Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 2, February -2015

LOW POWER MULTIPLIER USING BYPASSINGZERO ARCHITECTURE

Nishant Govindrao Pandharpurkar[1]and Antriksh sharma.[2]

[1]*Deparment of SENSE, VITUniversity, nishant.govindrao2013@vit.ac.in*
[2]*Department of SENSE, VIT University,antriksh.sharma2013@vit.ac.in*

Abstract- *This paper presents the implementation of multipliers with Bypassing the Zero architecture based on shift and add multiplication. Using this architecture a considerable amount of switching power can be reduced as compared to conventional shift and add multipliers. For this purpose, synchronous ring counter is used instead of conventional binary counter. Note that Shifting of only most significant partial products has been done. Simulation results shows that this architecture lowers switching cases and lowers power consumption up to 30%-40%.*

Keywords-*synchronous low power ripple counter, Bypass and Feeder register, Bypassing zero procedure, pmos array*

I. INTRODUCTION

Each and every computing system demands multiplication as their fundamental function. In digital CMOS design, multiplication serves as a basis for complex process. Among the multiplication algorithm, add and shift multiplication is the simplest. But it involvesa considerable switching activity which indirectly affects the power consumption and also a considerable time factor has to be given for the completion of shiftand addalgorithm. To reduce the dynamic power consumption, switching activities must be reduced. The above said phenomenon forms the basis of this research paper which proposes an architecture involving the bypassing adder circuit while calculating the intermediate partialproducts. Certain modification involves use of ringcounter, shifting the multiplier bits without using shift register and use of ripple carry adder.Among the various multipliers, tree multipliers is very popular for fast multiplication but it involves large area and also highest power consumption .Wallace multipliers are famous for fastest algorithm but it is complex as compared to other multipliers.CSA based radix multipliers employs a considerable number of active transistors which invariably resulting in greater power consumption. High Radix multipliers are also affected with same complication as number of multiplier and multiplicand bits increases. Inference from this analysis is that lower switching activities and number of intermediate partial products, dynamic power consumption can be reduced. Add and shift algorithm, being one of the simplest algorithm, certain genuine modifications will result in considerablereduction in power consumption. All modifications will be discussedin section III in this paper. All simulated results are obtained using 4x4 multiplication.

II. CONVENTIONAL ADD AND SHIFT MULTIPLICATION

An easy way to comply with the binary multiplication is using add and shift multiplication.Fig.1 depicts the flow chart of shift and add multiplication with multiplier Y and multiplicand X.

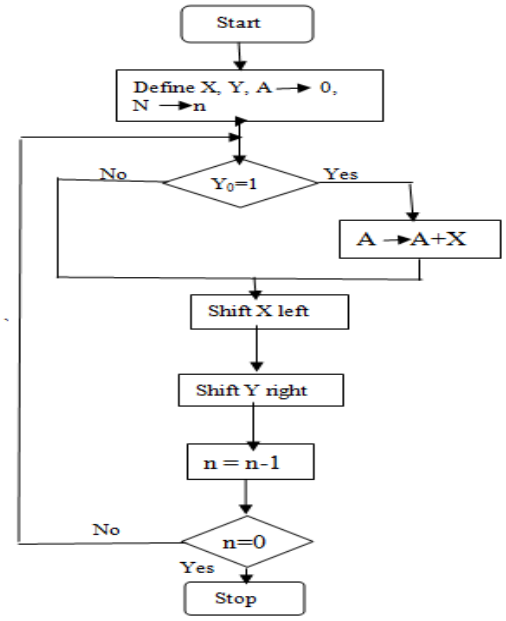

Fig.1 Flow Chart of add and shift algorithm

Basic add and shift multiplication employs last bit of multiplier to decide whether X(multiplicand) is added to zero or A(partial product) at each cycle, which results in unnecessarytransition when Y_0=0.Binary multiplication which involves Add and Shift procedure contains a regular shift register for shifting each bit of multiplier in each cycle. Basicbinary parallel in serial out shift register can be employed for this purpose. Algorithm flow is based on taking each bit of multiplier from right side to left side and then doing multiplication by multiplicand with each bit of multiplier and later placing the intermediate partial products appropriate positions from left to right.Fig.2 shows the circuit implementation of this algorithm. To achieve the final multiplication using shift and add algorithm, significant number of switching activities have been taken place within the circuit.

