International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Obfuscation Mechanism for DSP Protection

P. Sandeep
Assistant Professor,
ECE Department,
Vignan Institute of Technology and Science

P. Shiva Rama Krishna
Assistant Professor,
ECE Department,
Vignan Institute of Technology and Science

Mennaiah Batta
Assistant Professor,
ECE Department,
Vignan Institute of Technology and Science

D Kiran Kumar
Assistant Professor,
ECE Department,
Vignan Institute of Technology and Science

***Abstract*: This paper displays a novel way to deal with configuration jumbled circuits for DSP applications utilizing elevated level changes, a key-based FSM, and a reconfiguration. The intention is to design DSP circuits which are reutilized by the specific operational methods by the designer. The design aims at the HLT(high-level transformations) of repeated state graphs which have been utilized for speed, area and power compromises. Many meaningful modes are made used to reconfigurise the order of filter for various applications. Still, existing modes may be corresponding with non-meaningful modes. The configured data controls multiple modes of the circuit operation. Functional obfuscation is fulfilled by the right and unique key and configures data. Incorrect input key is unsuccessful to change the re-configurator and an incorrect configure data generates either a understandable however non-functional or non-understandable mode. Here we have a liability to perform some chance of activating the right mode, which ends up the reduced operations to an obfuscated DSP circuit. The efficiency of proposed implementation is verified with IMAGE SCALING WITH INTERPOLATION AND DECIMATION design; strong high-level obfuscation is proved and analyzed for various key sizes. The aim of the paper is to obfuscate the DSP circuits by using HLT transformations. By the obfuscation, we protect the hardware. The image scaling algorithms employed for confounding the circuits. The image scaling algorithms will change the size of the images. For the image scaling, I used only BILINEAR INSECURITISATION algorithm.**

Keywords:Obfuscation, reconfigure, cyber security,Up sampling, Down sampling, Highlevel transformations.

I. INTRODUCTION

Introduction:

The hardware security could be a severe problem that has leads to plenty of effort on hardware security in terms of piracy and intellectual property (IP). The protection of hardware is often generally classified into 2 ways

- Authentication based Protection,
- Obfuscation based Protection.

Confounding based approach is an interesting thing in the present thesis, an approach that changes a design or an application into one that is functionally identical but is more difficult for reverse-engineering. The hardware protection strategies are attained by neutering the human readability of the HDL code, or by encrypting the source code base cryptographic techniques. Now, varieties of hardware prevention schemes are grower up that modify the FSM representations to change the circuits. Obfuscation have not been proposed for DSP systems till now.

We are generating layout options of smaller dimensions, less than the wavelength (W) of the light, which requires progressively, advanced OPC and alternative DFM techniques at increasing layout space and price.

In the previous paper with the title Protecting DSP circuits through Obfuscation, I implemented the confounded code for filters like IIR, FIR with 1st order, 2nd order and 3rd order filters. In this proposed thesis confounding is used for DSP circuits like digital image with image scaling algorithm.

Filters:

Filtering is a principal function in image process applications, filtering techniques are used to win many roles like resampling, interpolation, and noise removing applications. In almost, all image processing systems filtering of image data is a principle process. The selection of filter is set by the character of the work executed by filter and kind of information

Filtering without Detection:

In this window, the mask is employed for filtering, within the method of filtering; window mask is proceeds across the detected image. The size of the mask is (2N+1)/2, where N is a positive number. During this, the picture element of concern is center of the element. The window mask does arithmetic operations while doing arithmetic operations it does not differentiate any picture element of image once it moves from right bottom corner to the left top corner of the image.

Detection followed by Filtering:

This filtering is carried in 2 steps. In the 1st step, it recognizes the noisy pixels of the image and in the 2nd step, it filters the pixels of the image which contain noise. For filtering a mask is moved across the image. To identify the noisy pixels of the image it does some arithmetic operations. After identifying the noisy pixels of the image which get in the first step, filtering is carried out by placing the non-noisy pixel of the image.

Hybrid Filtering:

A depraved position of a noisy image is filtered with 2 or more filters in the hybrid filtering scheme. The use of a particular filter is based on the conclusion taken by us.

