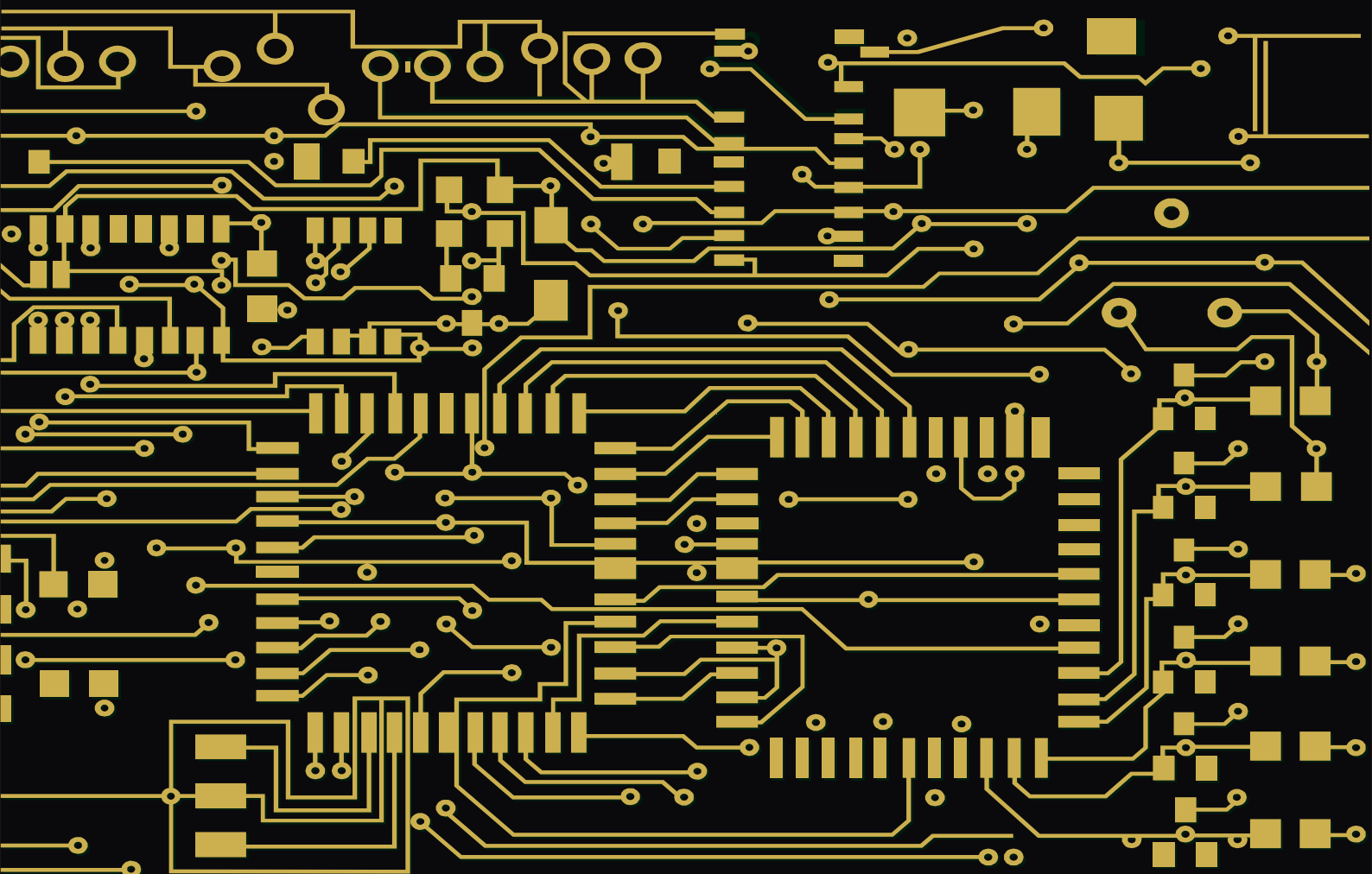


ISSN: 2153-1285 Vol.1, No.1, July 2010



Scientific
Research

Circuits and Systems



ISSN: 2153-1285



www.scirp.org/journal/cs

Journal Editorial Board

ISSN 2153-1285 (Print) ISSN 2153-1293 (Online)

<http://www.scirp.org/journal/cs>

Editor-in-Chief

Prof. Gyungho Lee Korea University, Korea (South)

Editorial Board

Prof. Radim Belohlavek	Palacky University, Czech
Prof. Dong-Ho Cho	KAIST, Korea (South)
Prof. John Choma	University of Southern California, USA
Dr. Bhaskar Choubey	University of Glasgow, UK
Prof. Wonzoo Chung	Korea University, Korea (South)
Prof. Aleksander Grytczuk	Western Higher School of Commerce and International Finance, Poland
Prof. Juwook Jang	Sogang University, Korea (South)
Prof. C. C. Ko	National University of Singapore, Singapore
Prof. Changhua Lien	Department of Marine Engineering, Taiwan (China)
Prof. Giuseppe Liotta	Universit�a degli Studi di Perugia, Italy
Dr. Zoltan Mann	Budapest University of Technology and Economics, Hungary
Prof. Shahram Minaei	Dogus University, Turkey
Prof. Fathi M. Salem	Michigan State University, USA
Prof. Victor Sreeram	University of Western Australia, Australia
Prof. Theodore B. Trafalis	The University of Oklahoma, USA
Prof. Fangang Tseng	National Tsing Hua University, Taiwan (China)
Prof. Chengchi Wang	Far East University, Taiwan (China)
Dr. Fei Zhang	Michigan State University, USA
Dr. Zuxing Zhang	Tohoku University, Japan

Editorial Assistant

Jane Xiong Scientific Research Publishing. Email: cs@scirp.org

TABLE OF CONTENTS

Volume 1 Number 1

July 2010

Two Simple Analog Multiplier Based Linear VCOs Using a Single Current Feedback Op-Amp

D. R. Bhaskar, R. Senani, A. K. Singh, S. S. Gupta.....1

Voltage Mode Cascadable All-Pass Sections Using Single Active Element and Grounded Passive

Components

J. Mohan, S. Maheshwari, D. S. Chauhan.....5

Fast Implementation of VC-1 with Modified Motion Estimation and Adaptive Block Transform

M. Tammen, M. El-Sharkawy, H. Sliman, M. Rizkalla.....12

FPGA Design of an Intra 16×16 Module for H.264/AVC Video Encoder

H. Loukil, I. Werda, N. Masmoudi, A. B. Atitallah, P. Kadionik.....18

The Design of an Intelligent Security Access Control System Based on Fingerprint Sensor FPC1011C

Y. Wang, H. L. Liu, J. Feng.....30

Circuits and Systems

Journal Information

SUBSCRIPTIONS

The *Circuits and Systems* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: \$50 per copy.

To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements

Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.

E-mail: sub@scirp.org

COPYRIGHT

Copyright© 2010 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assume no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: cs@scirp.org

Two Simple Analog Multiplier Based Linear VCOs Using a Single Current Feedback Op-Amp

Data Ram Bhaskar¹, Raj Senani^{2*}, Abdhesh Kumar Singh³, Shanti Swarup Gupta⁴

¹Department of Electronics and Communication Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India

²Division of Electronics and Communication Engineering, Netaji Subhas Institute of Technology, Azad Hind Fauj Marg, New Delhi, India

³Electronics and Communication Engineering Department, ITS Engineering College, Greater Noida, India

⁴Ministry of Commerce and Industry, Government of India, New Delhi, India

E-mail: bhaskar.ec@jmi.ac.in, senani@nsit.ac.in, abdheshks@yahoo.com, ss.gupta@nic.in

Received February 27, 2010; revised March 29, 2010; accepted April 5, 2010

Abstract

Two simple voltage-controlled-oscillators (VCO) with linear tuning laws employing only a single current feedback operational amplifier (CFOA) in conjunction with two analog multipliers (AM) have been highlighted. The workability of the presented VCOs has been demonstrated by experimental results based upon AD844 type CFOAs and AD534 type AMs.

Keywords: Voltage-Controlled Oscillators, Current Feedback Op-Amps, Current-Mode Circuits, Analog Multipliers

1. Introduction

Although a number of new building blocks and circuit concepts related to current-mode circuits have been investigated in recent literature [1-3], the use of current feedback operational amplifiers (CFOAs) as an alternative to the traditional voltage-mode op-amps (VOA), has attracted considerable attention (see [4-7] and the references cited therein) in various instrumentation, signal processing and signal generation applications due to their commercial availability as off-the-shelf ICs as well as due to the well known advantages offered by CFOAs over the VOAs. Because of these reasons, use of CFOAs has been extensively investigated in realizing oscillators, for instance, see [6,8-11] and the references cited therein. Although, a variety of CFOAs are available from various manufacturers, AD844 (from Analog Devices) which contains a CCII+ followed by a voltage buffer is particularly flexible and popular due to the availability of z-terminal of the CCII+ therein as an externally accessible lead which permits AD844 to be used as a CCII+ (one AD844) or as CCII- (employing two AD844s) or as a general 4-terminal building block [6].

