

RESEARCH

Open Access



Post algebras and ternary adders

Daniel Etiemble 

*Correspondence:
de@lri.fr

LISN, Bat 650, University Paris
Saclay, 91190 Gif sur Yvette,
France

Abstract

Except for qubits for which the different possible values are unordered, the different values of m -valued circuits either with voltage levels, current levels or charge levels are totally ordered. Either at the Math level (Post algebras) or at the circuit level, it means that each multiple valued level must be decomposed into binary levels, processed with binary computation and finally converted into a multiple valued level. Using ternary adders as example, we show that the ternary-to-binary decoding and binary encoding should be applied to the whole adder or to restricted parts of the adder. The second approach using multiplexers leads to the most efficient ternary adders. However, a comparison with binary adders shows that the ternary-to-binary and binary-to-ternary conversions is the reason for which the binary adders are more efficient.

Keywords: Post algebra, Ternary adders, Binary adders, Propagation delays, Power dissipation, Chip area

Introduction

Except for qubits, for which the different possible values are unordered, the different values of m -valued circuits are totally ordered. This is true whatever electrical support is used: voltage levels, current levels, number of charges. It means that the algebras corresponding to these m different values are some flavor of Post algebras. All variants of Post algebras decompose each multiple value into binary values. It means that the m -valued circuits use m -valued to binary decoders and binary to m -valued encoders. In this paper, the considered ternary full adders are compared to the corresponding binary ones. Two implementations of the ternary full adder are considered: a naive one and a MUX-based one. In both cases, ternary-to-binary and binary-to-ternary conversions are used. The results can be easily extended to ternary multipliers or extended to quaternary adders or multipliers.

The paper is organized as follows:

- Post algebra with 3 values is first presented.
- Two opposite techniques to synthesize a ternary full adder are then presented
- The methodology to compare MUX-based ternary adders and binary adders is presented
- The performance of a ternary full adder using a “state-of-the art” technique is compared with the performances of binary full adders.
- A 6-bit carry propagate adder (CPA) is compared with a 4-trit CPA.

- The conclusion summarizes why ternary adders are less efficient than the corresponding binary ones processing the same amount of information.

Post algebras

Post algebra has been introduced in 1921 [1]. There exist several systems of Post algebra, which are isomorphic [2]. The monotonic system of Post algebra is used, as it is the most suitable for circuit implementation.

Monotonic system of Post algebra when $m = 3$

The presentation is limited to $m = 3$ as the implementation of ternary circuits is studied.

Definition: Let $m = 3$. A monotonic algebraic system is a distributive lattice M with a null element 0 and a universal element 2 for which the following axioms are verified. To be consistent with the logical operators that will be presented later, the notation of the axioms is slightly changed while keeping their meaning.

Axiom 1: M has 3 elements e_0, e_1 et e_2 such as

- $0 = e_0 < e_1 < e_2 = 2$
- if $x, e_i \in M$ and $x.e_i = 0 (i \neq 0)$, then $x = 0$
- if $x, e_i, e_j \in M$ and $x + e_i = e_j (i < j)$, then $x = e_j$

Axiom 2: There exist a set of unary operators $X_n(x), X_p(x), \overline{X_n}(x), \overline{X_p}(x)$ such as

- $X_n(x) = 2$ if $x < 1$ else 0 if $x \geq 1$
- $X_p(x) = 2$ if $x < 2$ else 0 if $x = 2$
- $\overline{X_n}(x) = 0$ if $x < 1$ else 2 if $x \geq 1$
- $\overline{X_p}(x) = 0$ if $x < 2$ else 2 if $x = 2$

The unary operators translate a ternary input into a binary output, as shown in Table 1. The gates that implement the X_n and X_p unary operators are called negative inverter (NI) and positive inverter (PI). They are presented in Fig. 1. The binary-to-ternary conversion is implemented by the circuit shown in Fig. 2 corresponding to Table 3.

Synthesis of a ternary function

Let consider the example of the unary ternary function shown in Table 2.

$y = y_2 + y_1$ where y_2 is $y(a)$ for which $y=2$ and y_1 is $y(a)$ for which $y=1$.

- $y_2 = a_0 = an$

Table 1 Post-unary operators when $m=3$

X	Xn	\overline{Xn}	Xp	\overline{Xp}
0	2	0	2	0
1	0	2	2	0
2	0	2	0	2

Table 2 Ternary complement

a	y
0	2
1	1
2	0

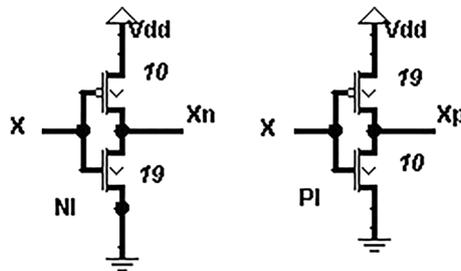


Fig. 1 Threshold detectors

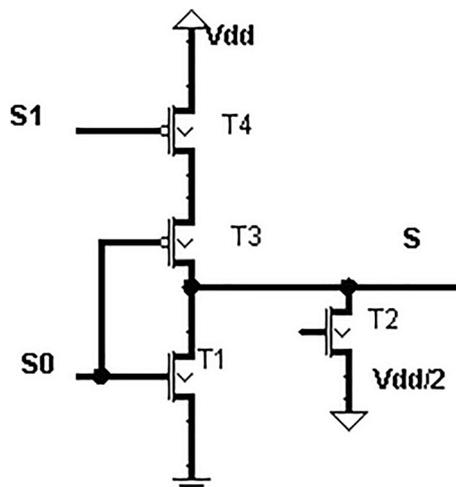


Fig. 2 Encoder circuit for the direct implementation

- $y_1 = a_1 = \overline{an}.ap$

$$y = an + \overline{an}.ap$$

While the unary operators $an, ap, \overline{an}, \overline{ap}$ are the ternary-to-binary decoders, the output of the function is obtained by a binary-to-ternary encoder. (y_1 and y_2 are the binary inputs of this encoder.)

Synthesis of a ternary full adder

The truth table of a ternary full adder is presented in Table 4. A, B and S are the ternary inputs and output, while C_{in} and C_{out} are the binary carries. It should be mentioned that ternary adders have binary carries and not ternary ones. While

ternary-to-binary decoding and binary-to-ternary encoding are mandatory, there are two opposite techniques to implement a ternary adder.

Direct implementation

The direct implementation corresponds to the general scheme of m-valued circuits presented in Fig. 3. The following notations are used: $A_i/B_i/S_i$ corresponds to $A/B/S=i$ ($i=0,1,2$). According to Table 4, when $C_{in}=0$, then

- $S_{0C_0} = A_0B_0 + A_1B_2 + A_2B_1$
- $S_{1C_0} = A_0B_1 + A_1B_0 + A_2B_2$
- $S_{2C_0} = A_0B_2 + A_1B_1 + A_2B_0$
- $C_{outC_0} = A_2B_1 + A_1B_2 + A_2B_2$

When $C_{in}=1$, then

- $S_{0C_1} = S_{2C_0}$
- $S_{1C_1} = S_{0C_0}$
- $S_{2C_1} = S_{1C_0}$
- $C_{outC_1} = A_2+B_2+A_1B_1$

In any case,

- $A_0 = A_n, A_1 = \overline{A_n}.A_p, A_2 = \overline{A_p}$
- $B_0 = B_n, B_1 = \overline{B_n}.B_p, B_2 = \overline{B_p}$

The methodology used to implement and simulate the ternary circuits will be detailed in the section Methodology. For the moment, we just mention

- CNTFET technology is used. It has the same circuit styles than CMOS technology.
- Ternary circuits are implemented with two power supplies V_{dd} and $V_{dd}/2$ as ternary circuits with only one power supply exhibit static power dissipation for level 1.