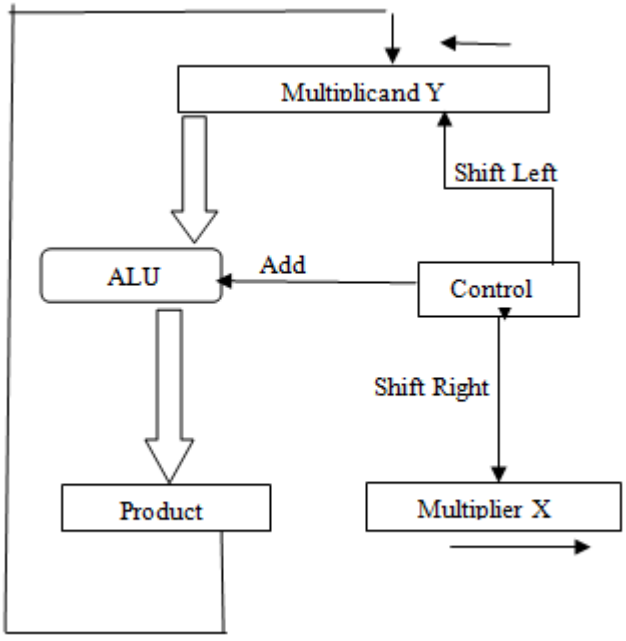

Fig.2 Circuit implementation of add and shift multiplication

2.1 Sources of switching activities

The switching activities involved are shift of multiplier register,switching between 0 and X, Activity in counter,Activity in adder, Activity of mux select signal are controlled by Y_0andContinuous feeding of partial product to adder total power consumption in any digitalcircuitry is defined as the algebraic sum of static, dynamic, switching, SC

power and small amount of leakage power. Dynamic power refers to the power consumed during charging and discharging of capacitances. Static power refers to the static path formed between supply and ground.SC power constitutes the power due to metallic conduction between two blocks. Mathematically,

$$P_{switching} = \alpha C V_{dd}^2 f_{clk} \_\_\_\_\_ (1)$$

Among all, switching power amounts for major share because it generates while actual transition of bits take place. It is impossible to nullify all these power heads, but it can be practically made lower either by employing modification in existing algorithm in constructive way or by employing certain low switching digital blocks in a circuitry. This paper emphasizeson both these aspects, which will be discussed in following sections.

2.2 Reducing Switching Activities

The advantage of using shift and add multiplication is that the algorithm is simple and easy to implement using basic digital blocks i.e. shift register and multiplexer. But as it involves various transitions and considerable power consumption. Slight constructive modification will be beneficial for lowering power and switching activities. Among these, switching activity during selection between 0 and A can be reduced by removing multiplexer and switching activity due to counter can be reduced by using synchronous ring counter.

Bypassing zero while multiplication can be achieved by neglecting adder stage when multiplier bit of Y_0 is 1 and partial product is shifted to right by one bit at each cycle. Also we define two special purpose registers Bypass and Feeder register which are helpful while bypassing is needed. Bypass register and Feeder register are clocked alternately according to shifted bit of multiplier at each stage.

III. TOWARDS LOW POWER MULTIPLICATION

This paper proposes the bypassing zero architecture for radix 4 multiplication using basic add and shift multiplication algorithm. Bypassing zero is needed when Y_0 bit of multiplier is 1.Also adder can be bypassed in this case to avoid unnecessary transition in calculation of partial products. Fig.3 shows block diagram of Bypassing Zero Architecture.

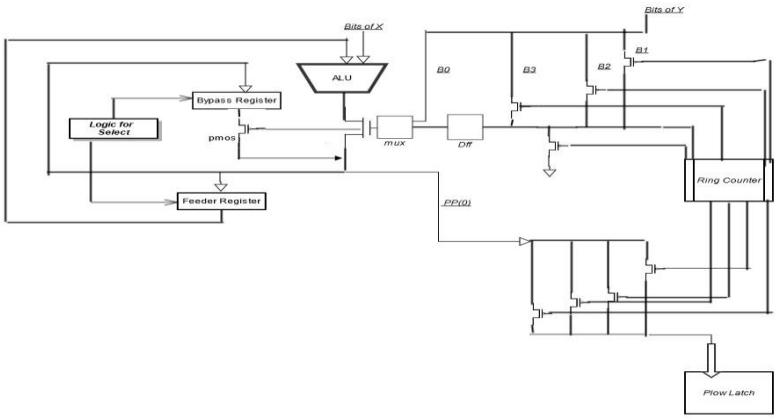

Fig.3 Multiplication using Bypassing Zero Architecture

Bypassing zero procedure gives optimised results in terms of power as well as area. Two special purpose registers ,Feeder and Bypass registers, have been introduced for getting the partial products in each cycle, depending on shifted bit of multiplier. Modifications in circuitry level involves direct feeding of multiplicand to the adder circuit, bypassing the adder circuit, use of ring counter instead of binary counter. Bypass and Feeder register has been employed to store the current partial product and these registers are clocked depending on coming bit of Y i.e. Y(n).In each cycle, product either comes from bypass or adder circuit. Ripple carry adder has been used because it has lowest average transition among other adder circuits like carry look ahead adder, carry save adder etc.

