods are generally used;

one is based on Chinese Remainder Theorem (CRT) and is given by:

$$X = \left| \sum_{i=1}^r a_i x_i \hat{M}_i \right|_M, \quad \text{where} \quad \hat{M}_i = \frac{M}{m_i} \quad \text{and}$$

$a_i = \left\lfloor \frac{1}{\hat{M}_i} \right\rfloor_{m_i}$ . The second uses Mixed Radix Conversion

(MRC) and is given by the following equation:

$$X = A_r \prod_{i=1}^{r-1} m_i + \dots + A_3 m_2 m_1 + A_2 m_1 + A_1,$$

where  $A_i \in [0, m_i)$  and

$$A_i = \left\lfloor \frac{1}{m_1 m_2 \dots m_{i-1}} (x_i - (A_{i-1} m_{i-2} \dots m_1 + \dots + A_1)) \right\rfloor_{m_i}.$$

In this paper an MRC based architecture is proposed.

## 2. High Speed Processor

The basic architecture of the processor is shown in Figure 1. It consists of a binary-to-residue converter, the processor modules and the reverse converter. Conventional designs use ROM and modulo adders for the implementation of the forward converter. But our design uses the one Hot Coded Residue (OHR) cells defined in [2] for the whole design. The look up operation

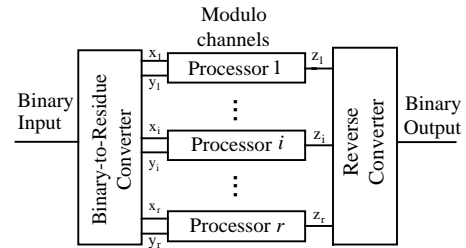


Figure 1. RNS Based Processor

is implemented simply by hardwiring the OHR cells. A binary to OHR converter is shown in Figure 2. The input binary number is divided into  $l$  bit fields that are directly

fed to the OHR cell through a decoder, where  $l = \lceil \log_2 m_i \rceil$ . The OHR cell delay is only a single gate/switching delay. Hence the binary-to-OHR conversion time is only a few gate delays. The modular adders and multipliers are implemented using barrel shifters. Modular multiplier design is based on index calculus approach [6], where modular multiplications are done using modular addition. Therefore, the processing modules take a single switching time, irrespective of its operation such as modulo multiplication or addition. For instance, a complete 32 coefficient Multiply Accumulate (MAC) operation can be completed in 32 clock cycles. Since each state in this architecture is represented by a separate active line, the coefficient ROM in the traditional filter design can be replaced by mere decoders which saves hardware and takes minimum delay. Moreover, all the modules with different bits/modulus can work synchronously because modulo multiply/add times are independent of their word length. Therefore, the moduli set selected need not be a balanced one.

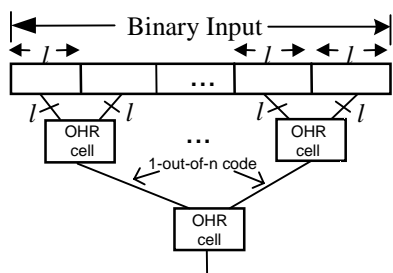


Figure 2. Binary-to-OHR Converter

The final conversion from residue to weighted number can be done using a residue-to-mixed radix converter. The error detection capability is built into the design due to its 1-out-of-n code nature, rather than by the addition of redundant moduli as in conventional RNS architectures. The mixed radix converter used in our approach is very similar to the one proposed in [3]. But [3] uses a set of look up tables to implement the mixed radix conversion ( $z = |(x - y)k|_m$ , where  $x$  and  $y$  are the input to the look up table and  $z$  is the output), whereas they are replaced by means of OHR cells in our design. All the modules work on 1-out-of-n code and hence error checking can be postponed to the last stage. Finally the 1-out-of-n code outputs from the final blocks are encoded to generate the MRC coefficients. The encoder is nothing but a set of OR gates. The cells that output the MRC coefficients are checked for errors using a TSC checker. That is, for an  $r$  moduli system the  $r$  MRC coefficient channels are checked for errors using self checking checkers. This is done by first converting the 1-out-of-n code from each channel into an m-out-of-2m code, and subsequently to a

1-out-of-2 code. The conversion from 1-out-of-n code to m-out-of-2m code uses a set of simple OR gates. The m-out-of-2m code to 1-out-of-2 code conversion uses majority detection circuits. These 1-out-of-2 codes from the individual channels are finally combined to form a final 1-out-of-2 code, which in turn displays the error.

### 3. Conclusion

The proposed architecture replaces all arithmetic modules in conventional designs by simple switching modules. The hardware overhead is comparable to that of the existing ones. Furthermore, an error detection feature is incorporated into the design without the use of any redundant moduli. The overhead for error detection is about 5% of the overall hardware of the system. Apart from the error detection capability, the proposed architecture exhibits regular layout, operand independent delay and minimum power-delay product. The processor is totally memoryless and is hardware efficient.

### References

- [1] M.A. Soderstrand, 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] W.A. Chren, "One- Hot Residue Coding for Low Delay-Power Product CMOS Design," *IEEE Trans. Circuits Syst.-II: Analogue and Digital Signal Processing*, vol. 45, no. 3, Mar. 1998, pp. 303-313.
- [3] D.F. Miller, W.S. McCormick, "An Arithmetic Free Parallel Mixed-Radix Conversion Algorithm," *IEEE Trans. Circuits Syst.-II: Analogue and Digital Signal Processing*, vol. 45, no. 1, Jan. 1998, pp. 158-162.
- [4] A.M Paschalis, D. Nikolos, and C. Halatsis, "Efficient Modular Design of TSC Checkers for M-out-of-2M Codes," *IEEE Trans. Comput.*, vol.37, no. 3, Mar. 1988, pp. 301-309.
- [5] D.A. Anderson, G. Metze, "Design of Totally Self-Checking Check Circuits for m-Out-of-n Codes," *IEEE Trans. Comput.*, vol. 22, no. 3, Mar., 1973, pp. 263-268.
- [6] D. Radhakrishnan and Y. Yuan, "Novel Approaches to Design of VLSI RNS Multipliers", *IEEE Trans. Circuit. Syst.-II: Analogue and Digital Signal Processing*, vol. 39, no. 1, Jan.1992, pp. 52-57.