Journal of Computer Science 6 (10): 1117-1122, 2010
ISSN 1549-3636

Low Power Multiplier Design Using Latches and Flip-Flops

[1]C.N. Marimuthu and [2]P. Thangaraj
[1]Department of Electronics and Communication Engineering,
Maharaja Engineering College, Avinashi,
[2]Department of Computer Technology, Kongu Engineering College, Perundurai,
Anna University, Tamil Nadu, India

Abstract: Problem statement: Power dissipation is designated as critical parameter in modern VLSI design field. In VLSI implementation low power concept is necessary to meet Moore's law and to produce consumer electronics with more back up and less weight. To save significant power consumption of a VLSI design, it is a good direction to reduce its dynamic power which is the major part of power dissipation. Multiplication occurs frequently finite impulse response filters, fast Fourier transforms, discrete cosine transform and other important DSP and multimedia kernels. Being one among the functional components of many digital systems the reduction of power dissipation in multipliers should be as much as possible. **Approach:** In this study a low power structure called Bypass Zero Feed A Directly (BZFAD) for shift and add multipliers was proposed for reducing the switching activity. **Results:** The simulation results showed conventional and proposed BZFAD 8 bit multipliers. **Conclusion:** From these results, BZFAD can attain considerable power reduction and area saving when compared to the conventional shift and add multipliers.

Key words: Hot block ring counter, low power multiplier, low power ring counter, reduction in switching activity

INTRODUCTION

The Golden formula for calculating dynamic power dissipation is P = C_LV^2f. Power reduction can be achieved by various manners. They are reduction of output load capacitance C_L, reduction of power supply voltage V and reduction of clock frequency f.

Many research efforts have been devoted to reduce the power dissipation in different multipliers (Chandrakasan *et al*., 1992; Shen and Chen, 2002; Chen *et al*., 2005). Among multipliers, tree multipliers are used in high speed applications such as filters, but these require large area (Chen and Chu, 2007; Huang and Ercegovac, 2005). 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 applications for their simplicity and relatively small area requirement (Nelson and Nagle, 1995; Wang *et al*., 2004).

Steps leading to a low power multiplier: Architecture of a conventional shift and add multiplier with multiplicand A and multiplier B is shown in Fig. 1.

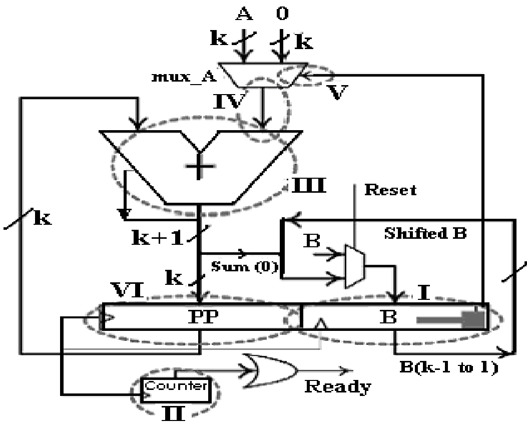


Fig. 1: Architecture of conventional shift and add multiplier with major source of switching activity

In the conventional shift and add multiplier, multiplier B is needed to be shifted in every cycle. The LSB obtained by shifting of B is connected to the select pin of multiplexer mux_A. It decides whether the multiplicand A or 0 is to be added with the partial

Corresponding Author: C.N. Marimuthu, Department of Electronics and Communication Engineering,
Maharaja Engineering College, Avinashi, Anna University, Tamil Nadu, India

product obtained in the previous cycle for forming the new partial product. From this operation of conventional shift and add multiplier we can see that there are many sources of switching activities present in the conventional architecture. There are 6 major sources of switching activity in the multiplier, which are marked in the dashed ovals in the Fig. 1. They are, (a) shifting of the B register, (b) activity in the counter, (c) activity in the adder, (d) switching between '0' and A in the multiplexer, (e) activity in the multiplexer select, (f) shift of the partial product register.

Power consumption can be lowered by minimizing or removing any of the above switching activity. More power reduction can be achieved by reducing the switching activity of nodes with higher capacitance. As an example B (0) is the selector line of the multiplexer which is connected to K gates for a K bit multiplier. If we somehow eliminate this node, noticeable power saving can be achieved.

