DESIGN OF LOW POWER MULTIPLIER WITH REDUCED SPURIOUS TRANSITION ACTIVITY TECHNIQUE FOR EFFICIENT NEURAL NETWORK

S.Saravanan
Department of ECE
K.S.R. College of Technology
Tiruchengode-637215, India.
saravanan_nivi@yahoo.com, 91-4288-274741

M.Madheswaran
Centre for Advanced Research
Department of ECE,
Muthayammal Engineering College,
Rasipuram - 647408, India
madheswaran.dr@gmail.com, 91-4287-226737

Abstract

This paper explores the implementation approaches of a low power Modified Booth Multiplier (MBM) with Reduced Spurious Transition Activity Technique (RSTAT) and its application on a low power (LP) neural network. This RSTAT approach has been applied on both the compression tree of multipliers and the modified Booth Encoder to enlarge the power clampdown, for high speed and low power purposes. To filter out the spurious switching power of the multiplier, there are two approaches, one is using registers and using AND gates, to assert the data signals of LP multipliers after the data transition has been proposed. The RSTAT approach leads to a 40% power consumption reduction and speed improvement when compared with the other power minimization technique. An artificial neural network is a system consisting of small processing units (called neurons) that perform specific tasks in parallel. The hardware implementation of such neural network will mainly consist of a multiplier circuit for the product term along with an adder circuit for the summation. The above low power multiplier can be used in the neural network for low power VLSI implementations.

Keywords - low power, modified booth multiplier, RSTAT, bit-pair recoding, neural network.

I. Introduction

Low-power and low-energy VLSI circuits have become an important issue in today's consumer electronics. The number of embedded devices that must run with battery power or parasitic power continues to grow. The traditional approaches for designing these systems vary according to the need of low power design. Enhancing the processing performance and reduce the power consumption of the circuit designs are undoubtedly having the challenges in low power VLSI Design. The addition of multiply capabilities to processor architecture can provide significant boost in performance for low power wireless multimedia and Digital Signal Processor (DSP) applications such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), quantization, and neural networks. It is well known that the clamp down approach of dynamic power which is the major part of total power dissipation may provide significant reduction in power consumption. This can be achieved by minimizing the transition capacitance.

The reduction of dynamic power consumption by minimizing the switched capacitance has been reported by many researchers [1-7]. Choi et al [1] proposed 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. The reported results show that the PGC can reduce power consumption by 10% to 44% in an array multiplier with 30% to 36% area overhead in speech related applications.

A 32-bit 2's complement adder equipping a dynamic-range determination (DRD) unit and a sign-extension unit was reported by Chen et al [2]. This design tends to reduce the power dissipation of conventional adders for multimedia applications. Later Chen et al [3] presented a multiplier using the DRD unit to select the input operand with a smaller effective dynamic range to yield the Booth codes and it saves 30% power dissipation than conventional ones. Benini et al [4] reported that, the technique for glitching power minimization by replacing some existing gates with functionally equivalent ones that can be "frozen" by asserting a control signal. This saves 6.3% of total power dissipation since it operates in the layout level environment which is tightly restricted. The double-switch circuit-block switch scheme capable of reducing power dissipation during down time by shortening the settling time after reactivation was proposed by Henzler [5]. Huang et [6] also presented the arithmetic details about the signal gating schemes and illustrates 10% to 45% power reduction for adders. The combination of the signal flow optimization (SFO), left-to-right leapfrog (LRLF) structure, and upper/lower split structure was incorporated in the design to optimize the array multipliers by Huang [8] and it is reported that the new approach can save about 20% power dissipation. Wen et al [9] reported that the turning off some columns in the multiplier array whenever their outputs are known can save 10% power consumption for random inputs. In order to improve the performance, the

architecture with accelerating multiplication is expected.

II. Modified Booth Multiplier

Consider a Modified Booth Multiplication with two numbers "2AC9" and "006A". The recoded output 0000+2-1-1-2 has been considered as new multiplier and this will reduce the multiplication process into half of its original step, this lead to the reduction of power consumption by half. The modified booth multiplied partial products are represented by PP0, PP1, PP2, PP3…etc as shown in the below Fig. 1.