An array of pmos is connected to bypass register and array of nmos is connected to feeder register so that in each cycle, either bypass will be clocked or feeder will be clocked. Counter has been employed so that it performs dual purpose of counting and also shifting the multiplier bit. Lower half of partial product is obtained through ring counter and PP (0), last bit of current partial product. The total product is given by 4 bits of bypass and 4 bits of P_{low}. The main advantage of using Bypassing zero architecture is that multiplication can be able to implement without using complex digital blocks. Also simple algorithm of add and shift is used making it customised.

3.1 Shifting multiplier bits using ring counter

In generating intermediate partial products, it is necessary to shift multiplier bits at each cycle i.e.Y0, Y1,Y2 and so on. In Bypassing procedure, multiplier bits are shifted with thehelp of ring counter as shown in the Fig 3. In this way, multiplier bits are produced without actual shifting of multiplier bits. Hence considerable amount of power can be reduced. This is shown with an example in table I.Fig.4 shows the shifting mechanism using ring counter.

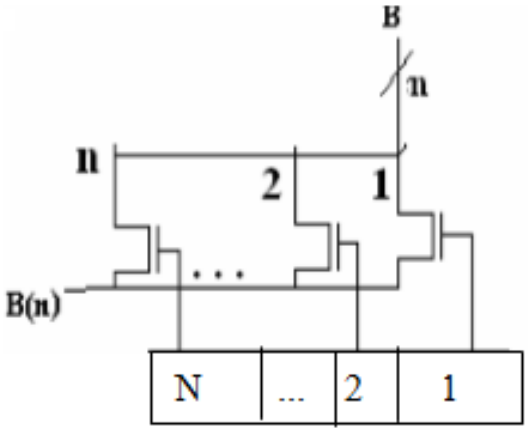

Fig.4 Shiftingusing ring counter

TABLE IEXAMPLE OF SHIFTING OF BITS USING RING COUNTER

Counter output	Required shifted bit
001	Y(0)
010	Y(1)
100	Y(2)

3.2 Special purpose register for reducing switching activityof adder

In bypassing zero architecture, two special purpose register Bypass and Feeder have been employed for reducing the switching activity of adder circuit. In each cycle, multiplier bit is obtained according to which either Bypass or Feeder register is clocked. WhenY (n) is zero, adder is not needed, in other words, adder has to be bypassed. In that case, Bypass register is clocked to get the current partial product or say (n+1) is 1, adder is needed and hence Feeder register is clocked to get current partial product. Selection between Feeder and Bypass register has been done by using ANDgate and invertergate using global clock as shown in Fig.5. Hence in each cycle, partial product is generated either from adder or from bypass register. These partial products get added successively to get bits of final product .These two registers(Fig.6) can be referred as peculiarity of bypassing zero architecture.

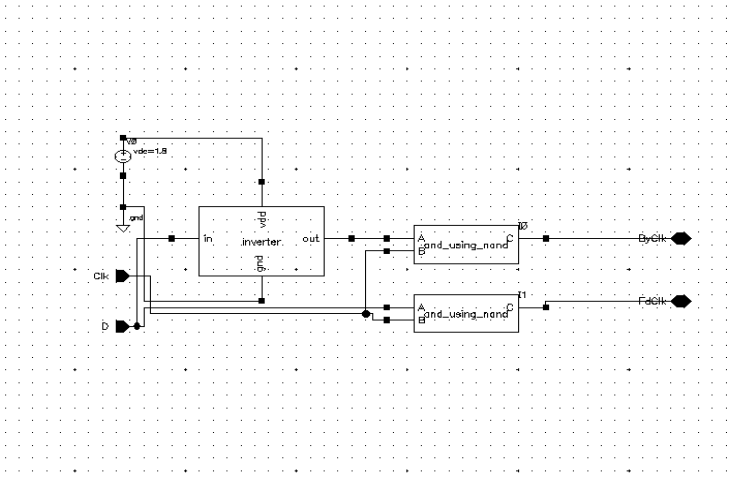

Fig.5Logic circuit for selecting Bypass and Feeder register

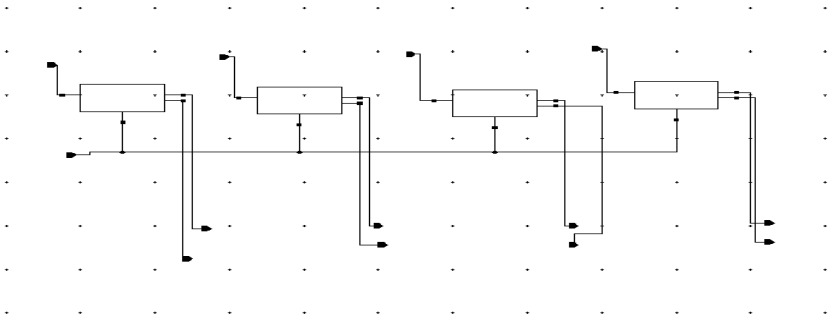

Fig.6Bypass Register using D flip flop

3.3 Shift of partial products

In basic add and shift multipliers, at each cycle partial products are being shifted and added successively with previous partial products. This leads to the considerable amount of power as well as time consumption .To eliminate this, in Bypassing zero architecture, only most significant bits have been shifted and lower 4 bits are stored in P_{low}latches. It is noted that lower 4 bits has been stored in k latches if multiplication employs k bit x kbit

