

## Research Article

# Implementation of Low Power, Delay and Area Efficient Shifters for Memory Based Computation

S. Soundarya, K. Prasanna

Department of Electronics and Communication Engineering, Arasu Engineering College,  
Kumbakonam-612501. India.

\*Corresponding author's e-mail: [soundaryaswaminathan@gmail.com](mailto:soundaryaswaminathan@gmail.com)

## Abstract

The efficient memory based computation is essential in DSP applications. The optimized Area is carried out for LUT's, and then delay will also reduce. In this paper, the barrel shifter which requires only one clock cycle for "n" number of shifts and it can shift all of the outputs up to three positions to the right side (LSB). In this brief, APC-OMS technique is used for LUT size reduction. The design also consists of Arbiter Shifter to select the order of access to shared resource among asynchronous requests. The FIFO algorithm is used in arbiter shifter for receiving the request and grant signals. This proposed system shows less area, delay and power compared to the existing shift register. The design synthesis and power analysis are carried out by using Xilinx 12.1 version software.

**Keywords:** Anti-symmetric coding; Odd Multiple Storage; First in First out; Look up table; Least significant bit.

## Introduction

Since 1970's VLSI plays a major role in communication and semiconductor devices. VLSI (Very Large Scale Integration) comprises thousands of transistors on a single IC Chips. VLSI is majorly linked with Low power, Area and Speed. Mainly CPU, ROM and glue logic all these functions are performed on a single VLSI Chip. The Power Consumption is important phenomenon in many applications. VLSI design built its structures as such as design analysis, design implementation, computer-aided design, simulation, testing. In VLSI modular technology it majorly deals with reducing interconnecting fabricating microchip Area [1]. It is the one where rectangular blocks are constructed by repetitive structures and they are connected by using wiring. For instance, the layout has been portioned into equal bit slices. In digital circuits shift registers is used to construct many applications and it is the basic building block. It is constructed by connecting flipflops for data transmission and it is majorly used for shifting the data. Flipflops deals with the timing problem. so in order to avoid timing problem we used Pulsed latches instead of flipflops. The pulsed latches also deal with timing problem but compared to pulsed latches, flipflops produces larger timing problem.

Shift registers are a type of sequential logic circuit, mainly for storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most of the registers possess no characteristic internal sequence of states. All flipflop is driven by a common clock, and all are set or reset simultaneously. A register allows each of the flip-flops to pass the stored information to its adjacent neighbor [2]. The storage capacity of a register is the total number of bits (1 or 0) of digital data it can retain. Each stage (flip flop) in a shift register represents one bit of storage capacity. Therefore the number of stages in a register determines its storage capacity. A computer or microprocessor-based system commonly requires incoming data to be in parallel format. But frequently, these systems must communicate with external devices that send or receive serial data. So, serial-to-parallel conversion is required. Generally to produce delay in the circuits, the serial in -serial out shift register can be used as a time delay device.

A N-bit shift Register by using register reusing concept. The pulsed latch has been used to reduce the time delay in the circuits [3]. SSASPL (static sense differential amplifier shared Pulsed Latch) which is the smallest latch with less number of transistors. The same similar

operation of latch and flip-flops is explained by using twisted ring counter or simply called as Johnson counter.

A logical shift is often used when its operand is being treated as a sequence of bits instead of as a number [4]. Shifting left by  $n$  bits on a signed or unsigned binary number has the effect of multiplying it by  $2^n$ . Shifting right by  $n$  bits on an unsigned binary number has the effect of dividing it by  $2^n$  (rounding towards 0).