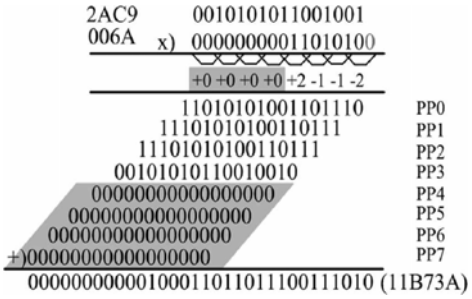

Fig. 1. Illustration of multiplication using Modified Booth Encoding.

The PP candidates are added using a RSTAT equipped adder to get the final result. When an application such as Fast Fourier Transforms (FFT), Discrete Cosine Transform (DCT), quantization, and filtering are concern multiplication is 16 bit and the speed of operation depends upon the processor. For DSP applications the input data is sophisticated, at that time the multiplication process is split down to most significant part (MSP) and least significant part (LSP). Then the operation has been carried out with the help of some latch to switch down some unwanted signals and transients power to get perfect output.

III. Reduced Spurious Transition Activity Technique

The implementation approaches of Reduced Spurious Transition Activity Technique (RSTAT) design concept is described in this section. This RSTAT approach can be applied on both the compression tree of multipliers and the Modified Booth Encoder to enlarge the transition power reduction. To illustrate the influence of the spurious power signal transitions, two cases of a 16-bit addition are explored as an example shown in Fig. 2.

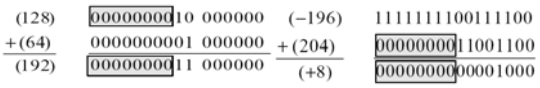

Fig. 2. Spurious power signal transition examples

The first case illustrates a transient state in which the unwanted transitions of carry signals occur in the MSP though the final result of the MSP are unchanged. The second case describes the situations of one negative operand adding another positive operand with carry from LSP. Here the results of the MSP are predictable; therefore the computations in the MSP are useless and can be neglected. Eliminating those spurious computations will not only save the power consumed inside the RSTAT multiplier but also decrease the glitching noises which will affect the next arithmetic circuits. The main contribution of RSTAT is to exploring two implementing approaches and comparing their efficiency to get different material for the reduction of power and to increase the speed of operation. Fig. 3. shows the simplified implementation approach for the addition ⁄ subtraction of two 16 bit numbers.

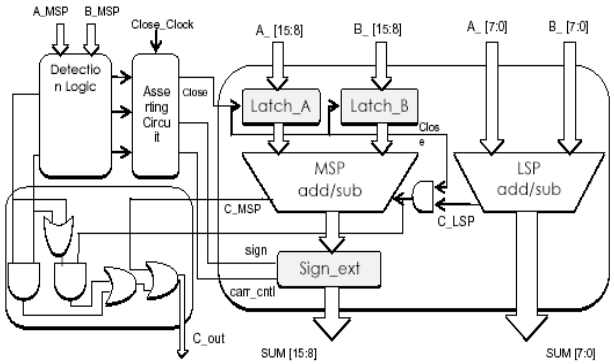

Fig. 3. Low-power adder⁄substractor example based on proposed RSTAT.

VI. Neural Networks

The artificial neural network is a system consisting of small processing units called neurons that perform specific task in parallel. The neurons are arranged in layers, with the output of the neurons in a layer becoming the input to the neurons in the next. Typically the first layer consists of a neuron for each input to the system. These neurons simply hold the input values and pass them on to the next layer. The next layer is called a hidden layer since the user of the network does have access to these neurons. The neurons themselves consist of a set of inputs from the first (input) layer, an implementation of mathematical function of those inputs, then an output that carries the result of the function to the next layer. This next layer can potentially be another layer whose neurons are now functions of the outputs of the previous layer. After the final hidden layer, a last layer, called the output layer, actually supplies the predicted the output values to the user. Some neural networks are capable of designing themselves. These networks are generally called self-organizing, and the process of setting up a network to perform a specific task is

called training. The mathematical functions in the hidden neurons are where the training actually occurs in the networks. Each input must be tested to see its effects on the output of the system. A comparison between the input and output results in a weight or coefficient that is associated with the input value.

