Design of 32-bit cell-based carry-save combinational multiplier with reduced area and propagation delay

Shruti Suman¹, Ngangbam Phalguni Singh², Selvakumar R³, Harshita Saini⁴

ECE Department, Koneru Lakshmaiah Education Foundation (K L University), Andhra Pradesh, India¹,²,³
C-DAC’s Advanced Computing Training School, Pune, India⁴

shrutisuman23@gmail.com

Abstract. In Very Large Scale Integrated (VLSI) and Digital Signal Processing (DSP) applications, a Conventional Combinational Multiplier uses a lot of hardware and propagation delay, and that is one of the main issues. The delay is determined by the ripple carries between the adders. Most techniques used in the combinational multiplier depend on the paper-and-pencil shift-and-add (PPSA) algorithm, which uses ripple carry adder architecture. This paper proposes an architecture called Cell-based Carry-Save Combinational Multiplier (CCCM) that focuses on the propagation delay and area. Each row represents a carry-save adder. The carry outputs are passed to the next row. For validating the proposed architecture, 16x16 multiplication is performed and the overall improvement in the delay is 73.32 % while there is a 3.31 % reduction in the area.

Keywords: Multiplier; paper-and-pencil shift-and-add algorithm; propagation delay; ripple carry; carry-save.

1. Introduction

Multiplication is one of the basic arithmetic operations. It operates between 2 to 8 cycles, performing two subtasks of pencil shift-and-add (PPSA) algorithm [1]. In high-performance processors, multipliers use adders for partial products. The most commonly used adders are ripple adder and carry-look-ahead adder [2-5]. Previous works focus on the performance analysis of the various adders and the multipliers with unit-gate models for area and delay [6]. Several algorithms are used to improve speed and minimize the number of partial products. Wallace Tree algorithm and Modified Booth algorithm employ a parallelism technique which leads to reduced speed, extended silicon area caused by the structure irregularity and increased power consumption because of interconnect extension resulting from complex routing. However, the Cell-based Carry-Save Combinational Multiplier is serial-parallel and has better speed, smaller area, and reduced power consumption. To design multipliers of low power and high-speed, either the overall design structure is altered or the adder units are redesigned [7].

1.1. Basic Multiplication Algorithm
S Multiplication is defined as a process of the addition of equal numbers. Here Partial Products are added consecutively to have the least hardware. Each of the partial product is added by a combinational circuit using a parallel multiplier. Still, it is feasible to adapt the compression technique to decrease the number of partial products before the final addition is done.

Algorithm:

Step 1: If the LSB of Multiplier is set ‘HIGH’, the multiplicand is added to the accumulator.

Step 2: Shift the multiplier to right and multiplicand towards left by one bit respectively.

Step 3: Stop as soon as all the bits of multiplier goes low.

Let’s consider the multiplication of two numbers, say a multiplicand ‘A’ of N bit long and a multiplier ‘B’ of length M bits.

\[ A = a_{N-1}a_{N-2} \ldots \ldots \ldots \ldots a_1a_0 \text{ Multiplicand} \]

\[ B = b_{M-1}b_{M-2} \ldots \ldots \ldots \ldots b_1b_0 \text{ Multiplier} \]

The partial products generated here will be of N*M-bit, which is generated using AND gates. Depending on the different algorithms adopted, partial products are also added up accordingly. Figure 1 depicts an example of multiplication of two numbers say ‘A’ and ‘B’ with 16-bits each, as the summation of \( a_i \times b_j \) terms, to generate a 32-bit product Y, i.e. \( y_{31} - y_0 \).

![Figure 1. Multiplication as a summation of \( a_i \times b_j \) terms](image)

The product equation is given as

\[ Y(N + M) = A(N)B(M) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} a_i b_j 2^{i+j} \quad (1) \]

The final products of the multiplication after adding the respective partial products are given by (1). Thus, the product term \( Y[31-0] \) is generated. While generating the product term, some delay is introduced. This paper presents a Cell-based carry-save combinational multiplier that saves the carry generated at the output of each stage and feeds them to the next row cells, where they are added parallelly. It reduces the delay and increases the operating speed.

