$$S = \sum_{i=0}^{k-1} \mathrm{FIR}\{(P_I \oplus V_i)\} \quad (5)$$

where FIR denotes finite impulse response filtering to limit the bandwidth of the transmitting signals, and ⊕ denotes the modulo-2 adder, which is implemented with a binary exclusive-OR gate. Using the definitions of eqn. 2 and the linearity of each operation in eqn. 5, we can rewrite eqn. 5 as

$$\begin{aligned} S &= \sum_{i=0}^{k-1} \mathrm{FIR}\{P_I + V_i - 2P_IV_i\} \\ &= \mathrm{FIR}\left\{\mathrm{EXOR}\left(P_I, \sum_{i=0}^{k-1} V_i\right)\right\} \end{aligned} \quad (6)$$

where EXOR, whose first argument has a binary logic value and second argument has a multilevel logic value, denotes the binary-multilevel exclusive-OR logic operation that is given by letting any one of its two input logics in eqn. 2 be binary. This binary-multilevel exclusive-OR logic operator can be easily implemented as shown in Fig. 2. In this Figure, the (radix-1)'s complement means the multilevel NOT operator defined in eqn. 2. The switch that can be implemented with the 2-to-1 multiplexer indicates the upper line when the binary input P_I is the logic **1**, and the lower line when P_I is the logic **0**.

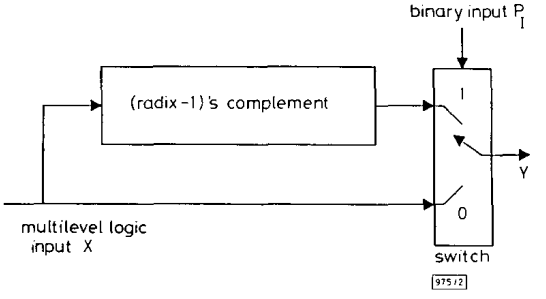

Fig. 2 *Binary-multilevel exclusive-OR logic operator*

Using this binary-multilevel exclusive-OR operator, we can modify the circuit shown in Fig. 1*a* into that shown in Fig. 1*b* because eqns. 5 and 6 always produce the same results. Comparing Fig. 1*a* with Fig. 1*b*, Fig. 1*a* contains k FIR filters, k modulo-2 adders and one arithmetic adder, whereas Fig. 1*b* contains only one FIR filter, one binary-multilevel exclusive-OR operator and one arithmetic adder. Furthermore, each input of the arithmetic adder in Fig. 1*a* must be represented with multiple bits, to maintain the resolution given by the FIR filter coefficients, whereas those in Fig. 1*b* require just 1 bit resolution because they are binary logic operations. For example, if each FIR filter outputs 8 bit filtered data, the arithmetic adder in Fig. 1*b* requires k 1 bit additions instead of k 8 bit additions in Fig. 1*a*. Therefore, it is possible to combine digitally at lower speed in the modified circuit than in the given circuit. Thus, Fig. 1 shows that it becomes easier and simpler to implement Fig. 1*b* than Fig. 1*a* because it has less components and requires lower speed processing.

Fig. 1 shows an example for design efficiency of the binary-multilevel exclusive-OR operation that is a special case of multilevel exclusive-OR of the MLOs. We can also get similar results for many other applications of AND, OR and NOT operations. We can apply this MLO concept to design in the base station modulator for the DS/CDMA digital cellular communications system to reduce the number of spreader and FIR filters <1% and Walsh covering 50% when compared to the conventional case in [4].

Conclusions: We have newly defined some MLOs that are extended from the conventional binary logic operations. Multilevel logic occurs in many digital logic circuits. Since it is possible to manipulate directly the multilevel logic by using the MLOs instead of the binary logic operations, it has many advantages for the digital circuits and VLSI design. It makes digital circuits and VLSI design easy and simple, and also makes the processing rate and power consumption low because it has fewer components and a lower data rate compared to binary logic.

This concept, extended from the binary logic operations, will be useful in designing digital circuits and VLSI when the condition of eqn. 3 is satisfied.

3 July 1995
Electronics Letters Online No: 19951110

Jin-up Kim (*Mobile Telecommunications Division, Electronics and Telecommunications Research Institute, PO Box 106, Yusong, Taejon 305-600, Korea*)

Jae-kyoon Kim (*Department of Electrics and Electronics Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-dong, Yusong-gu, Taejon 305-701, Korea*)

E-mail: jukim@dcnpo.etri.re.kr

References

1 MANO, M.M.: 'Computer system architecture' (Prentice-Hall, NJ, USA, 1982)
2 HURST, S.L.: 'Multiple-valued logic - Its status and its future', *IEEE Trans.*, 1984, **C-33**, (12), pp. 1160–1179
3 TIA/EIA/IS-95, 'Mobile station – base station compatibility standard for dual-mode wideband spread spectrum cellular system', Washington, DC, USA, July 1993
4 KIM, J.U., and KIM, J.K.: 'A new efficient construction of base station modulator for DS/CDMA digital cellular communications systems using the multilevel logic operations', submitted to *IEEE Pers. Commun.*, 1995

Number of nonlinear regular s-boxes

A.M. Youssef and S.E. Tavares

Indexing terms: Cryptography, Combinatorial mathematics

Nonlinearity is a crucial requirement for the substitution boxes in secure block ciphers. In the Letter, the probability of linearity in any nonzero linear combination of the output co-ordinates of a randomly selected regular substitution box is calculated.

