

Comparative Performance Study of Digital Multipliers under Various Conditions in Nanometer SPICE Technology

Hani O. Jamleh*, Abdoul Rjoub**

**(Department of Electrical Engineering, The University of Jordan, Amman, Jordan*

***(Computer Engineering Department, Jordan University of Science and Technology, Irbid, Jordan*

Corresponding Author : Hani O. Jamleh

ABSTRACT

There are many concerns regarding the power dissipation as well as the emergence of green and mobile electronics, as they raise the need for low-energy communicating and computing electronic modules.

This paper investigates, analyzes, and compares the performance of three popular digital multipliers using the state-of-the-art CMOS technologies: 90nm, 65nm, and 22nm, and it aims to evaluate the performance of each design in terms of different supply voltages and load capacitances. These multiplier circuits were built based on the conventional static CMOS Full adder modules, and the simulation results were obtained from HSPICE simulation with nanometer PTMs for CMOS.

Overall simulation results for all operands indicate that Wallace-Tree multiplier mostly outperformed the other two in the three technology nodes, while Bit-Array can be used as a complementary multiplier with Wallace-Tree in the 65nm node. However, the simulation results show that the Carry-Save and Bit-Array designs may have better performance than Wallace-Tree under some values of supply voltage and load capacitance operands, in which the designers may have many options to improve their design's performance.

Keywords – Low-power design, Digital multipliers, CMOS adder, Delay, Leakage current, Nanometer, HSPICE.

Date Of Submission: 25-07-2019

Date Of Acceptance: 06-08-2019

I. INTRODUCTION

In modern very-large-scale integration (VLSI) digital designs, the need for high-performance arithmetic digital multiplier with low-power design is crucial for many applications and systems such as digital signal processing, digital image processing, system on chips, network on chips, internet of things, and artificial neural network, especially in portable devices like smartphones.

Reduction of power and energy consumption in VLSI digital circuit design has become the main design concern in all digital integrated circuits. This concern puts challenges on selecting the technology node that suits most with the utilized digital multipliers and adders. One of the challenges is downsizing the CMOS transistors, wherein the power dissipation is increased. [1] And this certainly affect the speed of running an electronic circuit. Nevertheless, the low energy electronic modules have enabled us to design and build abundant mobile devices that can run on a single battery charge for an extremely long time.

In 2019, the planar CMOS technology continues to be the most commonly used structure in semiconductor devices. It is still efficient for integrated circuits manufacturing with technology

nodes smaller than 20nm before using more advanced technologies. [2]

Digital adders and multipliers are fundamental units used for arithmetic computations. In the literatures, many multiplier architectures and algorithms have been particularly proposed and developed in order to reduce the power consumption [3-9]. Actually, multiplication is one of the most power consuming module in digital designs. [10] Therefore, several techniques have been proposed to decrease the power dissipation for different parallel multiplier architectures [3-9] such as development of new designs [1, 11] and eliminating the spurious transitions [4, 12].

In the contemporary semiconductor technology, powers of leakage and switching are practically taken care of on side by side, and they can't be separated anymore. For better designs, the two powers should be wisely balanced. Moreover, the leakage power depends strongly on the supply voltage and inversely upon the threshold voltage V_{th} . [13]

There are mainly three essential techniques on how to design multipliers based on the input sequence; serial, in parallel, or hybrid (serial/parallel) topologies.[14] Parallel multiplier can improve the speed and leakage power [3, 14]; however, it increases the area complexity, but on the other good side, the speed and leakage power get improved.[3, 14] In this study, three parallel multipliers are designed, simulated using the HSPICE and AvanWaves tools, and then compared in terms of power consumption, speed, and leakage current at different technology nodes, namely 90, 65, and 22nm under a variety of conditions; supply voltage (V_{DD}) levels, and load capacitance (C_L) values. [15] However, these metrics give valuable information not only for the overall design but also for who are pointing one optimizing parameter.

The purpose of this investigation is not realizing a minimum-energy design but searching in the space of energy-delay tradeoff for the lowest energy design for a given performance. This is done by using electronic design automation (EDA) tools to tackle the trade-off space in a systematic approach. We aim to shed the light on how the physical limits of energy scaling will steer new designs in the future.

This paper is organized as the following sections: In Section II, the power consumption in digital CMOS is outlined. Section III describes the conventional static CMOS 28T Full-Adder circuit which is used in our designs. The multiplier under-test (MUT) architectures are described in Section IV. Section V overviews the simulation methodology. In Section VI, results of the simulations are discussed. Finally, Section VII concludes the paper.

II. POWER CONSUMPTION OF DIGITAL CMOS TRANSISTOR

There are two major source of power dissipation in a VLSI design: dynamic and static. The former essentially depends on the frequency and activity in the network, while the latter doesn't depend directly on any of them, but on the leakage current. In the nano-electronics era, the static power should be treated more cautiously. [1] The main source of dynamic power is charging/discharging capacitances, in addition to the dynamic hazards and short-circuit-currents which are considered as parasitic effects.

In a common digital CMOS gate, the average power dissipation is given by the following equation [16, 17]:

$$P_{avg} = P_{dynamic} + P_{short-circuit} + P_{static} \quad (1)$$

This equation (1) will be used with more elaboration in Section V.

III. CONVENTIONAL STATIC CMOS FULL ADDER CELL

The multiplier modules are fundamentally built from adders. Therefore, using efficient and fast full adders will make a major contribution to the performance of the whole digital systems.

In this part, a short description about one of the most common conventional adders is given. The static CMOS 28 Transistor (28T) circuit is used in the designs of the three multipliers. It can be described as a 1-bit full adder (FA) which gives two 1-bit outputs (sum and carry) from three 1-bit inputs (A, B, and C_{in}). The mathematical expressions that relate the inputs to the outputs are stated as the following two equations [1]:

$$C_{out} = A \cdot B + B \cdot C_{in} + A \cdot C_{in} \quad (2)$$

$$SUM = A \cdot B \cdot C_{in} + (A + B + C_{in}) \cdot \overline{C_{out}} \quad (3)$$

The schematic circuit for the described full adder module is illustrated in Figure 1. [1] The most left 10 transistors along with one CMOS inverter at the output produce the output carry C_{out} , while the remaining transistors generate the Sum output. As noted in equation (3), the delay for computing $\overline{C_{out}}$ directly influences the total propagation delay of the SUM output. In Figure 1, the structure of adder is big in size which occupies a large on-chip area. However, this is favorable to investigate the effect of CMOS downsizing in nano-scales on the multipliers' performance.

The structure of the 28T CMOS adder merges NMOS pull-down and PMOS pull-up networks to produce the considered outputs, namely Sum and C_{out} . In this fashion, all transistors are prearranged in entirely separate branches, which contain a number of sub-branches.

