

# Implementation of Convolutional Encoder by **XOR-Free Approach**

Pankaj Kumar<sup>1</sup>, Deepak Kumar<sup>2</sup>

M.Tech Scholar Department of Electronics and Communication Engineering, Vidhyapeeth Institute of Science &

Technology, Bhopal, India<sup>1</sup>

Assistant Professor, Department of Electronics and Communication Engineering, Vidhyapeeth Institute of Science &

Technology, Bhopal, India<sup>2</sup>

Abstract: Due to the excellent error control performance in many communication systems Convolution encoder and Viterbi decoder are widely used. In coding techniques the number of symbols in the source encoded message is increased in a controlled manner in order to facilitate two basic demand at the receiver one is Error detection and other is Error correction. The amount of error detection and correction required and its efficiency depends on the signal to noise ratio (SNR). The technology based on Non Line of Sight (NLOS) ability to make the system very attractive for users, but there will be a little higher BER at low SNR. Coding is a technique where redundancy is added to original bit sequence to increase the reliability of the communication. This paper presents a review on hardware implementation of Convolution Encoder with power efficient architecture. The results of this architecture will decrease the dynamic power and HW cost with lower design complexity as comparing to conventional method.

Keywords: Automatic Repeat Request (ARQ), Forward Error Correction (FEC), Convolutional Code (CC), Non Line of Sight (NLOS).

# I. INTRODUCTION

This In a communication system, error detection and relation how each possible input to the encoder impacts correction mechanisms are vital and numerous techniques on the output in shift register. Viterbi algorithmis used for exist for reducing the consequence of bit-errors and trying decoding. In communication, a convolution code is a type to make sure that the receiver eventually gets an error free of error-correcting code in which version of the packet. The most important technique used are fault detection with Automatic Repeat Each m-bit information symbol (each m-bit string) to Request (ARQ), Forward Error Correction (FEC) and be encoded is transformed into an n-bit symbol, hybrid forms of ARQ and FEC. This development focus on FEC techniques. Forward Error Correction (FEC) is an error control method for data transmission by adding redundant data to its messages to improve the capacity of a channel. This redundant data allows the receiver to detect and to correct a certain number of errors without asking the encoder to re-transmit more additional data. The process of adding this redundant information is known as channel coding. Mainly there are two major number of output bits. In this example, rate is 1/2 which kinds of channel coding: block codes like Reed -Solomon coding and Convolution coding. Block codes (b) Constraint length: The number of delay elements work with fixed length blocks of code. Convolution codes in the convolution coding for example with k = 3, there deal with data sequentially. Block codes become very complex as their length increases and are therefore harder c) Generator polynomial: Wiring of the input sequence to implement. Convolution codes in comparison to block codes are less complex and therefore easier to implement.

# Convolutional Code (CC)

In Non Line of Sight (NLOS) Communication such as previous input. The output from 58 = 1012 uses the XOR Mobile Wi-Max of CDMA part, the CC is the only required coding scheme. Its computations depend not only on the existing set of input symbols it also depends up Rate =  $\frac{1}{2}$ on some of the previous used input symbols. A lattice Constrain length, k=3description is used for convolution encoding that show

where m/n is the code rate  $(n \ge m)$ .

The transformation is a function of the last information symbols, where k is the constraint length of the code.

The convolutional codes are defined by three parameters which are as follow:

(a) Rate: Ratio of the number of input bits to the means there are two output bits for each input bit.

are two delaying elements.

with the delay elements to form the output. For example, generator polynomial is [7,5]8 = [111,101]2. The output from the 78 = 1112 arm uses the XOR of the current input, previous input and the previous to of the current input and the previous to previous input.

Generator polynomial is [7, 5]8 = [111, 101]2



Generic Methods for Decoding Convolution code

Convolution codes which are Feedback decoding, sequential decoding and maximum prospected decoding.

1. Threshold decoding – This decoding is called majority logic decoding. It is successfully applied only on definite classes of code. It applies to channel having a slight to good SNR. It is far away from optimal because of its inferior in bit error performance.

2. Sequential decoding –This decoding is sub optimal. Convolution operation is realized using a deterministic This decoding has better performance than the previously used method. Virtually independent from the length of the particular code is the advantage of it. Unpredictable decoding latency & variable decoding time is its drawback. Also, it requires a large memory.