A barrel shifter is a digital circuit that can shift a data word by a specified numbers of bits in one clock cycle [5]. It can be implemented as a sequence of multiplexors (mux), and in such an implementation the output of one mux is connected to the next mux in a way that depends on the shift distance. Barrel shifters are often utilized by embedded digital signal processors and general purpose processors to manipulate data. Shifting and rotating data is required in several applications, variable-length coding, and bit indexing. Barrel shifters are often utilized by embedded digital signal processors and general-purpose processors to manipulate data. Barrel shifter requires only one clock cycle for 'n number of shifts. An important form of arbiter is used in asynchronous circuits to select the order of access to a shared resource among asynchronous requests [6]. Its function is to prevent two operations from occurring at once when they should not.

## Research methodology

### *Anti-Symmetric product coding*

In earlier days, for memory based implementation of DSP algorithms involving Orthogonal transforms & digital filters had reported by several architectures but they could not find any significant work for LUT optimization [7]. Recently we introduced a new approach for LUT optimization in which only the odd multiples of fixed coefficient are to be stored which is termed as odd multiple-storage-scheme (OMS). An LUT size can also be reduced to half by another approach known as anti-symmetric product coding (APC) scheme whereas the product words are termed as anti-symmetric pairs.

A memory unit of  $[(2L/2) + 1]$  words of  $(W + L)$ -bit width is used to store the product values, where the first  $(2L/2)$  words are odd multiples of  $A$ , and the last word is zero.

A barrel shifter for producing a maximum of  $(L - 1)$  left shifts is used to derive all the even multiples of  $A$ .

The  $L$ -bit input word is mapped to the  $(L - 1)$ -bit address of the LUT by an address encoder, and control bits for the barrel shifter are derived by a control circuit

### *Odd Multiple Storage (OMS)*

As the name OMS itself specifies that it stores only odd multiples of fixed coefficient. The multiplication of a binary of binary word  $X$  of word size  $L$  with fixed coefficient  $A$ , instead of storing all possible  $2L$  product values, LUT stores only  $2L/2$  words corresponding to odd multiples of  $A$ , While all even multiples of  $A$  can be converted into odd multiples by left shift operations [8], from the above assumptions, the LUT for the multiplication of an  $L$ -bit input with a  $W$ -bit coefficient could be designed by the following strategy.

A memory unit of  $[(2L/2) + 1]$  words of  $(W + L)$ -bit width is used to store the product values, where the first  $(2L/2)$  words are odd multiples of  $A$ , and the last word is zero.

A barrel shifter for producing a maximum of  $(L - 1)$  left shifts is used to derive all the even multiples of  $A$ .

The  $L$ -bit input word is mapped to the  $(L - 1)$ -bit address of the LUT by an address encoder, and control bits for the barrel shifter are derived by a control circuits.

Product could be obtained by adding or subtracting the stored value  $(v - u)$  to or from the fixed value  $16A$  when  $x_4$  is 1 or 0, respectively.

Product word =  $16A + (\text{sign value}) \times (\text{APC word})$

### *APC-OMS combined structure*

The APC-OMS combined structure is shown in figure 1, consist of Address generator and control circuit, address decoder, LUT outputs and Barrel Shifter. The combined schemes of APC-OMS design of an LUT for  $L = 5$  for any coefficient width  $W$ . It consists of an LUT of nine words of  $(W + 4)$ -bit width, a four-to-nine-line address decoder, a barrel shifter, an address generation circuit, and a control circuit for generating the RESET signal and control word  $(s_1, s_0)$  for the barrel shifter is shown in figure. The APC-OMS combined optimization of the LUT can also be performed for signed

values of  $A$  and  $X$  [9]. When both operands are in sign-magnitude form, the multiples of magnitude of the fixed coefficient are to be stored in the LUT, and the sign of the product could be obtained by the XOR operation of sign bits of both multiplicands. When both operands are in two's complement forms, a two's complement operation of the output of the LUT is required to be performed for  $x_4 = 1$ . There is no need to add the fixed value  $16A$  in this case, because the product values are naturally in anti-symmetric form. For the multiplication of unsigned input  $X$  with signed, as well as unsigned, coefficient  $A$ , the products could be stored in two's complement representation, and the add/subtract circuit could be modified.

