

# Design of Baugh Wooley Multiplier with Adaptive Hold Logic

M.Kavia, V.Meenakshi

**Abstract** — Mostly, the overall performance of the arithmetic functional units depends on the throughput of the digital multiplier. In order to increase the performance, clock period has to be minimized. Baugh Wooley Multiplier is one of the high performance parallel multiplier but its propagation delay is high compared to other arithmetic units in the system. So variable latency technique is implemented using adaptive hold logic, it will analyze the inputs and determine the number of clock cycle required for each input pair. At normal condition multiplier output will be taken in one clock cycle, depend upon the complexity of input, number of clock period is increased and for worst case two clock cycles is used for multiplier. By means of this technique the performance of the system will be increased depend upon the clock frequency used.

**Index Terms**— Adaptive Hold Logic (AHL), variable latency, Baugh Wooley Multiplier (BWM), clock cycles, propagation delay, clock period

## 1 INTRODUCTION

In today's digital signal processing and various other applications multipliers plays a major role. With recent advancement in technology, many researchers are working to design multipliers which offer either of the following design targets - high speed, low power consumption and less area. Baugh wooley multiplier has the advantage of reducing the number of partial products. Baugh wooley multiplier is used where hardware cost is a major concern. Baugh wooley multiplier has to be designed with variable latency technique to operate for both worst case and best case conditions.

## 2 VARIABLE LATENCY DESIGN

Variable latency technique is used to increase the throughput of the design. The variable latency technique divides the circuit into two parts i) shorter paths and ii) longer paths. Shorter paths can execute correctly in one cycle whereas longer paths needs two cycles to execute. The telescopic units is one of the existing realizations of the variable latency design style [5]. The design of the hold logic in telescopic units has an impact on circuits throughput. The hold logic which was designed traditionally may be inaccurate. To obtain more accurate hold logic and improve the efficiency of telescopic units Shortest Path Activation Function (SPAF) algorithm is used. Non exact hold logic methodology is used to reduce the overhead of large circuits.

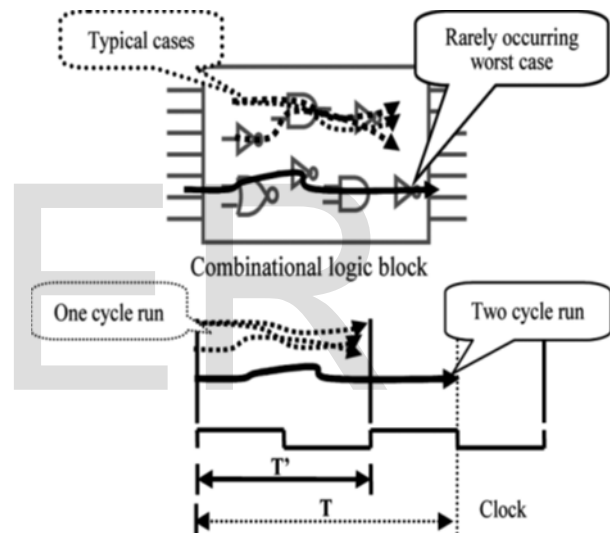


Fig.2.1 Concept of variable latency design

## 3 VARIABLE LATENCY ADDER

Carry Select Adder using variable latency technique called a variable latency adder [6]. The variable latency technique allows the adder to operate at low supply voltage when compared to conventional adders while maintaining the same throughput. While processing different operands, the variable latency adder concept exploits the differences in latency required by an adder. When short latency operation occurs the adder output will be obtained at single clock cycle. When long latency operation occurs the adder output will be obtained at two clock cycles. When the variable latency adder concept is applied to 64-bit carry save adder design, more than 40% of energy saving is obtained when compared to conventional adder while the same throughput is maintained.

- M.Kavia, PG Scholar, VLSI Design, Sona College of Technology, Salem, India, PH- 9840881518. E-mail: [kaviang@gmail.com](mailto:kaviang@gmail.com)
- V.Meenakshi, Assistant Professor, Department of ECE, Sona College of Technology, Salem, India, PH-9600516042. E-mail: [meena.vijay27@gmail.com](mailto:meena.vijay27@gmail.com)

#### 4 VARIABLE LATENCY VLIW PROCESSORS

VLIW (Very Long Instruction Word) processors have different functional units, compilers need to schedule operations onto different functional units [4]. All the functional units of same kind and same latency are considered. In such functional units, for scheduling the operation the conventional list scheduling algorithm selects the first free available functional unit. Functional units of same kind may have different latencies, due to the process variation in advanced processing technologies. Functional list scheduling algorithm may not yield good performance, when the functional units have different latencies. So mobility list scheduling algorithm was proposed to schedule the operation on non-uniform latency functional units. Mobility latency scheduling uses mobility information to schedule the operation. The mobility list scheduling achieves 20% performance when compared to CLS.