One of the most common training methods called back-propagation. In this method, an initial set of weights is assigned to each input. Then each input set from the training data is supplied to the network. This corresponding output is tested and compared to the expected output. A function of the error is used to update the coefficients. After testing all of the input sets from the training data, the network retests all of the data. This continues until the network determines that the coefficients are supplying the best output possible and it does not generate the optimal network architecture. This method can potentially required millions of iterations before the final transfer function are chosen.

A. Group Method of Data Handling

The GMDH algorithm was first presented by Ukrainian engineer and cyberneticist, A.C. Ivakhnenko, and his colleagues in 1968[10]. His intent was to increase a rival method to stochastic approximation. Originally designed to estimate higher order regression polynomials, this heuristic self-organization method has been applied to a large variety of fields including medicine, biology, manufacturing, environmental, ecological systems, psychology and economics. The method builds hierarchical polynomial regression networks to model complex input-output relationships [11].

The GMDH algorithm generates and tests all input-output combinations for a system. Each element of the system implements a function of two inputs. Coefficients of the elements are determined using a regression technique. A threshold is specified at each level to determine if the outputs of the elements in a layer are giving acceptable results. If the result from an element is within the threshold, it is passed on to the next layer. Those elements and variables that are least useful in predicting the proper output are filtered out. Each succeeding layer has more complex combinations. Layers are added until satisfactory results are reached [12-14]. It is almost a "Darwin" model: only those elements that are strong and give desired results are allowed to pass on to the next stage. Using this method, the algorithm chooses the optimal set of input variables, the degree of nonlinearity in the final model, and the structure and the degree of interaction terms in the final model [15]. There are 4 advantages to this method: (1) A small training set is required; (2) The multiple layer structure of the resulting system results in a feasible way of implementing a high degree multinomial; (3) The computational burden is reduced; (4) inputs/functions of inputs that have little impact on the output are automatically filtered out.

As with most neural networks, the first layer of this network would be comprised of a single neuron per input to the system. These neurons sole functions are to pass the input in to the first hidden layer. The hidden layers would have the actual transfer that will attempt to predict the proper output. Since there are many different types of relationships between inputs and the output of a system, the first step to developing a GMDH based network is to narrow down which type of relationship the network will explore. GMDH algorithms often combine linear and non-linear elements in order to better find the input-output relationship of a system. In this algorithm, linear and parabolic (input squared) relationships are chosen. In addition, each neuron will have two inputs. This results in 6 possible terms. The following quadratic polynomial is used as the output equation for the hidden neurons:

$$Y = b_0 + b_1 in_1 + b_2 in_2 + b_3 in_1^2 + b_4 in_2^2 + b_5 in_3 in_2 \quad (1)$$

V. Neural Networks Architecture with Multiplier

The neural network unit consists of different multiplier sections and each section is controlled by RSTAT. The main contribution of RSTAT is to exploring two implementing approaches and comparing their efficiency to get different material for the reduction of power and to increase the speed of operation. From "(1)", and Fig.4, that the hardware implementation of the neural network will mainly consist of a multiplier circuit for the product term along with an addition circuit for the summation term.

