Comparing Unbalanced and Balanced CNTFET Ternary Adders and Multipliers with the corresponding binary ones

Original Research Article

***Abstract*—This paper compares the performance of the ternary adders and multipliers using balanced and unbalanced set of values. We use the 1-trit adders to evaluate the two versions of a 4-trit propagate adder, which are compared with a 6-bit corresponding propagate adder. Similarly, we compare the two types of 2*2 trit multipliers with a 3*3 bit multiplier. The simulations with a 32-nm Carbon Nanotuble Field-Effect Transistor (CNTFET) technology show that the binary adders and multipliers are more efficient than the ternary ones that compute the same amount of information.**

I. Introduction

Ternary circuits can use two different sets of values:

- Unbalanced circuits use 0, 1 and 2 ternary values corresponding to 0, $V_{dd}/2$ and V_{dd} levels.
- Balanced circuits use -1, 0 and 1 ternary values corresponding to $-V_{dd}/2$, 0 and $V_{dd}/2$ levels. They are generally quoted as N (Negative), Z (Zero) and P (Positive) to present the ternary numbers.

Most of the proposed ternary full adders use the unbalanced representation. Table I presents a comparison of ternary unbalanced full adders presented in the last decade [1]–[11]. Some balanced ternary adders and multipliers have been presented. [12]–[14]. Unbalanced ternary multipliers have been presented [15]–[19].

The paper is organized as follows:

- We present the methodology: (1) CNTFET technology, evaluation of propagation delays, power and power-delay product, chip area. (2) Mux-based approch that is used to implement ternary circuits is presented. (3) Binary adders used for comparison are also presented.
- Balanced and unbalanced ternary adders are presented: balanced ones have ternary carry values, while unbalanced ones that are used to implement carry-propagate adders (CPAs) have binary carry values. Their performance are presented and compared. 4-bit CPAs are compared with the two types of 3-trit CPAs.
- Balanced and unbalanced 1-trit multipliers are presented. The balanced one does not generate a carry, while the unbalanced one generates a carry. Their performance are presented and compared. Then a 3*3 bit multiplier is compared with the two types of 2*2 trit multipliers. Without simulation, the performance of larger multipliers are also evaluated.
- The conclusion summarizes the results and explains why the binary adders and multipliers are more efficient than the ternary ones that compute the same amount of information.

1

II. Methodology

A. CNTFET technology

We use CNTFET technology as it is used in most papers presenting ternary or quaternary implementations of adders, multipliers, counters [20], etc. 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. All simulations are done with the 32nm CNTFET parameters of Stanford library [21] that are used by most m-valued CNTFET designers.

B. Propagation delays

For all combinational circuits, the important information is the propagation delay corresponding to the critical paths. We will only present the propagation delays corresponding to these critical paths.

C. 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, we use the input waveforms shown in Fig. 1 and Fig. 2 for ternary circuits. We have verified that the delays for 0-2 or 2-0 ternary transitions are always less than ternary transitions 0-1, 1-2, 2-1 or 1-0. The binary waveforms used for the binary adders are presented in Fig. 3.

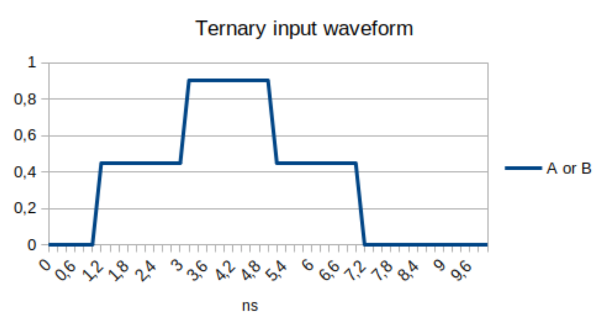

Fig. 1. Input waveforms for ternary circuits

D. 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, we use the diameter values presented in Table II.

TABLE I
TFAs COMPARISON

TFA / Year	CNTFETs Count	Power (μW)	Max. Delay (ps)	Max. PDP (x10^{-18} J)	Max. EDP (x10^{-27} J.s)	Technique
In [1] 2011	412	1.36	88	12	10.5	Decoder-Binary-Encoder
In [2] 2017	105	1.13	68	77	5.2	2 custom algorithms+TMuxes
In [3] 2017	74	0.82	146	120	17.5	TMUXes
In [4] 2018	98	0.16	192	31	5.9	TBDD algorithm
In [5] 2018	89	0.44	48	21	1	MUXes
In [6] 2019	142	4.62	94	434	40.8	Unary ops+MUXes+Encoder
In [7] 2020	106	0.13	269	35	9.4	Modified Quine-MCluskey algorithm
In [8] 2020	49	1.23	192	236	45.3	Majority-not based Full Adder
In [8] 2020	37	0.81	262	212	55.5	Majority-not based Full Adder
In [9] 2021	54	0.43	47	20	0.9	Unary ops + Decoders+Transmission gates
In [10] 2023 TFA1	59	0.46	27	12.5	0.3	Unary ops+MUXEs
In [10] 2023TFA2	55	0.22	34	7.5	0.25	Unary ops + Muxes
In [11] 2023	48/50					Unary ops+MUXES
In [11] 2023	118/128					Decoder-Binary-Encoder

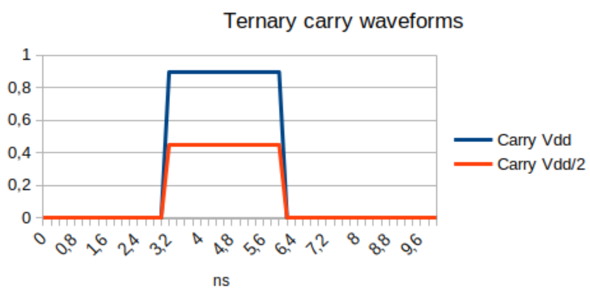

Fig. 2. Carry input waveforms for ternary circuits

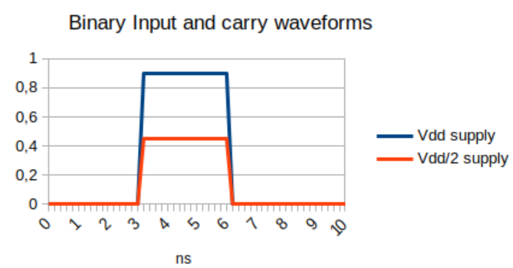

Fig. 3. Input waveforms for binary circuits

E. The Mux-based implementation

Implementation of ternary circuits can used different techniques: Paper [11] presents a detailed examination of the two opposite techniques, which are "Decoders-Binary-Encoders" and "MUX-based" implementations. The most efficient one is the MUX-based one, as shown by a detailed examination of Table I.

The MUX approach uses the operations that are shown in Table III. Ternary multiplexers are used to switch the correct values to the outputs. Similar circuits are used by unbalanced and balanced circuits.

A_n and A_p unary functions are implemented by the threshold detectors shown in Fig. 4. The A^1 and A^2 operators (Fig. 5) are derived from A_n and A_p outputs of the threshold detectors. The 3-input MUX circuit is shown in Fig. 6. Inverters delivering Bn and Bp are the threshold detectors presented in Fig. 4. Inverters delivering Bnb, Bnp, Bnpbb and Bpbb are usual binary inverters. Bnbb and Bpbb have higher driving capabilities than Bn and Bp.

