Implementation of 16 bit Hybrid Modulo 2^n-1 Adder

Uday J
Asst. Professor
SJEC, M'lore

Sudarshan Patwardhan
Scholar
SJEC,M'lore

Shishir Rai
Scholar
SJEC, M'lore

Sushmitha B
Scholar
SJEC, M'lore

Vaishnavi Devi
Scholar
SJEC, M'lore

***Abstract*:- There is a growing demand for fast and efficient adders. There are number of architectures proposed to cop up with the growing demands. A variety of prefix adders are discussed in literature to achieve area and performance optimization. Modular addition plays an important role in data encryption standard, RNS application. In this paper a newVLSI circuit architectures for addition modulo 2^n-1 are presented, which allows the implementation of highly efficient combinational circuits for modular arithmetic. To realize the architecture we use a new prefix operator known as Star operator and Sparse carry computation unit.**

The architectures are implemented on Xilinx Spartan III field-programmable gate array (FPGA) using ISE 8.1. The results indicate that, on an average, the implemented architectures are better in terms of slices, LUT's and memory utilization by comparing all formal proposals.

General Terms:- Slices, LUTs, delay, architecture.

Keywords:- Hybrid Parallel Prefix adders, Star, Sparse.

1. INTRODUCTION

Modulo 2^n -1 adders are used in various applications, ranging from Residue Number System (RNS) applications [1] fault-tolerant computer systems [2], and cryptography. In RNS logic, each operand is represented by its moduli with respect to a set of numbers comprising the base. Each of the numbers of the base must not have any common factor with any of the other numbers of the base. Most often, the base consists of three numbers: 2^n -1, 2^n , 2^n +1.
Many solutions have been presented for fast modulo 2^n -1 addition. In [5] modulo adders are proposed that use a parallel prefix carry computation unit along with an extra prefix level that handles the end-around-carry. In [6] it was shown that the recirculation of the end-around-carry can be performed within the existing prefix levels. Therefore, the need of the extra prefix level is cancelled and parallel-prefix modulo 2^n -1 adders are derived that can perform carry computation in $\log_2 n$ levels. However, the routing requirements are increased. In [7] select prefix modulo 2^n -1 adders have been proposed, that aim at reducing the area complexity of the parallel-prefix structures but suffer from significant delay penalties.
In this paper a new architecture of modulo 2^n -1 adder is implemented. This architecture uses new operator called star operator.

2. PARALLEL PREFIX ADDER

***Prefix*:** The system output is dependent on initial input. *Parallel*: Execution of an operation in parallel.
This is done by segmentation into smaller pieces that are computed in parallel. We know that computation of the carry input signal for each bit addition is the most critical and time consuming. The carry look ahead adders (CLA), gives an idea how to produce the carry input signals for individual bit addition. This is achieved by generating two signals, generate (g_i) and propagate (p_i) using the equations
$g_i = a_i$ **AND** b_i $p_i = a_i$ **OR** b_i (1)
The carry-in signal for any adder block is calculated by using the formula

$$c_{i+1} = g_i + (p_i \cdot c_i) \quad (2)$$

Addition of two operands, A= (a_0, a_1……a_{n-1}) with B = (b_0, b_1……b_{n-1}) using parallel prefix adder is shown in Fig.1. It mainly consists of three stages, first is Pre-Computation or Generation stage, where we find three signals, carry generation (g_i), carry propagation (p_i) using equation (1) and half sum (h_i) using the relation

$$h_i = a_i \text{xor } b_i \quad (3)$$

Second stage is Carry Computation, where we find carry bit for each input bit using carry generate (g_i) and carry

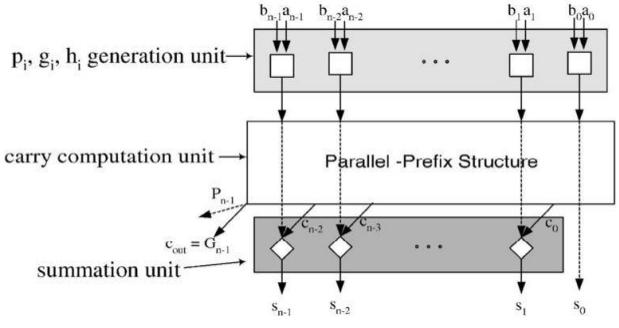

Fig 1: Block Diagram of parallel prefix adder

propagate (p_i) signal, using equation (2). For computation of carry bit signal we use different tree- architecture such as Han-Carlson, Kogge-stone, Brent-Kung, Ladner-Fischer, etc. Finally the last stage is Post-Processing or Summation stage, where we will find final sum for each bit using the following equation

$$S_i = h_i \text{xor } C_{i-1} \qquad (4)$$

Parallel Prefix adders compute carry-in at each level of addition by combining generate and propagate signals in a different manner (depending upon tree architecture). The implementation of □ ◇and ●, operators which are used in parallel prefix adders is shown in Fig.2