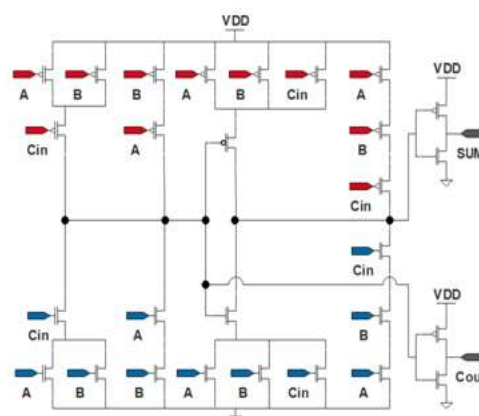


Figure 1. Conventional CMOS Adder Cell with 28 Transistors (28T).

IV. MULTIPLIERS UNDER-TEST (MUT) ARCHITECTURES

Built as multipart adder arrays, digital multipliers are found in many critical applications that needs to improve execution speed and reduce power dissipation.

In the literature, multipliers are introduced with many different architectures and algorithms with principally the same steps of processing. [1] The digital binary Half adders (HA) and Full adders (FA) are the basic modules to realize a digital multiplier. Moreover, the performance characteristics of each multiplier varies depending on the adopted algorithm.

One way to minimize multiplier's power consumption is by optimizing the used binary adder in terms of energy and power consumptions.

In this research, three basic popular multipliers have been chosen for the investigation and characterization; Bit-Array multiplier, Carry-Save multiplier, and Wallace-Tree multiplier.

The performance of each multiplier was evaluated by using static CMOS 28T adder. This adder was chosen since it is structured from a number of 28 MOS transistors which distinctly illustrates the effect of node technology downsizing and its consequences. Each MUT was implemented with operands of 4-bits size. In the following multipliers, X represents the multiplicand, Y represents the multiplier and Z indicates the product.

4.1 Carry-Save Array Multiplier

The circuit architecture of this multiplier has a regular structure. This design does not change the multiplication's result even when there is a carry bit generated to the output. Instead, the carry is passed diagonally downwards rather than in parallel fashion from right to left.

Each stage of partial product needs a vector-merging adder. The carry bits generated after each adder is saved for the next addition stage. Finally, the sums and carries are merged into carry-look-ahead adder.

This design has two advantages; the ease of pipelining, and the existence of only one critical path. Figure 2 illustrates design of this multiplier.

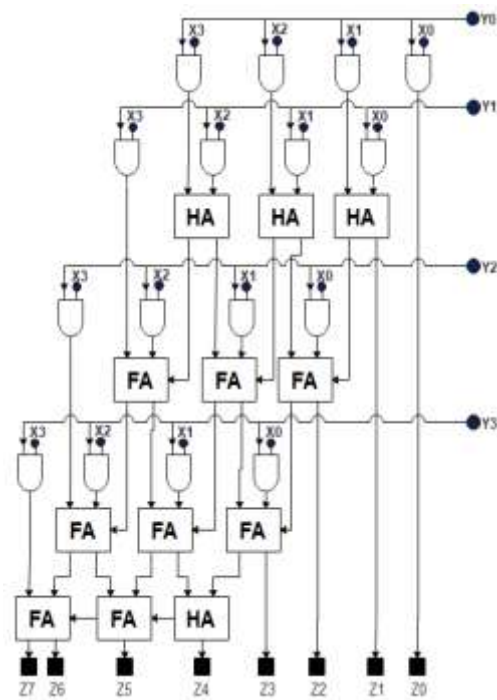


Figure 2: Carry-Save Array multiplier circuit architecture.

There are three delay times that control the total delay of this multiplier as shown in the following mathematical equation (4) assuming the propagation delay of the sum and carry generation are the same [1, 14]:

$$\Delta T = T_{and} + T_{final} + (S-1)T_{carry} \quad (4)$$

Where:

- T_{and} : Delay of the used AND gates for generating partial products,
- T_{final} : Delay of the final stage carry-lookahead adder,
- T_{carry} : Carry generating delay, and
- S: Number of partial product stages.

4.2 Bit-Array Multiplier

This multiplier is of generic type in which the carry bits propagate and get summed in the next adder cell instantly once it is generated from the previous one. Figure 3 shows a simple 4×4 bits multiplier which has regular structure and can be expanded to involve more bits easily. [1] The AND gates are used to generate partial products at each stage. The partial products are consequently shifted and added according to their bit orders. Based on the multiplier bits, the array multiplication should add all the generated partial products.

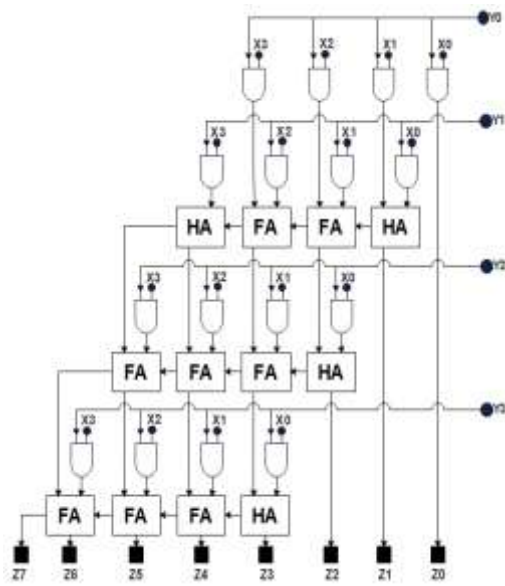


Figure 3: Bit-Array multiplier circuit architecture.

Just like in the previous Carry-Save one, this multiplier needs $X \times Y$ 2-bit AND gates in order to generate the needed partial products. This type of multipliers needs a large area. [14]

A simple routing can be implemented by shifting the partial products to an appropriate alignment, in which there is no need for any extra logic. Unlike the previous Carry-Save multiplier, this one has an architecture that produces many critical paths, which makes the propagation delay measurements a serious issue. The propagation delay equation can be approximated as [1, 14]:

$$\Delta T = T_{\text{and}} + T_{\text{sum}} + [(N-1)(M-2)]T_{\text{carry}} \quad (5)$$

Where:

- T_{and} : Delay of the used AND gates for generating partial products,
- T_{sum} : Full adder delay between the sum and the input carry bit,
- T_{carry} : Delay between the output and input carry, and
- N : Width of multiplicand X .
- M : Width of multiplier Y .

4.3 Wallace-Tree Multiplier

Wallace-Tree multiplier [1, 18, 19] has a tree structure. Figure 4 [1] shows a 4x4 Wallace-Tree multiplier based on AND gates and Full/Half Adders. There are mainly three steps to perform a multiplication operation:

1. Bit products formation.
2. Creating the Wallace tree by reducing number of bit products in step 1 into a two-row matrix.
3. The two rows matrix in step two are summed up to produce a product in fast carry-propagate adder fashion.