TABLE II
TRANSISTOR DIAMETERS

	n	Diameter(nm)
D1	10	0.783
D2	19	1.487
D3	29	2.270
D4	37	2.896

TABLE III
BASIC OPERATIONS FOR IMPLEMENTING TERNARY CIRCUITS

A	A_n	A_p	A1	A2	A001	A011	Etc
0	2	2	1	2	0	0	
1	0	2	2	0	0	1	
2	0	0	0	1	1	1	

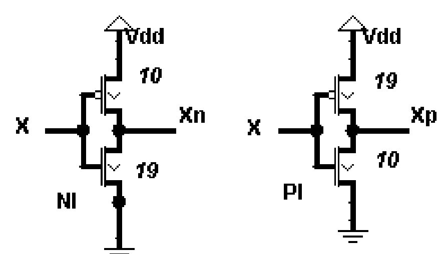

Fig. 4. Threshold detectors

F. Comparison with binary circuits

Any proposal of ternary circuits should be compared with the binary circuits computing the same amount of information, either to prove or disprove their interest:

- Ternary adders should be compared with binary adders.

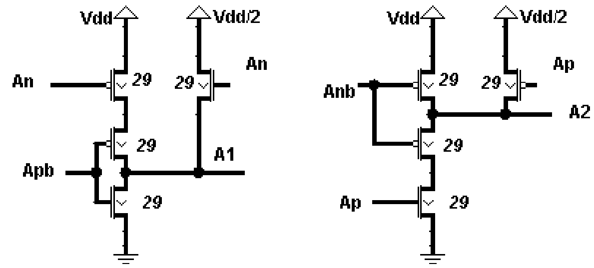

Fig. 5. A^1 and A^2 circuits

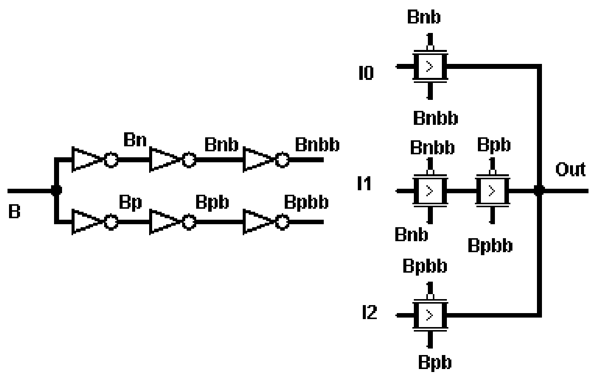

Fig. 6. 3-input MUX with ternary control

- Ternary multipliers use 1-trit multipliers and ternary adders. 1-bit multiplier is implemented with a AND gate.

We use two types of full adders

- The first one is the typical 28T full adder (figure 7).
- The second one is the 14T full adder presented in figure 8.
- A detailed analysis of the performance of these binary adders can be found in [11].

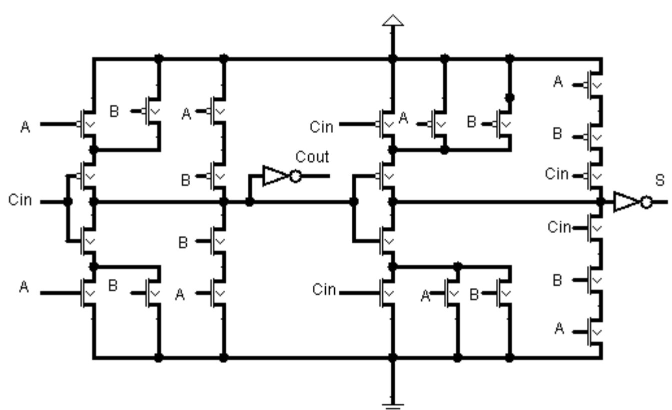

Fig. 7. 28T binary full adder

III. 1-trit adders

A. Unbalanced 1-trit adders (UTFA)

There are two types of unbalanced 1-trit adders:

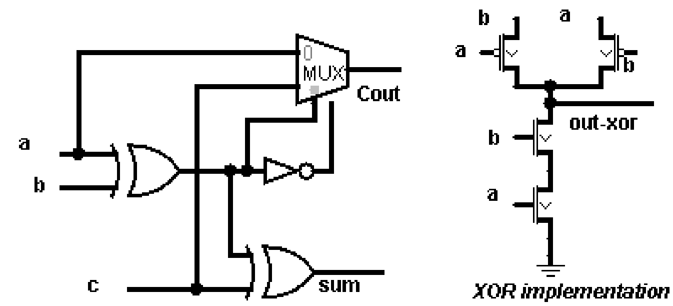

Fig. 8. 14T binary full adder

- The 1-trit adder for building n-trit adders. a and b inputs and S output are ternary. Input and output carries are binary.
- The ternary counter also called compressor for which all values (inputs, output and carries) are ternary.

Table IV presents the combined truth table. The whole table corresponds to the ternary counter. Only the columns C_{in}=0 and C_{in}=1 should be considered for the 1-trit adder.

TABLE IV
Truth table of a ternary full adder with ternary carries

C_{in}=0				C_{in}=1				C_{in}=2			
a	b	S_0	C_{out0}	a	b	S_1	C_{out1}	a	b	S_2	C_{out2}
0	0	0	0	0	0	1	0	0	0	2	0
0	1	1	0	0	1	2	0	0	1	0	1
0	2	2	0	0	2	0	1	0	2	1	1
1	0	1	0	1	0	2	0	1	0	0	1
1	1	2	0	1	1	0	1	1	1	1	1
1	2	0	1	1	2	1	1	1	2	2	1
2	0	2	0	2	0	0	1	2	0	1	1
2	1	0	1	2	1	1	1	2	1	2	1
2	2	1	1	2	2	2	1	2	2	0	2

The Mux approach is based on a different way to consider Table IV:

When C_{in}=0

- When B=0 then Sum=A
- When B=1 then Sum = (A+1) mod(3) quoted as A^1
- When B=2 then Sum = (A+2) 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

When C_{in}=2 (Only for the ternary counter)

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

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

In the two left columns of Table IV, 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). The two different versions named TFA1 and TFA2 are shown in Fig. 9.

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. 9. The V$_{dd}$/2 version uses the inverter implementing A_n 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.

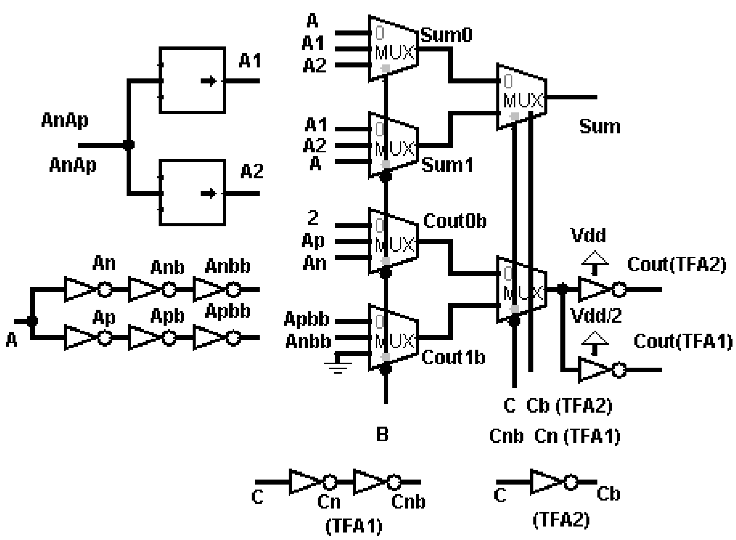

Fig. 9. Unbalanced 1-trit Full Adder (Mux approach)

The 1-trit unbalanced adder is shown in Fig. 9. They have a common part:

- Two 3-input MUXes controlled by B switch A, A^1, A^2 to Sum_0 and Sum_1. Two other 3-input MUXes 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 $\overline{Sum_0}$ or Sum_1 to Sum while another one switches either $\overline{C_{out0}}$ or $\overline{C_{out1}}$ to $\overline{C_{out}}$.

