Low power Multiplier Design using bypassing and feeder technique

[1]M. Ravindra Kumar Reddy, [2]K.Sudhakar
Project Student, Dept of ECE, Jaya Prakashnarayan College of Engineering, Mahabubnagar, India.
Associate Professor, Dept of ECE, Jaya Prakashnarayan College of Engineering, Mahabubnagar, India.

***Abstract*— The role of multiplier circuitry in real time applications such as microprocessor and digital processing is very prime. The power handling capacity of a multiplier is a key parameter in designing using VLSI technology because the consumption and dissipation of power is a factor which also effects the performance of multiplier. In present paper a technique called bypassing is implemented in conventional shift-and-add multiplier architecture. This technique eliminates unnecessary switching activities responsible for power consumption and also eliminates unnecessary iteration done in the conventional multiplier whenever a zero is encountered in the multiplication process.**

***IndexTerms*— lowpower, multiplier, shift-and-add, ring counter, feeder.**

I. INTRODUCTION

In a complex or portable applications, the power consumption by individual elements as well as power dissipation of elements playes important role on the performance of complete system. To have a beter performance every element in a circuitry must have lower levels of power consumption as well as dissipation. In present paper we considered the most fundamental component in every digital system, that is mmultiplier. To any multiplier power dissipation and speed are of prime concern. Most common and the best ways to reduce the dynamic power dissipation is to minimize the total switching activity, i.e., the total number of signal transitions of the system. The evaluation research in this field has brought some solution but the problem of nightmare is still remain.

Conventional shift and add multiplier

In order to convey the concept more effectively we considered to be began with conventional multiplier architecture. A Figure 1. is showing the architecture. The regions in the architecture are highlighted with ovals of dashed nature are playing major role in switching activities. The adder operation here is mainly performed in two stages but in single cycle. During the first stage shifting operation is performed where the bits in register B are getting out of it. In second stage the adder operation is performed. where the outcome bits of register B are attached to a multiplexer select pin, here represented as mux_A. As there is any change observed at select signal, the mux_A output is also going to change. The counter is responsible for checking the number of operation which are required generally for smoothening of operations are performed or not. The major activities in switching 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

In this process we found there is redundancy. If we could eliminate, there is possible to save some power and space in architecture and this facilitates to develop low power architecture.

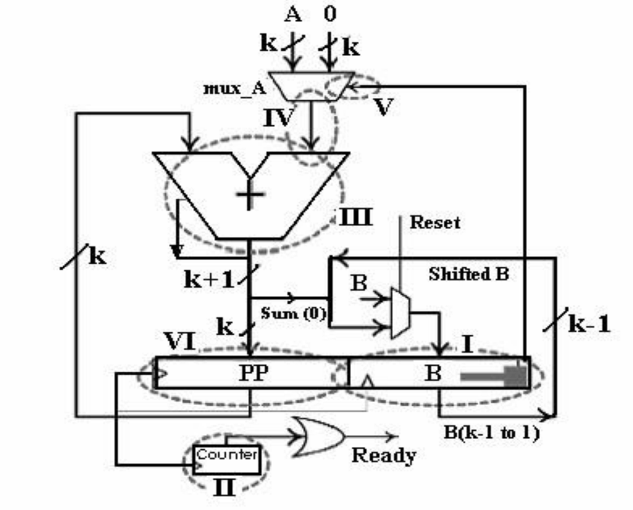
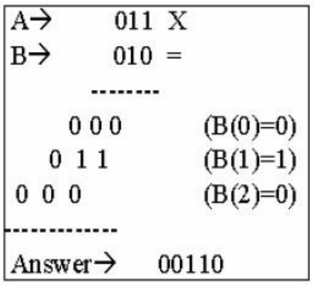


Fig. 1. Architecture of conventional shift and add multiplier with major source of switching activity

A. Shift of the B Register

An example of shifting of register is shown here



```
A→        011  X
B→        010  =
        --------
       0 0 0          (B(0)=0)
     0 1 1            (B(1)=1)
 0  0  0              (B(2)=0)
------------
Answer→    00110
```

Fig. 2. Shift and add multiplication example

In the traditional architecture (see Figure 2), to generate the partial product, *B(0)* is used to decide between *A* and 0. If the bit is ‘1’, *A* should be added to the previous partial product, whereas if it is ‘0’, no addition operation is needed to generate the partial product. Hence, in each cycle, register *B* should be shifted to the right so that its right bit appears at *B(0)*; this operation gives rise to some switching activity.

