

Dynamic Logic ALU Design with Reduced Switching Power

Ramdas Bhanudas Khaladkar*, A. Anita Angeline and V. S. Kanchana Bhaaskaran

School of Electronics Engineering Department, VIT University, Chennai Campus, Chennai-600127, Tamil Nadu, India; khaladkar.ramdas2013@vit.ac.in, anitangeline.a@vit.ac.in, kanchana.vs@vit.ac.in

Abstract

Background: The dynamic gates produce an extremely high power overhead due to their high switching activity if used in designing a compact and high speed circuit. This paper presents a dynamic logic based arithmetic and logical unit that can realize reduced switching power to alleviate such a problem. **Methods:** This is achieved by employing the modified dual VT domino logic and Limited Switch Dynamic Logic (LSDL) mechanisms. The LSDL technique provides a static like switching action which exhibits lesser area requirement and high speed performance. **Findings:** The design of the arithmetic and logical units using the conventional dynamic logic, modified dual VT domino logic and the limited switch dynamic logic are presented in the paper for validating the claim. **Conclusion:** The simulation results depict the fact that the use of the LSDL shows 52.86% reduction in the dynamic power in comparison with the conventional dynamic logic. The circuits are implemented using Cadence® Virtuoso tools and 45nm technology node library files. The power analyses are performed using Cadence® Virtuoso Spectre tool.

Keywords: ALU Design, Dynamic Circuits, Limited Switch Dynamic Logic (LSDL), Low Power

1. Introduction

The Arithmetic Logic Unit (ALU) is the basic building block of any processor for performing the arithmetic and logical operations. The data to be operated on (operands) and the code indicating the operations to be performed (mnemonics) are given as the inputs to the ALU. The ALU produces the result based on the operation performed. In general, the ALU consists of different computing components, such as the arithmetic adder and subtraction units, the multiplication unit, the division unit, and the logical unit to perform the various arithmetic and logical operations. The instruction unit performs the operation of fetching the instructions and the control unit coordinates the operations to be executed for the given instruction.

The power consumption of a CMOS circuit contains two primary components. The first component is the static power, which incurs mainly due to the leakage power components. The second component is the dynamic or the switching power which results in, due to the charging and discharging of the load capacitance at the nodes, and

the short circuit power component of the output nodes, when both the pull-down and pull-up devices are *on*.

The total switching power¹ is the sum of the power dissipation due to the two components mentioned above, and it is given by

$$P_{\text{SWITCH}} = P_{\text{CAP}} + P_{\text{SHORT}} \quad (1)$$

Here, P_{CAP} is the power dissipation due to the charging and discharging activities of the load capacitor. The P_{SHORT} component is the short circuit power dissipation component. Hence, the switching power for a single switching operation of a static CMOS gate can be expressed as

$$P_{\text{SWITCH}} = (0.5 \alpha C V_{\text{dd}}^2 f_{\text{CLK}}) + (\alpha I_{\text{SC}} V_{\text{dd}} f_{\text{CLK}}) \quad (2)$$

Where the switching probability of the output is given by α , C represents the effective switching capacitance of all the nodes, f_{CLK} is the clock frequency and I_{SC} is the short circuit current component. The power consumption of a static CMOS gate happens only when the output

* Author for correspondence

transitions or state switching occur, which identifies the static CMOS logic with capability to offer low power.

The switching power of a dynamic gate is expressed as

$$P_{\text{SWITCH}} = (P_{\text{out}(1)} C V_{\text{dd}}^2 f_{\text{CLK}}) + (P_{\text{out}(1)} I_{\text{SC}} V_{\text{dd}} f_{\text{CLK}}) + P_{\text{CLK}} \quad (3)$$

Here, $P_{\text{out}(1)}$ is the output probability of the gate to remain in its *HIGH* state. It depends on the gate topology and the input combinations. I_{SC} is the short circuit current and P_{CLK} is the power dissipation due to clock.

By comparing Equations (2) and (3) above, it can be found that the power dissipation of a static gate is output switching dependent, while the power dissipation of a dynamic gate is output state-dependent. It is explained as follows. In a dynamic gate, the switching activity during one clock cycle happens during the pre-charge phase and also during the evaluation phase if the logic is *True*. This makes the switching power twice as that of the equivalent static gate for a particular input condition when *True* and equivalent to that of a static gate when the input is *False*.