Two possible implementations can be considered for the direct approach:

Implementation with $A_0, A_1, A_2, B_0, B_1, B_2$

The corresponding sum circuit is shown in Fig. 4. It directly corresponds the previously written equations. The circuit is divided in three parts.

- A and B ternary inputs are decomposed into A_0, A_1, A_2, B_0, B_1 and B_2 binary outputs. A_n, A_p, B_n and B_p are the outputs of the circuits shown in Fig. 1 that implement the

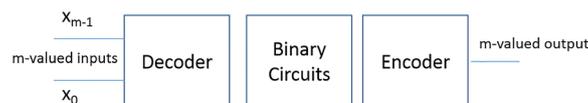


Fig. 3 General scheme of m-valued circuits

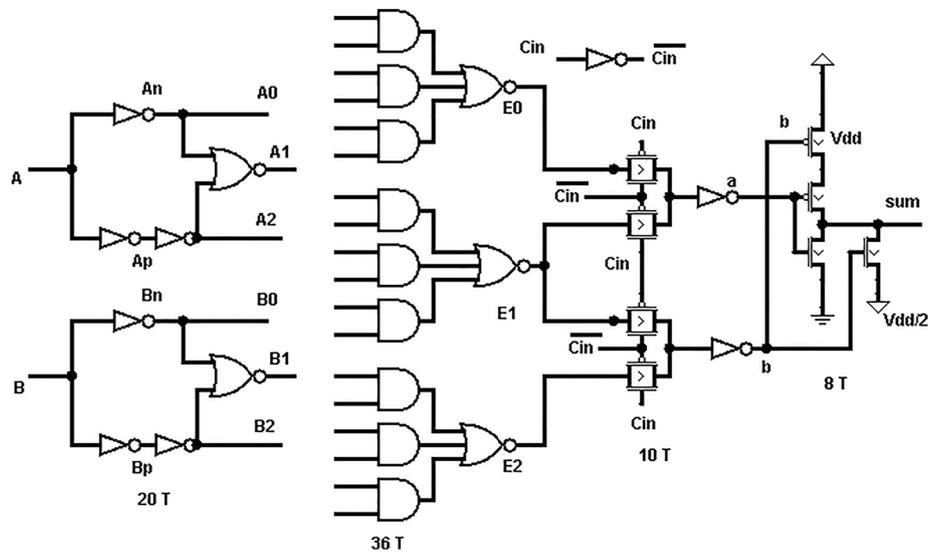


Fig. 4 Sum circuit-version 1

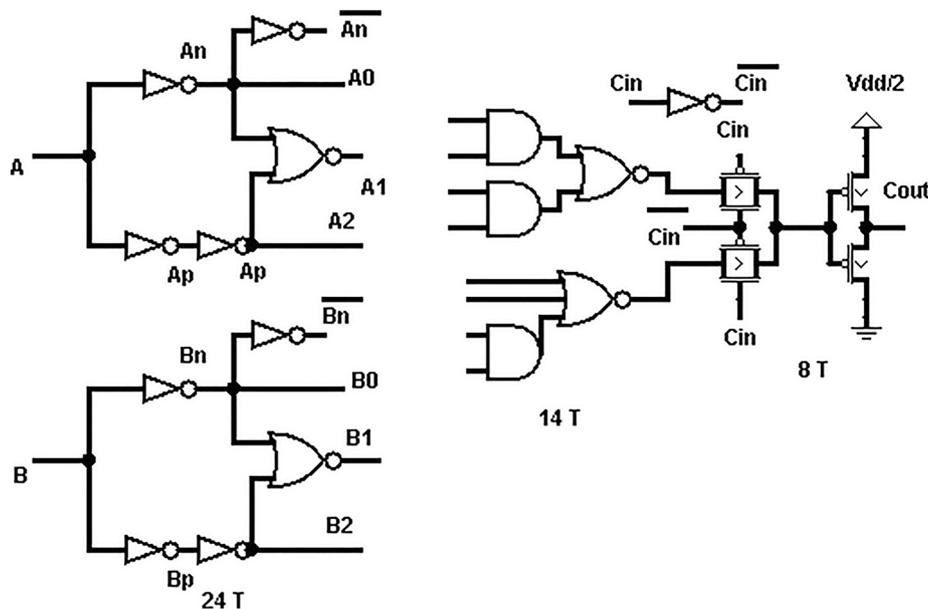


Fig. 5 Carry circuit-version 1

unary functions of Table 1. The inverters and NOR gates use the typical CMOS circuit style.

- The second binary part first computes $\overline{S0C0}$, $\overline{S1C0}$, $\overline{S2C0}$ using complex gates (combination of series/parallel patterns of transistors). Two multiplexers are controlled by C_{in} switches $\overline{S0C0}$, $\overline{S1C0}$, $\overline{S2C0}$ outputs to a and b inputs of the final encoder.
- The final encoder is presented in Fig. 2.

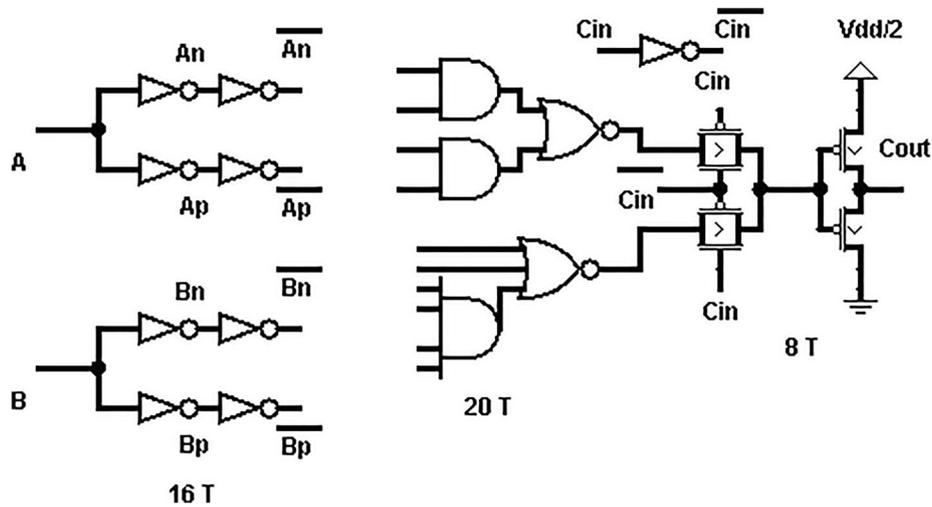


Fig. 7 Carry circuit-version 2

Table 3 Ternary encoder

S1	S0	S	T1	T2	T3	S4
0	0	2	Off	Off	On	On
2	0	1	Off	On	On	Off
0	2	0	On	Off	Off	On

Table 4 Truth table of a ternary full adder

$C_{in}=0$				$C_{in}=1$			
A	B	S_{C0}	C_{out0}	A	B	S_{C1}	C_{out1}
0	0	0	0	0	0	1	0
0	1	1	0	0	1	2	0
0	2	2	0	0	2	0	1
1	0	1	0	1	0	2	0
1	1	2	0	1	1	0	1
1	2	0	1	1	2	1	1
2	0	2	0	2	0	0	1
2	1	0	1	2	1	1	1
2	2	1	1	2	2	2	1

- When $B=2$, then $Sum = (A+2) \text{ mod}(3)$ quoted as A^2
- When $B=0$, then $C_{out}=0$
- When $B=1$, then $C_{out}=1$ when $A = 2$ else 0
- When $B=2$, then $C_{out}=1$ when $A > 0$ else 0

