International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 3, May - June 2013 pp-1587-1592 ISSN: 2249-6645

A Parallel Multiplier - Accumulator Based On Radix – 4 Modified Booth Algorithms by Using Spurious Power Suppression Technique

S. Tabasum, M. P. Chennaiah
M.Tech (VLSI), SSITS, Rayachoty, Kadapa dist India,
Associate prof, SSITS, Rayachoty,Kadapa dist India

Abstract: *In this paper, we proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. This can be implement by using radix-2 booth encoder .By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. This includes the design exploration and applications of a spurious-power suppression technique (SPST) which can dramatically reduce the power dissipation of combinational VLSI designs. Power dissipation is recognized as a critical parameter in modern VLSI field. In Very Large Scale Integration, Low power VLSI design is necessary to meet MOORE'S law and to produce consumer electronics with more back up and less processing systems. The proposed MAC accumulates the intermediate results in the type of sum and carry bits instead of the output of the final adder, which made it possible to optimize the pipeline scheme to improve the performance. The objective of a good multiplier is to provide a physically compact, good speed and low power consuming chip. To save significant power consumption of a VLSI design, it is a good direction to reduce its dynamic power that is the major part of power dissipation.*

Keywords: *low-power design, array multiplier, booth encoder, carries save adder, accumulation, SPST adder, multiplier and accumulator (MAC).*

I. Introduction

In present most digital signal processing methods use nonlinear *functions* such as discrete cosine transform (DCT) or *discrete* wavelet transform (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication *and* addition arithmetic's determines the execution speed and performance of the entire calculation and yield. **O**ne of the accompanying challenges in designing ICs for portable electrical devices is lowering down the power consumption to prolong the operating time on the basis of given limited energy supply from batteries. Therefore, dedicated low- power techniques are not important for high speed multiplication .

The design in proposes a concept called partially guarded computation(PGC), which divides the arithmetic units, e.g., adders and multipliers, into two parts and turns off the unused part to minimize the power consumption. In general, a multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree instead of the array of FAs., i.e., this multiplier mainly consists of the three parts: i.Booth encoder, ii.a tree to compress the partial products such as Wallace tree, and iii. adder. Because Wallace tree is to add the partial products from encoder as parallel as possible. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication proceeds a series of additions for the partial products. To reduce the number of calculation steps for the partial products, MBA algorithm has been applied mostly where Wallace tree has taken the role of increasing the speed to add the partial products. By using g Booth encoder we can reduce te number partial products it dependence radix. in place of booth encoder we can replace this with spst adder/sub it gives better result.
Thi*s* paper is organized as follows. In Section II, a simple introduction of a general MAC will be given, and the architecture for the proposed SPST will be described in Section V. In Section III, the BOOTH encoder is described. Finally, the conclusion will be given in Section VI.

II. Overview of Mac

In this section, basic MAC operation is introduced. A multiplier can be divided into three operational teps.

i. The first is radix-2 encoding in which a partial product is generated from the multiplicand (X) and the multiplier(Y).
ii. The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry.
iii. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry.

If the process to accumulate the multiplied results is included, a MAC consists of four steps, as shown

Fig. 1 which shows the operational steps explicitly
Step1. Booth encoding
Step2. Partial product summation and accumulation,
step3. Final addition

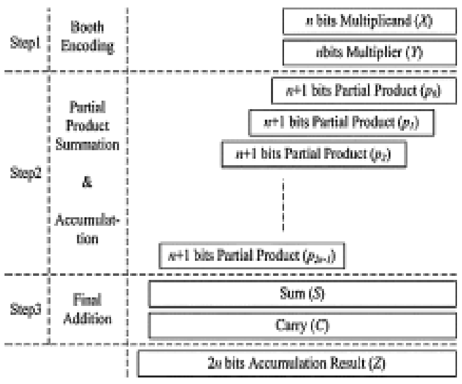

Step1
Booth Encoding
n bits Multiplicand (X)
nbits Multiplier (Y)
Step2
Partial Product Summation & Accumulation
n+1 bits Partial Product (p0)
n+1 bits Partial Product (p1)
n+1 bits Partial Product (p2)
n+1 bits Partial Product (p2n-1)
Step3
Final Addition
Sum (S)
Carry (C)
2n bits Accumulation Result (Z)

Fig. 1 steps for multiplication and accumulation.

