ALU Design using Pseudo Dynamic Buffer based Domino Logic

SIVA KUMAR AKURATI [1], A. ANITA ANGELINE[2], V S KANCHANA BHAASKARAN[3]
[1, 2, 3] School of Electronics Engineering, VIT University, Chennai Campus, Chennai-600127, Tamil Nadu, India
[1]sivakumar.akurati2014@vit.ac.in, [2]anitaangeline.a@vit.ac.in, [3]kanchana.vs@vit.ac.in

Abstract

Background: Domino logic is widely used in modern digital systems because of easy implementation with less number of transistors and high speed. The pre-charge and evaluation phases of the domino logic, leads to enormous transitions at the output. This switching of the output is undesirable as it leads to more dynamic power dissipation. **Methods:** This achieved by the using the structures such as True Single Phase Clocking (TSPC), Limited Switch Dynamic Logic (LSDL) and Pseudo Dynamic Buffer (PDB). The PDB structure reduces the dynamic power to a greater extent, without increase in the number of transistors. **Findings:** The design of Arithmetic and logical units using PDB is presented in the paper for validating the claim. **Conclusion:** This paper details the design of an ALU using PDB based domino methodology. The PDB based domino logic ALU demonstrates an average power 11.86 mw with a delay of 152.75 ps. Simulations are carried using Cadence® Virtuoso with 180nm technology library file.

Keywords: Arithmetic and Logic unit (ALU), domino logic, dynamic power, low power, Pseudo Dynamic Buffer (PDB).

1. Introduction

The design of a fast and compact processor underlies on the Arithmetic and Logic unit (ALU) design. The design of various ALU units using the domino logic enables in achieving high speed with compactness.

Power consumption of CMOS circuits consists of two primary components. The first component is the switching or dynamic power, which is due to the charging and discharging of the nodal capacitances and short circuit power component. The short circuit power component is due to the rail – to – rail current paths that exist between V_{dd} and *GND* rails during the input/output logical transitions. It occurs when both the pull-up and pull down networks are *ON*. The second component is the static power, which is dominated by the leakage power. Hence, the total power dissipation is given as in Eq.(1).

$$P_{TOTAL} = P_{DYN} + P_{STATIC} \quad (1)$$

where, P_{TOTAL} is the total power consumption, P_{DYN} is the dynamic power consumption and P_{STATIC} is the static power consumption and is given in eq. (2).

$$P_{DYN} = P_{NODE} + P_{SHORT} \quad (2)$$

Here, P_{NODE} is the power dissipation due to the charging and discharging of capacitance at the output node. The P_{SHORT} component is the short circuit power dissipation component. Hence, the dynamic power for a static CMOS circuit can be expressed in eq. (3).

$$P_{DYN} = (0.5\alpha\ C\ V_{dd}^2\ f_{CLK}) + (\alpha\ C\ V_{dd}^2\ f_{CLK}) \quad (3)$$

Here, α is the signal switching portability, C is the switching capacitance at the nodes, f_{CLK} is the clock frequency and I_{SHORT} is the short circuit current component. The power dissipation of static CMOS occurs at the output node when state or transitions occur. The dynamic power of dynamic gate is expressed in Eq. (4).

$$P_{DYN} = (Pout\ CV_{dd}^2 f_{CLK}) + (PoutI_{SHORT}V_{dd}f_{CLK}) + P_{CLK} \quad (4)$$

where, P_{out} is the probability of output in HIGH state. It depends on the input combinations. I_{SHORT} is the short circuit current and P_{CLK} is the clock power dissipation.

Comparing Eq. (3) and (4), it can be found that the power dissipation of dynamic gate is output state dependent and the power dissipation of static gate is output transition or switching dependent. In a dynamic gate, the switching activity happens in both pre-charge phase and also during evaluation phase if the logic is TRUE. This contributes more to the dynamic power dissipation.