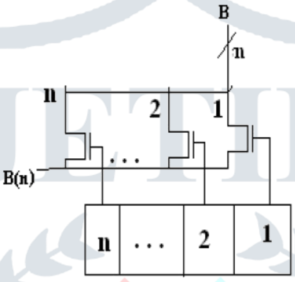


Fig. 3. Multiplier with ring counter

For a 3 bit multiplier 3 bit ring counter is used. Table 1 gives the required bit and counter output Combination

TABLE I. COUNTER OUTPUT WITH REQUIRED BIT

COUNTER OUTPUT	REQUIRED BIT
001	B(0)
010	B(1)
100	B(2)

II. THE PROPOSED LOW POWER MULTIPLIER: BZFAD

In order to obtain low power architecture, we concentrate our effort on eliminating or reducing the sources of the switching activity discussed in the previous section.

The architecture of BZ-FAD is proposed in this paper. The multiplexer (*M1*) used here uses one-hot encoded bus selector chooses a hot bit of *B* in each cycle. A ring counter of 32 bit is used here, it performs multiple tasks, one is selection of *B(n)* in *n*th cycle and second is blocking of *M2*. In order to reduce the power handling capacity of counter we utilized a low power ring counters. which is shown in Figure 4.

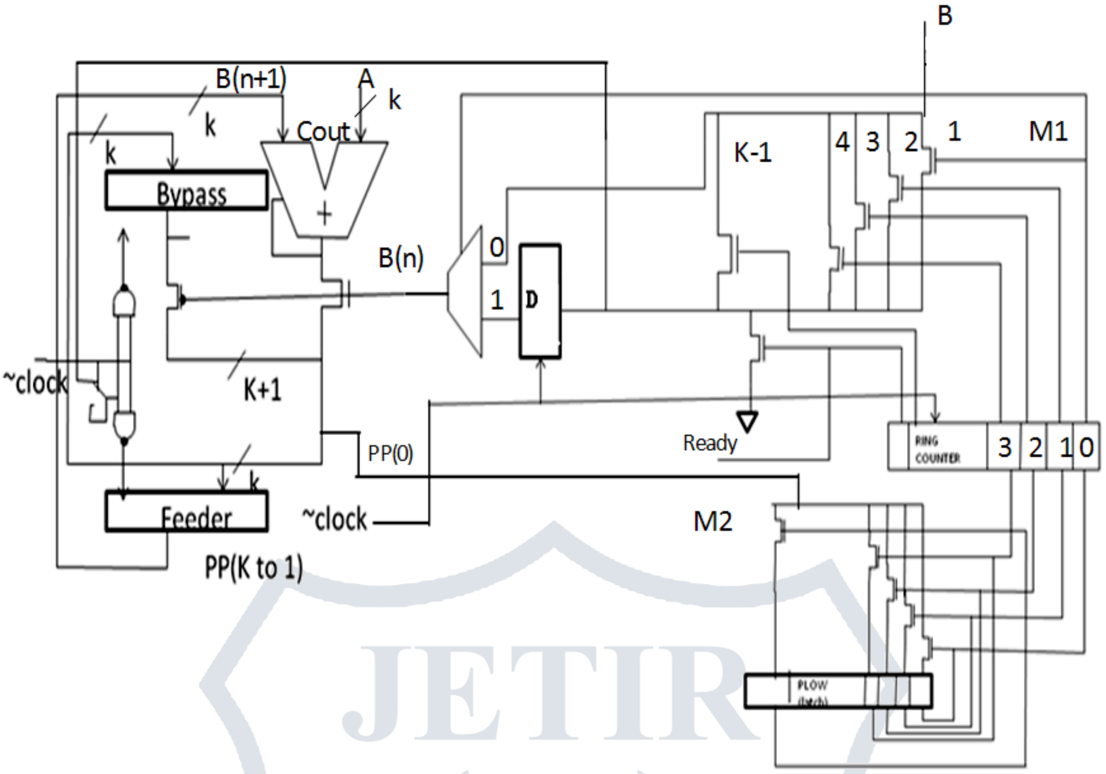


Fig. 4. BZFAD Architecture

III.SHIFTING OF MULTIPLIER REGISTER

A ring counter is used in proposed that eliminates right shifting operation as well as shifting dynamic power dissipation. The wider ring counter in BZFAD architecture helps to raise more transitions than its binary counterpart in the conventional architecture. The BZFAD architecture uses a multiplexer with a one hot encoded bus selector choosing the required bit of 'B' in each cycle. Consider a four bit multiplier 'B'. The shift and add multiplication requires the subsequent bits of 'B' starting from the LSB in each cycle. For getting the subsequent multiplier bits shifting was performed in conventional architecture. This action causes switching activities to occur. The BZFAD architecture makes use of a ring counter to obtain the required multiplier bit without multiplier shifting and thereby causes power saving. A multiplexer with a one hot encoded bus selector is used for getting the required multiplier bit without shifting. The ring counter used in the BZFAD architecture is considerably wider than its binary counterpart used in the conventional architecture. So a low power architecture is used for the ring counter.

