ISSN:0975-9646

FPGA Implementation of Fast Arithmetic Unit Based on QSD

Shrikesh A. Dakhane, A. M. Shah

Department of Electronics & Telecommunication,
Government College of Engineering, Amravati, India

***Abstract*—: Design of binary logic circuits is easily possible when the interconnection is less. As the number of inputs increases the interconnections is tedious work. The complexity increases, as chip area is reduced. To overcome the problem multi-valued logic is possible solution. The quaternary signed digit provides us to increase the logic levels. This number system based logic circuits gives less delay as compared to the binary circuits. In this paper the arithmetic unit is designed which gives better results than the binary circuits. The quaternary signed digit based addition prevents rippling of carry. This removes the additional carry handling circuit, which enables to build fast adders which can be used in other processors to give fast results. These adders give the constant delay invariant to the number of inputs. Using this adder, multipliers are designed which gives faster results and the design are less complex. The circuits are designed and verified using Xilinx 13.2 software. The simulation results are tested using Modelsim.**

***Keywords*— FPGA field programmable gate array**

I. Introduction

For any processors the addition is the basic arithmetic operation on which the speed depends. For increasing the speed of such arithmetic operation many researchers move towards the QSD number systems. The QSD number systems provide the carry free addition and borrow free subtraction. The multiplier is designed using these adders. The partial product generator and single bit multiplier is the key elements of the design. These units combine gives the fast results. The behavioural model of design of fast adder and multiplier is done using VHDL. The functional verification of the design is done using Xilinx project navigator and ModelSim simulator on Spartan 3E.

II. Quaternary Number System

Quaternary is the base 4 redundant number system. The degree of redundancy usually increases with the increase of the radix . The signed digit number system allows us to implement parallel arithmetic by using redundancy. QSD numbers are the sign digit numbers with the digit set as :{ $\bar{3}$ $\bar{2}$ $\bar{1}$ 0 1 2 3} where $\bar{3}$ $\bar{2}$ and $\bar{1}$ represents -3, -2 -1 respectively.

$$D = \sum_{i=0}^{n-1} X_i 4^i$$

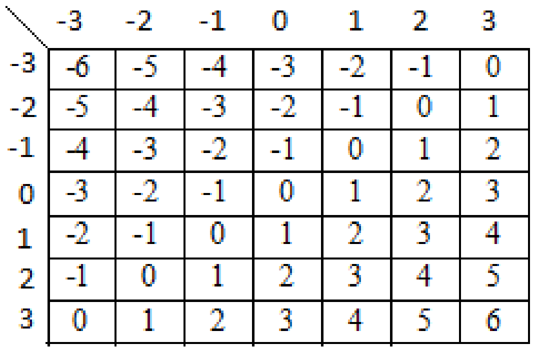
QSD numbers are represented using 3-bit 2’s complement notation. Each number can be represented

III. Algorithm For QSD Addition

The decimal numbers in the range of -3 to +3 are represented by one digit QSD number. As the decimal number exceeds from this range, more than one digit of QSD number is required. The two QSD number addition result lie in the range of +6 and -6 [1]. This result can be represents in two QSD digits. In the two digits QSD result the LSB digit represents the sum bit and the MSB digit represents the carry bit. There are two steps involved in the carry-free addition.

Step 1: Generate an intermediate carry and sum from the addend and augends.
Step 2: Combine the intermediate sum of the current digit with the carry of the lower significant digit.

Table I.
Possible Combination Addition Of Two Digits



	-3	-2	-1	0	1	2	3
-3	-6	-5	-4	-3	-2	-1	0
-2	-5	-4	-3	-2	-1	0	1
-1	-4	-3	-2	-1	0	1	2
0	-3	-2	-1	0	1	2	3
1	-2	-1	0	1	2	3	4
2	-1	0	1	2	3	4	5
3	0	1	2	3	4	5	6

.

To prevent carry from further rippling, we define two rules.

Rule 1: The magnitude of the intermediate sum must be less than or equal to 2 which means it should be in range of +2 to -2.

Rule 2: The magnitude of the carry must be less than or equal to 1 that means it should be in range of +1 to -1.

Consequently, the magnitude of the second step output cannot be greater than 3 which can be represented by a single-digit QSD number, hence no further carry is required. By exploiting the redundancy feature of QSD numbers we choose such QSD represented number which satisfies the above mentioned two rules.

