Design of LUT based RNS Reverse Converters

Wei Lam Kong and Nicholas Chan Hua Vun
School of Computer Engineering, Nanyang Technological University, Singapore

ABSTRACT

This paper presents the strategies to implement Residue Number System reverse converter based on the Look-Up Table (LUT) approach that is applicable for general moduli set. The approach makes use of partitioning to divide the LUT entries into multiple small LUTs in parallel, where their outputs can be further selected to obtain the equivalent binary number. Pipelining architecture is also incorporated to improve the operation speed. These techniques are hence suitable for general moduli set with large moduli value. Implementation results based on FPGA further demonstrate the feasibility and effectiveness of the proposed approach.

INTRODUCTION

Digital signal processing is now an integrated parts of many sophisticated consumer electronics products such as the smart phones and tablets. As such, it is important to have highly efficient techniques to execute these processes. One such approach is to make use of the Residue Number System (RNS) which allows the execution to be done in a parallel manner with small dynamic range data [1][2]. [3] further illustrates how RNS can also be applied directly to the hardware implementation of high speed analog to digital converter (ADC) to greatly simplify its internal circuit complexity compared to normal approach, while its RNS format output can also be processed directly and efficiently in the RNS domain [4]. In all these cases, the RNS outputs need to be eventually converted back to the equivalent conventional binary number format. This reverse conversion process can theoretically be performed by applying the Chinese Remainder Theorem (CRT) but in practice, is hard to be implemented efficiently in hardware except for specific combination such as the popular $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$ moduli set [5][6]. In this paper, we examine the implementation of the RNS reversion conversion based on the LUT approach which would then allow non-restrictive choice of moduli set, and hence would be suitable for more general classes of RNS based applications.

While a LUT approach does not require computational logic like those based on CRT, the cost, complexity and power consumption of LUT based implementations will typically increase drastically with its table size, and hence usually would not be suitable for moduli sets using large moduli values. The main focus of this paper is hence to propose efficient strategies to implement LUT based RNS reverse converter that is practical and extendable for any combination of moduli set. Performance results based on FPGA implementations are also given to demonstrate the feasibility and benefits of the proposed techniques.

PROPOSED ARCHITECTURES

The first technique is to partition the required LUT entries into smaller independent groups. The strategy is to first choose one modulus from the moduli set as a selector. This selector will hence determine the partition factor of the LUTs. The selector is preferably the biggest modulus number in the moduli set in order to maximize the reduction in terms of total combinations and input bits per partition block. In this manner, each partition block would then contain the LUT entries correspond to the same residue value of the selector. During operation, the residue value of the selector will then be used to select the appropriate partition block that provides the equivalent binary number output. Using the {7,8,9} moduli set as an example, Table 1 shows 504 unique input combinations that represents the equivalent binary number ranging from 0 to 503.

Table.1 Moduli set {7.8.9} entries

Inputs			**Output**
Residue 1 (Mod 7)	**Residue 2 (Mod 8)**	**Residue 3 (Mod 9)**	
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
0	7	7	7
1	0	8	8
2	1	0	9
3	2	1	10
4	3	2	11
5	4	3	12
. .	. .	. .	. .
5	6	7	502
6	7	8	503

Using the largest modulus value 9 as the selector, the entries can hence be divided into nine groups, each group contains entries that correspond to the same value shown under the Residue 3 column. The individual bit value of Residue 3 will then be used as control signals for a selector unit, which is made up of multiple multiplexers to select an output from one of the partition blocks as the final binary result. Figure 1 shows the block diagram illustrating the concept of the proposed partition architecture design.

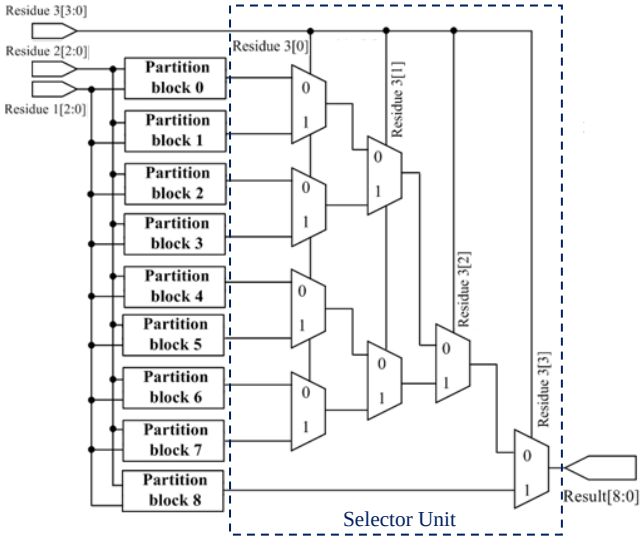

Fig.1 Moduli set {7.8.9} partition architecture

By using this partition architecture, the amount of resources required is greatly reduced compared to the non-partition LUT approach when implemented in FPGA. However, due to the inclusion of the selector unit, the overall speed of the partitioned architecture would theoretically be slower compared to the non-partitioned version. Hence the main weakness of the partition architecture would be the increasing complexity of selector unit that affects the speed performance when the number of partition blocks increases (i.e. with larger modulus value).

