Low Power and Area Efficient Folded Architecture for Row Bypassing Multiplier Using FPGA

[1]K.Sheela Shobana Rani, [2]S.Chithra, [3]R.Poornima

[1]Associate Professor, [2,3]Assistant Professor, Department of E.C.E.
[1]Sri Ramakrishna Institute of Technology, Coimbatore, Tamilnadu, India.
[2,3]Erode Sengunthar Engineering College, Perundurai, Tamilnadu, India
Email: [1]yuvsheka@gmail.com, [2]sd.chithra@gmail.com, [3]poornimavr7@gmail.com

Abstract- Optimizing power consumption and reducing hardware complexity are the most important criteria for the fabrication of DSP architectures. DSP operation must be completed within fixed time which demands the usage of high speed multiplier that consumes low power. In this work, a methodology to determine the best solution to this problem is presented by designing parallel and folded Row Bypassing Multiplier. The performance of Row Bypassing multiplier is analyzed using a 8bit multiplication operation. The design uses four 4×4 row bypassing multiplier blocks based on ripple carry adder. The proposed multiplier reduces power consumption by disabling adders resided in the multiplier when inputs are at zeros. To reduce area further, folding technique is used to realize a n× n multiplication using single n/2× n/2 multiplier. Control bit adder is designed to reduce the power consumption. Blocks are designed and simulated in using ALTERA QUARTUS II Performance of the Row Bypassing multiplier is analyzed in terms of logic cell counts, power dissipation, delay and power delay product.

Keywords-Bypassing multiplier, Ripple carry adder, Parallel folded multiplier, Compressor

I. INTRODUCTION

Modern technology demands circuits which can operate at ultra high speed and at the same time, which will consume least power. Low power consumption and smaller area are some of the most important criteria for the fabrication of DSP systems and high performance systems.

Optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas[3].To increase speed bypassing techniques are used for elimination of partial products when an input bit is zero. The operation of Row Bypassing multiplier is to disable adders based on multiplier bit, which reduces power consumption parallel architecture implementation can also be followed in such multiplier which shortens the delay. The Row Bypassed multiplier design significantly reduces the power consumption mainly due to less number of switching, which substantially improves the performances over the Array multiplier and other state-of-the-art high-performance multiplier designs.

A. Bypassed Multiplier

An 8 bit Row-Bypassed multiplier is design for reduced delay operation. The design uses two 8x4 Row Bypassed multiplier blocks including ripple carry adders based on bypassing technique, multiplexers and tristate buffers[6]. A CSA adder is adopted with bypassing ability in each row of adders. The reduced delay and power is achieved mainly due to the reduction in switching activity compared to the conventional logic design. Another technique to further reduce the power is to adopt ripple carry adder instead of carry save adder. The reason of adopting RCA adder instead of CSA adder is to achieve parallel architecture. Other power efficient multipliers such as Array multiplier, Wallace Tree multiplier were chosen for comparison with row bypassed multiplier, due to its simplicity in design while it has been demonstrated to consume minimum power compared to Array multiplier and other state-of-the-art designs [10]. Hence the reduced delay in row bypassed design compared to other multipliers ensures better performance over other multipliers.

B. Mathematical analysis of parallel multiplication

Consider A and B two n - bit numbers. Their 2n bit product is given by (1)

$$P=AB=\sum_{(i=0 \text{ to } 2n-1)} p_i 2^i$$
$$=\left(\sum_{(i=0 \text{ to } n-1)} b_i 2^i\right)\left(\sum_{(i=0 \text{ to } n-1)} b_i 2^i\right) \quad (1)$$

Fig.1.1 shows the architecture of n X n parallel multiplication. To obtain the final product each partial product is accumulated with the next. This accumulation results in high delay due to propagation of carry from LSB to MSB in during each addition. The delay due to carry propagation increases at the O (n^2) which limits the design of parallel multipliers for large values of n. To overcome this bypassing based multipliers are designed which just skip accumulation and partial product generation for multiplier bit equals to zero. An investigation of a class of bypassing multipliers is done in the following sections to analyse their merits and demerits.