The BZFAD architecture: By eliminating or reducing the sources switching activity described in the registers and counters low power architecture of multiplier can be derived. The proposed BZFAD architecture with multiplicand 'A' and multiplier 'B' is shown in Fig. 2. In the BZFAD architecture (Mottaghi *et al*., 2009), for getting the reduction of power consumption the main areas concentrated are described as in Fig. 1.

Shifting of b register (multiplier) using hot block ring counter: In the conventional architecture register B should be shifted to the right in every cycle. It's right bit appear as B (0). B (0) is used to select between A (multiplicand) and 0. If B (0) is one then A should be added to the previous partial product and, if B (0) is zero, then '0' should be added to the previous partial product.

The rights shifting of B in each cycle give rise to some switching activities. To avoid this we use a low power ring counter to select the required bit of B without shifting in each cycle. The BZFAD architecture uses a multiplexer with a one hot encoded bus selector choosing the hot bit of B in each cycle.

Operation in the adder with feeder and bypass registers: In the conventional architecture if LSB of B equal to zero then the current partial product is added to zero and if LSB of B equal to one then the current partial product is added to A. Addition of zero leads to unnecessary transitions in the adder that is, the adder can be bypassed if 0 is to be added and the partial product is required to be shifted to right by one bit. In BZFAD architecture the modifications are made by using Feeder and Bypass registers. The operations of the adder are optimized using these two registers.

Feeder and Bypass registers are used to bypass the adder in cycles in which the LSB of B is zero. In each cycle the hot bit of next cycle is checked and the following operations are performed:

- Feeder is clocked if LSB of B in next cycle equal to "1"
- Bypass is clocked if LSB of B in next cycle equal to "0"

Thus the current partial product is stored either in feeder or in bypass register. NAND and NOR gates are used to clock feeder and bypass registers. Since these are inverting logic inverted clock is fed to them. It is shown as ~Clock in Fig. 2. Thus the reduction of switching activity in adder is mainly due to the following reasons.

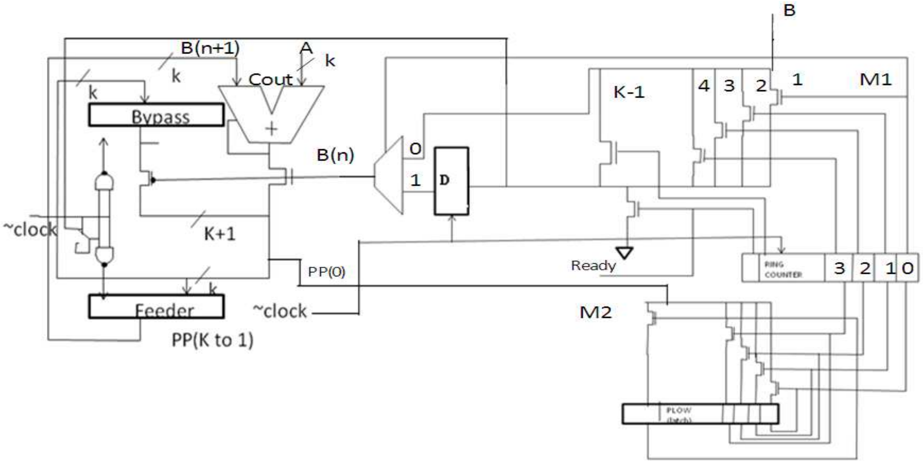


Fig. 2: Architecture of low power multiplier (BZFAD)

The right input of adder is A, which is constant during multiplication. This enables us to removing the multiplexer and feeding A directly to the adder, resulting in a noticeable power saving. In each cycle when the next hot bit is zero, feeder is not clocked and current partial product is stored in bypass register. So there is no transition in the adder input which also causes power saving.