Figure 1: Convolutional code with Rate 1/2, K=3, Generator Polynomial octal

3. Viterbi decoding - It is optimal algorithm for decoding of Convolution code. It is the dominant technique for Convolution codes. It has advantages like satisfactory bit error performance, low cost, fixed decoding time. most extensively decoding algorithms The for Convolution codes is Viterbi code proposed in 1967. It is very useful method for forward error correction. In many wireless communication systems like IEEE 802.11a/g, Wi-Max, WCDMA and GSM to improve capacity of communication channel it is widely used. Due to high demand of the portable wireless communication devices by user. So need of high speed viterbi decoder increasing.

### **II. METHOD**

Viterbi algorithm is the most possibility decode algorithm of convolution code. Viterbi decoder means the VLSI implementation of Viterbi algorithm. In the area of communication, convolution code is very popular, so how to improve the performance and reduce the power and area of the decoder is important. In the other hand, different protocols use different convolution code and varied applications have different requirement for throughput, area and power. So design of reusable Viterbi decoder is important, too. This decoder adopted the Process Element (PE) technique, which made it easy to adjust the throughput of the decoder by growing or falling the number of PE. By the method of Same Address Write Back (SAWB), we compact the number of register to half

in disparity with the method of ping-pong. This decoder There are many different decoding techniques for supported punctured convolution code and was datadriven, which means the circuit was able to work under different data rate and avoid those invalid operations.PE Process Element, was one of the most popular architecture in digital signal processing .The PE architecture of Viterbi decoder has been introduced. Fig1 is the basic structure of PE in Viterbi decoder. Which consists of PMU, ACSU, BMU and LRU?

> finite state machine (DFSM). Its hardware implementation requires a combinational circuit and memory elements. The discrete convolution for the encoded sequence (Cj) can be expressed in terms of information sequence (Ij) with the generator sequences (GI) by the following equation:

> Further shift register (SR) based realization of (1) for encoded sequence (Cj $\beta$ ) depends upon the length (L) of SR, the present input Ij and M previous input blocks [Ij1,...., Ij-M] to yield (2).

$$C_{j\beta} = \sum_{\alpha=1}^{n} \left[ \sum_{I=0}^{M} I_{j=I} \alpha g_{\alpha\beta}^{(I)} \right]$$
(2)



Figure 2: Structure of PE

### **III.RESULT ANALYSIS**

	E Design Overview		taplie Project	Status (06/24	(2016 - 13(06:54)			
Vex: () 🚯 Implementation 🛎 🛃 Simulation	108 Properties	Project File:	softer_prisk.torercide.size	Parser I	mme	No Errora		_
Betword         ■           Hearchy         ■           Hauchy         ■           Mather         =           Mather	Module Level Utilization     Timing Constraints     Pinout Report     Ceck Report	Hodule Rame:	topfe	Implem	ntation States	Placed or	d Risulted	
		Target Device:	valu 2001 Bragi D4	•1	TOPE	No Gross		
		Product Version:	DE 14.1	•1	lamings:	225.55m	ina (10m)	
	Static Timing	Design Goalt	Beimord	Routing Results:		Al.Scrab	Al Scrais Consisteix Routed	
	Application Messages     Application Messages     Application Messages     Application Messages	Design Strategy:	Wm Defait Unitsted	Timing Constraints:		42 Comb	All Construints Hert	
		forwanest	Seller Selling	final Tiering Score:		0 (Timin	© (Timine Report)	
		Device Utilization Summary						
P. 7	- Timing Messages	Sice Logic Utilization		Used	Available	Utilization	Note(s)	
No Processes Running	to Processes Running	Number of Skite Angelers	1		3 126,800	17	6	
Processes uut - topfile		Nation used as Fig.Fig.	фя.		3			
E Sim Smuletor	A Providence Proved	<ul> <li>Number used as Latche</li> </ul>	5		0			
Eehavioral Check Syntax Simulate Behavioral Model	Design Properties	Number used as Latch	trua		0			
Sentral Desired incom	Optional Design Summary Contents	Number used as 4/C/I	Riopa		0]			
	Prov Clock Report     Prov Failing Constraints     Prov Railing Constraints     Prov Railing     Prov Railing     Prov Railing	Number of Skit LUTs			7 63,400	17	6	
		Number used as logic			7 63,400	17	6	
		Number using OK ou	lput only		7			
		Number using 05 st	but only		0			
		Number using 05 an	d 06		0			
Start 💐 Design 🚺 Hies 🚺 Libraries	E Despis	Sumary	0					