When $C_{in}=1$

- When $B=0$, then $Sum=A^1$
- When $B=1$, then $Sum=A^2$

- When $B=2$, then $Sum=A$
- When $B=0$, then $C_{out}=1$ when $A = 2$ else 0
- When $B=1$, then $C_{out}=1$ when $A > 0$ else 0
- When $B=2$, then $C_{out}=1$

Post-ary functions (Table 1) are implemented by the threshold detectors shown in Fig. 1. The A^1 and A^2 operators (Fig. 8) are derived from A_n and A_p outputs of the threshold detectors. So the ternary-to-binary decoding (threshold detectors) and binary-to-ternary encoding (A^1 and A^2) process is limited to the generation of A^1 and A^2 outputs. Then, two 3-input MUXes are controlled by B switch A, A^1 , A^2 to Sum_0 and Sum_1 . Two other 3-input MUXes are controlled by B switch different binary carry values to $\overline{C_{out0}}$ and $\overline{C_{out1}}$. It should be noticed that these binary values are 0/2. One final MUX controlled by C_{in} switches either Sum_0 or Sum_1 to Sum , while another one switches either $\overline{C_{out0}}$ or $\overline{C_{out1}}$ to $\overline{C_{out}}$. The final 1/0 C_{out} is obtained using an inverter with $V_{dd}/2$ power supply.

The 3-input MUX circuit is shown in Fig. 9. The 2-input final MUXes are controlled by a binary value (C_{in}). They use the typical 2-input MUXes with binary control.

In Table 4, the binary input and output carry values are 0/1, while A and B inputs have 0/1/2 values. However, when implementing ternary adders, the carry levels can be 0 and $V_{dd}/2$ (corresponding to 0/1 values) or 0 and V_{dd} (corresponding to 0/2 values). V_{dd} carry swing can be used as C_{in} only controls the final MUXes and C_{out} can also have a V_{dd} swing. There are few differences between $V_{dd}/2$ and V_{dd} carry versions that are outlined in Fig. 10. The $V_{dd}/2$ version uses a NI inverter to get C_n , and the final carry inverter has a 0.45V power supply. For the V_{dd} version, C_{in} and C_{out} use inverters with V_{dd} power supply.

Some details should be mentioned:

- In Figs. 9 and 10, some inverters look redundant. The point is that NI and PI inverters (Fig. 1) have poor driving capabilities. The added inverters are used as buffers.
- The simplest circuit to get Sum and C_{out} with final MUXes is shown in Fig. 11. However, in carry propagate adders (CPAs) shown in Fig. 12, there could be a direct propagation of carry values through a series of transmission gates with the RC effect shown in Fig. 13 that degrade the switching and propagation delays. This is the reason why an inverter is used to improve the propagation delays (Fig. 14).

The transistor counts are, respectively, 50 T ($V_{dd}/2$ carry values) and 48 T (V_{dd} carry values).

Related works

A lot of ternary full adders have been published in the last decade [3–11]. They use different techniques quoted in Table 5 that range from direct implementation to MUX-based implementation. Transistor count is not a sufficient criterion to determine the best technique. However, considering Table 5 and a similar table comparing

Table 5 Proposed TFAs in the last decade

TFA/Year	CNTFETs Count	Technique
This work 2023	118 or 128	Decoders-Binary-Encoder
In [3] 2011	412	Decoders-Binary-Encoder
In [4] 2017	105	Two custom algorithm + TMUXes
In [5] 2017	74	TMUXes
In [6] 2018	89	TMUXes
In [7] 2018	98	TBDD algorithm
In [8] 2019	142	Unary ops +MUXes+Encoder
In [9] 2020	74	Pass transistors + MUXes
In [10] 2020	106	Modified Quine–McCluskey algorithm
In [11] 2021	54	Unary ops + Decoders + Transmission gates
This work 2023	50 or 48	Unary ops + Multiplexers

ternary half adders in [12], the technique using A^1 and A^2 operators and MUXes may be considered as the most efficient one.

Methodology to compare MUX-based ternary adders and binary ones

The significant figures to compare circuit designs include switching times, power dissipation, chip area, etc. The comparison is realized by using HSpice simulations and evaluating the chip area according to transistor sizes.

CNTFET technology

All simulations are done with the 32nm CNTFET parameters of Stanford library [13] as most papers presenting designs of ternary circuits in the last period use simulations with this 32 nm CNTFET technology. This allows us to compare our results with all published results on ternary circuits. One advantage of CNTFET technology is that the threshold levels of gates only depend on the diameter of individual transistors, which facilitates the design of m-valued circuits.

Propagation delays

In full adders, the important information is the propagation delay corresponding to the critical paths, i.e., from C_{in} or Inputs to C_{out} or *Sum*. For CPAs, the critical path is C_{in} to C_{out} , except for the first and last full adders. We will only present the propagation delays corresponding to the critical paths.

Power dissipation and power-delay product (PDP)

Both power dissipation and PDP directly depends on the duration of the input signals. It is important to use the same input signal for all designs. For all simulations, the input waveforms shown in Figs. 15, 16 and 17 are used. It has been verified that the delays for 0–2 or 2–0 ternary transitions are always less than for ternary transitions 0–1, 1–2, 2–1 or 1–0. These waveforms are used to compute the worst-case delays from Input (A or B) to *Sum*/ C_{out} and from C_{in} to *Sum*/ C_{out} .

Chip area

We use a rough evaluation of the chip area by summing the diameters of all the used transistors by each circuit. This rough evaluation is a little bit better than the transistor count. In this paper, the diameter values presented in Table 6 are used.

Circuit styles

Many techniques have been proposed to design full adders. Only techniques with the following properties are considered:

- No static power dissipation
- The circuit outputs have full swing. Reduced swings degrade noise margins and can degrade the operation of cascaded circuits, such as CPAs
- The circuits should have a sufficient driving capability.