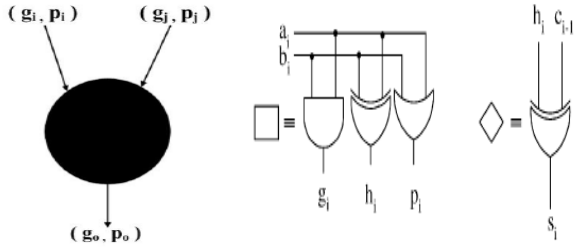

Fig. 2: Implementation of Operators

The black operator receives two sets of generate and propagate signals (g_i , p_i),(g_j , p_j), computes on it and produces a set of generate and propagate signals, (g_o , p_o) by using the following equations

$$(G_{i:j}, P_{i:j}) = (g_o , p_o) = (g_i , p_i) \text{ o } (g_j , p_j)$$

$$g_o = g_i + (p_i \cdot g_j) \text{ and } p_o = p_i \cdot p_j \qquad (5)$$

The "o" operator in (5) defines prefix operation between a pair of generate and propagate signals for carry computation.

2.1 Hybrid Parallel Prefix Adder A new parallel prefix adder architecture is developed by using black and Star operators, called Hybrid Parallel Prefix adder. Black operator as been already defined in section II. The Star operator is shown in Fig. 3.

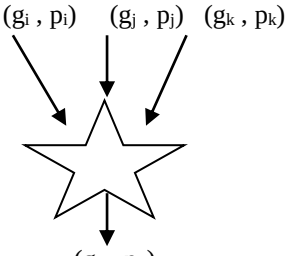

Fig 3: Star Operator

The Star operator, which takes three pairs of generate and propagate values $(\boldsymbol{g_i}, \boldsymbol{p_i})$, $(\boldsymbol{g_j}, \boldsymbol{p_j})$, $(\boldsymbol{g_k}, \boldsymbol{p_k})$ asinputs and produces a pair of generate and propagate output values (g_o , p_o) as follows

$$\left.\begin{aligned} g_o = g_i + (p_i \cdot g_j) + (p_i \cdot p_j \cdot g_k) \\ p_o = p_i \cdot p_j \cdot p_k \end{aligned}\right\} \qquad (6)$$

3. MODULO$2^n - 1$ADDER

4. 16-BIT MODULO$2^n - 1$ ADDER USING HYBRID-SPARSE STRUCTURE

For large number of bits, there will be more number of interconnection wiring between the black operators in the normal prefix structure, if only black operators are used, i.e. Using additional carry increment stage (Fig.4a) and EAC (Fig.4b).

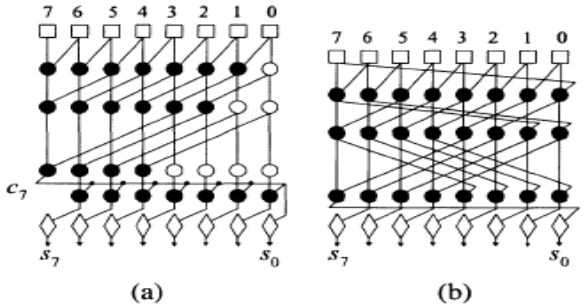

Fig 4: Modulo 2^8-1 Adder using (a) Additional Carry Increment, (b) End around[8]

This large number of prefix operator and wiring between them can be reduce using Hybrid parallel prefix operator, which is a combination of black and Star operator (Fig.3) Therefore by using sparse adder there will be a chance of saving considerable amount of area. Fig.5 shows the architecture of modulo 2^{16}-1 adder using Hybrid-sparse structure.