For TFA1, the final C_{out} with 0,1 values is obtained using an inverter with V$_{dd}$/2 power supply. The final inverter for the carry output is introduced to avoid a long chain of MUX3 in the carry propagation of Carry Propagate Adders (CPA) that would degrade the propagation times. The resulting 1-trit full adder 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. TFA2 uses a final binary inverter with V$_{dd}$ power supply. The control of the MUX2s are different for TFA1 and TFA2.

The (3,2) unbalanced ternary counter is shown in Fig. 10. It is not useful to implement n-trit adders. We present it as it is similar to the balanced 1-trit adder.

For unbalanced ternary adders, Fig. 11 presents the Input to Sum/C_{out} delays according to CL for the two versions (0.45 and 0.9V carry swings). The log-log scale shows that the delays roughly increase linearly with CL. Sum delays are similar for the two versions. The C_{out} delays are less sensitive to CL, and less for 0.45V than for 0.9V carry swing. Fig. 12 presents the C_{in} to Sum/C_{out} delays. The C_{out} delays are slower with 0.9 V swing, which is the reason to introduce it for CPAs. Fig. 13 presents the power dissipation according to CL. It increases more than linearly with CL.

B. Balanced 1-trit adders (BTFA)

Table V presents the truth table of a balanced ternary adder. Table VI presents the same table by replacing N by 0, Z by 1 and P by 2 to establish the correspondence between 0, V_{dd}/2 and V_{dd} and 0,1,2. This allow a direct comparison between the unbalanced and balanced implementations.

TABLE V
TRUTH TABLE OF A BALANCED TERNARY ADDER

C_{in}=N				C_{in}=Z				C_{in}=P			
b	a	sN	CtN	a	b	sZ	CZ	a	b	sP	CP
N	N	Z	N	N	N	P	N	N	N	N	Z
N	Z	P	N	N	Z	N	Z	N	Z	Z	Z
N	P	N	Z	N	P	Z	Z	N	P	P	Z
Z	N	P	N	Z	N	N	Z	Z	N	Z	Z
Z	Z	N	Z	Z	Z	Z	Z	Z	Z	P	Z
Z	P	Z	Z	Z	P	P	Z	Z	P	N	P
P	N	N	Z	P	N	Z	Z	P	N	P	Z
P	Z	Z	Z	P	Z	P	Z	P	Z	N	P
P	P	P	Z	P	P	N	P	P	P	Z	P

TABLE VI
TRUTH TABLE OF A BALANCED TERNARY ADDER (N=0; Z=1; P=2)

C_{in}=0				C_{in}=1				C_{in}=2			
b	a	S_0	C_0	a	b	S_1	C_1	a	b	S_2	C_2
0	0	1	0	0	0	2	0	0	0	0	1
0	1	2	0	0	1	0	1	0	1	1	1
0	2	0	1	0	2	1	1	0	2	2	1
1	0	2	0	1	0	0	1	1	0	1	1
1	1	0	1	1	1	1	1	1	1	2	1
1	2	1	1	1	2	2	1	1	2	0	2
2	0	1	1	2	0	1	1	2	0	2	1
2	1	1	1	2	1	2	1	2	1	0	2
2	2	2	1	2	2	0	2	2	2	1	2

The operation of BTFA is described below:
When C_{in}=0

- When B=0 then Sum=A
- When B=1 then Sum = (A+1) mod(3) quoted as A^1
- When B=2 then Sum = (A+2) 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

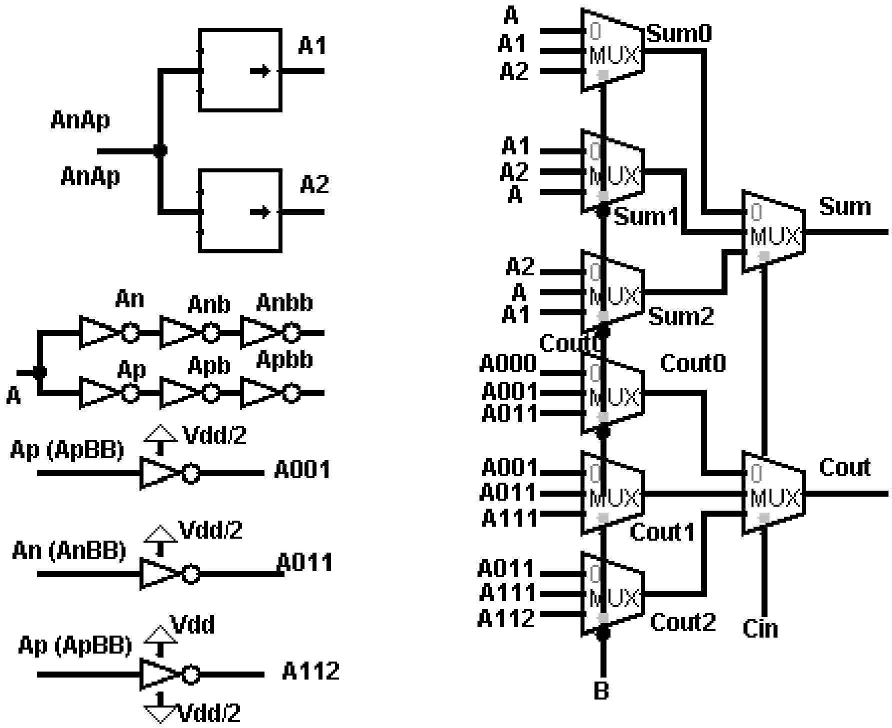

Fig. 10. Unbalanced Ternary (3,2) counter

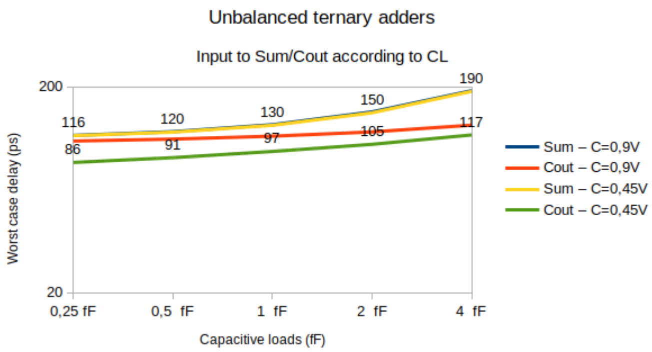

Fig. 11. Input to Sum/C_{out} of unbalanced ternary adders according to CL

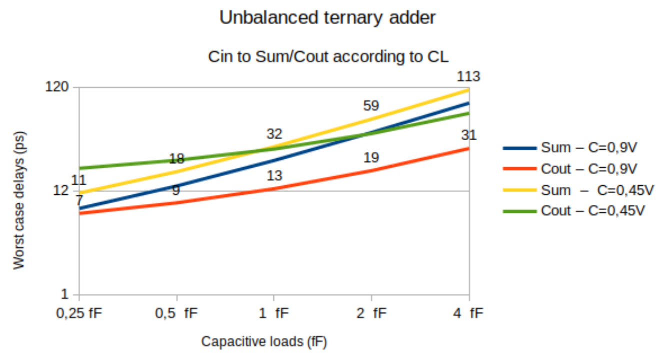

Fig. 12. C_{in} to Sum/C_{out} of unbalanced ternary adders according to CL

When C_{in}=1

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

When C_{in}=2 (Only for the ternary counter)

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

We define two versions of the balanced ternary adder.

