WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

An Area Efficient and High Throughput 32-bit IoT Processor Design Using Monotonic-Static CMOS

Suruchi Rajput*[*a], Divyanshu Rao[b], Ravimohan[c]

[a]*M. Tech. Schohlar,Department of ECE, SRIT,Jabalpur, India*
[b,c]*Assistant Professor,Department of ECE, SRIT,Jabalpur, India*

A B S T R A C T

The arithmetic logic unit (ALU) is the core of a CPU in a computer. In ALU, adders play a major role not only in addition but also in performing many other basic arithmetic operations like subtraction, multiplication, etc. The most executed operation in the data path is addition, which requires a binary adder that adds two given numbers. Adders also play a vital role in more complex computations like multiplication, division and decimal operations. Hence, an efficient implementation of binary adder is crucial to an efficient data path. Using the Monotonic-Static CMOS (MS-CMOS) logic, A IoT processor ALU has been designed with a feature size of 25nm. MS-CMOS is a low power, high speed logic family which can be seen as an intermediate logic design style between standard Static CMOS and Dynamic CMOS. In the present work we have shown successful implementation of Critical units of the ALU using MS-CMOS logic while rest of the modules are implemented using static CMOS. The designed ALU operates at a frequency of 1.25 GHz with a dual supply of 1.2-0.6 Volt. A modified carry-lookahead adder is used in the ALU design which is considerably faster than ripple carry adder. Since Adder is the main performance bottleneck in the ALU design, the improved adder design reduces critical path delay and improves overall performance. Also, concept of dual power supply is applied to reduce power consumption of whole circuit. Higher power supply (1.2V) is used to drive the arithmetic unit components which lie in the critical path whereas lower power supply (0.6V) drives the logical unit. This saves nearly 75% of the power consumption in logical unit and accounts for more than 18% of the total power consumed in ALU. The designed ALU, performs basic arithmetic operations like addition, subtraction, increment, decrement, transfer and logical operations like AND, OR, XOR, NOT with logical, arithmetic and circular shifts in both direction

Keywords: Arithmetic logic unit, Monotonic-Static -CMOS, CPU, VLSI, Dynamic CSMOS

1. Introduction

With the ever-increasing demand of high performance and low power digital processors, it is becoming more and more difficult for traditional design approaches to meet the system level specifications. Migration to newer technology process after a regular interval has only made this gap wider. This has led designers to think about novel design approaches and apply them in their design flow. This thesis **explores** the possibility of implementing MS-CMOS logic in the design of an arithmetic logic unit (ALU).An Internet of Things Device processor (IOT) is a microprocessor particularly to process the digital signals. It has a specialized architecture which is perfect for the fast operational requirement of digital signal processing. An Internet of Things Device processor (IOT) is particularly for those applications that can't tolerate delays because the main feature of IOT is to process the data in real time. Internet of Things Device processors take a digital signal and improves the quality of that signal. For ex. it makes the sound very clear of that digital signal, gives the faster data or sharper images. Internet of Things Device processors use that type of signals that have been digitized like video, voice, audio, temperature or position signals etc and then manipulate mathematically. To perform these mathematical functions rapidly IOT is designed. So the information contained in signals can be displayed or converted to another type of signal after the process of the signals.There are number of mathematical operations required in IOT algorithm to perform fast operations and repeatedly on a series of data samples. In IOT systems firstly analog signals that may be audio or video is converted into digital signals, then manipulated digitally, and again it is converted into analog signals.

Analog Devices' 32-Bit Floating-Point IoT Processors are based on a Super Harvard architecture that balances exceptional core and memory performance with outstanding I/O throughput capabilities. This "Super" Harvard architecture extends the original concepts of separate program and data memory busses

* *Corresponding author.* Tel.: +91-7389973999
E-mail address: rajputsuruchisingh@gmail.com

by adding an I/O processor with its associated dedicated busses. In addition to satisfying the demands of the most computationally intensive, real-time signal-processing applications, SHARC processors integrate large memory arrays and application-specific peripherals designed to simplify product development and reduce time to market.

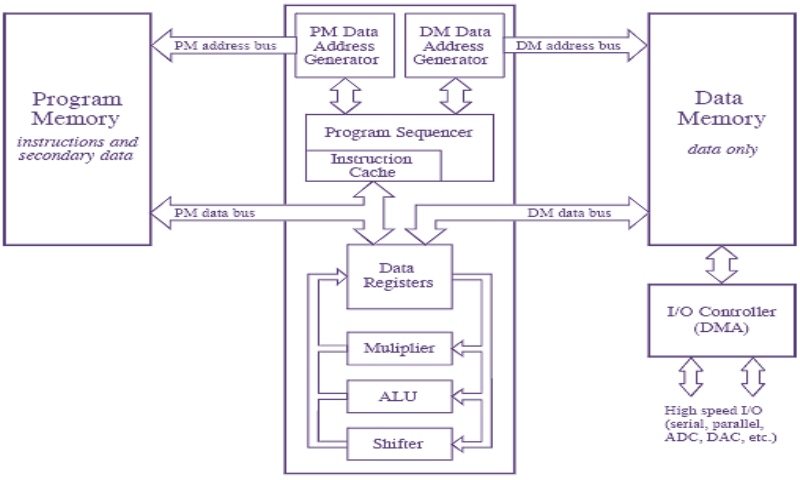


Figure 1: Analog Devices SHARC IOT.