2. Methodologies

2.1 Conventional Domino Logic:

The basic structure of domino logic is shown in Figure 1. It consists of N-type evaluation network connected between one P-type clocked transistor and one

N type clocked transistor and the output is followed by a static inverter[1]. The clocked transistors enable the operation in two phases, such as the pre-charge and the evaluation phases.

During the pre-charge phase, the clock signal is *LOW*, which turns the PMOS transistor *M1* to be *ON* and NMOS transistor *M2* to be in *OFF* state. This enables the dynamic node X to pre-charge to HIGH.

In the evaluation phase, the clock signal is HIGH and it causes PMOS transistor *M1* to be in *OFF* state and NMOS transistor *M2* to be in *ON* state. Assuming the inputs are in HIGH state, the PDN is *ON* and makes the dynamic node X to be pulled down to LOW and makes the output node to be HIGH state. Assuming the inputs are at LOW state, PDN is *OFF* and makes the dynamic node X pulled up to V_{dd} and the output node discharges to *LOW* via the static inverter. Thus, unnecessary switching occurs during the pre-charge phase, even when there is no change in the input logical levels during consecutive evaluation phases. Hence, there exists a need to reduce this dynamic or switching power. This is accomplished by modifying the output part of the domino logic circuit structure. The various logic styles such as Pseudo Dynamic Buffer (PDB), True Single Phase Clock (TSPC) and Limited Switch Dynamic Logic (LSDL) retains the previous state till the next evaluation phase. Almost, all these circuits lead to extra transistor circuitry causing increased area. On the other hand, in the case of PDB based domino logic, there exists no increased transistor count but a mere modified connectivity. Furthermore, in domino logic the keeper transistor is utilized to reduce the charge sharing and charge leakage problems.

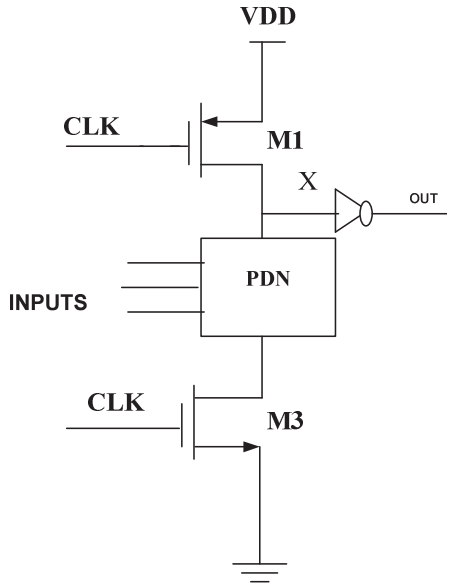

Figure 1. Domino logic circuit.

2.2 Pseudo dynamic buffer mechanism:

In domino logic, the switching or transient power is very high. The usage of Pseudo Dynamic Buffer (PDB) along with the domino logic retains the same output till the next evaluation phase. The PDB based domino logic as shown in Figure 2 comprises M5 transistor, where the source terminal of it is connected to node A (which is the drain of the footer M3 transistor) instead of connecting to the ground [2, 3]. This prevents the pre-charge pulse from propagating to the output node. This causes reduction in the switching power dissipation.

When the input signal is *LOW,* the dynamic node X is always *HIGH* and the output is *LOW* in both cases of pre-charge and evaluation phases.

When the input signal is high

- during the pre-charge phase i.e. when clock signal is *LOW*, node X is charged to *HIGH*, NMOS transistor M5 is ON and the output is LOW.
- during the evaluation phase i.e. when clock signal is high, node X is discharged to ground, PMOS transistor *M4* is ON and output is charged to *HIGH.*

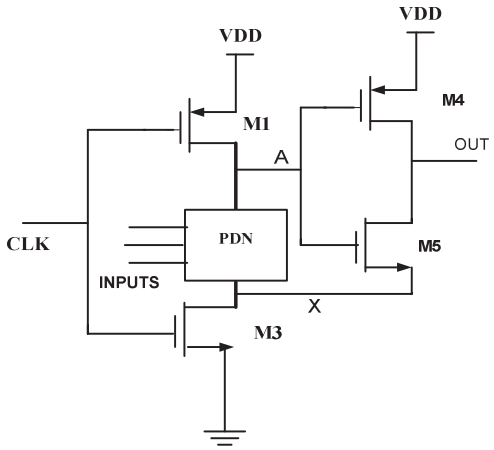

Figure 2. PDB logic circuit.

