International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 22nd Sept2017. Link :http://www.ijiemr.org/downloads.php?vol=Volume-6&issue=ISSUE-8

Title: **FPGA IMPLEMENTATION OF LMS BASED FIR ADAPTIVE FILTER FOR REAL TIME DSP APPLICATIONS**

Volume 06, Issue 08, Pages: 304– 311.
Paper Authors

B.SRIKANTH REDDY,MR.M.HARINATH REDDY

Sphoorthy Engineering College, Sagar Road,Nadargul Village, Vanasthalipuram, Saroornagar Mandal, Hyderabad, Telangana

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

FPGA IMPLEMENTATION OF LMS BASED FIR ADAPTIVE FILTER FOR REAL TIME DSP APPLICATIONS

[1]B.SRIKANTH REDDY,[2]MR.M.HARINATH REDDY

[1]M.Tech Scholar,Dept Of E.C.E, Sphoorthy Engineering College, Sagar Road,Nadargul Village, Vanasthalipuram, Saroornagar Mandal, Hyderabad, Telangana

[2]Assistant Professor,Dept Of E.C.E, Sphoorthy Engineering College, Sagar Road,Nadargul Village, Vanasthalipuram, Saroornagar Mandal, Hyderabad, Telangana

ABSTRACT: Existing designs proposed for the FPGA implementation of an LMS adaptive filter. Excess use of multipliers and longer cycle periods are a few of the issues associated with the existing structures. Based on this study, we propose a design for an FPGA implementation of an LMS based adaptive filter using the Xilinx DSP48. The proposed architecture uses one set of multipliers for both filter output and weight increment term computation. We have simulated the proposed design for a 12-tap adaptive filter in Xilinx system generator and implemented the filter using the Vivado tool set. Implementation results shows that the proposed architecture uses 3 DSP48 units compared to 36 DSP48 units for the existing architecture with the same filter of size 12, and our implementation supports a 75% higher clocking frequency than the existing design. Additionally, the proposed architecture consumes nearly 4.7 times less dynamic power than the existing architecture. Therefore, the proposed architecture is suitable for efficient FPGA realization of an LMS FIR adaptive filter for real-time digital signal processing applications.

Keywords: FPGA; ASIC; LMS Adaptive Filters

1.0 INTRODUCTION TO VLSI:

The complexity of VLSI is being designed and used today makes the manual approach to design impractical. Design automation is the order of the day. With the rapid technological developments in the last two decades, the status of VLSI technology is characterized by the following:

HISTORY OF VLSI:

VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors. This is the field which involves packing more and more logic devices into smaller and smaller areas. VLSI circuits can now be put into a small space few millimeters across.. VLSI circuits are everywhere our computer, our car, our brand new state-of-the-art digital camera, the cell-phones, and what we have.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

VARIOUS INTEGRATIONS:

Over time, millions, and today billions of transistors could be placed on one chip, and to make a good design became a task to be planned thoroughly. In the early days of integrated circuits, only a few transistors could be placed on a chip as the scale used was large because of the contemporary technology, and manufacturing yields were low by today's standards. As the degree of integration was small, the design was done easily. Over time, millions, and today billions of transistors could be placed on one chip, and to make a good design became a task to be planned thoroughly.

MSI TECHNOLOGY

The next step in the development of integrated circuits, taken in the late 1960s, introduced devices which contained hundreds of transistors on each chip, called "medium-scale integration" (MSI).They were attractive economically because while they cost little more to produce than SSI devices, they allowed more complex systems to be produced using smaller circuit boards, less assembly work (because of fewer separate components), and a number of other advantages.