Table 6 Transistor diameters

	<i>n</i>	Diameter(nm)
D1	10	0.783
D2	19	1.487
D3	29	2.27
D4	37	2.896

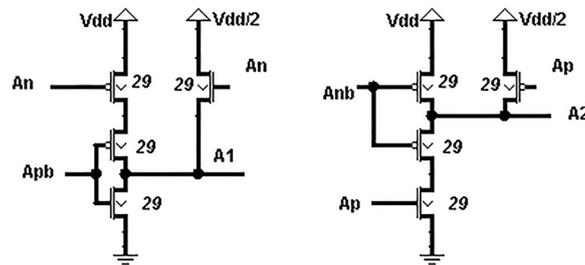


Fig. 8 A¹ and A² circuits

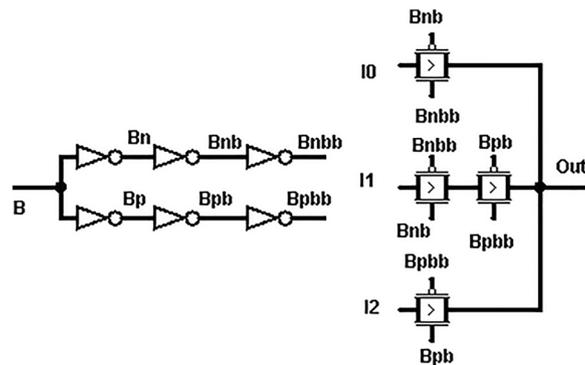


Fig. 9 3-Input MUX with ternary control

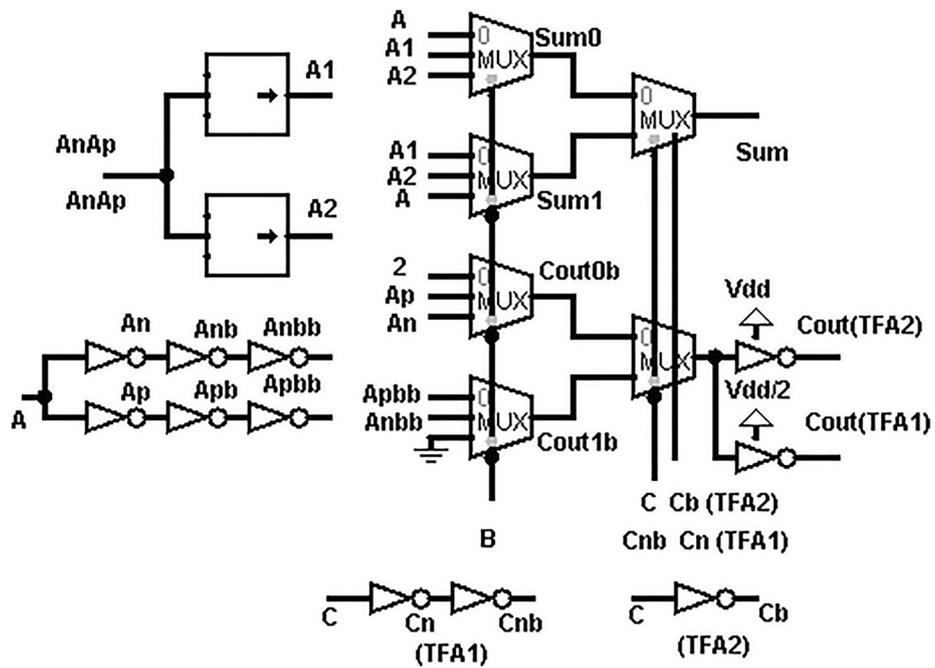


Fig. 10 1-Trit full adder (MUX approach)

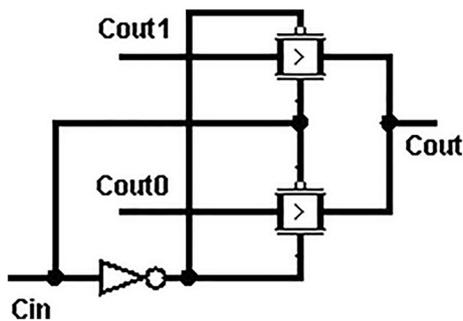


Fig. 11 C_{in} to C_{out} carry propagation in a full adder

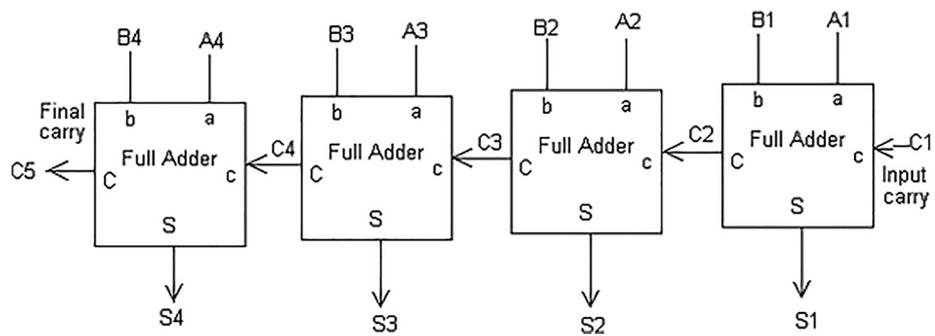


Fig. 12 4-Digit carry propagate adder

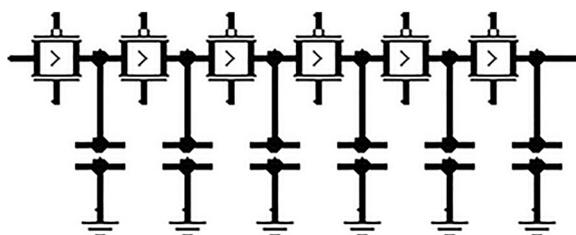


Fig. 13 RC effect with series of transmission gates

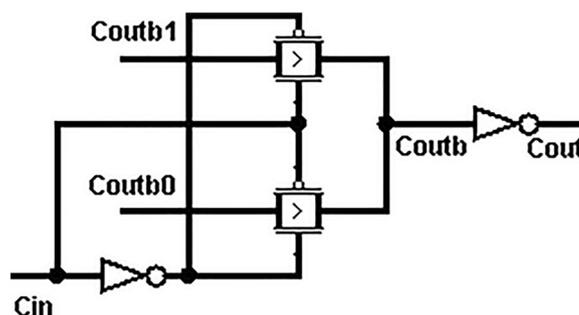


Fig. 14 C_{in} to C_{out} carry improved propagation with capacitive loads in a full adder

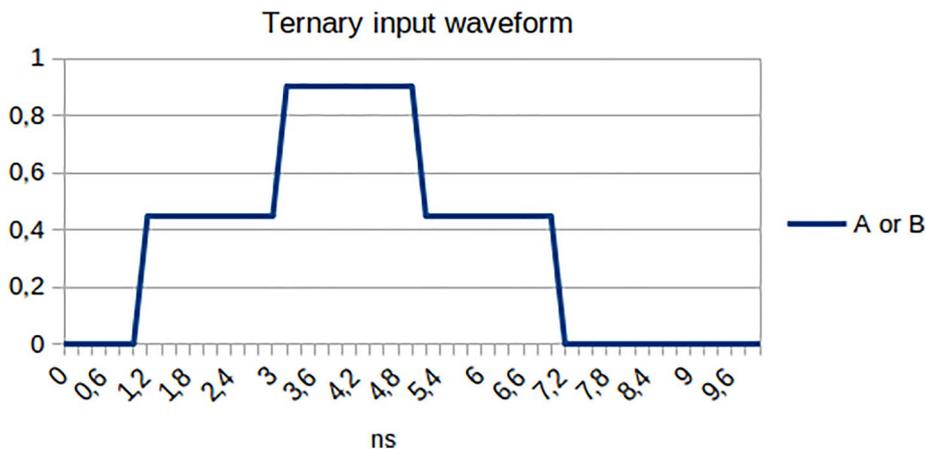


Fig. 15 Ternary input waveform

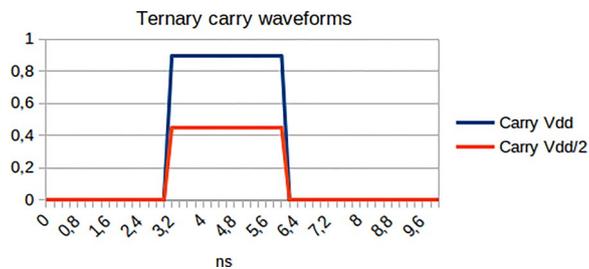


Fig. 16 Ternary carry waveforms

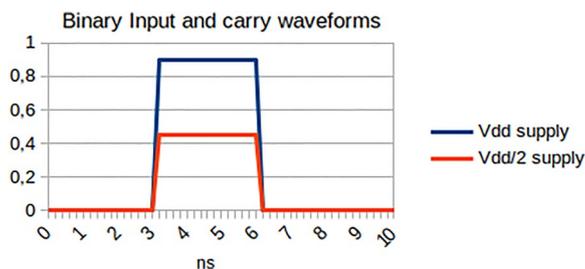


Fig. 17 Binary input and carry waveforms

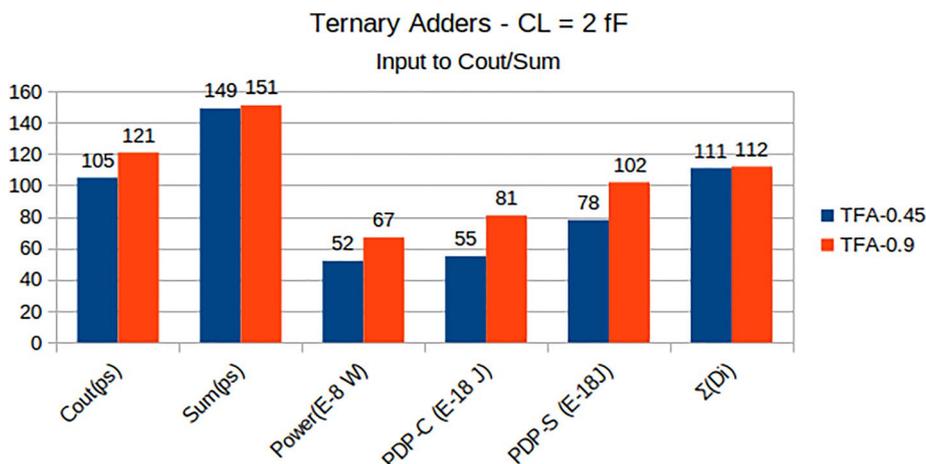


Fig. 18 Input to C_{out}/Sum performance of ternary adders with 0.45V and 0.9V carry values