The paper is arranged as follows. Section 2 elaborates the methodology of operation of the three logic gate styles, namely, the dynamic CMOS logic, the limited switch dynamic logic and the modified dual V_T domino logic. Section 3 discusses the arithmetic and logical unit, functional units and the integration. Section 4 presents the simulation and the comparisons of the individual modules and the ALU designed in the three different approaches, viz. simple dynamic logic, modified dual V_T logic, limited switch dynamic logic methodology. Section 5 concludes.

2. Methodologies

2.1 Dynamic CMOS Logic

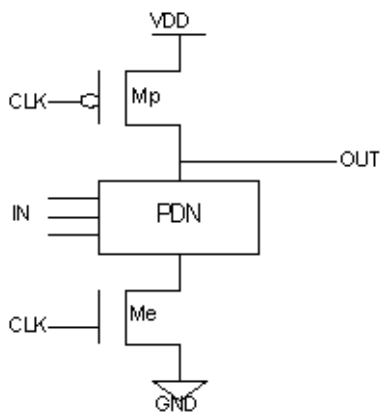


Figure 1. Basic dynamic CMOS circuit.

The basic structure of a dynamic CMOS logic gate² is as shown in Figure 1. The operation of the circuit comprises of two phases, namely, the *pre-charge* and *evaluation* phases.

2.1.1 Pre-charge Phase

During this phase, the outputnode (*OUT*) is pre-charged to V_{DD} through the pre-charge transistor *Mp* and $\text{CLK}=0$ during this time. At the same time, the evaluation transistor *Me* is *OFF*.

2.1.2 Evaluation Phase

With $\text{CLK}=1$, the pre-charge transistor *Mp* is in *OFF* state and the evaluation transistor *Me* is in *ON* state. The output depends on the input values and the pull-down path. Assuming the input is *True* or *HIGH*, the pull down network is *ON* and it makes the output pulled down *LOW* thus discharging the output node to ground through the footer device. The PDN remains in *OFF* condition, when the inputs are *LOW* or *False*. Hence, the pre-charged value on the output node (*OUT*) is retained.

The primary feature of the dynamic circuits is that the evaluation can happen only once per every clock cycle. To elaborate, during the period of the evaluation, the single path available at the dynamic output node to the GND rail is through the PDN applied with $\text{CLK} = \text{HIGH}$. And, it means that the input can be changed only once during the evaluation phase. Even if the input changes when CLK remains *HIGH* in the same clock cycle, it cannot reflect in any output change. When the output is discharged, the next pre-charge operation is necessary for the next evaluation to take place. Thus, it can be said that the dynamic evaluation necessitates an intervening *HIGH* logic by the pre-charge operation, even when any two successive outputs happen to be *LOW*. This feature causes switching at the output which leads to unnecessary dynamic power consumption. Hence, the need for reducing the switching power of the dynamic logic is to be incorporated, and this paper focuses on the two styles which attempts retaining the previous state when the inputs remain the same.

2.2 Modified Dual V_T Domino Logic

The dual V_T domino logic shown in Figure 2 is an enhancement of the conventional dynamic CMOS logic style. It offers reduction in the sub threshold leakage as presented in reference³. The structure of the modified dual V_T domino logic comprises of an inverter formed by

P3 and N2 at the output along with an NMOS transistor N3 in series with N2. The clock is fed as an input to the transistor N3. The evaluation network consists of low threshold voltage devices. This augments for the speedier evaluation process. Since the remaining devices possess normal threshold voltage levels, the possible leakage power dissipation incurred by the evaluation network would be controllable. The operation of the dynamic circuit is explained as follows.

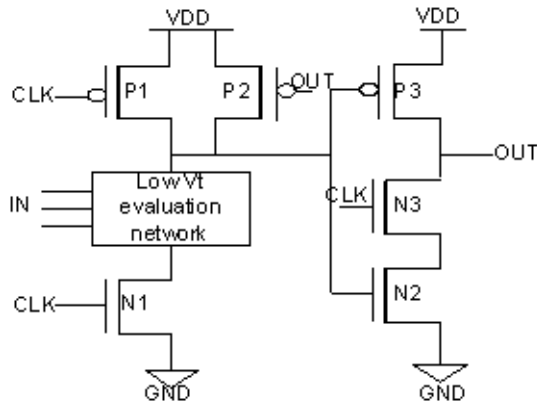


Figure 2. Modified dual V_T domino logic.

