An Approximate Sequential Multiplier with Segmented Carry Chain and Variable Accuracy

Jorge Echavarria, Stefan Wildermann, Faramarz Khosravi, Jürgen Teich
Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 91058 Erlangen, Germany
Email:{jorge.echavarria, stefan.wildermann, faramarz.khosravi, juergen.teich}@fau.de

Abstract—Presented is an approximate multiplier based on sequential accumulations. The particularity of this architecture is variability in the accuracy of the partial product additions, and hence of the overall outcome by splitting the carry chain of the accumulators to reduce the length of the critical path. Our approximate multiplier thus trades-off accuracy for shortened delays while also saving resources in comparison to combinatorial approaches. Also presented is an error analysis of the implementation of the multiplier.

Index Terms—Approximate Sequential Multiplier.

I. INTRODUCTION

APPROXIMATE COMPUTING (AC) takes advantage of the tolerability of inaccuracy and errors of computations in many applications. That is, rather than computing fully accurate results, user-defined goals like higher performance, lower chip area, and/or power reductions can be achieved.

From this perspective, and in hand with the fact that arithmetic operations are important application areas of approximate computing, e.g., for multimedia processing, we propose a novel sequential approximate multiplier design that operates with reduced carry chains. This capability allows us to shorten the critical path (CP) within the combinatorial part, and thus increase the operational frequency.

II. PROPOSED DESIGN

The so-called grade-school $n \times n$ multiplication algorithm multiplies each digit (bit) of an $n$-bit multiplicand by an $n$-bit multiplier. Representing a multiplication by the radix, each partial product is shifted to the left by the amount ($m$) equal to the power of the corresponding bit of the multiplicand. As the addend is shifted to the left $m$ places, the $m$ least significant bits (LSBs) of the augend do not take part in the addition. These extra $m$ bits are simply right-concatenated with the $n$-bit resultant accumulation. Thus, only an $n$-bit adder is required to perform this accumulation. Clearly, computing this algorithm using a sequential multiplier requires fewer resources than a combinatorial approach. Furthermore, some power savings are expected due to less interconnect, which in turn means that the capacitive load on the signals in the design decreases. With a lower load, dynamic power consumption is reduced.

Formally, let $p(a, b): \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n}$ represent the accurate product of two $n$-bit binary numbers $a$ and $b$, such that $a \in \mathbb{A}$, $b \in \mathbb{B}$, and $p \in \mathbb{P}$, where $\mathbb{A} \subseteq \mathbb{B}$ and $\mathbb{B} \subseteq \mathbb{B}^n$, and $p(a_{n-1}, \ldots, a_0, b_{n-1}, \ldots, b_0) = (p_{2n-1}, \ldots, p_0)$. Let $a_i(b_i)$ be the $i$-th ($j$-th) bit of multiplier $a$ (multiplicand $b$). $S_{ij}^l$ be the $i$-th bit in the accumulated sum during the $j$-th clock cycle, and $C_{ij}$ the $i$-th carry-bit in the $j$-th carry chain.

Verma et al. showed that on average a carry propagates only a small fraction of the carry chain [2]. Based on this assumption, approximate addition via segmented carry chains has become a well-investigated methodology, resulting in multiple works on its error analysis [3], [4]. Consequently, the adder within our following approximate sequential multiplier is based on carry chain segmentation at a splitting point.

Figure 1: Schematic of the presented sequential approximate multiplier. For the sake of simplicity, the controller and clock lines are not shown. Shift registers $A_{LSP}$, $A_{MSP}$, and $B$ have synchronous inputs for parallel load. Shifting to the right with left serial input, and clear to set to 0. The D flip-flops have asynchronous clear inputs. A decrement unit is also shown which, besides informing the controller about sequence completion, enables the multiplexing of the least significant $n+t$ bits according to the carry-out of the last accumulation—VHDL and Verilog codes publicly available [1].