3. Arithmetic and Logical Unit

ALU being the primary block for any processor, its power consumption is the major component of the total power consumption of any typical processor based system next to the memory array [4, 5]. Hence, the ALU design is focused on PDB based domino logic design which has lower dynamic power dissipation, compared to other logical designs such as conventional domino, TSPC and LSDL. The individual modules of ALU are designed using PDB based domino logic.

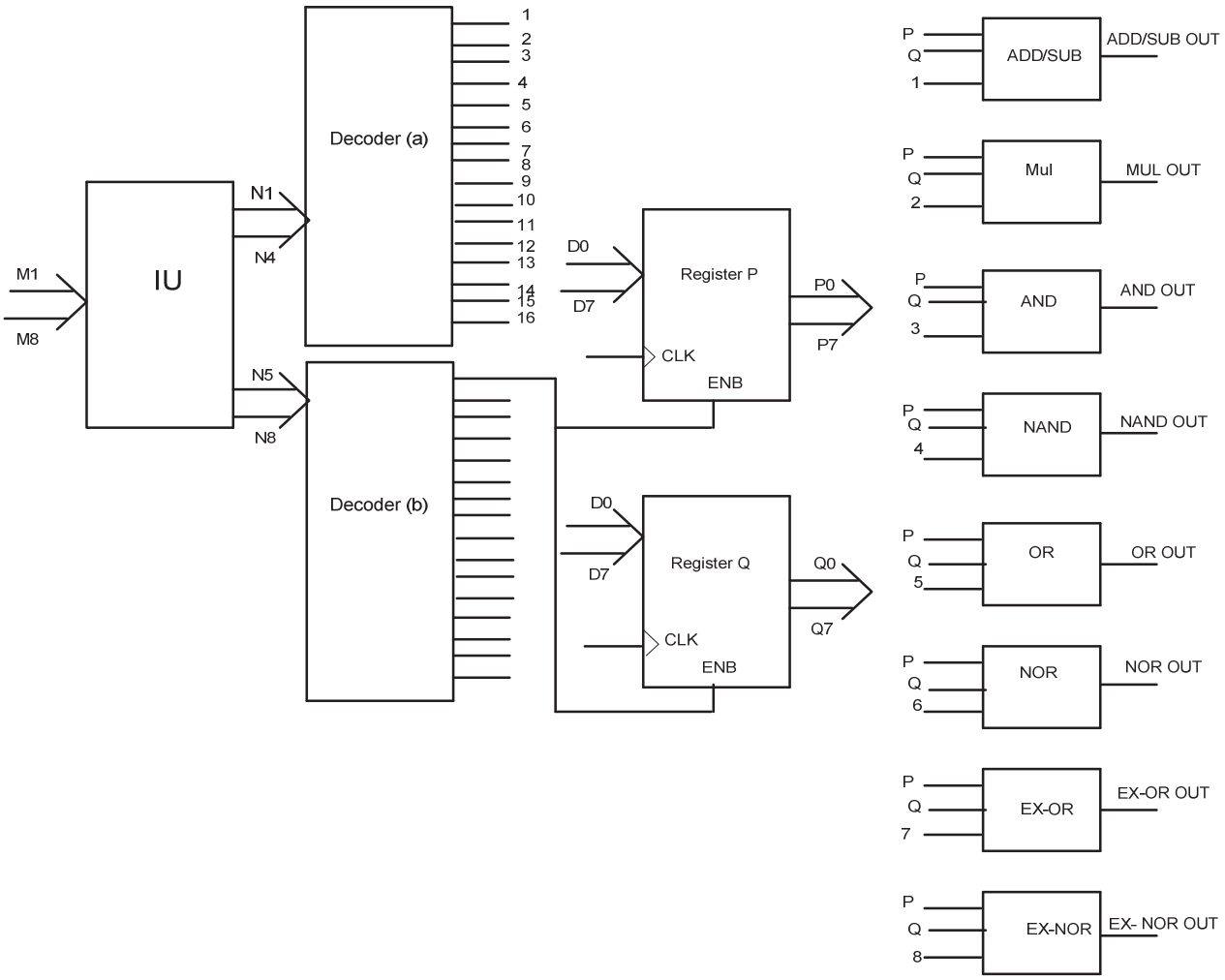

Figure 3. ALU module .

Figure 3 shows the block diagram of the ALU. It consists of instruction unit, set of decoders, registers and different arithmetic and logical blocks.

An 8 bit instruction is given to the instruction unit which is shown in Figure 4. The instruction is divided into two parts:

4 bit MSB: signifies the op-code which is used to select the type of operation that is to be performed by the ALU.

4 bit LSB: for selection of registers containing the d

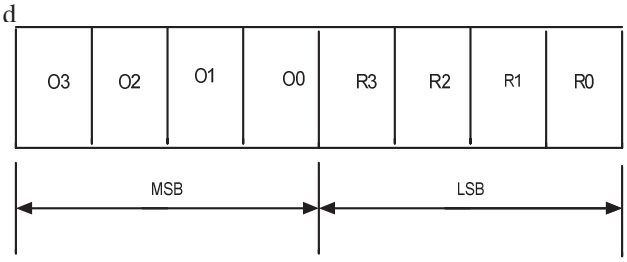

Figure 4. Format of 8 input Instruction Register.

Depending on the MSB, the decoder (a) decodes the appropriate arithmetic or logical operation to be performed. The 4 bit LSB is used for selecting the registers, which contains the data to be operated using the decoder (b). Also, the resultant register is indicated through the bits of the instruction.