CMOS logic family has always been a preferred choice among designers to implement their complex designs which usually consists millions of transistors. This is because CMOS technologies in general show highest densities and lowest power per gate [9]. They are easier to 'scale-down' thus favouring technology updateability. Within CMOS logic family various sub-families have evolved over time, namely static and dynamic CMOS. Static CMOS can be characterized as the one having low power consumption and robust design whereas dynamic CMOS is known for its high speed at the cost of higher power consumption. This chapter explores various aspects of Monotonic Static CMOS sub-family which is an intermediate class between static and dynamic.

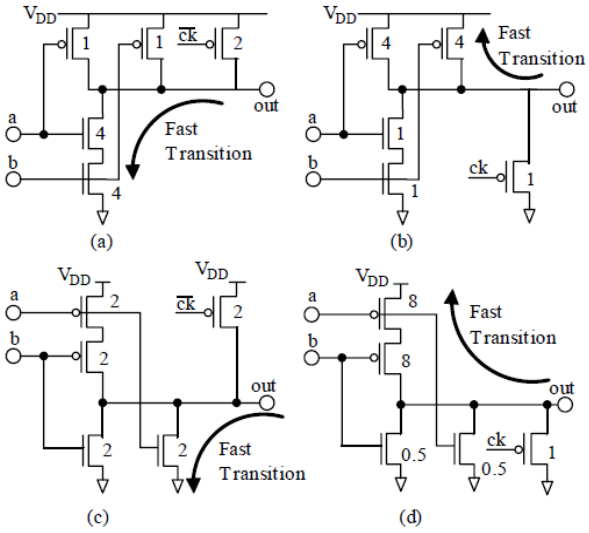


Figure 2Four basic MS-CMOS logic gates (a) LS-NAND (b) HS-NAND (c) LS-NOR, and (d) HS-NOR

In above figure 1, 4, 2 and 0.5 is the Skew Ratio, Skew ratio (SR) is defined using a static CMOS gate with roughly matched rise and fall time as reference [7]. It is ratio of transistor width of a MS-CMOS gate to the corresponding transistor width of a standard static CMOS. For LS gate skew ratio is ratio between NMOS transistor width of MS-CMOS gate to standard static CMOS gate. All PMOS of LS MS-CMOS gates are scaled down with the same skew ratio or to their minimum size allowed by technology rules. In other words, size of NMOS transistors of LS MS-CMOS gates are obtained by multiplying standard NMOS size of static CMOS with skew ratio and size of PMOS transistors is obtained by dividing PMOS size of static CMOS with skew ratio. For a HS gate PMOS transistor width is obtained by multiplying and NMOS transistor width by dividing. Note that a MS-CMOS gate with SR=1 is simply a static CMOS (unskewed one).As illustrated in Figure 2, MS-CMOS logic circuit has both low-skewed (LS) and high skewed (HS) NAND/NOR gates. The trip point of LS gate is lower than Vdd / 2 and that of HS gate is higher than Vdd / 2 [8]. LS gates are precharged when Φ is low and evaluated when Φ is high. HS gates are similar to LS gates except the phase. Like dynamic circuits, the precharge time can be hidden by using multiphase clocks [Harris]. As shown in Figure 2, the evaluation path of skewed gates can consist of either parallel connected transistors (EP), or serially connected transistors (ES). In

MS-CMOS logic LS NOR gate and HS NAND gates are preferred to get high-speed evaluation. This is because they are of EP type which is essentially faster than ES type gates [8].

2. MS-CMOSbased ALUDesign

The arithmetic logic unit (ALU) is a digital circuit that performs an arithmetic operation (addition, subtraction, etc.) and logic operations (Exclusive-OR, AND, etc.) between two numbers. The ALU is a fundamental building block of the central processing unit of a computer. In this chapter design methodology of MS-CMOS ALU is presented, which can operate on a IoT processor (8 byte) quad-word. Also, a detailed overview of dual supply implementation is given which is used to reduce the power consumption of the whole unit by exploiting the timing slack present in different subsections.Arithmetic Logic Unit (ALU) is the part of a computer processor (CPU) that carries out arithmetic and logic operations on the operands in computer instruction words. In most of the processors, ALU is divided into two units, an arithmetic unit (AU) and a logic unit (LU). Some processors contain more than one AU - for example, one for fixed-point operations and another for floating-point operations. In personal computers floating point operations are sometimes done by a floating-point unit on a separate chip called a numeric coprocessor.