LARGE SCALE INTEGRATION:

Further development, driven by the same economic factors, led to "large-scale integration" (LSI) in the mid-1970s, with tens of thousands of transistors per chip.Integrated circuits such as 1K-bit RAMs, calculator chips, and the first microprocessors, that began to be manufactured in moderate quantities in the early 1970s, had under 4000 transistors. True LSI circuits, approaching 10,000 transistors, began to be produced around 1974, for computer main memories and second-generation microprocessors.Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors. VLSI stands for "Very Large Scale Integration". This is the field which involves packing more and more logic devices into smaller and smaller areas. Thanks to VLSI, circuits that would have taken board full of space can now be put into a small space few millimeters across! This has opened up a big opportunity to do things that were not possible before. VLSI has been around for a long time, there is nothing new about it, but as a side effect of advances in the world of computers, there has been a dramatic proliferation of tools that can be used to design VLSI circuits. Alongside, obeying Moore's law, the capability of an IC has increased exponentially over the years, in terms of computation power, utilization of available area, yield.

VLSI DESIGN STYLES:

Several design styles can be considered for chip implementation of specified algorithms or logic functions. Each design style has its own merits and shortcomings, and thus a proper choice has to be made by designers in order to provide the functionality at low cost.

GATE ARRAY DESIGN:

In view of the fast prototyping capability, the gate array (GA) comes after the FPGA. While the design implementation of the

FPGA chip is done with user programming, that of the gate array is done with metal mask design and processing. Gate array implementation requires a two-step manufacturing process: The first phase, which is based on generic (standard) masks, results in an array of uncommitted transistors on each GA chip. These uncommitted chips can be stored for later customization, which is completed by defining the metal interconnects between the transistors of the array Since the patterning of metallic interconnects is done at the end of the chip fabrication, the turn-around time can be still short, a few days to a few weeks. a corner of a gate array chip which contains bonding pads on its left and bottom edges, diodes for I/O protection, nMOS transistors and pMOS transistors for chip output driver circuits in the neighboring areas of bonding pads, arrays of nMOS transistors and pMOS transistors, underpass wire segments, and power and ground buses along with contact windows.

Standard-cells based design

The standard-cells based design is one of the most prevalent full custom design styles which require development of a full custom mask set. The standard cell is also called the police. In this design style, all of the commonly used logic cells are developed, characterized, and stored in a standard cell library. A typical library may contain a few hundred cells including inverters, NAND gates, NOR gates, complex AOI, OAI gates, D-latches, and flip-flops. Each gate type can have multiple implementations to provide adequate driving capability for different fan outs. For instance, the inverter gate can have standard size transistors, double size transistors, and quadruple size transistors so that the chip designer can choose the proper size to achieve high circuit speed and layout density. The characterization of each cell is done for several different categories. It consists of

- delay time vs. load capacitance
- circuit simulation model
- timing simulation model
- fault simulation model
- cell data for place-and-route
- mask data

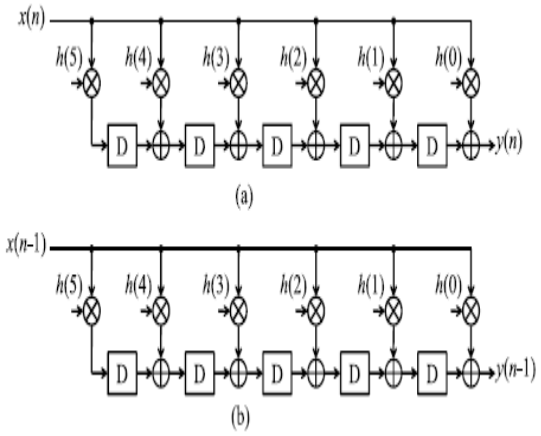
2.0 LITERATURE SURVEY:

FINITE-IMPULSE response (FIR) digital filter is widely used in several digital signal processing applications, such as speech processing, loud speaker equalization, echo cancellation,adaptive noise cancellation, and various communication applications, including software-defined radio (SDR) and so on Many of these applications require FIR filters of large order to meet the stringent frequency specifications Very often these filters need to support high sampling rate for high-speed digital communication The number of multiplications and additions required for each filter output, however, increases linearly with the filter order. Since there is no redundant computation available in the FIR filter algorithm, real-time implementation of a large order FIR filter in a resource constrained environment is a challenging task.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COMPUTATIONAL ANALYSIS:

The data-flow graphs (DFG-1 and DFG-2) of transpose form FIR filter for filter length $N = 6$, as shown in Fig. 1, for



Dfg Of Transpose Form Structure For $N = 6$. (A) Dfg-1 For Output $Y(N)$. (B) Dfg-2 For Output $Y(N-1)$.