The behavior of this multiplier is different from the other mentioned ones above in a way that the ANDed terms are added all just before entering the full-adder array (Figure 4). This will result in an asymmetrical structure, which will shorten the longest path toward the final addition. In the final stage, an adder topology will be used to carry out the final result.

To illustrate the process in a clearer way, constructing and manipulating the Wallace-Tree is shown in Figure 5 as a transformation process of 4x4 bit multiplication with partial products presented as dots. For each partial product stage, it contains 4-bits width products as shown in Figure 5(a).

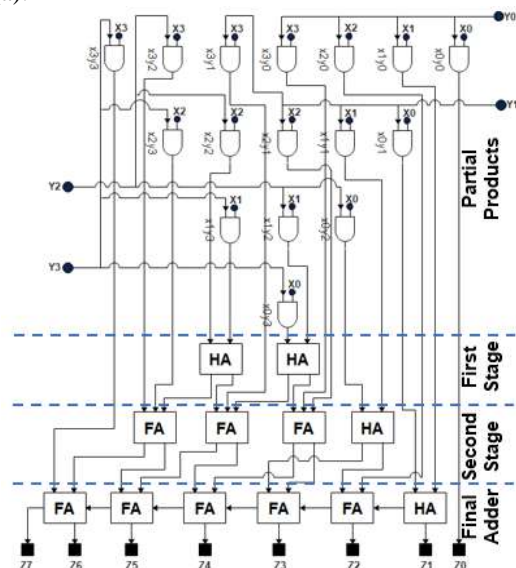


Figure 4: Wallace-Tree multiplier circuit architecture.

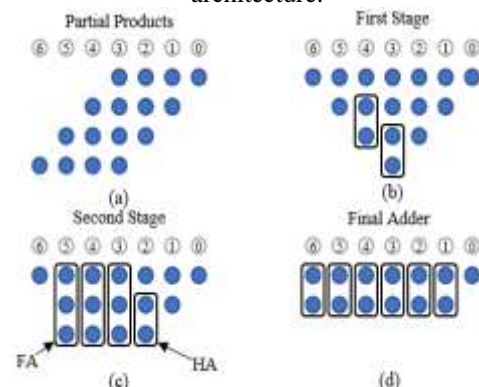


Figure 5: Transformation of (a) partial products into (b), (c), (d) Wallace Tree.

The main concept of this multiplier is to generate tree design with a minimum number of Full and Half adders. One way to do so is by starting with the deepest column number 3 and its neighbor column 4, in which one half adder is used in each (Figure 5(b)), then the resultant tree (Figure 5(c)) is

reduced more by introducing one half-adder and three full-adders. Eventually, the tree is reduced (Figure 5(c)) in which the final generated tree of depth two can be added using any conventional adder (Figure 4). [20]

Resulted from the 4x4 Wallace-Tree reduction process described above, the maximum delay will be only six adder delays, and thus the propagation delay will be of order $O(\log_{3/2}(N))$. [1] Despite the high performance of Wallace-Tree multiplier, its structure is complex and irregular, making it difficult to layout and cause a wastage in the occupied area and power dissipation.

V. SIMULATION METHODOLOGY

The operation and performance of each designed circuit was initially verified through simulation. The schematics of the designed circuits were realized as layouts, and then HSPICE editor was used to analyze the performance of each multiplier. Lastly, AvanWaves tool was used to display and analyze the results. Different SPICE model parameters were used according to the technology size based on the Predictive Technology Model (PTM). This model is developed by NIMO Group at Arizona State University (ASU) for each technology node and can be obtained from their website. [21]

At the simulation stage, a random of 10,000 inputs were generated to cover almost all possible transitions. A delay of 10ns was given between input signals in order to stabilize the voltage at the output.

All the multipliers were analyzed for the performance characteristics, namely; power consumption, propagation delay time and leakage current. For CMOS digital circuits, the power dissipation is estimated using the following equation [1]:

$$P_{avg} = P_{dynamic} + P_{short-circuit} + P_{static}$$

$$= V_{DD} \cdot f_{clk} \cdot \sum_i (\alpha_i \cdot V_{iswing} \cdot C_{iload}) + V_{DD} \sum_i I_{isc} + V_{DD} \cdot I_l \quad (6)$$

Where:

- f_{clk} is clock frequency,
- V_{iswing} : Swinging voltage at node i and approximately equals to V_{DD} ,
- C_{iload} : Capacitance at the output at node i ,
- α_i is the activity factor,
- I_{isc} : Short circuit current, and
- I_l : Leakage current.

Delays are measured by averaging the two propagation delays, (t_{PLH} and t_{PHL}). For each multiplier, 30 simulation runs were conducted, and

the worst value among the measurements was considered.

To determine the leakage current of each multipliers, we use a specific function in HSPICE, (i.e.: .MEAS t1 INTEGRAL power FROM=1ns TO=50ns). Conventionally, we cannot calculate the leakage current because each multiplier contains a large number of transistors (e.g. approximately up to 188 PMOS and 188 NMOS transistors).

The static power is divided into four intervals that are caused mainly by the leakage current (Figure 6). The static power over the intervals were integrated and the results were summed up which represents the total static power consumed in the multiplier. By dividing the total static power by the operating voltage source V_{DD} , the leakage current was obtained. Each of the four periods has 50ns interval as illustrated in Figure 6.

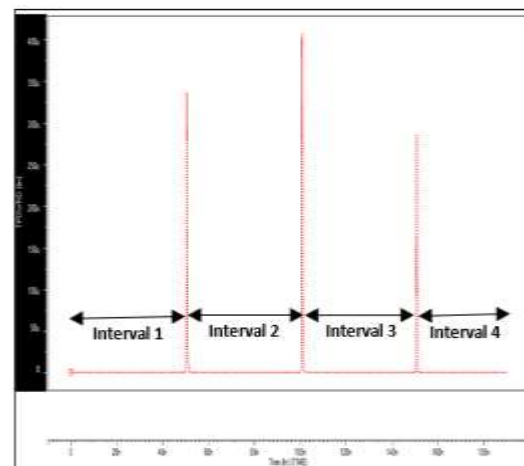


Figure 6: Static power intervals.

VI. SIMULATION RESULTS

In this section, performance measurements of all the three 4-bit multipliers at three technology nodes, 90, 65, and 22nm, using the conventional static CMOS adder are presented. These results were obtained from HSPICE simulations with one common index for all comparisons, in which the design constraints were the same for all the multipliers. While low power consumption is the objective of our designs, the delay and leakage current characteristics were measured as they are indicators of good performance. Each performance characteristic is measured for each technology node.

6.1. Power Consumption

In this subsection, the power measurement for all designed MUTs under different technology nodes for a variety of voltage supply levels and load capacitances are presented and discussed as the following.