A general hardware architecture of this MAC is shown in Fig. 2. It executes the multiplication operation by multiplying the input multiplier X and the multiplicand Y.

The N -bit 2's complement binary number X can be expressed as

$$XxY = —2^{N-1}x_{N-1}+ \sum_{i=0}^{N-2} 2^{i} \qquad (1)$$

If (1) is expressed in base-4 type redundant sign digit form in order to apply the radix-2 Booth's algorithm.

$$X = \sum_{i=0}^{\frac{N}{2}-1} d_i 4_i \qquad (2)$$

$$d_i = —2x_{2i+1}+x_{2i}+x_{2i-1} \qquad (3)$$

If (2) is used, multiplication can be expressed as

$$XxY=\sum_{I=0}^{\frac{N}{2}-1} 2^{2i}\, Y \qquad (4)$$

If eq used after multiplication and accumulation

$$P=X+Y+Z=\sum_{i=0}^{\frac{N}{2}-1} d_i\, 2^{i}Y+\sum_{J=0}^{2N-1} z_i\, 2^{i} \qquad (5)$$

Each of the two terms on the right-hand side of eq(5) is calculated independently and the final result is produced by adding the two results. The MAC architecture implemented by eq(5) is called the standard design [6]. If N-bit data are multiplied, the number of the generated partial products is proportional to N. In order to add them serially, the execution time is also proportional to the architecture of a multiplier, which is fatest,uses fastest, uses radix-2 Booth encoding that generates partial products and a Wallace tree based on CSA as the adder array to add the partial products. If radix-2 Booth encoding is used, the number partial products, i.e., the inputs to the Wallace tree, is reduced to half, resulting in the decrease in CSA tree step. Each block is decoded to generate the correct partial product. The encoding of the multiplier Y, using the modified booth algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table 1.

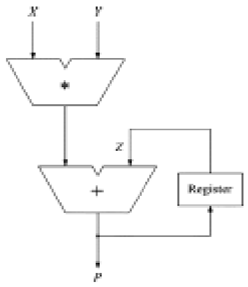


Fig. 2.MAC hardware architecture

III. Modified Booth Encoder

To Booth recode the multiplier term, consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Figure 3 the grouping of bits from the multiplier term for use in modified booth encoding. sum and carry.

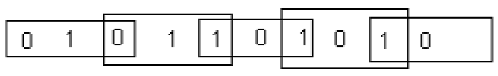


Fig.3. Grouping of bits from the multiplier term

Table.1.Recoding of bits

Block	Re - coded digit	Operation on X
000	0	0 X
001	+1	+1 X
010	+1	+1 X
011	+2	+2 X
100	-2	-2 X
101	-1	-1 X
110	-1	-1 X
111	0	0 X

. IV. Proposed Mac Architecture

In this section, the expression for the new arithmetic will be derived from equations of the standard design. From this result, VLSI architecture for the new MAC will be proposed. In addition, a hybrid-typed CSA architecture that can satisfy the operation of the proposed MAC will be proposed

A. Derivation of MAC Arithmetic

1) Basic Concept: If an operation to multiply two -bit numbers and accumulate into a 2-bit number is considered, the critical path is determined by the 2 -bit accumulation operation. If a pipeline scheme is applied for each step in the standard design of Fig. 1, the delay of the last accumulator must be reduced in order to improve the performance of the MAC. The overall performance of the proposed MAC is improved by eliminating the accumulator itself by combining it with the CSA function. If the accumulator has been eliminated, the critical path is then determined by the final adder in the multiplier. The basic method to improve the performance of the final adder is to decrease the number of input bits. In order to reduce this number of input bits, the multiple partial products are compressed into a sum and a carry by CSA. The number of bits of sums and carries to be transferred to the final adder is reduced by adding the lower bits of sums and carries in advance within the range in which the overall performance will not be degraded. A 2-bit CLA is used to add the lower bits in the CSA. In addition, to increase the output rate when pipelining is applied, the sums and carrys from the CSA are accumulated instead of the outputs from the final adder in the manner that the sum and carry from the CSA in the previous cycle are inputted to CSA. Due to this feedback of both sum and carry, the number of inputs to CSA increases, compared to the standard design and [17].