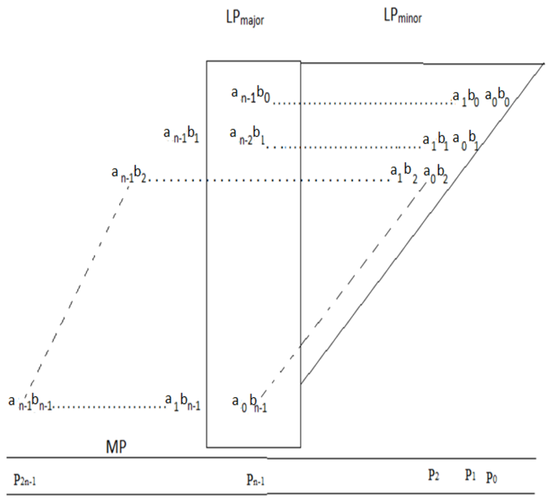

Fig. 1.1. Partial products for n*n multiplication

II. BYPASSING BASED ON RCA

The power consumption can be lower if the transitions of these input signals can be less frequent. Therefore, the most effective way to reduce the power of array based multiplier is to disable the transition of adder. The CSA based bypassing multiplier can save certain power consumption[5]. The additional circuits by adopting bypassing method can degrade the operation speed of multiplier. CSA based multiplier can achieve faster operation speed compared to RCA based multiplier. However, hardware cost is 50 percent more compared to conventional array multiplier.

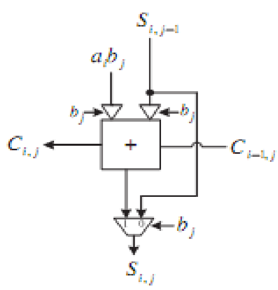

Fig.2.1.Modified Ripple Carry Full Adder

The proposed multiplier adopts the ripple-carry adder with fewer hardware components and parallel architecture. The new bypassing architecture is proposed to enhance operating speed and reduce power consumption of ripple-carry adder at same time. A RCA adder is adopted with bypassing ability in each row of adders. The reason of adopting RCA adder instead of CSA adder is to achieve parallel architecture.

A. Operation of Modified RCA Block

The two tri-state buffers are placed at two inputs of full adder to disable the operation of full adder when bj is 0. The tri-state buffer is designed by transmission gate (TG). The multiplexer is placed at the sum output of full adder. The value of sum can be selected from the bypassing value or sum output of full adder according to the value of bj. The proposed design does not need to add multiplexer for carry output and tri-state buffer for carry input of full adder. The reason is that two inputs of full adder in jth row need to be disabled while the value of bj is 0. Thus carry outputs of the full adders in the same row cannot be changed since two out of three-input full adder is disabled.

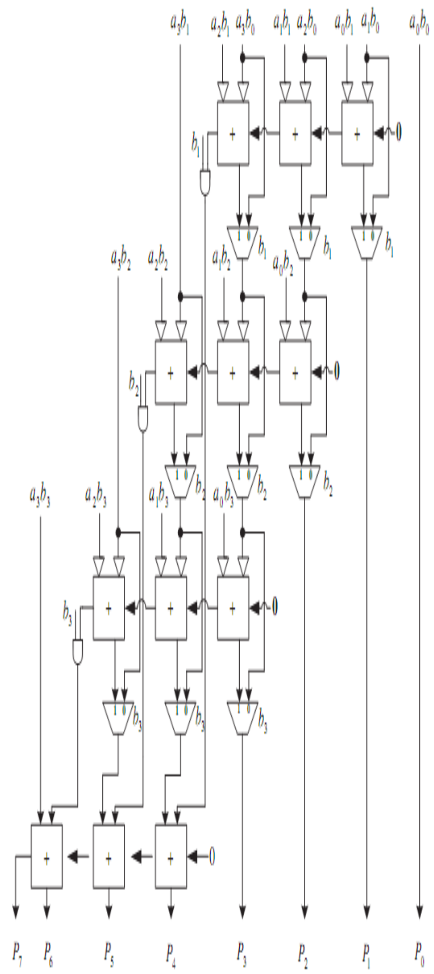

Fig.2.2.4×4 bypassing multiplier based on ripple carry array

Therefore, significant portion of extra hardware can be saved without degrading speed performance. In addition, power consumption also can be reduced as a result of reduced hardware activities [1]. The proposed RCA full adder only needs two tri-state buffers and one multiplexer. On the other hand, the CSA full adder designs needs three tri-state buffers and two multiplexers. It is evident that the proposed design can reduce hardware area.