6.1.1. Power Under Various Supply Voltages

The power consumptions for various supply voltages (V_{DD}) with a fixed load capacitance $C_L=10fF$ for all investigated technology sizes of the three MUTs are presented in

Table I and its related Figure 7. We find that at 90nm and for all the operands V_{DD} , Carry-Save consumed noticeably less power compared to the Bit-Array and Wallace-Tree multipliers.

Likewise, for technology node 65nm, when $V_{DD}=0.8V, 1V,$ and $1.4V,$ Carry-Save multiplier again consumed lowest power. However, when $V_{DD}=1.2V$ and $1.6V,$ Bit-Array multiplier was the lowest in power consumption. For the two technology nodes mentioned above, Wallace-Tree multiplier consumed largest power for all the operands $V_{DD}.$

Finally, for 22nm, when $V_{DD}=0.8V$ the Wallace-Tree multiplier consumed the lowest power, but when the V_{DD} increased, it became the largest power consumer. Bit-array became lowest power consumer when $V_{DD}=1V$ and above.

6.1.2. Power Under Various Load Capacitances

After comparing the power consumed in each multiplier based on V_{DD} operands, a comparison among the multipliers based on various load capacitance conditions for all the investigated technology sizes for a fixed $V_{DD}=0.8V$ is provided. The results are recorded in Table II and its Figure 8, we find that for 90nm technology node, the Carry-Save consumed the lowest power from $C_L=10fF$ to just before $C_L=50fF,$ and after that, the Wallace-Tree multiplier showed more efficiency than the other two. Whereas, the Wallace-Tree multiplier had the

largest power consumption in the range from $C_L=10fF$ until just before $C_L=40fF.$

For the same experiment but for 65nm multipliers, we find that when $C_L=10fF,$ both Carry-Save and Bit-Array multipliers consumed nearly the same amount of power, but when $C_L=20fF$ and $30fF,$ Carry-Save was the power efficient multiplier. However, Wallace-Tree became the most efficient multiplier when $C_L=40fF$ and above.

Finally, for the 22nm multipliers, we find that for all the operands $C_L,$ the Wallace-Tree multiplier consumed considerably less power compared to the other two multipliers, in which the Bit-Array was in the second place. This might be concluded from the previous experiment, when Wallace-Tree consumes least power at the operand $V_{DD}=0.8V,$ in which it shows better capability to drive large load capacitance C_L with least power consumption.

6.1.3. Power Versus Technology Node

Figure 9 shows the relationship between the power dissipation and both the technology sizes and supply voltages. Some conclusions can be obtained from it, the power dissipation increases by increasing the supply voltage. Furthermore, the power dissipation decreases by decreasing the technology size from 90nm to 65nm, and it increases by decreasing the technology size from 90nm to 22nm and from 65nm to 22nm, this is due to the increasing in the static power caused by increasing in the leakage current which has the largest amount at 22nm.

Table I. Power Consumption (mW) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF,$ for different technology sizes.

Multipliers Power Dissipation (mW)									
Node	90 nm			65 nm			22 nm		
V_{DD}	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
0.8	16.1	14.3	21.4	16.1	16.1	19.4	13.9	16.6	11.4
1	36.4	27.3	46.1	35.7	35.3	41.8	41.8	45.4	43.8
1.2	74.5	72.8	85.5	66.9	71.3	84.1	92.3	97.5	104
1.4	122	118	141	111.3	110	134.0	181	200	215
1.6	187	179	227	154.2	162	209.2	509	510	528

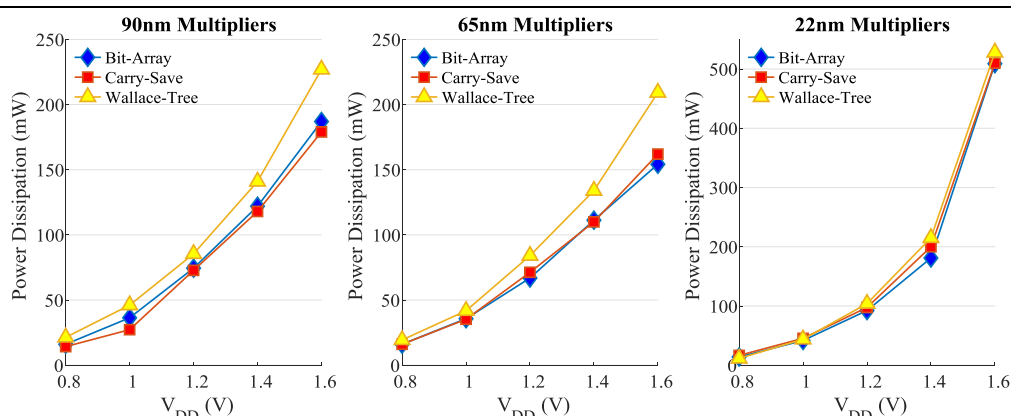


Figure 7: Power Consumption (mW) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10\text{fF}$, for different technology sizes.

Table II. Power Consumption (mW) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8\text{V}$, for different technology sizes.

Multipliers Power Dissipation (mW)									
Node	90 nm			65 nm			22 nm		
$C_L(\text{fF})$	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
10	16.1	14.3	21.4	16.1	16.1	19.4	13.9	16.6	11.4
20	22.8	21	23.3	17.25	17.2	20.3	17.8	20.6	15.9
30	24.5	21.6	27.2	20.7	19.9	21.2	20.2	23.5	16.9
40	28	24.6	27.3	22.4	22	21.5	23	23.7	18.5
50	30.8	28.9	28.4	26.5	22.6	22.1	23.2	24.9	18.7

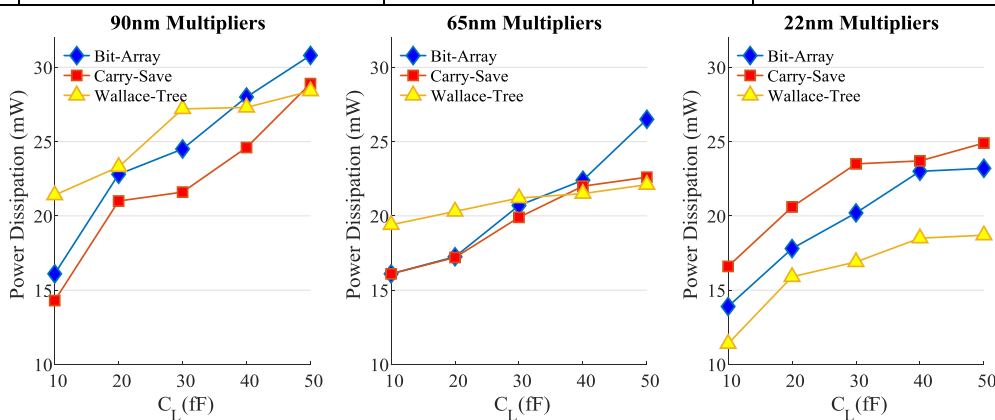


Figure 8: Power Consumption (mW) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8\text{V}$, for different technology sizes.