TABLE II
INTERMEDIATE SUM AND CARRY REPRESENTATION

Sum	*QSD represented number*	*QSD coded number*
-6	$\bar{2}2, \bar{1}\bar{2}$	$\bar{1}\bar{2}$
-5	$\bar{2}3, \bar{1}\bar{1}$	$\bar{1}\bar{1}$
-4	$\bar{1}0$	$\bar{1}0$
-3	$\bar{1}1, 0\bar{3}$	$\bar{1}1$
-2	$\bar{1}2, 0\bar{2}$	$0\bar{2}$
-1	$\bar{1}3, 0\bar{1}$	$0\bar{1}$
0	00	00
1	$01, 1\bar{3}$	01
2	$02, 1\bar{2}$	02
3	$03, 1\bar{1}$	$1\bar{1}$
4	10	10
5	$11, 2\bar{3}$	11
6	$12, 2\bar{2}$	12

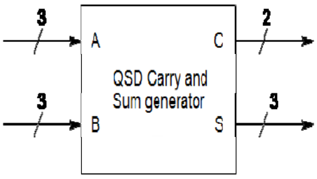
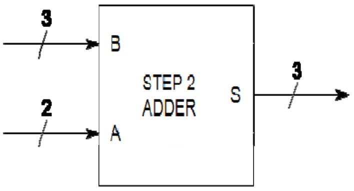


Fig 1.Intermediate sum and carry generator



Fg 2. Step 2 adder

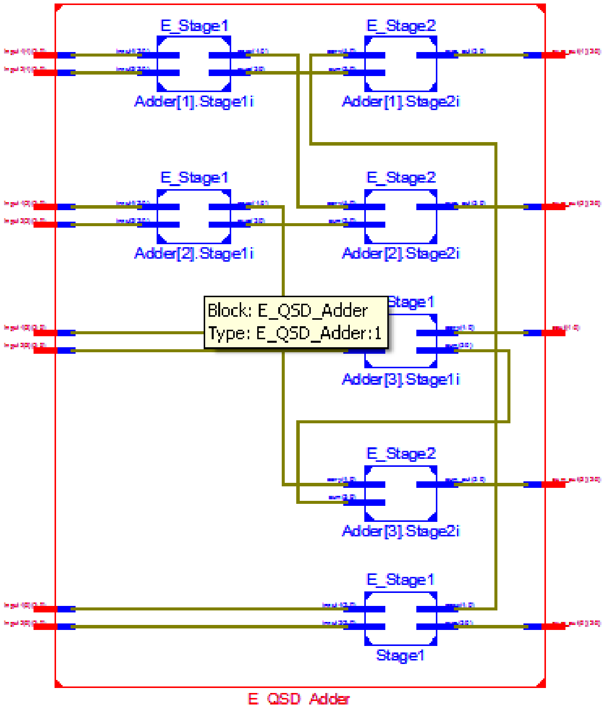


Fig 3. 4 Digit QSD Adder

IV. QSD MULTIPLIER:

Multiplication is done with partial product generators and QSD adders. Partial product generator consists of single digit multiplier and adders. For n x n QSD digit multiplication, n partial product generator circuit and n-1 QSD adders are required. Multiplication requires partial product generator and QSD adder [2]. Partial product generator produces M as result of multiplication between n digit input An-1-A0 with a single digit input Bi where i=0 to n-1. Partial product generator is a combination of a single digit multiplication unit and QSD added. The n digit partial product uses n single digit QSD multiplier.

TABLE III .
OUTPUTS OF THE STEP 1 QSD ADDER.

	-3	-2	-1	0	1	2	3
-3	9	6	3	0	-3	-6	-9
-2	6	4	2	0	-2	-4	-6
-1	3	2	1	0	-1	-2	-3
0	0	0	0	0	0	0	0
1	-3	-2	-1	0	1	2	3
2	-6	-4	-2	0	2	4	6
3	-9	-6	-3	0	3	6	9