Introduction: Gordon and Retkin [1] calculated the probability that any of the output co-ordinates of a random reversible substitution box (i.e. a permutation) are affine functions. After both differential cryptanalysis [2] and linear cryptanalysis [3] were introduced, it was realised that the cryptographic strength of a multi-output function depends not only on the strength of its individual output co-ordinates but also on the strength of every nonzero linear combination of these co-ordinates [4].

One requirement in substitution-box (s-box) design is to have a regular s-box (also known as a balanced s-box). This means that each output symbol should appear an equal number of times when the input is varied over all possible values.

We calculate the probability that any nonzero linear combination of the output co-ordinates of a regular s-box is an affine function. We enumerate the number of ways in which we can construct a regular $n \times m$ s-box (described by the multi-output Boolean function $f(X) : Z_2^n \rightarrow Z_2^m$, $n \geq m$) for which the first k output functions are affine.

By noting that every nonzero linear combination of the output co-ordinates of regular s-boxes is a balanced function, the first function can be chosen in $(2^{n-1} - 2)$ ways, which is the total number of balanced affine functions. The second function can be chosen from the set of balanced affine functions not including the first one or its complement, i.e. in $(2^{n-1} - 4)$ ways. The third one can be chosen from the set of balanced affine functions not including any linear combination of the first two functions. Proceeding as above, the first k output functions can be chosen in $2^k \prod_{i=0}^{k-1} (2^n - 2^i)$ ways. Since we can partition the input X into 2^k distinct sets, all Xs in a given set are assigned the same common value of these k bits. We still need to assign the remaining $n - k$ output bits for each X. Each X within a given set must be assigned a distinct $(n - k)$-tuple of the remaining output bits for 2^{n-m} times. Each set can be assigned in

$$\frac{2^{n-k}!}{(2^{n-m}!)2^{m-k}}$$

ways, so the remaining bits can be assigned in

$$\left(\frac{2^{n-k}!}{(2^{n-m}!)2^{m-k}}\right)^{2^k}$$

ways. Thus the number of regular s-boxes in which the first k output co-ordinates are linear functions is given by

$$R(n,m,k) = 2^k \left(\frac{2^{n-k}!}{(2^{n-m}!)2^{m-k}}\right)^{2^k} \prod_{i=0}^{k-1}(2^n - 2^i)$$

Consider the function Φ: $Z_2^n \rightarrow Z_2^{2m-1}$ constructed from every nonzero linear combination of the output co-ordinates of f. Counting the number of nonlinear regular s-boxes corresponds to counting the number of functions Φ with no affine co-ordinates.

The number of ways we can choose l co-ordinates of Φ such that k of them are linearly independent is equivalent to the number of $l \times m$ binary matrices (without taking the order of rows into account, i.e. two matrices with the same rows but in different orders are counted once) with nonzero distinct rows which have rank k. This is given by [4, 5]

$$LI(m,l,k) = \binom{m}{k}_2 \sum_{j=0}^{k}(-1)^j 2^{\binom{j}{2}} \binom{2^{(k-j)}-1}{l} \binom{k}{j}_2$$

where

$$\binom{m}{k}_2 = \begin{cases} 1 & k = 0 \\ \dfrac{\prod_{i=0}^{k-1}(2^m - 2^i)}{\prod_{i=0}^{k-1}(2^k - 2^i)} & n \geq k > 0. \end{cases}$$

It is clear that for every k linearly independent co-ordinates of Φ, denoted by $(\phi i_1, \phi i_2, ..., \phi i_k)$, we can find $(m-k)$ co-ordinates of Φ, denoted by $(\phi i_{k+1}, \phi i_{k+2}, ..., \phi i_m)$, such that

$$(\phi_{i_1} \ \phi_{i_2} \ .. \ .. \ \phi_{i_m})^t = A(f_1 \ f_2 \ .. \ .. \ f_m)^t$$

where A is an $m \times m$ invertible binary matrix and $(f_1 f_2 .. .. f_m)$ denotes the output co-ordinates of f. This means that as f varies over all the set of distinct regular s-boxes, $(\phi i_1, \phi i_2 .. .. \phi i_m)$ scans the whole set but in a different order. From the above argument, it is clear that the number of ways of constructing certain k linearly independent co-ordinates of Φ from affine functions is also given by $R(n, m, k)$.

