PROTECTING DSP CIRCUITS THROUGH OBFUSCATION VIA HIGH-LEVEL TRANSFORMATIONS

P.BujjiBabu[1], S.Pragathi[2]
[1]Asst. Professor, ECE Department, [2]PG scholar in VLSI Design,

Abstract: Obfuscation is a technique that makes comprehending adesign difficult and hides the secrets in the design. Obfuscation is called best-possible if the obfuscated design leaks no more information thanany other design of the same function. In this paper, we prove that any best-possible obfuscation of a sequential circuit can be accomplished by asequence of four operations: retiming, resynthesis, sweep, and conditionalstuttering. Based on this fundamental result, we also develop a key-based obfuscation scheme to protect design Intellectual Properties (IPs) againstpiracy. The novel obfuscation method embeds a secret key in the powerup state of IC, which is only known by the IP rights owner. Without the key, the IC still functions but its efficiency will be much degraded. Unlike existing IC metering techniques, the secret key in our approach is implicitthus it can also be used as a hidden watermark. Potential attacks andthe countermeasures are thoroughly examined, and experimental resultsdemonstrate the effectiveness of the method.

Index Terms: Digital signal processing (DSP), functional obfuscation, hardware security, high-level transformations, intellectual property (IP) protection, obfuscation, reconfigurabledesign, and structural obfuscation.

I. INTRODUCTION

The business model of semiconductor industry has changed significantly in last decade. With increasing complexity and cost of modern ICs, a design house has to seek the aid from various externalagencies, such as EDA companies, IP vendors, library providers, and fabrication foundries. The active participation of external entities inthe design and manufacturing flow has produced numerous hardware security issues. Among all the hardware security problems, piracy islikely to be the most ubiquitous and expensive one. Most leading edge design houses have outsourced their fabrication to the offshorefoundries for the sake of lower labor and manufacturing cost. However, many offshore foundries are hard to be trusted since they are incountry without consummate enforcement law for IP protection. A variety of techniques has been proposed for fighting againsthardware piracy. There are two main classes of approaches. Oneapproach is hardware metering, which enables design houses tohave post-fabrication control on the produced ICs. By metering, thedesigner can count the number of fabricated ICs, monitor their usage,and even remotely lock/unlock the ICs. Hardware watermarking, as another popular approach to IP protection, is inspired by thetraditional digital watermarking technique. It inserts certain identityinformation into behavioral specification or sequential structure ofthe design. Watermarking is more passive compared with metering.But since watermarking has a unified signature for all ICs and doesnot involve any designer-manufacturer interaction, it will usually beless expensive. Physical circuits may (and usually do) perform not exactlyaccording to the design because of several reasons such as tolerance of circuit elements, environmental effects(temperature, humidity etc.) and aging. The design processshould include an analysis of the effects of the parametersvariation on the overall circuit response. The transient sensitivity of nonlinear circuits is a challenging problem as itmay require substantial amount of storage and CPU time alsoon modern computers. The adjoint based approach is aclassical method for sensitivity analysis of linear and nonlinear circuits; as known two system analyses are sufficient to obtainthe sensitivities of the response of a circuit regardless thenumber of design parameters. The problem of hardware security is a serious concern thathas led to a lot of work on hardware prevention of piracyand intellectual property (IP), which can be broadly classified into two main categories: 1) authentication-based approachand 2) obfuscation-based approach. The authenticationbased approaches include physical unclonable functions(PUFs)-based authentication, digital water marking, key-locking schem, and hardware metering. Thefocus of this paper is on obfuscation, which is a techniquethat transforms an application or a design into one that is functionally equivalent to the original but is significantlymore difficult to reverse engineer. Some hardware protectionmethods are achieved by altering the human readability of the hardware description language (HDL) code, or by encryptingthe source code based on cryptographic techniques.Recently, a number of hardware obfuscation schemes have been proposed that modify the finite-state machine (FSM)representations to obfuscate the circuits. The key contribution of this paper is a novel approach todesign obfuscated DSP circuits by high-level transformationsduring the design stage. The DSP circuits are obfuscated by introducing an FSM whose state is controlled by a key. TheFSM enables a reconfigurator that configures the functionalitymode of the DSP circuit. High-level transformations lead to many equivalent circuits and all these create ambiguityin the structural level. High-level transformations also allowdesign of circuits using same datapath but different controlcircuits. Different variation modes can be inserted into the DSP circuits for obfuscation. While some modes generateoutputs that are functionally incorrect, these may representcorrect outputs under different situations, since the output is meaningful from a signal processing point of view. Othermodes would lead to nonmeaningful outputs. The initializationkey and the configure data must be known for the circuit towork properly. Consequently, the proposed design methodology leads to a DSP circuit that is both structurally andfunctionally obfuscated.