B. Signed Bypassing Multiplier Design

The multiplier in the previous section is used to compute unsigned numbers. However, it is essential to design signed multipliers because computer system usually manipulates signed numbers [5]. With regard to signed multiplier design, some signed multiplication algorithms are proposed. In conventional array multipliers such as Braun multiplier, signed multipliers can be realized through Baugh–Wooley multiplication algorithm, often used to deal with signed multiplication. The algorithm uses 2's complement to represent the signed numbers and also uses the same framework of array multiplier. The advantage of this algorithm is accomplishing signed multiplication without expanding sign bits. Consequently, additional hardware cost is not increased; thus, not dissipating extra power. Only the AND gate to NAND gate for corresponding operands is changed and an inverter is inserted at the final carry output.

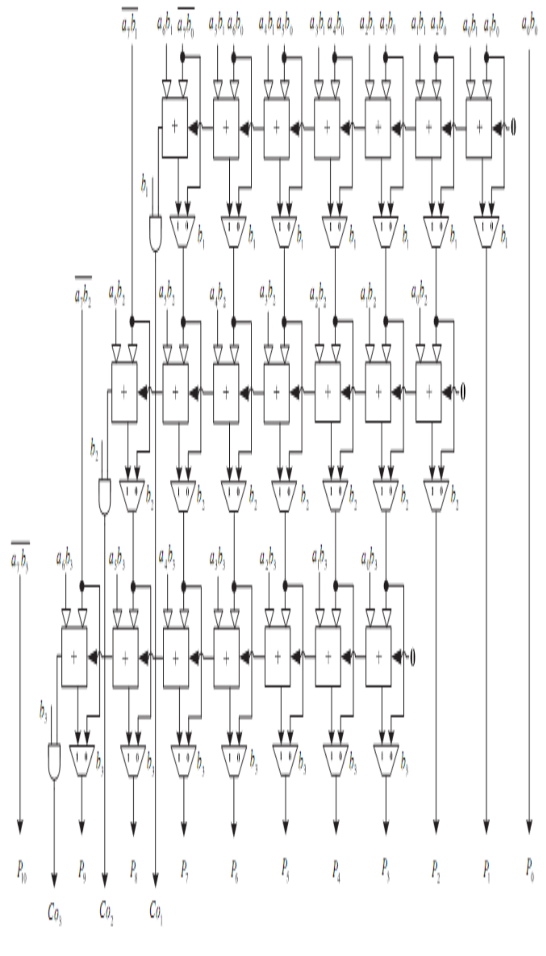

Fig. 2.3. An 8×4 signed bypassing multiplier.

The proposed multiplier also adopts the Baugh–Wooley algorithm for signed number multiplication. Considering an 8×8 signed multiplication, all operands are separated into two parts.

Full adders are used to compute the last row of operands. Therefore, two different 8×4 bit signed ripple-carry array multipliers need to be designed. Blocks 1 and 2 are the two 8×4 bit signed ripple- carry array multipliers, respectively. Therefore we have four parallel blocks, each working in parallel and generating partial and generating partial products. The addition operation in the columns of each block can be performed by performed by choosing the half adders, full adders, 4:2 compressors and 5:2 compressors according to the number of bits to be added. Thereby, full adder only needs two tri-state buffers and one multiplexer. Moreover an AND gate is inserted into the last carry output in each row of full adder for correcting output when the value of bj is 0.