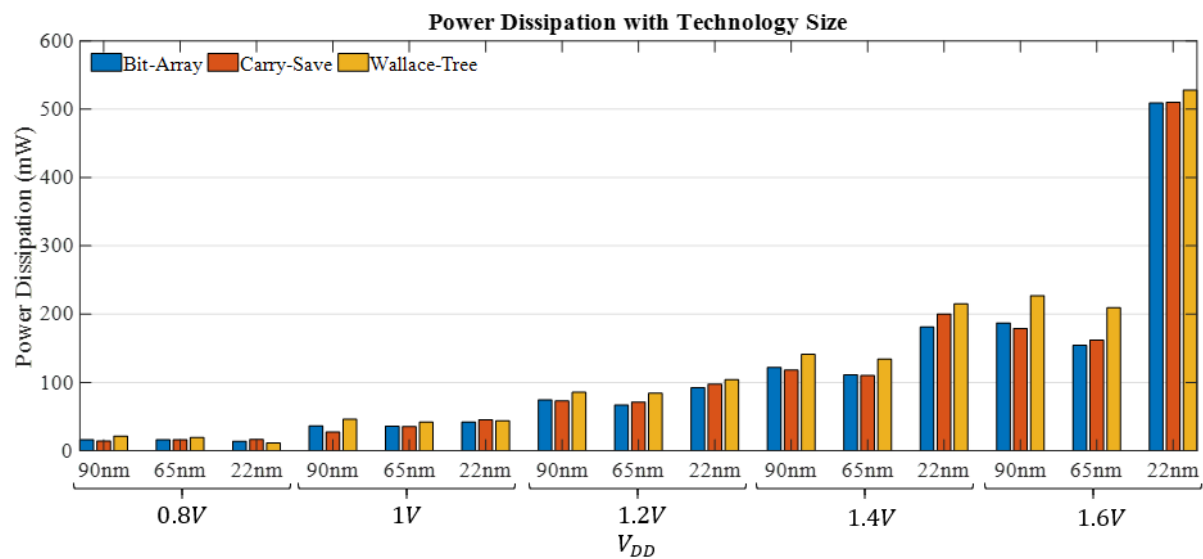


Figure 9: Multiplier Power Dissipation (mW) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF$ for different technology sizes.

6.2. Delay Time

Propagation delay is usually used as a measure of speed performance of a digital circuit. In this subsection, a delay time comparison among various multipliers used in our study at different technology sizes are presented, first using various supply voltages at fixed load capacitance C_L , and then fixing V_{DD} with various load capacitances.

6.2.1. Delay Under Various Supply Voltages V_{DD}

Table III shows the measured delay performance characteristics for the three MUTs for different technology sizes given with a capacitance load equals to 10fF and various supply voltages.

From the results in Table III and its Figure 10, we find that the delay data for Wallace-Tree is substantially less than the other two multipliers at 90nm technology node. The Carry-save multiplier has the largest propagation delay time.

At 65nm technology size, we see that the Wallace-Tree again had substantially less delay than the two other multipliers, in which the Carry-save came the next.

Finally, at 22nm technology size and when $V_{DD}=0.8V$ Bit-Array had the lowest delay time, but when $V_{DD}=1V$, Carry-Save multiplier had the lowest delay time, however we note that the Wallace-Tree outperformed the other two multipliers for operand $V_{DD}=1.2V$ and above.

6.2.2. Delay Under Various Load Capacitances C_L

Table IV shows the propagation delay time based on various load capacitances and a fixed $V_{DD}=0.8V$ for the MUTs implemented in various technology sizes.

The results from Table IV and its related Figure 11 show that the propagation delay time for Carry-Save is substantially more than the other multipliers at 90nm technology nodes. When the capacitance is in the range from $C_L=10fF$ to 34fF, Bit-array multiplier has the lowest propagation delay time, but when C_L is greater than 34fF, Wallace-Tree became the lowest propagation delay time multiplier.

At 65nm technology node, when the load capacitance is in the range from $C_L=10fF$ to 15fF, Wallace-Tree multiplier has the lowest propagation delay time, but when C_L is in the range from 15fF to 37fF Bit-Array became the lowest delay multiplier. Wallace-Tree returns to have the lowest propagation delay time when $CL=37fF$ and above.

Table III. Multiplier Delay Performance (ns) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF$ for different technology sizes.

Multipliers Delay Time (ns)									
Node	90 nm			65 nm			22 nm		
V_{DD}	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
0.8	0.420	0.467	0.420	0.382	0.415	0.35	0.238	0.254	0.270
1	0.365	0.405	0.345	0.325	0.330	0.279	0.158	0.146	0.203
1.2	0.335	0.380	0.320	0.286	0.305	0.262	0.124	0.132	0.120
1.4	0.312	0.365	0.305	0.249	0.284	0.241	0.105	0.115	0.092
1.6	0.307	0.335	0.295	0.245	0.280	0.237	0.053	0.0601	0.036

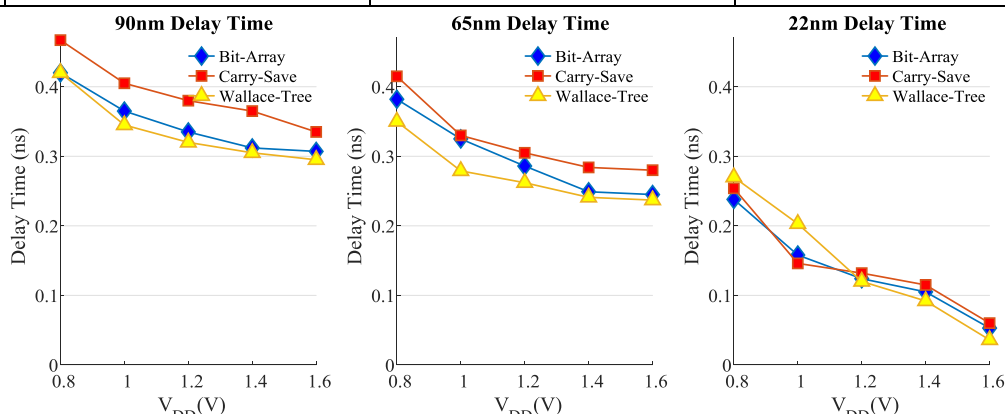


Figure 10: Multiplier Delay Performance (ns) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF$ for different technology sizes.

Table IV. Multiplier Delay Performance (ns) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8V$ for different technology sizes.