II. RELATED WORK

Most popular traditional approaches include: (a) FSMwatermarking based on Unused Transitions: the authors in [18]introduced the first IP protection using FSM watermarking. The algorithm is based on extracting the unused transitions in a statetransition graph (STG) of the behavioral model. In their solution,extra transitions are added to satisfy the design oals. (b) FSMwatermarking by Property Implanting: the author in [13] tried tomanipulate the STG of the finite state machine to implant the watermark as a property. The property was topological in natureand was defined in terms of visited states (s → s → ⋯ → s). Inorder to define the topological property, the author added extra states and state transitions in a systematic way to satisfy a specifictopological requirement. (c) FSM watermarking by Integration ofTwo Distinct FSMs: the authors in [6] designed a completely newFSM as a watermark and then the watermark FSM was combinedwith the original FSM to create an integrated composite FSM.Constructing a new watermark FSM was done by adding new states and transitions. More recently, a FSM watermarking scheme by making theauthorship information a non-redundant property of the FSM wasproposed in [3]. In this work, the watermark bits were added intothe outputs of the existing and free transitions of STG. Anothermethod was proposed in [11]. In this work, a set of edges were added as a dummy entity. This was done by assigning state encoding values. The new edges created by this method werepaired with an unused state input combination, and the output wasspecified as a don’t-care condition. Despite these popular methods which can be effective inprotecting IPs of FSMs as demonstrated in these works, theseapproaches are fundamentally based on expanding the original FSM to an enlarged FSM with new states and/or state transitions.

III. EXISTING METHOD

As this paper is the first attempt to develop a methodology to obfuscate DSP circuits by utilizing high-eveltransformations, it is hard to compare with other existingobfuscation methods which are general to a wide variety of designs. Therefore, we have introduced two metricsto analyze the security, Most of the hardware obfuscation techniques in this paper can also be applied to DSPcircuits. However, the use of high-level transformationsfrom a security perspective has not been incorporatedinto any of these prior hardware obfuscation techniques. In addition, other circuit locking techniques only achieveprotection at one-level (i.e., encrypt the normal functionality by a key), while our proposed methodology provides a two-level protection (i.e., structural obfuscationand functional obfuscation). The main advantage of the proposed methodology is the generation of meaningful variation modes from a signalprocessing point of view, since the meaningful modescreate ambiguity to the adversary such that it is hard forthe adversary to distinguish the desired functionalityfrom other variation modes. Other existing methods, such as [6], [7], are not specific to DSP circuits, which wouldnot be able to ensure meaningful variation modes from asignal processing point of view. In addition, meaningfulvariation modes enable our proposed design methodology to be adaptable to reconfigurable applications. Finally,when considering the metrics of the design performance,our proposed methodology is also superior. While ourproposed approach only alters the logic of switches, mostof the existing methods are based on explicit FSM modifications (e.g., the technique proposed in [13]), which are not scalable since the construction of the FSM is not practical for even moderate-sized circuits, not to mention thatthe number of added obfuscation states can be relativelylarge as compared with the original FSM. In our proposedmethodology, area consumption is slightly increased dueto the increased cost of the control logic for the obfuscated switches.