Performance of the ternary full adder

We now present the simulation results for the two versions of the ternary full adder presented in Fig. 10: One version has V_{dd}/2 carry levels (quoted as 0.45), and the second one has V_{dd} carry levels (quoted as 0.9) as V_{dd}=0.9V.

Performance with a 2 fF capacitive load

Figure 18 presents the Input to C_{out}/Sum performance with a C_L = 2 fF capacitive load. Figure 19 presents the C_{in} to C_{out}/Sum performance with the same load.

The following remarks can be made when comparing V_{dd}/2 and V_{dd} carry swings

- Chip areas are equivalent
- For Input to C_{out}/Sum performance, the 0.45V version is slightly better than the 0.9V one.
- However, the 0.9V version is better for C_{in} to C_{out}/Sum performance. For C_{in} to C_{out} delay, which is the critical one in CPAs, the 0.9V delay is more than x2 reduced compared to the 0.45V version. The reason is that the final inverter with 0.9V power supply has more driving capability as the inverter with 0.45V power supply.

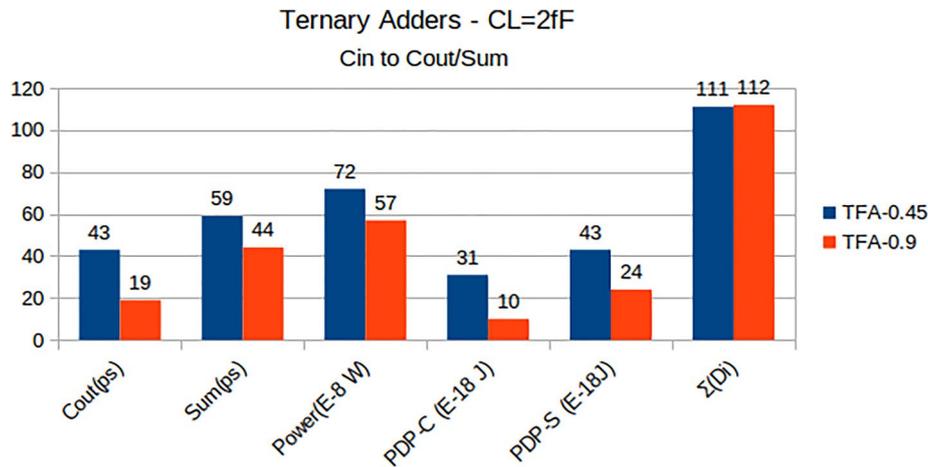


Fig. 19 C_{in} to C_{out} /Sum performance of ternary adders with 0.45V and 0.9V carry values

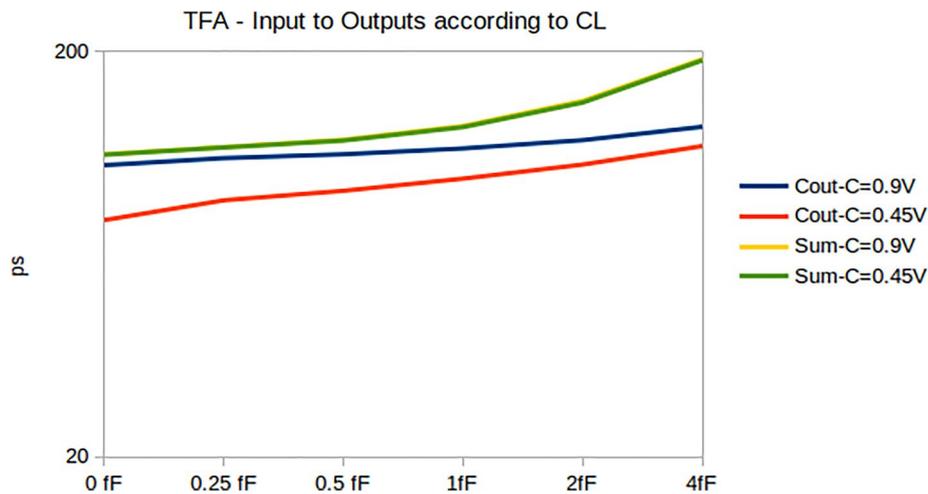


Fig. 20 TFA-Input to C_{out} /Sum delays according to C_L

Delays and power according to capacitive load

With a log–log scale (except for $C_L = 0$ fF), Fig. 20 presents the input to outputs delays according to C_L . Figure 21 presents the same information for C_{in} to outputs delay, while Fig. 22 presents the evolution of power according to C_L . Considering the different curves between $C_L = 0.25$ fF and $C_L = 4$ fF, it may be observed that the delay evolution is close to a linear one, with different slopes. Power increases more than linearly according to C_L .

C_{in} to C_{out} path is through a multiplexer and an inverter, while C_{in} to Sum is just through a multiplexer. The inverter restores the signal and has more driving capability than the multiplexer. It explains why the sum delay is more sensitive to capacitive load. Input to C_{out} and Sum paths include the whole circuit. The final inverter delay for C_{out} has a limited impact on the overall delay compared to Sum delay, which explain why these large delays do not increase much when C_L is multiplied by 16. Power increases from x2 to x3.

