



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

To Reduce Power Consumption By Add and Shift Multiplier Design Using BZFAD Architecture

Badikala Rameshbabu

RTL Design Verification Engineer, S.H Informatics, India

badikala.rameshbabu@gmail.com

Abstract

A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processors and microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. The architecture considerably lowers the switching activity of conventional multipliers. The modification to the multiplier which multiplies A by B Include the removal of the shifting register, direct feeding of A to the adder, bypassing the adder whenever possible, using a ring counter instead of a binary counter and removal of the partial product shift. The simulation result for 8 bit multipliers shows that the proposed low power architecture lowers the total power consumption by 35.25% and area by 52.72 % when compared to the conventional architecture. Also the reduction in power consumption increases with the increase in bit width.

Keywords: Low power multiplier, low power ring counter, sources of switching activities.

Introduction

A multiplier is one of the key hardware blocks in most digital and high performance systems such as FIR filters, digital signal processor microprocessors etc. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in multiplier. Thus making them suitable for various high speed, low power and compact VLSI implementations. However area and speed are two conflicting constraints. So improving speed results always in larger areas. So here we try to find out the best trade off solution among them. Generally as we know multiplication goes in two basic steps. Partial product and then addition. Hence here, we first try to design different adders and compare their speed and complexity of circuit i.e. the area occupied. Considering the design of Wallace tree multiplier then followed by Booth's Wallace multiplier and comparing the speed and Power consumption in them. MULTIPLIERS are among the fundamental components of many digital systems and, hence, their power dissipation and speed are of prime concern. For portable applications where the power consumption is the most important parameter, one should reduce the power dissipation as much as possible. One of the best ways to reduce the dynamic power dissipation, henceforth

referred to as power dissipation in this paper, is to minimize the total switching activity, i.e., the total number of signal transitions of the system. Many research efforts have been devoted to reducing the power dissipation of different multipliers (e.g., [1]–[3]). The largest contribution to the total power consumption in a multiplier is due to generation of partial product. Among multipliers, tree multipliers are used in high speed applications such as filters, but these require large area. The carry-select-adder (CSA)-based radix multipliers, which have lower area overhead, employ a greater number of active transistors for the multiplication operation and hence consume more power. Among other multipliers, shift-and-add multipliers have been used in many other applications for their simplicity and relatively small area requirement [4]. Higher-radix multipliers are faster but consume more power since they employ wide registers, and require more silicon area due to their more complex logic. In this work, we propose modifications to the conventional architecture of the shift-and-add radix-2 multipliers to considerably reduce its energy consumption.

A Motivation

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal

processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tools, power consumption has become a critical concern in today's VLSI system design. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processin capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low- power VLSI system design. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. We study different **adders** and compare them, so that we can judge to know which adder was best suited for situation. **Ripple Carry Adder** has a smaller area while having lesser speed. **Carry Select Adders** are high-speed but posses a larger area. **Carry Look Ahead Adder** is in between the spectrum having a proper tradeoff between time and area complexities. Coming to **Multipliers**, we consider different Multipliers starting from Array Multiplier to Wallace Tree, Booth Multipliers, both Radix-2 and Radix-4. Many research efforts have been devoted to reducing the power dissipation of different multipliers (e.g., [1]–[3]). The largest contribution to the total power consumption in a multiplier is due to generation of partial product. Among multipliers, tree multipliers are used in high speed applications such as filters, but these require large area. The carry-select-adder (CSA)-based radix multipliers, which have lower area overhead, employ a greater number of active transistors for the multiplication operation and hence consume more power. Among other multipliers, shift-and-add multipliers have been used in many other applications for their simplicity and relatively small area requirement [4]. Higher radix multipliers are faster but consume more power since they employ wider registers, and require more silicon area due to their more complex logic.

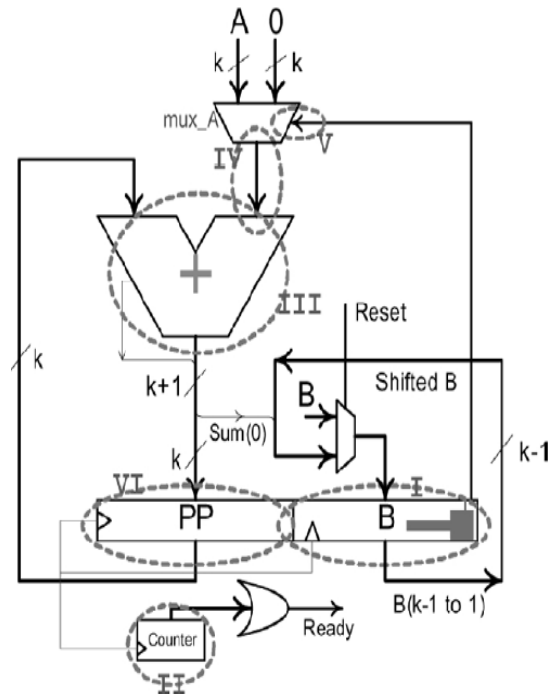


Fig 1.0 Architecture of Shift-and-add Architecture

Shift and Add Multipliers

Figure 1 shows the architecture of a conventional shift and add multiplier[4]. The dashed oval show the major sources of switching activities. The multiplier is shifted in each cycle and the bit which getting out of register B is connected to the select pin of multiplexer, mux_A. As the select signal changes, the output of mux_A also changes. This causes the adder operation. partial product is required to be shifted in every cycle. The counter is for checking whether the required number of operations has been performed. The major sources of switching activities are summarized as below

- Shifting of the 'B' register
- Activity in the counter
- Activity in the adder
- Switching between '0' and 'A' in the multiplexer
- Activity in the multiplexer select
- Shifting of the partial product register

By eliminating or reducing the switching activity described above, low power architecture can be derived.