- The first one is presented in Fig. 14. It corresponds to the truth table.
- The second one is presented in Fig. 15. As for the unbalanced version, it uses an inverter to avoid a long chain of MUX3 in CPAs. However, as the balanced carries are ternary, this inverter is a ternary one. The used ternary inverter is shown in Fig. 16. A lot of ternary inverters have been proposed, with 4, 5, 6, 8 transistors. However, most of them have conflicting behaviors of transistors for the

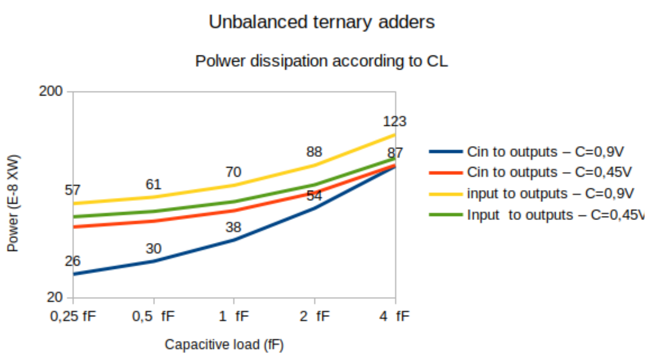

Fig. 13. Power dissipation of unbalanced ternary adders according to CL

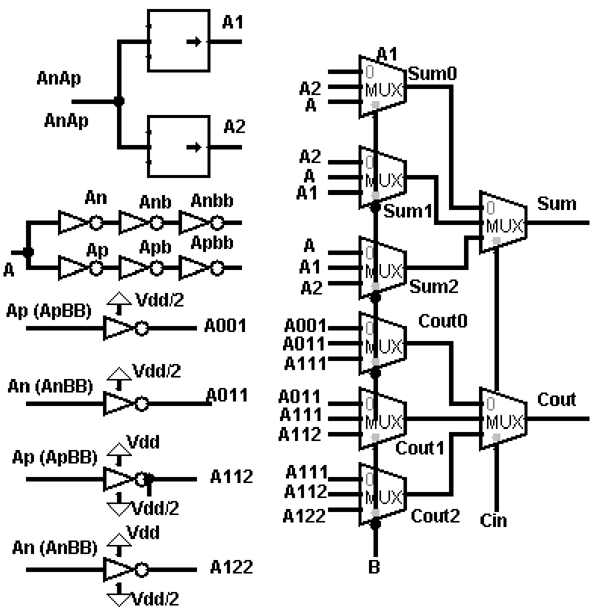

Fig. 14. Balanced ternary adder - Version 1

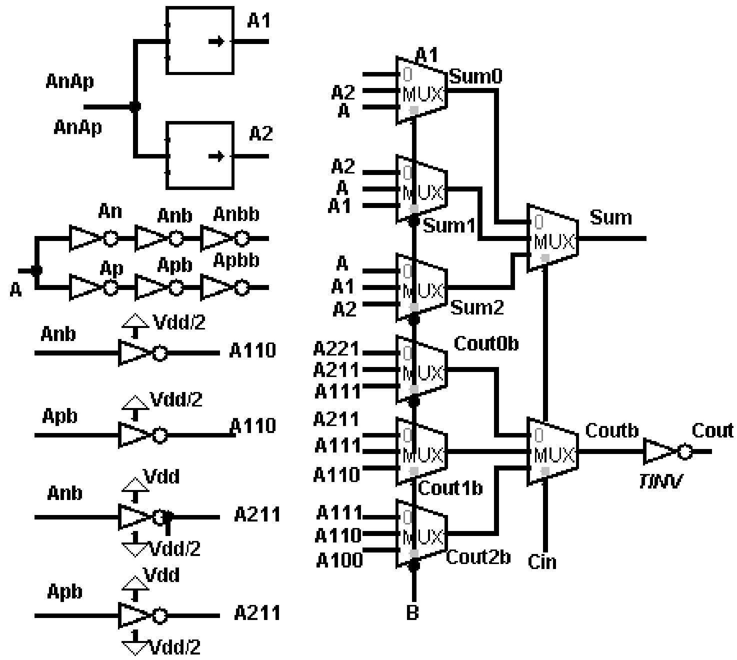

Fig. 15. Balanced ternary adder - Version 2

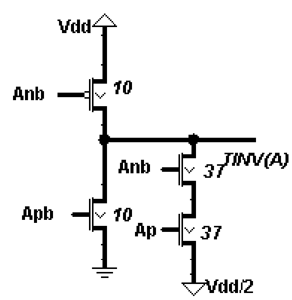

Fig. 16. Ternary inverter

intermediate level, which increase power dissipation. Our implementation has a total of 12 transistors (including 4 inverters to generate a_n, a_{nb}, a_p and apb) but has less power dissipation than most implementations with less transistors.

Fig.17, 18 and 19 respectively present the Input to Sum/C_{out} delays, Carry to Sum/C_{out} delays and power dissipation according to CL. There are few differences between V1 and V2 versions.

C. Comparing the balanced and unbalanced 1-trit adders

Fig. 20 compares these two 1-trit adders:

- ΣDi for the balanced version is roughly x3 ΣDi of the unbalanced version. It comes from using ternary versus binary carries : 8 MUX3 versus 4 MUX3 and 2 MUX2, more circuits to generate the carry values, etc.
- In Carry-Propagate Adders, the critical path is C_{in} to C_{out}. The unbalanced version has smaller C_{in} to C_{out} delays (x3 smaller delay when using the 0.9V carry swing).
- Input to Sum/Output are smaller for the balanced version. However, Input to C_{out} delay is significant only for the first stage of a CPA.

Clearly, the unbalanced version is the best one for implementing CPAs.

Fig. 21 compares balanced and unbalanced 4-trit CPAs with 6-bit CPAs:

- The two versions of the balanced 4-trit CPA have similar performance. The two versions of unbalanced CPAs have also similar performance. However, the unbalanced CPAs has delays x2 smaller than the balanced ones.
- There are few differences between the binary CPAs using 14T and 28T. The 28T version has less chip area than 14T version because it only uses transistor with Di=1.487 nm while the 14T version uses transistors with 2,896 nm for XOR gates and MUX2. When 0.45V V_{dd} is used, power