E Project Navigetor (P.15x1) - Elvortree, convoluti File Edit View Project Source Process		ng multi				pĺ
980 X 0 0 X 0 8 6		THE REPORT OF A DESCRIPTION OF A DESCRIP				
+06x	B Design Overview	·	vice Utilization Summary			
Verc () 🐉 Indenentation 🔋 📓 Smulation	C ICE Properties	Sice Logic Utilization	Used	Available	Utilization	Note(s)
Behavioral	O Module Level Utilization     O I ming Constraints	Note of Sice Repiters	3		1%	mucial
inversion of the second se	D Sand June	Number used as Rip Flops	3	10.50	10	
na inter convolucionada a V 🛛 🖥 Cock Report	Cock Report	E Note use a lattes				
a munee m house my	State Timing	Number used as Latch-frus	0	-		
∃ 🛿 wt-topfie (topfiex) - 🕜 gl - shiftegister (shiftegiste	Parser Messages	Number used as AVIC (CR logis	0			
	(A) Synthesis Messages	Note of Sice LUTs	1		1%	
gi - muki (muki.v)	Tanslation Messages	Number used as logic	1	63,40	18	
	Place and Route Messages	Number using Disculput only	1	0,10		
( <u> </u>	Timing Messages	Number using DS output only	0			
No Processes Running	- 🗋 Bitgen Messages - 👔 Al Implementation Messages	Number using 05 and 06	0			
	🖯 Detailed Reports	A CONTRACTOR	-	_		
tocesses uit - topfie	Synthesis Report	+ Number used as ROM	0			
Sim Smulator	Design Properties	- Nurber used as Henry	0	19,000	0%	
Simulate Behavioral Model	🗏 📄 Enable Message Filtering	Nurber used exclusively as nute-finus	0		نمر.	
	Optional Design Summary Contents 	Noter of coupled Sites	4	15,851	1%	
	- Show Failing Constraints	Number of WIT Flip Flip pairs used	8			
	- 🗄 Show Warnings	Number with an unused Flip Flop	5			
	- 🗄 ShowErrors	Number with an unused UUT		<u> </u>	2010	
		Number of fully used LUT FF pairs	2		25%	
1						
m Console 🗿 Brurs 🧕 Mamoga 🍓 Find	nfieleus					
unsk D dus 🔬 temp 🕷 ind	n File Stads					Ê 4 () 1
0 8 1 2	1	ico Sarrel				<b>6 1</b> 1
9 🗿 🛓 🌶	iorendeluche andutorendeuiz-1	icy Surre)			*	
Pojet Naigetor (P.S.M Exortee consolid A biet New Projet Source Process	iorendeluche andutorendeuiz-1					<b>6 1</b> 1
Pojet Naigetor (P.S.M Exortee consolid A biet New Projet Source Process	j tionenade/yorher, consolutionencodessie - 15 Tools Window Layout Help					<b>6 1</b> 1
Pojet Narjet/ PEdi - Ekone consult       a Sit View Pojet Source Proces	Torenade unter annulationenade size - [ Tore Tarlet annulationenade size - [ Tore Tarlet annulationenade size - ] Tore Tarlet annulationenade s	□ ₽ ₽ ११ ► 2 ¶ ¶ Nutentianus/FpRp			626	<b>6 1</b> 1
Poject Tairgetor (P.E.fr) - Excitner, consolid a 587 View Poject Source Process P - D - D - D - D - D - D - D - D - D -	ionenzáře tortes (analdatorecodes use - ) Tots Windou Ispot Hép				626	8 4 U 6
Anger Bauger (25th - Bonner ganadi E lith Tear Anger Same Anger B B B B B B B B B B B B B B B B B B B	Tore material free considerance lie - [ Toris lindon Layout Help	□ ₽ ₽ ११ ► 2 ¶ ¶ Nutentianus/FpRp		1	Q%	8 4 U 6
Prior Langton P.25M - Exchange model Prior Langton P.25M - Exchange model E Hit like - Prior Linear Anness B B B Linear Linear Anness B B Linear Linear Anness B S S S S S S S S S S S S S S S S S S S	Tore Under Suffree canadidation and solar - 1 Tore Under Layout Help	Instantial and a musel file Rep           Instantial and a musel file Rep	1	8	£2%	8 4 U 6
9         2         >           Argiet longtor (254)         Bonfree genedities         >           1         10         >         >           1         10         >         >         >           1         10         >         >         >         >           1         10         >	Tote Vortes annolationerande vortes annolationerande vortes Tote Vindow Layoz Hép	Image: Second	1	8	62% 12% 25%	8 4 U 6
