International Journal of LNCT, Vol 2(5)
ISSN (Online): 2456-9895

Implementation aspects of Radix-N adder with analysis of Radix-4

Deepak Jain[1] and M. Zahid Alam[2]
M. Tech Scholar, Department of Electronics and Communication Engineering, Lakshmi Narain College of Technology, Bhopal, (M.P), India[1]
Associate Professor, Department of Electronics and Communication Engineering, Lakshmi Narain College of Technology, Bhopal, (M.P), India[2]

Abstract
Increasing of calculation in various operations arithmetic circuits needs to be increased capability. With various Improvements in the VLSI technology submicron dimension the various other algorithms and implementation concepts should be improved. Traditionally Radix-2 binary logic circuits used which have 0,1 two symbol to represents then Radix–4 quaternary logic which have 0,1 ,2,3 symbols representation has been used. Then we can further have increased the concept to the Radix- N logic implementation which can have N symbols to represent. Various advantages of power utilization and improved switching delays will be there.

Keywords
Higher radix logic, Full adder, Multivalued logic, Quaternary logic, Standard CMOS Technology.

1.Introduction
In binary the number system the two voltage level indicates the one binary bit i.e. logic '1' and logic '0'. The voltage level 0V represent the binary logic '0' and the voltage level of 5V represent binary logic '1'. While the ternary logic number system uses three voltage levels of 0V, Vdd/3 and Vdd V, whereas the quaternary number system represents the four voltage levels of 0V, Vdd/2, 2Vdd/3, Vdd V. This voltage level may differ for both ternary and quaternary logic system. The encoder and decoder circuits to convert binary logic to multivalued logic and multivalued logic to binary logic. The number system logic in ternaruy uses logic 0,1 and 2 while the quaternary number system uses logic 0,1,2 and 3.

2.MVL aritmatic addition
Addition rules are the same as in the decimal system. The sum or product of two digits may only produce one or two digit numbers. In the latter case, if necessary, the first digit is carried over to the next operation (on the left.) For example, in base 7, 36 + 144 = 213. Indeed, from right to left, 6 + 4 = 13. Then 3 + 4 + 1 = 11, and finally 1 + 1 = 2.

As everyone knows, 2 + 2 = 4. This is true in all base systems. That is, except bases 2, 3, and 4. In base 4, we have 2 + 2 = 10. In base 3, 2 + 2 = 11. However, recollect that (4)10 = (10)4 = (11)3, and everything falls into its right place again. Numbers equal in one base are equal in any other base. Conversion between bases does not violate arithmetic identities. In base 2, 2 + 2 = 4 appears as 10 + 10 = 100 - looking differently but having exactly the same meaning.

The same, of course, is true of 2 × 2 = 4 which is true in all bases starting with 5. In bases 4,3, and 2 it appears as
2 × 2 = 10
2 × 2 = 11
10 × 10 = 100,
respectively.

Table 1 Following Tables Shows the Arithmetic Addition of Multi Value Logic Number Systems.

1. Ternary addition

+ Radix Number	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

2. Quaternary Addition

+ Radix Number	0	1	2	3
0	0	1	2	3
1	1	2	3	10
2	2	3	10	11
3	3	10	11	12

3. Senery Addition

A+/B	**0**	**1**	**2**	**3**	**4**	**5**
0	0	1	2	3	4	5
1	1	2	3	4	5	10
2	2	3	4	5	10	11
3	3	4	5	10	11	12
4	4	5	10	11	12	13
5	5	10	11	12	13	14

4. Duodecimal Addition

A+/B	0	1	2	3	4	5	6	7	8	9	A	B
0	0	1	2	3	4	5	6	7	8	9	A	B
1	1	2	3	4	5	6	7	8	9	A	B	10
2	2	3	4	5	6	7	8	9	A	B	10	11
3	3	4	5	6	7	8	9	A	B	10	11	12
4	4	5	6	7	8	9	A	B	10	11	12	13
5	5	6	7	8	9	A	B	10	11	12	13	14
6	6	7	8	9	A	B	10	11	12	13	14	15
7	7	8	9	A	B	10	11	12	13	14	15	16
8	8	9	A	B	10	11	12	13	14	15	16	17
9	9	A	B	10	11	12	13	14	15	16	17	18
A	A	B	10	11	12	13	14	15	16	17	18	19
B	B	10	11	12	13	14	15	16	17	18	19	1A

