ISSN 2348 – 9928
IJAICT Volume 6, Issue 7, July 2019 Doi:01.0401/ijaict.2016.11.05 Published on 05 (7) 2019

MULTIPLIER DESIGN USING ADAPTIVE HOLD LOGIC

Mrs. S. Prema ***Assistant Professor,***
Ms. S. Divya ***UG Student,***
Ms. G. Divya Karunya ***UG Student,***
Ms. R. Kokila ***UG Student,***

Electronics and Communication Engineering,
KPR Institute of Engineering and Technology,
Coimbatore, Tamilnadu, India

Abstract – ***Low power consumption and reduced time delay has been an important part in VLSI system design. Digital multipliers are key components of many high performance systems such as FIR filters, microprocessors and digital signal processors. The overall performance of these systems depends on the performance of the multipliers and if the multipliers are too slow, the performance of entire circuits will be reduced. Hence, it is important to design high performance multipliers. Therefore, a multiplier design with a novel Adaptive Hold Logic (AHL) is proposed. The multiplier is able to provide higher performance through the variable-latency and can adjust the AHL circuit to reduce maximum power consumption and delay. The timing violations are reduced based on the idea of razor flip flop and Adaptive Hold Logic. In the fixed-latency technique, usage of clock cycles is increased. The re-execution of clock cycles is reduced by using variable-latency. The experimental results show that our proposed architecture with column- and row-bypassing multipliers achieves better performance in power consumption and delay compared with fixed-latency column- and row-bypassing multipliers.***

Keywords – Adaptive Hold Logic (AHL), Fixed-Latency, Variable-Latency, Razor Flip Flop.

I. INTRODUCTION

Digital multipliers are among the most critical arithmetic functional units in many applications such as the Fourier Transform, Discrete Cosine Transforms, and digital filtering. The performance of these systems is generally determined by the performance of the multiplier because the multiplier isgenerally the slowest element in the system. Hence, the variable-latency design is proposed to reduce maximum power consumption and timing waste of traditional circuits. The variable-latency technique divides the circuit into two parts, they are, the shorter paths and the longer paths. Shorter paths can execute correctly in one cycle. In case of the longer paths, it needs two cycles to execute. When shorter paths are activated frequently, the average latency of variable-latency designs is better than that of traditional designs. Latency is the delay from input into a system to desired outcome. Also, it is well known that multipliers consume most of the power in DSP computations. Hence, low power column-bypassing multipliers and row-bypassing multipliers have been proposed to reduce the number of delay as well as power consumption. The delay and power reduction depends on the input bit coefficient. This means that if the input bit coefficient is zero, corresponding row or column of adders need not be activated.

II. ARRAY MULTIPLIER

The array multiplier (AM) is a fast parallel multiplier. The multiplier array consists of (n-1) rows of carry save adder (CSA). Each row of CSA contains (n-1) full adder (FA) cells. Each full adder in the carry save adder has two outputs, one is the sum and the other is the carry. The sum output goes down and the carry bit goes to the lower left full adder. The last row is a ripple adder for carry propagation. The full adders in the array multiplier are always active regardless of input states.

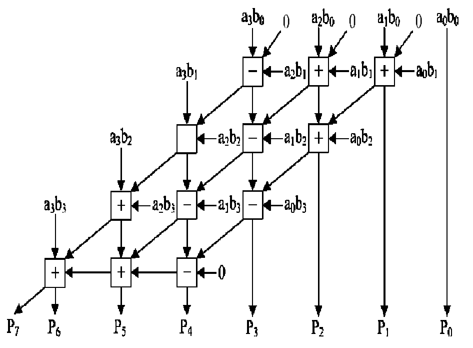

Fig 1: 4x4 array multiplier

III. COLUMN BYPASSING MULTIPLIER

A column-bypassing multiplier is an improvement on the normal array multiplier. A low-power column-bypassing multiplier design is proposed in which the full adder operations are disabled if the corresponding bit in the multiplicand is zero. Suppose, the input bits are 1010*1111, it can be notedthat for the full adders in the first and third diagonals, two of the three input bits are 0: the carry bit from its upper right full adder and the partial product a_ib_i. Hence, the output of the adders in both the diagonals is zero. The output sum bit is equal to the third bit, which is the sum output of its upper full adder. The full adder is modified to add two tristate gates and one multiplexer. The multiplicand bit, a_i can be used as the selector of the tristate gate to turn off the input path of the full adder. If a_i is 0, the full adder inputs are disabled, and the sum bit of the current full adder is equal to the sum bit from its upper full adder. Thus, the power consumption of the multiplier is reduced.