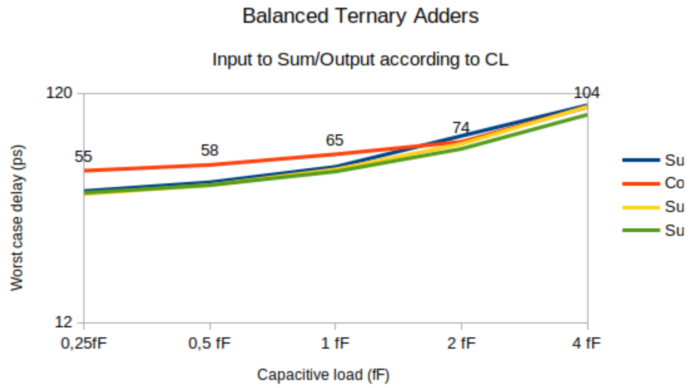

Fig. 17. Input to Sum/C_{out} of balanced ternary adders according to CL

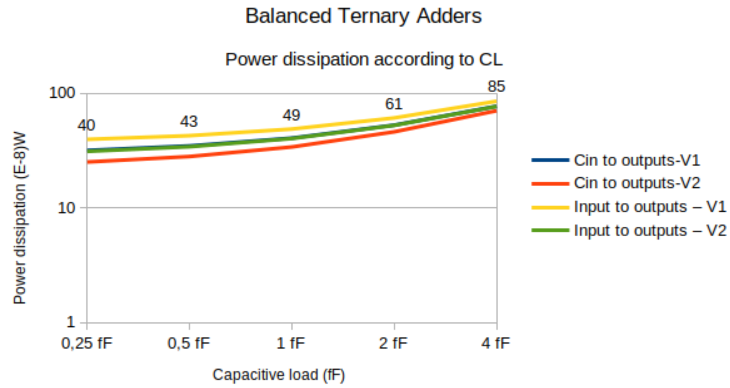

Fig. 19. Power dissipations of balanced ternary adders according to CL

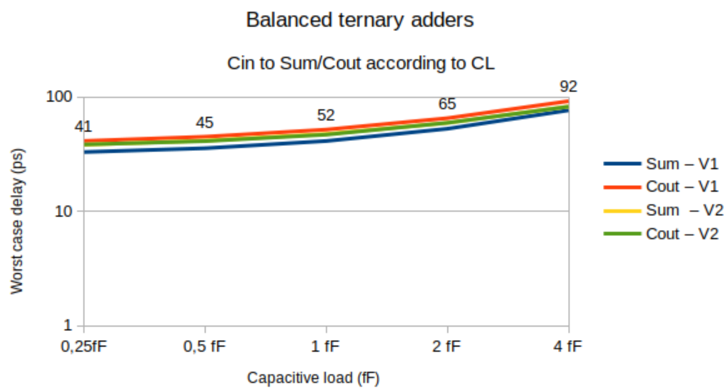

Fig. 18. C_{in} to Sum/C_{out} of balanced ternary adders according to CL

is roughly divided by 4.

- The delays of the 4-trit unbalanced CPAs are smaller than the corresponding ones of the 6-bit CPAs only for the 0.45V binary V_{dd}.
- The power of the same unbalanced CPAs are smaller than the power of the 6-bit CPAs only for the 0.9V binary V_{dd}.
- The ΣDi of the binary CPAs are more than x2 smaller than the unbalanced CPAs and x4.5 smaller than the balanced CPAs.

Clearly, the 6-bit CPA are more efficient than the 4-trit CPAs. The balanced 4-trit CPAs are outperformed by the unbalanced ones.

IV. 1-TRIT MULTIPLIERS

A. Unbalanced 1-trit multipliers

Table VII presents the truth table of the unbalanced 1-trit multiplier. This multiplier generates both a product term and a carry term. For a mux-based implementation, Table VII can be rewritten as

- When Bi=0 then Pi=0 and Ci=0
- When Bi=1 then Pi=Bi and Ci=0
- When Bi=2 then Pi= {0,2,1} and Ci={0,0,1} for Ai={0,1,2}

The X021 and X001 unary operators and a ternary multiplexer are needed to implement the 1-trit multiplier.

TABLE VII
TRUTH TABLE OF A 1-TRIT UNBALANCED MULTIPLIER

Bi	Ai	Pi	Ci
0	0	0	0
0	1	0	0
0	2	0	0
1	0	0	0
1	1	1	0
1	2	2	0
2	0	0	0
2	1	2	0
2	2	1	1

Fig. 22 presents the delay and power of the unbalanced multiplier according to CL.

B. Balanced 1-trit multipliers

Table VIII presents the truth table of a balanced 1-trit multiplier. The left part uses the N(-1), Z(0) and P (+1) notations of the balanced ternary representation. The right part presents the correspondence with the ground level (0), middle level (1) and V_{dd} (2) of the corresponding circuits. It should be noticed that the 1-trit balanced multiplier does not generate a carry output.

TABLE VIII
TRUTH TABLE OF A 1-TRIT MULTIPLIER

Bi	Ai	Prod.		Ai	Bi	Prod.
N	N	P		0	0	2
N	Z	Z		0	1	1
N	P	N		0	2	0
Z	N	Z		1	0	1
Z	Z	Z		1	1	1
Z	P	Z		1	2	1
P	N	N		2	0	0
P	Z	Z		2	1	1
P	P	P		2	2	2

For the mux based implementation, Table VIII can be rewritten as:

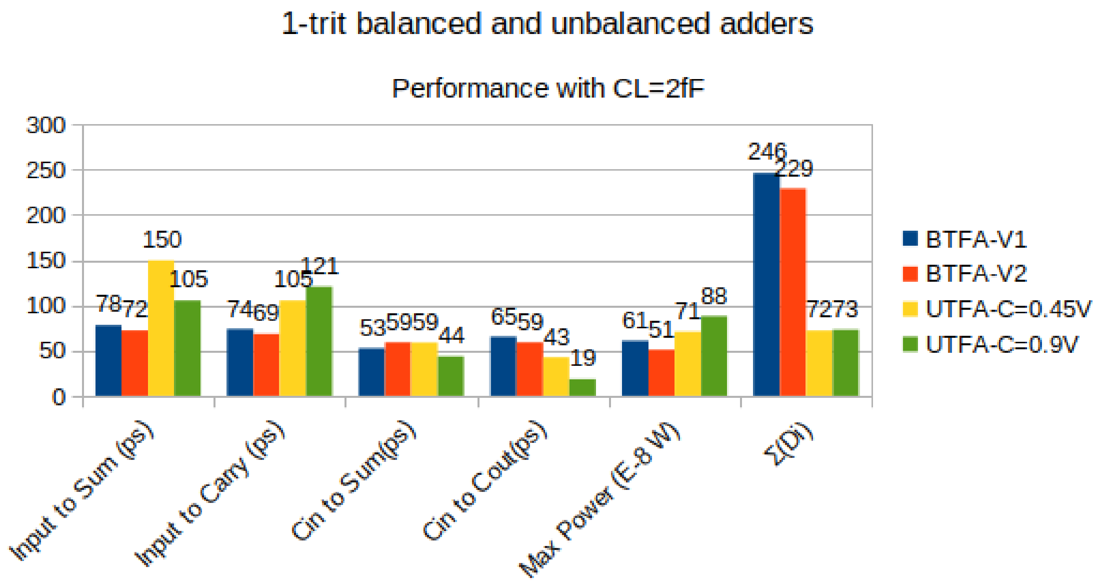

Fig. 20. Comparing 1-trit balanced and unbalanced adder (CL=2fF)

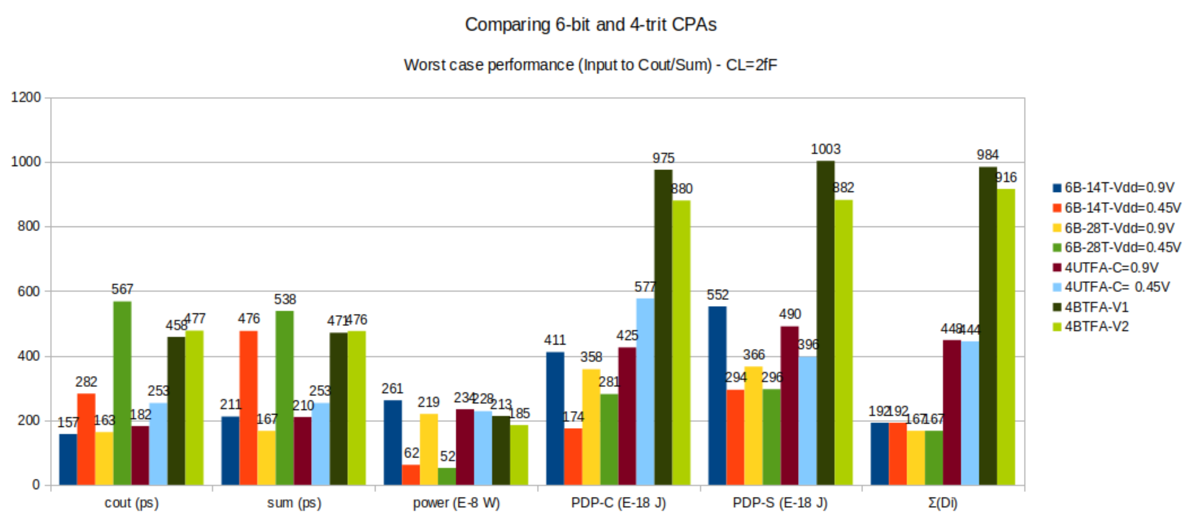

Fig. 21. Comparing 4-trit and 6-bit CPAs (CL=2fF)