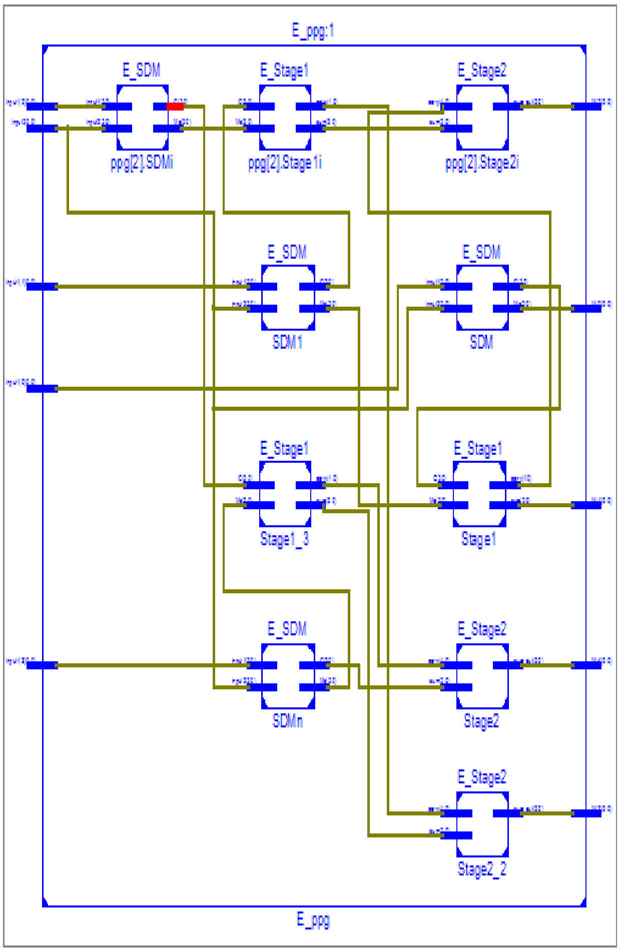


Fig 4. Partial Product Generator

The 4x1 bit partial product is shown in fig4.The n digit partial product uses n single digit QSD multiplier. The output of single-digit QSD multiplier is gathered by QSD Adder. Fig 5 4x4 QSD multiplier architecture. The table V shows representation of M and C.

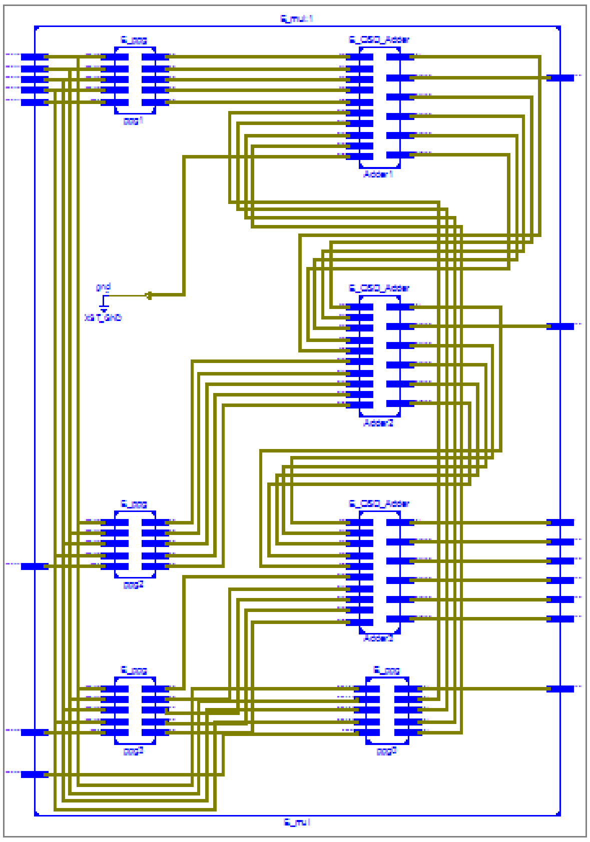
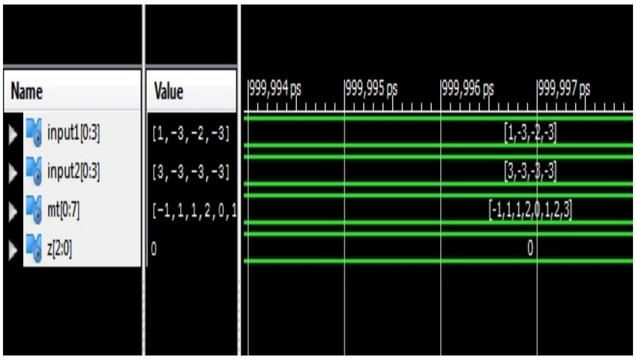


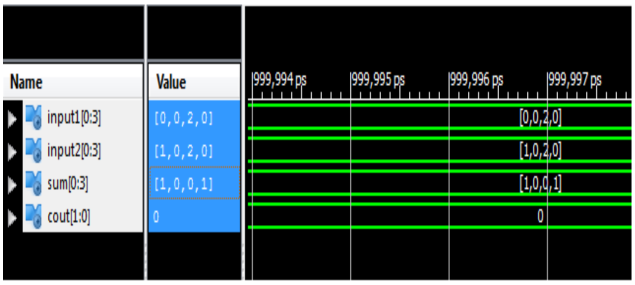
Fig 5. 4 Bit Multiplier

V. Simulation Results



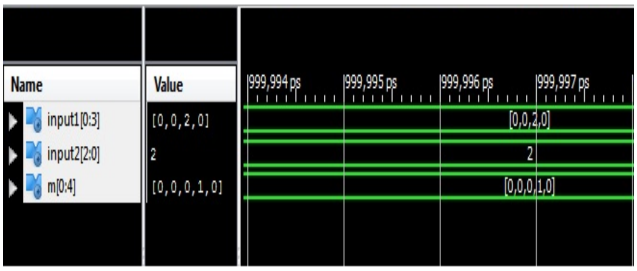
Name	Value	
input1[0:3]	[1,-3,-2,-3]	[1,-3,-2,-3]
input2[0:3]	[3,-3,-3,-3]	[3,-3,-3,-3]
mt[0:7]	[-1,1,1,2,0,1	[-1,1,1,2,0,1,2,3]
z[2:0]	0	0