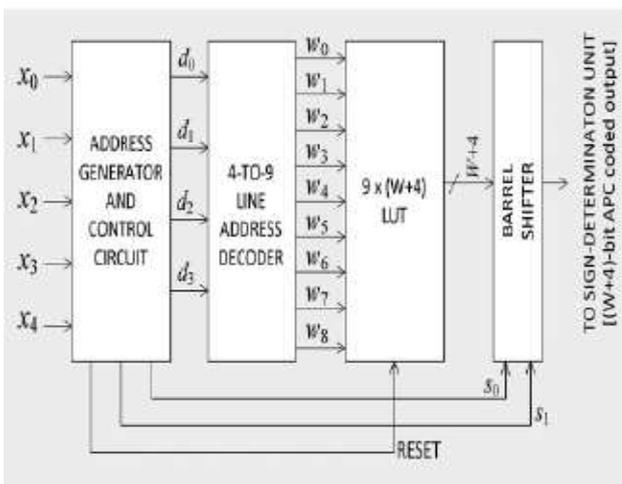


Figure 1. APC-OMS combined structure

**Arbiter shifter**

Many input ports which are requestor want to access a common physical channel resource. In this case, an arbiter is required to determine how the physical channel can be shared amongst many requestors. When we think about arbitration Logic, we have to consider many factors. If many flits arrive at buffers from several virtual channels and these flits are destined for one physical channel, an arbiter receives request signals from buffer such as FIFO empty or full signals [9]. These FIFO empty and full signals are generated by comparing write pointers and read pointers.

For example, as shown in figure 2, if a write pointer has the same value as a read pointer, FIFO empty signal will be generated. On the other hand, if a write pointer has more value than read pointers, FIFO full signal will be made. The figure below shows general arbitration flow in FIFO (First In First Out)

based request and a grant system. When a new flit arrives at FIFO, a write pointer gets incremented and request signal is generated. An arbiter receive  $N$  request signals and grant only one buffer, and this grant signal increases a read pointer of corresponding FIFO. This type of arbitration flow is used to implement a NoC router VA and SA logic. Fairness is a key property of an arbiter. In other words, a fair arbiter support equal service the different requests [10]. In FIFO environment, requesters are served in the order they made their requests. Even though an arbiter is fair, if traffic congestion is not fair in a NoC environment, the system cannot be fair.

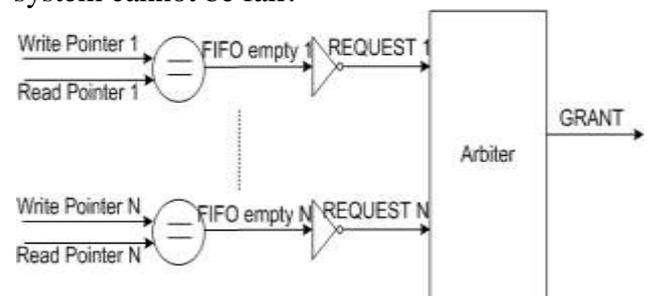


Figure 2. Example of arbitration in FIFO

**Block diagram of arbiter shifter**

The figure 3 shows the block diagram of a common node in any Network on Chip. Typically there are 5 entry points into the node (North, South, West, East, and Access Point). We did not consider other node configurations (e.g. hexagonal networks have previously been suggested for parallel processing). The channel FIFOs transmit and receive data and metadata to and from the node [11]. How and what is transmitted is modularized to involve only the arbiter and the channel FIFO's channel metadata signals.

