Variable Latency Multiplier Design with Adaptive Hold Logic
Sreeja Raj¹, Aneesh S Perumprath²
M.Tech Scholar in VLSI and Embedded Systems¹, Assistant Professor²
Department of ECE
Musalar College of Engineering and Technology, Kerala, India

Abstract:
High speed and low power consumption is one of the most important design objectives in integrated circuits. As multipliers are the most widely used components in such circuits, the multipliers must be design efficiently. Digital multipliers are most critical functional units. The overall performance of these systems depends on the throughput of multiplier design. Aging problem of transistors has a significant effect on performance of these systems and in long term, the system may fail due to timing violations. Aging effect can be reduced by using over-design approaches, but these leads to area, power inefficiency. Hence to reduce the maximum power consumption and delay, variable latency multiplier with adaptive hold logic is used. The proposed multiplier is able to provide higher throughput through the variable latency and can adjust the AHL circuit to mitigate performance degradation that is due to the aging effect. Based on the idea of razor flip flop and adaptive hold logic the timing violations are reduced. Moreover proposed architecture can be applied to a column or row bypassing multiplier. This multiplier design can be applied to digital filter so as to enhance its performance. The VHDL language is used for coding, synthesis was done by using Xilinx ISE and simulated by using Model-Sim.

Index Terms: Adaptive hold logic (AHL), negative bias temperature instability (NBTI), positive bias temperature instability (PBTI), reliable multiplier, variable latency.

I. INTRODUCTION
Multiplication is an essential arithmetic operation for common DSP applications, such as filtering and fast Fourier transform (FFT). To achieve high execution speed, parallel array multipliers are widely used. These multipliers tend to consume most of the power in DSP computations, and thus power-efficient multipliers are very important for the design of low-power DSP systems. If the multipliers are too slow, the performance of entire circuits will be reduced. Meanwhile, the negative bias temperature instability effect occurs when a pMOS transistor is under negative bias (Vgs = -Vdd), increasing the threshold voltage of the pMOS transistor, and reducing multiplier speed. A similar phenomenon, positive bias temperature instability, occurs when an nMOS transistor is under positive bias. Both effects degrade transistor speed, and in the long term, the system may fail due to timing violations. Therefore, it is important to design reliable high-performance multipliers. Traditional circuits use critical path delay as the overall circuit clock cycle in order to perform correctly. However, the probability that the critical paths are activated is low. In most cases, the path delay is shorter than the critical path. For these noncritical paths, using the critical path delay as the overall cycle period will result in significant timing waste. Hence the variable-latency design was proposed to reduce the timing waste of traditional circuits. The variable-latency design divides the circuit into two parts: 1) shorter paths and 2) longer paths. Shortest paths are assigned to be executed within one cycle and longest paths within two or more cycle. When shorter paths are activated frequently, the average latency of variable-latency designs is better than that of fixed latency designs. The main objective of the work is to multiply two images using low power variable latency multiplier with AH logic. The coding can be synthesized by the Xilinx ISE Design Suite, simulated using Model-Sim simulator. The major application of these digital multipliers are in the field of digital filtering and signal processing.

II. PAPER CONTRIBUTION
In this paper, a reliable multiplier design with a adaptive hold logic (AHL) circuit is proposed. The multiplier is based on the variable-latency technique and can adjust the AHL circuit to achieve reliable operation to reduce the error and reexecution of clock cycle. The adaptive hold logic (AHL) circuit can decide whether the input patterns requires one or two cycles and can adjust the judging criteria to ensure that there is minimum error detection and reexecution of clock cycle. Based on the idea of razor flip flop and adaptive hold logic the timing violations are reduced. Moreover proposed architecture can be applied to a column or row bypassing multiplier.

III. METHODOLOGY
Performance of most of the digital circuits depends on throughput of multipliers. The primary objective is power reduction with small area and delay overhead. Low power Variable latency multiplier design with AH logic introduces a multiplier, in which AHL circuit associated with it adjusts the circuit when timing delays occurs so as to ensure minimum performance degradation.

1.1 Array multiplier
The array multiplier is a fast parallel multiplier and is shown in Figure.1 and it consists of (n-1) rows of carry save adder, in
which each row contains (n-1) full adders. Each full adder in the carry save adder array has two outputs they are the sum bit goes down and the carry bit goes to the lower left full adder. The last row is a ripple adder for carry propagation.

1.2 Column-bypassing multiplier
A column-bypassing multiplier is a modified array multiplier. The column-bypassing multiplier is an excellent candidate for the variable design since we can simply examine the number of zeros in the multiplicand to predict whether the operation requires one cycle or two cycles to complete. The FA operations are disabled if the corresponding bit in the multiplicand is 0.

Figure 2 shows a 4x4 column-bypassing multiplier. Supposing the inputs are 1010*1111, it can be seen that for the full adders (FAs) in the first and third diagonals, two of the three input bits are 0: the carry bit from its upper right FA and the partial product $a_i * b_i$. Therefore, the output of the adders in both diagonals is 0, and the output sum bit is simply equal to the third bit, which is the sum output of its upper FA.