Anjert Rangton (2514) - Bonfree ganarial Project Rangton (2514) - Bonfree ganarial te the file of a fore Annese and a foregan and a foregan and a second a foregan and	Tots Window Layoz Hép Tots Window Layoz Hép P P B P P B P B B B B B B B B B B B B B	Image: Second	1	8	0% 2% 2%	<b>6 1</b> 1
9         2         2           Prijet laugtor P.Daft, Boofner, consult         2         2           Prijet laugtor P.Daft, Source Processon         2         2           10         10         2         2           10         10         2         2         2           10         10         2         2         2         2           10         10         2         3         2         3         2         3         2         3	Tote Vortes considerende v	Image: Section 1         Image: Section 1         Image: Section 1           Image: Section 1         Image: Section 1         Image: Section 1           Image: Section 1         Image: Section 1         Image: Section 1           Image: Section 1         Image: Section 1         Image: Section 1           Image: Section 1         Image: Section 1         Image: Section 1           Image: Section 1         Image: Section 1         Image: Section 2           Image: Section 1         Image: Section 2         Image: Section 2	1 2 1 5 21	8 125,400 221	62% 12% 25% 1%	8 4 U 6
Apert laugus (25th - Storing concil - Apert laugus (25th - Storing concil + Exit Iva - Apert Sure: Anes: 	Tots Window Layoz Hép Tots Window Layoz Hép P P B P P B P B B B B B B B B B B B B B	Image: State of the s	1 2 1 5 21 21	8 125,800 210 125	62% 12% 23% 1% 0%	<b>6 1</b> 1
Piper langtor PEAR - Bonfree consol Piper langtor PEAR - Bonfree consol e Stit Han Piper Source Process PERF Source Processor PERF Source Processor PERF Source Processor PERF Source Processor PERF Source Processor PERF PROCESSOR	innersatief software, smooth sover some sover some sover sov	Image: Section 1     Image: Section 1       Imag	1 2 1 5 21 21 8 0	8 125,000 210 135 237	2% 2% 2% 1% 1% 0%	8 4 U 6
Apert laugus (25th - Storing concil - Apert laugus (25th - Storing concil + Exit Iva - Apert Sure: Anes: 		Image: Section 1         Image: Section 1           Image: Section 1	1 2 1 5 21 21 8 0 0	8 8 126,800 210 135 230 230 330 330	2% 2% 2% 1% 1% 0%	<b>6 1</b> 1
Chiper Bayder (25th - Storing concil Erity Chiper Bayder (25th - Storing concil E this fair fairs fairs fairs E this fair fairs fairs fairs E this fairs fairs fairs E this fa		Image: State of the state o	1 2 1 5 22 8 0 0 1 1	8 125,800 210 135 135 137 137	2% 2% 2% 1% 1% 0%	<b>6 1</b> 1
Constant PEAA - Evolve constant     Effect langtar PEAA - Evolve constant     Effect langt Source Process     Constant PEAA - Evolve constant     Evolve constant PEEAA     Constant PEEAAA     Constant PEEAAA     Constant PEEAAA     Constant PEEAAAA     Constant PEEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	Tos Undor Layot Hép Tos Undor Layot Hép P / P & P / Q Device Desp Device	Image: Control of the second secon	1 2 1 5 21 21 8 0 0	8 125,800 210 135 135 137 137	2% 2% 2% 1% 1% 0%	<b>6 1</b> 1
Chyser Bangtor P2304 - Scotter general Entyper Bangtor Source Process Entyper Bangtor Source Process Entyper Bangtor Source Process Entyper Bangtor Source Process Entyper Bangtor Bangtor Entyper Bangtor Bangtor Entyper Ba		Image: State of the state o	1 2 1 5 22 8 0 0 1 1	8 126,000 210 210 210 210 210 210 210 210 210	73 73 75 75 76 76 76 76 76 76 76 76 76	<b>6 1</b> 1
le ist Ven Proct Source Process n - O # X O O A s - O # X O O A Betroid Heatoly - O determine the Source - O anter, consistence - O anter, cons	inneratie tortes consistence of the constant o	Image: State of the state o	1 2 1 5 22 21 0 0 0 1 1 1 1 0 8	8 8 126,800 210 135 270 320 300 300	83 83 83 83 83 83 83 83 83 83 83 83 83 8	<b>6 1</b> 1