### 2. Cell-based carry-ripple combinational multiplier

Normally multipliers based on adder uses two M-bit coarse adders. In this architecture, the basic building block is a 1-bit full-adder cell unit. It lets the architecture to traverse a more efficient circuit with a fine structure of multiplier. The multiplication operation can be considered in a two-dimensional pattern, as a summation over \( a_i b_j \) terms in equation (1). This term \( a_i b_j \) transmits a 1-bit value and two
such terms can act as inputs for a 1-bit full adder along with a 1-bit carry-in $c_i$, giving a sum bit, $S_{out}$ and a carry-out, $C_{out}$. However, most of the ASIC technologies use full-adder cells predesigned in the device library which gets generated during synthesis.

**Fig 2.** The architecture of carry-ripple multiplier with propagation delay contour lines

Full adders are arranged in a two-dimensional structure of multiplication operation, summing the terms $a_i b_j$. Carry propagates from LSB to MSB at every stage. Figure 2, $FA_{ij}$ indicates a 1-bit full adder placed at the $i^{th}$ row and the $j^{th}$ column. Array-of-arrays is used to define the bit-product, sum, and carry signals inside. Figure 2 illustrates signal propagation from the LSB in top row to the bottom row MSB. Contour lines represent the propagation of the signal. Likewise, this multiplier architecture can be divided into similar propagation delay stages.

It is observed that all the signals are blocked to the next row until the carry-bit is available in the MSB. If the full-adder cell propagation delay is $T_{fa}$ and the delay in getting $a_i * b_j$ is negligible, then the M-bit multiplier has M-1 rows where each row consists of M full-adder cells.

Here the critical path is formed by the M cells in the top row and two cells each from rest of the M-2 rows. Therefore, the propagation delay is given by (2).

$$MT_{fa} + 2(M-2)T_{fa} = (3M - 4)T_{fa} \quad (2)$$

**3. Cell-based carry-save combinational multiplier**

From Figure 2 and equation (2), it is evident that the carry generated from each stage of the carry-ripple architecture forms a cascading chain, and introduces a delay of $(3M - 4)T_{fa}$. In the worst case, inputs to the least significant adder can affect the most significant bit (MSB) of the product. However, the carry outputs can be saved and passed to the cells in the next row, instead of propagating to the next cell in the same row. In Cell-based carry-save combinational multiplier, carry outputs are passed to the cells in the next row instead of providing it to the next cell of the same row. To design a multiplier, 1-bit full-adder cell is used as a basic unit.
The architecture of a carry-save multiplier using 1-bit full-adders as the basic unit cells is shown in Figure 3. Each row is connected in a cascading chain to form a ripple adder. Each partial product obtained is combined with the next product component using subsequent adders. The cells placed in the last row are organized as regular carry-ripple adders, where all carries are hooked up in the usual way and are allowed to ripple from the LSB to the MSB. The last row, thus adds the carry-out from the last carry-save adder and provides the final output.

Figure 3 shows that for the next step i.e. step $j+1$, the carry output from $i^{th}$ bit during step $j$ is applied to the carry input for $(i+1)^{th}$ bit. It is also obvious that the carry-save multiplier is divided into identical delays stages which remodel it to a pipelined design. To reduce cluttering, the pipeline registers for operands are not included in Figure 3. Figure 4 illustrates the RTL schematic of a 32-bit Cell-based Carry-Save Combinational Multiplier.

![Figure 3. The architecture of a carry-save multiplier with propagation delay contour lines](image1)

![Figure 4. RTL Schematic of 32-bit Cell-based Carry-Save Combinational Multiplier](image2)
Table 1. delay and area cooperation of 32-bit multipliers

| Type of Multiplier | Total CLB’s | Delay due to logical structure (ns) | Delay due to Routing (ns) | Maximum Combinational Path Delay (ns) |
|--------------------|-------------|-------------------------------------|--------------------------|-------------------------------------|
| Carry-Ripple Multiplier | 363 | 22.531 (33.5%) | 44.684 (66.5%) | 67.215 |
| Carry-Save Multiplier | 351 | 1.376 (6.7%) | 16.727 (93.3%) | 17.931 |
| Improvement (%) | 3.31 | 93.89 | 62.57 | 73.32 |

In this design, the propagation of the carries can be traced easily. For example, the critical path for an M-bit multiplier will have M-1 cells in the last row and one cell each from rest of the M-1 rows. Therefore, the propagation delay is given by (3), and total area is given by (4).

\[
(M-1)T_{fa} + (M-1)T_{fa} = 2(M-1)T_{fa} \quad (3)
\]

\[
Total \ Area = (M-1)M * Area_{FA} \quad (4)
\]

4. Results and discussion

Table 1 represents the delay and area cooperation of 32-bit carry-ripple and carry-save multipliers.