Mathematical Formulation Of The Transpose Form Block Fir Filter

Suppose in every cycle, the block FIR filter takes a block of L new input samples, and processes those to produce a block of L output samples. The kth block of filter output $\mathbf{y}k$is computed using the relation

$$\mathbf{y}k = \mathbf{X}k \cdot \mathbf{h} \qquad (3)$$

where the weight vector $\mathbf{h}$ is defined as
$\mathbf{h} = [h(0), h(1), \ldots, h(N-1)]T$.
The input matrix $\mathbf{X}k$is defined as

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k^0 & \mathbf{x}_k^1 & \ldots & \mathbf{x}_k^4 & \ldots & \mathbf{x}_k^{N-1} \end{bmatrix}$$

where$\mathbf{x}ik$is the $(i + 1)$th column of $\mathbf{X}k$are defined as

$$\mathbf{x}ik=[x(kL- i\ ) x(kL- i - 1) \cdot \cdot \cdot x(kL- i - L + 1)]T. \qquad (5)$$

Substituting (4) in (3), the matrix-vector product is expressed in the form of scalar–vector

$$\mathbf{y}_k = \sum_{i=0}^{N-1} \mathbf{x}_k^i \cdot h(i). \qquad (6)$$

Suppose N is a composite number and decomposed as
$N = ML$, then index i is expressed as $i = l + mL$, for $0 \le l \le L - 1$, and $0 \le m \le M - 1$. Substituting $i = l + mL$ in (5), we have

$$\mathbf{x}_k^{l+mL} = \mathbf{x}_{k-m}^l. \qquad (7)$$

Substituting (7) in (4), we have $\mathbf{X}k= \_\mathbf{x}0\ k$ $\mathbf{x}1$

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x}_k^0 & \mathbf{x}_k^1 & \cdots & \mathbf{x}_k^{L-1} & \mathbf{x}_{k-1}^0 & \mathbf{x}_{k-1}^1 & \cdots & \mathbf{x}_{k-1}^{L-1} & \cdots \\ \mathbf{x}_{k-M+1}^0 & \mathbf{x}_{k-M+1}^1 & \cdots & \mathbf{x}_{k-M+1}^{L-1} \end{bmatrix}. \qquad (8)$$

Substituting (8) in (3), we have

$$\mathbf{y}_k = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} \mathbf{x}_{k-m}^l \cdot h(l + mL). \qquad (9)$$

The input matrix $\mathbf{X}k$of (8) has an interestin feature. The data block $\mathbf{x}0\ k$ is the current block,while $\{\mathbf{x}0\ k-1, \mathbf{x}0\ k-2, \ldots, \mathbf{x}0\ k-M+1\}$ are blocks delayed by $1, 2, \ldots, (M - 1)$ cycles. The overlapped blocks $\{\mathbf{x}1\ k-1, \mathbf{x}1\ k-2, \ldots, \mathbf{x}1\ k-L+1\}$ are, respectively, 1 clock cycle, 2 clock cycles, . . . , $(M - 1)$ cycles delayed version of overlapped block $\mathbf{x}1\ k$. To take the advantage of this feature, the input-matrix $\mathbf{X}k$is decomposed into M small matrices $\mathbf{S}l\ k$, such that $\mathbf{S}0k$ contains L inputblocks$\{\mathbf{x}0\ k, \mathbf{x}1\ k, \ldots, \mathbf{x}L-1\ k\}$, and $\mathbf{S}1k$ contains input blocks $\{\mathbf{x}0\ k-1, \mathbf{x}1\ k-1, \ldots, \mathbf{x}L-1\ k-1\}$. Similarly, the input block $\{\mathbf{x}0\ k-M+1, \mathbf{x}1\ k-M+1, \mathbf{x}L-1\ k-M+1\}$constitute the matrix $\mathbf{S}M-1\ k$.
The coefficient vector $\mathbf{h}$ is also decomposed into small weight vectors $\mathbf{c}m = \{h(mL), h(mL + 1), \ldots, h(mL + L - 1)\}$. Interestingly, $\mathbf{S}mk$is symmetric and satisfy the following identity:

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

$$\mathbf{S}_k^m = \mathbf{S}_{k-m}^0. \quad (10)$$

According to (10), **S***mk*(for $1 \le m \le M-1$) are *m* clock cycle delayed with respect to **S**0*k*

. Computation of (9) can be expressed in matrix-vector product using **S**0*k* −*m* and **c***m* as

$$\mathbf{y}_k = \sum_{m=0}^{M-1} \mathbf{r}_k^m \quad (11a)$$

$$\mathbf{r}_k^m = \mathbf{S}_{k-m}^0 \cdot \mathbf{c}_m. \quad (11b)$$

The computations of (11) may be expressed in a recurrence form

$$\mathbf{Y}(z) = \mathbf{S}^0(z)[(z^{-1}(\cdots(z^{-1}(z^{-1}\mathbf{c}_{M-1} + \mathbf{c}_{M-2}) + \cdots) + \mathbf{c}_1) + \mathbf{c}_0]$$

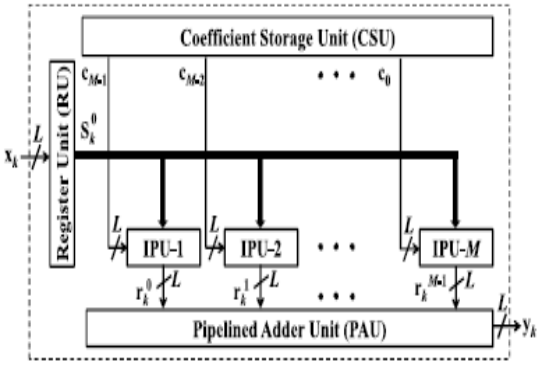
where**S**0(*z*) and **Y**(*z*) are the *z*-domain representation of **S**0*k* and **y***k*, respectively. The DFG-4 of block transpose form type-II configuration (shown in Fig. 5 for $N = 6$ and $L = 2$) can be derived using the recurrence relation of (12). The delay operator {*z*−1} of (12) represents a delay for a block of data in the transpose form type-II structure that stores the product of **S**0*k* and **c***m*.The proposed structure (transpose form type-II) is presented in Section III.

3.0 PROPOSED STRUCTURES:

There are several applications where the coefficients ofFIR filters remain fixed, while in some other applications,like SDR channelize that requires separate FIR filters ofdifferent specifications to extract one of the desired narrowbandchannels from the wideband RF front end. These FIRfilters need to be implemented in a RFIR structure to supportmultistandard wireless communication [6]. In this section, wepresent a structure of block FIR filter for such reconfigurableapplications. In this section, we discuss the implementation ofblock FIR filter for fixed filters as well using MCM scheme.

STRUCTURE FOR TRANSPOSE FORM BLOCK FIR FILTER FOR RECONFIGURABLE APPLICATIONS

The proposed structure for block FIR filter is [based onthe recurrence relation of (12)] shown in Fig. 6 for theblock size $L = 4$. It consists of one coefficient selectionunit (CSU), one register unit (RU), *M* number of innerproductunits (IPUs), and one pipeline adder unit (PAU).The CSU stores coefficients of all the filters to be usedfor the reconfigurable application. It is implemented using*N* ROM LUTs, such that filter coefficients of any particularchannel filter are obtained in one clock cycle, where *N* is thefilter length. The RU [shown in Fig. 7(a)] receives **x***k*duringthe *k*th cycle and produces *L* rows of **S**0*k*in parallel.