In general, the ALU includes storage places for input operands, operands on which calculations has to be performed, the accumulated result which is stored in an accumulator, and shifted results. The size of the accumulator indicates the word size of the processor. The flow of bits and the operations performed on them in the subunits of the ALU is controlled by gated circuits. The gates in these circuits are controlled by a sequence logic unit that uses a particular algorithm or sequence for each operation code. In the arithmetic unit, multiplication and division are done by a series of adding or subtracting and shifting operations. An ALU must process numbers using the same format as the rest of the digital circuit. There are several ways to represent negative numbers. Modern processors normally use two's complement binary number representation for negative numbers. Early computers used a wide variety of number systems, including one's complement, sign-magnitude format, and even true decimal systems. ALU's for each one of these number systems had different designs, and this is reason two's complement system is preferred over others as it requires less complex hardware to implement addition and subtraction. The design of the ALU is obviously a critical part of the processor and new approaches to speeding up instruction handling are continually being developed.

ALU Architecture: Basic task of an ALU is to perform a set of arithmetic and logical operations on of its inputs. This operation may be on both the operands (inputs) or on single operand depending upon the type of operation to be performed. Among the many functions that it can perform, the desired one is chosen by select signals. Therefore, number of select signals depends on the number of different functions that a particular ALU can perform. Implementation of all the functions and interconnection between its different components is decided by the architecture of the ALU. Performance of any ALU largely depends upon its architecture. Even a small flaw at the architectural level, cannot be overcome by any level of circuit design expertise. Therefore, role of architecture of an ALU in its performance and efficiency is very important. The architecture of designed ALU is presented in Figure 3.

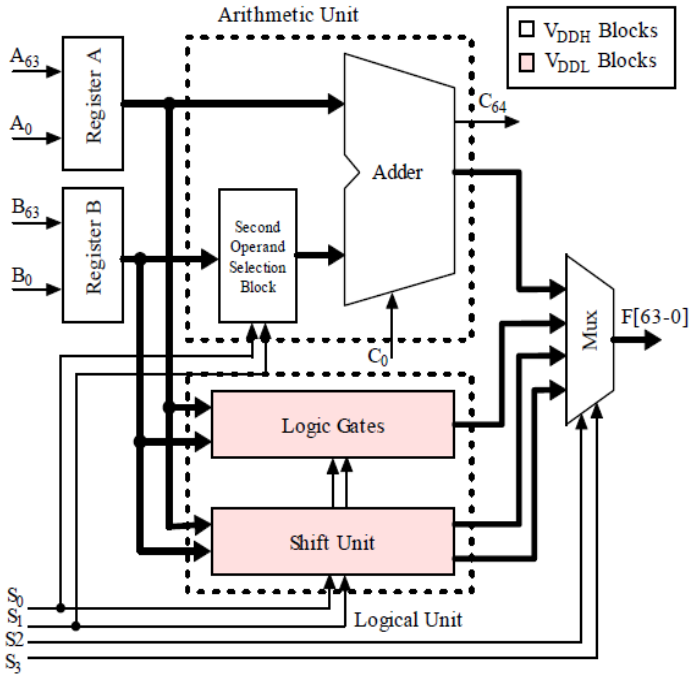


Figure 3 Architecture of the designed IoT processor ALU

The design of the ALU can be divided into two main parts- Arithmetic Unit and Logical Unit. In the arithmetic unit (AU) inputs A and B are fetched from two separate registers, with A being directly fed to the input of adder unit. B is connected to second operand selection block which computes second input to the adder depending upon two lower order select signals, enabling various arithmetic operations to be performed. Logical Unit (LU) performs logical operations between two IoT processor input values along with the left and right shift operation. There are four select signals provided which are used to select a particular operation of ALU. Complete list of ALU functions and corresponding select signals is given in Table 1. The upper two select signals S3

and S2 are used to select the type of operation whether arithmetic, logical, shift right or shift left. Lower two select signals S1 and S0 selects the specific operation among them. For arithmetic operations input Cin also works as a select signal and generates two different output for Cin = 0 and Cin = 1 as shown in Table 1. For logical and shift operations Cin has no effect on output and is denoted by don't care symbol (x).

Table 1: Function Table of the designed ALU

Operation Select					Operation	Function
S_3	S_2	S_1	S_0	C_{in}		
0	0	0	0	0	$F = A + B$	Addition
0	0	0	0	1	$F = A + B + 1$	Add with carry
0	0	0	1	0	$F = A + \overline{B}$	Subtract with borrow
0	0	0	1	1	$F = A + \overline{B} + 1$	Subtraction
0	0	1	0	0	$F = A$	Transfer A
0	0	1	0	1	$F = A + 1$	Increment A
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	Transfer A
0	1	0	0	x	$F = A \wedge B$	AND
0	1	0	1	x	$F = A \vee B$	OR
0	1	1	0	x	$F = A \oplus B$	XOR
0	1	1	1	x	$F = \overline{A}$	Complement A
1	0	0	0	x	$F = shr\ A$	Logical Right Shift
1	0	0	1	x	$F = ashr\ A$	Arithmetic Right Shift
1	0	1	0	x	$F = cir\ A$	Circular Right Shift
1	0	1	1	x	---	---
1	1	0	0	x	$F = shl\ A$	Logical Left Shift
1	1	0	1	x	$F = ashl\ A$	Arithmetic Left Shift
1	1	1	0	x	$F = cil\ A$	Circular Left Shift
1	1	1	1	x	---	---

A 4x1 multiplexer is placed at the output to choose among arithmetic, logical, shift right and shift left output. The output values are selected at bit-level using two higher order select signals (S3 and S2). The designed ALU provides a choice of eight arithmetic, four logical and six shift operation. The first eight are arithmetic operations and are selected with S3S2 = 00. The next four are logical operations, selected by S3S2 = 01. Shift right operations are selected by combination S3S2 = 10 while S3S2 = 11 selects shift left operation. Task of multiplication and division can be performed by repeated additions or subtractions respectively. Alternatively, a separate block can be optimally designed to carry out multiplication and division since they are both, most time consuming and power consuming operations.