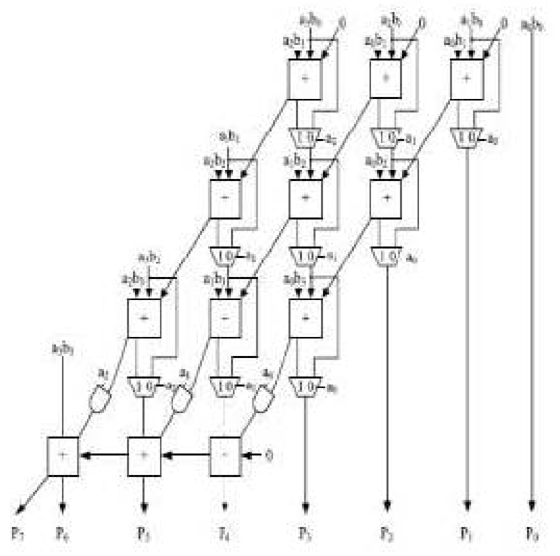

Fig 2: 4x4 column-bypassing multiplier

IV. ROW-BYPASSING MULTIPLIER

A low-power row-bypassing multiplier isalso proposed in order to reduce the power consumption of the array multiplier. The operation of the low-power row-bypassing multiplier is similar to that of the low-power column-bypassing multiplier, but the selector of the multiplexers and the tristate gates use the multiplicator. Each input is connected to a full adder through a tristate gate. Supposing the inputs are 1111*1001, the two inputs in the first and the second rows are zero for full adders. Since b_1 is zero, the multiplexers in the first row select a_ib_0 as the sum bit and select 0 as the carry bit. The inputs are bypassed to the full adders in the second rows, and the tristate gates turn off the input paths to the full adders. Therefore, in the first-row full adders, no switching activities occur; in return, power consumption is reduced. Similarly, since b_2 is 0, no switching activities will occur in the second-row full adders. The full adders in the third row must be active because b_3 is not 0.

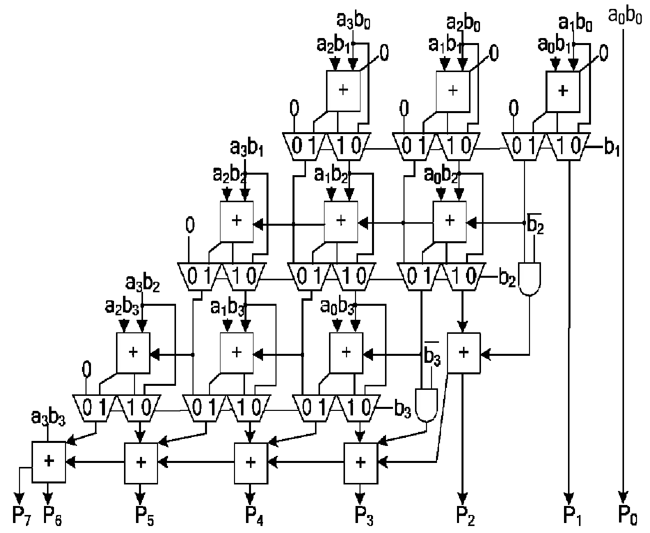

Fig 3: 4x4 row-bypassing multiplier

V. VARIABLE LATENCY DESIGN

The variable latency design is proposed to reduce the timing waste occurring in traditional circuits that uses the critical path cycle as an execution cycle period. The basic concept is to execute shorter paths using shorter cycles and longer paths using two cycles. Since most paths execute in a cycle period that is much smaller than the critical path delay, the variable-latency design has smaller average latency. For example, in an 8-bit variable-latency ripple carry adder,*A*8–*A*1, *B*8–*B*1 are the 8-bit inputs, and *S*8–*S*1 are the outputs. Supposing the delay for each full adder is one, and the maximum delay for the adder is 8.Through simulation, it can be determined that the possibility of the carry propagation delay being longer than 5 is low.