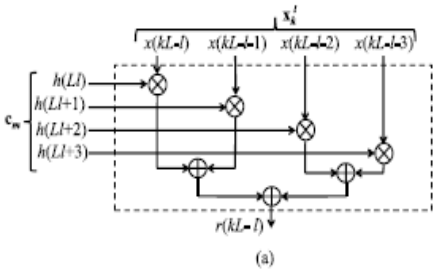
Proposed Structure For Block Fir Filter

MCM-BASED IMPLEMENTATION OF FIXED-COEFFICIENT FIR FILTER:

The derivation of MCM units for transpose formblocks FIR filter, and the

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

design of proposed structure for fixedfilters. For fixed-coefficient implementation,



(a)

(a) Internal structure of (l + 1)th IPC for L = 4

longer required, since the structure is to be tailored for only one given filter. Similarly, IPUs are not required. The multiplications are required to be mapped to the MCM units for a low-complexity realization. In the following, we show that the proposed formulation for MCM-based implementation of block FIR filter makes use of the symmetry in input matrix **S**0k to perform horizontal and vertical common sub expression elimination.

$$\mathbf{Y}(z) = z^{-1}\cdots z^{-1}(z^{-1}\mathbf{r}_{M-1} + \mathbf{r}_{M-2} + \mathbf{r}_{M-3}) + \cdots + \mathbf{r}_1 + \mathbf{r}_0.$$

To illustrate the computation of (14) for L = 4 and N = 16,we write it as a matrix product given by (16). From (16),we can observe that the input matrix contains six-input samples {$x(4k)$, $x(4k-1)$, $x(4k-2)$, $x(4k-3)$, $x(4k-4)$, $x(4k-5)$, $x(4k-6)$}, and multiplied with several constant coefficients,

FPGA IMPLEMENTATION:

The proposed architecture is implemented in Xilinx 7K325T FPGA. The logiCORE component DSP48 is used for implementation of multiplication and addition. The design is simulated using Xilinx system generator tool and the corresponding HDL code synthesized and implemented in FPGA using Xilinx Vivado software. The simulation was performed using Vivado simulator.

POWER ESTIMATION COMPARISON:

	Proposed	Yongbo et al [2]
Dynamic	0.051W	0.195W
Clock	0.021W	0.018W
Signals	0.010W	0.037W
Logic	0.004W	0.017W
DSP	0.015W	0.089W
I/O	0.001W	0.033W
Device Static	0.158W	0.159W

4.0 VERILOG HARDWARE DESCRIPTION LANGUAGE:

Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i.e., the switch level or, it might describe the logical gates and flip flops in a digital system, i.e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all of these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

NUMBER REPRESENTATION:

Verilog permits numbers to be represented using a binary, octal, hex or decimal representation. Apart from the normally permissible values (0 and 1 for binary, 0 through 7 for octal, 0 through F for hex, and

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

0 through 9 for decimal), each digit may take on the values x (unknown) or z (high impedance). A number may be represented in a *sized* or *unsized*form, depending on whether the number of bits is specified or not. A sized number is represented in the form size ' base_format number where size corresponds to the number of bits, base_format takes on one of the possible values of b (binary), o (octal), h (hex) or d (decimal), and number is the actual value of the number. It is important to note that regardless of the base that is used, the size refers to the number of binary bits in the representation.