2) Equation Derivation: The aforementioned concept is applied to (5) to express the proposed MAC arithmetic. Then, the multiplication would be transferred to a hardware architecture that complies with the proposed concept, in which the feedback value for accumulation will be modified
and expanded for the new MAC. First, if the multiplication in (4) is decomposed and rearranged, it becomes

$$\text{XxY}=d_0 2Y + d_1 2^2 Y + d_2 2^4 \text{Y}+ \ldots\ldots d_{\frac{N}{2}-1} 2^{N-2} \qquad (6)$$

If (6) is divided into the first partial product, sum of the middle partial products, and the final partial product, it can be expressed as eq(7). The reason for separating the partial X product addition as eq(7) is that three types of data are fed

$$XxY=d_0 2Y + \sum_{i=0}^{\frac{N}{2}-2} d_i 2^{2i} Y + d_{\frac{N}{2}-1} 2^{N-2} Y \qquad (7)$$

Now, the proposed concept is applied to Z in (5). If Z is first divided into upper and lower bits and rearranged, (8) will be derived. The first term of the right-hand side in (8) corresponds to the upper bits. It is the value that is fed back as the sum and the carry. The second term corresponds to the lower bits and is the value that is fed back as the addition result for the sum and carry .

$$\text{Z}=\sum_{i=0}^{N-1} z_i 2^i + \sum_{i=N}^{2N-1} z_i 2^i \qquad (8)$$

The second term can be separated further into the carry term and sum term as

$$\sum_{i=N}^{2N-1} z_i 2^i = \sum_{i=0}^{N-1} z_{N+i} 2^{i+N} = \sum_{i=0}^{N-2} (c_i + s_i) 2^{i+N} \qquad (9)$$

Thus, (8) is finally separated into three terms as

$$\text{Z}=\sum_{i=0}^{N-1} z_i 2^i + \sum_{i=0}^{N-2} c_i 2^i 2^N + \sum_{I=0}^{N-2} s_i 2^i 2^N \qquad (10)$$

If (7) and (10) are used, the MAC arithmetic in (5) can be expressed as

$$\text{P}=d_0 2Y + \sum_{i=1}^{\frac{N}{2}-2} d_i 2^{2i} Y + d_{\frac{N}{2}-1} 2^{N-2} Y + \sum_{i=0}^{N-1} z_i 2^i 2^N + \sum_{i=0}^{N-2} c_i 2^i 2^N + \sum_{i=0}^{N-2} s_i 2^i 2^N \qquad (11)$$

If each term of (11) is matched to the bit position and rearranged, it can be expressed as (12), which is the final equation for the proposed MAC. The first parenthesis on the right is the operation to accumulate the first partial product with the added result of the sum and the carry.The second parenthesis is the one to accumulate the middle partial products with the sum of the CSA that was fed back. Finally, the third parenthesis expresses the operation to accumulate the last partial product with the carry

$$P=d_0 2Y + \sum_{i=0}^{N-1} z_i 2^i) + \sum_{i=1}^{\frac{N}{2}-1} d_i 2^{2i} Y + \sum_{i=0}^{N-2} c_i 2^i 2^N) + (d_{\frac{N}{2}-1} 2^{N-2} Y + \sum_{i=0}^{N-2} s_i 2^i 2^N \qquad (12)$$

B. Proposed MAC Architecture

If the MAC process which the MAC is organized into three steps. When shown in Fig. 1, it is easy to identify the difference that the accumulation has been merged into the process of adding the partial products. Another big difference from Fig. 1 is that the final addition process in step 3 is not always run even though it does not appear explicitly in Fig. 3. Since accumulation is carried out using the result from step 2 instead of that from step 3, step 3 does not have to be run until the point at which the result for the final accumulation is needed.