2.2.1 Pre-charge Phase

When the $CLK=0$, the transistor $P1$ is ON and the transistors $N1$ and $N2$ are OFF. With the dynamic node = HIGH, the transistor $P3$ is OFF. Hence, the output retains the previous state and the device $P2$ operates as per the value existing at the node OUT .

2.2.2 Evaluation Phase

When the $CLK=1$, the transistor $P1$ is in OFF state and the transistor $N1$ becomes ON. This commences the evaluation phase of the dynamic circuit. If the pull down network is ON or True, then the output of the dynamic node is pulled down LOW. It makes the transistor $P3$ to be ON and the device $N2$ is OFF. This yields the output of the second inverter HIGH thus following the input as defined by the logic network. Note that at this time with a HIGH in OUT node, the device $P2$ is OFF.

2.3 Limited Switch Dynamic Logic (LSDL)

This technique consists of providing the dynamic logic evaluation^{4-6, 8-10} with the use of a static latch. As shown in Figure 3, the static latch is constituted by the transistors $M1, M2, M3, M4$ and $M5$, with the clock CLS controlling

the device $M3$ at its gate. The transistors $M4, M5$ provides a non-inverted output. The transistors $M1$ and $M2$ prevent the back propagation to the dynamic node. The clocked footer transistor $M3$ prevents the forward propagation during the pre-charge phase. The inverter $M6$ and $M7$ configures the inverter included for phase correction.

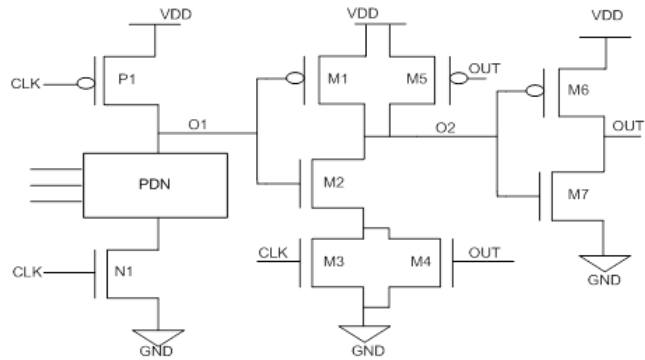


Figure 3. Limited switch dynamic logic.

2.3.1 Pre-charge Phase

When $CLK=0$, the PMOS device $P1$ is ON and the NMOS devices $N1$ and $M3$ are OFF. The transistor $M4$, depending on the previous output either pulls $O2$ down or keeps it HIGH based on the OUT available from the previous evaluation phase.

2.3.2 Evaluation Phase

When $CLK=1$, the transistor $P1$ becomes OFF state and the footer transistor $N1$ is ON. If the inputs are HIGH or True the PDN block evaluates, thus bringing the output of the dynamic node $O1$ LOW. This makes the transistor $M1$ to be ON and the transistor $M2$ to be OFF and there is no discharge path from the node $O2$ to GND. Thus, the node $O2$ attains logic HIGH which in turn will be inverted by the inverter formed by $M6$ and $M7$ transistors to produce the output node as LOW with respect to the input PDN.

If the inputs are LOW or the PDN is not True, then the dynamic output node $O1$ stays at HIGH. Then, the transistor $M1$ is OFF and the transistor $M2$ is ON. $M3$ transistor is ON which is enabled by the clock CLK , thus providing a ground path to $O2$. This makes the node $O2$ to be LOW, which in turn is inverted by the inverter of $M6$ and $M7$, making the output node switch to HIGH level.

To elaborate, assume the input condition that will remain the same through two successive precharge-evaluate cycles. After the first evaluate phase with HIGH

or *True* input condition, the dynamic node would have become LOW. This would have made O2 HIGH and OUT node LOW. Now, assume, the next precharge takes place making O1 HIGH. Consider the latch with its O2 node HIGH. It will remain so, since the CLK is LOW and OUT = LOW, thus making M3 and M4 OFF. Hence, even if the node O1 becomes HIGH during the subsequent pre-charging phase, it does not reflect on the node O2. In other words, the previous value is retained at the output node O2 and hence OUT irrespective of the precharge operation and an unwanted discharging of the node O2 and OUT are avoided.