Hence, the cycle period is set to 5. Hold logic is added to notify the system whether the adder can complete the operation within a cycle period. Another key point is that the path delay for an operation is strongly tied to the number of zeros in the multiplicands in the column-bypassing multiplier and multiplicators in the row-bypassing multiplier. It can be seen that as the numberof zeros increases, delay distribution is left shifted and average delay is reduced.

VI. PROPOSED MODEL

The proposed multiplier architecture includes two m-bit inputs (m is a positive number), one 2m-bit output, one column- or row-bypassing multiplier, 2m 1-bit Razor flip flops, and an AHL circuit.

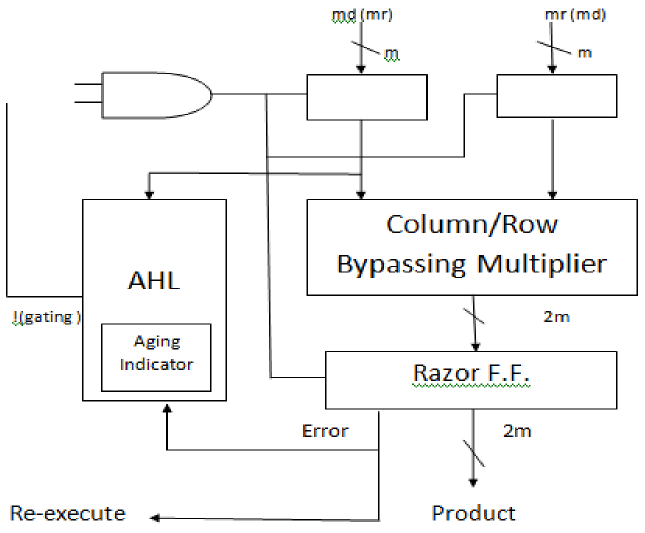

Fig 4: Proposed architecture

The column- and row-bypassing multipliers can be examined by the number of zeros in either the multiplicand or multiplicator to predict whether the operation requires one cycle or two cycles to complete.The two multipliers can be implemented using similar architecture, and the difference between the two bypassing multipliers lies in the input signals of the AHL. According to the bypassing selection in the column or row-bypassing multiplier, the input signal of the AHL in the architecture with the column-bypassing multiplier is the multiplicand, whereas that of the row-bypassing multiplier is themultiplicator.

6.1. Razor flip flop

Razor flip-flops can be used to detect whether timing violations occur before the next input pattern arrives. A 1-bit Razor flip-flop contains a main flip-flop, shadow latch, XOR gate, and multiplexer. The main flip-flop catches the execution result for the combination circuit using a normal clock signal.The shadow latch catches the execution result using a delayed clock signal, which is slower than the normal clock signal.

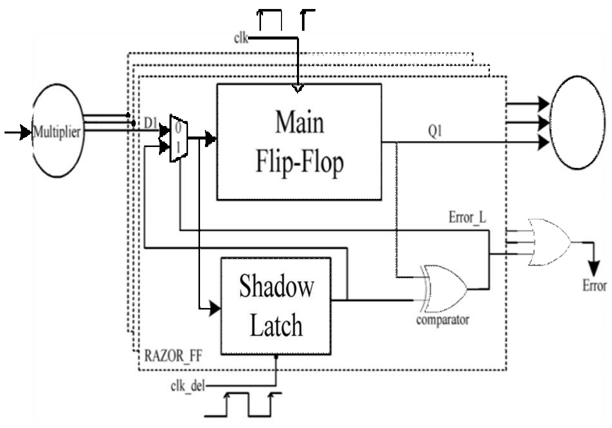

Fig 5: Razor flip flops

If the latched bit of the shadow latch is different from that of the main flip-flop, this means the path delay of the current operation exceeds the cycle period, and the
main flip-flop catches an incorrect result. If errorsoccur, the Razor flip-flop will set the error signal to 1 to notify the system to re-execute the operation. It also notifies the AHL circuit that an error has occurred.