#### 5 VARIABLE LATENCY PIPELINED MULTIPLIER

The variable latency multiplier architecture combines second order booth algorithm with a split carry save array pipelined organization including multiple row skipping and completion predicting carry select final adder [10]. The variable latency multiplier architecture for both synchronous and asynchronous design can overcome the performance offered by fixed latency multipliers

#### 6 ROW BYPASS MULTIPLIER WITH AHL

The variable latency multiplier architecture for  $m$  bit is shown in the fig.6.1, which includes a  $m$  bit row-bypassing multiplier, normally it will operate at the one clock cycle (minimum 50% of multiplicand is at logic "0" if less than that output of the multiplier is unstable because of less clock period),  $2m$  1-bit Razor flip-flops and an AHL circuit [1].

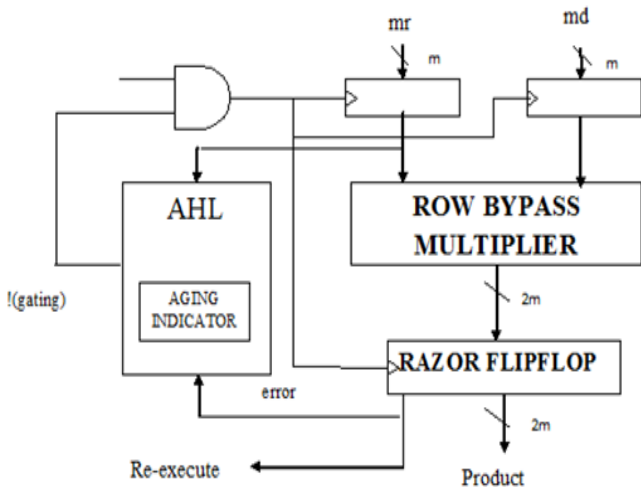


Fig.6.1 Row Bypass Multiplier with AHL

The two inputs are given to the row bypass multiplier. The

inputs are multiplicand and the multiplier. Row bypassing technique is based on number of zeros in the multiplier bits. If the corresponding multiplier bit is 0, the multiplexer selects  $a_i b_j$  as the sum bit and zero as the carry bit. Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives. If errors occur, the Razor flip-flop will set the error signal to 1 to notify the system to re-execute the operation and notify the AHL circuit that an error has occurred. The AHL circuit can decide whether the input patterns require one or two cycles to complete and it holds the inputs of the multiplier.

#### 6.1 Row Bypass Multiplier

By passing with reference to multiplier means turning off some columns or rows or both in the multiplier array whenever certain multiplier or multiplicand or both bits are zero. This Row bypassing technique is based on number of zeros in the multiplier bits. In this multiplier, some of the rows of adders in the basic multiplier array are disabled during operation, to save the power.

For a low-power row-bypassing multiplier, if the bit  $b_j$  in the multiplier is 0, i.e., all partial products,  $a_i b_j$ , are zero; the addition operation in the corresponding row can be bypassed which results in reduction of power. Fig.6.2 shows a  $4 \times 4$  row-bypassing multiplier. When the inputs are  $1011 * 1001$ , the multiplexers in the first row select  $a_i b_0$  as the sum bit and select 0 as the carry bit because  $b_1$  is 0. Then the inputs are bypassed to FAs in the second row. Therefore, no switching activities occur in the first-row FAs because the tristate gates turn off the input paths to the FAs and the power consumption is also reduced. In the same way, no switching activities will occur in the second-row FAs because  $b_2$  is 0. However, the FAs must be active in the third row because the  $b_3$  is not zero.