As the selector unit forms the critical path of the design and determines the speed performance of the proposed architecture, pipelining can be further incorporated into the system. The strategy is to insert pipelines into the design to eliminate long critical paths with trade off in initial latency. This would significantly improve the speed and throughput of the design, albeit at the expense of bigger area. Figure 2 illustrates the pipelined architecture applying to the selector unit for moduli set {7,8,9} reverse converter of Figure 1.

SYNTHESIS RESULTS

The proposed architectures are synthesized for a Xilinx Spartan-6 LX45 FPGA. Table 2 presents the comparisons of the various performance parameters between the standard LUT implementation, the proposed partition LUT architecture and pipelined partition LUT architecture.

Table.2 Performance Comparisons

Resource Usage	Without Partition	With Partition	Pipelined Partition
# BELS	**186**	**81**	**81**
# INV	1	1	1
# LUT4	6	4	4
# LUT5	11	15	15
# LUT6	132	61	61
# MUXF7	36	-	-
# FlipFlops/Latches	**52**	**19**	**133**
# FDR	52	19	133
# Clock and I/O Buffers	**21**	**21**	**21**
# BUFGP	1	1	1
# IBUF	11	11	11
# OBUF	9	9	9
Max Freq (MHz)	**220.5**	**216.8**	**370.8**

As shown in Table 2, the partition architecture enables significant improvement in terms of area saving mainly due to the reduced usage of LUTs. However, there is a slight reduction in terms of speed due to the delay introduced by the selector unit. By including the pipelined architecture, the speed performance can be greatly enhanced, but at the expense of higher number of logic gates.

SUMMARY

Applying partitioning and pipelining techniques to LUT implementations enable the RNS reverse converter design to be more scalable and extendable to general moduli set of unrestricted values. These techniques can potentially be further refined by also sub-dividing the partition blocks and arranged them in a pipeline-like architecture, which would be particular suitable for moduli set with large modulus value.

REFERENCES

[1] N.S.Szabo, R.I.Tanaka, " Residue Arithmetic and its Applications to Computer Technology", McGraw-Hill, New York, 1967.

[2] A.Omondy, B.Premkumar, "Residue Number System theory and implementation", Chapter 8, Imperial College Press, 2007.

[3] Vun, C.H.; Premkumar, A.B., "RNS Encoding Based Folding ADC," *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on* , vol., no., pp.814,817, 20-23 May 2012.

[4] C.H.Vun, A.PremKumar, "A New RNS based DA Approach For Inner Product Computation", accepted for publication in *Circuits and Systems I, IEEE Transactions on*, 2013.

[5] S. Andraos and H. Ahmed, "A new efficient memoryless residual to binary converter", *IEEE Trans. Circuits & Systems*, vol. 35, pp. 1441-1444, Nov. 1988.

[6] Zhongde Wang G.A Jullien and W.C Miller, "An improved Residue to binary converter" *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol.47, no.9, pp.1437,1440, Sep 2000.

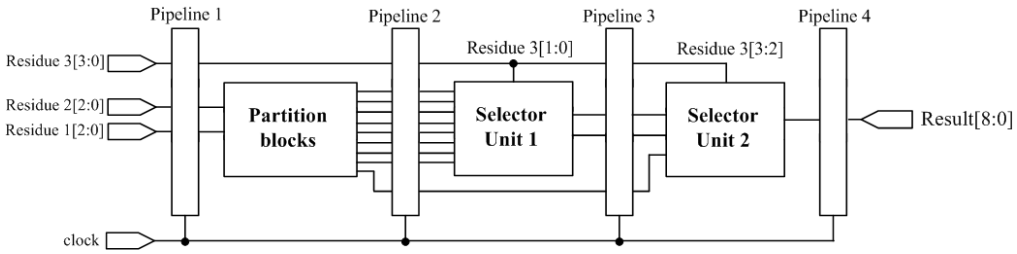

Fig.2 Pipelined architecture