3. Arithmetic and Logical Unit (ALU)

The ALU is the prime component of any processor. The power consumption of the ALU majorly contributes for the total power consumption of the processor. Hence, the appropriate logical style of the design of the combinational circuits used in the ALU can significantly lead to the overall power reduction of the processor. In this direction, the individual modules of the ALU are designed using the low power dynamic switching reduction techniques.

3.1 Functional Blocks of the ALU

This section elaborates the design and testing of ALU using the three dynamic logic styles. The arithmetic and logical bitwise operations are performed by the ALU. The primary modules of the ALU are the following

3.1.1 Adder-subtractor Unit

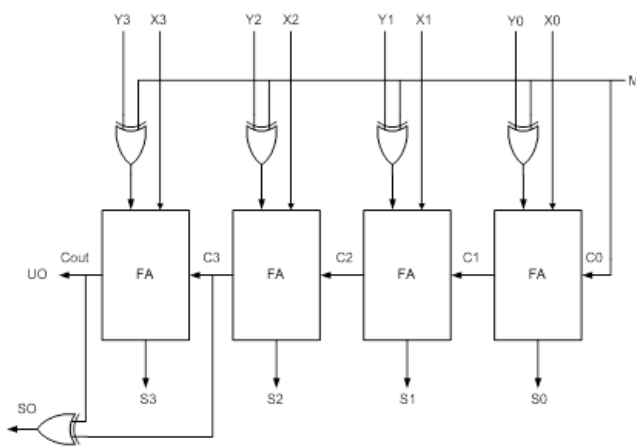


Figure 4. Adder- Subtractor unit.

The adder-subtractor unit shown in Figure 4 is capable of performing the addition and the subtraction operations. It is so designed that it can handle either of the two operations based on the mode signal M. This avoids the use of two individual units. The addition is performed for $X+Y$ using the full adder modules when the mode control signal $M = 0$. However, for the subtraction process, the operation $\{X+(-Y)\}$ will be performed which is in effect, the subtraction operation. Hence, the modification in the adder circuit which will invert the Y bits before the addition process is incorporated in the adder-subtractor module. Note that a '1' is added to the first full adder module as a carry bit C0 using the mode control signal M. When $M=1$ meaning the subtraction operation, then the Y bits are complemented by the use of the XOR gates and the addition of bit 1 from M makes it a two's complemented addition of Y with X.

In other words,

$$S = X + (-Y)$$

On the other hand, when the mode control signal $M=0$, the addition operation will be carried out.

And, then

$$S = X + Y$$

3.1.2 Wallace Tree Multiplier

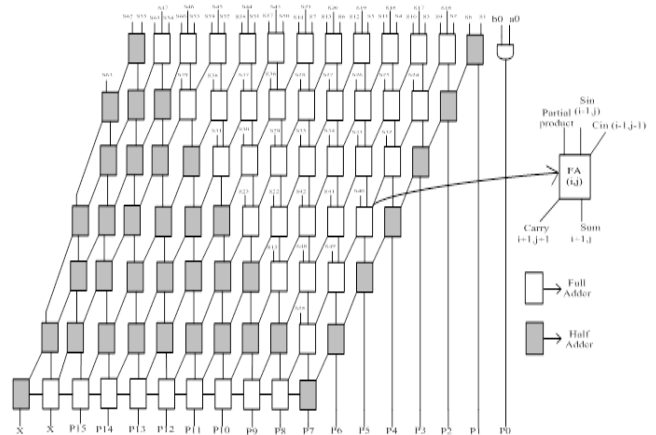


Figure 5. Wallace Tree Multiplier.

Figure 5 shows the Wallace Tree Multiplier structure⁷ which carries out the multiplication operation of the ALU. The shaded block represents the half adders and the unshaded rectangle represents the full adder. The Wallace tree architecture is constituted by three steps as follows

- The partial products are calculated.
- The partial products are arranged in a number of two row matrices, and it is followed by forming the *sum* of

rows of each set of row matrix bits, thus forming the partial product rows for subsequent addition process of the row matrices.

- The resulting rows of the previous step are added again using adders to obtain the intermediate sum of product bits and this continues in succession to obtain the final output.

In this work, an approach to add a larger number of rows to save timing and enhance speed performance is attempted. Fulladders are utilised to add three bits of a column at a time for the vertically aligned bits as explained above for the 2-bit additions of rows. Hence, it is referred to as the three input *Wallace Tree* structure. The output *sum* of the first stage adder is taken as an input to the next adder in succession. In the same way, the *carry* bit of full adder is given as an input to the next full adder while executing the step two. This process is repeated for all the stages and across all steps incurred in the 8-bit multiplier architecture.