6.2. Adaptive Hold Logic

The key component in variable-latency multiplier is the AHL circuit. The AHL circuit contains an aging indicator, two judging blocks, one multiplexer, and one D flip-flop.

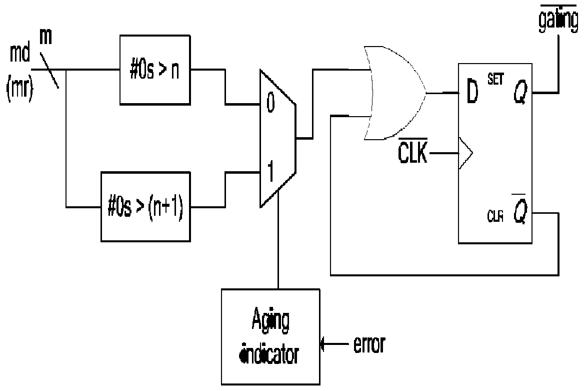

Fig 6: Adaptive Hold Logic

The aging indicator indicates whether the circuit has suffered significant performance degradation due to the aging effect. When input patterns arrive, the column- or row-bypassing multiplier, and the AHL circuit execute simultaneously. According to the number of zeros in the multiplicand (multiplicator), the AHL circuit decides if the input patterns require one or two cycles. If the input pattern requires two cycles to complete, the AHL will output 0 todisable the clock signal of the flip-flops. Otherwise, the AHL will output 1 for normal operations. When the column- or row-bypassing multiplier finishes the operation, the result will be passed to the Razor flip-flops.

VII. SIMULATION RESULTS

The following results are obtained by carrying out the simulation using Xilinx. Instead of fixed-latency technique, variable latency is used and the power and delay are reduced.