Assuming, that the bit pattern 00000000 *is* fed to the instruction decoder, the 4 bit MSB 0000 makes the output of the first line of decoder (a) to be HIGH. This in-turn will enable the adder-subtracter block HIGH. Similarly, the LSB of the op-code being *0000* makes the output of the first line of the decoder (b) to be *HIGH*. This results in the selection of the registers P and Q which contains the data. If the data at register P and register Q are 10111010 and 1001001 respectively an output of 10000011 is obtained.

The various functional blocks of ALU are given below.

3.1 Manchester Carry Chain (MCC) adder

The MCC adder offers a fast, regular, and simple structure [6] and implemented using PDB based domino logic. The computations are performed for odd and even carry chain separately as shown in Figure 5 and Figure 6.

Let $C = c_{n-1}c_{n-2}\cdots c_1c_0$, (5)

$D = d_{n-1}d_{n-2}\cdots d_1d_0$, (6)

eq. (5),eq. (6) represents the two binary numbers to be added and eqn (7)

$E = e_{n-1}e_{n-2}\cdots e_1e_0$ (7)

be their sum.

And the carry signal represented in eq. (8) as,

$f_i=h_i+k_i.f_{i-1}$. (8)

Here, $h_i = c_i \cdot d_i$ and k_i are the carry generate and the carry propagate terms.

The carry could be calculated either using inclusive OR or exclusive OR to facilitate the speed of operation. For inclusive OR adders, it is defined as $k_i = o_i = c_i + d_i$. And for exclusive OR adders, it is defined as $k_i = r_i = c_i \oplus d_i$. The length of carry chains is limited to only 4 bits due to signal clamping.

The new generate and propagate equations given in eq.(9), eq.(10).

$H_i = h_i+h_{i-1}$ (9)

and

$R_i = r_i \cdot r_{i-1} \cdot o_{i-2}$. (10)

For i = 0, $k_0 = o_0$, $f_0 = h_0+o_0 f_{-1}$.