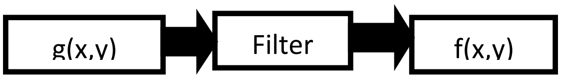


Figure 1.1 Description of filter

Image Scaling:

The resizing of a digital image is explained by image scaling in digital imaging and computer illustration. The magnification of a digital image is understood as an up-scaling or resolution improvement in video technology.

Image Scaling Algorithms:

Image scaling will be taken as a sort of image resampling or image reconstruction from the view of the Nyquist sampling theorem. The two different types of scaling operations are up_sampling and down_sampling. In the scrutiny of up_sampling, a reconstruction filter is used which act like anti-aliasing filter. A progressively advanced way to deal with upscaling regards the issue as a converse issue, explaining the topic of producing a conceivable picture which, when downsized, would appear to be like the information picture. A variety of techniques applied for this, including optimization techniques with regularization terms and the use of machine learning from examples.

Nearest Neighbor Interpolation: This is the simplest way of enlarging image size by substituting each pixel with multiple pixels of the same color.

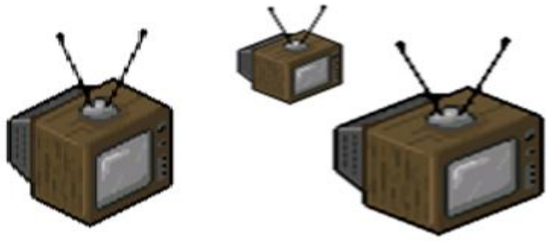


Figure 1.2 Nearest-Neighbor Interpolation (left), Bilinear interpolation (right) image scaling

Bilinear and Bi-Cubic Algorithms: Interpolating picture element color values by introducing endless transition into the output even wherever the material has separate changes.

Sinc and Lanczos Resampling: Sinc resampling in the theory provides the most effective attainable reconstruction for a superbly band-limited signal. Lanczos resampling is approximated as Bi-Cubicinterpolation.

Box Sampling: The bilinear, Bi-Cubicand associated algorithms are sampling a range of pixels. Once downscaling below a specific threshold, the algorithms may sample non-adjacent pixels, which leads to information loss and causes rough results. This is overcome with the box sampling that is to contemplate the target picture element a box on the initial image, and sample all pixels within the box. This guarantees that the output is contributed to everyone input pixels.

Mipmap: Another answer to the downscale fault of bi-sampling scaling is Mipmaps. A Mipmap is pre-scaled set of downscale copies. This can be an algorithmic rule and simple to optimize. It's normal in several frameworks like OpenGL.

Fourier Transform Methods: The Fourier transform-based interpolation depends on filling the frequency domain with zero elements. Besides the great conservation of details, notable is that the ringing and also the circular hurt of content from the left border to the right border.

Edge-directed interpolation: Edge-directed interpolation algorithms point to conserve edges within the image once scaling, in contrast to alternative algorithms which might introduce stairway artifacts.

HQX: For amplifying PC illustrations with low goals or potentially few hues (as a rule from a couple of to 256 hues), great outcomes are picked up withHQX or elective picture component workmanship scaling calculations.

Vectorization: Vector extraction gives a different answer. Vectorizations initially will produce an intention to independent vector illustration of the graphic to be scaled. This method is employed by Adobe artist Live Trace and Inkscape. Scalable Vector Graphics are similar temperament to easy geometric pictures, whereas images don't fare well with Vectorization because of their complexness.

II. METHODOLOGY

In the implementation, image input module designed in the first stage, the image module is designed with counter after that an FSM control module designed in this the states are customized by key values. For every key-value, we have different states are assigned according to the key values the states are changed. For different states, different wait-cycles are assigned according to key-in values. Modelsim software is utilized to obtain simulation results.

The block diagram mainly contains five blocks each block has own importance in implementation. The different blocks are image input block, main control block, line register block, combined filter block, and Interpol bilinear block. The operation of each block is described herein in the preceded headings. The program execution of each block is also explained with flow charts.