Arithmetic Unit: The basic component of an arithmetic unit is the parallel adder and second operand selection block. By controlling data inputs to the adder, it is possible to obtain different types of arithmetic operations. In the previous chapter discussion about various issues of adder design and implementation was presented. In this section we shall talk about interconnection of adder block with other blocks to perform arithmetic operations. Figure 4illustrates the design components of arithmetic unit [2].

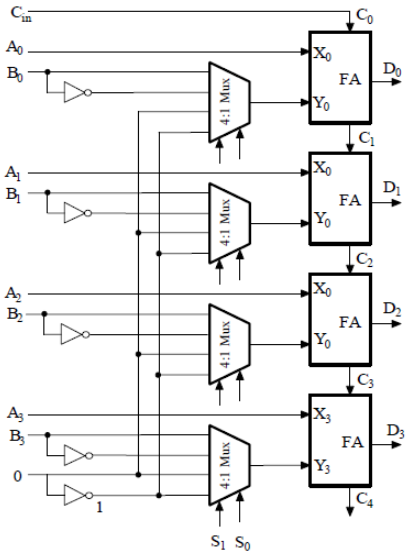


Figure 4 Design of 4-bit arithmetic unit

The above diagram is for 4-bit data and contains four full adder circuits that constitute the 4-bit adder and four multiplexers in second operand selection block for choosing different operations. There are two 4-bit inputs A and B and a 4-bit output D. The four inputs from A go directly to the X inputs of the

binary adder. Each of the four inputs from B is connected to the data inputs of the multiplexers. The multiplexers data input also receive the complement of B. The other two data inputs are connected to logic-0 and logic-1. Logic-0 input can be connected to ground and logic-1 input to supply voltage. The four multiplexers are controlled by two selection inputs, S1 and S0. Transistor level schematic of a two-level 4:1 multiplexer is shown in Figure 5.

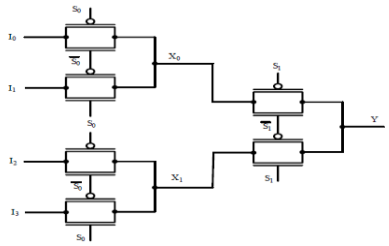


Figure 5 Transistor-level schematic of a 4:1 MUX

The input carry $C_{in}$is connected to the carry input of the full adder in the least significant position. The output of the binary adder is calculated from the following arithmetic sum:

$D = A + Y + C_{in}$

where A is the 4-bit binary number at the X inputs and Y is the 4-bit binary number at the Y inputs of the binary adder. $C_{in}$is the input carry, which can be equal to 0 or 1. Note that the symbol + in the equation above denotes an arithmetic plus. By controlling the value of Y with the two selection inputs $S_1$and $S_0$and making C_{in}equal to 0 ot 1, it is possible to generate the eight arithmetic operations listed in Table 1.

When S1S0 = 00, the value of B is applied to the Y inputs of the adder. If C_{in}= 0, the output D = A + B. If C_{in}= 1, output D = A + $\overline{B}$+ 1. Both cases perform the add operation with or without adding the input carry. For S_1S_0= 01, the complement of B is applied to the Y inputs of the adder. If C_{in}= 1 then D = A + B +1. This produces A plus 2's complement of B, which is equivalent to A – B. When C_{in}= 0, then D = A +$\overline{B}$. This is equivalent to a subtract with borrow, that is, A – B – 1. When S_1S_0= 10, the inputs from B are neglected, and instead, all 0's are inserted into the Y inputs. The output becomes D = A + 0 + C_{in}. This gives D = A when Cin = 0 and D = A + 1 when C_{in}=1. In the first case we have a direct transfer from input A to output D. In the second case, the value of A is incremented by 1. When S_1S_0= 11, all 1's are inserted into the Y inputs of the adder to produce the decrement operation D = A – 1 when C_{in}= 0.

This is because a number with all 1's is equal to the 2's complement of 1 (the 2's complement of binary 0001 is 1111). Adding a number,A to the 2's complement of 1 produces D = A + 2's complement of 1 = A – 1. When C_{in}= 1, then D = A – 1 + 1 = A, which causes a direct transfer from input A to output D. Note that the operation D = A is generated twice, so there are only seven distinct operations in the arithmetic unit.

Logical Unit:In the architecture of the designed ALU, logical unit can be broadly divided into two parts. First block is of logic gates and is used for performing logical operations between two input operands. Second block is called shift unit and contains circuitry for performing various shift operations. Structure of one-bit logic gate is shown in Figure 6.