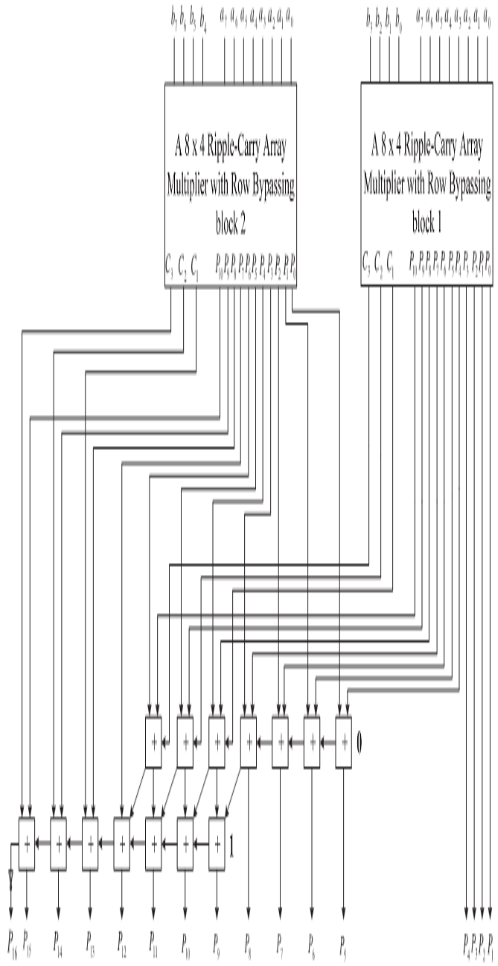

Fig.2.4. 8-Bit Row Bypassing Parallel Multiplier

III. PROPOSED HIGH SPEED ROW BYPASSING MULTIPLIERS

A. Parallel RB Multiplier

In Parallel Row Bypassing multiplier, 8×8 multiplier is designed using four 4×4 multiplier blocks. The design used half adders, full adder and compressor blocks. The delay is reduced to a considerable amount. Therefore we have four parallel blocks, each working in parallel and generating partial and generating partial products. The addition operation in the columns of each block can be performed by performed by choosing the half adders, full adders, 4:2 compressors and 5:2 compressors according to the number of bits to be added.

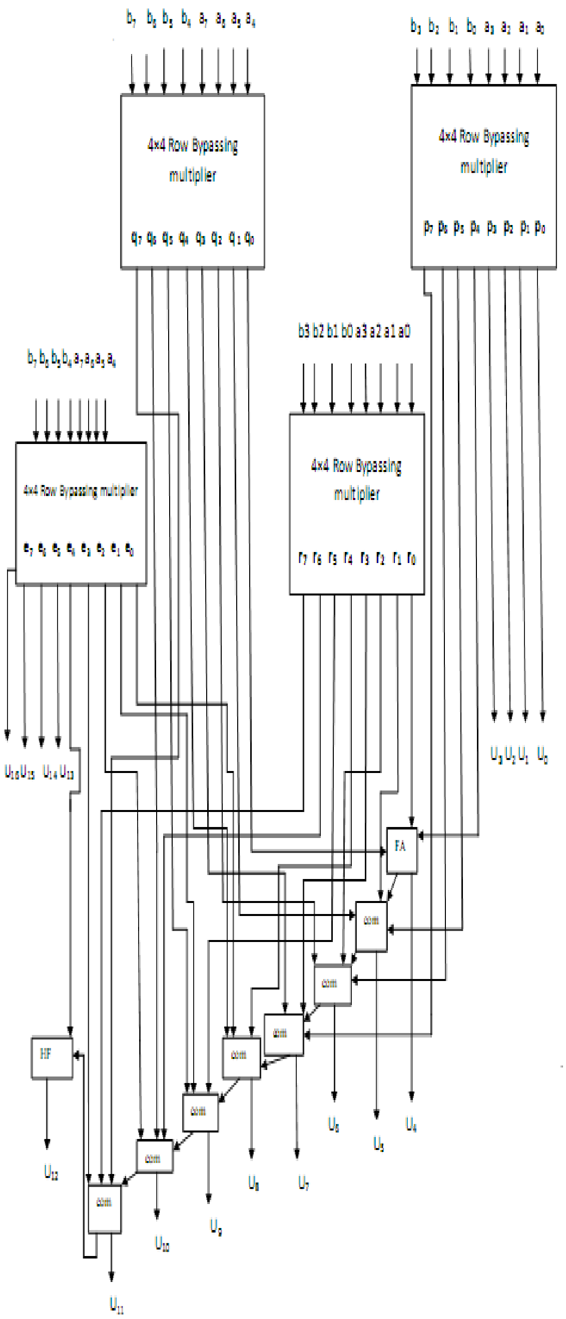

Fig. 3.1. 8-Bit Row Bypassing Parallel Multiplier

IV.PROPOSED FOLDED RBP MULTIPLIER

Here the partial products are grouped into quarters as shown in the figure4.1. Each quarter is named and they are executed simultaneously. Finally using compressors and full adders the partial products are summed to get the resulting values from s_0 to s_{15} Thus by using the concept of folding in Row Bypassing Multiplier considerable amount of delay can be reduced due to parallel operation. The area overhead can be reduced by dividing an eight bit multiplier into four (4*4) blocks, activating each block based on input clock pulse forced, then final resultant is obtained by accumulating partial products[4].

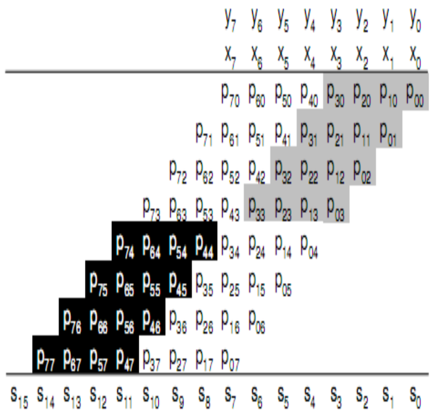

Fig.4.1. Partial Product Representation of a 4-Bit Multiplication in an 8-Bit Multiplier.