- When Bi=0 then Prod={2,1,0} when A={0,1,2}. It is implemented by a ternary inverter.
- When Bi=1 then Prod=1
- When Bi=2 then Prod=Ai

Fig. 23 presents the delay and power of the unbalanced multiplier according to CL.

C. Comparing balanced and unbalanced 1-trit multiplier

Fig. 24 summarizes the comparison between the two different versions

- There is no significant difference for worst-case propagation delays
- Power dissipation is roughly 3x lower for the balanced version, due to the absence of carry generation.
- For the same reason, ΣDi is about x2 smaller for the balanced version.

As the balanced version does not generate a carry: there are n lines in the reduction tree of a n*n trit multiplier, when there are 2n lines in the unbalanced version.

D. Comparing 2*2 trit balanced and unbalanced multipliers with a 3*3 bit multiplier

While it makes sense to compare the two versions of 1-trit multiplier, it is not fair to compare them with the 1-bit multiplier, which is only an AND gate. We compare now a

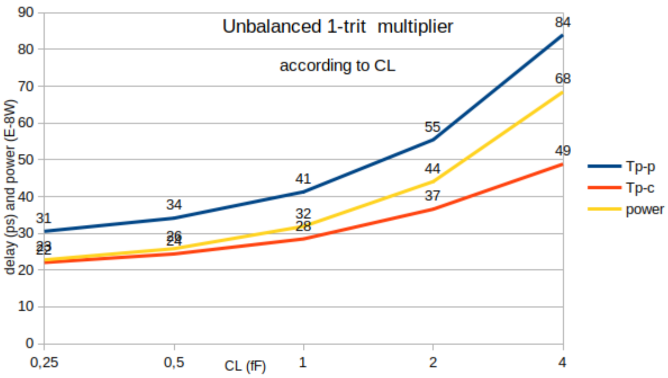

Fig. 22. Unbalanced ternary multiplier performance according to CL

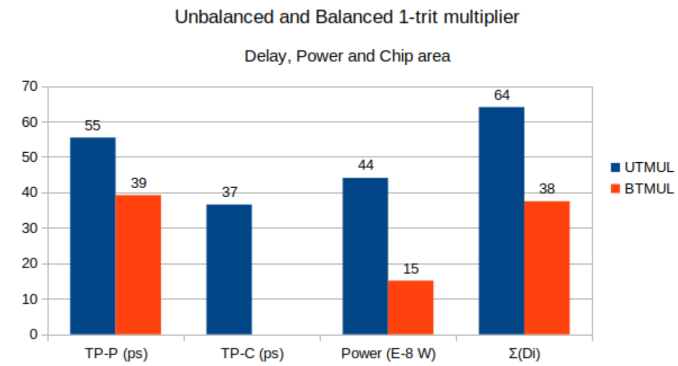

Fig. 24. Unbalanced and Balanced 1-trit multiplier with 2fF capacitive loads

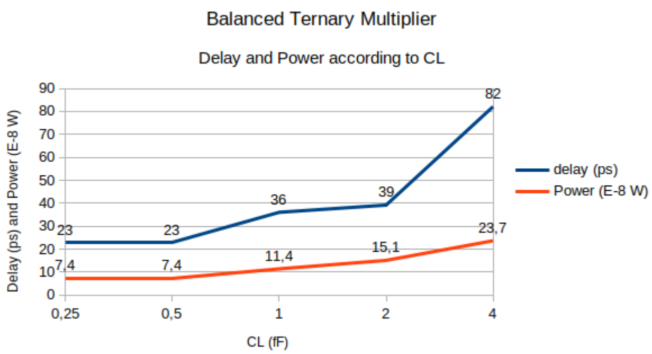

Fig. 23. Balanced ternary multiplier performance according to CL

3*3 bit multiplier, with the two versions of 2*2 trit multipliers (Fig. 25). In this figure, 3 corresponds to a ternary value and 2 to a binary one. The 3*3 bit multiplier computes 6 bits of information versus 6.34 bit for the 2*2 trit multipliers. These multipliers are quite simple. The ternary balanced multipliers only use two balanced ternary half adders.

The 3*3 bit multiplier use 9 and gates (NAND+Inverter), 3 binary half adders and 3 binary full adders. For our simulations, we used 6 binary full adders.

The worst-case delay (WC) of the 3*3 bit multiplier is obtained by multiplying x11*111 with x={0,1,0}. The corresponding outputs are then {010101, 110001, 010101}. m5 outputs are {0,1,0} and m2 outputs are {1,0,1}. For the ternary multipliers, x2*22 delivers the WC delay with x={0,1,2,1,0}. The corresponding outputs are then {0121, 1111, 2101, 1111, 0121}. m3 outputs are then {0,1,2,1,0} and m1 outputs are {2,1,0,1,2}. Fig. 26 summarizes the comparison:

- The two binary multipliers have similar performance.
- The delay of the ternary multipliers are close. However, the unbalanced chip area is close to x2 the balanced one, while the power dissipation of the balanced one is close to x3 the power of the unbalanced one.
- The two ternary multipliers are outperformed by the two binary ones. Delays are x1.5 to x2 greater. Power is x8 to x25 greater. Chip area is equivalent for the balanced version. However, it is x2 for the unbalanced one.

E. Comparing larger multipliers without simulations

Simulating larger ternary and binary multipliers would be more significant, but would need huge simulation times. However, without simulation, it is possible to evaluate the chip area of different multipliers computing the same amount of information. It is also possible to evaluate the worst case propagation delay.

Higher part of Fig. 27 compares 4*4 trit and 6*6 bit multipliers. Lower part of this figure compares 8*8 trit and 12*12 bit multipliers. The different multipliers use (3,2) adders in the Wallace reduction trees and carry-propagate adders for the final sum (this is the only difference with actual combinational multipliers that would use fast adders for the final sum). The unbalanced ternary versions use more chip area than the balanced one: this is due to the carries generated by the multipliers that double the number of lines to reduce. In both case, the ternary versions use more than x1.8 chip area than the binary one for the unbalanced version and x1.3 and x1.7 for the balanced version.

For a 6*6 bit multiplier, Fig. 28 shows how the worst case propagation delay can be evaluated. There is the propagation delay of the 1-digit multiplier. Then the worst case delay corresponds to the vertical longest path in the reduction tree followed by the horizontal left propagation delay for the remaining bits of the carry propagate adder. These delays have a green color on the two parts of Fig. 28. We use the following notation:

- $TpMul$ is the delay of the 1-digit multiplier.
- $TpFAI2Sum$ is the input to Sum delay of the 1-digit Full Adder.
- $TpFAI2Cout$ is the input to C_{out} delay of the 1-digit Full Adder.
- $TpFAC_{in}2C_{out}$ is the C_{in} to C_{out} delay of the 1-digit Full Adder.