Voltage-Controlled Oscillators (VCO) are important building blocks in several instrumentation, electronic and

communication systems, such as in function generators, in production of electronic music to generate variable tones, in phase locked loops and in frequency synthesizers [12-17].

A known method of realizing VCOs is to devise an RC-active oscillator configuration with two analog multipliers (AM) appropriately embedded to enable independent control of the oscillation frequency through an external control voltage V_C applied as a common multiplicative input to both the multipliers. When two AMs are appropriately embedded in such a configuration, this technique gives rise to a linear tuning law of the form

$$f_0 \propto V_C \quad (1)$$

Based upon this approach, a number of VCO configurations have been proposed by various researchers in the past [12,14-17] employing traditional VOAs and AMs.

A family of eight, CFOA-based linear VCOs of the above kind has recently been presented in [18]; however, all the circuits presented therein require two CFOAs along with two AMs. The main object of this communication is to highlight two simple linear VCOs of the above kind, which are realizable with only a single CFOA along with two AMs. Experimental results using AD844 CFOAs have been given and the advantages of the new

Choosing the same V_{ref} for both the multipliers, the circuits of **Figures 1(a), 1(b)** have the condition of oscillation (CO) and frequency of oscillation (FO) given by

$$\text{CO: } (C_1 - C_2) \leq 0 \quad (2)$$

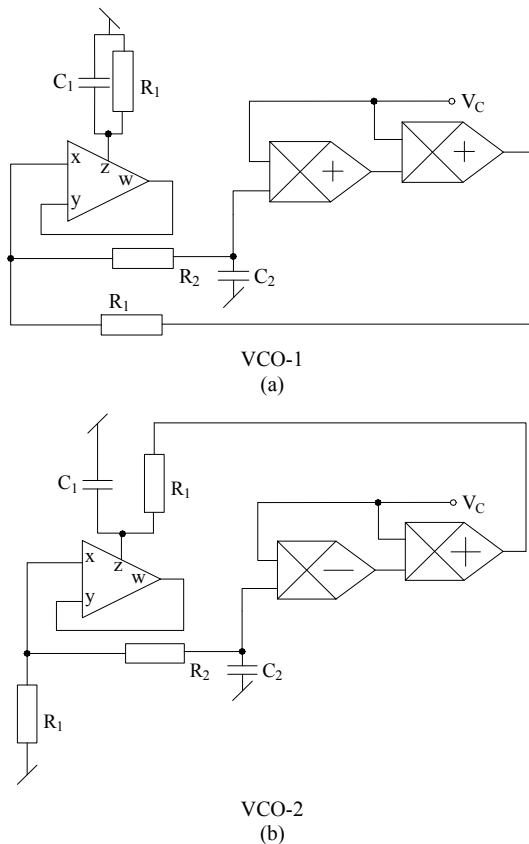


Figure 1. CFOA-based VCOs.

$$\text{FO: } f_0 = \frac{\beta}{2\pi} \sqrt{\frac{1}{C_1 C_2 R_1 R_2}}; \text{ where } \beta = \frac{V_C}{V_{ref}} \quad (3)$$

Thus, it is seen that f_0 is linearly controllable by the external control voltage V_C , as desired. However, as per Equation (2), one needs a variable capacitor to adjust CO.

3. Consideration of CFOA Parasitics

The prominent non-idealities of the CFOAs include—a finite non-zero input resistance r_x at port- x (typically around 50 Ω), y -port parasitics consisting of a parasitic resistance R_y (typically 2 M Ω) in parallel with a parasitic capacitance C_y (typically 2 pF) and z -port parasitic impedance consisting of a parasitic resistance R_p (typically 3 M Ω) in parallel with a parasitic capacitance C_p (typically, between 4-5 pF). In case of an analog multiplier, the finite non-zero output resistance r_{out} , as per datasheet of AD534, is merely 1 Ω and hence, can be ignored in all the cases. On the other hand, the input impedance of the AM, being 10 M Ω , is sufficiently high and hence, its effect can be ignored since usually, R_2 and $1/\omega C_2$ would be relatively much smaller over the frequency range of interest. For a quantitative assessment of the effect of the various CFOA non-idealities, we have carried out a re-analysis of both the VCOs and the nonideal expressions for CO and FO are given in **Tables 1** and **2** respectively. It may be noted that since CFOA has $y-w$ shorted, the y -port parasitics do not affect the operation of the circuits and hence, do not appear in the nonideal expressions of CO and FO.

It is easy to infer from the expressions given in **Tables 1** and **2** that the errors caused by the influence of CFOA can be kept small by choosing all external resistors to be much larger than r_x but much smaller than R_p and choosing both external capacitors to be much larger than C_{p1} and C_{p2} .

Table 1. Ideal and non-ideal conditions of oscillation.

VCO	Ideal CO	Nonideal CO
1	$C_1 = C_2$	$C_1 = C_2 \frac{\left\{ 1 - \frac{R_2}{R_p} - R_x \left(\frac{1}{R_1} + \frac{1}{R_p} \right) \left(1 + \frac{R_2}{R_1} \right) \right\}}{1 + R_x \frac{(1 - \beta^2)}{R_1}} - C_p$
2	$C_1 = C_2$	$C_1 = C_2 \frac{\left\{ 1 - \frac{R_2}{R_p} - R_x \left(\frac{1}{R_1} + \frac{1}{R_p} \right) \left(1 + \frac{R_2}{R_1} \right) \right\}}{1 + \frac{R_x}{R_1}} - C_p$

Table 2. Ideal and non-ideal frequency of oscillation.

VCO	Ideal FO	Nonideal FO
1	$\omega_0^2 = \frac{\beta^2}{C_1 C_2 R_1 R_2}$	$\omega_0^2 = \frac{\beta^2 - \frac{R_1}{R_p} + R_s(1 - \beta^2) \left(\frac{1}{R_1} + \frac{1}{R_p} \right)}{(C_1 + C_p) C_2 R_1 R_2 \left\{ 1 + R_s \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \right\}}$
2	$\omega_0^2 = \frac{\beta^2}{C_1 C_2 R_1 R_2}$	$\omega_0^2 = \frac{\beta^2 + \frac{R_1}{R_p} + R_s \left(\frac{1}{R_1} + \frac{1}{R_p} \right)}{(C_1 + C_p) C_2 R_1 R_2 \left\{ 1 + R_s \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \right\}}$

An inspection of **Tables 1** and **2** shows that including the non-ideal parameters of the CFOA will result in a condition of oscillation that can not be controlled without disturbing the frequency of oscillation. Thus, it appears that the advantage of the proposed circuits in providing non-interacting controls of CO and FO is lost. However, the same can be circumvented by proper selection of the values of the resistors. This limits the ease of the design process and affects the highest frequency realizable by the proposed circuits.

A quantitative comparison of f_0 calculated from non-ideal formula and values obtained practically is given in Section 5.

4. Frequency Stability Properties

Frequency stability is an important figure of merit for oscillators. The frequency stability factor S_F is defined as

$$S_F = \frac{d\phi(u)}{du} \text{ where } u = \frac{\omega}{\omega_0} \text{ is the normalized frequency and } \phi(u) \text{ is the phase of the open-loop transfer function of the oscillator circuit. The expressions for the frequency stability factors for both the VCOs for } C_1 = C_2 = C, R_1 = R \text{ and } R_2 = R/n \text{ have been derived and are found to be } -\frac{2\sqrt{n}}{n+1}.$$

It can be easily deduced that for $n = 1$, S_F is exactly 1 for both the VCOs. This figure is better than several classical oscillators such as Wein bridge oscillator and thus, both the proposed VCOs can be regarded to be quite satisfactory from this viewpoint.

5. Experimental Results

Both the VCOs have been experimentally studied using AD844 type CFOAs and AD534 type AMs biased with ± 15 volts DC power supplies and it has been possible to

generate oscillation frequencies from tens of kHz to several hundreds of kHz with tolerable errors in the frequency.

Experimental results of the proposed VCOs are shown in **Figures 2 (a)** and **2(b)**. The component values chosen were as under: For the circuits of **Figure 1**, $R_1 = R_2 = 1 \text{ k}\Omega$, $C_1 = 1 \text{ nF}$ and C_2 was taken as a variable capacitor to adjust the CO. **Figure 2(a)** shows the variation of oscillation frequency with control voltage V_C for the VCO of **Figure 1(a)** whereas **Figure 2(b)** shows a typical waveform ($f_0 = 68.7 \text{ kHz}$, $V_{0p-p} = 3 \text{ V}$) obtained from the VCO of **Figure 1(b)**.

The measured values of the THD for the two VCOs are found to be 1.88% and 1.52% respectively. The practical results show a reasonably good correspondence between the theoretical and experimental values and confirm the workability of the two VCOs.