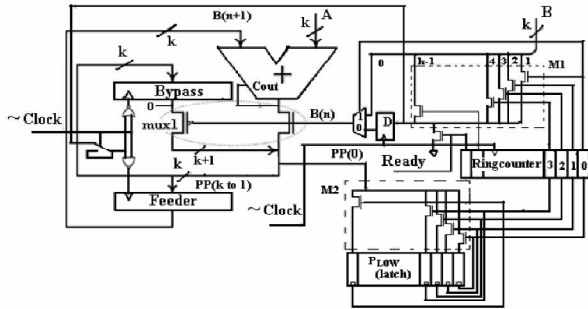


Fig 2.0 BZFAD Architecture

Hot Block Ring Counter

In the proposed multiplier, we make use of a ring counter the architecture of which is described in this section. In a ring counter always a single “1” is moving from the right to the left. Therefore in each cycle only two flip-flops should be clocked. To reduce the switching activity of the counter, we propose to partition the counter into \bar{E} blocks which are clock-gated with a special multiple-bit clock gating structure shown in Fig. 4, whose power and area overheads are independent of the block size. In the proposed counter, called *Hot Block* ring counter (see Fig. 3) fewer superfluous switching activity ex-ists and there are many flip-flops whose outputs do not go to any clock gating structure. This noticeably reduces the total switching activity of the ring counter. We have utilized the property that in each cycle, the outputs of all flip-flops, except for one, are “0”. Thus in the partitioned ring counter of Fig. 3, there is exactly one block that should be clocked (except for the case that the “1” leaves a block and enters another). We call this block the *Hot Block*. Therefore, for each block, the clock gating structure (CG) should only know whether the “1” has entered the block (from the right) and has not yet left it (from the left). Passing the clock pulses to the block once the “1” appears at the input of the first flip-flop of the block. It shuts off the clock pulses after the “1” leaves the left-most flip-flop of the block.

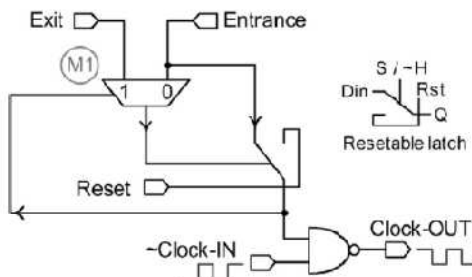


Fig 3.0 Clock gating structure

The clock gating structure (CG) proposed for the Hot Block ring counter is shown in Fig. 4. It is composed of a multiplexer 33_3 , a NAND gate, and a resettable latch. In this work, the multiplexers are implemented with transmission gates. In addition to the *Reset* and clock signals, there are two other signals called *Entrance* and *Exit*, coming from the neighboring left and right blocks. These are used to determine whether the “1” is present in the block to which the output of Multiplier Based On Add And Shift Method By Passing Zero 4 the CG goes. When the active high Reset signal is “1”, the latch is reset which causes the value of the Entrance signal to be placed on the 33_3 line of the latch through multiplexer M1. This in turn causes the latch to read the *Entrance* signal, which was previously reset to “0”, since the whole ring counter is reset and all the bits except the first are reset to “0”. After a sufficiently long interval, *Reset* goes to “0” and since *Entrance* has a value of “0”, the latch keeps holding “0” on its output, forcing *Clock-OUT* to “1” after the CG is reset. This condition should persist until the “1” is about to enter to the block.

Results and Analysis

After understanding the architecture of both conventional and BZFAD multipliers, next step was to implement it. In order to accomplish this we write a code in Very High Speed Integrated Circuit- Hardware Descriptive Language [VHDL]. This code was synthesized using Xilinx and simulated using ISE simulator [isim], and was implemented by burning on Spartan2 FPGA kit. Simulation results, timing summary, area utilization and power analysis report is shown below.