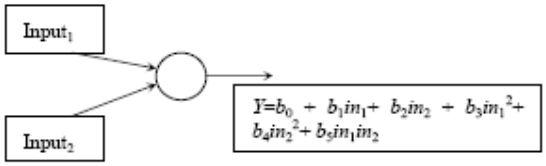

Fig. 4.Diagram of Basic Neuron

VI. Implementation of Low Power Multiplier using Rstat

The computation of a multiplier manipulates two input data to generate many partial products for subsequent addition operations, which in the CMOS circuit design, require many switching activities. Thus, switching activities within the functional units of a multiplier account for the majority of the power dissipation of a multiplier, as given in the following,

$$Ps_{witching} = \alpha\, C\, V_{dd}^{\,2}\, f_{clk}$$

Where α is the switching activity parameter, C is the loading capacitance ,V_{dd} is the operating voltage and f_{clk} is the operating frequency αC can also be viewed as the effective switching capacitance of the

transistors nodes on charging and discharging. Therefore, minimizing switching activities can effectively reduce power dissipation without impacting the circuit's operational performance. The low power multiplier is designed by equipping the RSTAT on a tree multiplier unit. There are two distinguishing design considerations in designing the proposed multiplier and RSTAT, as listed in the following.

A Applying the RSTAT on the MBE

In Fig.1, the shadow denotes that the numbers in this part of Booth multiplication are all zero so that this part of the computations can be neglected. Saving those computations can significantly reduce the power consumption caused by the transient signals. According to the analysis of the multiplication the RSTAT equipped Modified Booth Encoder (MBE); this is controlled by a detection unit. The detection unit has one of the two operands as its input to decide whether the Booth encoder calculates redundant computations. The proposed RSTAT equipped multiplier is illustrated in Fig.5.

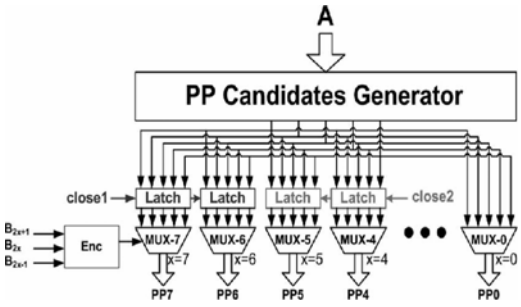

Fig. 5. RSTAT equipped MBE

The PP generator generates the candidates of the partial products, which are then selected according to the Booth encoding results of the operand B. Moreover, when the operand besides the Booth encoded one has a small absolute value, there are opportunities to reduce the spurious switching power dissipated in the compression tree. According to the redundancy analysis of the additions, we replace some of the adders in compression tree of the multiplier with the RSTAT equipped adders, which are marked with oblique lines in Fig. 6.

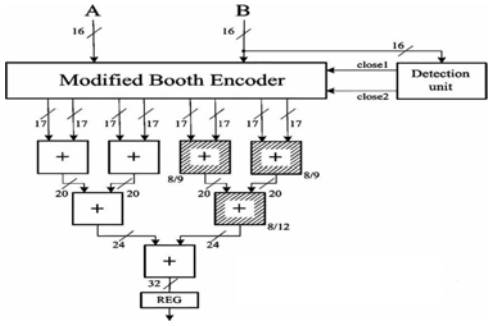

Fig. 6. RSTAT adder

B. Detection Logic Circuit Design

A detection logic circuit design using register is as shown in the Fig. 7. The shadow section indicates the registers used in this circuit for to control the latch, carry and sign extension. Again to clampdown the power consumption, the register is suppressed in to an AND gate as shown in Fig. 8, to control the signal assertion. When speed is seriously concerned, this implementing

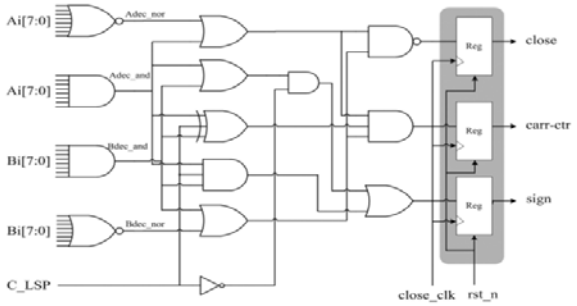

Fig. 7. Detection Logic Circuit Using Registers

approach enables better flexibility on adjusting the data asserting time of RSTAT equipped multipliers.