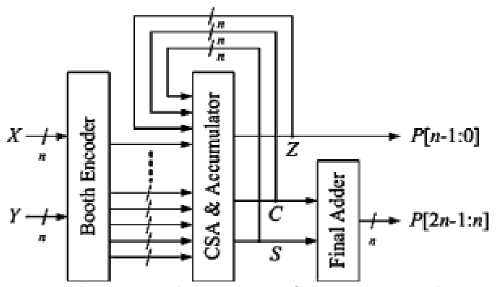


Fig. 4 . Hardware architecture of the proposed MAC

The n -bit MAC inputs, X and Y, are converted into an (n+1)-bit partial product by passing through the Booth encoder. In the CSA and accumulator, accumulation is carried out along with the addition of the partial products. As a result, n -bit S ,C and Z (the result from adding the lower bits of the sum and carry) are generated. These three values are fed back and used for the next accumulation. If the final result for the MAC is needed, P[2n-1:n] is generated by adding and C in the final adder and combined with P[n-1:0] that was already generated.

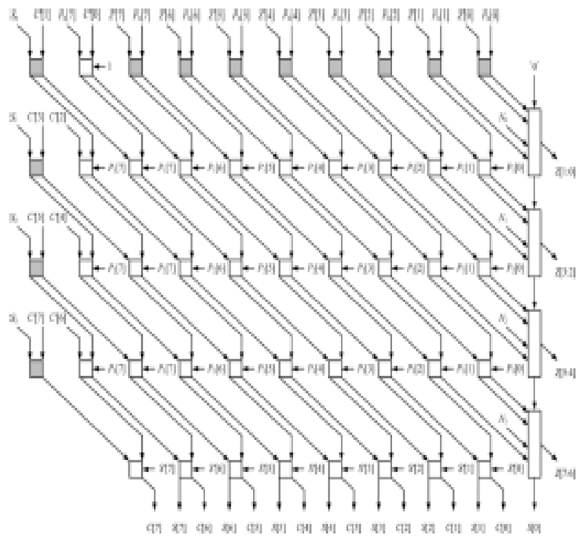


Fig.5 proposed MAC architecture

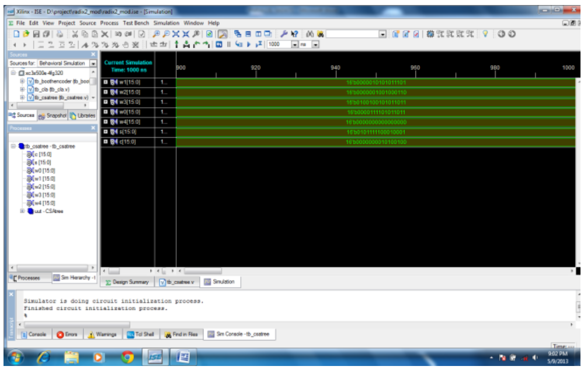
C. Proposed CSA Architecture

The architecture of the hybrid-type CSA that complies with the operation of the proposed MAC is shown in Fig. 5, which performs 8X 8-bit operation. It was formed based on (12). In Fig. 6, Si is to simplify the sign expansion and Ni is to compensate 1's complement number into 2's complement number. S[i] and C[i] correspond to the i th bit of the feedback sum and carry. Z[i] is the i th bit of the sum of the lower bits for each partial product that were added in advance and is the previous result. In addition, Pj[i] corresponds to the i th bit of the j th partial product. Since the multiplier is for 8 bits, totally four partial products (P0[7:0] ~P3[7:0]) are generated from the Booth encoder. In (11), d0Y and d N/2-1 2^(N-2)Y correspond toP0[7:0] andP3[7:0] respectively.

This CSA requires at least four rows of FAs for the four partial products. Thus, totally five FA rows are necessary since one more level of rows are needed for accumulation. For an n x n -bit MAC operation, the level of CSA is (n/2+1) The white square in Fig. 5 represents an FA and the gray square is a half adder (HA). The rectangular symbol with five inputs is a 2-bit CLA with a carry input.