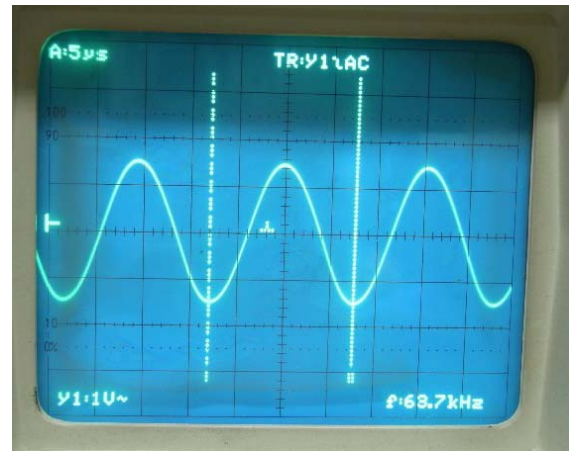
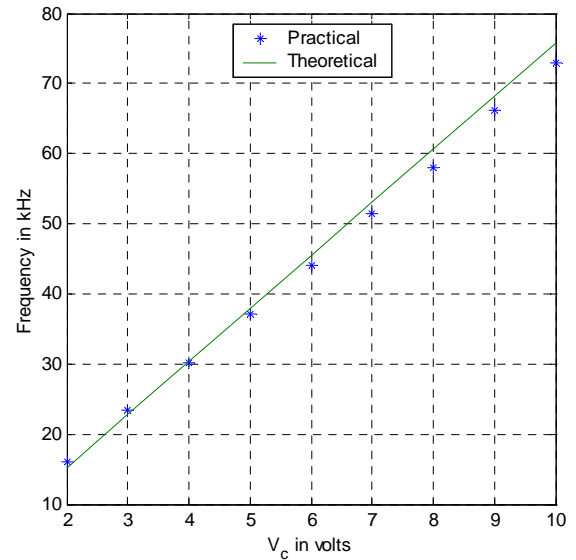


Figure 2. Experimental results of the VCOs: (a) Variation of frequency with VC for VCO-1; (b) A typical waveform generated from VCO-2 ($f_0 = 68.7 \text{ kHz}$, $V_{0p-p} = 3 \text{ V}$).

6. Concluding Remarks

In this communication, we highlighted two simple CFOA-AM-based VCOs, providing linear tuning laws, which are derivable from previously known op-amp-AM based VCOs published in [14]. The workability of the new VCOs has been confirmed by the experimental results, based upon AD844 type CFOAs and AD534 type analog multipliers.

A comparison with the previously known linear VCOs of [12,14-17] is now in order. The VCOs presented here have the advantage of requiring fewer AMs than those of [15], fewer resistors than those of [14,16,17] and fewer active elements than those of [15-17]. When the single-CFOA-RC VCOs are compared with single-op-amp-RC VCOs of [14] (**Figure 2** therein), they have the advantage of employing two less resistors. Furthermore, when compared to op-amp-RC VCOs of [12,14-17], the CFOA-RC VCOs presented here not only offer relatively higher operational frequency range (several hundred kHz as compared to only a few kHz available from the op-amp-RC VCOs) they also exhibit lower distortion level (THD being less than 2%). Lastly, when compared with the CFOA-AM-RC circuits presented recently in [18] all of which require two CFOAs, the circuits presented here have the advantage of employing only a single CFOA.

7. References

- [1] J. W. Horng, "Current Conveyors Based All Pass Filters and Quadrature Oscillators Employing Grounded Capacitors and Resistors," *Computers & Electrical Engineering*, Vol. 31, No. 1, 2005, pp. 81-92.
- [2] A. Toker, O. Cicekoglu and H. Kuntman, "On the Oscillator Implementations Using a Single Current Feedback Op-Amp," *Computers & Electrical Engineering*, Vol. 28, No. 5, 2002, pp. 375-389.
- [3] A. M. Soliman and A. S. Elwakil, "Wien Oscillators Using Current Conveyors," *Computers & Electrical Engineering*, Vol. 25, No. 1, 1999, pp. 45-55.
- [4] C. Toumazou and F. G. Lidgey, "Current Feedback Op-Amps: A Blessing in Disguise?" *IEEE Circuits Devices Magazine*, Vol. 10, No. 1, 1994, pp. 34-37.
- [5] F. J. Lidgey and K. Hayatleh, "Current-Feedback Operational Amplifiers and Applications," *Electronics and Communication Engineering Journal*, Vol. 9, No. 4, 1997, pp. 176-182.
- [6] R. Senani, "Realization of a Class of Analog Signal Processing/Signal Generation Circuits: Novel Configurations Using Current Feedback Op-Amps," *Frequenz*, Vol. 52, No. 9-10, 1998, pp. 196-206.
- [7] A. M. Soliman, "Applications of the Current Feedback Amplifier," *Analog Integrated Circuits and Signal Processing*, Vol. 11, No. 3, 1996, pp. 265-302.
- [8] R. Senani and S. S. Gupta, "Synthesis of Single-Resistance-Controlled Oscillators Using CFOAs: Simple State-Variable Approach," *IEE Proceedings on Circuits, Devices and Systems*, Vol. 144, No. 2, 1997, pp. 104-106.
- [9] S. S. Gupta and R. Senani, "State Variable Synthesis of Single of Single-Resistance-Controlled Grounded Capacitor Oscillators Using only two CFOAs: Additional New Realizations," *IEE Proceedings on Circuits, Devices and Systems*, Vol. 145, No. 6, 1998, pp. 415-418.
- [10] A. M. Soliman, "Current-Feedback Operational Amplifier Based Oscillators," *Analog Integrated Circuits and Signal Processing*, Vol. 23, No. 1, 2000, pp. 45-55.
- [11] D. R. Bhaskar and R. Senani, "New CFOA-Based Single-Element-Controlled Sinusoidal Oscillators," *IEEE Transactions on Instrumentation and Measurement*, Vol. 55, No. 6, 2006, pp. 2014-2021.
- [12] R. Senani, D. R. Bhaskar and M. P. Tripathi, "On the Realization of Linear Sinusoidal VCOs," *International Journal of Electronics*, Vol. 74, No. 5, 1993, pp. 727-733.
- [13] R. Senani and D. R. Bhaskar, "New Active-R Sinusoidal VCOs with Linear Tuning Laws," *International Journal of Electronics*, Vol. 80, No. 1, 1996, pp. 57-61.
- [14] D. R. Bhaskar and M. P. Tripathi, "Realization of Novel Linear Sinusoidal VCOs," *Analog Integrated Circuits and Signal Processing*, Vol. 24, No. 3, 2000, pp. 263-267.
- [15] S. K. Saha, "Linear VCO with Sine Wave Output," *IEEE Transactions on Instrumentation and Measurement*, Vol. 35, No. 2, 1986, pp. 152-155.
- [16] S. K. Saha and L. C. Jain, "Linear Voltage Controlled Oscillator," *IEEE Transactions on Instrumentation and Measurement*, Vol. 37, No. 1, 1988, pp. 148-150.
- [17] V. P. Singh and S. K. Saha, "Linear Sinusoidal VCO," *International Journal of Electronics*, Vol. 65, No. 2, 1988, pp. 243-247.
- [18] D. R. Bhaskar, R. Senani and A. K. Singh, "Linear Sinusoidal VCOs: New Configurations Using Current Feedback Op-Amps," *International Journal of Electronics*, Vol. 97, No. 3, 2010, pp. 263-272.

Voltage Mode Cascadable All-Pass Sections Using Single Active Element and Grounded Passive Components

Jitendra Mohan^{1*}, Sudhanshu Maheshwari², Durg S. Chauhan³

¹*Department of Electronics and Communications, Jaypee University of Information Technology,
Solan, India*

²*Department of Electronics Engineering, Z. H. College of Engineering and Technology,
Aligarh Muslim University, Aligarh, India*

³*Department of Electrical Engineering, Institute of Technology, Banaras Hindu University,
Varanasi, India*

E-mail: jitendramv2000@rediffmail.com

Received April 23, 2010; revised May 27, 2010; accepted June 2, 2010

Abstract

In this paper, four new first order voltage mode cascadable all-pass sections are proposed using single active element and three grounded passive components, ideal for IC implementation. The active element used is a fully differential current conveyor. All the proposed circuit possess high input and low output impedance feature which is a desirable feature for voltage-mode circuits. Non-ideality aspects and parasitic effects are also given. As an application, a multiphase oscillator is designed. The proposed circuits are verified through PSPICE simulation results using TSMC 0.35 μm CMOS parameters.

Keywords: Analogue Signal Processing, All-Pass Filter, Current Conveyor, Voltage-Mode

1. Introduction

Voltage mode (VM) active filters with high-input and low-output impedance are of great interest because several cells of this kind can be directly connected in cascade within voltage-mode systems without additional voltage buffers. On the other hand, the use of grounded capacitor is beneficial from the point of integrated circuit implementation and also having less parasitic compared to floating counterparts [1].