The evaluation of WC delays of binary multipliers have been done with 28T binary full adders.

For the 6*6 bit multiplier, the WC delay is: $T_pMul + 3T_pFAI2Sum + T_pFAI2C_{out} + 5T_pFAC_{in}2C_{out}$

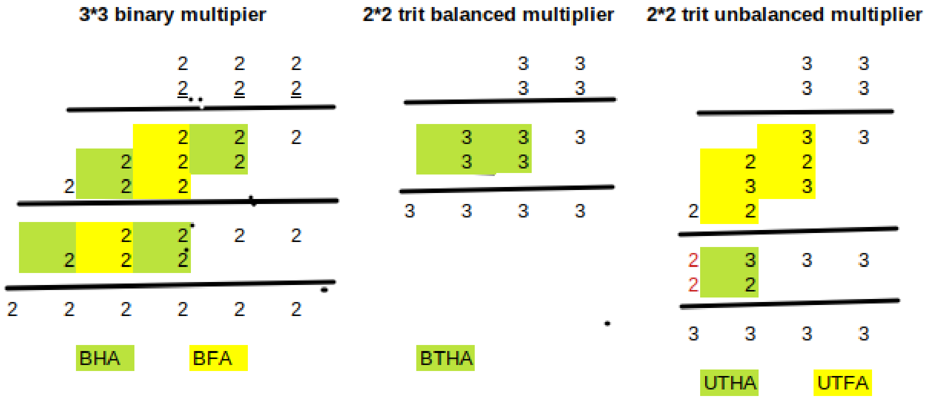

Fig. 25. 3*3 bit and 2*2 trit multipliers

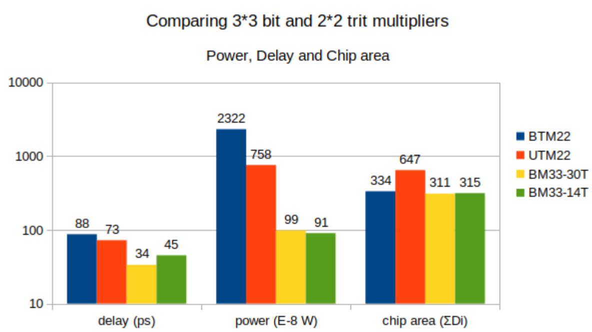

Fig. 26. Comparing 2*2 trit and 3*3 bit multipliers

Higher part of Fig. 29 compares the evaluated WC delays of the two types of 4*4 ternary multipliers and a 6*6 bit multiplier. Lower part of the Fig. compares 8*8 trit multipliers with a 12*12 bit multiplier. WC delays of the two types of ternary multipliers are close. However, they are x1.7 to 2.1 the propagation delays of the binary multipliers computing the same amount of information. These results are not surprising:

- There are x2.25 more binary multipliers than ternary ones. However, the ternary multipliers are x2.7 and x4.6 slower than the binary ones.
- There are less half adders and full adders for the ternary versions, but the ternary half adders and full adders are slower than the corresponding binary ones.

The evalution of WC delays that has been done is not as precise as simulations. However, it provides a significant information.

The delays and chip area ratio between ternary and binary adders and multipliers should not be considered as absolute values. For the different components, delays and chip area depend on the chosen diameter values of the transistors. However, as the ratios are between x1.7 and x2.3, different design diameter sizes would not change the results of the comparisons.

V. Concluding Remarks

Two versions of balanced ternary adders have been compared with two versions of unbalanced ones. The balanced version does not provide advantage as it uses far more chip area. The results are similar when implementing 4-trit CPAs. The two versions are less efficient than the binary ones when comparing 4-trit CPAs and 6-bit CPAs.

1-trit balanced and unbalanced multipliers have been compared. The balanced version has two advantages:

- As it does not generate a carry, it has smaller delays, smaller power dissipation and smaller chip area.
- The balanced version directly operates positive and negative numbers, while the unbalanced one only operate on positive numbers.

However, when comparing 2*2 trit multipliers, the balanced version uses x2 less chip area, but has a small disadvantage in delay and x3 more power dissipation.

While 2*2 trit multipliers and 3*3 bit multipliers have similar chip areas, the ternary multipliers have larger delays and power dissipation from x7 to x25 the power of the binary ones. It comes from the larger complexity of the 1-trit adders compared to a AND gate and the larger complexity of the 1-trit adders compared to the binary ones.

This study shows that the ternary approach does not improve the performance of the binary approach for typical combinational circuits such as adders and multipliers.

References

[1] S. Lin, Y.-B. Kim, and F. Lombardi, “CNTFET-Based Design of Ternary Logic Gates and Arithmetic Circuits", IEEE Trans. Nanotechnology, vol. 10, no. 2, pp. 217-225, Mar. 2011, doi: 10.1109/TNANO.2009.2036845.

[2] B. Srinivasu and K. Sridharan, “A Synthesis Methodology for Ternary Logic Circuits in Emerging Device Technologies", IEEE Trans. Circuits Syst. I, vol. 64, no. 8, pp. 2146-2159, Aug. 2017, doi: 10.1109/TCSI.2017.2686446.

[3] S. Tabrizchi, A. Panahi, F. Sharifi, K. Navi, and N. Bagherzadeh, “Method for designing ternary adder cells based on CNFETs”, IET Circuits, Devices & Systems, vol. 11, no. 5, pp. 465-470, Sep. 2017, doi: 10.1049/iet-cds.2016.0443.

	Number			ΣDi/unit			ΣDi (total		
	TB4*4	TU4*4	B6*6				TB4*4	TU4*4	B6*6
MUL	16	16	36	38	64	14	608	1024	504
HA	4	7	11	92	78	18	368	546	198
FA	8	17	23	178	104	47	1424	1768	1081
							2400	**3338**	**1783**
							1,3	1,9	1

	Number			ΣDi/unit			ΣDi (total		
	TB8*8	TU8*8	B12*12				TB4*4	TU4*4	B6*6
MUL	64	64	144	38	64	14	2432	4096	2016
HA	17	20	31	92	78	18	1564	1560	198
FA	47	73	107	178	104	47	8366	7592	5029
							12362	**13248**	**7243**
							1,7	1,8	1

Fig. 27. $\Sigma(Di)$ of 4*4 trit and 6*6 bit multipliers and $\Sigma(Di)$ of 8*8 trit and 12*12 bit multipliers