Shifting of the partial product register using P_{Low} latch: The computation process of a multiplier manipulates two input data to generate many partial products for subsequent addition operations, which require a lot of switching activities. Thus switching activities within the functional units of a multiplier accounts for the majority of power dissipation of a multiplier as described in the following equation:

$$P_{Switching} = \alpha C V^2_{dd} f_{clk} \quad (1)$$

Where:

α = The switching activity parameter
C = The loading capacitor
V_{dd} = The operating voltage
f_{clk} = The operating frequency

Minimizing switching activities can efficiently lower power dissipation without affecting the circuit operation performance (Chen *et al*., 2003).

The least significant bits of partial products need not to be shifted for completing the multiplication. We take advantage of this observation that is the multiplication can be completed by processing the most significant bits of partial products. The BZFAD architecture uses P_{Low} latch to store the lower half of partial product. It has K latches for K bit multiplier.

In the first cycle, the LSB of partial product become finalized and is stored in the right most latch of P_{Low}. In the subsequent cycles the next LSBs are finalized and stored in the proper latches. The ring counter output is used to open the proper latch. Using this method no shifting of lower half of partial product is required; shifting is required only for the higher half of the partial product.

Hot block ring counter: The ring counter used in the BZFAD multiplier is noticeably wider than the binary counter used in the conventional architecture. Therefore an ordinary ring counter, if used in BZFAD would raise more transitions than its binary counterpart in the conventional architecture. To minimize the switching activity of the counter we use low power ring counter.

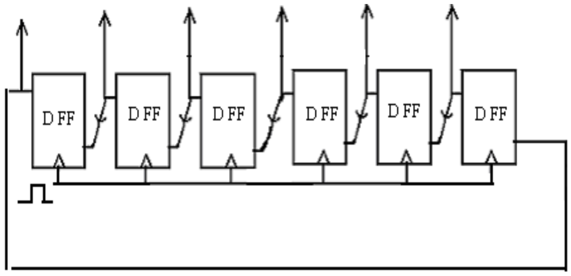


Fig. 3: Conventional synchronous ring counter

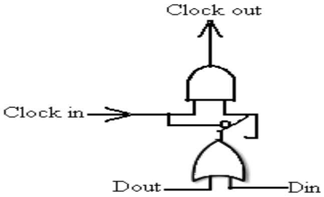


Fig. 4: Clock gator logic with flip flop's input and output

Unnecessary transitions in the conventional ring counter: An n-bit synchronous ring counter is built by cascading n D flip-flops in a chain as shown in Fig. 3.

In the above architecture all flip-flops have a common clock signal and each clock pulse is applied to all flip flops whereas inspecting the movement of the '1' in the counter chain reveals that each clock pulse must be applied to only 2 flip flops (not all of them).Therefore on each clock pulse (n-2) × s unnecessary transitions are raised in which s is the total number of transitions raised in a single flip flop and n is the number of flip flops.

Steps towards hot block ring counter: According to the previous discussion some flip-flops can be clock gated leading to fewer switching activities. A flip-flop in a ring counter must be clocked if and only if either its input or its output is "1" immediately before the triggering clock edge comes. Therefore only 2 flip flops must be clocked in each cycle.

The clock gating logic in the Fig. 4 ORs the value of flip flop's input and output on positive clock edges stores the result in a latch. The output of the latch determines whether or not to gate the clock signal. This clock gator is positive edge triggered.

If we want to avoid all the unnecessary transitions raised by the clock signal we should provide each flip-flop with the clock gating circuitry of Fig. 4, but this solution ends up with a large area overhead plus due to transitions in clock gator themselves the resulting ring counter will not have fewer switching activity. A better solution is used in the BZFAD architecture.

One of the important properties of the ring counter is that its output is one hot encoded in Fig. 5. This property of the ring counter makes its output wide especially as the counter size increases. As an example, consider a 5 bit binary counter which counts from 0-31. A ring counter with the same counting range is 32 bit wide. To reduce the switching activity of the counter the counter is partitioned in to a number of blocks which are clock gated with a special clock gating structure whose power and area overheads are independent of the block size. The clock gating structure is shown in Fig. 6.

In the partitioned ring counter there is exactly one block should be clocked, it is the block which is having a “1” in it at that time (except for the case that one leaves a block and enters another). The block which is clocked is called hot block. Therefore for each block the Clock Gating structure (CG) should only know whether “1” has entered the block (from the right) and has not yet left it (from the left). The CG starts 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.