Fig.6 simulation result of multiplier



Name	Value	
input1[0:3]	[0,0,2,0]	[0,0,2,0]
input2[0:3]	[1,0,2,0]	[1,0,2,0]
sum[0:3]	[1,0,0,1]	[1,0,0,1]
cout[1:0]	0	0

Fig 7. Simulation result of 4 bit adder



Name	Value	
input1[0:3]	[0,0,2,0]	[0,0,2,0]
input2[2:0]	2	2
m[0:4]	[0,0,0,1,0]	[0,0,0,1,0]

Fig8. Simulation result of partial product generator

TABLE IV
DEVICE UTILIZATION OF 4 BIT QSD MULTIPLIER

Logic Utilization	Used	Available	Utilization
No Slice Registers	488	4656	10%
Number of Slice LUTs	277	9312	2%
Number of Fully Used LUT-FF Pairs	898	9312	9%
Number of Bonded IOBs	51	232	21%

TABLE V
DEVICE UTILIZATION OF 4 BIT QSD ADDER

Logic Utilization	Used	available	Utilization
No Slice Registers	46	4656	2%
Number of Slice LUTs	15	9312	1%
Number of Fully Used LUT-FF Pairs	86	9312	2%
Number of Bonded IOBs	38	232	12%

Table VI compares the delays of proposed design with previous designs. It has been observe that delays of previous design are more than delay of QSD adder.

TABLE VI
COMPARISION OF QSD ADDER

Adder	Delay (ns)
Carry propagate adder	15.243
Carry look ahead adder	13.567
Carry skip adder	14.767
Qsd adder	6.21

It has been observed that while on increasing the number of bits of operation it gives constant delay. Fig. 7 shows the simulation result of 4bit QSD adder for various combinations of inputs and delay is found out to be 4.287 ns. Fig. 8 shows simulation result of partial product generator and Fig. 6 shows simulation result of 4x4 QSD multiplier and delay is found out to be 11.258 ns. Device Utilization Summary of the QSD adder and QSD multiplier is given in Table V and Table IV respectively. Were VHDL code is use for designing and Xilinx 13.2 design software is use for simulation where Targeted device used is belongs to Spartan 3, XC3S500E device, 4FG320 package.

REFERENCES

[1]. SachinDubey, Reena Rani, SarojKumari, Neelam Sharma, "VLSI implementation of Fast Addition using Quaternary Signed Digit Number System", *IEEE*2013.

[2] . Nagamani A. N, Nishchai S, "Quaternary High Performance Arithmetic Logic Unit Design", *14th Euromicro Conference on Digital System Design 2011 IEEE.*

[3] P. K. Dakhole, D.G. Wakde, " Multi Digit Quaternary adderon Programmable Device : Design and verification" International *conference on Electronic Design*,2008, 1-3 Dec, pp. 1-4.

[4] N. Takagi, H. Yasuura, and S. Yajima, ªHigh-Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree,º *IEEE Trans. Computers*, vol. 34, no. 9, pp. 789-796, Sept 1985.

[5] I. M. Thoidis, D. Soudris, J. M. Fernandez, A. Thanailakis, "The circuit design of multiple-valued logic voltage-mode adders," *IEEE*, 2001.

.

[6]. SongpolOngwattanakul, PhaisitChewputtanagul, David J. Jackson, Kenneth G. Ricks, "Quaternary Arithmetic Logic Unit on a Programmable Logic Device", *proc. IEEE conference*, 2001.

[7] J. U. Ahmed, A. A. S. Awwal, "Multiplier design using RBSD number system", Proceedings of the *1993 National Aerospace and Electronics Conference*, pp. 180-184, Vol. 1, 1993.

[8] A. Avizienis, "Signed-Digit Number Representation for Fast Parallel Arithmetic, "*IRE Transaction Electron. Comp.*, EC-10, pp. 389-400, 1961.

[9] Reena Rani, UpasanaAgrawal, Neelam Sharma, L.K Singh, "High Speed Arithmetic Logical Unit using Quaternary Signed Digit Number System" *International Journal Of Electronic Engineering Research*, ISSN 0975 – 6450, Volume 2 Number 3, 2010 pp. 383–391.

[10]. Vasundhara Patel K. S. Gurumurthy "Design of High Performance Quaternary Adders " *International journal of computer theory andEngg*,V01.2,No.6,Dec.,2010,1793-8201