First order all-pass filters are an important class of analogue signal processing circuits which have been extensively researched in the technical literature [2,3] due to their utility in communication and instrumentation systems, for instance as a phase equalizer, phase shifter or for realizing quadrature oscillators band pass filters etc. Numerous first-order voltage-mode all-pass sections (VM-APSs) employing different types of active element such as current conveyors and its different variations have been reported in the literature [4-24]. Among the cited references, several VM-APSs employ a single standard current conveyor [6-16,19-22]. Such circuits aim at realizing the first order all-pass function using optimum number of passive components, rather using grounded components or offering high input and low output impedance

feature together. The circuits reported in [17,18] enjoy high input impedance but uses two active element and three grounded passive components. Some of the circuits described in [15,21-23] fall in the separate category of tunable, resistorless realizations, the most recent of these [22,23] enjoys high input and/or low output impedance. A recent published all-pass filter circuit in [24] employ two DVCCs and two passive components with the advantage of high input impedance and low output impedance, which is ideal for cascading, but it still suffer from floating resistor. But a careful survey reveals that none of the reported works realizes a VM-APS using single active element, grounded passive components, and also providing high input impedance and low output impedance features simultaneously.

This paper proposes four new first order VM cascadable all-pass sections, with high input and low output impedance using single active element and three grounded passive components, which are ideal for IC implementation. Each circuit employs two grounded resistors and one grounded capacitor. The proposed circuits are based on fully differential second generation current conveyor (FDCCII), an active element to improve the dynamic range in mixed mode application, where fully differential signal processing is required [25]. As an application, the

proposed circuit realizes a multiphase oscillator. PSPICE simulation results using TSMC 0.35 μm CMOS parameters are given to validate the circuits.

The paper is organized as follows: in Section 2, the proposed all-pass filters using FDCCII are presented. In Section 3, parasitic and non-ideal analyses of the proposed circuits are given. In Section 4, to verify the theoretical study the first order all-pass filters were constructed and simulated with PSPICE program. In Section 5, a multi-phase oscillator is implemented to show the usefulness of the proposed circuits as an illustrating example and finally the conclusion in Section 6.

2. Proposed Circuit

The FDCCII is an eight terminal analog building block with a describing matrix equation of the form [25]

$$\begin{bmatrix} I_{Y1} \\ I_{Y2} \\ I_{Y3} \\ I_{Y4} \\ V_{X+} \\ V_{X-} \\ I_{Z+} \\ I_{Z-} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{Y1} \\ V_{Y2} \\ V_{Y3} \\ V_{Y4} \\ I_{X+} \\ I_{X-} \\ V_{Z+} \\ V_{Z-} \end{bmatrix} \quad (1)$$

The symbol and CMOS implementation of FDCCII

are shown in **Figure 1** [25]. The Y_1 , Y_2 , Y_3 , and Y_4 terminals are high-impedance terminals, while $X+$ and $X-$ terminals are low-impedance ones. The $Z+$ and $Z-$ terminals are high impedance nodes suitable for current outputs. FDCCII is a useful and versatile active element for analog signal processing. The applications of FDCCII in filters and oscillators design using only grounded passive components were demonstrated in [26,27].

The voltage transfer function of an all-pass filter can be given as

$$\frac{V_{OUT}}{V_{IN}} = K \frac{s\tau - 1}{s\tau + 1} \quad (2)$$

where K is the gain constant and its sign determines whether phase shifting is from 0° to -180° or from 180° to 0° , and τ is the time constant. The four proposed first order VM cascable all-pass sections using a single FDCCII and three grounded passive components are shown in **Figures 2(a)-2(d)**. The four circuits in **Figures 2 (a)-2(d)** are characterized by the voltage transfer function as

$$\frac{V_{OUT}}{V_{IN}} = K \left(\frac{s - (1/C_1)[(1/R_2) - (1/R_1)]}{s + (1/R_1 C_1)} \right) \quad (3)$$

For $R_2 = R_1/2$ Equation (3) becomes

$$\frac{V_{OUT}}{V_{IN}} = K \frac{sR_1 C_1 - 1}{sR_1 C_1 + 1} \quad (4)$$

where the value of $K = +1$ for Circuit-I and Circuit-II (**Figures 2(a)** and **2(b)**) and the value of $K = -1$ for Circuit-III and Circuit-IV (**Figures 2 (c)** and **2(d)**).

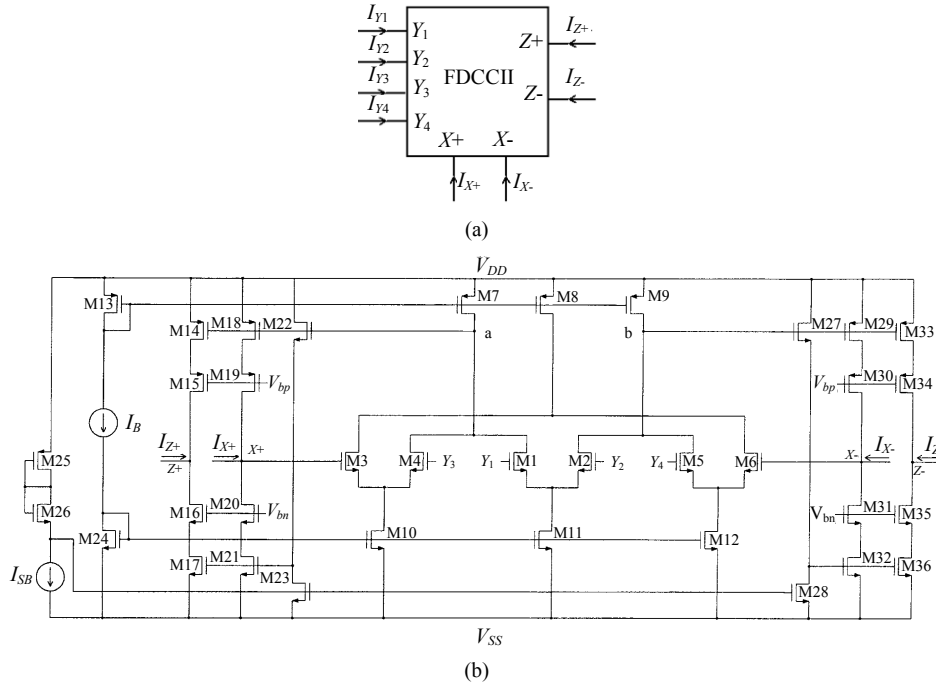


Figure 1. Fully differential second generation current conveyor (a) symbol (b) CMOS implementation [25].

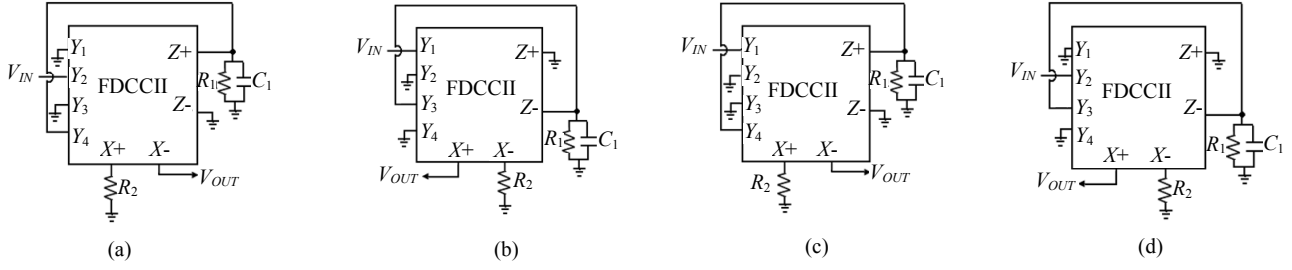


Figure 2. Proposed first order cascable all-pass filters (a) circuit-I; (b) circuit-II; (c) circuit-III and (d) circuit-IV.

The salient features of the four proposed circuits are high input and low output impedance, single active element and use of grounded passive components; the three features are not exhibited together in any of the available works, including the most recent circuits [4-24]. It may be noted that the new circuits are based on a topology with input at Y and output at one of the X terminals, the other X -terminal being terminated by a resistor. The four proposed circuits with similar properties being derivable from a topology are a result of the versatility of FDCCII.

It is also worth mentioning that four additional new circuits can further be obtained from the proposed circuits by replacing the resistor (R_2) with a capacitor (C_2). However these circuits would employ a capacitor at X terminal, thus degrading high frequency operation. This aspect will not be further elaborated for brevity reasons.