The detection logic circuit is used to detect the zero's present in the input of the multiplier. When we are using this low power multiplier for FFT, IDFT, DFT calculation, before the multiplication process we would identify the zeros in the input to effectively use RSTAT. The output of detection logic circuit consists of three signals namely close, carr-ctr and sign. Close signal is used to control the latch, Carr-ctr is used to control the carry signals and sign is used to maintain the sign of the output.

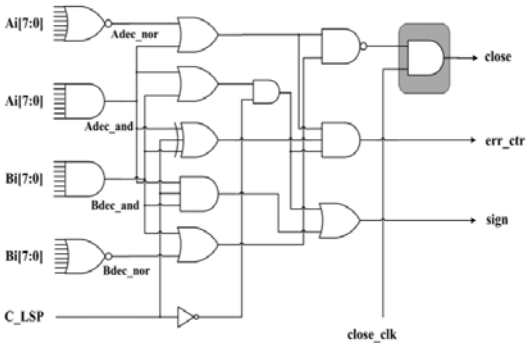

Fig. 8. Detection Logic Circuit Using ANDGate

The three output signals of the detection logic are given a certain amount of delay before they assert, demonstrated in the timing diagram shown in Fig. 9. The delay Φ, used to assert the three output signals, must be in the range of $\psi<\Phi<\Delta$ to filter out the glitching signals as well as to keep the computation results correct. Where ψ represents the transient period, Δ represents earliest required time of all the inputs. The range has been represented as

shadow in the timing diagram. However the restriction that Φ must be greater than ψ to guarantee the registers from latching the wrong values of control signals usually decrease the overall speed of the applied designs.

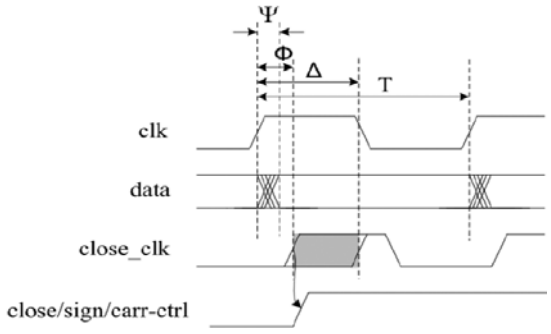

Fig. 9. Timing Diagram of the Control Signals of Detection Logic Circuits After Assertions

VII. Results

Design	Power in (mW)
RSTAT Using register	3.94
RSTAT Using AND gate	2.69

Table 1

Simulation Results

Fig. 10 and Fig. 11 shows the simulation results of detection logic circuit using registers and AND gates respectively.

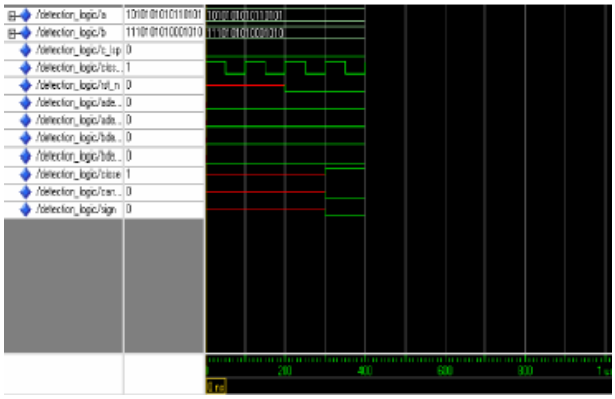

Fig. 10. Simulation Results of Detection Logic Circuit Using Registers

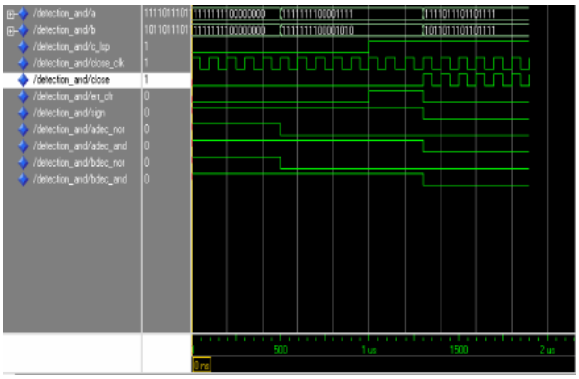

Fig. 11.Simulation Results of Detection Logic Circuit Using AND Gates

VIII. Conclusion

In this paper, multiplier unit adopting the RSTAT approach using register and AND gates in the detection logic unit is presented. The simulation results show that the power reduction of the new approach, which leads to a 40% switching power reduction when compared with the existing techniques in 0.18µm CMOS technology.