Operation In the adder with register

Adder operation is optimized using registers as shown in Figure 5. Feeder and Bypass registers are for optimizing the adder operation.

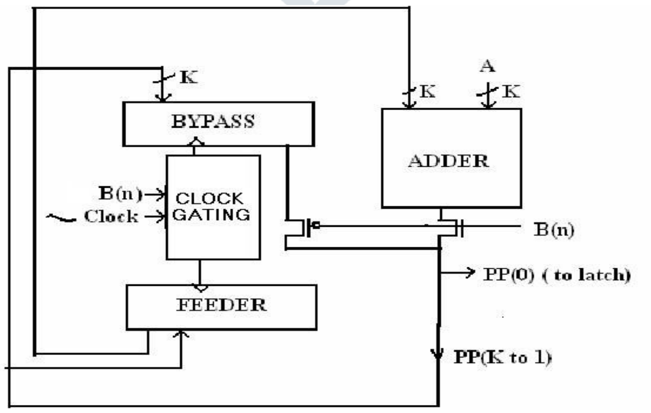


Fig. 5. Adder with Registers

In the conventional architecture if LSB of 'B' is equal to zero, then the current partial product is added to zero and if LSB of 'B' is 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.

Feeder and bypass registers

Feeder and bypass registers are used to optimize the adder operation. In every cycle, the partial product obtained in the previous cycle is available at the input of these two registers. A clock gating structure is given to feeder and bypass registers. The clock gator performs the following functions. Feeder is clocked if LSB of ‘B’ obtained by using the low power ring counter is equal to ‘1’. Bypass is clocked if LSB of ‘B’ obtained by using the low power ring counter is equal to ‘0’. Thus the current partial product is stored either in feeder or in bypass register.

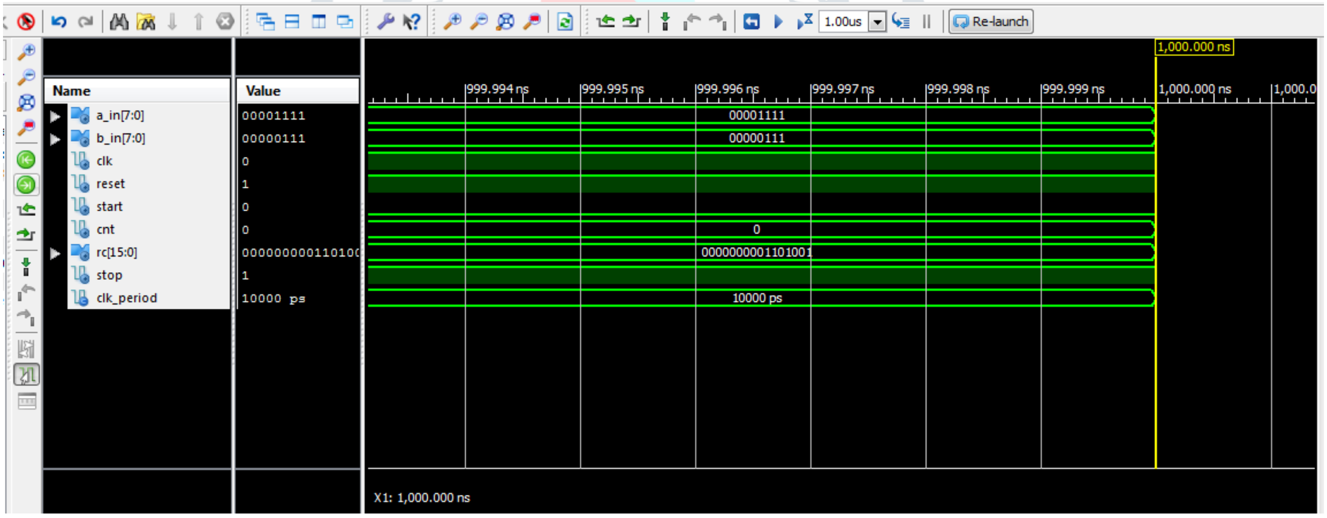
NAND and NOR gates are used to clock feeder and bypass registers. If feeder is clocked, then only the adder operation is performed. If bypass register is clocked, it means that bit obtained from the multiplier by using the low power ring counter is zero, and no addition operation is required. So feeder performs only shifting. The reduction of switching activity in adder is mainly due to the following reasons. The right input of adder is ‘A’, which is constant during multiplication. This enables to remove the multiplexer and feeder ‘A’ directly to the adder, resulting in a noticeable power saving. In each cycle when the LSB of multiplier obtained by using the ring counter is zero, the 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.