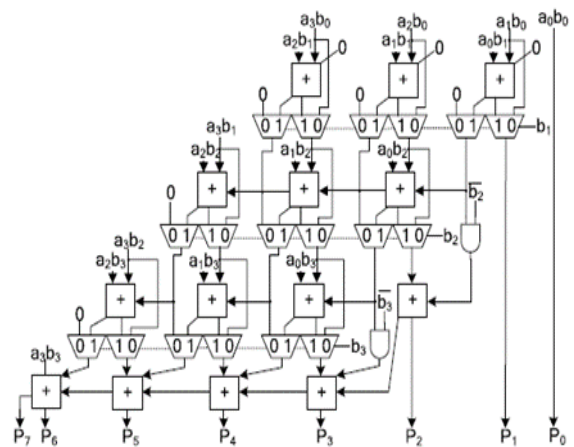


Fig.6.2 Row Bypass Multiplier

#### 6.2 Razor Flipflop

Razor flip-flops are used to detect whether timing violations occur before the next input pattern arrives. A main flip-flop,

shadow latch, XOR gate, and multiplexer are contained in 1-bit Razor flip-flop. Using a normal clock signal, the main flip-flop catches the execution result for the combination circuit and using a delayed clock signal, which is less than the normal clock signal the shadow latch catches the execution result.

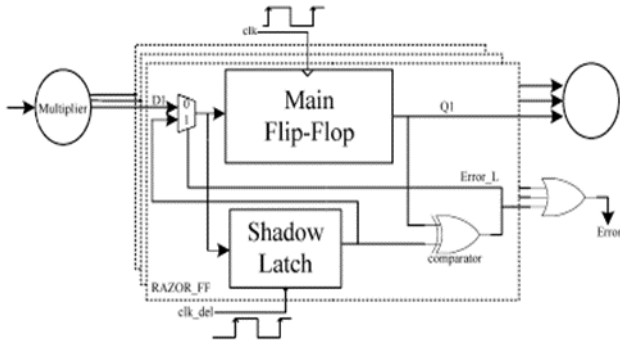


Fig.6.3 Razor Flip Flop

If the latched bit of the shadow latch is different from that of the main flip-flop, this condition occurs only if number of logic "1" more than 50% in multiplicand therefore this operation requires two clock periods, and the main flip-flop catches an incorrect result. Razor flip-flop will set the error signal to notify the system to re-execute the operation continue for two clock cycle by notifying the AHL circuit that an error has occurred.

### 6.3 Adaptive Hold Logic

The AHL circuit is the key component in the variable-latency multiplier. The AHL circuit contains an aging indicator, a mux, and a D flip-flop.

The aging indicator is implemented in a simple counter that counts the number of errors over a certain amount of operations and is reset to zero at the end of those operations. If the cycle period is too short, throw-bypassing multiplier is not able to complete these operations successfully, causing timing violations which will be caught by the Razor flip-flops, which generate error signals. If errors happen repeatedly, the aging indicator will output signal 1; otherwise it will output 0. The multiplexer selects one of its inputs based on the output of the aging indicator. Then an OR operation is performed between the result of the multiplexer, and the Q signal which can determine the input of the D flip-flop. The output of the multiplexer is 1 when the pattern needs single cycle.

The input flip flops will latch new data in the next cycle when the gating signal will become 1. On the other hand, when the output of the multiplexer is 0, which means the input pattern requires two cycles to complete, the OR gate will output 0 to the D flip-flop. Therefore, the gating signal will be 0 to disa

ble the clock signal of the input flip-flops in the next cycle. Note that only a cycle of the input flip-flop will be disabled because the D flip-flop will latch 1 in the next cycle.

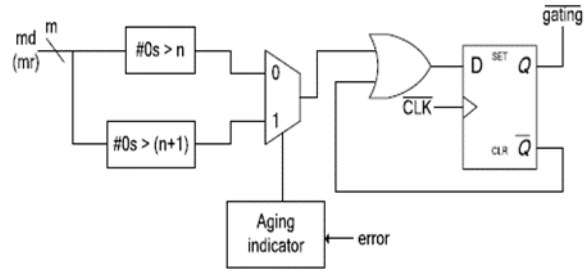
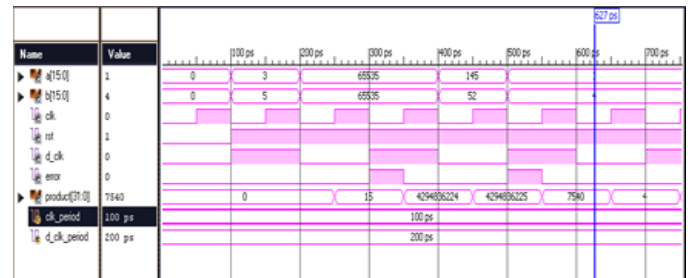