5. Quodradecimal Addition

A+/B	0	1	2	3	4	5	6	7	8	9	A	B	C	D
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D
1	1	2	3	4	5	6	7	8	9	A	B	C	D	10
2	2	3	4	5	6	7	8	9	A	B	C	D	10	11
3	3	4	5	6	7	8	9	A	B	C	D	10	11	12
4	4	5	6	7	8	9	A	B	C	D	10	11	12	13
5	5	6	7	8	9	A	B	C	D	10	11	12	13	14
6	6	7	8	9	A	B	C	D	10	11	12	13	14	15
7	7	8	9	A	B	C	D	10	11	12	13	14	15	16
8	8	9	A	B	C	D	10	11	12	13	14	15	16	17
9	9	A	B	C	D	10	11	12	13	14	15	16	17	18
A	A	B	C	D	10	11	12	13	14	15	16	17	18	19
B	B	C	D	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	10	11	12	13	14	15	16	17	18	19	1A	1B	1C

3.Methodology

Figure 1 shows the schematic design for two stage four-bit ripple adder for design of 8-bit carry select/save adder. The stage 1 is design using four-bit ripple carry adder. In stage 2 a two blocks of 4-bit ripple carry adder is design with assuming that previous carry is ‘0’ or ‘1’. In this stage, first block is use to add four most significant bit with assuming that the previous carry is ‘0’ and the second block is use to add four most significant bit with assuming that the previous carry is ‘1’.

This design uses 12 full adders and 5 two input and one output multiplexers. The carry out bit of first stage ripple carry adder is connected to the select/saveion input of 2X1 multiplexer. If this carry is ‘1’ then it select/save the output of second stage block of ripple carry adder with previous carry ‘1’ and if this carry is ‘0’ then it select/save the output of second stage block of ripple carry adder with previous carry ‘0’ [2,3,4,5].

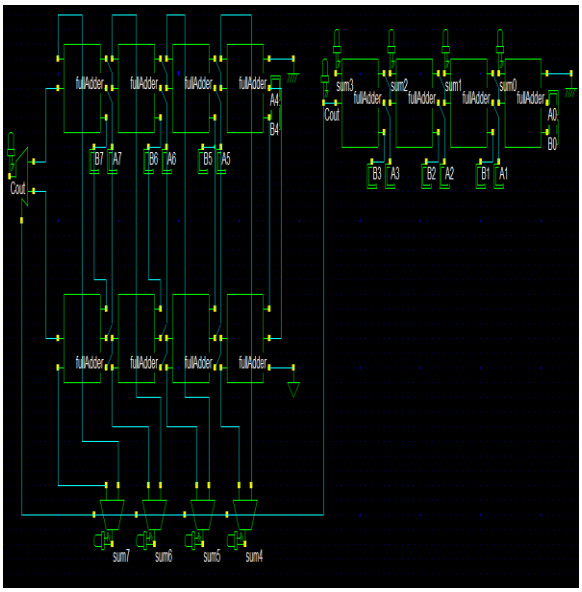

Figure 1 Eight Bit Carry Select/save adde

Figure 2 shows the timing simulation of 8-bit carry select/save addition. The four least significant bits are added by ripple carry addition. And the carry out of this addition will select/save the ripple carry addition of next stage. For example, let's take the addition of two eight bit numbers $(11101111 \text{ and } 10111101)_2$.