IV.SIMULATION RESULTS

Both the architectures , conventional shift and add multiplier and Bypass zero feed a direct multiplier are implemented in VHDL and are simulated using Xilinx 13.1 in Isim simulator and the results obtained are shown in following sections.

A. *Conventional Multiplier output*

The simulation result for 8 bit multiplier using conventional architecture is highlighted in Figure 5.1. The inputs given are 00001111(15) and 000000111(7). Multiplier is shifted in each cycle and the corresponding partial products are formed. The conventional multiplier uses a binary counter. From the simulation results, it can be seen that the output of multiplier comes out to be 0000000001101001(105).



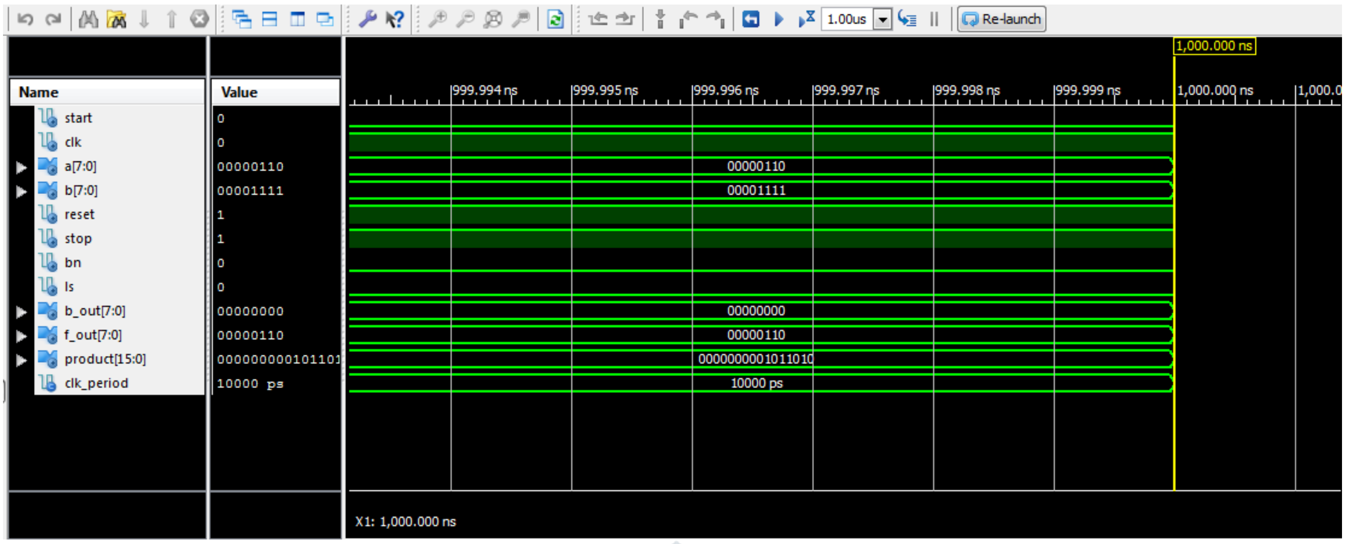
Name	Value
a_in[7:0]	00001111
b_in[7:0]	00000111
clk	0
reset	1
start	0
cnt	0
rc[15:0]	000000000110100
stop	1
clk_period	10000 ps

X1: 1,000.000 ns

Fig 5.1 Simulation results of conventional multiplier

B. *BZFAD multiplier output*

The simulation result for 8 bit multiplier using low power architecture is shown in Figure 5.2. The inputs given are 00001111(15) and 00000110(6).The multiplication operation is performed using the BZFAD architecture. The low power multiplier uses a low power ring counter. From the simulation results, it can be seen that the output comes out to be 0000000001011010(90)