3. Parasitic and Non-Ideal Analysis

3.1. Parasitic Effects

A study is next carried out on the effects of various parasitic of the FDCCII used in the proposed circuits. These are port Z parasitic in form of R_Z/C_Z , port Y parasitic in form of R_Y/C_Y and port X parasitic in form of series resistance R_X [13]. The proposed circuits are reanalyzed taking into account the above parasitic effects. The voltage transfer function (assuming $R \ll R_Y$ or R_Z and $R_X \ll R$), for the circuits of **Figures 2(a)-2(d)**, is given as

$$\frac{V_{OUT}}{V_{IN}} = K \left(\frac{s - (1/(C_1 + C_p))[(1/R') - (1/R_1)]}{s + (1/R_1(C_1 + C_p))} \right) \quad (5)$$

where, $R' = R_2 + R_X$, (for **Figures 2(a)-2(d)**), $K = +1$, (for **Figures 2(a)** and **2(b)**), $K = -1$, (for **Figures 2(c)** and **2(d)**), and $C_p = C_{Z+} + C_{Y4}$ (for **Figures 2(a)** and **2(c)**), and $C_p = C_{Z-} + C_{Y3}$ (for **Figures 2(b)** and **2(d)**).

From (5), it is seen that the gain is unity and the pole-frequency is

$$\omega_o = \frac{1}{R_1(C_1 + C_p)} \quad (6)$$

From (5), the parasitic resistance/capacitances merge with the external value. Such a merger does cause slight

deviation in circuit's parameters, which can be eliminated by pre-distorting the element values to be used in the circuit. It is seen that the pole-frequency would be slightly deviated (in deficit) due to these parasitics. The deviation is expected to be small for an integrated FDCCII; the actual value would be given in the 'simulation results'.

3.2. Non-Ideal Analysis

Taking the non-idealities of the FDCCII into account, the relationship of the terminal voltages and currents can be rewritten as

$$\begin{bmatrix} I_{Y1} \\ I_{Y2} \\ I_{Y3} \\ I_{Y4} \\ V_{X+} \\ V_{X-} \\ I_{Z+} \\ I_{Z-} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \beta_1 & -\beta_2 & \beta_3 & 0 & 0 & 0 & 0 & 0 \\ -\beta_4 & \beta_5 & 0 & \beta_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{Y1} \\ V_{Y2} \\ V_{Y3} \\ V_{Y4} \\ I_{X+} \\ I_{X-} \\ V_{Z+} \\ V_{Z-} \end{bmatrix} \quad (7)$$

where α_i ($i = 1, 2$) accounts for current transfer gains and β_i ($i = 1, 2, 3, 4, 5, 6$) accounts for voltage transfer gains of the FDCCII. These transfer gains differ from unity by the voltage and current tracking errors of the FDCCII. More specifically, $\alpha_i = 1 - \delta_i$, ($|\delta_i| \ll 1$) δ_1 is the current tracking error from $X+$ to $Z+$ and δ_2 is the current tracking error from $X-$ to $Z-$. Similarly, $\beta_i = 1 - \varepsilon_i$, ($|\varepsilon_i| \ll 1$) where, voltage tracking errors are ε_1 (from Y_1 to $X+$), ε_2 (from Y_2 to $X+$), ε_3 (from Y_3 to $X+$), ε_4 (from Y_1 to $X-$), ε_5 (from Y_2 to $X-$), and ε_6 (from Y_4 to $X-$). The circuits of **Figures 2(a)-2(d)** are reanalyzed using (7) and the non-ideal voltage transfer functions are found as

Circuit-I:

$$\frac{V_{OUT}}{V_{IN}} = \beta_5 \left(\frac{s - [(\alpha_1 \beta_2 \beta_6 R_1 - \beta_3 R_2) / \beta_5 C_1 R_1 R_2]}{s + (1 / C_1 R_1)} \right) \quad (8)$$

Circuit-II:

$$\frac{V_{OUT}}{V_{IN}} = \beta_1 \left(\frac{s - [(\alpha_2 \beta_3 \beta_4 R_1 - \beta_1 R_2) / \beta_1 C_1 R_1 R_2]}{s + (1 / C_1 R_1)} \right) \quad (9)$$

Circuit-III:

$$\frac{V_{OUT}}{V_{IN}} = -\beta_4 \left(\frac{s - [(\alpha_1 \beta_1 \beta_6 R_1 - \beta_4 R_2) / \beta_4 C_1 R_1 R_2]}{s + (1 / C_1 R_1)} \right) \quad (10)$$

Circuit-IV:

$$\frac{V_{OUT}}{V_{IN}} = -\beta_2 \left(\frac{s - [(\alpha_2 \beta_3 \beta_5 R_1 - \beta_2 R_2) / \beta_2 C_1 R_1 R_2]}{s + (1 / C_1 R_1)} \right) \quad (11)$$

From (8)-(11), it is seen that the pole-frequency is unaltered by FDCCII non-idealities for all the transfer functions of the respective circuit, but the filters gain are slightly modified due to the FDCCII non-idealities. Thus the pole-frequency sensitivity to the FDCCII nonidealities is zero, and the filters gain sensitivity to these non-idealities is found within unity in magnitude. This suggests a good sensitivity performance for the proposed circuits.

4. Simulation Results

To verify theoretical result the proposed filter circuits were simulated by the PSPICE simulation program. The FDCCII was realized based on the CMOS implementation as shown in **Figure 1** [25] and simulated using TSMC 0.35 μm , level 3 MOSFET parameters as listed in **Table 1**. The aspect ratio of the MOS transistors are listed in **Table 2**, with the following DC biasing levels $V_{dd} = -V_{ss} = 3.3$ V, $V_{bp} = V_{bn} = 0$ V, and $I_B = I_{SB} = 1.7$ mA. The circuit-I (**Figure 2(a)**) was designed with $C_1 = 50$ pF,

Table 1. 0.35 μm level 3 MOSFET parameters.

NMOS:
LEVEL = 3 TOX = 7.9E - 9 NSUB = 1E17
GAMMA = 0.5827871 PHI = 0.7 VTO = 0.5445549 DELTA = 0
UO = 436.256147 ETA = 0 THETA = 0.1749684
KP = 2.055786E - 4 VMAX = 8.309444E4 KAPPA = 0.2574081
RSH = 0.0559398 NFS = 1E12 TPG = 1 XJ = 3E - 7
LD = 3.162278E - 11 WD = 7.04672E - 8 CGDO = 2.82E - 10
CGSO = 2.82E - 10 CGBO = 1E - 10 CJ = 1E - 3 PB = 0.9758533
MJ = 0.3448504 CJSW = 3.777852E - 10 MJSW = 0.3508721
PMOS:
LEVEL = 3 TOX = 7.9E - 9 NSUB = 1E17
GAMMA = 0.4083894 PHI = 0.7 VTO = -0.7140674
DELTA = 0 UO = 212.2319801 ETA = 9.999762E - 4
THETA = 0.2020774 KP = 6.733755E - 5 VMAX = 1.181551E5
KAPPA = 1.5 RSH = 30.0712458 NFS = 1E12 TPG = -1
XJ = 2E - 7 LD = 5.000001E - 13 WD = 1.249872E - 7
CGDO = 3.09E - 10 CGSO = 3.09E - 10 CGBO = 1E - 10
CJ = 1.419508E - 3 PB = 0.8152753 MJ = 0.5
CJSW = 4.813504E - 10 MJSW = 0.5

$R_1 = 2$ k Ω and $R_2 = 1$ k Ω . The theoretical designed pole frequency was 1.59 MHz. The phase and gain plots are shown in **Figure 3**. The phase is found to vary with frequency from 180° to 0° with a value of 90° at the pole frequency, and the pole frequency was found to be 1.54 MHz, which is close to the theoretical value. The circuit was next used as a phase shifter introducing 90° shift to a sinusoidal voltage input of 1 Volt peak at 1.59 MHz. The input and output waveforms are given in **Figure 4** which verify the circuit as a phase shifter. The THD variation at the output for varying signal amplitude at 1.59 MHz was also studied and the results shown in **Figure 5**. The THD for a wide signal amplitude (few mV-1V) variation is found within 4.27% at 1.59 MHz. The Fourier spectrum of the output signal, showing a high selectivity for the applied signal frequency (1.59 MHz) is also shown in **Figure 6**. Both theoretical and simulated pole frequencies are found to closely match; the discrepancy (deficit) in simulated frequency being the result of various parasitic discussed in Section 3.

Table 2. Transistor aspect ratios for the circuit shown in Figure 1.

Transistors	W(μm)	L(μm)
M1-M6	60	4.8
M7-M9, M13	480	4.8
M10-M12, M24	120	4.8
M14,M15,M18,M19,M25,M29,M30,M33,M34	240	2.4
M16,M17,M20,M21,M26,M31,M32,M35,M36	60	2.4
M22,M23,M27,M28	4.8	4.8

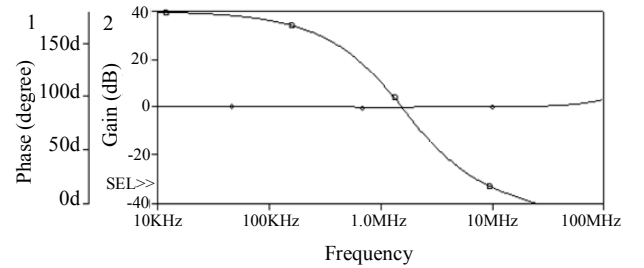


Figure 3. Gain and phase responses for the circuit-I.