IV. RESULTS AND SIMULATION

Inthis paper, results have been shown for ring counter and multiplier(Fig.7 and Fig.8)circuit. Cadencevirtuoso -64 has been used for simulation purpose on 180 nm technology.

4.1 Ring Counter

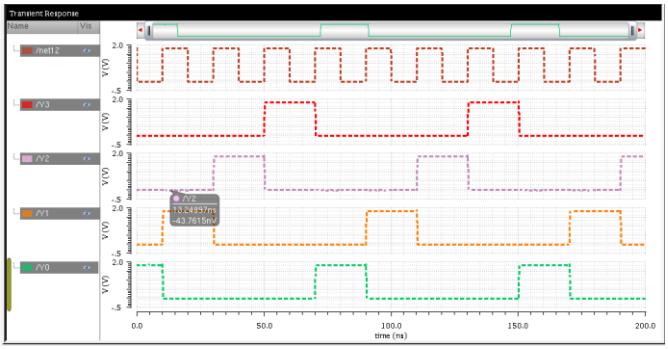

Fig.7 Ring counter

The ring counter is used for dual purposes, viz., counting the steps required for multiplication and also shifting of multiplier bits .Hence power has been reduced up to 30-40% as compared to the same circuit employing binary counter. Also only most significant bits of multiplier has been processed for multiplication, Therefore a considerable switching power is reduced and time required to complete this process is also low. This serves the advantage of using bypassing zero architecture instead of basic multiplier design.

Circuit simulation results on Cadence Virtuoso -64 shows that delay has been reduced up to 15% and two specialpurpose registers Bypass and Feeder is clocked at eachcycle, contributing to less delay and also less switching power. Simulation results shows that as radix number increases the gate delay factor also increases significantly in binary multiplication.

Simulation results shows that bypassing zero architecture can be used when power consumption and gate delay is a major concern.

TABLE II

Comparisons between bypassing architecture and conventional add and shift multiplier

Multiplier	**Power Consumption**	**Delay**
Bypassing Architecture	862.3µW	139ns
Conventional Add and Shift Multiplier	634.04µW	142ps

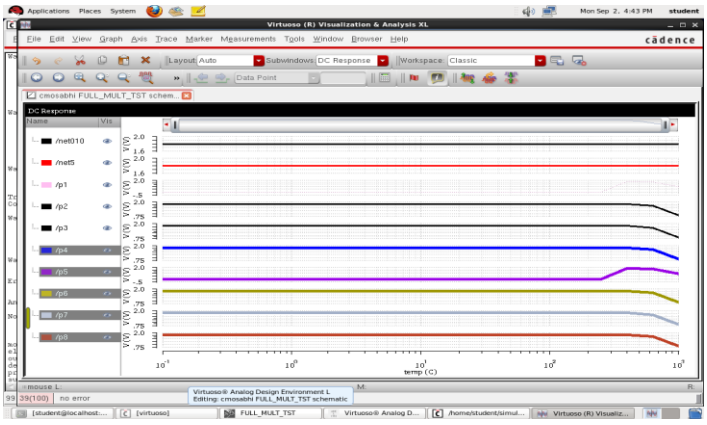

Fig.8 Multiplication of 1111 with 1111

CONCLUSIONS

Table II shows that power consumption of Bypassing multiplier has been reduced to 30-40% in comparison to conventional add and shift multiplier.Bypassing Architecture has advantages of minimum switching activities being

encountered but also employed simple algorithm. Besides this, use of global clock makes this structure customised. Compared to other multiplier architecture, delay factor also reduces to significant amount.

REFERENCES

[1] M. Mottaghi-Dastjerdi, A.Afzali-Kusha, and M. Pedram. " BZ-FAD: A Low-Power Low-Area Multiplier based onShift-and-Add Architecture"
[2] O.Chen, S.Wang, and Y.W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 3, pp. 418–433, Jun. 2003.
[3] Levin Hood, "Structure of Computer Systems",Oct1999
[4] Kuan-Hung Chen and Yuan-Sun Chu, "A Low-Power Multiplier With
the Spurious Power Suppression Technique," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 15, no. 7, July 2007, pp. 846-850.
[5] International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010
[6]Anahita Naghilou, " Hot Block Ring Counter: A Low Power Synchronous Ring Counter",2009