IV. HIDING FUNCTIONALITY BY HIGH-LEVEL TRANSFORMATIONS

High-level transformations have been known for a long time and have been used in a wide range of applications, such as pipelining, interleaving, folding, unfolding, and look-ahead transformations (e.g., quantizer loops, multiplexer loops, relaxed lookahead, annihilation reordering look-ahead), and havebeen used in synthesis of DSP systems. These techniques can be applied at the algorithm or the architecture level to achieve a tradeoff among different metrics of performance, such as area, speed, and power [25]. However, the use ofhigh-level transformations from a security perspective hasnot been studied before. High-level transformations alter thestructure of a DSP circuit, while maintaining the original functionality. These transformations may lead to architectureswhose functionalities are not obvious. Take an extreme case,for example, many filters can be folded into one multiplyaccumulator (MAC), but their functionalities are not the same.In other words, one MAC with proper switches can implementmany different digital filters. Therefore, we can conclude that high-level transformations naturally provide a means to obfuscate DSP circuits both structurally and functionally. Structuralobfuscation and functional obfuscation are defined as follows.

1) Structural Obfuscation: Any algorithm can be implemented by a family of architectures by using high-leveltransformations. These architectures enable structuralobfuscation where the functionalities of the algorithmscan be hidden. This can be considered as a passive modelfrom attacker’s perspective.

2) Functional Obfuscation: This is realized by encryptingthe normal functionality of a DSP circuit with one ormore sets of keys. The DSP circuit cannot functioncorrectly without the keys. This corresponds to an activemodel from attacker’s perspective.

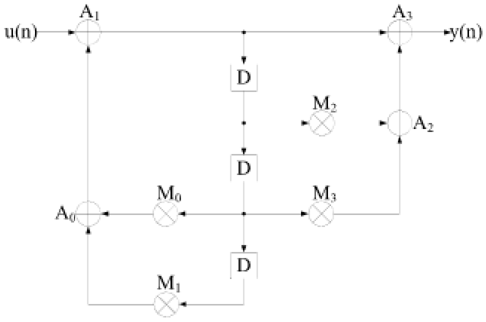

Fig. 1. Third-order IIR filter

Folding is such an example of high-level transformation,which could be utilized to achieve design obfuscation. Thefolding transformation generates folded variants based on the folding set, which is the reverse of the unfolding transformation [18]. The choice of folding set is critical to theperformance of the folded structure, since an appropriatechoice of folding order can lead to an architecture with lowerarea and power. Folding sets can be designed intuitively to meet the performance requirements or can be obtained from a high-level synthesis system. The details and otherexamples (e.g., interleaving) of how to hide the functionalitiesof DSP circuits by high-level transformations are described. We can observe that: 1) circuits with different functionalities can have a similar structure, and circuits withthe same functionality may have very different structures;2) structural obfuscation can be achieved by high-level transformations; and 3) if the switch instances are invisible to the adversary, then the DSP systems will be harder to reverseengineer, since the functionality of a DSP circuit is not obviousdue to obfuscation achieved by high-level transformations.As a result, the adversary who only has knowledge of the structural information but lacks knowledge of the switch instancescannot easily discover the functionality of a DSP circuit. As an example, we consider a third-order IIR digital filter given by transfer function H(z) = (1 + m2z−1+m3z−2)/(1 − m0z−2 − m1z−3), as shown in Fig. 1. The coefficients mi correspond to the multiplication Mi. We assume the availability of one 1-stage pipelined adder and one 3-stagepipelined multiplier. The filter is folded with folding factorN = 4 using the following folding sets:

$$M = \{M_0, M_1, M_2, M_3\}$$
$$A = \{A_0, A_1, A_2, A_3\}.$$

Folding sets represent the order of operations executed bythe same hardware. For a folded system to be realizable, the folding equations, *DF(U →* e *V) = Nw(e) − PU + v − u*,must be greater or equal to 0 for all the edges in the diagram,where *N* is the folding factor, *w(e)* is the number of delays from *U* to *V* , *PU* represents the pipelining level of hardwarefunctional unit for operation type *U*, and *u* and *v* representthe folding orders of *U* and *V* , respectively. Retiming andpipelining can be used to satisfy this property (or it canbe determined that the folding sets are not feasible), as a preprocessing step prior to folding. The folded architecture isshown in Fig. 2. Fig. 3 presents the structure that the switchinstances are designed to be invisible. Null operations areincorporated into the switches.