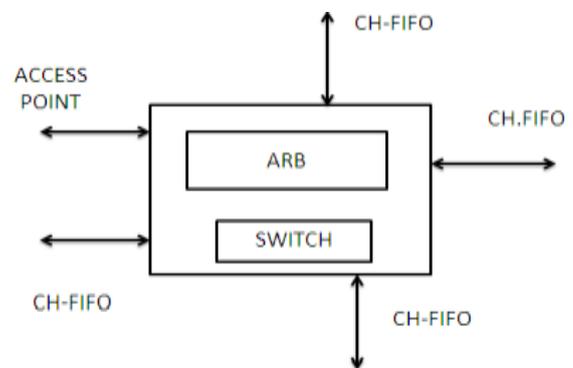


Figure 3. Arbiter shifter structure

The switch itself is an all-to-all mux that allows multiple paths of communication simultaneously. Any of these components can be

switched out to add various NoC concepts (virtual channels, QoS, reservation systems) without affecting the remainder of the system.

**Architecture of entire structure of arbiter unit**

The switch architecture consists of five input buffers and an arbitration unit as shown in the figure 4, which collects the control information and makes the arbitrations, a crossbar and a central cache to temporally store the head packets from the buffers [12]. The NoC architecture has a great advantage on the bus architecture. It has better latency and throughput than the bus architecture.

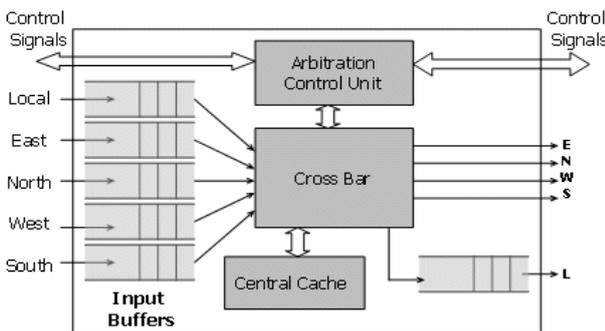


Figure 4. Arbiter unit

NoC architecture can be described by its strategy for routing, flow control, switching, arbitration and buffering and the topology used in this architecture Arbitration is responsible to arrange the use of channels and buffers for the messages. Switching is the mechanism that gets data from an input channel of a router and places

it on an output channel [13]. Architecture a small central cache is embedded into every switch to reduce the deadlock problems. It increases the throughput and average latency of the system. The head packet of any buffer can be stored in the caches if any packets calls and it stored in the buffer so the blocked packets to be bypassed without delay. The arbitrator collects the information from the neighboring switches. If any of the resources is available, the arbitrator will check the cache and lookup tables for the input buffers (in the order of cache, East, North, West and South buffers; obviously cache has the top priority in the sequence) and forward to the output port.

**Results and discussions**

**Barrel shifter and arbiter shifter**

The APC-OMS technique is used to analyse the Barrel Shifter. The Anti-symmetric Product Coding is used to reduce the LUT Size [14]. Odd Multiple Storage is used in left shifting of even multiples of A. The Barrel shifter is responsible for shifting operations as shown in the figure 5. The Arbiter shifter is used for explaining the concept of shifting operations in memory based architecture using the algorithm of FIFO algorithm as shown in the figure 6, implemented by using Xilinx 12.1 software tool.