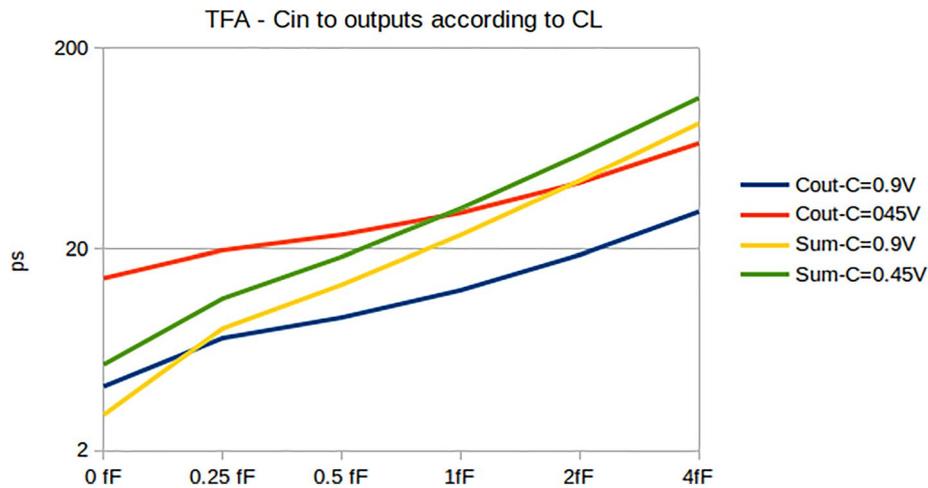


Fig. 21 TFA-C_{in} to C_{out}/Sum delays according to C_L

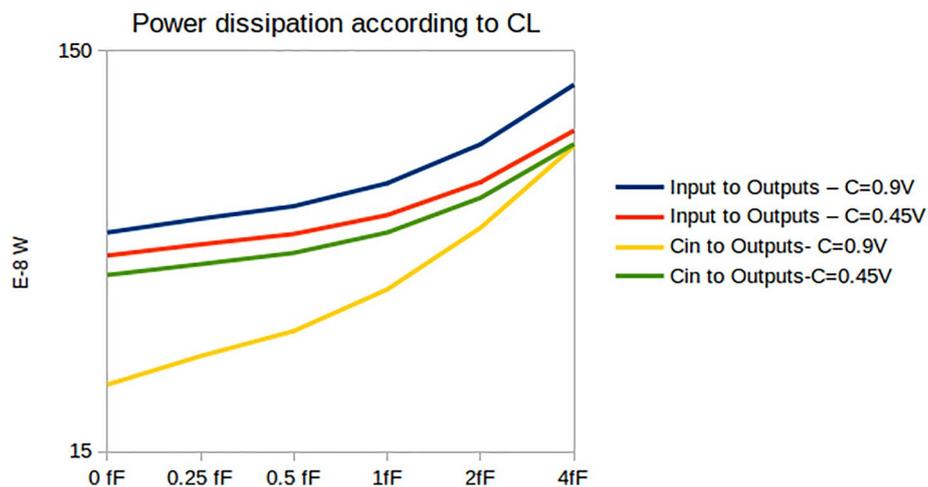


Fig. 22 TFA-Power dissipation according to C_L

The Binary Full Adders

The considered ternary adders have 2 power supplies: V_{dd} and $V_{dd}/2$. It means that some transistors operate with a $V_{dd}/2$ voltage swing. To compare the ternary adders with binary adders, it makes sense to use two different power supplies for the binary adders: either V_{dd} or $V_{dd}/2$. Using $V_{dd}/2$ instead of V_{dd} roughly divides by four the dynamic power dissipation.

The 14T binary full adder (BFA) presented in Fig. 23 is used. It corresponds to the following equations:

- Sum = $a \oplus b \oplus c$
- If $a \oplus b = 1$, then $C_{out} = C_{in}$ else $C_{out} = a$

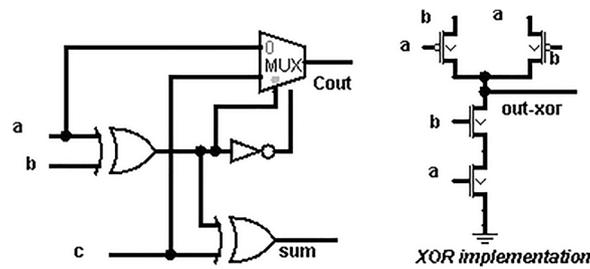


Fig. 23 14T Binary full adder-BFA

Performance with a 2 fF capacitive load

Figure 24 presents the Input to C_{out}/Sum performance with $C_L = 2$ fF. Figure 25 presents the C_{in} to C_{out}/Sum performance with the same capacitive load. All powers for 0.45 V_{dd} are roughly 1/4 of the powers of 0.9 V_{dd} versions, leading to PDP slightly smaller or equivalent for both V_{dd} . In [17], this binary adder has been compared with two other