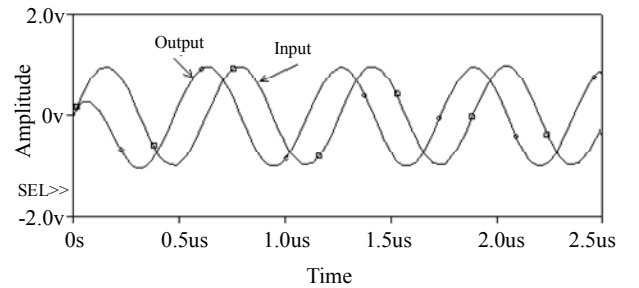


Figure 4. Input/output waveshapes for Circuit-I at 1.59 MHz.

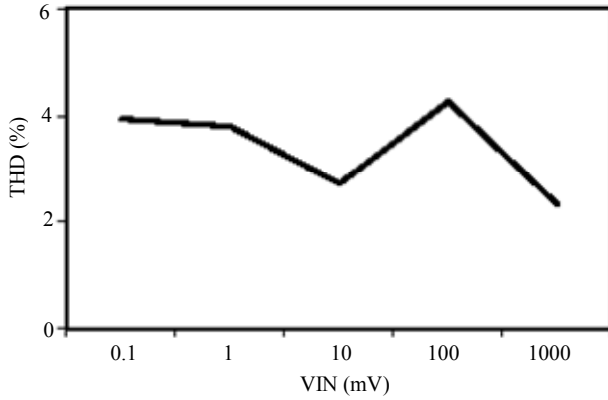


Figure 5. THD variation at output with signal amplitude at 1.59 MHz.

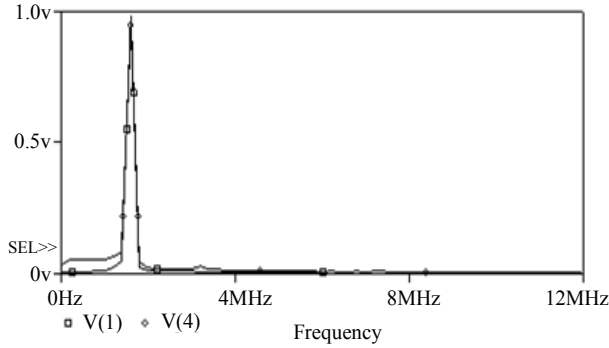


Figure 6. Fourier spectrum of Input-output signal at 1.59 MHz.

Similarly, the circuit-III (Figure 2(c)) was designed with the same values and frequency as above. The phase and gain plots are shown in Figure 7. The phase is found to vary with frequency from 0 to -180° with a value of -90° at the pole frequency, and the pole frequency was found to be 1.54 MHz, which is close to the theoretical value. The circuit was next used as a phase shifter introducing -90° shift to a sinusoidal voltage input of 1 Volt peak at 1.59 MHz. The input and output waveforms are given in Figure 8 which verify the circuit as a phase shifter.

5. Application Example

To further illustrate the utility of the proposed circuits a sinusoidal oscillator producing a number of quadrature signals was realized using the Circuit-I (Figure 2(a)). By connecting Y_2 terminal, the input node to the Z-terminal of FDCCII, connecting resistor (R') and capacitor (C') at X^- and Z^- terminals of FDCCII. The resulting circuit is shown in Figure 9. The circuit analysis yields the following characteristic equation

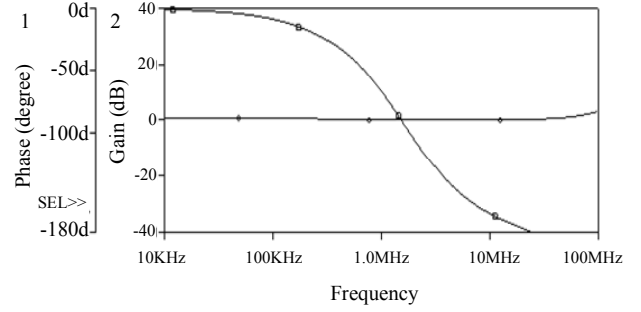


Figure 7. Gain and phase responses for the circuit-III.

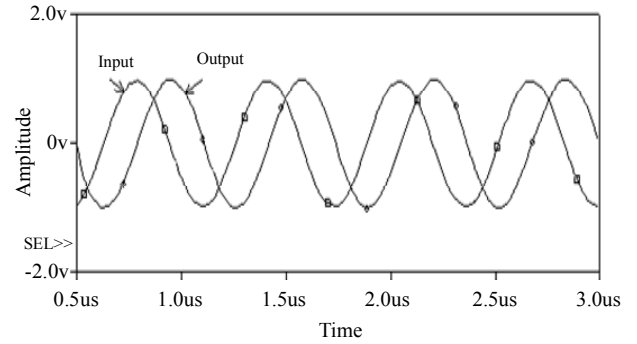


Figure 8. Input/output waveshapes for Circuit-III at 1.59 MHz.

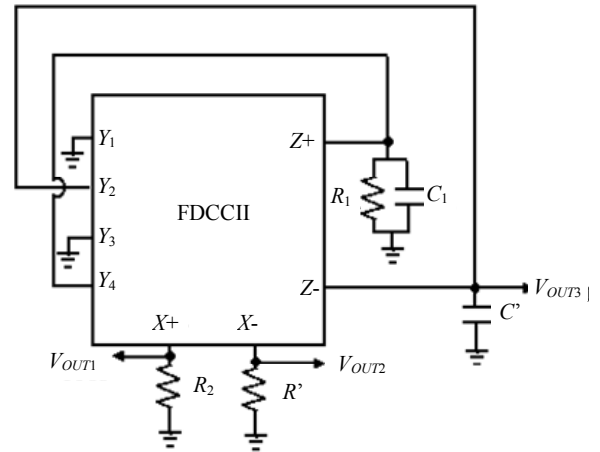


Figure 9. Multiphase oscillator using Circuit-I of Figure 2(a).

$$s^2 + s \left[\frac{1}{C_1 R_1} - \frac{1}{C' R'} \right] + \frac{R_1 - R_2}{C_1 C' R_1 R_2 R'} = 0 \quad (12)$$

At the frequency of oscillation, with $s = j\omega$, the Equation (5) gives the frequency of oscillation (FO) and condition of oscillation (CO) as

$$FO: \omega_o = \sqrt{\frac{R_1 - R_2}{C_1 C' R_1 R_2 R'}}; CO: C' R' \geq C_1 R_1 \quad (13)$$

Assuming $R' = R_1 = 2R_2$; $C' = C_1$

$$FO : \omega_o = \frac{1}{2C'R'} \quad (14)$$

From **Figure 9**, at oscillating frequency the circuit provides three quadrature voltage outputs (V_{OUT1} , V_{OUT2} and V_{OUT3}) whose phasor relationship is shown in **Figure 10**.

The circuit was designed with $C_1 = C' = 50$ pF, $R_1 = R' = 2$ k Ω , and $R_2 = 1$ k Ω , the theoretical frequency of oscillation was around $f_o = 1.59$ MHz, whereas the simulated values as found from the result was $f_o = 1.54$ MHz. The quadrature oscillations are shown in **Figure 11**.

6. Conclusions

This paper has presented four new first-order VM cascadable all-pass sections, each employing single FDCCII, two resistors and one capacitor. The salient features of all the proposed circuits are high input and low output impedance, single active element and use of grounded passive components. The proposed circuits with grounded components in each case are suited for IC implementation in CMOS technology. Non-ideality aspects and parasitic effects are also studied. The circuits are verified through PSPICE simulations using TSMC 0.35 μ m CMOS parameters. Application example in the form of multiphase oscillator is also given and also verified with good results. The proposed circuits are ideal for voltage-mode applications as well as good for IC implementation, making them a future prospect for integration.

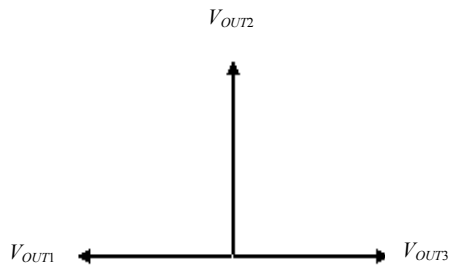


Figure 10. Phasor diagram of multiphase oscillator.

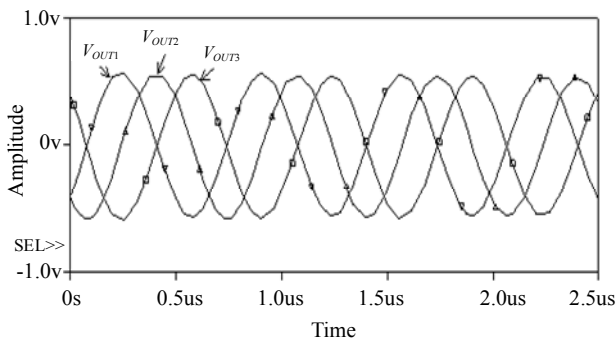


Figure 11. Quadrature voltage outputs of multiphase oscillator.

7. Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments.

8. References

- [1] M. Bhusan and R. W. Newcomb, "Grounding of Capacitors in Integrated Circuits," *Electronics Letters*, Vol. 3, No. 4, 1967, pp. 148-149.
- [2] D. Bielek, R. Senani, V. Biolkova and Z. Kolka, "Active Elements for Analog Signal Processing: Classification, Review, and New Proposals," *Radioengineering*, Vol. 17, No. 4, 2008, pp. 15-32.
- [3] S. J. G. Gift, "The Application of All-Pass Filters in the Design of Multiphase Sinusoidal Systems," *Microelectronics Journal*, Vol. 31, No. 1, 2000, pp. 9-13.
- [4] A. M. Soliman, "Inductorless Realization of an All-Pass Transfer Function Using the Current Conveyor," *IEEE Transactions on Circuits Theory*, Vol. 20, 1973, pp. 80-81.
- [5] A. M. Soliman, "Generation of Current Conveyor-Based All-Pass Filters from Op Amp-Based Circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 44, No. 4, 1997, pp. 324-330.
- [6] M. Higashimura and Y. Fukui, "Realization of All-Pass Network Using a Current Conveyor," *International Journal of Electronics*, Vol. 65, No. 22, 1988, pp. 249-250.
- [7] O. Cicekoglu, H. Kuntman and S. Berk, "All-Pass Filters Using a Single Current Conveyor," *International Journal of Electronics*, Vol. 86, No. 8, 1999, pp. 947-955.
- [8] I. A. Khan and S. Maheshwari, "Simple First Order All-Pass Section Using a Single CCII," *International Journal of Electronics*, Vol. 87, No. 3, 2000, pp. 303-306.
- [9] A. Toker, S. Özcan, H. Kuntman and O. Çiçekoglu, "Supplementary All-Pass Sections with Reduced Number of Passive Elements Using a Single Current Conveyor," *International Journal of Electronics*, Vol. 88, No. 9, 2001, pp. 969-976.
- [10] M. A. Ibrahim, H. Kuntman and O. Cicekoglu, "First-Order All-Pass Filter Canonical in the Number of Resistors and Capacitors Employing a Single DDCC," *Circuits, Systems, and Signal Processing*, Vol. 22, No. 5, 2003, pp. 525-536.
- [11] N. Pandey and S. K. Paul, "All-Pass Filters Based on CCII- and CCCII-," *International Journal of Electronics*, Vol. 91, No. 8, 2004, pp. 485-489.
- [12] K. Pal and S. Rana, "Some New First-Order All-Pass Realizations Using CCII," *Active and Passive Electronic Component*, Vol. 27, No. 2, 2004, pp. 91-94.
- [13] S. Maheshwari, I. A. Khan and J. Mohan, "Grounded Capacitor First-Order Filters Including Canonical Forms," *Journal of Circuits, Systems and Computers*, Vol. 15, No. 2, 2006, pp. 289-300.
- [14] J. W. Horng, C. L. Hou, C. M. Chang, Y. T. Lin, I. C.

- Shiu and W. Y. Chiu, "First-Order All-Pass Filter and Sinusoidal Oscillators Using DDCCs," *International Journal of Electronics*, Vol. 93, No. 7, 2006, pp. 457-466.
- [15] S. Minaei and O. Cicekoglu, "A Resistorless Realization of the First Order All-Pass Filter," *International Journal of Electronics*, Vol. 93, No. 3, 2006, pp. 177-183.
- [16] H. P. Chen and K. H. Wu, "Grounded Capacitor First Order Filter Using Minimum Components," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Science*, Vol. E89-A, No. 12, 2006, pp. 3730-3731.
- [17] S. Maheshwari, "High Input Impedance VM-APSS with Grounded Passive Elements," *IET Circuits Devices and Systems*, Vol. 1, No. 1, 2007, pp. 72-78.
- [18] S. Maheshwari, "High Input Impedance Voltage Mode First Order All-Pass Sections," *International Journal of Circuit Theory and Application*, Vol. 36, No. 4, 2008, pp. 511-522.
- [19] S. Maheshwari, "Analog Signal Processing Applications Using a New Circuit Topology," *IET Circuits Devices Systems*, Vol. 3, No. 3, 2009, pp. 106-115.
- [20] B. Metin and O. Cicekoglu, "Component Reduced All-Pass Filter with a Grounded Capacitor and High Impedance Input," *International Journal of Electronics*, Vol. 96, No. 5, 2009, pp. 445-455.
- [21] D. Biolek and V. Biolkova, "Allpass Filter Employing one Grounded Capacitor and one Active Element," *Electronics Letters*, Vol. 45, No. 16, 2009, pp. 807-808.
- [22] D. Biolek and V. Biolkova, "First Order Voltage-Mode All-Pass Filter Employing One Active Element and One Grounded Capacitor," *Analog Integrated Circuit and Signal Processing*, Vol. 45, No. 16, 2009, pp. 807-808.
- [23] T. Tsukutani, H. Tsunetsugu, Y. Sumi and N. Yabuki, "Electronically Tunable First-Order All-Pass Circuit Employing DVCC and OTA," *International Journal of Electronics*, Vol. 97, No. 3, 2010, pp. 285-293.
- [24] S. Minaei and E. Yuce, "Novel Voltage-Mode All-Pass Filter Based on Using DVCCs," *Circuits, Systems, and Signal Processing*, Vol. 29, No. 3, 2010, pp. 391-402.
- [25] A. A. El-Adway, A. M. Soliman and H. O. Elwan, "A Novel Fully Differential Current Conveyor and its Application for Analog VLSI," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 47, No. 4, 2000, pp. 306-313.
- [26] C. M. Chang, B. M. Al-Hashimi, C. I. Wang and C. W. Hung, "Single Fully Differential Current Conveyor Biquad Filters," *IEE Proceedings on Circuits, Devices and Systems*, Vol. 150, No. 5, 2003, pp. 394-398.
- [27] J. W. Horng, C. L. Hou, C. M. Chang, H. P. Chou, C. T. Lin and Y. Wen, "Quadrature Oscillators with Grounded Capacitors and Resistors Using FDCCIs," *ETRI Journal*, Vol. 28, No. 4, 2006, pp. 486-494.

Fast Implementation of VC-1 with Modified Motion Estimation and Adaptive Block Transform

Michael Tammen¹, Mohamed El-Sharkawy², Hisham Sliman¹, Maher Rizkalla¹

¹Purdue School of Engineering and Technology, Indianapolis, USA

²Egypt Japan University of Science and Technology, Borg Elarab, Alexandria, Egypt

E-mail: melshark@iupui.edu

Received March 18, 2010; revised April 20, 2010; accepted April 25, 2010

Abstract

The Society of Motion Picture and Television Engineers (SMPTE) Standard 421M, commonly known as VC-1, is a state-of-the-art video compression format that provides highly competitive video quality, from very low through very high bit rates, at a reasonable computational complexity. First, this paper presents fast motion compensation methods. The four motion estimation methods examined are fast, three step search, varying diamond, and 2D logarithmic. These methods use less search points than the full spiral scan used in the VC-1 reference software, which allows for faster motion estimation. Second, this paper presents a residual texture based choice of the block size for the Discrete Cosine Transform (DCT). To determine the block size, data is examined after the residual texture has been calculated. This is in contrast to the VC-1 reference software, which uses calculations at the block level to determine the block size. The residual texture of each block is small and uniform, allowing for simplified block choices.

Keywords: VC-1, Motion Estimation, Discrete Cosine Transform, Video Compression

1. Introduction

VC-1 is a state-of-the-art video compression format that provides highly competitive video quality, from very low through very high bit rates, at a reasonable computational complexity [1-4]. At a high level, VC-1 is similar to other popular video standards since the First Moving Picture Expert Group Standard (MPEG-1). They have similarities in many different areas, one of which is block-by-block motion compensation. The motion compensation is done using a motion vector from a previously reconstructed frame to determine the displacement. On the decoder side, quantized transform coefficients are entropy-decoded, dequantized, and inverse-transformed to produce an approximation of the residual error, which is then added to the motion-compensated prediction to generate the reconstruction [5].

An important feature of VC-1 is adaptive block size transforms for inter-frame coding. Instead of using a fixed block size for every transform, like many previous standards, it has the ability to choose a transform size based on the information in the block or macroblock. VC-1 can choose 8×8 , 8×4 , 4×8 , or 4×4 transforms based on the information in the block. The ability to choose a transform size allows VC-1 to deal with areas of conti-

nunity and discontinuity with more accuracy and less ringing artifacts than any other video standard [1]. An 8×8 block may be transformed by either one 8×8 block, two horizontally stacked 8×4 s, two vertically stacked 4×8 s, or four 4×4 blocks. The block sizes can be seen in Figure 1.