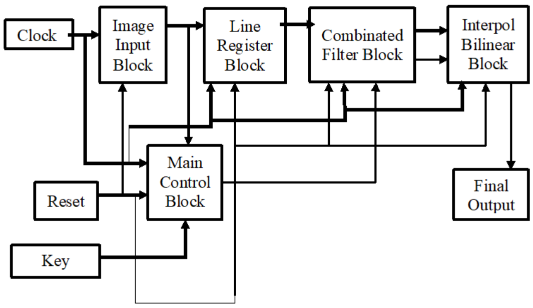


Figure 2.1 Block Diagram.

The image input isn't given by me it's internally generated signal. Whenever reset value is high i.e., logic one the output is 000000……whenever reset value is logic 0 and correct key's given to regulate, then the correct or desired output we get in the output for the execution of program clock is given as input.

Image Input Block:

This block is nothing however a counter that generates count values 0 to 255 which is 256 bits of data. This information is organized as a 16x16 matrix type that is a grey scaled image. For every one bit, different colors are assumed. Here is a flow chart that explains how the count values generated and counting value increased.

Main Control Block:

This block explains about the states which are changed from one to another when timeout occurs and key values are given. For every timeout condition and different key values, transition of states changed. This is observed in the finite state machine diagram.

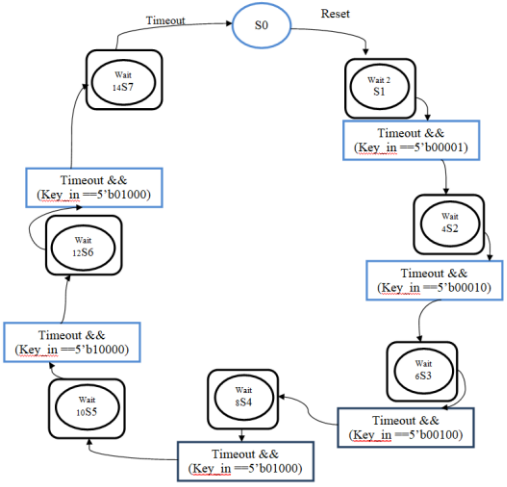


Figure 2.2 Main Control Mechanism.

Line Register Block:

In this block shifting and delay of input data takes place. In this registers which are flip flops only used for storing, delaying and shifting operations. The program execution of each flip flop is same and the flow is explained in the flow chart. Shifting of data takes in flip flops.

Combined Filter Block:

In this module the data is chooses depends on left or right shifting operations in line register block by using a multiplexer. Based on left shift or right shift the up_scaling or down_scaling operation is takes place. In this we have two filters one upscale filter and another one is downscale filter. The multiplexer output is based on the filter output values which are up-scaled or which are downscaled. The flow chart gives the information about execution of program.

The design procedure is described below:

Step1: DSP algorithm: - This step creates the DSP algorithm based on DSP application

Step2: High-level transformation choice: - Based on the precise application, applicable high-level transformation ought to be chosen consistent with the performance demand.

Step3: Obfuscation via HLT techniques: -

Suitable HLT techniques are applied. With obfuscation variation modes, and a totally different configurations of the switch instances are designed.

Step4: Secure controller or switch design: - The secure controller is meant for the development of HLT techniques.

Step5: 2-level FSM generation: - The re-configurator and therefore the confounded FSM are comprised of the DSP design. The configuration key is generated at this stage.

Step6: Design specification: - This step comprises the HDL design, netlist creation and simulation of the DSP system.

The proposed design methodologies don't need vital change to established validation and examine flows. Actually, the confounded DSP circuit with proper key behaves rather like the original circuit. Here we tend to use the DSP circuits are to be confounded via HLT techniques by suitably designing the switches in an exceedingly secure manner.

The switches which are created with HLT techniques are periodic with period N to one switch.

These switches will be enforced as multiplexers and its control signals are obtained from ring counters as shown in Figure 3.10.

Thus, the protection of the switch depends upon the implementation of ring counters i.e., the outputs of a ring counters are going to be obfuscated. An FSM is often outlined by a 6-tuple (I, O, S, S0, F, G), where S, S0, F, G may be a finite set of internal states, I and O represent the inputs and outputs of the FSM, severally, F is that the next state, G is that the output function, and S0 is that the initial state. But, in distinction to general FSMs, the FSM of the ring counter is input independent, such that it endlessly transits to future state depends on the present state. As a result, control signals are periodic.