As $h_i = h_i \cdot o_i$,

we obtain $f_0=o_0\cdot(h_0+f_{-1}) = o_0 \cdot m_0$. (11)

The even carry equations are

$m_0=h_0+f_{-1}$ (12)

$m_2 = h_2+h_1+r_2r_1o_0\,(h_0+f_{-1})$ (13)

$m_4 = h_4+h_3+r_4r_3o_2\,(h_2+h_1+r_2r_1o_0\,(h_0+f_{-1}))$ (14)

$m_6=h_6+h_5+r_6r_5o_4(h_4+h_3+r_4r_3o_2(h_2+h_1+r_2r_1o_0(h_0+f_{-1}))$ (15)

the structure pertaining to the above eq.(12), (13), (14) and (15) are shown in Figure 5.

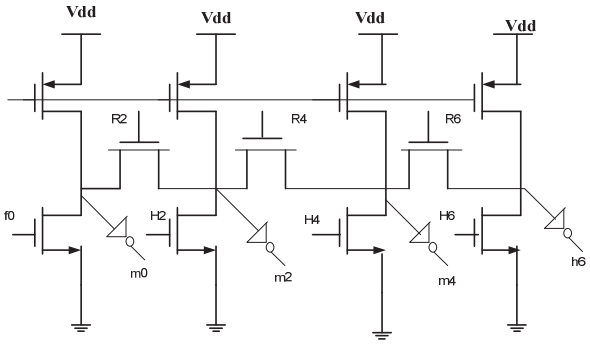

Figure 5. Even carry chain.

Odd carry equations are given by the following eq.(16), (17), (18) and (19) are depicted in Figure 6.

$m_1 = h_1+ h_0+r_1r_0f_{-1}$ (16)

$m_3 = h_3+h_2+r_3r_2o_1\,(h_1+h_0+r_1r_0f_{-1})$ (17)

$m_5 = h_5+h_4+r_5r_4o_3\,(h_3+h_2+r_3r_2o_1\,(h_1+h_0+r_1r_0f_{-1}))$ (18)

$m_7=h_7+h_6+r_7r_6o_4(h_5+h_4+r_5r_4o_3(h_3hh_2+r_3r_2o_1(h_1+h_0+r_1r_0 f_{-1})))$ (19)

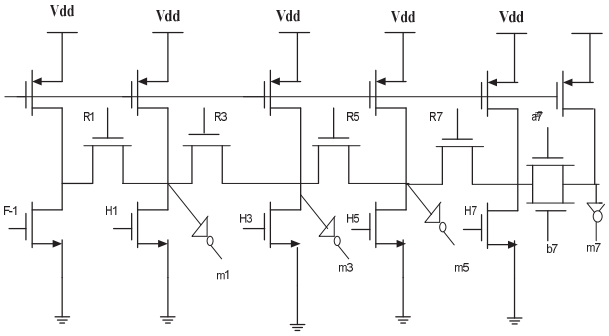

Figure 6. Odd carry chain.

The sum equation is eq.(20) and is shown in Figure 7.

$E_i = m_{i-1}\, r_i + m_{i-1}(r_i \oplus o_{i-1})$ (20)

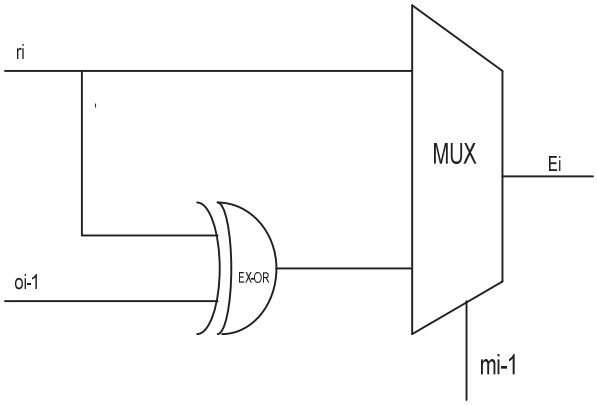

Figure 7. Sum bit generator.