3.1.3 Logical Operations Unit

The ALU supports the logical operations such as the AND, OR, NOT, NAND, NOR and XOR for the 8 bit data. The input data are received from registers or the source (locations) based on the instruction, and the results after the logical operations are stored in the accumulator.

3.1.4 Instruction Decoder

The instruction unit enables the computation and register selection based on the op-code and the operand. The instruction decoder is employed to decode the instruction and accordingly enable the corresponding operation, by strobing the source and destination registers. In other words, the decoders are deployed for the operation selection and the register selection. The input to the decoder asserts one among the n output lines, depending on the m-bit input instruction code. In general, the decoder with m-bit input lines has $n=2^m$ output lines. For example, in a 3X8 decoder, if the decoder input is 100, then the 4th output line will assert itself, while the rest of the output lines are de-asserted.

3.1.5 Registers

The registers are meant for holding the input data to be given to the arithmetic and logic blocks and for holding

the result. The registers are identified to a certain maximum number based on the number of operations identified and the decoder bits.

3.1.6 Control unit

The control unit regulates the sequence of the operations of ALU. This is designed by considering the possible states of data flow. It operates in coordination with the instruction unit and controls the arithmetic and logical units, as well as the registers for storing and retrieval of data.

3.2 Integration of ALU

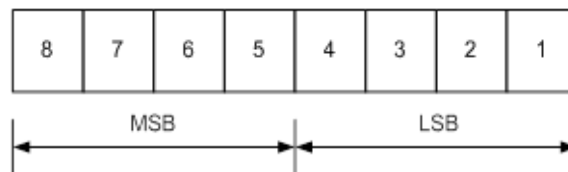


Figure 6. 8-bit Instruction Register Format.

A typical flow with the ALU is depicted in this section. Assume an 8 bit instruction shown in Figure 6 is fed to the instruction unit. Figure 7 demonstrate the flow in the integration module of the ALU. The 4-bit MSB of Figure 6 consists of the op-code used for the operation selection using the decoder *a* as indicated in Figure 7. The arithmetic/logical block will perform the computation based on the selection of the required operation by the decoder *a*. The LSB is used for the register selection using decoder *b* and the input data are fed to the computation unit.

To discuss the operation of circuit, considering the binary instruction defined the bit pattern 0 0 0 1 0 0 0 0 fed to the instruction decoder. The MSB consists of the op-code 0 0 0 1, which makes the output of the second line of decoder to be *HIGH*. This in-turn will enable the adder-subtractor block *HIGH* as shown in Figure 7. The LSB consists of the op-code 0 0 0 0. This makes the output of the first line of the decoder *b* to be *HIGH*. This results in the selection of the register A and register B. If the data at register A and register B are 1 0 1 1 1 0 1 0 and 1 1 0 0 1 0 0 1 respectively, an output of 1 0 0 0 0 1 1 is obtained as addition operation is asserted by the opcode.