Number of OLOGICE2[OLOGICE3]OSERCESE2s

Number of PHASER, DI PHASER, DI PHAS

Number of PHASER\_OUT/PHASER\_OUT\_PHIs

Number of BSCAVIS

Number of BURHCEs

Number of BURRs

Design Sunnary

30 65

0%

05 z

65 5

65

+D8X

• 8 4 0 18 M

1 24 65

į 2

i

t 24

🗄 Enable Message Filtering

Show Clock Report

Show Errors

Show Failing Constraints Show Warrings

tional Design Summary Contents

- () Behavioral Check Syntax Simulate Behavioral Model

🌶 Start 💐 Design 🚺 Files 🐧 Libraries 👔

Br.4

A D D L L D D X D D		1.					
N RT IN LOND IN AN OLIVER A	1	• //88/8 🖉 🔂 581					
+08×	0	🖯 Design Overview 🔹 🔺	Intel Accentionare inserses	-	300	0%	
Vien: () 🗟 Indenentation 🖲 📓 Smulation 👘 👘 🕞 108 Properties		Number of PHIKSER, JN, PHIKSER, JN, PHIK	0	24	0%		
etarioral •	onal		Number of PHASER_OUT_PHASER_OUT_PH16	0	24	0%	
ierardy		Number of BSCANs	0	4	0%		
onfree_convolutionencoder	Ø	Pinout Report     Occk Report	Number of BUFHCEs	0	96	0%	
upmer_convolucinencade     unite convolucinencade     unite convolucinencade     unite convolucinencade     unite convolucinencade     unite convolucinencade     unite convolucinencade     unite convolucinencade		Static Timing	Number of BUFRs	٥	24	0%	
E wut - topfile (topfile.v)	1	G Errors and Warnings	Number of CAPTURES	0	1	0%	
gl - shiftregister (shiftregist	¥	a non manga	Number of DNA_PORTs	0	1	0%	
- 😧 g2 - mu321 (mu321.ii) - 🔽 a3 - mu41 (mu41.ii)	A	Translation Messages	Number of OSP44E1s	0	240	0%	
C P martineau		Map Messages	Number of BPUSE_USRs	0	1	0%	
		<ul> <li>Place and Route Messages</li> <li>Timing Messages</li> </ul>	Number of FRAME_ECCs	0	1	0%	
		- 🗍 Bitgen Messages	Number of ICAPs	0	2	0%	
No Processes Running		All Implementation Messages	Number of DELAYCTRUS	0	6	0%	
Processes uut - topfile	Synthesis Report	Number of DI_FEFOs	0	24	0%		
🐐 TSim Smulator		A fundain frend	Number of MICHE2_ADVs	0	6	0%	
8ehavioral Check Syntax Simulate Behavioral Model		Design Properties Enable Message Filtering	Number of OUT_FEFOs	0	24	0%	
Surger foreigne wide		Optional Design Summary Contents 	Number of PCIE_1_1s	0	1	0%	
			Number of PHASER_REFs	0	6	0%	
			Number of PHT_CONTROLs	6	6	0%	
			Number of PLIE2_ADVs	0	6	0%	
			Number of STARTUPs	0	1	0%	
			Number of XADCs	0	1	0%	





1

🔋 Console 🏮 Errors 🤰 Marrings 👔 Find in Files Results

A









Table 1: - Comparison of Resource Utilization for the
Architectures

Parameter	Conventional Encoder (K=1/3)	Reduced ROM XOR- FREE
FFs	12	08
LUTs	04	04
LUT-FF Pairs	08	03
BUFG/BUFGCTRLs	01	01
F <sub>max</sub> (MHz)	544.0	589.2