These switches will be enforced as multiplexers and its control signals are obtained from ring counters as shown in Figure 3.10.

Thus, the protection of switch depends on the implementation of ring counters i.e., the outputs of ring counters are going to be obfuscated. An FSM is often outlined by a 6-tuple (I, O, S, S0, F, G), where S, S0, F, G may be a numerable set of internal states, I and O represent the input and output of the FSM, severally, F is that the next state, G is that the output, and S0 is that the initial state. But, in distinction to general FSMs, the FSM of a ring counter is input independent; such it endlessly transits to future state depends on the current state. As a result, control signals are periodic.

Our scaling methodology needs low computational quality and solely one line memory buffer, thus it's appropriate for low price VLSI implementation. Figure 3.2 shows diagram of the each stage VLSI design for our scaling methodology. The design contains seven main blocks: counter module (CM), register bank (RB), Interpolator and decimator and therefore the controller. Every of them is represented concisely within the following subsections.

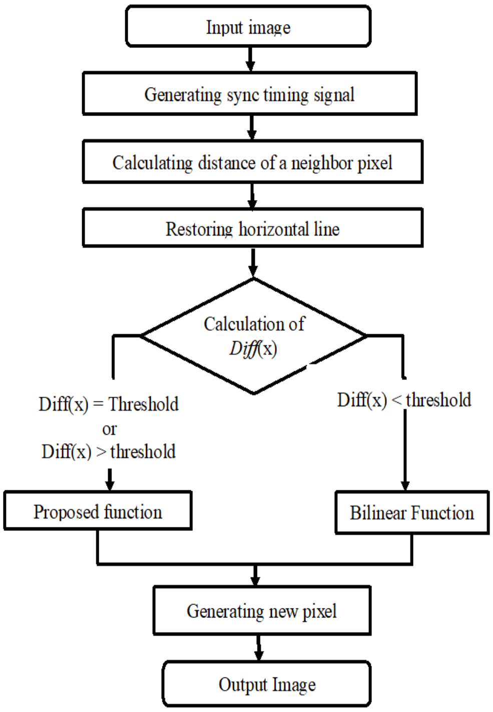


Figure 2.3 Algorithm and Flow Chart for Proposed Design.

1. **Counter Module:** This module is mainly utilized for sequence generation of image pattern for the specified application.
2. **Register Bank**: Here the purpose of the register bank is to provide the different shifting and storing of the image data in different levels.
3. **Interpolation:** Interpolation is a technique of constructing new data points in the well-known data points.
4. **Decimation:** Decimation by a number factor M, will be explained with identical implementation.
    a. Decrease high frequency signal component with a digital LPF.
    b. Down-sample the filtered signal by M, i.e., keep only every M_sample, remaining values are deleted.

Down-sampling alone roots high frequency signal elements to be misinterpreted by later users of the information that may be a kind of distortion known as aliasing. The primary stepis to suppress aliasing to a suitable level. During this application, the filter is termed as an anti-aliasing filter, and its design is mentioned below. Conjointly see under-sampling for data concerning down-sampling bandpass functions and signals.When anti-aliasing filter is an IIR design, it depends on feedback from output to input before the down-sampling step. With FIR filtering, it's an easy meet to calculate solely each Mth output. The calculation performed by a decimating FIR filter for the ordinal output sample could be a real.

$$y(n) = \sum_{k=0}^{k-1} x[nM - k].h[k]$$

IV. ADVANTAGES AND DISADVANTAGES

Advantages:

- Increased number of predicates to deduce by the hacker that is more resources needs to be used by the hacker.
- Put dead and irrelevant code to the original code in order to confuse the hacker.
- The Introduced state diagram code and the code that manipulates the states, blends well with the source code because it uses keying values.
- The predicate design codes are user defined so they resemble the given code.
- In run time the behavior is not determined, because of the randomness mechanism of the code.