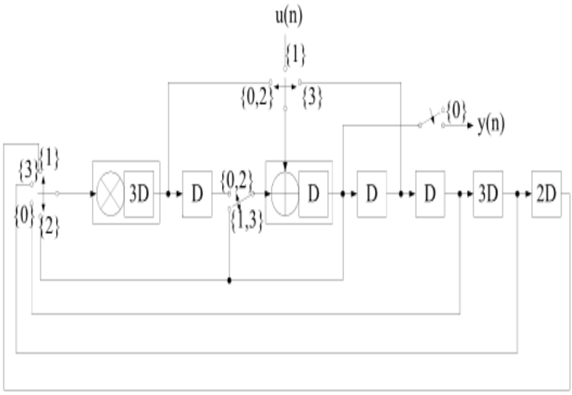

Fig. 2. Folded structure of the third-order IIR filter in Fig. 1. The switch instance *i* corresponds to clock cycle 4*l* + *i*.

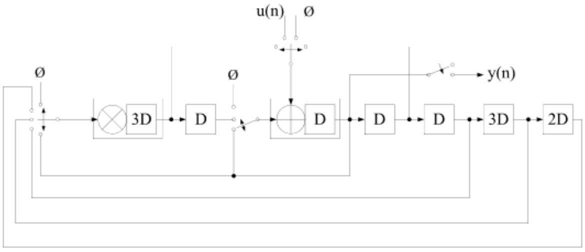

Fig. 3. Folded structure of the third-order IIR filter with invisible switches.

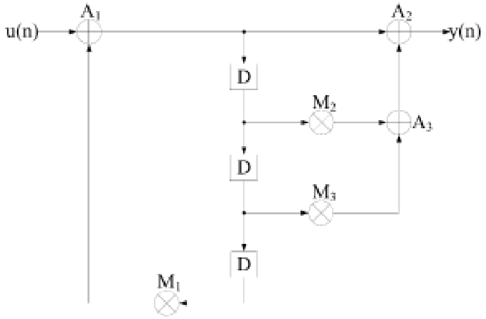

Fig. 4. Another third-order IIR filter

We consider the implementation of another third-order IIR filter given by transfer function *H(z) = (*1 + *m*2*z*−1+*m*3*z*−2*)/(*1 − *m*1*z*−3*)* as shown in Fig. 4.In order to achieve obfuscation, architecture can be designed to be configurable as a third-order IIR filter shown in either Fig. 1 or 4. These two modes are considered as meaningful modes. In fact, a folded architecture of Fig. 4 using the following folding sets can be obtained by assigning different switchinstances to the structure in Fig. 3, which is shown in Fig. 5.

$$M = \{\emptyset, M_1, M_2, M_3\}$$
$$A = \{\emptyset, A_1, A_2, A_3\}.$$

The folding factor is 4, while there are only three multipliersand three adders in the DSP circuit. Therefore, if we considerthe functionality of Fig. 4 as the desired mode, one computation cycle is wasted every four cycles. The latency will alsobe increased. Note that we could use clock gating techniquesto reduce the power for the null operation cycles.However, we can extend the periodicity of the switches toovercome the hardware underutilization. For instance, we can fold the third-order IIR filter in Fig. 4 by folding factor 3 withthe folding sets

$M = \{M3, M1, M2\}$
$A = \{A2, A1, A3\}$.

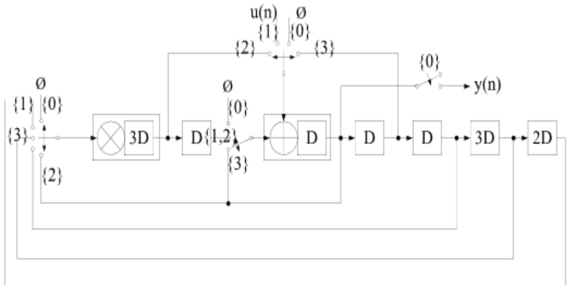

Fig. 5. Folded structure of the third-order IIR filter in Fig. 4. The switchinstance *i* corresponds to clock cycle 4*l* + *i*.