V. Simulation Results

Csa output:

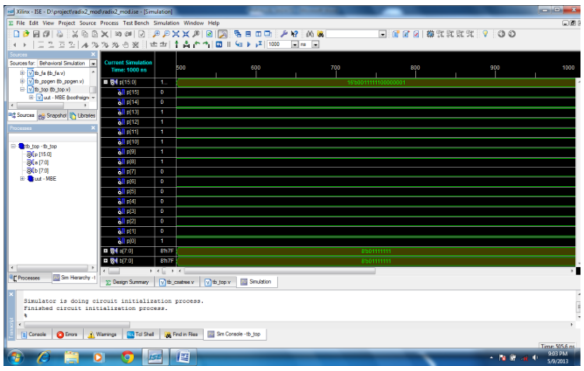


```
Current Simulation
Time: 1000 ns
w1[15:0]  1...  16'b0000010101011101
w2[15:0]  1...  16'b0000001001000110
w3[15:0]  1...  16'b0100100101011011
w0[15:0]  1...  16'b0000111101011011
w4[15:0]  1...  16'b0000000000000000
s[15:0]   1...  16'b0101111100010001
c[15:0]   1...  16'b0000000010100100
```

```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Fig.6. Csa outputs

Mac output:



```
Current Simulation
Time: 1000 ns
p[15:0]  1...   16'b0011111100000001
  p[15]  0
  p[14]  0
  p[13]  1
  p[12]  1
  p[11]  1
  p[10]  1
  p[9]   1
  p[8]   1
  p[7]   0
  p[6]   0
  p[5]   0
  p[4]   0
  p[3]   0
  p[2]   0
  p[1]   0
  p[0]   1
a[7:0]   8'h7F  8'b01111111
b[7:0]   8'h7F  8'b01111111
```