Using the inclusion-execlusion principle, the number of linear regular s-boxes, i.e. regular s-boxes with the property that one or more of the nonzero linear combinations of their output co-ordinates are affine, is given by

$$RL(n,m) = \sum_{l=1}^{2^m-1}(-1)^{l-1} \sum_{k=1}^{min(l,m)} LI(m,l,k)\, R(n,m,k).$$

To express the above count as a fraction of the total number of regular s-boxes, denoted by $FRL(n, m)$, we divide by the total number of $n \times m$ regular s-boxes

$$\frac{2^n!}{(2^{n-m}!)2^m} \qquad n \geq m$$

To give a numerical example, for $n = 6$ and $m = 4$, which is the size of DES s-boxes, $FRL(6,4) = 2.46 \times 10^{-16}$. We can easily get an upper bound for $FRL(n, m)$ by noting that

$$RL(n,m) < (2^m - 1)\, R(n,m,1)$$

and hence that

$$FRL(n,m) < \frac{2(2^n-1)(2^m-1)(2^{n-1}!)^2}{2^n!} = O\left(\frac{2^{5n/2}}{2^{2^n}}\right).$$

Conclusion: We have derived an exact expression for the number of regular s-boxes with the property that one or more of the nonzero linear combinations of their output co-ordinates are affine. From the above, it is clear that this fraction decreases dramatically with the number of inputs.

15 June 1995
Electronics Letters Online No: 19951086

A.M. Youssef and S.E. Tavares (*Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, K7L 3N6, Canada*)

S.E. Tavares: corresponding author

E-mail: tavares@ee.queensu.ca

References

1 GORDON, J., and RETKIN, H.: 'Are big S-boxes best?'. Lecture Notes in Computer Science: Proc. Workshop on Cryptography, (Springer-Verlag, 1982), pp. 257–262
2 BIHAM, E., and SHAMIR, A.: 'Differential cryptanalysis of DES-like cryptosystems'. Advances in Cryptology: Proc. Crypto '90, (Springer-Verlag, 1991), pp. 1–21
3 MATSUI, M.: 'Linear cryptanalysis method for DES cipher'. Advances in Cryptology: Proc. Eurocrypt '93, 1994, (Springer-Verlag), pp. 366–397
4 NYBERG, K.: 'Perfect nonlinear S-boxes', Advances in Cryptology: Proc. Eurocrypt '91, (Springer-Verlag, 1992), pp. 378–386
5 STRONG, R.: Private communication
6 GOLDMAN, J., and ROTA, G-C: 'On the foundation of combinatorial theory IV. Finite vector spaces and Eulerian generating functions', *Stud. Appl. Math.*, 1970, **XLIX**, (3), pp. 239–258

Reduced-complexity circuit for neural networks

S.S. Watkins and P.M. Chau

Indexing terms: Neural networks, Reduced instruction set computing

The Letter demonstrates that a 10 bit reduced-complexity VLSI circuit can be used in place of a 32 bit floating-point processor to speed up some neural network applications, reducing circuit area and power consumption by 88% with a negligible increase in RMS error. Applications were executed on a radial basis function neurocomputer using the reduced-complexity circuit implemented with FPGA technology. One application produced better results than had been previously obtained for a NASA data set using either neural network or non-neural network approaches.

Introduction: Today's hardware capabilities are limiting the development of neural network research. Neural networks learn by adjusting weights on input and internal signals by very small increments until the network has converged on a solution that is satisfactory for all training patterns, and this process can take days, weeks or months on a modern workstation. A neural network usually exhibits a significant amount of potential parallelism, and hardware accelerators can reduce the learning time by orders of magnitude by exploiting this parallelism. Many applications require less than 32 bits of floating-point precision [1], and this fact can be used to reduce the cost of the accelerator circuits in terms of area and power. For one application described below, the use of a unique 10 bit reduced-complexity multiply/accumulate circuit resulted in an area and power saving of 88% over a full 32 bit floating-point circuit, while the learning results as measured by RMS error were within 0.03% of the 32 bit results.

Radial basis function neural networks: Our research has focused on implementing radial basis function (RBF) neural networks with reduced-complexity VLSI circuits as a means to accelerate learning while minimising costs in terms of area and power. The RBF network uses a radial basis function (usually a Gaussian) as the transfer function of a neuron rather than the traditional sigmoid function. Radial basis functions have been used to solve mapping and function estimation problems with positive results [2]. The equations describing an RBF neuron's output x_j in terms of an input vector and stored weights are:

$$z_j = \sum_k (C_{jk} - I_k)^2 \qquad (1)$$