5.0 XILINX:

The Integrated Software Environment (ISE™) is the Xilinx® design software suite that allows you to take your design from design entry through Xilinx device programming. The ISE Project Navigator manages and processes your design through the following steps in the ISE design flow

PROJECT NAVIGATOR OVERVIEW:

Project Navigator organizes your design files and runs processes to move the design from design entry through implementation to programming the targeted Xilinx® device. Project Navigator is the high-level manager for your Xilinx FPGA and CPLD designs, which allows you to do the following:

- Add and create design source files, which appear in the Sources window
- Modify your source files in the Workspace
- Run processes on your source files in the Processes window
- View output from the processes in the Transcript window

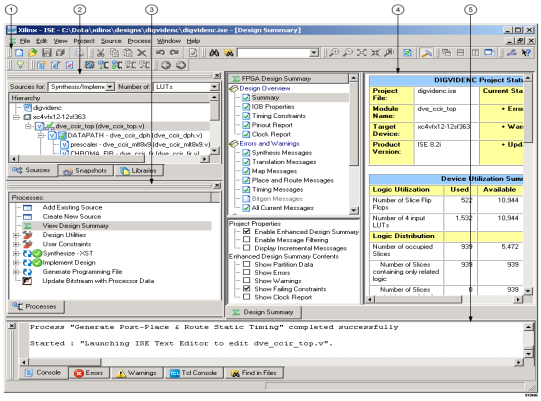


Figure Project Navigator

Depending on the source file and tool you are working with, additional tabs are available in the Sources window:

- Always available: Sources tab, Snapshots tab, Libraries tab
- Constraints Editor: Timing Constraints tab
- Floor plan Editor: Translated Netlist tab, Implemented Objects tab
- iMPACT: Configuration Modes tab
- Schematic Editor: Symbols tab
- RTL and Technology Viewers: Design tab
- Timing Analyzer: Timing tab.

CONCLUSION

In this it is explored the possibility of realization of block FIR filters in transpose form configuration for area delay efficient realization of both fixed and reconfigurable applications. A generalized block

formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications.We have presented a scheme to identify the MCM blocks for horizontal and vertical subexpression elimination in the proposed block FIR filter for fixed coefficients to reduce the computational complexity. Performance comparison shows that the proposed structure involves significantly less ADP and less EPS than the existing block direct-form structure for medium or large filter lengths while for the short-length filters, the existing block direct-form structure has less ADP and less EPS than the proposed structure. Application-specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involve 42% less ADP and 40% less EPS than the best available FIR filter structure of for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves 13% less ADP and 12.8% less EPS than that of the existing direct-from block FIR structure.

BIBILOGRAPHY:

[1] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing:Principles, Algorithms and Applications*. Upper Saddle River, NJ, USA:Prentice-Hall, 1996.

[2] T. Hentschel and G. Fettweis, "Software radio receivers," in*CDMA Techniques for Third Generation Mobile Systems*. Dordrecht,The Netherlands: Kluwer, 1999, pp. 257–283.

[3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, "A new adaptive noisecancellation scheme in the presence of crosstalk [speech signals]," *IEEETrans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 10,pp. 681–694, Oct. 1995.

[4] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering andvariable gain ranging seismic data acquisition system," in *Proc. IEEESoutheastcon*, Apr. 1993, p. 1–6.

[5] J. Mitola, *Software Radio Architecture: Object-Oriented Approaches toWireless Systems Engineering*. New York, NY, USA: Wiley, 2000.

[6] A. P. Vinod and E. M. Lai, "Low power and high-speed implementationof FIR filters for software defined radio receivers," *IEEE Trans. WirelessCommun.*, vol. 7, no. 5, pp. 1669–1675, Jul. 2006.

[7] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, andK. Roy, "Computation sharing programmable FIR filter for low-powerand high-performance applications," *IEEE J. Solid State Circuits*,vol. 39, no. 2, pp. 348–357, Feb. 2004.

[8] K.-H. Chen and T.-D.Chiueh, "A low-power digit-based reconfigurableFIR filter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8,pp. 617–621, Aug. 2006.

.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org