3.2 ADD/SUB block

This block performs both addition and subtraction based on C_{-1} value. If C_{-1} is *0,* it performs addition and if it is *1,* it performs subtraction. Figure 8 shows the block diagram of ADD/SUB unit. It contains inverters and 2x1muxltiplexers whose select lines are connected to C_{-1}. If C_{-1} is *0* then the data D is fed directly to the ADD/SUB block and performs addition. If C_{-1} is *1,* then the data D is complimented and added which is basically the subtraction operation. *E0 -E7* represents the output of the ADD/SUB block[7].

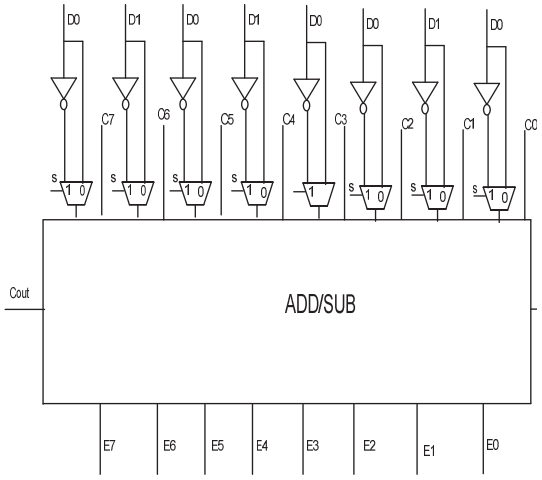

Figure 8. ADD/SUB block .

3.3 Booth Multiplier

Booth multiplication overcomes the drawbacks of array multiplication as two signed number multiplication could be performed at reduced circuitry and high speed [8]. This algorithm is applicable for both positive and negative numbers. Figure 9 shows the flow chart of an 8 bit Booth multiplier and the Figure 10 shows the block diagram of 8 bit Booth multiplier [9].

As the multiplication operation comprises repeated additions, subtractions and shifting of bits, the Booth multiplier unit consists of adder, subtracter and arithmetic right shift block (ASHR). The C_{-1} signal which is the output of MUX determines addition or subtraction operation. The ADD/SUB receives the inputs bits for multiplication and this block gets activated with the carry C_{-1}.If C_{-1} is 0, it performs addition and if C_{-1} is 1 it performs subtraction. An arithmetic shift operation is performed using ASHR unit. Then, the LSB's Q_n and Q_{n+1} bits are compared and depending on that appropriate operations are performed which is shown in Table 1.

TABLE I
BOOTH MULTIPLIER OPERATION

Q_n	Q_{n+1}	Operation
0	0	Arithmetic shift right
1	0	Addition and Arithmetic shift right
1	1	Subtraction and Arithmetic shift right
1	1	Arithmetic shift right

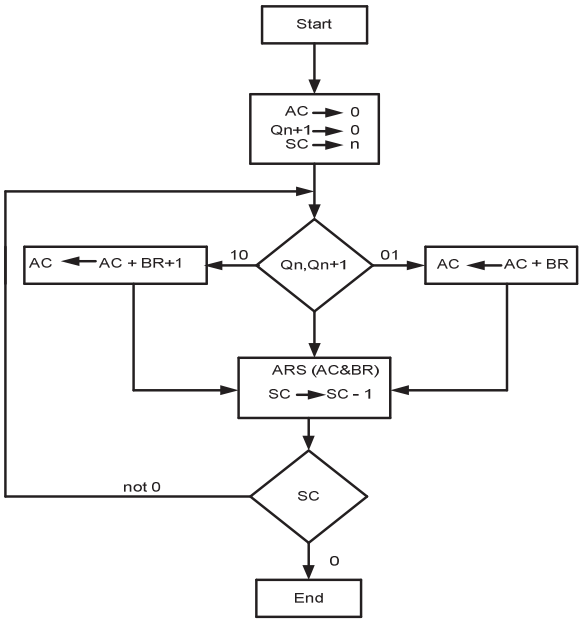

Figure 9. Flow chart of Booth Multiplier .