Multipliers Delay Time (ns)									
Node	90 nm			65 nm			22 nm		
C_L (fF)	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
10	0.420	0.467	0.420	0.382	0.415	0.352	0.238	0.254	0.270
20	0.450	0.535	0.465	0.392	0.458	0.415	0.294	0.278	0.278
30	0.478	0.545	0.486	0.426	0.458	0.432	0.332	0.335	0.313
40	0.542	0.617	0.526	0.452	0.460	0.447	0.400	0.398	0.383
50	0.583	0.642	0.552	0.535	0.575	0.479	0.429	0.411	0.4225

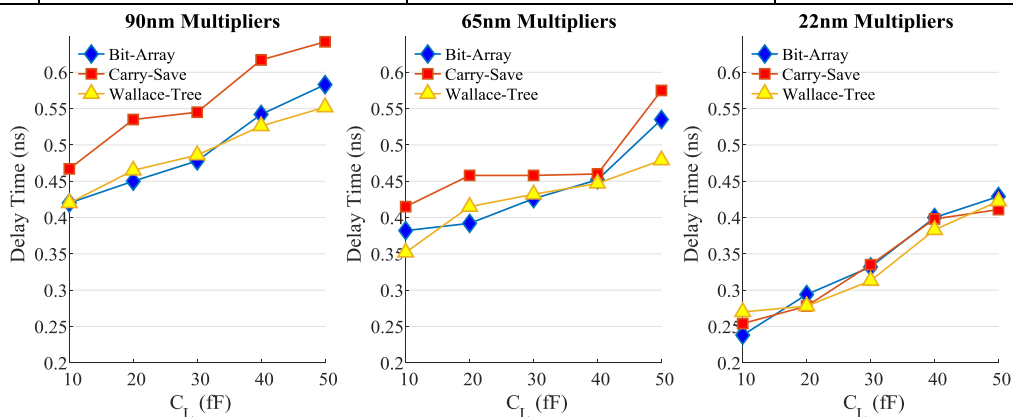


Figure 11: Multiplier Delay Performance (ns) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8V$ for different technology sizes.

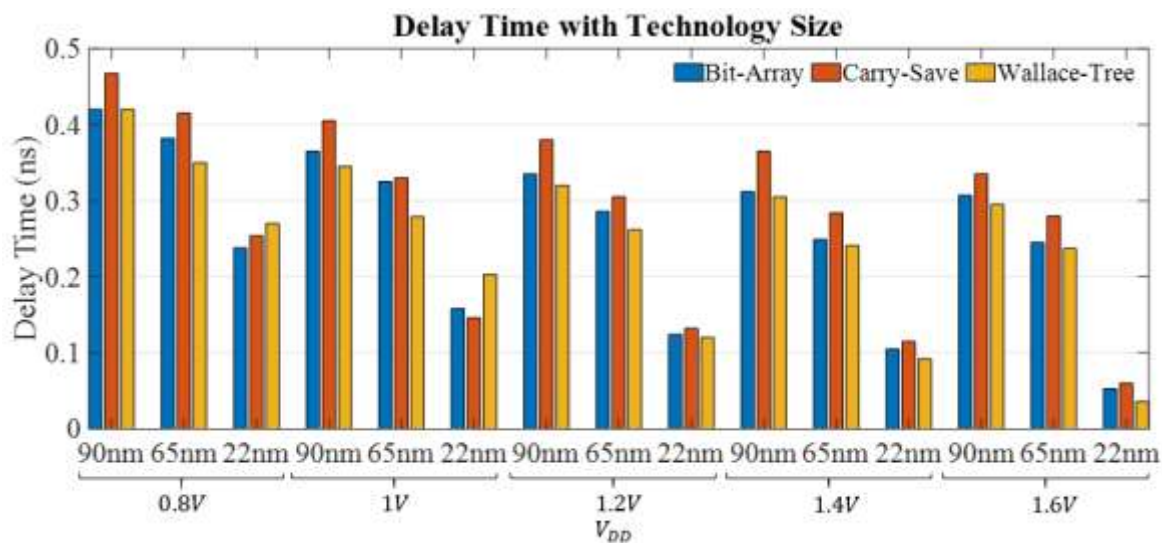


Figure 12: Multiplier Delay time (ns) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF$ for different technology sizes.

Finally, at 22nm technology size, Table IV and its related Figure 11 show that Bit-Array had the lowest propagation delay time in the range form $C_L=10fF$ to 15fF. Carry-Save multiplier had the lowest propagation delay time when the capacitance is in the range from $C_L=15fF$ to 20fF then it returned back to have the lowest delay time again when $C_L=40fF$ and above. Wallace-Tree had the lowest propagation delay time in the range form $C_L=20fF$ to 45fF.

6.2.3. Delay Versus Technology Nodes

The relationship between the delay time with both the technology sizes and supply voltages is shown in Figure 12, it is noted for such multipliers, the behavior of the delay time with supply voltage V_{DD} is inversely related, in that sense, as we increase the supply voltage less delay time appears, and vice versa. The relationship of the delay time with the technology size is as expected, the delay time decreases by decreasing the technology size and increasing the supply voltage V_{DD} .

6.3. Leakage Current

Leakage current characteristic play an important role in low power VLSI since it appears when transistors are in the switched-off mode, besides this current is responsible for the consumed static power in the circuit. In this subsection, we present a comparison based on the leakage current that appears in the three MUTs.

6.3.1. Leakage Under Various Supply Voltages

Table V shows the leakage current for the three multipliers at different technology sizes, with a load capacitance equals to 10fF at various supply voltages.

As we see in Table V and its Figure 13, at 90nm technology node, Wallace-Tree had the lowest leakage current change according to the changing in supply voltage in the range form $V_{DD}=0.8V$ to 25V. However, in the range from $V_{DD}=1.25V$ to 1.52V, Carry-Save had lowest leakage current, nevertheless when $V_{DD}=1.52V$ and more, Bit-Array became the multiplier with the lowest leakage current.

At 65nm technology size, both the Wallace-Tree and Bit-Array multipliers has the lowest leakage current, but as the supply voltage increases, Bit-Array became the lowest leakage current consumer than the other two multipliers.

Finally, at 22nm technology size, we find that for all V_{DD} operands, Carry-Save multiplier consumed considerably less leakage current compared to the other multipliers. Noting that differences between them didn't exceeded 4.640%.

6.3.2 Leakage Under Various Load Capacitances

Table VI shows the leakage currents for the multipliers at different technology sizes, with various load capacitances and a constant supply voltage equals to 0.8V.

As we can see in Table VI and its Figure 14 for all the operands C_L 's at 90nm technology node that the Wallace-Tree multiplier has considerably less leakage current compared to the Bit-Array and Carry-Save multipliers. Noting that the differences between them did not exceed 17.900%.