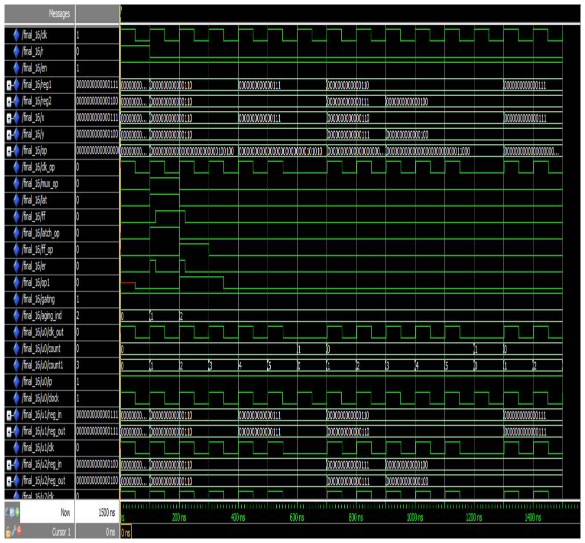

Fig 7: Fixed-latency column-bypassing multiplier

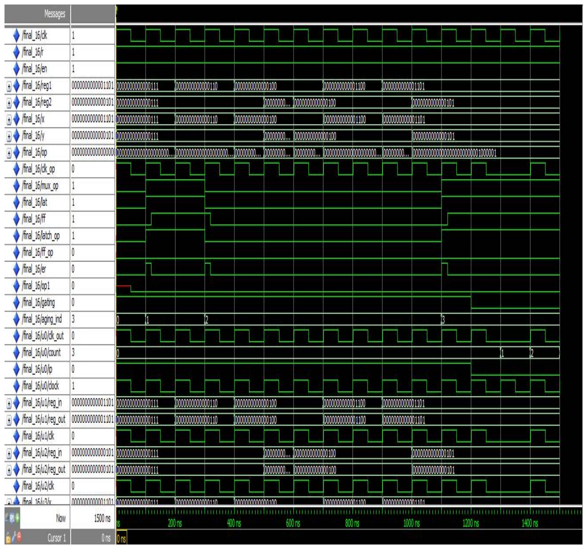

Fig 8: Variable-latency column-bypassing multiplier

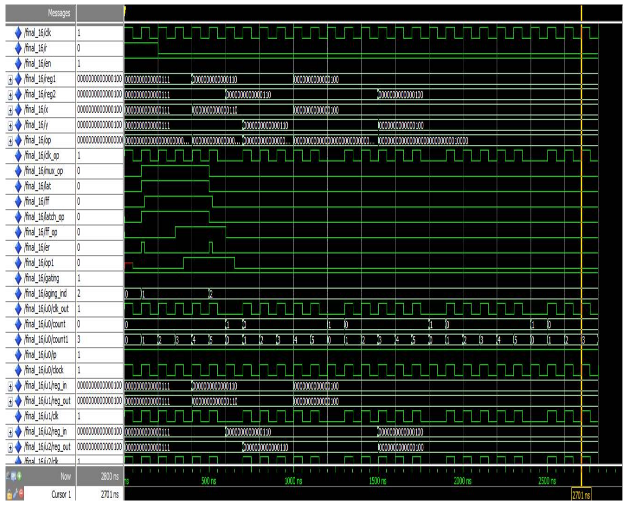

Fig 9: Fixed-latency row-bypassing multiplier

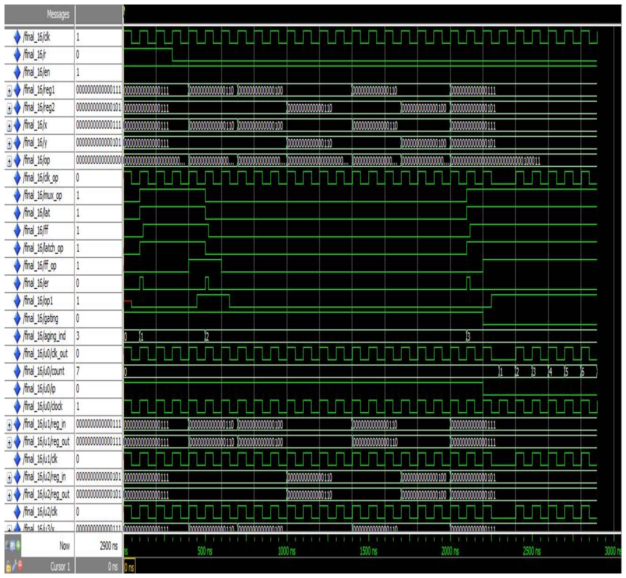

Fig 10: Variable-latency row-bypassing multiplier

Table 1: Comparison of Power consumption and Delay in fixed- and variable-latency technique

LATENCY	POWER(mW)		DALAY(ns)	
	ROW	COLUMN	ROW	COLUMN
FIXED	199	192	10.001	10.001
VARIABLE	179	180	6.237	6.237

Hence from the above table, it is observed that variable-latency technique consumes less power and has reduced time delay compared to fixed-latency design.

VIII. CONCLUSION

The variable latency technique has been shown to be very useful in reducing the power consumption and time delay. The experimental results show that our proposed architecture with the 16x16 row- and column-bypassing multipliers has better performance compared with the 16x16 fixed-latency row- and column-bypassing multiplier.

References

[1] Ing-Chao Lin, Yu-Hung Cho, and Yi-Ming Yang , "Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic",IEEE Transactions on Very Large Scale Integration (VLSI) Systems,VOL.23,No.3,pp.544-556,March 2015.

[2] 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.

[3] J. Ohban, V.G. Moshnyaga, and K. Inoue, " Multiplier energy reduction through bypassing of partial products", in Proc. APCCAS, 2002, pp.13-17.

[4] K. Du, P.Varman, and K. Mohanram, "High performance reliable variable latency carry select addition", in Proc.DATE, 2012, pp.1257-1262.

[5] A. K. Verma, P. Brisk, and P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design", in Proc. DATE, 2008, pp. 1250-1255.

[6] K.-C. Wu and D.Marculescu, "Aging-aware timing analysis and optimization considering path sensitization", in Proc. DATE, 2011, pp.1-6.

[7] D.Baneres, J. Cortadella, and M.Kishinevsky, "Variable-latency design by function speculation", in Proc. DATE, 2009, pp.1704-1709.