Fig.6.4 Adaptive Hold Logic

## 7 ROW BYPASS MULTIPLIER WITH AHL OUTPUT



## 8 PROPOSED WORK

In the proposed method, variable latency Baugh Wooley Multiplier is designed with Adaptive Hold Logic (AHL). Baugh Wooley Multiplier is a parallel multiplier which uses fewer adders and less iterative steps. This is very important criteria in case of fabrication of chips and high performance system requires components which are as small as possible. Consider the input bits as multiplicand and multiplier to the Baugh Wooley Multiplier. The comparator circuit is used to compare the previous and present input of the multiplier. The comparator circuit produces 32 bit output. The output of the comparator is passed to the error signal generator. The error signal generator produces the error signal as one when the comparator output contains more than 16 1's otherwise it produces error signal as zero. The output of error signal generator is passed to the system clock generator. If error signal is one, the system clock generator produces the multiplier output in two clock cycles else if the error signal is zero, the system clock generator produces the multiplier output in single clock cycle. The proposed circuit not only improves the accurate performance but also reduces the hardware complexity and also less power consumption. The clock period has been reduced by 50% by using variable latency because normally clock period will depend on the maximum operating time.

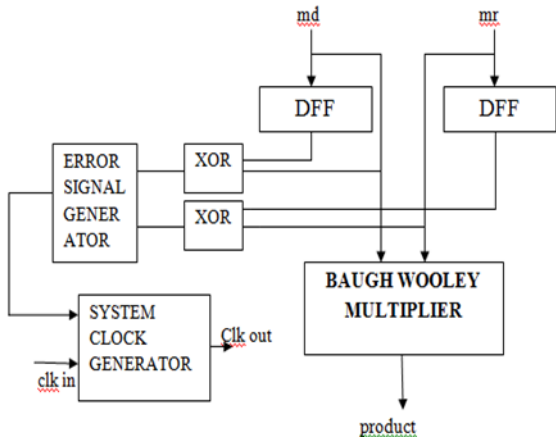


Fig.8.1 Baugh Wooley Multiplier with AHL

By using this technique, for the worst case, two clock cycles are used for execution and at the normal conditions the output will be obtained in one clock cycle

### 8.1 Baugh Wooley Multiplier

Baugh Wooley Multiplier was used for both signed and unsigned multiplication. Baugh Wooley Multiplier is a parallel multiplier which uses fewer adders and lesser amount of iterative steps which occupy lesser space when compared to serial multiplier[3].

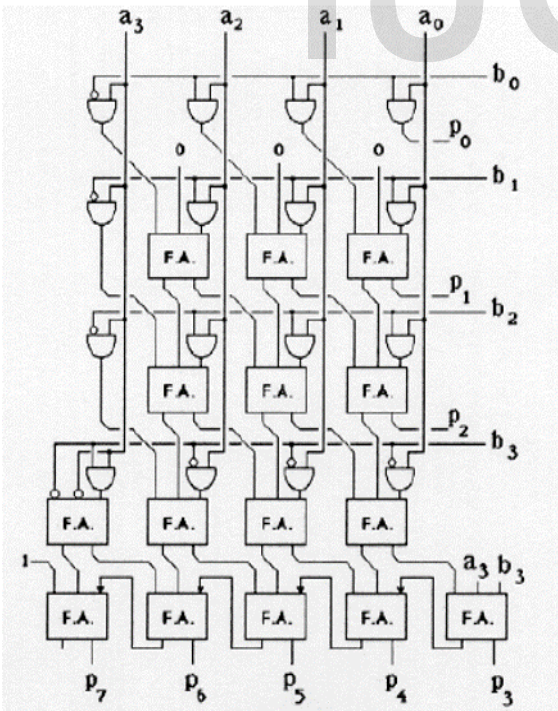
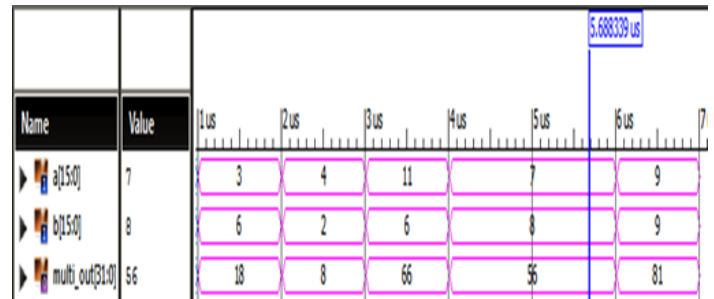