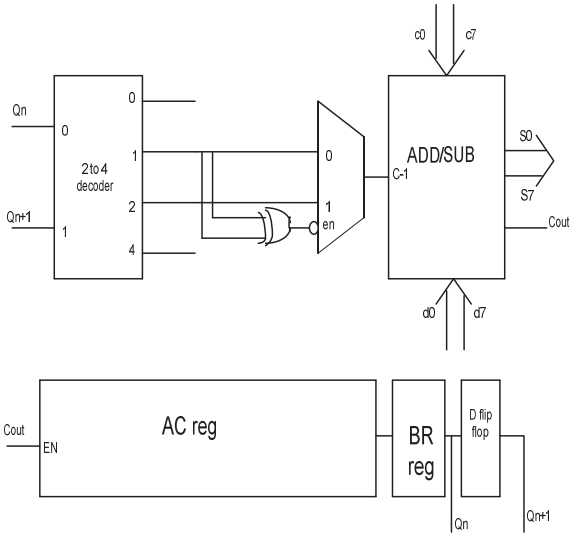

Figure 10. Block diagram of Booth multiplier.

4. Simulation and Analysis

The various sub blocks of the ALU design are designed and simulated using PDB based domino logic design style using Cadence® Virtuoso at 180nm library node. The power and delay are calculated for 8 bit MCC adder, 8 bit adder-subtraction and Booth multiplier as in Table II. The graphical representation of the dynamic power is shown in Figure 11 for various arithmetic blocks of the PDB.

TABLE II
Power comparison of the ALU Arithmetic Units

Module	Dynamic power(mw)	Leakage power(pw)	Delay(ps)
MCC Adder	6.077	110.25	75.61
ADD/SUB	6.23	125.02	81.78
Booth multiplier	8.03	234.69	106.87

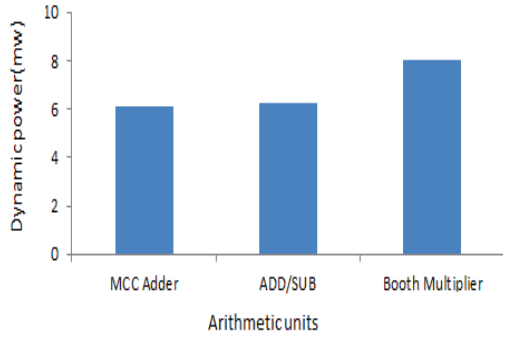

Figure 11. Dynamic power comparisons of ALU

power and delay are calculated for 8 bit AND gate, 8 bit OR gate, 8 bit NAND gate, 8 bit NOR gate, 8 bit XOR gate and 8 bit XNOR gate. Table III shows power and delay for these logical circuits. Also, the graphical representation of the dynamic power for logical blocks of ALU is depicted in Figure 12.

TABLE III
Power comparison of 8 input Logical Units

Module	Dynamic power(uw)	Leakage power(pw)	Delay(ps)
AND	16.08	98.45	29.36
OR	12.52	103.26	32.54
NAND	11.91	114.47	60.59
NOR	12.93	102.96	75.45
XOR	22.63	154.01	89.17
XNOR	33.28	178.18	95.42

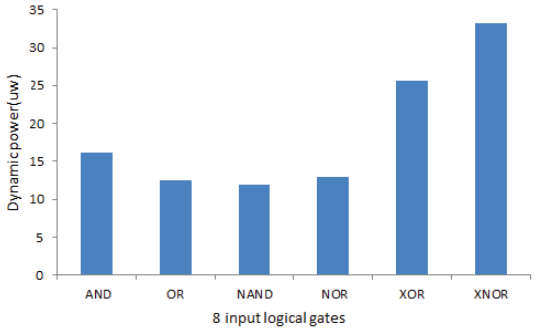

Figure 12. Dynamic power comparisons of ALU_Logical Units.