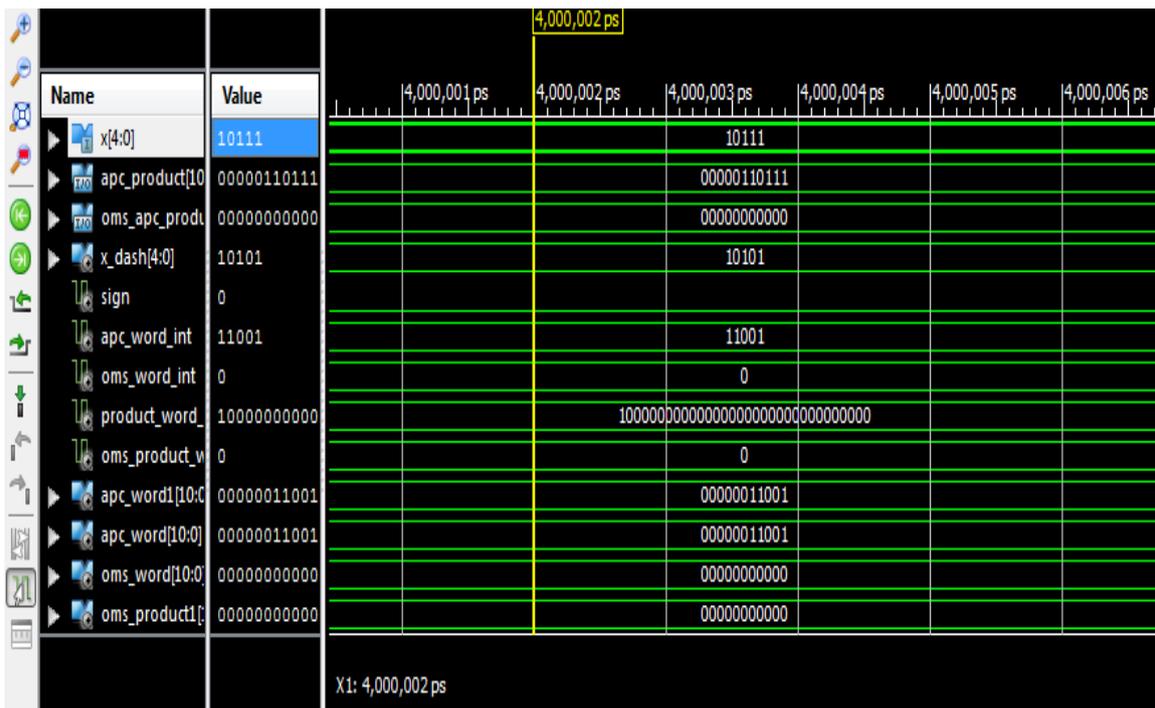


Figure 5. Barrel shifter

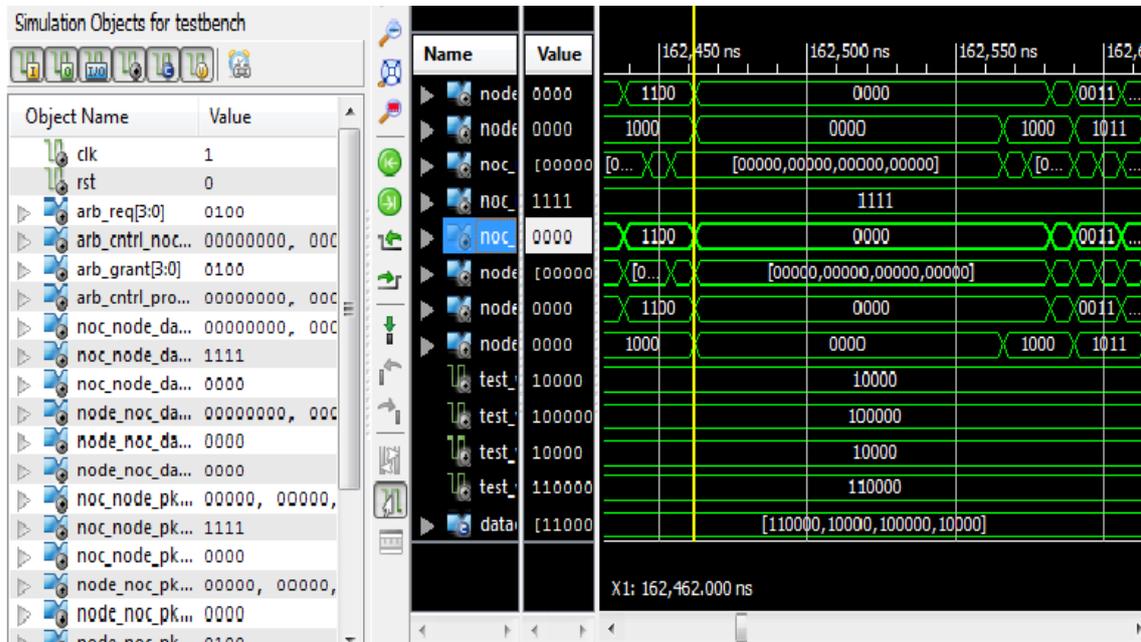


Figure 6. Arbiter shifter

## Conclusions

The technique of APC-OMS method is used to reduce the LUT size. Anti-symmetric Product Coding is used for sign magnitude function. Moreover shifting operations are done by using Barrel shifter. It is used to shift the odd multiples using OMS method. The Arbiter shifter enables the request-grant signals to analyse the operation of memory structure. The FIFO algorithm is used in Arbiter shifter for receiving the request and grant signals. Thus less area, delay and power efficient shifters are designed compared to existing shift register is implemented by using Xilinx 12.1 version software tool

## Conflict of interest

Authors declare there are no conflicts of interest.

## References

- [1] Anusya S, Bhubaneswari PR. Design and Implementation of Motion Artifact Reduction Asic for wearable ECG recording. *American-Eurasian Journal of Scientific Research*. 2014;10(3):154-159.
- [2] Bhavin DM, Altaf D. VHDL Implementation of 8Bit Vedic Multiplier using Barrel Shifter. *International Journal of Scientific Research and Development*. 2014;2(1):156-158.
- [3] Feng C, Lu Z. Addressing Transient and Permanent Faults in Noc with efficient Fault Tolerant Deflection Router. *IEEE Transactions on Very Large Integration Systems*. 2013;21(6):1053 - 1066.
- [4] Liya P, Reen P. Modified APcOMS technique for Memory Based Computing. *Global Colloquium in Recent Advancement and Effectual Researchers in Engineering Science and Technology*. 2016.