```
Simulator is doing circuit initialization process.
Finished circuit initialization process.
```

Fig.7.mac outputs

VI. Future Scope of Project

SPST technique:

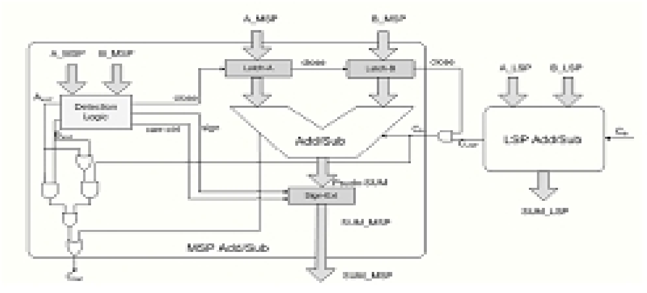


Fig.8 Low-power adder/subtractor design example adopting the proposed SPST

Moreover, we propose the novel glitch-diminishing technique by adding three 1-bit registers to control the assertion of the *close*, *sign*, and *carr-ctrl* signals to further decrease the transient signals occurred in the cascaded circuits which are usually adopted in VLSI architectures designed for multimedia/DSP applications. Hence, the transients of the detection-logic unit can be filtered out; thus, the data latches can prevent the glitch signals from flowing into the MSP with tiny cost..

1) *When the detection-logic unit turns off the MSP:* At this moment, the outputs of the MSP are directly compensated by the SE unit; therefore, the time saved from skipping the computations in the MSP circuits shall cancel out the delay caused by the detection-logic unit.

2) *When the detection-logic unit turns on the* MSP circuits must wait for the notification of the detection- logic unit to turn on the data latches to let the data in. Hence, the delay caused by the detection-logic unit will contribute to the delay of the whole combinational circuitry, the 16-bit adder/subtractor in this design example.

3) ***When the detection-logic unit remains its decision:*** No matter whether the last decision is turning on or turning off the MSP, the delay of the detection logic is negligible because the path of the combinational circuitry (i.e., the 16- bit adder/subtractor in this design example) remains the same.

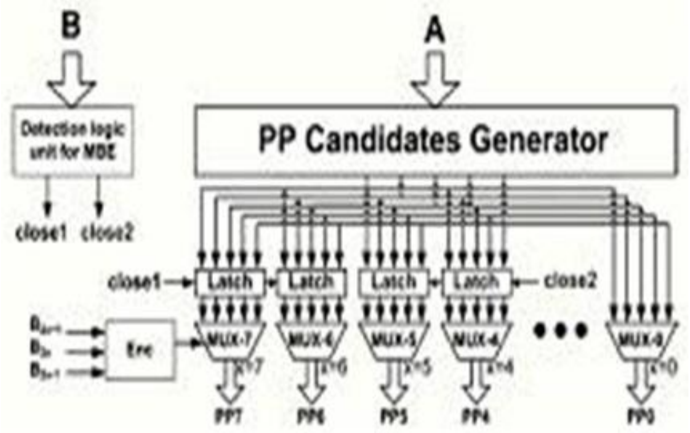


Fig.9 Illustration of multiplication using modified Booth encoding.

From the analysis earlier, we can know that the total delay is affected only when the detection-logic unit turns on the MSP. However, the detection-logic unit should be a speed- oriented design. When the SPST is applied on combinational circuitries, we should first determine the longest transitions of the interested cross sections of each combinational circuitry, which is timing characteristic and is also related to the adopted technology.

The longest transitions can be obtained from analyzing the timing differences between the earliest arrival and the latest arrival signals of the cross sections of a combinational circuitry. Then, a delay generator similar to the delay line used in the DLL designs [16], [17],comprising several invertors and some capacitors, can be used to generate a proper delay to control the "*close*," "*sign*," and "*carr-ctrl*" signals.

VII. Conclusion

In this paper, a new MAC architecture to execute the multiplication- accumulation operation, which is the key operation, for digital signal processing and multimedia information processing efficiently, was proposed.By removing the independent accumulation process that has the largest delay and merging it to the compression process of the partial products, the overall MAC performance has been improved almost twice as much as in the previous work. The proposed SPST can obviously decrease the switching (or dynamic) power dissipation, which comprises a significant portion of the whole power dissipation in integrated circuits. The performance comparisons also illustrate that the SPST-equipped designs are very competitive with the existing designs. Furthermore, the proposed SPST is a fully static CMOS circuit technique which does not aggravate the problems of leakage power, signal racing, and voltage dropping. While the delay has been increased slightly compared to the previous research, actual performance has been increased to about twice if the pipeline is incorporated. Consequently, we can expect that the proposed architecture can be used effectively in the area requiring high throughput such as a real-time digital signal processing.

Reference

[1] O. L. MacSorley, "High speed arithmetic in binary computers," Proc. IRE, vol. 49, pp. 67-91, Jan. 1961.
[2] A D. Booth, "A signed binary multiplication technique," Quart. J. Math., vol. IV, pp. 236-240, 1952.
[3] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14-17, Feb. 1964.
[4] A. R. Cooper, "Parallel architecture modified Booth multiplier," Proc. Inst. Electr. Eng. G, vol. 135, pp. 125- 128, 1988.
[5] N.R.Shanbag and P. Juneja, "Parallel implementation of a 4x4-bit multiplier using modified Booth's algorithm," IEEE J. Solid-State Circuits, vol. 23, no. 4, pp. 1010-1013, Aug. 1988.
[6] G Goto,T. Sato, M. Nakajima, and T. Sukemura, "A 54x54 regular structured tree multiplier," IEEE J. Solid- State Circuits, vol. 27, no. 9, pp. 1229-1236, Sep. 1992.
[7] J.Fadavi-Ardekani, "MN Booth encoded multiplier generator using optimizedWallace trees," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, pp. 120-125, Jun. 1993.
[8] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka,A. Shimizu,K. Sasaki,and Y. Nakagome, "A 4.4 ns CMOS 54x54 multiplier using pass-transistor multiplexer, IEEE J. Solid-State Circuits, vol. 30, no. 3, pp. 251-257, Mar. 1995.
[9] A.Tawfik, F. Elguibaly, and P. Agathoklis, "New realization and implementation of fixed-point IIR digital filters," J. Circuits, Syst., Comput., vol. 7, no. 3, pp. 191- 209, 1997.