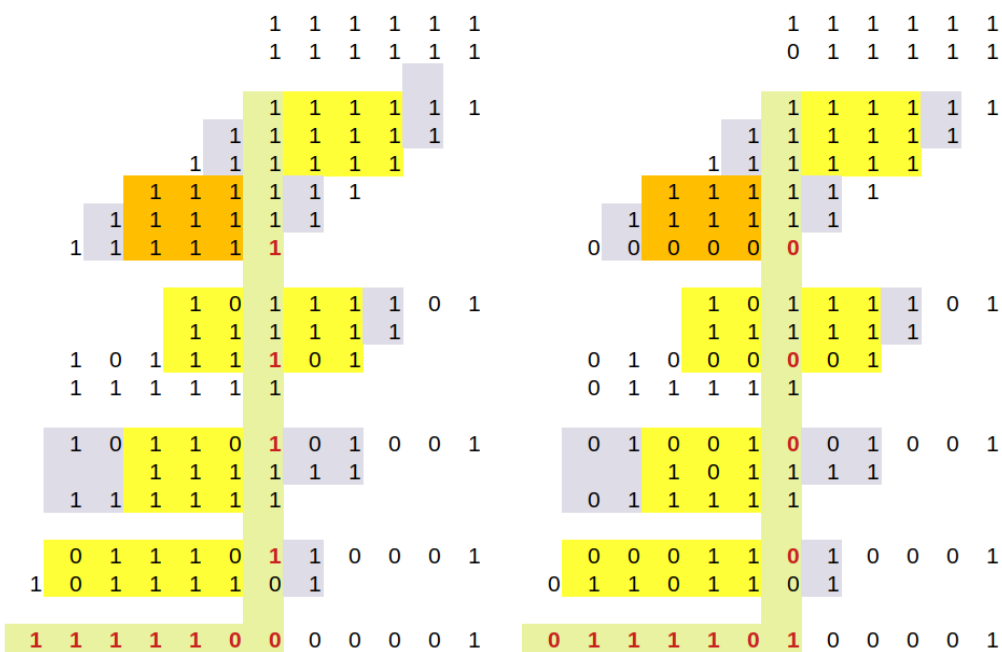

Fig. 28. Worst case delay through multipliers and Wallace reduction tree

[4] C. Vudadha, A. Surya, S. Agrawal, and M. B. Srinivas, “Synthesis of Ternary Logic Circuits Using 2:1 Multiplexers", IEEE Trans. Circuits Syst. I, vol. 65, no. 12, pp. 4313-4325, Dec. 2018, doi: 10.1109/TCSI.2018“.2838258.

[5] E. Shahrom and S. A. Hosseini, “A new low power multiplexer based ternary multiplier using CNTFETs", AEU - International Journal of Electronics and Communications, vol. 93, pp. 191-207, Sep. 2018, doi: 10.1016/j.aeue.2018.06.011.

[6] T. Sharma and L. Kumre, “CNTFET-Based Design of Ternary Arithmetic Modules", Circuits Syst Signal Process, vol. 38, no. 10, pp. 4640-4666, Oct. 2019, doi: 10.1007/s00034-019-01070-9.

[7] S. Kim, S. -Y. Lee, S. Park, K. R. Kim and S. Kang, “A Logic Synthesis Methodology for Low-Power Ternary Logic Circuits," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 9, pp. 3138-3151, Sept. 2020, doi: 10.1109/TCSI.2020.2990748.

[8] I. Mahmoudi Salehabad, K. Navi, and M. Hosseinzadeh, “Two novel inverter-based ternary full adder cells using CNFETs for energy-efficient applications", International Journal of Electronics, vol. 107, no. 1, pp. 82-98, Jan. 2020, doi: 10.1080/00207217.2019.1636306.

[9] S. A. Hosseini and S. Etezadi, “A Novel Low-Complexity and Energy-Efficient Ternary Full Adder in Nanoelectronics", Circuits Syst Signal Process, vol. 40, no. 3, pp. 1314-1332, Mar. 2021, doi: 10.1007/s00034-020-01519-2.

[10] R.A. Jaber, A.M. Haidar,A. Kassem, F. Zahoor “Ternary Full Adder Designs Employing Unary Operators and Ternary Multiplexers". Micromachines 2023, 14, 1064. https://doi.org/10.3390/mi14051064

[11] D. Etiemble, “Post algebras and ternary adders". Journal of Electrical Systems and Inf Technol 10, 20 (2023). https://doi.org/10.1186/s43067-023-00088-z

[12] M. Toulabinejad, M. Taheri, K. Navi, N. Bagherzadeh, “Toward efficient implementation of basic balanced ternary arithmetic operations in CNFET technology", Microelectronics Journal, Volume 90, 2019, Pages 267-277,

	Number			Delay/Unit			Delay		
	TB44	TU44	B66	TB	TU	Bin	TB44	TU44	B66
Multiplier	1	1	1	36	55	15	36	55	15
FAI2Sum	1	2	3	72	105	27	72	210	81
FAI2Cout	1	1	1	105	19	47	105	19	47
FACin2Cout	4	3	5	59	44	21	236	132	105
FACIN2Sum		1			121	24		121	0
							449	416	248
							1,8	1,7	1

	Number			Delay/Unit			Delay		
	TB88	TU88	B1212	TB	TU	Bin	TB44	TU44	B66
Multiplier	1	1	1	36	55	15	36	55	15
FAI2Sum	3	4	4	72	105	27	216	420	108
FAI2Cout	1	1	1	105	19	47	105	19	47
FACin2Cout	8	7	11	59	44	21	472	308	231
FACIN2Sum		1	1		121	24		121	24
							829	802	401
							2,1	2,0	1

Fig. 29. WC delays of 4*4 trit and 6*6 bit multipliers and WCDelay of 8*8 trit and 12*12 bit multipliers

ISSN 0026-2692, https://doi.org/10.1016/j.mejo.2019.05.010.

[13] S. Kim, T. Lim and S. Kang, “An optimal gate design for the synthesis of ternary logic circuits," 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju, Korea (South), 2018, pp. 476-481, doi: 10.1109/ASPDAC.2018.8297369.

[14] J. Yoon, S. Baek, S. Kim and S. Kang, “Optimizing Ternary Multiplier Design With Fast Ternary Adder," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 70, no. 2, pp. 766-770, Feb. 2023, doi: 10.1109/TCSII.2022.3210282.

[15] E. Shahrom, S.A. Hosseini, “A new low power multiplexer based ternary multiplier using CNTFETs", AEU - International Journal of Electronics and Communications, Volume 93, 2018, Pages 191-207, https://doi.org/10.1016/j.aeue.2018.06.011.

[16] S. Tabrizchi, A. Panahi, F. Sharifi, H. Mahmoodi, A.H. Badawy, “Energy-Efficient Ternary Multipliers Using CNT Transistors". Electronics 2020, 9, 643. https://doi.org/10.3390/electronics9040643

[17] R. A. Jaber, A. M. Haidar and A. Kassem, “CNTFET-Based Design of Ternary Multiplier using Only Multiplexers," 2020 32nd International Conference on Microelectronics (ICM), Aqaba, Jordan, 2020, pp. 1-4, doi: 10.1109/ICM50269.2020.9331806.

[18] R. A. Jaber, H. Bazzi, A. Haidar, B. Owaidat and A. Kassem, “1-trit Ternary Multiplier and Adder Designs Using Ternary Multiplexers and Unary Operators," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 2021, pp. 292-297, doi: 10.1109/3ICT53449.2021.9581366.

[19] E. Abbasian, S. A. Sofimowloodi, “A High-Performance and Energy-Efficient Ternary Multiplier Using CNTFETs". Arab J Sci Eng (2023). https://doi.org/10.1007/s13369-023-07618-x

[20] D. Etiemble and R.A. Jaber, “Design of (3,2) and (4,2) CNTFET Ternary Counters for Multipliers. Asian Journal of Research in Computer Science". 16, 3 (Jul. 2023), 103-118. DOI:https://doi.org/10.9734/ajrcos/2023/v16i3349.

[21] J. Deng and H. -. P. Wong, “A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application-Part II: Full Device Model and Circuit Performance Benchmarking", in IEEE Transactions on Electron Devices, vol. 54, no. 12, pp. 3195-3205, Dec. 2007. https://doi.org/10.1109/TED.2007.909043.

[22] W. J. Townsend, E. E. Swartzlander Jr. and J. A. Abraham, “A comparison of Dadda and Wallace multiplier delays", Proc. SPIE 5205, Advanced Signal Processing Algorithms, Architectures, and Implementations XIII, (24 December 2003); https://doi.org/10.1117/12.507012