Specifications for target device:

![Technology architecture of 32-bit Cell-based Carry-Save Combinational Multiplier](image)

**Fig 5.** Technology architecture of 32-bit Cell-based Carry-Save Combinational Multiplier
Table 2. Comparison of boundary and non-boundary cells

| Specific patterns for non-boundary cells | Specific patterns for boundary cells |
|----------------------------------------|-------------------------------------|
| • The \( c_i \) port is fed to the \( c_{i-1,j} \) signal. | • Top row: The \( s_0 \) is connected to the \( a_0b_0 \) term. Note that the \( b_{15} \) bit doesn’t exist and the leftmost term (i.e., \( a_0b_{15} \) in the diagram) is used for naming convention. The \( a_0b_{15} \) term is set to ‘LOW’. |
| • The \( c_0 \) port is fed to the \( c_{i,j} \) signal, which is treated as the carry-in of the \( FA_{i+1,j-1} \) cell. | • Bottom row: Take \( rc_0 = 0 \). The \( s_0 \) port is \( rc_j \), the \( c_0 \) port is \( rc_{j+1} \) with \( a_i \) connected to \( s_{N-1,j+1} \). \( b_l \) is connected to \( c_{N-1,j} \) while \( c_i \) is connected to \( rc_j \). |
| • The \( s_0 \) port is connected to the \( s_{i,j} \) signal, which is then fed to the bi port of the \( FA_{i+1,j-1} \) cell. | • Leftmost column: The \( s_{i,N-1} \) port of the \( FA_{i,15} \) cell is coupled with the \( a_i b_{N-1} \) from the leftmost cell in the previous row. |
| • The \( a_l \) port is connected to the \( a_{l,b} \) term. | |
| • The \( b_l \) port is coupled with the \( s_{i-1,j-1} \) signal, which is the \( s_0 \) signal of the \( FA_{i-1,j+1} \) cell. | |

5. Conclusions

Several digital processors and VLSI architectures employ multipliers as essential building blocks. This paper presents a carry-save array multiplier with considerable processing time and hardware resources, via Vivado Design Suite HLx Editions (Virtex 7 technology). Results are analyzed based on the delay, area, and power. Further, the analysis also shows the distinction of carry-save array multiplier over carry-ripple multiplier.

References

[1] P Ramakrishna, M. Nagarani, K Hari Kishore “A Low Power 8-Bit Current-Steering DAC Using CMOS Technology” International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6S, Page No: 137-140, April 2019.

[2] Avinash Yadlapati, K Hari Kishore “Implementation of Asynchronous FIFO using Low Power DFT” International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-8, Issue No: 6S, Page No: 152-156, April 2019.

[3] P Ramakrishna, K Hari Kishore “Implementation of Low Power and Area Efficient 7-Bit Flash Analog to Digital Converter” Journal of Computational and Theoretical Nanoscience, ISSN: 1546-1955, Volume-16, Issue No: (5-6), Page No: 2213-2217, June 2019.

[4] Mahesh Madavath, K Hari Kishore “Design and Analysis of Receiver Front-End of CMOS Cascode Common Source Stage with Inductive Degeneration Low Noise Amplifier on 65 nm Technology Process” Journal of Computational and Theoretical Nanoscience, ISSN: 1546-1955, Volume-16, Issue No: (5-6), Page No: 2628-2634, June 2019.

[5] K Hari Kishore, Fazal Noorbasha, Katta Sandeep, D. N. V. Bhupesh, SK. Khadar Imran, K. Sowmya “Linear convolution using UT Vedic multiplier” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 409-418, March 2018.

[6] K Hari Kishore, B. K. V. Prasad, Y. Manoj Sai Teja, D. Akhila, K. Nikhil Sai, P. Sravan Kumar “Design and comparative analysis of inexact speculative adder and multiplier” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 413-426, March 2018.