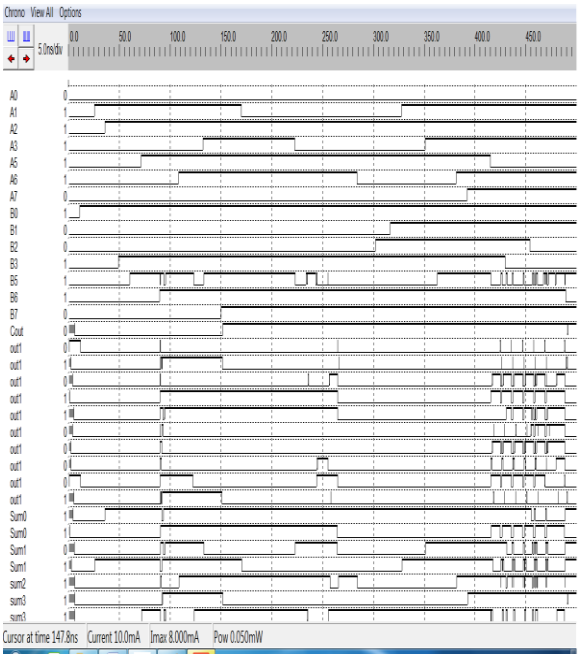

Figure 2 Timing analysis of propose eight bit carry select/save adder The addition of four LSB numbers 1111and 1101 gives the four LSB sum of 1100 along with carry bit '1'. The ripple carry addition of four most significant (MSB) number gives MSB sum of 1001 and 1010 for previous carry assuming as '0' and '1' respectively. The result of MSB is decided by the carry bit of first stage addition. Thus the final result of this addition is $(110101100)_2$.

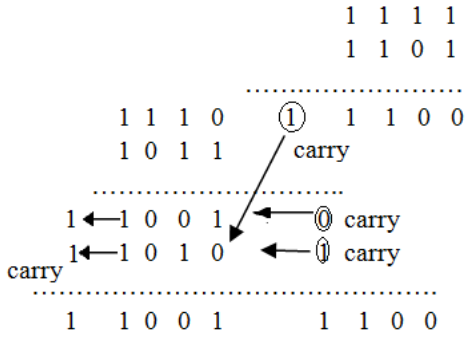

Table 2 Truth table for layout of quaternary Full adder logic circuit

Binary equivalent					QA input	QB input	QC input	Sum	Carry
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	1	0
0	0	0	1	0	0	0	2	2	0
0	0	0	1	1	0	0	3	3	0
0	0	1	0	0	0	1	0	1	0
0	0	1	0	1	0	1	1	2	0
0	0	1	1	0	0	1	2	3	0
0	0	1	1	1	0	1	3	0	1
0	1	0	0	0	0	2	0	2	0
0	1	0	0	1	0	2	1	3	0
0	1	0	1	0	0	2	2	0	1
0	1	0	1	1	0	2	3	1	1
0	1	1	0	0	0	3	0	3	0
0	1	1	0	1	0	3	1	0	1
0	1	1	1	0	0	3	2	1	1
0	1	1	1	1	0	3	3	2	1
1	0	0	0	0	1	0	0	1	0
1	0	0	0	1	1	0	1	2	0
1	0	0	1	0	1	0	2	3	0
1	0	0	1	1	1	0	3	0	1
1	0	1	0	0	1	1	0	2	0
1	0	1	0	1	1	1	1	3	0
1	0	1	1	0	1	1	2	0	1
1	0	1	1	1	1	1	3	1	1

Binary equivalent					QA input	QB input	QC input	Sum	Carry
1	1	0	0	0	1	2	0	3	0
1	1	0	0	1	1	2	1	0	1
1	1	0	1	0	1	2	2	1	1
1	1	0	1	1	1	2	3	2	1
1	1	1	0	0	1	3	0	0	1
1	1	1	0	1	1	3	1	1	1
1	1	1	1	0	1	3	2	2	1
1	1	1	1	1	1	3	3	3	1
0	1	1	1	1	0	3	3	2	1

Addition rules are the same as in the decimal system. The sum of two digits or its product of might produce one or two binary digit numbers. In the latter case, if necessary, the first digit is carried over to the next operation (on the left.) For example, in base 7, 36 + 144 = 213. Indeed, from right to left, 6 + 4 = 13. Then 3 + 4 + 1 = 11, and finally 1 + 1 = 2. As everyone knows, 2 + 2 = 4. This is true in all base systems. That is, except bases 2, 3, and 4. In base 4, we have 2 + 2 = 10. In base 3, 2 + 2 = 11. However, recollect that (4)10 = (10)4 = (11)3, and everything falls into its right place again. Numbers equal in one base are equal in any other base. Conversion between bases does not violate arithmetic identities. For the number system having radix 2 have the addition of decimal number 2 with 2is 4 shows as binary 10 + 10 = binary 100 appears another way but having exactly the same meaning.