Table IV shows power and delay of ALU for different instructions. ALU is fed with different instructions, and the instruction execution power and delay of a PDB based ALU is calculated. When different instructions are fed to the ALU block, the unit specific to the instruction gets activated. For example, when 00010000 if fed to the instruction register of ALU, this instruction is decoded, the data is fetched from the appropriate registers and subtraction using ADD/SUB block is performed. Thus, it leads to the variation in the power consumption for various instructions.

TABLE IV
Power comparison of ALU for different instructions

Instruction	Operation	Dynamic power(mw)	Leakage power(pw)	Delay(ps)
00000000	ADD	38.19	581.56	108.1
00010000	SUB	38.20	590.13	116.8
00100000	MUL	44.26	634.61	152.7
00110000	AND	1.676	269.23	56.85
01000000	OR	1.32	289.54	34.92
01010000	NAND	1.89	353.45	80.14
01100000	NOR	1.76	376.23	76.29
01110000	XOR	2.41	323.12	89.16
10000000	XNOR	2.86	367.18	94.12

The switching power incurred in the conventional domino logic during the pre-charge phase is eliminated using PDB design. Also, the charge sharing and charge leakage problems are eliminated using PMOS keeper transistor at the output. The ALU consumes maximum power of 44.26mw for the multiplication operation. This is because of Booth Multiplier block getting activated which has the highest hardware circuitry. Similarly, the OR operation consumes reduced power of 1.32mw owing to the minimal hardware.

6. Conclusion

Arithmetic and logical unit (ALU) using Pseudo Dynamic Buffer based domino logic is designed and implemented using Cadence® Virtuoso tool at 180 nm technology library. The switching power incurred in the conventional domino logic is minimized in the PDB based circuitry by reducing the transitions (which occurs during the pre-charge phase) at the output node. The power and delay for ALU module and its sub-blocks are tabulated and it is inferred that the multiplication operation consumes the maximum power of 44.26mw with a delay of 152.7 ps. The incorporation of various other low power techniques will further decrease the power consumption to a greater extent.

References

1. F. Mendoza-Herna´ndez, M. Linares-Aranda and V. Champac Noise-tolerance improvement in dynamic CMOS logic circuits IEE Proc.-Circuits Devices Syst., Vol. 153, No. 6, December 2006.
2. Fang Tang , AmineBermak,ZhouyeGu Low power dynamic logic circuit design using a pseudo dynamic buffer using a INTEGRATION, the VLSI journal 45 (2012) 395–404.

3. Fang Tang, Amine Bermak Lower-power TSPC-based Domino Logic Circuit Design with 2/3 Clock Load Energy Procedia 14 (2012) 1168 – 1174.
4. Ramdas Bhanudas Khaladkar, A. Anita Angeline and V. S. Kanchana Bhaaskaran Dynamic Logic ALU Design with Reduced Switching Power, Indian Journal of Science and Technology, Vol 8(20), IPL0194, August 2015.
5. M. Anders, S. Mathew, B. Bloechel, S. Thompson, R. Krishnamurthy, K. Soumyanath, S. Borkar, A 6.5 GHz 130 nm single-ended dynamic ALU and instruction-scheduler loop, IEEE ISSCC (2002) 410–411.
6. G. Vijayakumar, M. Poorani Swasthika, S. Valarmathi, and A. Vidhyasekar, "Implementation of Carry Look-Ahead in Domino Logic" International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 12, December – 2013
7. K. Vishnuvardhan Rao*, A. Anita Angeline and V. S. Kanchana Bhaaskaran, Design of a 16 Bit RISC Processor, Indian Journal of Science and Technology, Vol 8(20), IPL0221, August 2015
8. Xu-guang Sun, Zhi-gang Mao, Feng-chang Lai, A 64 bit parallel CMOS adder for high performance processors, in: Proceedings of the IEEE Asia-Pacific Conference on ASIC, 2002, pp. 205–208.
9. Hwang Wei, R.V. Joshi, W.H. Henkels, A 500-MHz, 32-word64-bit, eight-port self-resetting CMOS register file, IEEE Journal of Solid-State Circuits Jan. (1999) 56–67.