[7] G Siri Vennela, K Hari Kishore, E Raghuveera “High Accurate and Power Efficient ECG-Based Processor for Predicting Ventricular Arrhythmia” Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 10, Issue No: 2, Page No: 1180-1121, May 2018.
Avinash Yadlapati, K Hari Kishore “Low Power Synthesis for Asynchronous FIFO using Unified Power Format (UPF)” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 7-9, March 2018.

K Hari Kishore, Fazal Noorbasha, Katta Sandeep, D. N. V. Bhupesh, SK. Khadar Imran, K. Sowmya “Linear convolution using UT Vedic multiplier” International Journal of Engineering and Technology(UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 409-418, March 2018.

K Hari Kishore, B. K. V. Prasad, Y. Manoj Sai Teja, D. Akhila, K. Nikhil Sai, P. Sravan Kumar “Design and comparative analysis of inexact speculative adder and multiplier” International Journal of Engineering and Technology(UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 413-426, March 2018.

G Siri Vennela, K Hari Kishore, E Raghuveera “High Accurate and Power Efficient ECG-Based Processor for Predicting Ventricular Arrhythmia” Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 10, Issue No: 2, Page No: 1180-1121, May 2018.

Avinash Yadlapati, K Hari Kishore “Low Power Synthesis for Asynchronous FIFO using Unified Power Format (UPF)” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 7-9, March 2018.

Chella Santhosh, K. Hari Kishore, G. Pavani Lakshmi, G.Kushwathanth, P. Rama Krishna Dharma Teja, R. S. Ernest Ravindran, Sree Vardhan Cheeralu, M. Ravi Kumar “Detection of Heavy Metal Ions using Star-Shaped Microfluidic Channel” International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-12, Page No: 768-771, December 2019.

B Srikanth, M. Siva Kumar, J.V.R. Ravindra, K. Hari Kishore “Double Precession Floating Point Multiplier using Schonhage-Strassen Algorithm used for FPGA Accelerator” International Journal of Emerging Trends in Engineering Research, ISSN: 2347-3983, Volume-7 Issue-11, Page No: 677-684, December 2019.

K Divya Madhuri, K Hari Kishore “Implementation of 4-bit Ripple Carry Adder by Adopting Subthreshold Adiabatic Logic for Ultralow-Power Application” Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 12, Issue No: 6, Page No: 11-17, May 2020.

Avinash Yadlapati, Hari Kishore Kakarla “Design and Verification of Asynchronous FIFO with Novel Architecture Using Verilog HDL” Journal of Engineering and Applied Sciences, ISSN No: 1816-949X, Vol No: 14, Issue No: 1, Page No: 159-163, January 2019.

Ch. Naga Babu, P. Naga Siva Sai, Ch.Priyanka, K Hari Kishore, M.Bindu Bharagavi, K.Karthik “Comparative Analysis of High Speed Carry Skip Adders” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.24, Page No: 121-125, April 2018.

Avinash Yadlapati, K Hari Kishore “Low Power Synthesis for Asynchronous FIFO using Unified Power Format (UPF)” International Journal of Engineering and Technology (UAE), ISSN No: 2227-524X, Vol No: 7, Issue No: 2.8, Page No: 7-9, March 2018.

B. Srikanth, Dr. M. Siva Kumar, Dr. K Hari Kishore, Dr. J.V.R. Ravindra “Towards Reducing Area And Power Of A Multiplier With Double Precision Floating Point Computations Using FPGA Accelerators” Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 9, Special Issue No: 18, Page No: 2780, December 2017.

M. Vasudha, B. Sri Pravallika, Ch. Sai Kiran, P. Subhani, G. Rakesh Chowdary, M Durga Prakash, K Hari Kishore, T.V. Ramakrishna “Design and Performance Analysis of A Nonvolatile Memory Cell” Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 9, Special Issue No: 14, Page No: 2320, December 2017.

A Murali, K Hari Kishore, D Venkat Reddy “Integrating FPGAs with Trigger Circuitry Core System Insertions for Observability in Debugging Process” Journal of Engineering and
Applied Sciences, ISSN No: 1816-949X, Vol No.11, Issue No.12, page: 2643-2650, December 2016.

[22] Nazeer Hussain, K Hari Kishore, ‘’Heuristic Approach to Evaluate the Performance of Optimization Algorithms in VLSI Floor Planning for ASIC Design’’, Studies in Computational Intelligence, Volume: 885, ISSN: 1860-949X, Pp: 213-225, 5th February 2020.