4.Radix – N adder implementation

The reference [1] shows the implementation of Radix-4 adder with the CSA (carry save adder) which then been implemented in 50nm and 130nm submicron technology and compared with the similar technology implementation with Radix-2 adder. According to reference [2] the 130nm implementation is compared in the reference [3] paper. As discussed in the section II and the III the implementation in the higher radix digital logic will exploit the use of various voltage level. As the in Radix – N the value of N increases the voltage of 0 to 5V will be divided in the more number of levels that like in Radix- 4 voltage levels are divide into 0V, 1.6V, 3.4V and 5V. If we further increase the value of N the 0 to 5V to be divided into 5,6,7,8 and so on. So our design will become more prone to false level triggering of noise margin drastically decrease. And the further behavior of design become more unpredictable in outcome.

To resolve the level, constrain on the design the various algorithm and the logic family which provide more voltage range in the VLSI fabrication should be used. For the level Radix -8 the bipolar power supply can be used so that more levels will available but still this comes with disadvantage s and limitation of using the bipolar power in the VLSI fabrication.

5.Analysis and comparison

Table 3 Analysis table

Parameters	Technology	Power Dissipation	Output load	Switching Delays	Nos of T	Channel Length	Channel Width
Reference [2]	130nm	140 µW	10 pF	1.5 to 4.4ns	220	0.13 µm	0.26 µm
Reference [3]	130nm	111 µW	1.21fF	0.001 to 0.003 ns	282	0.13 µm	0.24 µm

Table 4 Parametric Analysis

Module Design	Technology	Power Dissipation	Output load	Switching Delays	Nos of T	Channel Length	Channel Width
Binary CSA adder	50 nm	30.42 µW	1.21 fF	0.002 to 0.009 ns	318	0.05 µm	0.1 µm
Radix-4 Adder	130nm	111 µW	1.21 fF	0.001 to 0.003 ns	282	0.13 µm	0.24 µm
Radix-4 Adder	50 nm	76.187 µW	1.21 fF	0.002 to 0.009 ns	282	0.05 µm	0.1 µm

6.Conclusion

This paper thus sums up the comparison of the Radix -4 full adder comparison discussion with the radix -2 full adder with results of power dissipation and switching delays. The discussion shows that the radix –N adder implementation is not possible with the existing methodology of radix -4 implementation as the extraction of voltage levels assigned to the symbols are less and the prone to the false level switching and undetermined operations.

References

[1] Kornerup P. Reviewing high-radix signed-digit adders. IEEE Transactions on Computers. 2015; 64(5):1502-5.

[2] Brito D, Rabuske TG, Fernandes JR, Flores P, Monteiro J. Quaternary logic lookup table in standard CMOS. IEEE Transactions on very large scale integration (VLSI) systems. 2015; 23(2):306-16.

[3] Zlatanovici R, Kao S, Nikolic B. Energy–delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example. IEEE Journal of Solid-State Circuits. 2009; 44(2):569-83.

[4] Romero ME, Martins EM, dos Santos RR, Gonzalez ME. Universal set of CMOS gates for the synthesis of multiple valued logic digital circuits. IEEE Transactions on Circuits and Systems I: Regular Papers. 2014; 61(3):736-49.

[5] Sarica F, Morgül A. Basic circuits for multi-valued sequential logic. Analog Integrated Circuits and Signal Processing. 2013; 74(1):91-6.

[6] Andréasson J, Pischel U. Smart molecules at work-mimicking advanced logic operations. Chemical Society Reviews. 2010 ;39(1):174-88.

[7] Chattopadhyay T, Sarkar T. Logical design of quaternary signed digit conversion circuit and its effectuation using operational amplifier. Bonfring International Journal of Power Systems and Integrated Circuits. 2012; 2(4):7.