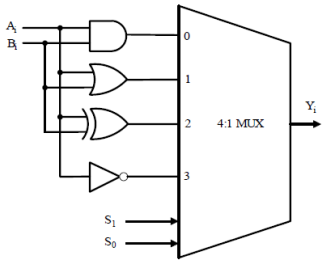


Figure6 one-bit logic gate structure

Although there are many logic operations, most computers use only four – AND, OR, XOR, and complement – from which all others can be derived. Designed logical unit performs these four basic operations. It consists of four gates and a multiplexer as shown in Figure 6 Each of the four logic operations is generated through a gate that performs the required logic. The outputs of the gates are applied to the data inputs of the multiplexer. The two selection inputs $S_1$and S_0choose one of the data inputs of the multiplexer and direct its value to the output. The diagram shows one typical stage with subscript i. For the logic circuit with 64 bits, this stage is repeated 64 times for i = 0, 1, 2,…, 62, 63. The selection inputs are applied to all stages. Figure 7 shows transistor level implementation of NOT, AND, and XOR. Note that OR operation can be derived from NAND gate with inverted inputs.

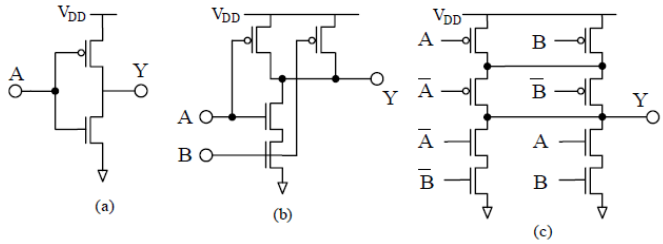
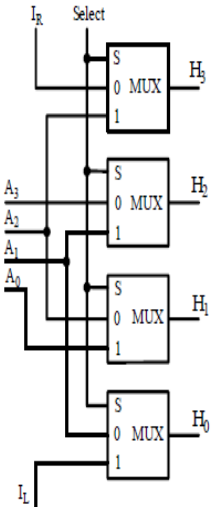


Figure 7: Transistor level static CMOS implementation of (a) NOT Gate (b) NAND Gate (c) XOR Gate

The second block of the logical unit is shift unit. This unit performs all shifting operations. Shift operations are used for serial transfer of data. They are also used in conjunction with arithmetic, logic, and other data-processing operations. A possible choice for the hardware implementation of a shift unit can be a bidirectional shift register with parallel load. Data can be transferred to the register in parallel and then shifted to the right or left. In this type of configuration, a clock pulse is needed for loading the data into the register, and another pulse is needed to initiate the shift. In a processor unit with many registers, it is more efficient to implement the shift operation with a combinational circuit. In this way the content of a register that has to be shifted is first placed onto a common bus whose output is connected to the combinational shifter, and the shifted number is then loaded back into the register. This requires only one clock pulse for loading the shifted value into the register [5].

A combinational shifter can be constructed with multiplexers as shown in Figure 8 The given diagram works on 4-bit data inputs and is extended to IoT processor in similar fashion. This shifter unit has four data inputs, A0 through A3, and four data outputs $G_0$to G_3. There are two serial inputs, one for shift left (IL) and the other for shift right (IR). When the selection input S is 0, the input data is shifted right (down in the diagram). When S = 1, the input data is shifted left (up in the diagram). The function table beside the diagram shows which input goes to each output after the shift. In a similar manner, a shifter with IoT processor data inputs needs 64 multiplexers.



Function Table

Select	Output			
S	H_3	H_2	H_1	H_0
0	I_R	A_3	A_2	A_1
1	A_2	A_1	A_0	I_L

Figure 8A 4-bit combinational shifter with function table.

3. Results and Discussions

Simulation results show that critical path delay of the designed ALU is 795 pico-seconds. This is the maximum time between the application of input signals and output of final multiplexer. Critical path lies in the computation of F[6] signal when arithmetic operations are performed. Worst case response is obtained from the same set of inputs which produce worst case delay in adders. Out of 795 ps of critical delay, 694 ps is introduced by the adder and the rest 101 ps comes from multiplexers which are implemented using pass transistor logic. The ALU operates on dual power supplies of V_{DDH} = 1.2V and V_{DDL}= 0.6V with a power dissipation of 1.61 mW. This power dissipation includes power drawn from V_{DDH}, $V_{DDL}$and clock.All the delay measurements given in Table 2 are in pico-seconds (ps). It can be seen that delay of logical gates increase considerable on decreasing supply voltage from 1.2V to 0.6V. The analytical relation between propagation delay and supply voltage. Table 2 also shows that among all gates XOR gate has maximum delay.