For the multipliers at 65nm technology size, once again Wallace-Tree multiplier has considerably less leakage current compared to the Bit-Array and Carry-Save multipliers, for all the

operand C_L . Noting again that the differences between them did not exceed 8.250%.
 Finally, at 22nm technology size, Bit-Array and Carry-Save multipliers have considerably less leakage current compared to Wallace-Tree multiplier. Noting that when C_L equals to 10fF, 20fF and 40fF, Bit-Array and Carry-Save multipliers have

the lowest and almost equal leakage current, when $C_L=30$ fF Carry-Save has lower leakage current than Bit-Array multiplier by about 0.704%. But when $C_L=50$ fF, Bit-Array leakage current is less than Carry-Save one by 0.568%.

Table V. Multiplier Leakage Current (pA) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10$ fF for different technology sizes.

Multipliers $L_{Leakage}$ (pA)									
Node	90 nm			65 nm			22 nm		
V_{DD}	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
0.8	0.148	0.163	0.142	0.161	0.165	0.161	0.134	0.134	0.137
1	0.158	0.164	0.157	0.189	0.193	0.192	0.897	0.884	0.922
1.2	0.189	0.194	0.178	0.258	0.267	0.263	6.30	6.17	6.47
1.4	0.286	0.232	0.292	0.444	0.463	0.454	36.5	35.8	37.1
1.6	0.574	0.604	0.588	0.961	1.150	0.985	16.4	16.2	16.8

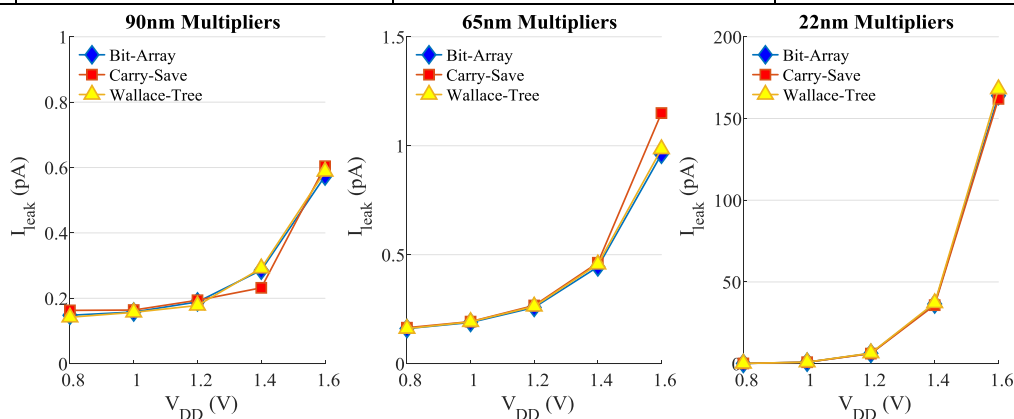


Figure 13: Multiplier Leakage Current (pA) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10$ fF for different technology sizes.

6.3.3. Leakage Versus Technology Node

Figure 15 shows the relationship between the leakage current with both the technology size and supply voltage. From the figure shown, we note that the relationship of the leakage current with supply voltage is dramatically rising, in that sense, as we increase supply voltage, leakage current increases. The leakage current is inversely related to

the technology size; it increases by decreasing the technology size and increasing the supply voltage as well. However, it is clearly shown that the leakage current highly increased for all the multipliers at 22nm node and especially when the supply voltage (V_{DD}) is greater than 1.2V. Hence, the power consumption will be noticeably increased at those parameters.

Table VI. Multiplier Leakage Current (pA) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8$ V, for different technology sizes.

Multipliers $L_{Leakage}$ (pA)									
Node	90 nm			65 nm			22 nm		
C_L (fF)	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree	Bit-Array	Carry-Save	Wallace Tree
10	0.148	0.163	0.142	0.161	0.165	0.161	0.134	0.134	0.137
20	0.173	0.179	0.149	0.169	0.174	0.163	0.135	0.135	0.139
30	0.192	0.201	0.165	0.182	0.187	0.172	0.142	0.141	0.146
40	0.213	0.227	0.187	0.198	0.206	0.189	0.155	0.155	0.160
50	0.239	0.255	0.212	0.221	0.230	0.214	0.175	0.176	0.181

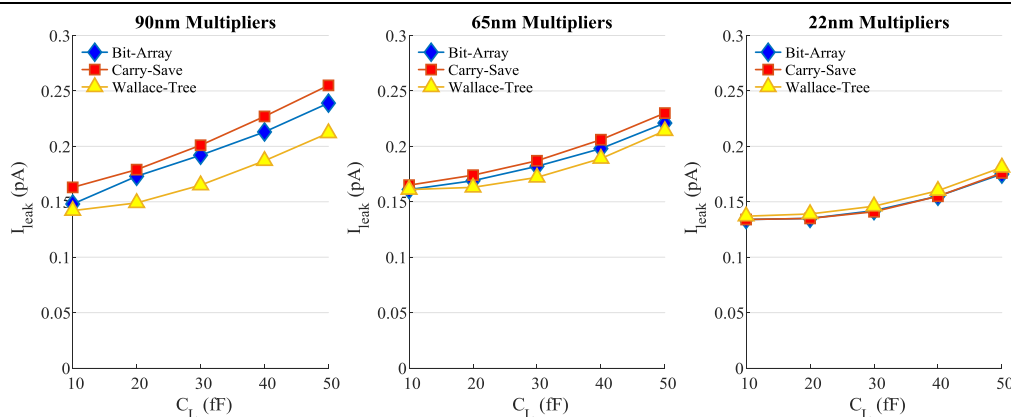


Figure 14: Multiplier Leakage Current (pA) of all the multiplier circuits with varying C_L at fixed $V_{DD}=0.8V$, for different technology sizes.