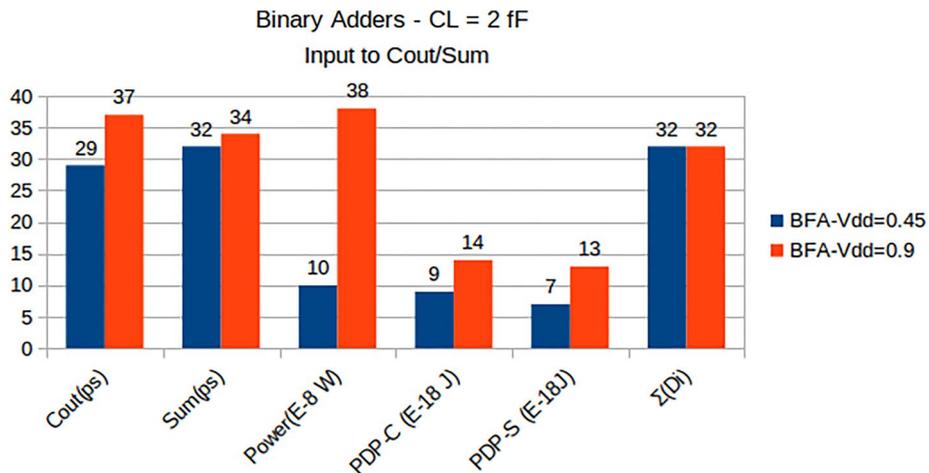


Fig. 24 Binary adders-Input to C_{out}/Sum - $C_L = 2$ fF

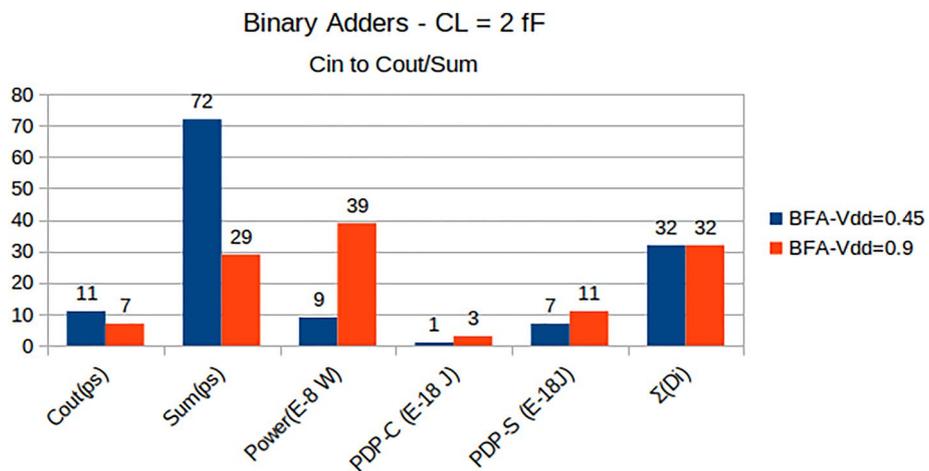


Fig. 25 Binary adders- C_{in} to C_{out}/Sum - $C_L = 2$ fF

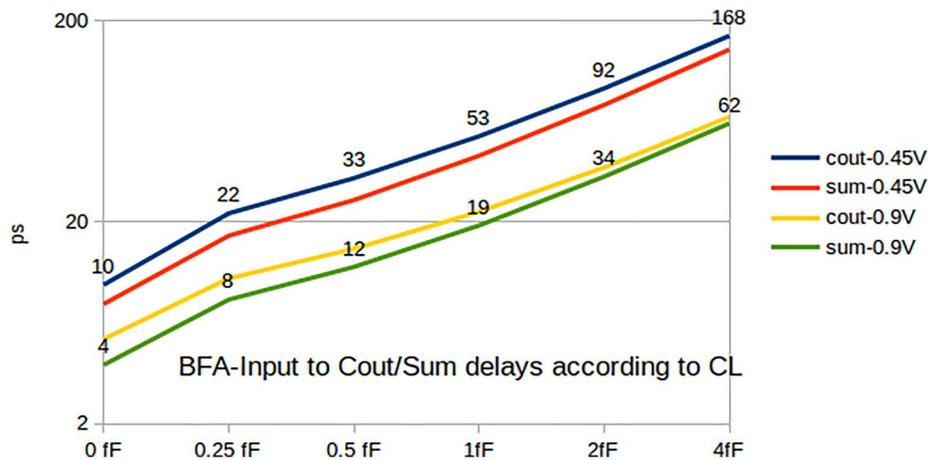


Fig. 26 BFA-input to C_{out}/Sum delays according to C_L

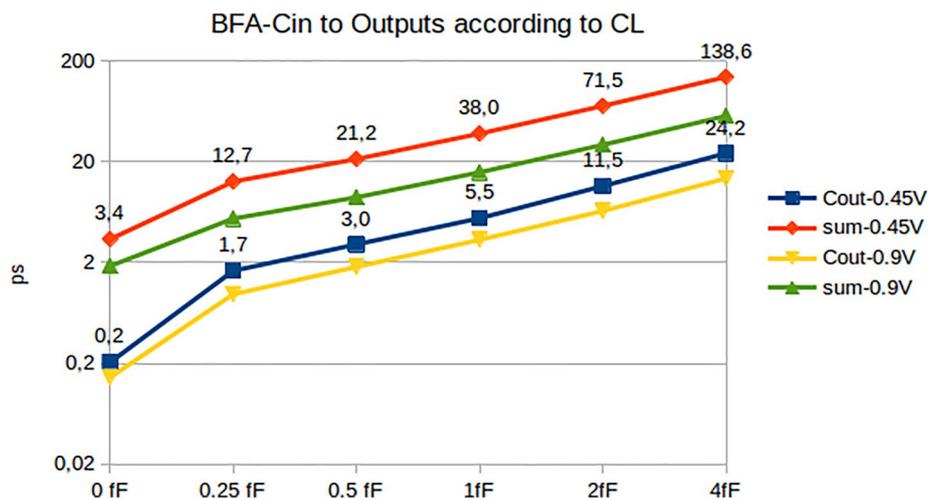


Fig. 27 BFA- C_{in} to C_{out}/Sum delays according to C_L

ones: the 28T typical CMOS implementation and a 34T MUX-based implementation. The simulated BFA (Fig. 23) is globally the most efficient one in terms of delays, PDP and ΣDi for the two different power supplies.

Delays and power according to capacitive load

The performance of the BFA according to capacitive loads are now presented. With a log–log scale, Fig. 26 presents the input to outputs delays according to C_L . Figure 27 presents the same information for C_{in} to outputs delays, while Fig. 28 presents the evolution of power according to C_L . There is a quasi-linear evolution of delay and power according to C_L . However, the binary adder structure is different of the MUX-based ternary adder structure: There is one MUX for C_{out} , but not a series of MUXes as in the Sum output of ternary adders. Globally, the binary adder is more sensitive to capacitive loads than the ternary ones.

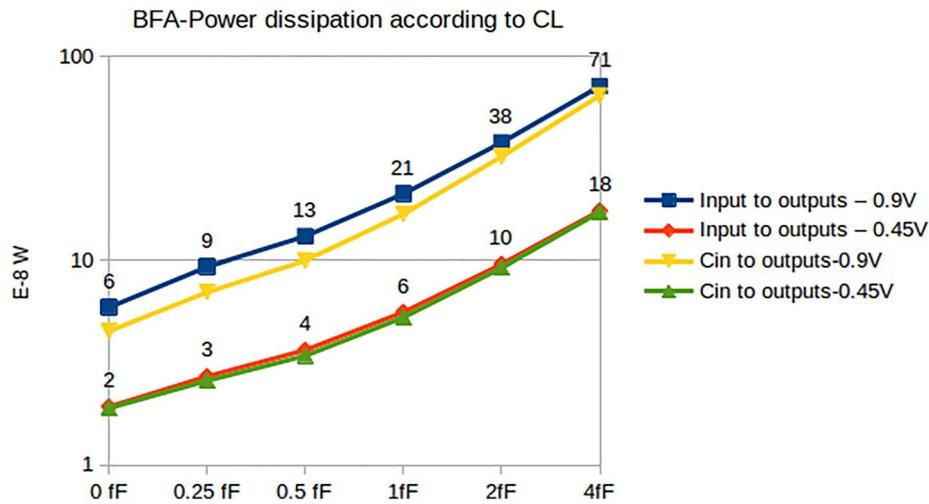


Fig. 28 BFA-power dissipation according to C_L

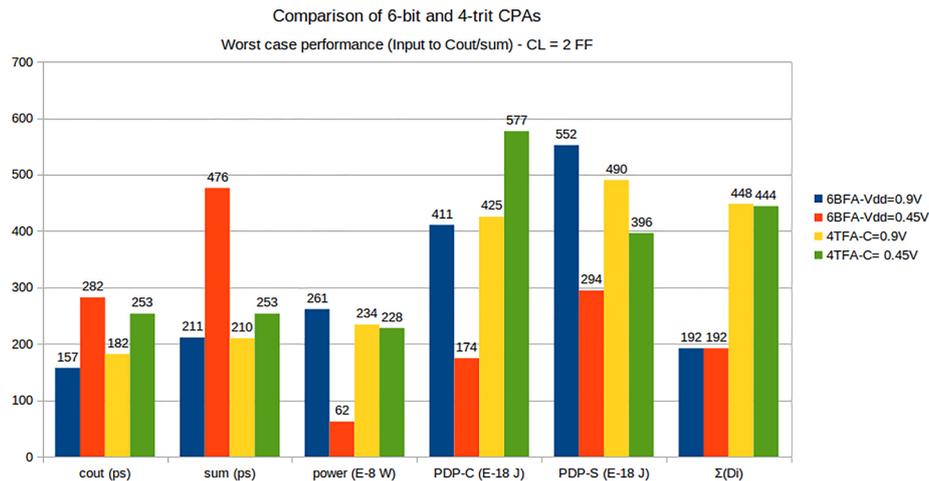


Fig. 29 Comparing 6-bit and 4-trit CPAs with $C_L = 2$ fF

Comparing 6-bit and 4-trit Carry Propagate Adders (CPAs)

The considered MUX-based ternary and binary adders can be used to build CPAs. The most significant information is to compare CPAs computing the same amount of information. 6-bit CPAs compute 6 bits of information, while 4-trit CPAs computes 6.34 bits of information, i.e., 6% more information.

Several 4-trit CPAs have been presented in the literature [5, 14, 15 and 16].

Both for binary and ternary adders, Input to C_{out} delay is greater than C_{in} to C_{out} delay. In CPAs, the critical path is thus from Input to C_{out} for the first adder, then C_{in} to C_{out} for the next ones and finally C_{in} to Sum for the last one. It means that Input to C_{out}/Sum provides the worst-case delays.

Figure 29 compares the performance of these two CPAs with the following variants: The ternary one uses $0-V_{dd}/2$ or $0-V_{dd}$ carry swing, and the binary one uses V_{dd} or $V_{dd}/2$ power supplies. The simulation has been done with a $C_L = 2$ fF capacitive load and

$T = 25^{\circ}\text{C}$ temperature. Other loads or temperatures would not change the results of the comparisons. From Fig. 29, the following conclusions can be deduced:

- While the binary CPA uses more full adders, its estimated chip area is $\times 0.45$ the chip area of the ternary CPAs.
- The ternary CPAs have less propagation delays when using full carry swing than when using $V_{dd}/2$ carry swing
- The $0.45 V_{dd}$ binary CPAs have the smallest power dissipation, from 1/2 to 1/4 power dissipation of the other CPAs. While its input to sum delay is the worst one, this CPA has the lowest PDP both for sum and carry outputs.

While ternary CPAs have less full adders, they suffer from larger chip areas and do not provide significant advantages in terms of delays. The best CPA is the binary one with $V_{dd} = 0.45V$ supply. Reducing power supply is possible with binary circuits, but is not possible with ternary circuits, as they would need a larger V_{dd} to handle the different voltage levels.