Name	Value
start	0
clk	0
a[7:0]	00000110
b[7:0]	00001111
reset	1
stop	1
bn	0
ls	0
b_out[7:0]	00000000
f_out[7:0]	00000110
product[15:0]	000000000101101
clk_period	10000 ps

X1: 1,000.000 ns

Fig 5.2 Simulation results of BZFAD multiplier

C. Device utilization summary

TABLE II. DEVICE UTILIZATION SUMMARY OF CONVENTIONAL MULTIPLIER

Device Utilization Summary(estimated values)			
Logic Utilization	Used	Available	Utilization
Number of slices	81	960	8%
Number of slice flip flops	88	1920	4%
Number of 4 input LUTs	123	1920	6%
Number of bonded IOBs	54	83	65%
Number of GCLKs	1	24	4%

TABLE III. Device utilization summary of BZFAD multiplier

Device utilization summary			
Logic utilization	Used	Available	Utilization
Number of slice flip flops	55	1920	2%
Number of 4 input LUTs	91	1920	4%
Logic Distribution			
Number of occupied slices	64	960	6%
Numder of slices containing related logic	64	64	100%
Number of slices containing unrelated logic	0	64	0%
Total Number of 4 input LUTs	123	1920	6%
Number used as logic	91		
Number used as a route-thru	32		
Number of bonded IOBs	68	66	103%
IOB flip flops	8		

Number of BUFGMUXs	1	24	4%

D. Timing summary

1). Conventional multiplier minimum period= 6.599 ns

2). BZFAD multiplier minimum period= 6.472 ns.

TABLE IV. SUMMARY OF SIMULATION RESULTS

Multiplier type	Conventional	BZFAD
Vendor	Xilinx	Xilinx
Device and family	Spartan 3E	Spartan 3E
Power dissipation	34mW	27mW

V. CONCLUSION

In this project, sources of power dissipation in VLSI circuits are studied and methods to reduce power dissipation are explained. Switching activity is found to be active source of power dissipation in conventional shift and add architecture. A low power architecture known as Bypass Zero Feed A Direct (BZFAD) architecture is proposed. Both the architectures are implemented using VHDL in Xilinx and results thus obtained are analyzed.

The proposed architecture occupies lesser area compared to that of the conventional multiplier but it takes more time to produce the output, hence the proposed multiplier can be used in applications where speed is not a primary concern. The power dissipated in the modified circuit is reduced by 20% compared to conventional multiplier architecture.

VI. Future Scope

The project can be implemented on CADENCE or SYNOPSIS tool to obtain more reliable power calculations. Multiplier architectures other than shift and add multiplier can be implemented, analyzed and compared with one another to select an appropriate architecture for our purpose. Apart from switching activity other sources of power dissipation can be considered and addressed.

REFERENCES

[1] Ercegovac M.D. and Huang Z. (March 2006) “High performance low power left to right array multiplier design” IEEE Trans. Comput., Vol-54, no-2, pp 272-283.

[2] C. N.Marimuthu, Dr. P. Thangaraj, Aswathy Ramesan “Low power shift and add multiplier design” International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010.

[3] Prof Prasann D.Kulkarni, Prof.S.P.Deshpande, Dr.G.R.Udupi” low power add and shift multiplier design Bzfad architecture” International Journal of Computer and Electronics Research [Volume 2, Issue 2, April 2013]

[4] 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.

[5] Peiyi Zhao and Zhongfeng Wang,”Low Power design of VLSI circuits and systems” .ASIC,2009. ASICON “09. IEEE 8th International Conference

[6] A.Chandrakasan and R. Brodersen, “Low Power CMOS Digital Design”, IEEE J. Solid State Circuits, Vol.27, no.4, pp 473-484, Apr 1992

[7] M. Mottaghi-Dastjerdi, A. Afzali-Kusha, and M. Pedram,”BZ-FAD :A low power low area Multiplier based on Shift and Add architecture” IEEE transactions on very large scale integration(VLSI) systems, Vol.17, No.2, February 2009

[8] 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

[9] Zamin Ali Khana ,S. M. Aqil Burneyb, , Jawed Naseemc, Kashif Rizwand, “Optimization of Power Consumption in VLSI Circuit”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011

[10] 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.

[11] V. P. Nelson, H. T. Nagle, B. D. Carroll, and J. I. David,”Digital Logic Circuit Analysis & Design.” Englewood Cliffs, NJ: Prentice-Hall, 1996.