In Figure 3.2 the input of the 2 bit counter is global clock. The output of the 2 bit counter is given to the input selector and 16 bit shift accumulator. The function of the input selector is to select one of the 4-bit input pair from 8 bit inputs A and B. Usually the input selector will be 4-1 mux with select signals as 2bit counter output.

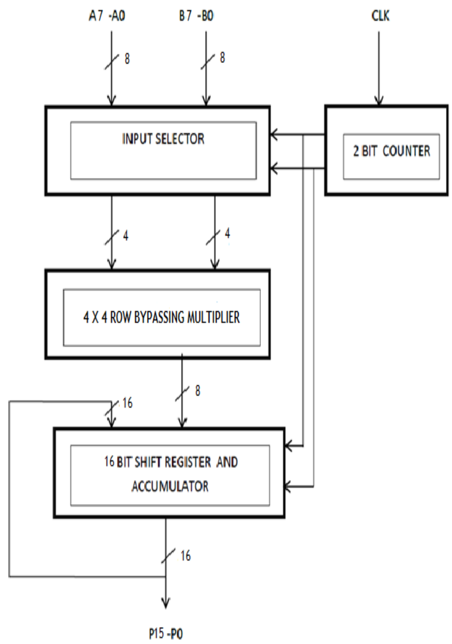

Fig.4.2. Block Diagram of Proposed 8x8Row Bypassing Multiplier Using Folding

The main components of proposed architecture are input selector, 4×4 multiplier 2 bit counter, product register and accumulator. Based on the counter output, the particular pair of inputs are fed through the 4×4 multiplier as shown in table 4.1.

The 4x4 Row bypassing multiplier produces 16 bit output for each clock cycle. The output of the multiplier is stored in a 16bit product register. The product register is shifted by counter output before the contents of product register are accumulated with the contents of accumulator [8] .The number of bits to be shifted for each counter output is shown in table 1 and depends on the bit positions. During each cycle a partial product is realized and accumulated with accumulator. After 4 cycles the final product of 8X8 multiplication is realized.

Table 4.1 Operation of 8x8 Optimized Row Bypassing Multiplier

2 Bit counter output	Inputs	Operation on product register
00	$A_{(3-0)}$ & $B_{(3-0)}$	no shift in 8bit output
01	$A_{(3-0)}$& $B_{(7-4)}$	4bit left shift in 8 bit output
10	$A_{(7-4)}$& $B_{(3-0)}$	4bit left shift in 8bit output
11	$A_{(7-4)}$ & $B_{(7-4)}$	4 bit left shift in 8 bit output

V. PERFORMANCE ANALYSIS

The parameters considered for the analysis of the proposed Parallel Row Bypassing Multiplier are Power, Delay and Area. The parameters are obtained using Quartus II

A. PDP Analysis

The table 5.1 illustrates that the proposed Row Bypassing Parallel Multiplier and folded RBP has reduced delay compared to other multipliers. The power consumption of parallel multiplier is much more less when Wallace multiplier. Thus the speed of the multiplier is enhanced when compared to Array multiplier and conventional Row Bypassing multipliers.

B. ADP Analysis

From the table 5.2 it is clear that the folded multiplier occupies less area when compared to Array multiplier, Wallace tree multiplier and other bypassing designs.

Thus the area overhead is is reduced in the proposed 8×8 multiplication which is realized using single 4 ×4 multiplier triggered by clock pulse for each pair of inputs.

Table 5.1 Comparison of Power, Delay and Power-Delay Product (PDP) of proposed Parallel Row Bypassing Multiplier and other Standard Designs.

Parameters / Design	Total Power Dissipation (mW)	Delay (ns)	PDP (pJ)
Array multiplier [5]	168.23	26.53	4463.14
Conventional RB [1]	168.40	27.60	4647.84
RB (8*4) Cascading [1]	169.56	24.45	4145.742
Wallace [3]	323.65	20.7	6699.55
Proposed RB (4*4) cascading	167.70	24.40	4091.88
Proposed - folded R.B	243.39	22.5	5476.275

VI. PROPOSED LOW POWER DESIGN

A. Control Bit Adder

The adder is designed along with control input ‘p’. The selection bit of multiplexer is used to select either the previous value or else the output of the full adder block. This adder along with control input replaces the multiplexer in the existing bypassing structure. Based on this controlled input any one of the value is bypassed. The switching activity gets minimized; thereby propagation delay is also minimized. Considerable amount of power savings can also be done.