In this paper, binary and ternary CPAs have been compared. The overall results are similar for quaternary CPAs [17]. Paper [18] also shows that binary multipliers are more efficient than quaternary ones. It means that binary circuits are more efficient than ternary or quaternary ones to implement combinational circuits.

Concluding remarks

The ordered set of ternary values ($0 < 1 < 2$) implies using some flavor of Post algebras. The monotonic Post algebra is the best form to implement ternary circuits. With totally ordered set of values, ternary values should be decomposed into binary values (threshold decoders) and the binary values should be encoded as ternary values. Using binary computation within ternary circuits cannot be avoided. Two opposite approaches to implement ternary adders have been detailed:

- The naive approach decomposes A and B ternary inputs into binary A_i and B_i for which $A_i/B_i=2$ when $A/B=i$ (else $A_i/B_i=0$). Then, S_0 , S_1 and S_2 binary outputs are computed as functions of A_0 , A_1 , A_2 , B_0 , B_1 , B_2 . Finally, The ternary sum is computed by the final encoder as a function of S_0 , S_1 , S_2 and C_{in} . The output carry is computed using the same approach.
- The MUX-based approach limits the ternary-to-binary decoding and binary-to-ternary encoding to the implementation of A^1 and A^2 functions for which $A^1=(A+1) \bmod 3$ and $A^2=(A+2) \bmod 3$. Then, the ternary values A , A^1 and A^2 are switched to the output sum according to B and C_i values using multiplexers. The carry output is computed using the threshold decoder outputs and multiplexers.

It turns out that the MUX-based approach outperforms the naive one. All the proposed ternary adders in the last decade fits within these two opposite approaches. The proposed and simulated MUX-based ternary adder is probably close to the best possible one. Two possible implementations differ with the carry values: either $V_{dd}/2$ or V_{dd} . It should be mentioned that too long series of MUXes should be avoided as they degraded

the switching times and propagation delays. For CPAs that propagate carries through the successive full adders, the adder carry output should be restored by an inverter.

We have evaluated the performance of this ternary adder and a 14T binary one in terms of worst-case propagation delays, power and PDP for Input to C_{out}/Sum and C_{in} to C_{out}/Sum . The ternary and binary adders are compared with the implementation of a 6-bit CPA and a 4-trit CPA. These two CPAs compute approximately the same amount of information. Globally, the 4-trit CPAs are less efficient than the 6-bit CPAs:

- The ternary CPAs use more than 2x the binary chip areas
- When the ternary CPAs use a V_{dd} power supply, the binary ones can use either a V_{dd} or a $V_{dd}/2$ power supply. Using $V_{dd}/2$ power supply, the binary CPAs outperform the ternary ones in terms of power dissipation and PDP.

The fundamental weakness of ternary (and quaternary) combinational circuits comes from the mandatory ternary-to-binary decoding and binary-to-ternary encoding that exist both at the math (Post algebra) and the circuit levels. This allows to understand why ternary combinational circuits have been unsuccessful in the last 50 years.

Circuits using an ordered set of values can be successful in small niches. It is the case of m-valued flash memories that use different levels of electrical charges. 4-valued (MLC) flash memories store two bits per cell. 8-valued (TLC) memories store 3 bits per cell. In 2018, ADATA, Intel, Micron and Samsung have launched some SSD products using QLD NAND-memory with 4 bits per cell. They can be used as flash memory access times are not critical. While binary flash memories have the advantage of faster write speeds, lower power consumption and higher cell endurance, M-valued flash memories provide higher data density and lower costs

Author contributions

The author read and approved the final manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Received: 7 December 2022 Accepted: 13 March 2023

Published online: 27 March 2023

References

1. Post EL (1921) Introduction to a general theory of elementary propositions. *Am J Math* 43:163–185
2. Nutter RS, Swartwout RE, Rine DC (1974) Equivalence and transformation for post multivalued algebras. *IEEE Trans Comput* C23:294–300
3. Lin S, Kim Y-B, Lombardi F (2011) CNTFET-based design of ternary logic gates and arithmetic circuits. *IEEE Trans Nanotechnol* 10(2):217–225. <https://doi.org/10.1109/TNANO.2009.2036845>
4. Srinivasu B, Sridharan K (2017) A synthesis methodology for ternary logic circuits in emerging device technologies. *IEEE Trans Circuits Syst I* 64(8):2146–2159. <https://doi.org/10.1109/TCSI.2017.2686446>
5. Tabrizchi S, Panahi A, Sharifi F, Navi K, Bagherzadeh N (2017) Method for designing ternary adder cells based on CNFETs. *IET Cir Devices Syst* 11(5):465–470. <https://doi.org/10.1049/iet-cds.2016.0443>
6. Shahrom E, Hosseini SA (2018) A new low power multiplexer based ternary multiplier using CNTFETs. *AEU-Int J Electron C* 93:191–207. <https://doi.org/10.1016/j.aeue.2018.06.011>
7. Vudadha C, Surya A, Agrawal S, Srinivas MB (2018) Synthesis of ternary logic circuits using 2:1 multiplexers. *IEEE Trans Circ Syst I* 65(12):4313–4325. <https://doi.org/10.1109/TCSI.2018.2838258>
8. Sharma T, Kumre L (2019) CNTFET-based design of ternary arithmetic modules. *Circuits Syst Signal Process* 38(10):4640–4666. <https://doi.org/10.1007/s00034-019-01070-9>

9. Mahmoudi Salehabad I, Navi K, Hosseinzadeh M (2020) Two novel inverter-based ternary full adder cells using CNFETs for energy-efficient applications. *Int J Electron* 107(1):82–98. <https://doi.org/10.1080/00207217.2019.1636306>
10. Kim S, Lee S-Y, Park S, Kim KR, Kang S (2020) A logic synthesis methodology for low-power ternary logic circuits. *IEEE Trans Circuits Syst I Regul Pap* 67(9):3138–3151. <https://doi.org/10.1109/TCSI.2020.2990748>
11. Hosseini SA, Etezadi S (2021) A novel low-complexity and energy-efficient ternary full adder in nanoelectronics. *Circuits Syst Signal Process* 40(3):1314–1332. <https://doi.org/10.1007/s00034-020-01519-2>
12. Jaber RA, Owaidat B, Kassem A, Haidar AM (2020) A novel low-energy CNTFET-based ternary half-adder design using unary operators. In: 2020 International conference on innovation and intelligence for informatics, computing and technologies (3ICT), pp 1–6. <https://doi.org/10.1109/3ICT51146.2020.9311953>
13. Deng J, Wong H-P (2007) A compact SPICE model for carbon-nanotube field-effect transistors including nonidealities and its application-part II: full device model and circuit performance benchmarking. *IEEE Trans Electron Devices* 54(12):3195–3205. <https://doi.org/10.1109/TED.2007.909043>
14. Mahboob Sardroudi F, Habibi M, Moayeri MH (2021) A low-power dynamic ternary full adder using carbon nanotube field-effect transistors. *AEU-Int J Electron C* 131:153600. <https://doi.org/10.1016/j.aeue.2020.153600>
15. Hosseini SA, Etezadi S (2021) A novel low-complexity and energy-efficient ternary full adder in nanoelectronics. *Circuits Syst Signal Process* 40(3):1314–1332. <https://doi.org/10.1007/s00034-020-01519-2>
16. Jaber RA Two improved designs for ternary full adders using unary operators and ternary multiplexers. *Personal communication*
17. Etiemble D Ternary and Quaternary CNTFET Full Adders are less efficient than the corresponding binary ones for the Carry-Propagate Adders [arxiv:2207.04839](https://arxiv.org/abs/2207.04839)
18. Etiemble D CNTFET quaternary multipliers are less efficient than the corresponding binary ones [arxiv:2206.03252](https://arxiv.org/abs/2206.03252)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