Fig.8.2 Baugh Wooley Multiplier

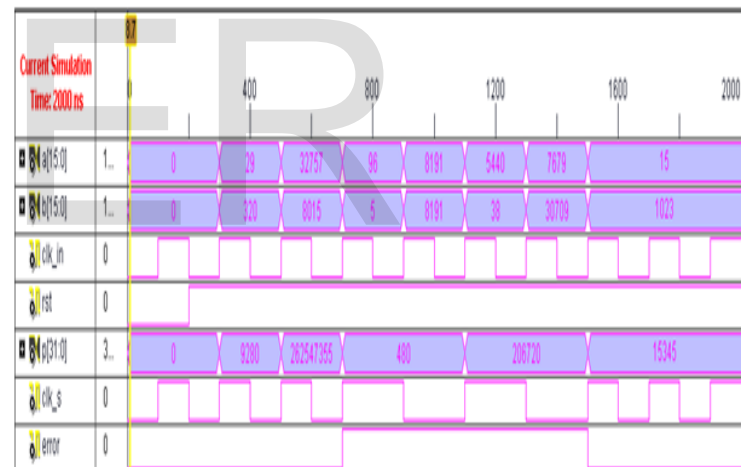
This multiplier is used because it gives the advantage of re-

ducing the number of partial products and it is used where hardware cost is a major concern. Fig 8.2 shows the Baugh Wooley Multiplier design where the first three rows are referred to as PM (partial products with magnitude part) and generated by one NAND and three AND operations. The fourth row is called as PS (partial products with sign bit) and generated by one AND gate and three NAND operations with a sign bit.

### 8.2 Baugh Wooley Multiplier Output



### 9 BAUGH WOOLEY MULTIPLIER WITH AHL OUTPUT



### 10 CONCLUSION

In the existing papers the carry select adder, synchronous and asynchronous pipelined multiplier, functional units of VLIW processors and row bypass multiplier are designed using variable latency technique. In this proposed paper variable latency technique is incorporated with Baugh Wooley Multiplier which enhances the performance of the design. Baugh Wooley Multiplier using variable latency technique results in 50% of area and power reduction when compared to row bypass multiplier using variable latency technique.

## REFERENCES

- [1] J Ing-Chao Lin, Yu-Hung Cho and Yi-Ming Yang, "Aging aware reliable multiplier with Adaptive Hold Logic," IEEE TRANS. VERY LARGE SCALE INTEGR. (VLSI) SYST., 2014.
- [2] Joshin Mathew Joseph, V.Sarada, "Reconfigurable High Performance Baugh Wooley Multiplier for DSP applications", ISSN (PRINT) : 2320 - 8945, vol -1, Issue -4, 2013.
- [3] PramodiniMohanty, RashmiRanjan, "An Efficient Baugh Wooley Architecture for Both Signed & Unsigned Multiplication", INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND ENGINEERING TECHNOLOGY, vol. 3, no. 4, April 2012.
- [4] N. V. Mujadiya, "Instruction scheduling on variable latency functional units of VLIW processors," in PROC. ACM/IEEE ISED, Dec. 2011, pp.307-312.
- [5] Y.-S. Su, D.-C. Wang, S.-C. Chang, and M. Marek-Sadowska, "Performance optimization using variable-latency design style," IEEE TRANS. VERY LARGE SCALE INTEGR. (VLSI) SYST., vol. 19, no. 10, pp. 1874-1883, Oct. 2011.
- [6] Y. Chen *et al.*, "Variable-latency adder (VL-Adder) designs for low power and NBTI tolerance," IEEE TRANS. VERY LARGE SCALE INTEGR. (VLSI) SYST., vol. 18, no. 11, pp. 1621-1624, Nov. 2010.
- [7] D. Baneres, J. Cortadella, and M. Kishinevsky, "Variable-latency design by function speculation," in PROC. DATE, 2009, pp. 1704-1709.
- [8] A. K. Verma, P. Brisk, and P. Jenne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," in PROC. DATE, 2008, pp. 1250-1255.
- [9] Ming-chen Wen, Syng-Jyan Wang, and Yen Nan Lin, "Low Power Parallel Multiplier with Column Bypassing", IEEE, 2005.
- [10] M. Olivieri, "Design of synchronous and asynchronous variable-latency pipelined multipliers," IEEE TRANS. VERY LARGE SCALE INTEGR. (VLSI) SYST., vol. 9, no. 4, pp. 365-376, Aug. 2001.