Based on control input p,

if p=0

sum<=a

else

sum<= a xor b xor c

carry<= (a and b) or (b and c) or (a and c)

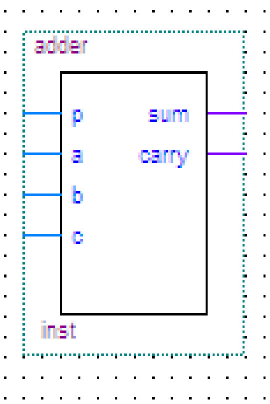

Fig.6.1. Control Bit Adder

Table 5.2 Comparison of Area, Delay and Area-Delay Product (ADP) of proposed Parallel Row Bypassing Multiplier and other Standard Designs

Parameters / Design	Delay (ns)	Area (logic cell count)	ADP
Array multiplier [5]	26.53	156	4138.68
Conventional RB [1]	27.60	135	3734.1
RB (8*4) Cascading [1]	24.45	198	4841.1
Wallace [3]	20.7	161	3332.7
Proposed RB (4*4) cascading	24.40	171	4172.4
Proposed - folded R.B	22.5	125	2812.5

VII. CONCLUSION

The design and operation of row bypassing multiplier based on ripple carry adder and carry save adder is described. The proposed multiplier reduces power consumption by disabling adders resided in the multiplier when input bits are zero. Delay of the proposed multiplier is also reduced since parallel structures are employed for parallel product generation. The circuits are designed and structured in ALTERA QUARTUS II. Simulation analysis shows that the proposed RBP and folded RBP have reduced delay compared to conventional Row Bypassing Multiplier. To reduce area further, folding technique is used to realize a n× n multiplication using single n/2× n/2 multiplier.

Future Work

Further enhancements can be done by extending the proposed methodology to higher bit multiplier. Also comparative analysis can be made by implementing various adder structures in the proposed design.

REFERENCES

[1] Bipul C.Paul, 'ROM-Based Logic (RBL) Design: A Low-Power 16 Bit Multiplier', IEEE Journal of Solid-State Circuits, VOL,44, NO. 11, pp. 2935-2941, 2009.

[2] C.H. Han, H.J. Park, L.S. Kim , 'A low-power array multiplier using separated multiplication technique', IEEE Transactions on Circuits and Systems-II, Analog and Digital Signal Processing 48 866–871, 2001.

[3] Chen L.H, Chen O.T.-C, Wang T.Y, 'Multiplication-accumulation computation unit with optimized compressors and minimized switching activities', in: Proceedings of the IEEE International Symposium on Circuits and Systems', pp. 6118–6121, 2005.

[4] Chang-Young Han, Hyoung-Joon Park, and Lee-Sup Kim , 'A Low-Power Array Multiplier Using Separated Multiplication Technique', IEEE Transactions On Circuits And Systems—II: Analog And Digital Signal Processing, VOL. 48, NO. 9, pp.866-871, 2001.

[5] Ito M, Chinnery D., Keutze K, 'Low power multiplication algorithm for switching activity reduction through operand decomposition', in:Proceedings of the 21st International Conference on Computer Design, pp.21–26, 2003.

[6] Ko-Chi Kuo ,Chi-Wen Chou, 'Low power and high speed multiplier design with row bypassing and parallel architecture', Microelectronics Journal No.41,pp.639–650.

[7] Kwen-Siong Chong, Bah-Hwee Gwee , 'A Micropower Low-Voltage Multiplier With Reduced Spurious Switching',IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, VOL. 13, NO. 2, pp.255-265,2 005.

[8] Pascal C. H. Meier, Rob A. Rutenbar and L. Richard Carley, 'Exploring Multiplier Architecture and Layout for Low Power, IEEE Conference Custom Integrated Circuits,1996.

[9] Pekmestzi K.Z, 'Multiplexer-based array multiplier', IEEE Transactions on Computers,Vol.48, No.2, pp.15–23, 1999.

[10] Rizwan Mudassir, H. El-Razouk and Z. Abid ,'New Designs of Signed Multipliers', IEEE Conference on Custom Integrated Circuits, 1996.