Disadvantages:

- In run time, the application needs more space resources for the diagram that is being build, that is used for the not transparent establish conditions.
- The application includes more code lines to be executed (dead code, graph manipulation code) so that it takes more time to execute.This disadvantage can harm an application that uses its resources carefully (like cellular applications).

V . RESULT ANALYSIS

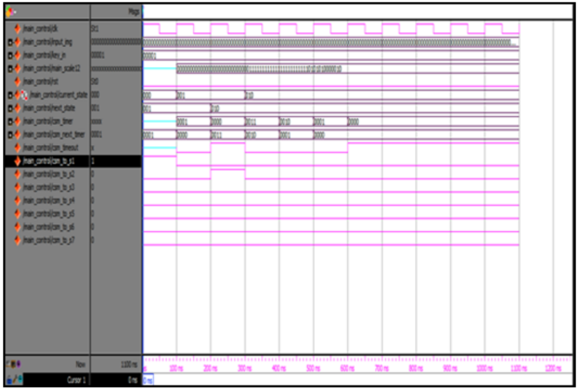
Output Waveforms:

Image input block output:

This waveform output is exported from modelsim. For every clock cycle the output of image input block are changed whenever the reset value is zero. If reset value is one the output of block is not changed the value is like 000000………

Main control block output:

In main control block output the states are changed. When the key value 00001 is given the state is altered from s0 to s1.Ifthe key values are not given the transition of states not happen.

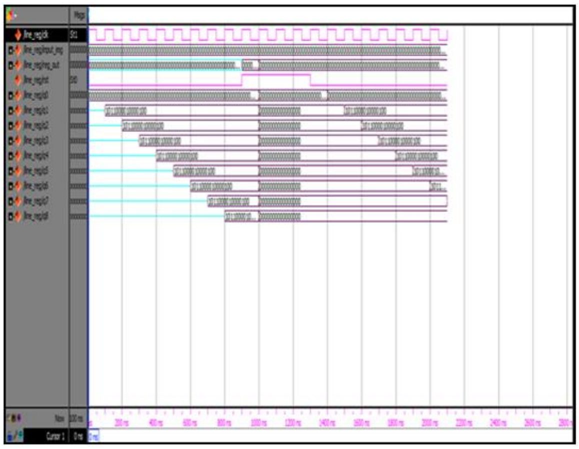


Line registers block output:

In line register block shifting of information bits takes place that is determined within the waveforms that takes only the reset value is zero. If reset value is one the shifting of bits not happen in waveforms.

Combined filter block output:

In the combined filter block the upscaling and downscaling operations of the image takes place. The upscaled and downscaled waveforms values of image are observed here.

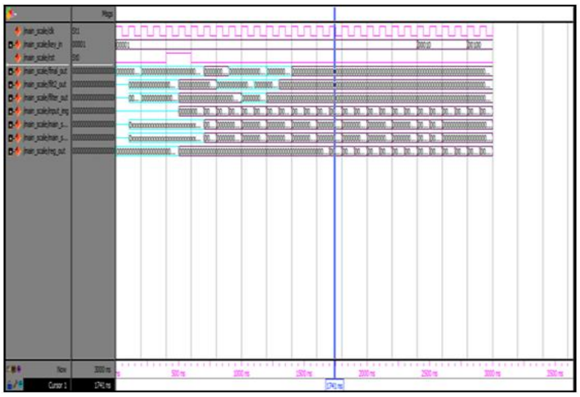


Interpol bilinear block output:

The output of Interpol bilinear block is shown in the waveform. The output image scaled values of the 2 filters are compared once comparison the larger value filter output is come into view as output.

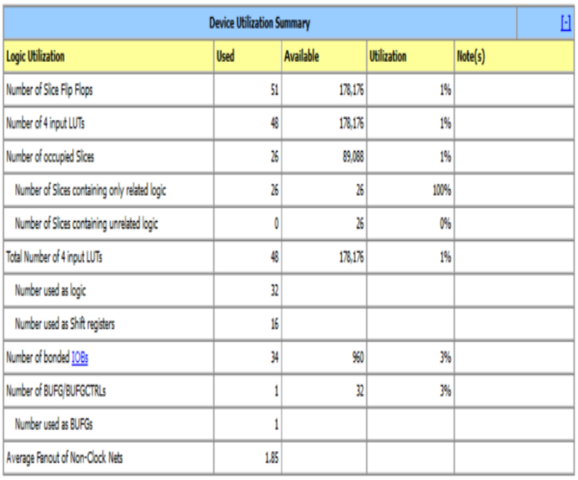
Final output:

The final output waveform values are observed here whenever the key values are changed the final output values are changed. In waveform the key values 00001, 01000 are given, the output observed with both key values is different in nature.