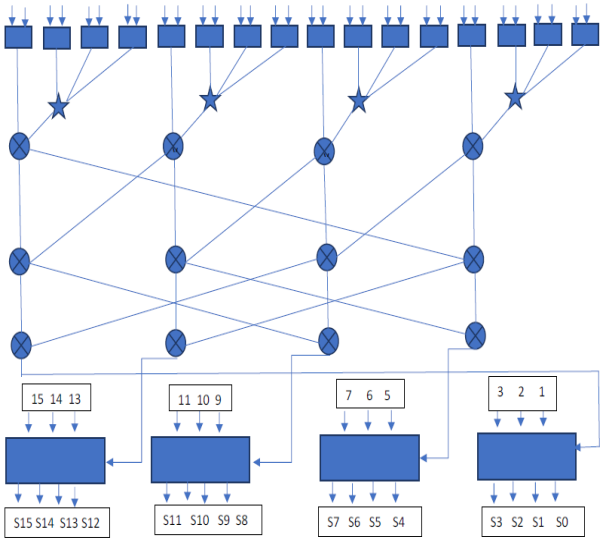

Fig 5: Modulo 2^8-1 Adder using Hybrid Sparse

5. SYNTHESIS RESULTS AND COMPARISON

The Hybrid-Sparse modulo 2^n-1 adder for 8-bit and 16-bit are compared with additional carry increment structure, EAC, Hybrid with Sparse All architectures of modulo adder are implemented using Xilinx ISE 8.1. The device used to implement the architecture is Spartan-3 XC3s400-4tq144. Table I, shows Synthesis Result. Fig.6 shows the snapshots simulation of modulo (2^{16}-1) adder

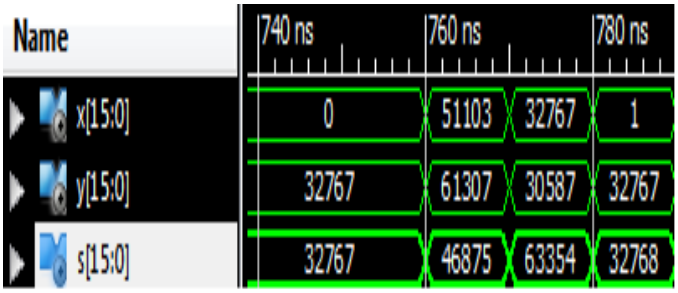

Fig. 6 Modulo $2^{16} - 1$ adder waveform

Table I: Synthesis Result

Parameter	Additional Carry Increment		End around carry		Hybrid with Sparse	
	8-bit	16-bit	8-bit	16bit	8bit	16-bit
Slices	20	46	23	64	16	35
4-i/p LUT's	34	80	40	112	27	60
Propagation delay (ns)	26.63	34.91	14.06	15.91	18.522	18.66

6. CONCLUSIONS

Analysis of tabulated result tells that the 8-bit and 16-bit modulo 2^n-1 adder and multiplier based on Hybrid-Sparse tree is area-efficient, since in proposed architecture the number of slices and LUT's are less compared with other architectures. Hybrid Sparse-tree architectures are used in less area utilization applications.

7. REFERENCES

[1] Koren, Computer Arithmetic Algorithms, Prentice-Hall,1993.
[2] T. R. N. Rao and E. Fujiwara, Error Control Coding for Computer Systems, Prentice-Hall, 1989.
[3] F. Halsall, Data Communications, Computer Networks and Open Systems, Addison Wesley, 1996.
[4] R. V. K. Pillai et aI., "A Low Power Approach to Floating Point Adder Design," in Proc. of the IEEE International
[5] R. Zimmerman, "Efficient VLSI Implementation of Modulo ($2^n \pm$ 1) Addition and Multiplication," in Proc. of14th Symp. Computer Arithmetic, April 1999, pp. 158-167.
[6] L. Kalampoukas et aI., "High-Speed Parallel-Prefix Modulo 2^n- 1 Adders," IEEE Trans. on Computers, vol. 49, no. 7,pp. 673-680,JuI. 2000.Conference on Computer Design, Oct. 1997, pp. 178-185.
[7] Efstathiou et aI., "Modulo 2^n- 1 Adder Design Using Select Prefix Blocks," IEEE Trans. on Computers, vol. 52, no. 11,pp. 1399-1406,2003.
[8] G. Dimitrakopoulos et aI., "A Family of Parallel-Prefix Modulo 2^n- 1 Adders," in Proc. of IEEE ASAP, 2003, pp. 326-336.
[9] Haridimos T. Vergos and andGiorgosDimitrakopoulos, et all, "New Architecture for Modulo $2^n - 1$ Adder", IEEE Transaction on computer, IEEE 2009