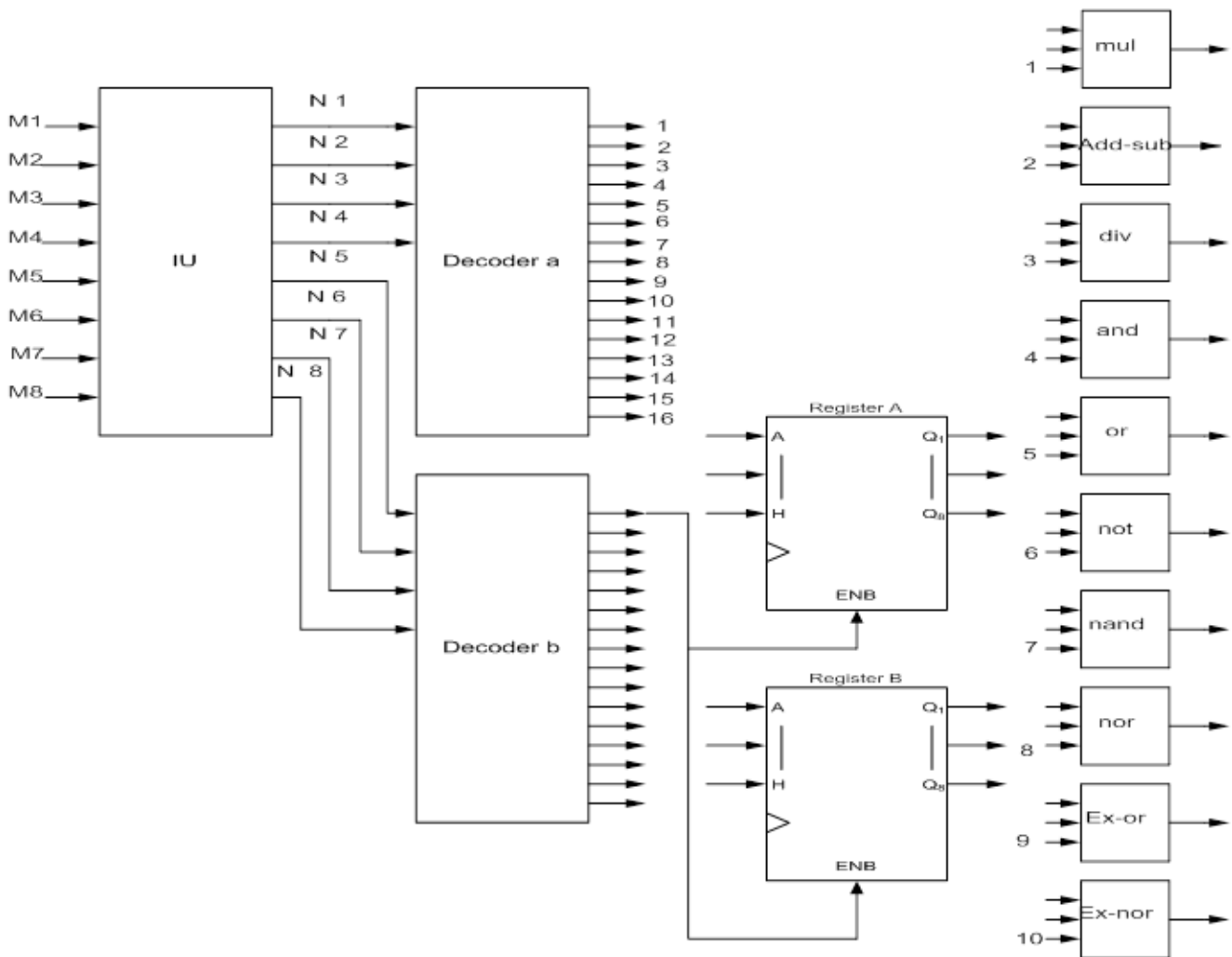


Figure 7. Integration Module of ALU.

4. Simulation and analysis

The various sub blocks of the ALU design are simulated using the design styles, namely, the simple dynamic, the modified dual V_T domino logic and the limited switch dynamic logic (LSDL).

Table 1. Power Comparison of the ALU_Arithmetic Units.

Module	Dynamic Power (uW)		
	Simple Dynamic	Modified Dual V_T	Limited switch dynamic
Adder-Subtraction	88.8	17.87	14.88
Wallace Tree Multiplier	185	174.4	57.36

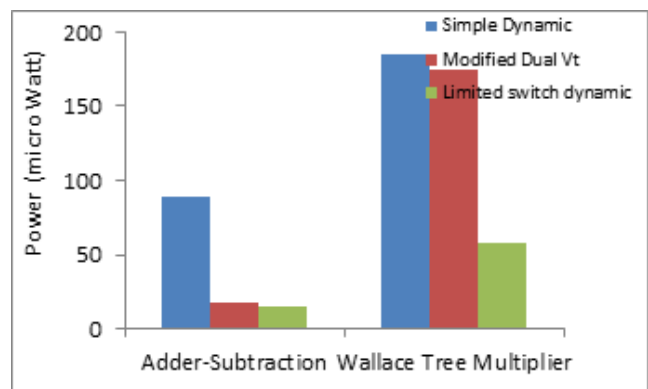


Figure 8. Power Comparison of Arithmetic Units.

Table 1 shows the power dissipation comparison of the three types of dynamic logic methodologies for

the individual arithmetic blocks, namely, the adder-subtraction unit and the Wallace tree multiplier unit. It depicts the simple dynamic and modified dual V_t types consuming 88.8uW and 17.87uW of power as against the 14.88uW by the limited switch dynamic logic, for the adder-subtraction operation. It can also be seen that the LSDL demonstrates only 31% and 32% of the power dissipation incurred by the simple dynamic and modified dual V_t dynamic logic styles respectively as shown in Figure 8.