Table 2: Delay measurement of logical gates with different VDD

	τ_{PHL}	τ_{PLH}	τ_P
AND	766.6	479.7	623.15
OR	396.2	764.7	580.45
XOR	385	952.7	668.85
NOT	172.8	159	165.9

(e) Supply = 0.4V

From this graph a value of 0.6 volt is chosen as VDDL to drive logical unit, as it seems to be a good compromise between delay and power. Application of this dual supply to IoT processor ALU where this VDDL is used to drive only logic unit, reduces its power consumption by 75% and results in total power saving of 18%. Table 3 gives the exact values of power dissipation among different blocks of ALU, with and without dual-supply. It can be seen that besides adder circuit, multiplexers and other circuits used in the design dissipate a considerable amount of power. Figure 9 shows a comparison between total power consumption of the ALU with single and dual supply.

Table 3: Distribution of Power Consumption among various blocks

	Power Dissipation in µW For Single Supply (1.2 V)	Power Dissipation in µW For Dual Supply (1.2-0.6 V)
Adder	835	835
Logic gates	**485.8**	**122.3**
Clock	13.2	13.2
Others (Muxes)	643	643
Total	**1977**	**1613.5**

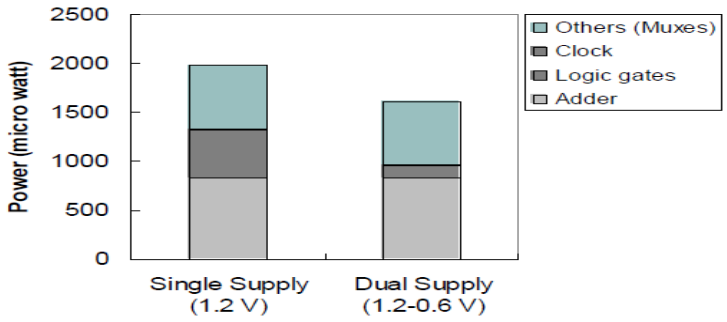


Figure 9: Power reduction using Dual Supply

Figure 9 shows dependence of frequency and power consumption on higher supply voltage VDDH. Inputs which generate worst case delay in the circuit may not be the one which trigger maximum power consumption. Therefore, separate inputs for speed and power measurement are chosen. To measure frequency of operation those inputs are applied which have the largest propagation path from bit-0 to 63. For measurement of power, inputs that produce large activity ratio are selected.Finally, the output in the Waveform (figure 10,11 and 12) shows the progressive transition of signals when performing arithmetic operation. Largest delay is found in signal F[63]. Note that signal X[63] which is logical output for 64th bit is computed much ahead of F[63] which denotes critical delay of the design. Waveform 8 shows output for all possible combinations of operations performed by ALU. The total design took around 9.5k transistors and dissipates 1.61 mW of power while operating at 1.2V-0.6V dual supply. The maximum frequency of the ALU is found to be 1.25 GHz.

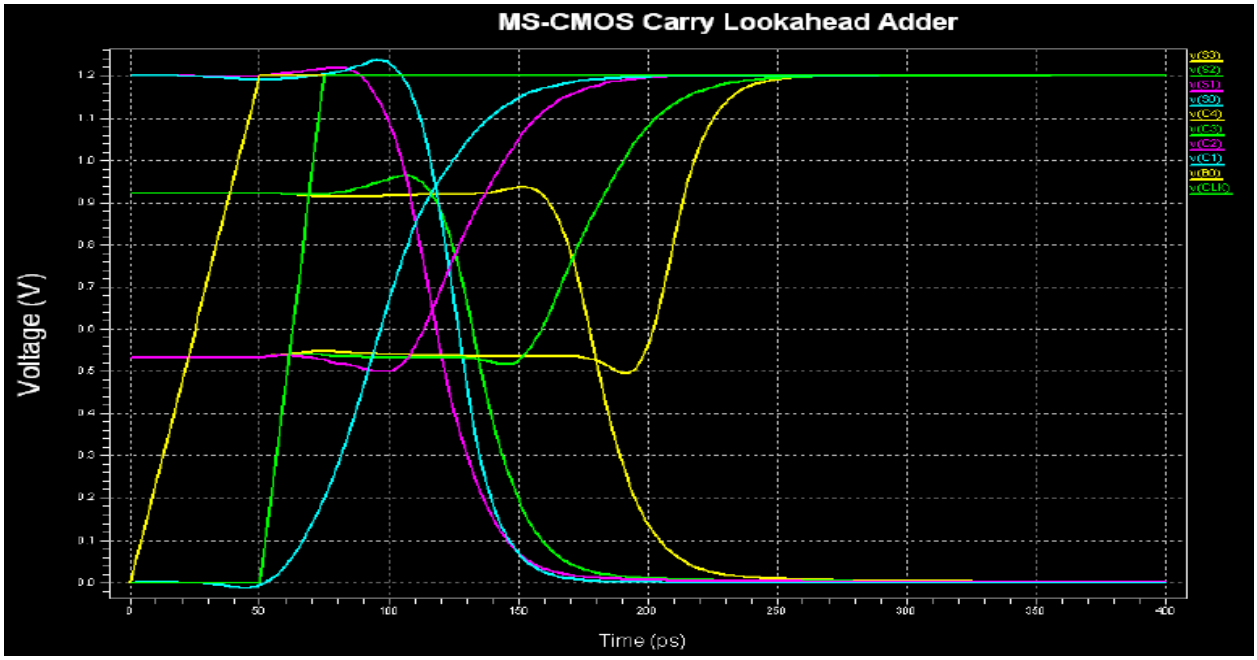


Figure 10:Sum and carry signal transitions for 4-bit MS-CMOS Carry Lookahead Adder