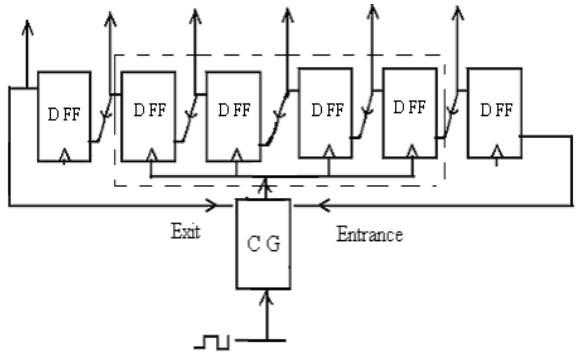


Fig. 5: Hot block architecture for ring counter with block of size 4

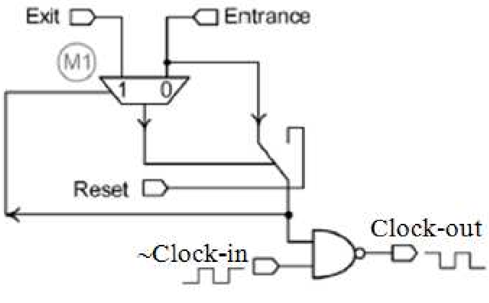


Fig. 6: Clock gating structure for the Hot Block ring counter

In Fig. 6 the signals entrance and exit is coming from the neighboring right and left blocks. The entrance and exit signals have special meanings as follows. When”1” entrance means that “1” is about to enter the block in the next cycle. This line is connected to the input of the left most flip flop in the right hand block. The exit signal indicates that “1” has left the block and hence it should no longer be clocked.

In Fig. 2 the s M1 and M2 are controlled by the same hot block ring counter. So the low power ring counter has the main role in BZFAD architecture.

MATERIALS AND METHODS

In this study, we propose a low-power, low-area multiplier using BZFAD architecture. The BZFAD architecture avoids the unwanted addition and thus minimizes the switching power dissipation. The conventional (Chen *et al*., 2005; 2006) and proposed design of 8 bit multiplier functionality can be verified using Model-Sim software and synthesized for getting power and delay report with Xilinx software back end tool.

RESULTS

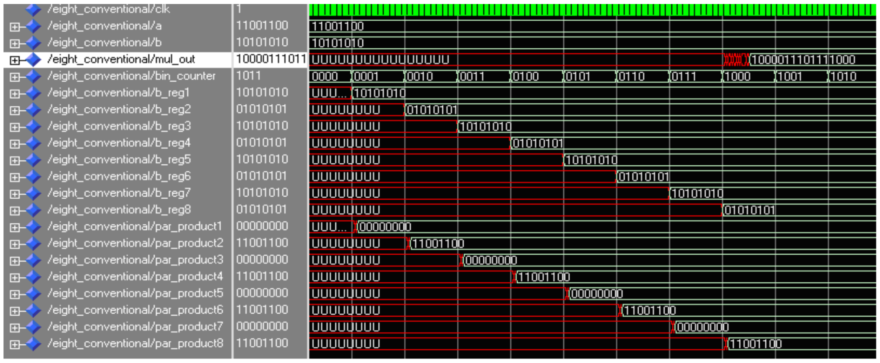
The simulation results of conventional and proposed BZFAD 8 bit multipliers are shown in Fig. 7 and 8. The Fig. 9 and 10 shows the power reports of both multipliers.