In addition, it can be seen that the performance can also be improved considering the design owns equivalent low power performance but also leads to a higher maximum speed when compared with the other approaches. Moreover, the proposed RSTAT equipped neural network also has better power efficiency when compared with the existing modern neural network unit.

References

[1] J. Choi, J. Jeon, and K. Choi, "Power minimization of functional units by partially guarded computation," in Proc.*IEEE International Symposium Low Power Electron. Devices*, pp. 131–136. 2000.

[2] O. Chen, R. Sheen, and S. Wang, "A low power adder operating on effective dynamic data ranges," *IEEE Transaction. Very Large Scale Integration. (VLSI) Syst.*, vol. 10, no.4, pp. 435–453, Aug. 2002.

[3] O. Chen, S. Wang, and Y. W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Transaction. Very Large Scale Integration. (VLSI) Syst.*, vol. 11, no.3, pp. 418–433, Jun. 2003.

[4] L. Benini, G. D. Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Glitching power minimization by selective gate freezing," *IEEE Tran. Very Large Scale Integration. (VLSI) Syst.*, vol. 8, no. 3, pp. 287–297, June 2000.

[5] S. Henzler, G. Georgakos, J. Berthold, and D.Schmitt-Landsiedel, "Fast power-efficient circuit-block switch off scheme," *Electronics Letter.* vol. 40, no. 2, pp. 103–104, Jan. 2004.

[6] Huang and M. D. Ercegovac, "On signal gating schemes for low power adders," in Proc. 35th *Asilomar Conference. Signal, Systems. Computer.* 2003.

[7]Z. Huang, "High-level optimization techniques for low-power multiplier design," *Ph.D. dissertation*, Department of Computer Science., Univ. California, Los Angeles, 2003.

[8] Z. Huang and M. D. Ercegovac, "High performance low power left-to- right array multiplier design," *IEEE Transaction on Computer.*, vol. 54, no. 3, pp. 272-283,March 2005.

[9] M. C. Wen, S. J. Wang and Y. N. Lin. "Low-power parallel multiplier with column by passing," *Electron. Lett*. vol. 41, no. 12, pp. 581–583, May 2005.

[10] A.G.Ivakhnenko, "heuristic self-organization in problems of engineering cybernetics," *Automatica*, 6(2), 1970.

[11] M.C.Acock, Y.A.Pachepsky, "estimating missing weather data for agricultural Simulations using group method of data handling," *Journal of Applied meteorology*, 39(2), pp.1176-1184, 2000.

[12] T.Kondo, A.S.pandya, J.M.Zurada, "logistic GMDH-type notification neural networks and their application to the identification of the X-ray film characteristic curve, "proc. of *IEEE international Conference on System, Man, and Cybernetics*, pp. 437-442, 1999.

[13] T.Kondo, A.S.Pandya and H.Nagashino, "GMDH-type neural network algorithm with a feedback loop for structural identification of RBF neural network, "*International Journal of Knowledge-Based Intelligent Engineering systems*, 11(7), pp.157-168, 2007.

[14] T.Kondo, J.Ueno and, A.S.Pandya, "Multi-layered GMDH-type neural networks with radial basis functions and their application to the 3-dimentional medical image recognition of the liver," *Journal of Advanced Computational Intelligence*, 11(7), pp. 157-168, 2007.

[15] T.Konda, A.S.Pandya, "GMDH-type neural network algorithm with sigmoid functions," *International Journal of Knowledge-Based Intelligent Engineering systems*, 7(4), pp.198-205, 2003.