$t, t \leq \frac{n}{2}$. Figure 1 shows the schematic of the circuit implementing the approximate multiplier. As can be seen, one $(n-t)$-bit most significant part (MSP) adder and one $t$-bit least significant part (LSP) adder, two D flip-flops, one $(n-t)$-, one $t$- and one $n$-bit right shift registers are sufficient—note that shifting the augend to the right is equivalent to shifting the addend to the left. By doing so, the next partial product may be added directly to the result. Observe the shortened carry chain w.r.t. a fully accurate sequential multiplier. At clock cycle $j=0$, the multiplicand $b$ is stored in the shift register $B$, and shift registers $A_{MSP}$ and $A_{LSP}$ are set to zero. During each shift to the right, the LSB of register $B$, i.e., $b_j$, is disposed to $B_{lsb}$, the LSB of register $A_{LSP}$, i.e., $S_{lsb}^l$, is introduced from the left to register $B$, the LSB of register $A_{MSP}$, i.e., $S_{lsb}^l$, is introduced from the left to register $A_{LSP}$, the D flip-flop storing the carry-out $C_{out}$ from the LSP Adder in the previous addition, i.e., $C_{out}^{j-1}$, is introduced as the carry-in $C_{in}$ of the MSP Adder, and the D flip-flop storing the carry-out $C_{out}$ from the MSP Adder in the previous addition, i.e., $C_{out}^{j-1}$, is introduced from the left to register $A_{MSP}$. The inputs of the MSP Adder and the LSP Adder are 1) the result of ANDing bit $B_{lsb}$ with multiplier $a$, and 2) the previous addition—located in registers $A_{MSP}$ and $A_{LSP}$, right-shifted once. At clock cycle $j=0$, shift registers $A_{MSP}$ and $A_{LSP}$ hold the $n$ most significant bits (MSBs), and shift register $B$ holds the $n$ LSBs of the approximate $2n$-bit product $p$. As seen, our approximate multiplier is made of two fully accurate adders. However, segmenting the carry chain of the partial product accumulation with an MSP $t$-bit adder and an MSP $n-t$-bit adder makes it possible to reduce the latency theoretically by $1 - \max(t, n-t)/n$.

The carry-out of the LSP Adder, as shown in Figure 1, being connected to the D flip-flop driving the carry-in of the MSP Adder causes a delay of the carry propagation by one clock cycle. When such a carry is 1 during the last clock cycle, i.e., $C_{out}^{n-1} = 1$, a zero detect signal enables the multiplexers setting all $n+t$ LSBs to 1,
representing the decimal value $2^{n+1} - 1$, that is, the closest decimal value to the disregarded overflow, thus, as we will show, considerably reducing the magnitude of the absolute error distance \( |ED| \). Such a fix-to-1 instrumentation reduces the mean error distance (MED) when considering absolute error distances (EDs). However, for error compensation in, for example, cascaded approximate multipliers, it may be disabled to allow for negative EDS, and hence, reduce the global MED.

Formally, our approximate multiplier performs the following computations:

\[
\hat{S}_i = \begin{cases} 
\int(s_i^0), & \text{if } r \in [0, n-1) \text{ and } \hat{c}_{n-1} = \perp \\
1, & \text{if } r \in [0, n-1) \text{ and } \hat{c}_{n-1} = \top \\
\int(s_{r-1}^{n-1}), & \text{if } r \in [n-1, t+n) \text{ and } \hat{c}_{n-1} = \perp \\
1, & \text{if } r \in [n-1, t+n) \text{ and } \hat{c}_{n-1} = \top \\
\int(s_{r-1}^{n-1}), & \text{if } r \in [n, 2n-1],
\end{cases}
\]

where \( \int(x) : \{\perp, \top\} \rightarrow \{0, 1\} \) corresponds to the non-negative integer representation of the boolean value \( x \in \{\perp, \top\} \).

Let the arithmetic error rate (AER) define the error rate (ER) of an approximate adder, and \( S^j \) the \( j \)-th accumulated sum—cf. [4]. According to the general disjunction rule for \( n-t \) number of events, it can be shown that

\[
ER(p, \hat{p}) = \sum_{k=1}^{n-t} (-1)^{k-1} \times \sum_{i \in \mathbb{C}_{n-t}} \rho \left( \hat{S}_{i}^{n-1} \right)_i \neq 0 \\
\cap BER(S_0^{n-t-2}, S_0^{n-t-2})_{i_k} \neq 0 \cap \ldots \cap BER(S_0^{n-1}, S_0^{n-1})_{i_0} \neq 0,
\]

for \( n > 4 \) and \( t \leq \frac{n}{2} \), where \( \rho(F) \) represents the probability of an expression \( F \) to be true, \( \mathbb{C}_{n-t} \) the set of all ordered \( k \)-tuples \( l_1, \ldots, l_k \) of \( \{1, \ldots, n-t\} \), and AER(\( S^j \)) the AER of an erroneous accumulated sum:

\[
AER(S^j) = \rho \left( \left( S_{i}^{j-1} \wedge a_{i-1} \wedge b_j \right) \vee \left( \bigwedge_{i=0}^{t-2} (S_{i+1}^{j-1} \wedge a_i \wedge b_j) \right) \right).
\]

when \( j > 0 \), and \( 0 \) when \( j = 0 \). The bit error rate (BER) \( BER(S_0^{j}, S_0^{j}) \) of the 0-th bit in the \( j \)-th approximately accumulated sum \( S_0^{j} \) is obtained as:

\[
BER(S_0^{j}, \hat{S}_0^{j}) = \frac{1}{|\mathbb{A} \times \mathbb{B}|} \sum_{A \times B} \int(s_i^0 \oplus \hat{s}_i^0),
\]

where \( \mathbb{A} \times \mathbb{B} \subseteq \{(a, b) | a \in \mathbb{A}, b \in \mathbb{B}\} \) is the Cartesian product of subsets \( \mathbb{A} \) and \( \mathbb{B} \).

Next, we calculate the maximum absolute error (MAE). Obviously, the action of setting all bits in shift registers \( A_{\text{SP}} \) and \( B \) to 1, whenever a carry chain generated within the LSP Adder cannot be propagated to the MSP Adder—right before the last shift to right operation during clock cycle \( n \)—results in fixing the approximated product bits \( \hat{p}_{n-t-1, \ldots, 1} \) to 1. Thus, the MAE occurs when there is a carry propagated at bit position \( t-1 \) in the second last partial accumulation, and there is no carry at all at the same position but in the last partial accumulation. Therefore, the probability of the approximated product \( \hat{p}(a, b) \) evaluating to an erroneous result with an ED as large as the MAE is:

\[
\rho(ED(p, \hat{p}(a, b))) = MAE(p, \hat{p}) = \rho \left( \hat{C}_{t-1}^{n-2} \wedge \hat{C}_{t-1}^{n-1} \right).
\]

Moreover, as the MAE occurs when \( \hat{C}_{t-1}^{n-2} = \top \) and \( \hat{C}_{t-1}^{n-1} = \perp \), an error is introduced at bit position \( t \) in the last partial accumulation, that is, \( S_{t,n}^{n-1} \), with a magnitude of \( 2^t \). Furthermore, during the calculation of \( S_{n-1}^{n-1} \)-shift register \( B \) carries the \( n-1 \) LSBs of \( \hat{p} \), incrementing the magnitude error introduced in \( S_{n-1}^{n-1} \) by \( 2^{n-1} \). Finally, the \( t+1 \) LSBs are fully accurate whenever there is not a fixing to 1 operation, reducing the MAE by \( 2^{t+1} \):

\[
MAE(p, \hat{p}) = 2^{n-t-1} - 2^t.
\]

The MED considers all the erroneous results to calculate the average error:

\[
MED(p, \hat{p}) = \sum_{\delta \in [0, \text{MAE}(p, \hat{p})]} \sum_{b \in \mathbb{B}} \Pr(a) \times \Pr(b) \cdot \rho(ED(p(a, b), \hat{p}(a, b)) = \delta),
\]

where \( \Pr(x) \) represents the probability density function (PDF) of a binary number \( x : x \in \mathbb{B}^n \). Note that \( \delta \) iterates over all possible EDS, with the ED defined as follows:

\[
ED(p(a, b), \hat{p}(a, b)) = \sum_{i=0}^{2n-1} \left( \int(p_i(a, b) \oplus \hat{p}_i(a, b)) \times 2^i \right) \times \text{sgn} \left( \int(p_i(a, b)) - \int(\hat{p}_i(a, b)) \right),
\]

where \( \text{sgn}(x) = \frac{x}{|x|} \) with \( x \in \mathbb{Z}^n \).

III. CONCLUSION

We presented an approximate sequential multiplier with segmented carry chain and variable accuracy, and we discussed the most relevant error metrics. In order to ease further comparisons we made our designs implemented in Verilog and VHDL available online. The delay of sequential multipliers is higher than that of combinatorial ones due to the number of cycles required for obtaining the product. Thus, we want to study other versions of our sequential approximate multiplier that is able to compute \( k > 1 \) partial products within each clock cycle. Further discussion on the error metrics and their complexity, area and power estimations targeting multiple technologies, and comparisons with the state-of-the-art multipliers have been left out in this extended abstract due to space limitations.

REFERENCES