Area report for conventional shift and add multiplier is given below:

```
Design Information
Command Line: Map -p xc2s100-fg456-5 -cm area -k 4
-c 100 -tx off
eight_conventional.ngd
Target Device: x2s100
Target Package: fg456
Target Speed: -5
Mapper Version: spartan2 -- $Revision: 1.58 $
Mapped Date: Mon Dec 21 17:14:33 2009
Design Summary
 Number of errors: 0
 Number of warnings: 0
 Number of Slices: 662 out of 1,200 55%
```

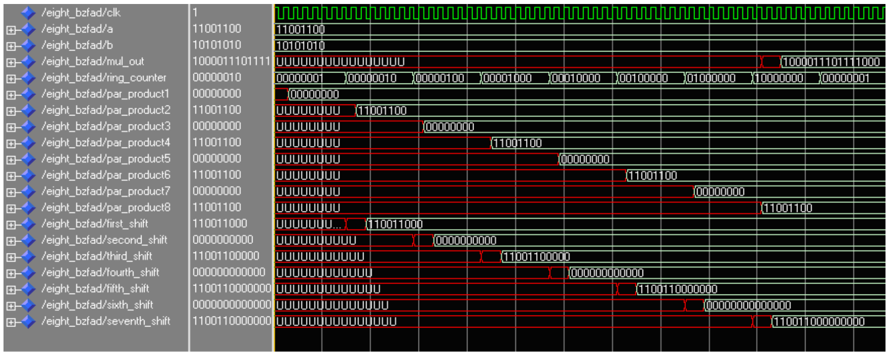
Area report for BZFAD multiplier is given below:

```
Design Information
Command Line: map -p xc2s100-fg456-5 -cm area -k 4
-c 100 -tx off
eight_bzfad.ngd
```



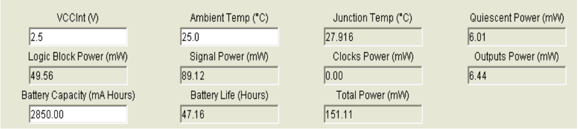
```
/eight_conventional/clk              1
/eight_conventional/a                11001100
/eight_conventional/b                10101010
/eight_conventional/mul_out          10000111011
/eight_conventional/bin_counter      1011
/eight_conventional/b_reg1           10101010
/eight_conventional/b_reg2           01010101
/eight_conventional/b_reg3           10101010
/eight_conventional/b_reg4           01010101
/eight_conventional/b_reg5           10101010
/eight_conventional/b_reg6           01010101
/eight_conventional/b_reg7           10101010
/eight_conventional/b_reg8           01010101
/eight_conventional/par_product1     00000000
/eight_conventional/par_product2     11001100
/eight_conventional/par_product3     00000000
/eight_conventional/par_product4     11001100
/eight_conventional/par_product5     00000000
/eight_conventional/par_product6     11001100
/eight_conventional/par_product7     00000000
/eight_conventional/par_product8     11001100
```

Fig. 7: Simulation results for conventional 8 bit multiplier



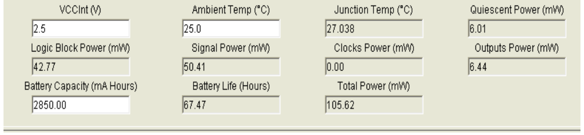
```
/eight_bzfad/clk              1
/eight_bzfad/a                11001100
/eight_bzfad/b                10101010
/eight_bzfad/mul_out          1000011101111
/eight_bzfad/ring_counter     00000010
/eight_bzfad/par_product1     00000000
/eight_bzfad/par_product2     11001100
/eight_bzfad/par_product3     00000000
/eight_bzfad/par_product4     11001100
/eight_bzfad/par_product5     00000000
/eight_bzfad/par_product6     11001100
/eight_bzfad/par_product7     00000000
/eight_bzfad/par_product8     11001100
/eight_bzfad/first_shift      110011000
/eight_bzfad/second_shift     0000000000
/eight_bzfad/third_shift      11001100000
/eight_bzfad/fourth_shift     000000000000
/eight_bzfad/fifth_shift      1100110000000
/eight_bzfad/sixth_shift      0000000000000
/eight_bzfad/seventh_shift    1100110000000
```

Fig. 8: Simulation results for 8 bit multiplier using BZFAD architecture



VCCInt (V)	Ambient Temp (°C)	Junction Temp (°C)	Quiescent Power (mW)
2.5	25.0	27.916	6.01
Logic Block Power (mW)	Signal Power (mW)	Clocks Power (mW)	Outputs Power (mW)
49.56	89.12	0.00	6.44
Battery Capacity (mA Hours)	Battery Life (Hours)	Total Power (mW)	
2850.00	47.16	151.11	