Table 2:- Comparison of Resource Utilization for the Architectures

Parameter	Conventional Encoder (K=1/2)	Reduced ROM XOR- FREE
FFs	10	09
LUTs	03	03
LUT-FF Pairs	10	08
BUFG/BUFGCTRLs	01	01
F <sub>max</sub> (MHz)	420.6	556.5

# **IV.CONCLUSION**

The improvement in delay can be done by the use of Convolution encoder Vedic multiplier. Which provide faster speed than the normal convolution encoder? Parallel generation of partial products and eliminates unwanted multiplication steps by this. A fast multiplication process and achieves a significantly less computational complexity over its conventional counterparts is allow by this algorithm.Viterbi decoder using parallel processing improves processing speed than normal viterbi decoder because the decoder do not need to wait trace back. It means trace back and decoder can simultaneously work. the design complexity will get reduced for inputs of large number of bits and will a provide faster speed. It will also used to design a PN sequence generator and spread spectrum modulation to improve the utilization of bandwidth.

# REFERENCES

- [1] G. Purohit, K. S. Raju, V. K. Chaubey, "A New XOR-Free Approach for Implementation of Convolutional Encoder" IEEE EMBEDDED SYSTEMS LETTERS, VOL. 8, NO. 1, MARCH 2016
- [2] J. Dielissen, Eindhoven, N. Engin, S. Sawitzki, and K. van Berkel, "Multistandard FEC decoders for wireless devices,"IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 3, pp. 284–288, Mar. 2008.
- [3] J. Hagenauer, "Forward error correcting for CDMA systems," in Proc. Int. Symp. Spread Spectr. Tech. Appl. Proc., Mainz, Sep. 1996, pp.566–569.



- [4] A. J. Viterbi, "Convolutional codes and their performance in communication systems,"IEEE Trans. Comm. Technol., vol. 19, no. 5, pp.751–772, Oct. 1971.
- [5] Y.Yibin, K.Roy, and R.Drechsler, "PowerconsumptioninXORbased circuits," inProc. ASP-DAC, Jan. 1999, pp. 299–302.
- [6] R. Pasko, P. Schaumont, V. Derudder, S.Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions,"IEEE Trans. Comput. Des. Integr. CircuitsSyst., vol. 18, no. 1, pp. 58–68, Jan. 1999
- [7] C. Huang, J. Li, and M. Chen, "Optimizing XOR-based codes," U.S. Patent 8209577 B2, Jun. 26, 2012.
- [8] H. J. Kang and I. C. Park, "A high-speed and low-latency Reed– Solomon decoder based on a dual-line structure," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., May 2002, vol. 3, pp. 3180–3183
- [9] H. Samueli, "An improved search algorithm for the design of multipli-erless FIR filters with powers-of-two coefficients,"IEEE Trans. CircuitsSyst., vol. 36, pp. 1044–1057, July 1989.
- [10] M. Potkonjak, M. B. Shrivasta, and P. A. Chandrakasan, "Multiple constant multiplication: Efficient and versatile framework and algorithms for exploring common subexpression elimination,"IEEE Trans.Computer-Aided Design, vol. 15, pp. 151–161, Feb. 1996.
- [11] M. Mehendale, S. D. Sherlekar, and G. Vekantesh, "Synthesis of multi-plierless FIR filters with minimum number of additions," in Proceedings of the 1995 IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos, CA: IEEE Computer Society Press, 1995, pp.668–671.
- [12] R. I. Hartley, "Sub expression sharing in filters using canonic signed digit multipliers," IEEE Trans. Circuits Syst. II, vol. 43, pp. 677–688,Oct. 1996.
- [13] A. G. Dempster and M. D. Mcleod, "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, vol. 42, pp.569–577, Sept. 1995.
- [14] D. R. Bull and D. H. Horrocks, "Primitive operator digital filters," Proc. Inst. Elect. Eng., vol. 138, pt. G, no. 3, pp. 401–411, June 1991.
- [15] Y. C. Lim and S. R. Parker, "Discrete coefficient fir digital filter design based upon an LMS criteria," IEEE Trans. Circuits Syst., vol. CAS-30, pp. 723–739, Oct. 1983.