- [5] Manoj MK, Sunita P. Analysis of Different Multiplication Algorithm and FPGA Implementation of Recursive Barrel Shifter Method for Multiplication. *International Research Journal of Engineering and Technology*. 2016;3(1):1141-1142.
- [6] Mallela U, Ravi J. Design of Digital FIR Filter Using LUT Based Multiplier. *International Journal of Electronics Communication and Computer Technology*. 2013;3(5):477-479.
- [7] Mohanamma MP, Vamsee K. Design and Implementation of LUT Based Multiplier using APCOMS Technique. *International Journal of Scientific Research of Development*. 2013;1(7):476-479.
- [8] Pavar Kumar, P, Neelima R. Digital FIR Filter Implementation using Combined APC-OMS Technique. *International Journal of VLSI and Embedded Systems*. 2014;4:06122.
- [9] Pragati S, Anchal K, Anita D, Pallavi G. Barrel Shifter. *International Journal of Science and Engineering and Technology*. 2014;2(7):1434-1440.
- [10] Kamal R, Yadav N. Noc and Bus Architecture a Comparision. *International*

- Journal of Engineering Science and Technology. 2012;4(4):1438-1442.
- [11] Pappachan R, Vijayakumar V. Design and Analysis of a 4Bit Low Power Barrel Shifter in 20 nm FINFET Technology. International Journal of Engineering and Sciences. 2013;2(3):17-25.
- [12] Ramya, Sudha M. LUT Optimization using Combined APCOMS Technique for Memory Based Computation. International Journal of Computer Applications in Engineering Sciences. 2013;3:119-127.
- [13] Sreelakshmi K, Srinivasa Rao A. An Advanced and Area Optimized LUT Design using APC and OMS. International Journal of Computer Science and Information Technologies. 2012;3:4265-4269.
- [14] AbhisekT, Ketan D, Patel N. An Approach for Implementation of Bus Arbitration Techniques. International Journal of Scientific Research of Development. 2016;4(2):112-116.

\*\*\*\*\*