Fig. 9: Power report for conventional multiplier



VCCInt (V)	Ambient Temp (°C)	Junction Temp (°C)	Quiescent Power (mW)
2.5	25.0	27.038	6.01
Logic Block Power (mW)	Signal Power (mW)	Clocks Power (mW)	Outputs Power (mW)
42.77	50.41	0.00	6.44
Battery Capacity (mA Hours)	Battery Life (Hours)	Total Power (mW)	
2850.00	67.47	105.62	

Fig. 10: Power report for BZFAD multiplier

Target Device: x2s100
Target Package: fg456
Target Speed: -5
Mapper Version: spartan2 -- $Revision: 1.58 $
Mapped Date: Mon Dec 21 17:22:28 2009
Design Summary
Number of errors: 0
Number of warnings: 0
Number of Slices: 344 out of 1, 200 28%

Table 1: Comparison between two multipliers

Multiplier type	Conventional shift and add multiplier	BZFAD architecture
Vendor	Xilinx	Xilinx
Device and family	Spartan 2	Spartan 2
Estimated area	662 slices	344 slices
Power dissipation	151.11 mW	105.62 mW

Delay report for conventional shift and add multiplier is given below:

Timing Summary:
Minimum period: 52.071ns (Maximum Frequency: 19.205 MHz

Delay report for BZFAD multiplier is given below:

Timing Summary:
Minimum period: 48.812ns (Maximum Frequency: 20.487 MHz)

DISCUSSION

Based on the reports generated at the time of synthesis process, a comparison is made between two multipliers and it is shown in Table 1.

CONCLUSION

The proposed BZFAD architecture lowers the power dissipation and area when compared to a conventional shift and add multiplier. 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 BZFAD architecture makes use of a low power ring counter. The design can be verified using Model-Sim with VHDL code and Xilinx tool for synthesis process. From these results, BZFAD can attain considerable power reduction and area saving when compared to the conventional shift and add multipliers.

REFERENCES

Chandrakasan, A.P., S. Sheng and R.W. Brodersen, 1992. Low power CMOS digital design. IEEE J. Solid State Circ., 27: 73-484.

Chen, K.H. and Y.S. Chu, 2007. A low power multiplier with spurious power suppression technique. IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 15: 846-850.

Chen, O.T.C., S. Wang and Y.W. Wu, 2003. Minimization of switching activities of partial products for designing low power multipliers. IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 11: 418-433.

Chen, K.H., K.C. Chao, J.S. Wang, Y.S. Chu and J.I. Guo, 2005. An efficient Spurious Power Suppression Technique (SPST) and its applications on MPEG-4 AVC/H.264 transform coding design. Proceeding of the 2005 International Symposium on Low Power Electronics and Design, Aug. 8-10, IEEE Xplore Press, USA., pp: 155-160.

Chen, K.H., Y.M. Chen and Y.S. Chu, 2006. A versatile multimedia functional unit design using the spurious power suppression technique. Proceeding of the IEEE Asian Solid State Circuits Conference, Nov. 13-15, IEEE Xplore Press, Hangzhou, pp: 111-114.

Huang, Z. and M.D. Ercegovac, 2005. High performance low power left to right array multiplier design. IEEE Trans. Comput., 54: 272-283.

Mottaghi, M.D., A. Afzali-Kusha, and M. Pedram, 2009. BZ-FAD: A low power low area multiplier based on shift-add-architecture. IEEE Trans. VLSI Syst., 17: 302-306.

Nelson, V.P. and H.T. Nagle, 1995. Digital Logic Circuit Analysis and Design. 2nd Edn., Prentice Hall, Englewood Cliffs, NJ., ISBN: 0134638948, pp: 842.

Shen, N.Y. and O.T.C. Chen, 2002. Low power multipliers by minimizing switching activities of partial products. Proc. IEEE Int. Symp. Circ. Syst., 4: 93-96.

Wang, J.S., C.N. Kuo and T.H. Yang, 2004. Low power fixed width array multipliers. Proceeding of the IEEE Symposium Low Power Electron and Design, Aug. 9-11, ACM Press, Newport Beach, California, USA., pp: 307-312.