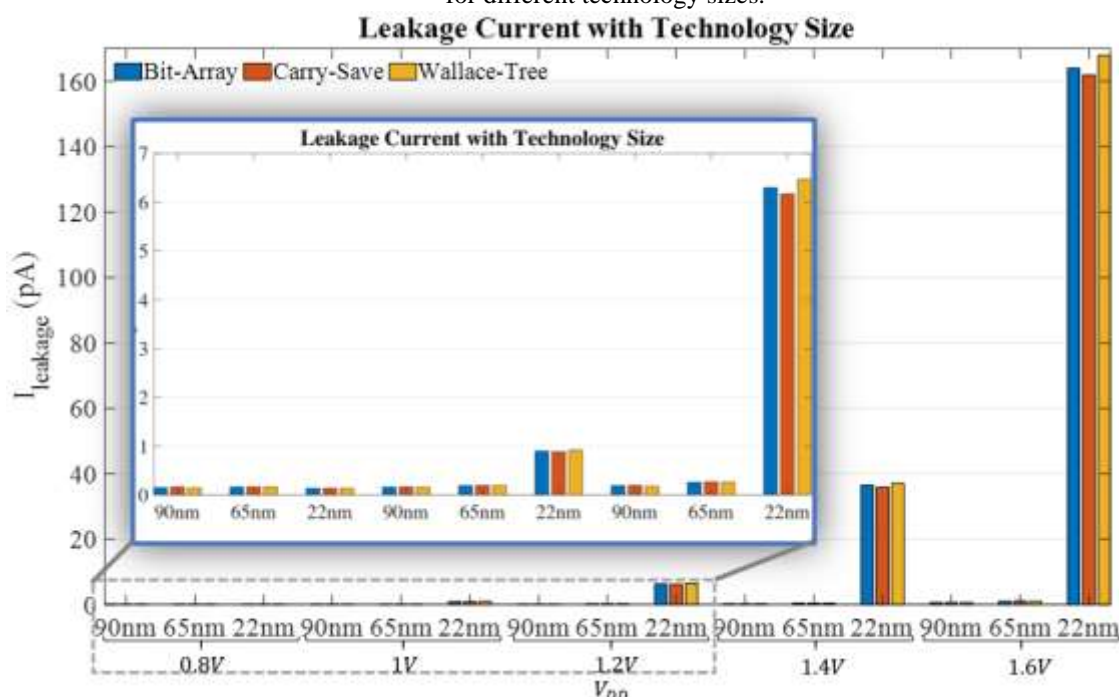


Figure 15: Multiplier Leakage current (pA) of all the multiplier circuits with varying V_{DD} at fixed $C_L=10fF$ for different technology sizes.

VII. CONCLUSIONS

This research has analyzed, evaluated, and compared the power consumption, speed performance, and leakage current characteristics for three popular digital multipliers implemented using the conventional static CMOS 28T adder. Accordingly, we analyzed the three 4×4 multiplier architectures, namely: Carry-Save, Bit-Array, and Wallace-Tree. In order to make this research study comprehensive, three technology nodes have been investigated for these designs: 90nm, 65nm, and 22nm. All for various supply voltage levels and load capacitances.

From the examined characteristics for 90nm technology node, Carry-Save showed better power performance when compared to Wallace-Tree and Bit-Array multipliers. However, Wallace-Tree

multiplier exhibited better delay time performance and leakage current when compared to the other two multipliers.

For the same multipliers, but with 65nm technology node, Bit-Array showed better power and leakage characteristics for various operand V_{DD} when compared to the other two multipliers, while Wallace-Tree has better delay and leakage performance when varying the operand C_L .

Now for all the measured characteristics of the 22nm technology node, Wallace-Tree exhibited the best performance among the other multipliers, making it the best choice for smaller technology sizes. At this node, the results exhibited no big difference in multipliers' performance when compared for leakage current characteristic.

These results can give the VLSI digital multiplier designers a closer look upon which algorithm fits better for their future design in the realm of low-power applications under various conditions.

REFERENCES

- [1]. J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits, 2 ed. Pearson, 2013.
- [2]. Sambit. (July 3, 2019). Global FinFET Technology Market 2024 Growth Analysis by Manufacturers, Regions, Type and Application, Forecast Analysis. Financial Planning.
- [3]. D. Kudithipudi, P. Nair, and E. John, "On estimation and optimization of leakage power in CMOS multipliers," in 2007 50th Midwest Symposium on Circuits and Systems, 2007, pp. 859-862.
- [4]. G. Ma and F. J. Taylor, "Multiplier policies for digital signal processing," IEEE ASSP Magazine, vol. 7, no. 1, pp. 6-20, 1990.
- [5]. D. R. Gandhi and N. N. Shah, "Comparative analysis for hardware circuit architecture of Wallace tree multiplier," in 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, pp. 1-6.
- [6]. S. Shah, A. J. Al-Khalili, and D. Al-Khalili, "Comparison of 32-bit multipliers for various performance measures," in ICM 2000. Proceedings of the 12th International Conference on Microelectronics. (IEEE Cat. No.00EX453), 2000, pp. 75-80.
- [7]. I. S. Abu-Khater, A. Bellaouar, and M. I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," IEEE Journal of Solid-State Circuits, vol. 31, no. 10, pp. 1535-1546, 1996.
- [8]. S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, 2017.
- [9]. M. Jhamb, Garima, and H. Lohani, "Design, implementation and performance comparison of multiplier topologies in power-delay space," Engineering Science and Technology, an International Journal, vol. 19, no. 1, pp. 355-363, 2016/03/01/ 2016.
- [10]. A. Bellaouar and M. I. Elmasry, Low-Power Digital VLSI Design: Circuits and Systems. Kluwer Academic Publishers, 1995.
- [11]. D. Radhakrishnan, "Low voltage CMOS full adder cells," Electronics Letters, vol. 35, no. 21, pp. 1792-1794, 1999.
- [12]. M. Liao, C. Su, C. Chang, and A. C. Wu, "A carry-select-adder optimization technique for high-performance Booth-encoded Wallace-tree multipliers," in 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353), 2002, vol. 1, pp. I-I.
- [13]. L. Raja, B. M. Prabhu, and K. Thanushkodi, "Design of Low Power Digital Multiplier Using Dual Threshold Voltage Adder Module," Procedia Engineering, vol. 30, pp. 1179-1186, 2012/01/01/ 2012.
- [14]. D. Kudithipudi and E. John, "Implementation of Low Power Digital Multipliers using 10 - Transistor Adder Blocks," J. Low Power Electronics, vol. 1, pp. 286-296, 2005.
- [15]. A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," IEEE Journal of Solid-State Circuits, vol. 27, no. 4, pp. 473-484, 1992.
- [16]. J. P. Uyemura, CMOS Logic Circuit Design, 1 ed. Springer US, 2001.
- [17]. N. Weste and K. Eshraghian, Principles of CMOS VLSI design: a systems perspective. Addison-Wesley, 1993.
- [18]. A. Skavantzoz and P. B. Rao, "New multipliers modulo 2^N-1 ," IEEE Transactions on Computers, vol. 41, no. 8, pp. 957-961, 1992.
- [19]. C. S. Wallace, "A Suggestion for a Fast Multiplier," IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14-17, 1964.
- [20]. S. e. Abed, Y. Khalil, M. Modhaffar, and I. Ahmad, "High- performance low- power approximate Wallace tree multiplier," International Journal of Circuit Theory and Applications, vol. 46, no. 12, pp. 2334-2348, 2018.
- [21]. Y. K. Cao. Nanoscale Integration and Modeling (NIMO) Group, ASU. Available: <http://ptm.asu.edu/>

Hani O. Jamleh" Comparative Performance Study of Digital Multipliers under Various Conditions in Nanometer SPICE Technology" International Journal of Engineering Research and Applications (IJERA), Vol. 09, No.08, 2019, pp. 53-65