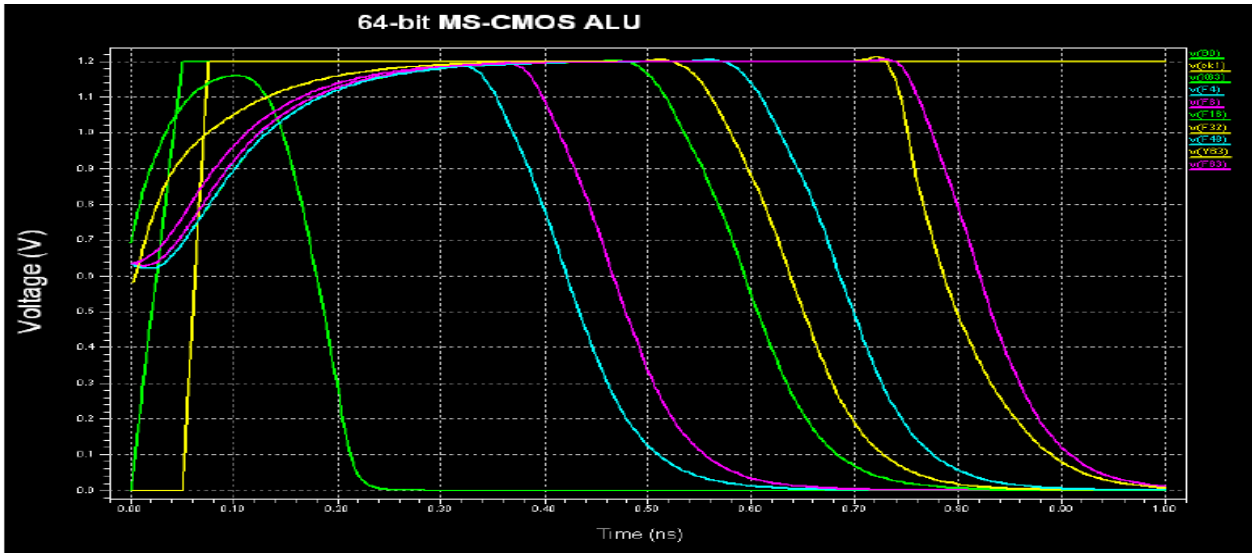


Figure 11:IoT processor MS-CMOS ALU output showing critical path delay

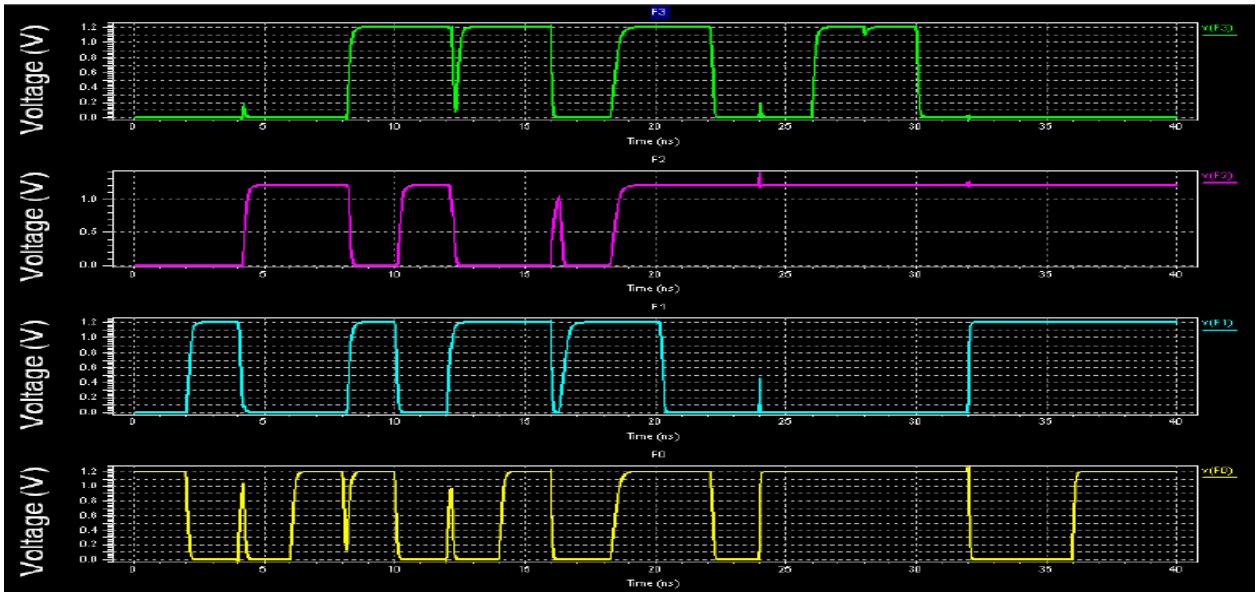


Figure 12:MS-CMOS ALU output for all operations

From figure 10,11 and 12 it my observe that proposed ALU for IoT processor is The simulation waveform shows that the proposed design of 64-bit ALU is working correctly.