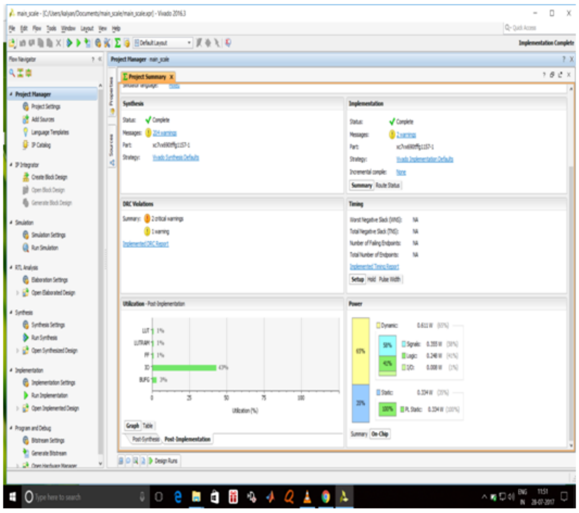


Xilinx Results:

The Xilinx results give the information about how the power ingestion is bring to a small value and area of chip is also bring to small value. The static and dynamic power dissipation values are observed. The static power ingestion is concerning 35% and dynamic power ingestion is concerning 65%. Within the dynamic power ingestion much power is dissipated in signals subsequently logics than in input output section. The on chip power is 0.945w, the junction temperature is 26.3^{0}C which observed in Xilinx results. The utilization of power is more in IO blocks.



Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	51	178,176	1%	
Number of 4 input LUTs	48	178,176	1%	
Number of occupied Slices	26	89,088	1%	
Number of Slices containing only related logic	26	26	100%	
Number of Slices containing unrelated logic	0	26	0%	
Total Number of 4 input LUTs	48	178,176	1%	
Number used as logic	32			
Number used as Shift registers	16			
Number of bonded IOBs	34	960	3%	
Number of BUFG/BUFGCTRLs	1	32	3%	
Number used as BUFGs	1			
Average Fanout of Non-Clock Nets	1.85			



VI. CONCLUSION AND FUTURE SCOPE

It's concluded that confirming the equality of DSP circuits by using HLT techniques are tougher if switches may be made in such a way that they are inconvenient to trace, a configurable switch design is included within the projected design scheme to improve the protection. An entire design flow is given within the proposed confounding methodology, the variation modes and therefore further confounded circuits might even be designed which consistently supported the HLT techniques. Obfuscated and reconfigure FSM modes of which reduce the area of performance speed improved to 341.53MHZ.

REFERENCES

[1] R. S. Chakraborty and S. Bhunia, “RTL hardware IP protection using key-based control and data flow obfuscation,” in Proc. 23rd Int. Conf. VLSI Design, Jan. 2010, pp. 405–410.

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

[3] 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

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

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

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

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

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

[9] 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.

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

[11] C.-Y. Wang and K. K. Parhi, "High-level DSP synthesis using concurrent transformations, scheduling, and allocation," IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 14, no. 3, pp. 274–295, Mar. 1995.

[12] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York, NY, USA: Wiley, 1999.

[13] K. K. Parhi, "Verifying equivalence of digital signal processing circuits," in Proc. 46th Asilomar Conf. Signals, Syst. Comput., Nov. 2012, pp. 99–103.

[14] K. K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," IEEE Trans. Circuits Syst., vol. 38, no. 4, pp. 358–375, Apr. 1991.

[15] Y. Alkabani, F. Koushanfar, and M. Potkonjak, "Remote activation of ICs for piracy prevention and digital right management," in Proc. Int. Conf. Computer.-Aided Design, Nov. 2007, pp. 674–67