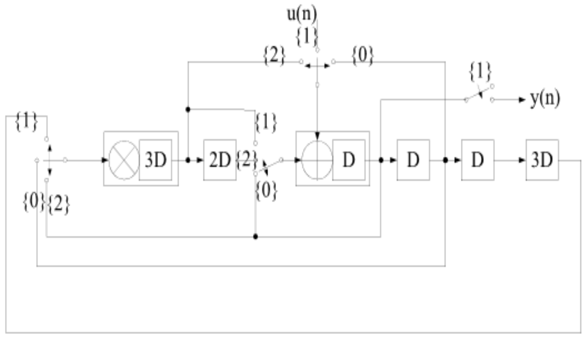

Fig. 6. Another folded structure of the third-order IIR filter in Fig. 4. The switch instance i corresponds to clock cycle 3l + i.

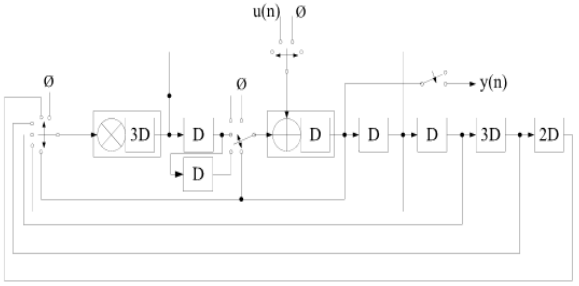

Fig. 7. Obfuscated structure, which can be configurable as athird-order IIR filter shown in either Fig. 1 or 4.

The folded structure is shown in Fig. 6. We can accommodate the two meaningful modes with no increase in latency for the second mode by extending the periodicity of the switches to the least common multiple of the folding factors of thetwo modes [i.e., lcm(3, 4) = 12]. Similar extensions of switchperiods have been considered in design of digit-serial DSParchitectures [28]. For example, switch instance 4*l* + *i* can be rewritten as 12*l* + *i*, 12*l* + 4 + *i*, and 12*l* + 8 + *i*, for *I* ranging from 0 to 3, in Fig. 2; while switch instance 3*l* + *I* can be rewritten as 12*l* + *i*, 12*l* + 3 + *i*, 12*l* + 6 + *i*, and12*l* + 9 + *i*, for *i* ranging from 0 to 2, in Fig. 6. As a result,for each meaningful mode as the desired mode, the latencyremains the same as the original folded structure. This is achieved by increasing the complexity of the switch and theexpense of hardware overhead associated with this step. Thefinal obfuscated architecture for these two meaningful modes is shown in Fig. 7. The switch instances are obfuscated andthe two correct configurations of the switches correspond totwo meaningful modes.

V. CONCLUSION

This paper presents a novel low-overhead solution to design DSP circuits that are obfuscated both structurally and functionally by utilizing high-level transformation techniques. It is shown that verifying the equivalence of DSP circuits by employing high-level transformations will be harder if some switches can be designed in such a way that are difficult to trace. A secure reconfigurable switch design is incorporated into the proposed design scheme to improve the security. A complete design flow is presented. In the proposed obfuscation methodology, the variation modes and the additional obfuscating circuits could also be designed systematically based on the high-level transformations. Compared with other existing obfuscation methods, another advantage of the proposed methodology is the generation of meaningful variation modes from a signal processing point of view, since the meaningful modes create ambiguity to the adversary such that it is hard for the adversary to distinguish the correct functionality from other variation modes. Experimental results have demonstrated the effectiveness of the proposed methodology. This paper, for the first time, considers the security perspective of high-level transformations. Future work will explore the algorithmic aspect of different high-level transformations for design obfuscation. Ongoing work includes the validation of the security performances of meaningful modes and non-meaningful modes. We are also interested in addressing the attack methods of DSP circuits.We intend to exploit the security perspective of the proposed methodology by performing various attacks to the obfuscated DSP circuits. Future work will be directed toward developing a complete design flow which can generate the target structure and obfuscation variation modes automatically based on the specific application performance requirement. The ultimate goal is to develop an electronic design automation synthesis tool which can incorporate large number of design obfuscation algorithms based on high-level transformations for DSP system design.