Table 4: Comparative Results

Author	**Outcomes**
[1]	Static power obtain is 9 mW and Dynamic power obtain is 60 mW
[2]	Static Power obtain for the multiplier only is 3.175 mW
[3]	They observe total static power consumption of for 64 ALU 12mW
Proposed	Proposed work is the design based on modifies 64 CSA using MS-CMOS logic and 25nm transistor and Double Pass Transistor (DPL) based FA, The static power observe for the 64 bit Multiplier is 2.982 mW which is better as compare with Mansi Jhamb et al [2], the static power foe the fu ALU design is 7.58 mW which is better as compare with Rajesh Pidugu et a [3] and Aishwarya Verma et al [1], and the dynamic power observe is 48.65 mW which is also better then Aishwarya Verma e al [1], The results are observe at Tanner tool.

4. Conclusion

In this paper, potential of MS-CMOS logic for the use of ALU design has been highlighted. The designed ALU can operate at 1.25 GHz at 1.2-0.6 volt dual-supply and can be pushed to 1.9 GHz at 2.4-1.0 volt dual-supply with roughly five times increase in power consumption. Its performance can be further improved by increasing skew ratio which is optimally set at value 2. The total design took around 9.5k transistors to implement. Overall power dissipation of the design was found to be 1614μW. Out of this around 52% is consumed in adder circuit, 8% in logical gates and rest 40% in other circuits especially multiplexers. Power dissipation in clock signal is negligible as compared to other units. Implementation of dual supply has saved a power of 75% in logical unit which accounts for more than 18% saving in total power consumption of the ALU.

REFERENCES

[1] G G. Verma, "Design and Analysis of ALU for Low Power IOT Centric Processor Architectures," 2020 Global Conference on Wireless and Optical Technologies (GCWOT), 2020, pp. 1-5, doi: 10.1109/GCWOT49901.2020.9391609.

[2] M. Maheepala, M. A. Joordens and A. Z. Kouzani, "Low Power Processors and Image Sensors for Vision-Based IoT Devices: A Review," in IEEE Sensors Journal, vol. 21, no. 2, pp. 1172-1186, 15 Jan.15, 2021, doi: 10.1109/JSEN.2020.3015932.

[3] Ibrahim, "Systolic Processor Core for Finite-Field Multiplication and Squaring in Cryptographic Processors of IoT Edge Devices," in IEEE Internet of Things Journal, vol. 9, no. 2, pp. 1354-1360, 15 Jan.15, 2022, doi: 10.1109/JIOT.2021.3087274.

[4] Y. Zhang, Z. Guo, J. Li, F. Cai and J. Zhou, "AnnikaCore: RISC-V Architecture Processor Design and Implementation for IoT," 2021 IEEE 15th International Conference on Anti-counterfeiting, Security, and Identification (ASID), 2021, pp. 200-203, doi: 10.1109/ASID52932.2021.9651690.

[5] Aishwarya Verma, Karan Jain, Anu Mehra1, Nidhi Gaur, Design and Implementation of 64 bit VLIW Microprocessor on 20nm and 28nm Technologies, International Conference onInformation Science (ICIS),2016, DOI: 10.1109/INFOSCI.2016.7845329

[6] Mansi Jhamb, Garima, Himanshu Lohani, Design, implementation and performance comparison of multiplier topologies in power-delay space, GGSIPU, Sector-, Engineering Science and Technology, an International Journal 19 (2016) 355–363, Production and hosting by Elsevier

[7] Pragati Nagdeote1, Prof. Manisha Waje, Design of IoT processor Arithmetic Logic Unit (ALU) Based on BSIM4 Model Using Tanner, IJSART - Volume 2 Issue 10 –OCTOBER 2016 ISSN [ONLINE]: 2395-1052

[8] Priyanka Yadav, Gaurav Kumar, Sumita Gupta, Design and Implementation of 4-Bit Arithmetic and Logic Unit Chip with the Constraint of Power Consumption, IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834, p- ISSN: 2278-8735.Volume 9, Issue 3, Ver. V (May - Jun. 2014), PP 36-43

[9] Rajesh Pidugu, P. Mahesh Kannan, DESIGN OF 64-BIT LOW POWER ALU FOR DSP APPLICATIONS, ISSN(Online): 2278 – 8875, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 4, April 2013

[10] Neil H. E. Weste, David Harris, and Ayan Banarjee, "CMOS VLSI Design – A Circuits and Systems Perspective," 3rd edition, Pearson Education, 2007.

[11] Kaijian Shi, David Howard, "Challenges in Sleep Transistor Design and Implementation in Low-Power Designs," DAC 2206, July 24-28, 2006.

[12] Ali Bastani, Charles A. Zukowski., "A Low-Leakage High-Speed Monotonic Static CMOS 64b Adder in a Dual Gate Oxide 65-nm CMOS Technology," Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED'2006).