Table 2. Power Comparison of the ALU_Logical Units

Module	Dynamic Power (nW)		
	Simple Dynamic	Modified Dual V_t	Limited switch dynamic
LOGICAL NOT	7.61	10.4	4.85
LOGICAL NAND	8.31	21.23	3.74
LOGICAL NOR	7.98	21.31	3.24
LOGICAL AND	13.03	7.71	3.19
LOGICAL OR	13	8.99	2.86
LOGICAL XOR	8.14	3.13	3.43

Table 2 presents the power dissipation values of the logical units of the ALU. It is observed that for an 8 bit logical NOT, NAND and NOR operations using the simple dynamic style shows less power consumption of 7.61nW, 8.31nW and 7.9nW as compared to that of the modified dual V_t domino style, which consumes 10.4nW, 21.23nW, 21.31nW respectively. However, it can be noted that the dynamic switching activity is reduced. The increase in the circuit area leads to increased power consumption in the modified dual V_t circuits. On the other hand, the LSDL style of implementation shows reduced power consumption of 4.85nW, 3.74nW and 3.24nW respectively for the same logic operations. This demonstrates that the LSDL style of implementation incurs reduction of 46%, 17% and 15% of the power consumption respectively as compared to the modified dual V_t methodology. In the case of AND, and OR operations, the power consumed by the LSDL logic is less such as 2.8nW and 3.43nW respectively. It exhibits 41% and 31% power consumption respectively, over the power consumption by the static logic and 24% and 21% power reduction over the modified dual V_t logic style.

Table 3 depicts the power consumption of the ALU for the addition operation using the three styles such

as: the simple dynamic logic, modified dual V_t domino logic, and limited switch dynamic logic methodologies. It demonstrates that simple dynamic style consumes 12.03 μ W power and the modified dual V_t domino logic and LSDL shows reduced power consumption of 9.8 μ W and 5.67 μ W respectively.

Table 3. Power Comparison of the ALU Unit

Component	Dynamic Power (uW)		
	Simple Dynamic	Modified Dual V_t	Limited Switch Dynamic
ALU	12.03	9.8	5.67

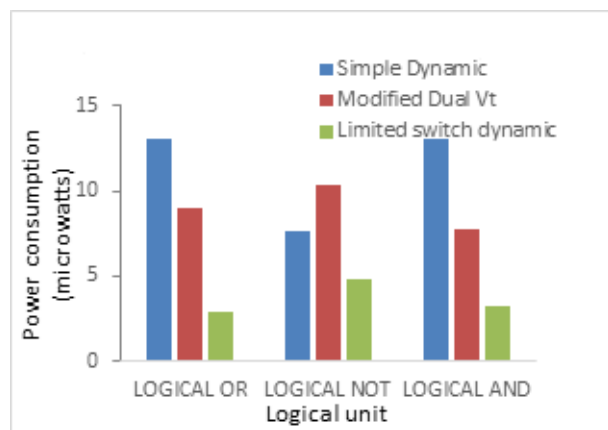


Figure 9. Power Comparison of Logical Units.

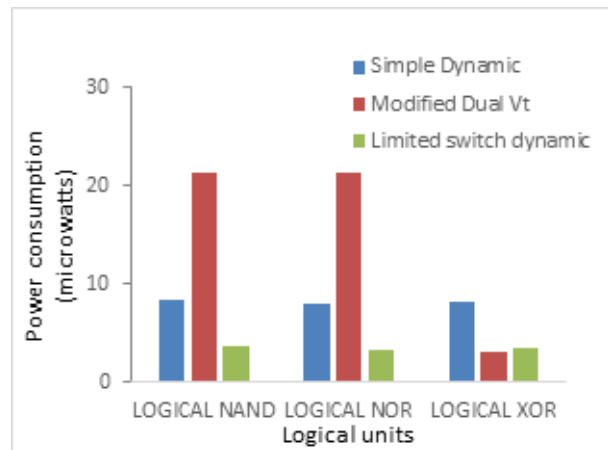


Figure 10. Power Comparison of Logical Units.

Figure 8, Figure 9, and Figure 10 Shows the power comparison of ALU arithmetic and logical blocks in all the three different approaches.