Figure 2. 4 x 4 Column-bypassing multiplier

The FA is modified to add two tristate gates and one multiplexer. The multiplicand bit $a_i$ can be used as the selector of the multiplexer to decide the output of the FA, and $a_i$ can also be used as the selector of the tristate gate to turn off the input path of the FA. If $a_i$ is 0, the inputs of FA are disabled, and the sum bit of the current FA is equal to the sum bit from its upper FA, thus reducing the power consumption of the multiplier. If $a_i$ is 1, the normal sum result is selected.

1.3 Row-bypassing multiplier
A low-power row-bypassing multiplier is also proposed to reduce the activity power 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.

1.4 Variable-latency design
The variable-latency design was proposed to reduce the timing waste occurring in traditional circuits. The basic concept is to execute a shorter path using a shorter cycle and longer path 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. Fig. 3 is an 8-bit variable-latency ripple carry adder (RCA). A8–A1, B8–B1 is 8-bit inputs, and S8–S1 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, and hold logic is added to notify the system whether the adder can complete the operation within a cycle period.

Figure 3. 8 bit RCA with a hold logic circuit

IV. PROPOSED MULTIPLIER
1.5 Proposed model
The multiplier architecture, which 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 as shown in Fig. 4. 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. When input patterns are random, the number of zeros and ones in the multiplicator and multiplicand follows a normal distribution. Therefore using the number of zeros or ones as the judging criteria results in similar outcomes. Hence, 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 by passing 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 the multiplicator.

Figure 4. Proposed architecture
1.6 Razor flip-flop
Figure 5 shows the block diagram of the razor flip-flop. Input to the razor flip-flop is the 2m bit output from column bypassing multiplier. For each m bit a 1-bit razor flip-flop is used. A 1-bit Razor flip-flop contains a main flip-flop, shadow latch, XOR gate and MUX. The main flip-flop catches the execution result for the combination circuit using a normal clock signal, and the shadow latch catches the execution result using a delayed clock signal, which is slower than the normal clock signal. If the latch 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 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. Razor flip-flops is used to detect whether an operation that is considered to be a one-cycle pattern can really finish in a cycle. If not, the operation is re-executed with two cycles.

Figure 5. Razor flip-flop

1.7 Adaptive hold logic
The Adaptive Hold Logic (AHL) circuit is the key component of variable-latency multiplier. Block diagram of adaptive hold logic is shown in figure 6. The AHL circuit contains decision block, MUX and a D flip-flop. If the cycle period is too short, the column-bypassing multiplier is not able to complete these operations successfully, causing timing violations. These timing violations will be caught by the Razor flip-flops, which generate error signals. If errors happen frequently, it means the circuit has suffered significant timing degradation due to the aging effect.

Figure 6. Diagram of AHL

The operation of the AHL circuit are as follows: when an input pattern arrives, decision block will decide whether the pattern requires one cycle or two cycles to complete and pass both results to the multiplexer. The multiplexer selects one of either result based on the output of the razor flip-flop. Then an OR operation is performed between the result of the multiplexer, and the Q_bar signal is used to determine the input of the D flip-flop. When the pattern requires one cycle, the output of the multiplexer is 1. The !gating) signal will become 1, and the input flip-flops will latch new data in the next cycle. 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 disable the clock signal of the input flip-flops in the next cycle.

1.8 Overall working of proposed multiplier
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 to disable 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. The Razor flip-flops check whether there is the path delay timing violation. If timing violations occur, it means the cycle period is not long enough for the current operation to complete and that the execution result of the multiplier is incorrect. Thus, the Razor flip-flops will output an error to inform the system that the current operation needs to be re-executed using two cycles to ensure the operation is correct. In this situation, the extra re-execution cycles caused by timing violation incurs a penalty to overall average latency. However, our proposed AHL circuit can accurately predict whether the input patterns require one or two cycles in most cases. Only a few input patterns may cause a timing variation when the AHL circuit judges incorrectly. In this case, the extra re-execution cycles did not produce significant timing degradation.

V. SIMULATION RESULTS

Figure 7. Simulation result of 16 x 16 fixed latency column bypassing

Figure 8. Simulation result of 16 x 16 variable latency column bypassing
VI. APPLICATIONS
This multiplier can be used in many applications, such as fourier transform, discrete cosine transforms, digital filtering. The throughput of these applications depends on speed multipliers. This multiplier can be used to multiply two images on a pixel by pixel basis, thus blending the two images into a single output image.

VII. CONCLUSION
This paper proposed an efficient aging-aware reliable multiplier design with the AHL. The multiplier is able to adjust the AHL to mitigate performance degradation due to increased delay. Variable-latency design minimizes the timing waste of the noncritical paths. The Razor flip-flops detect the timing violations and re-execute the operations using two cycles. Variable-latency design can adjust clock cycle required by input patterns to minimize performance degradation. And hence variable latency multipliers have less performance degradation when compared with traditional fixed latency multipliers, which needs to consider the degradation by the NBTI effect and use the worst case delay as the cycle period. In the variable latency lesser number of clock cycle are used, the error is reduced so that the area, power and delay are reduced. The experimental results show that our proposed architecture with the 16 x 16 column-bypassing multipliers can attain up to 62.88% performance improvement compared with the 16 x 16 fixed-latency column-bypassing multipliers.

VIII. REFERENCES