A. Simulation Results

The simulation results for both the conventional and BZFAD architectures follow in the order given below,
 4 Bit conventional multiplier
 8 Bit Conventional Multiplier
 4 Bit BZFAD Multiplier
 8 Bit BZFAD Multiplier

B. Timing Summary

	Conventional 8 bit	BZFAD 8 bit
Minimum period	8.258 ns	6.975 ns
Maximum frequency	121.094 Mhz	143.362 Mhz
Minimum input arrival time	8.426 ns	7.167 ns

	Conventional 16 bit	BZFAD 16 bit
Minimum period	9.946 ns	6.564 ns
Maximum frequency	100.540 Mhz	152.352 Mhz
Minimum input arrival time	10.281 ns	7.502 ns

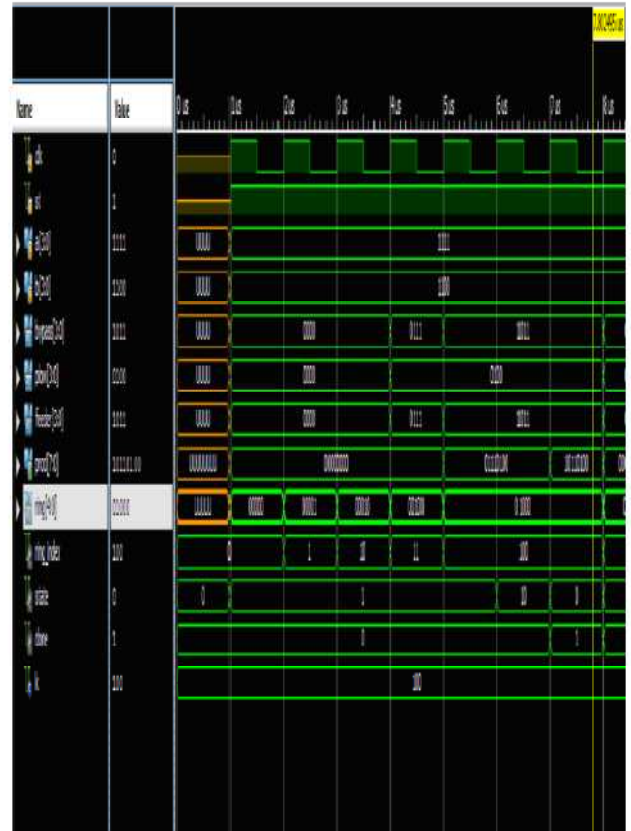


Fig Simulation For BZFAD Architecture

C Area Utilization

Number of	Conventional 4 bit			BZFAD 4 bit		
	Utilized	Available	Percentage utilization	Utilized	Available	Percentage utilization
Slices	52	768	6%	36	768	4%
Slices F/F	49	1536	3%	47	1536	3%
4 input LUTs	95	1536	6%	66	1536	4%
Bonded IOBs	19	146	13%	18	146	12%
Glck	2	4	50%	2	4	50%



Fig Simulation for Conventional

Conclusion

The proposed architecture lowers the power dissipation and area when compared to a conventional shift and add multiplier shown in Figure 15 . A multiplexer with one hot encoded bus selector is used for avoiding the switching activity due to the shifting of the multiplier register. Feeder and bypass registers are used for avoiding the unnecessary additions. The proposed architecture makes use of bit width control logic and a low power ring counter The design can be verified using Modelsim 6.5 with VHDL code and power consumption is analyzed using Xilinx software. From Table 3 and Figure 15 , proposed architecture can attain 35.25% power reduction and 52.75% area saving when compared to the conventional shift and add multipliers. Also from Table 4 and Figure 16 reduction of power consumption in multipliers can be increases with the increase in bit width of operands, whereas in design [5] reduction in power consumption decreases with the increase in bit width.

References

- [1] A.Chandrakasan and R. Brodersen, "Low Power CMOS Digital Design", IEEE J. Solid State Circuits, Vol.27, no.4, pp 473-484, Apr 1992
- [2] N.Y.Shen and O.T.C.Chen."Low power multipliers by minimizing switching activities of partial products" in Proc. IEEE Int.Symp.Circuits Syst., May 2002, Vol.4, pp 93-96.
- [3] O.T.Chen, S.Wang and Y, W.Wu "Minimization of switching activities of partial products for designing low power multipliers" IEEE Trans. Very Large Scale Integer .(VLSI)Syst. , Vol .11, No-3, pp418-433, June 2003 International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010 22
- [4] B.Parhami Computer arithmetic algorithms and Hardware designs 1 st ed.Oxford U.K. Oxford Univ, Press 2000.
- [5] K.H.Chen and Y.S.Chu , "A low power multiplier with spurious power suppression technique" , IEEE Trans. Very Large Scale Integr .(VLSI)Syst. , Vol.15 , no-7,pp846-850, July 2007.
- [6] A. Chandrakasan and R. Brodersen, "Low-power CMOS digital de-sign," *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, Apr. 1992.
- [7] N.-Y. Shen and O. T.-C. Chen, "Low-power multipliers by minimizing switching activities of partial products," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2002, vol. 4, pp. 93–96.
- [8] B. Parhami, *Computer Arithmetic Algorithms and Hardware Designs*, 1st ed. Oxford, U.K.: Oxford Univ. Press, 2000.
- [9] O. Chen, S. Wang, and Y. W. Wu, "Minimization of switching activities of partial products for designing lowpower multipliers," *IEEE Trans. Very Large Scale Integr.*
- [10] *(VLSI) Syst.*, vol. 11, no. 3, pp. 418–433, Jun. 2003.