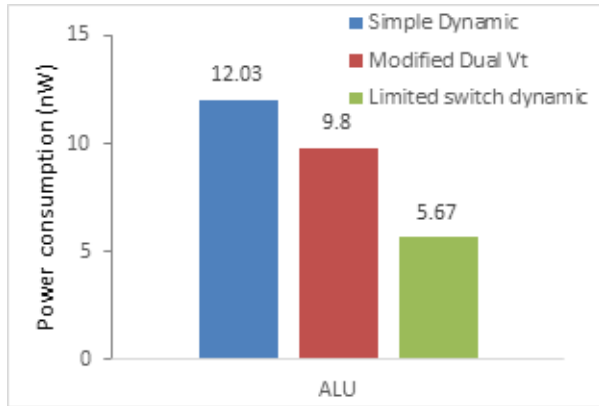


Figure 11. Power Comparison of ALU Unit.

Figure 11 shows the power comparison of the ALU module designed and implemented in all the 3 styles. It reveals the fact that the LSDL ALU consumes 5.67nW of power while the simple dynamic and modified dual V_T approaches consume 12.03nW and 9.8nW respectively. The reduction in the power consumption has been realized due to the decrease in the switching activity at the output node. As stated in Eq.2, in the modified dual V_t dynamic logic style and LSDL, the switching power dissipation is reduced since the previous state is made to be retained during the precharge operation.

5. Conclusion

Arithmetic and logic units are designed and implemented for the three dynamic logic styles, namely, the conventional dynamic logic, modified dual V_T domino logic and the LSDL logic using 45nm technology library. The switching power overhead incurred in the conventional dynamic logic during the pre-charge/evaluation operations is reduced in the modified dual V_T and the limited switch dynamic logic through the additional feature of retaining the previous output state at the dynamic node during the subsequent pre-charge phase also. The LSDL ALU realizes a power reduction 52.86% as compared to

the simple dynamic logic ALU. The modified dual V_T domino logic realizes 42.1% of reduction in the power as compared with the simple or the conventional dynamic logic. The modified dual V_T domino logic is proved to be quite suitable in the design of larger circuits due to this fact. In the case of a smaller circuit, the additional circuit overhead poses area penalty.

6. References

1. Charbel JA, Magdy AB. Single-phase SP – domino: A limited-switching dynamic circuit technique for low-power wide fan-in logic gates, *IEEE Trans on Circuits and Systems*. 2008; 55:141–5.
2. Rabaey JM, Chandrakasan A, Nikolic B. *Digital Integrated Circuits*. Second Edition. Prentice–Hall of India Private Limited; 2004.
3. Wang C-C, Huang C-C, Lee C-L, Cheng T-W. A low power high-speed 8-bit pipelining CLA Design Using Dual-Threshold Voltage Domino Logic. *IEEE Trans on Very Large Scale Integration (VLSI) Systems*. 2008 May; 16(5):594–8.
4. Belluomini W et al. Limited switch dynamic logic circuits for high speed low-power circuit design. *IBM J Res & Dev*. 2006 May; 50(2/3):277–86.
5. Jayakumaran S, Hung CN, Kevin JN, Robert K, Brown B. Controlled-load limited switch dynamic logic circuit. *IEEE Conference on Computer Society*; 2005. p. 1–6.
6. Belluomini WA, Montoye RK, Ngo HC. Limited switch dynamic logic circuit. U.S. patent 6,690,204 B1; 2004 Feb.
7. Vinoth C, Kanchana Bhaaskaran VS, Brindha B, Sakthikumar S, Kavinilavu V, Bhaskar B, Kanagasabapathy M, Sharath B. 8-10 A Novel Low Power and High Speed Wallace Tree Multiplier for RISC Processor. 3rd International Conference on Electronics Computer Technology (ICECT); 2011 Apr. p. 330–4.
8. Ji-ren Y, Karlsson I, Svensson C. A true single-phase-clock dynamic cmos circuit technique. *IEEE J Solid-State Circuits*; 1997 Oct. 22(5):899–901.
9. Fang T, Amine B, Gu Z. Low power dynamic logic circuit design using a pseudo dynamic buffer Integration. *The VLSI journal*. 2012 Sep; 45(4):395–404.
10. Pandey AK, Mishra RA, Nagaria RK. Low power dynamic buffer circuits. *International Journal of VLSI Design & Communication Systems (VLSICS)*. 2012 Oct; 3(5):53–65.