Future Work

The approach presented in this paper will prevent piracy from overproduction and mask theft, because the manufacturer would not have access to either the initialization key or the configure data. These keys could be programmed by another honest vendor after the chips have been fabricated or provided to the customers by the designer. Therefore, overproduced chips without the correct keys cannot function properly.

REFERENCES

[1]. J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Brand and IPprotection with physical unclonable functions," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 3186–3189.

[2]. G. E. Suh and S. Devadas, "Physical unclonable functions for deviceauthentication and secret key generation," in *Proc. 44th Annu. DesignAutom. Conf.*, Jun. 2007, pp. 9–14.

[3]. A. L. Oliveira, "Techniques for the creation of digital watermarks insequential circuit designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, pp. 1101–1117, Sep. 2001.

[4]. D. Kirovski, Y.-Y. Hwang, M. Potkonjak, and J. Cong, "Intellectualproperty protection by watermarking combinational logic synthesis solutions," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 1998,pp. 194–198.

[5]. A. B. Kahng *et al.*, "Watermarking techniques for intellectual propertyprotection," in *Proc. 35th Annu. Design Autom. Conf.*, Jun. 1998, pp. 776–781.

[6]. F. Koushanfar and Y. Alkabani, "Provably secure obfuscation of diverse watermarks for sequential circuits," in *Proc. Int. Symp. Hardw.-OrientedSecurity Trust*, Jun. 2010, pp. 42–47.

[7]. J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy ofintegrated circuits," in *Proc. Conf. Design, Autom. Test Eur.*, Mar. 2008, pp. 1069–1074.

[8]. W. P. Griffin, A. Raghunathan, and K. Roy, "CLIP: Circuit level ICprotection through direct injection of process variations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 5, pp. 791–803,May 2012.

[9]. Y. M. Alkabani and F. Koushanfar, "Active hardware metering forintellectual property protection and security," in *Proc. USENIX SecuritySymp.*, Aug. 2007, pp. 291–306.

[10]. T. Batra. (2005). *Methodology for Protection and Licensing of HDL IP*[Online]. Available: http://www.design-reuse.com/articles/12745

[11]. R. S. Chakraborty and S. Bhunia, "Hardware protection and authentication through netlist level obfuscation," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 674–677.

[12]. R. S. Chakraborty and S. Bhunia, "RTL hardware IP protection usingkey-based control and data flow obfuscation," in *Proc. 23rd Int. Conf.VLSI Design*, Jan. 2010, pp. 405–410.

[13]. R. S. Chakraborty and S. Bhunia, "HARPOON: An obfuscationbased SoC design methodology for hardware protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 10,pp. 1493–1502, Oct. 2009.

[14]. Y. Lao and K. K. Parhi, "Protecting DSP circuits through obfuscation,"in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014.

[15]. [15] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proc. IEEE*, vol. 77, no. 12, pp. 1879–1895,Dec. 1989.

[16]. K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters. I. Pipelining using scattered look-aheadand decomposition," *IEEE Trans. Acoust., Speech, Signal Process.*,vol. 37, no. 7, pp. 1099–1117, Jul. 1989.

[17]. K. K. Parhi, C. Y. Wang, and A. P. Brown, "Synthesis of control circuitsin folded pipelined DSP architectures," *IEEE J. Solid-State Circuits*, vol. 27, no. 1, pp. 29–43, Jan. 1992.

[18]. K. K. Parhi and D. G. Messerschmitt, "Static rate-optimal schedulingof iterative data-flow programs via optimum unfolding," *IEEE Trans. Comput.*, vol. 40, no. 2, pp. 178–195, Feb. 1991.

[19]. K. K. Parhi, "Pipelining in algorithms with quantizer loops," *IEEE Trans.Circuits Syst.*, vol. 38, no. 7, pp. 745–754, Jul. 1991.

[20]. K. K. Parhi, "Low-energy CSMT carry generators and binary adders,"*IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 4, pp. 450–462, Dec. 1999.

[21]. K. K. Parhi, "Design of multigigabit multiplexer-loop-based decisionfeedback equalizers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*,vol. 13, no. 4, pp. 489–493, Apr. 2005.