Another key innovation in VC-1 is the handling of an entire zero block or macroblock. When it happens, it does not send any transform information along, since the inverse transform of a zero block is going to be zero. Intra frames and intra blocks in predicted frames use 8×8 transform by default, meaning no calculations need to be preformed [5].

The way VC-1 signals the transform size is new and unique. It can either be signaled at the frame, macroblock, or block level. For frame level signaling, every block within the frame will use the same block size. Frame level signaling is useful for low-rate situations to keep the overhead low. If macroblock level signaling is used, then

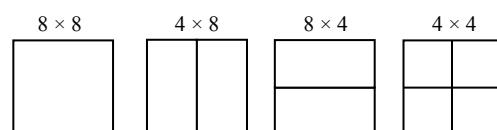


Figure 1. Transform sizes.

every block in the macroblock (six 8×8 blocks) is transformed using the same size. Block level signaling is used only for one 8×8 block. These different signaling types allow VC-1 to handle both nonstationary data (macroblock and block level signaling) and low-rate situations (frame level signaling) better than previous standards.

Currently, VC-1 uses half difference and half sum energies for the transform size decision for a block. It splits the block into four quadrants: left, right, top, and bottom. The left-right half difference, the left-right half sum, the top-bottom half difference, and the top-bottom half sum are then computed. Each of the above values is a running total for all the values in the quadrant.

To get the block size, it compares the left-right sum to the left-right difference and determines to chop vertically or not. Then it compares the top-bottom sum and top-bottom difference to determine if it should chop horizontally. If both conditions are met a 4×4 is used, while if none are met an 8×8 is used.

Motion estimation (ME) is the process of comparing the macroblock to be coded with all macroblocks within the search area in the reference frame [6-10]. The macroblock in the reference frame which is closest to the macroblock to be coded is chosen as the reference macroblock. Once it is chosen, a motion vector (MV) is assigned. The motion vector provides the coordinates of the best match in an (x, y) form, using the center of the search area as (0, 0). An illustration can be seen below in Figure 2.

2. Motion Estimation Techniques

The VC-1 reference software currently uses the full spiral search shown in Figure 3 (a search size of 4 is used in the figure) to find the best match. A best match is found by calculating the cost at each location. The spot

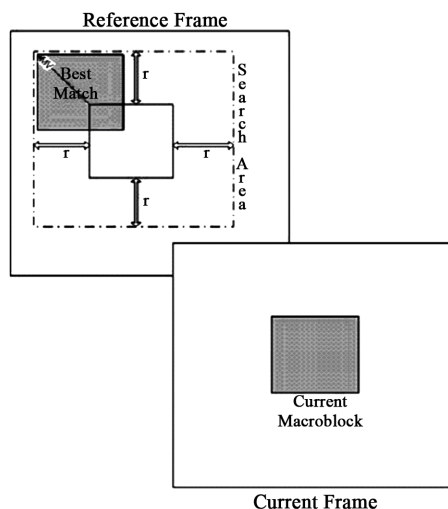


Figure 2. Motion estimation [11].

with the lowest cost is chosen as the best match. This is currently the only method for motion estimation within VC-1 reference software. It does allow for the search size to be modified but that is all.

The first implemented method was the fast motion estimation. It is based on the H.264 fast motion estimation but is not completely similar. It calculates the cost of each position in a diamond shape. The size of the diamond grows or shrinks based on the search size. The pattern is shown below in Figure 4 with a search size of 4. Also, the cost for the center is calculated before this search starts and if no cost in the fast search is lower than the center, then the center is chosen.

Second, we implemented a three or four step search. Figure 5 shows the search pattern for this search. First,

	-4	-3	-2	-1	0	1	2	3	4
4	74	75	76	77	78	79	80	81	50
3	73	44	45	46	47	48	49	26	51
2	72	43	22	23	24	25	10	27	52
1	71	42	21	8	9	2	11	28	53
0	70	41	20	7	1	3	12	29	54
-1	69	40	19	6	5	4	13	30	55
-2	68	39	18	17	16	15	14	31	56
-3	67	38	37	36	35	34	33	32	57
-4	66	65	64	63	62	61	60	59	58

Figure 3. Full spiral search.

	-4	-3	-2	-1	0	1	2	3	4
4					×				
3				×		×			
2			×				×		
1		×						×	
0	×								×
-1		×						×	
-2			×				×		
-3				×		×			
-4					×				

Figure 4. Fast motion estimation.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
6									2		2		2
5													
4			1			1		2		1			2
3							3	3	3				
2							3	2	3	2			2
1								3	3	3			
0			1			0					1		
-1													
-2													
-3													
-4			1			1					1		
-5													
-6													

Figure 5. Three step search.

the cost is computed at the center (0) and eight positions (1) around the center. The positions distance from the center is calculated by 2^{N-1} , where N is 3 if the search size is less than or equal to 12, or 4 if the search size is greater than 12. This allows for better accuracy when the search size is large, because the normal three step search will only be able to search about half of the search area if the search size is greater than 12. The least cost position is chosen (**1**) and N is decremented. The cost is then calculated at eight positions (2) around **1**. This again allows for another least cost position to be picked (**2**) and N is decremented again. Lastly, eight positions (3) have their cost calculated around **2** and the least cost position is chosen as the best match, **3**. A four step search would simply have one more step and works the same way. Obviously, the best match is not always going to require three or four steps, so if at the end of any step the best cost belongs to the center of the search (0, **1**, and **2** for reference) the search terminates. For example, if the cost of 0 is less than the cost at each of the eight positions (1) then the search terminates and 0 is chosen as the best match.

Third, a 2D log search was implemented. **Figure 6** is an illustration of how the log search works. To determine the distance from the center in which the search is started, $d = \text{floor}(2 \times (\log_2 S - 1))$, is computed, where S is the search size. The cost is then calculated at the center (0) and four other positions (1) surrounding it. The position with the least cost is chosen (**1**) and d remains the same. Next, the cost is computed at the three surrounding positions (2) and if their cost is not lower than the center then d is halved. If d is an odd number, 1 is subtracted from it and then it is halved.

The last search we implemented was the varying diamond search. This borrows ideas from the three step search, but instead of using eight positions it uses four. The option of a three step or a four step is allowed in this search as well. **Figure 7** illustrates what a varying diamond search will look like when it uses three steps. The bold positions are those with the lowest cost, and the lower the number, the earlier it is searched.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
6													
5													
4													
3							1						
2													
1													
0				1			0			1			
-1													
-2						3	4						
-3				2	3	1	3	4	2				
-4						3	4						
-5													
-6							2						

Figure 6. 2D log search.

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
6													
5													
4							1						
3													
2			2										
1													
0	2		1		2		0			1			
-1			3										
-2		3	2	3									
-3			3										
-4							1						
-5													
-6													

Figure 7. Varying diamond search.

3. Modified Adaptive Block Size Transform

To go along with the motion estimation, a modified adaptive block size transform algorithm is implemented. The current method for choosing a transform size has some drawbacks. Multiple calculations are performed at the block level, which is very costly.

Through research, we discovered that the block size is dominated by the residual, which is the prediction error [12]. Using (1), the residual data is computed. The texture that results is small in magnitude and uniform. Compared to the original data, there is a large difference since the original data may not be uniform. Since the data is uniform the block size selection is simplified.

We assume that motion estimation has been performed. The algorithm is as follows:

- 1) Initialize residual texture threshold T_8 and T_4 for 8×8 and 4×4 blocks. We perform motion estimation for the current macroblock to get the residual. We assign T_8 and T_4 to 250 and 50 respectively.
- 2) Partition the macroblock into 24 (4×4) blocks.
- 3) Calculate the residual block texture of each 4×4 block using Equation (1).

$$C_{m,n} = \frac{1}{16} \sum_{i=0}^3 \sum_{j=0}^3 (x_{i,j} - \bar{x})^2 \quad (1)$$

where m is the number of 8×8 blocks and n is 4×4 block number in the 8×8 block, beginning in the upper-left and moving right.

- 4) Sum all the 4×4 blocks' texture in an 8×8 block as the 8×8 block's texture. The macroblock texture is the sum of all 6 of the 8×8 blocks' texture.

$$C_m = \sum_{n=0}^3 C_{m,n} \quad (2)$$

$$C_{MB} = \sum_{m=0}^3 C_m \quad (3)$$

- 5) Create a count of each 4×4 residual block whose texture is greater than the threshold T_4 and record it in

count8. Also track the position of each block that is over the threshold in the variable *Track_b4*. *Track_b4* contains the value of 1-4 based on which is the most recent block to be over the threshold value.

6) The block size selection is done through the following method:

a) If *count4* = 0, 8×8 is chosen for the block size.

b) If *count4* > 1 and $C_m > T_8$, 4×4 is chosen as the block size.

c) If *track_b4* = 0 or 3 and $C_1 \geq C_2$ choose 4×8 , otherwise choose 8×4 .

d) If *track_b4* = 1 or 2 and $C_3 \leq C_0$ choose 8×4 , otherwise choose 4×8 .

The above procedure is carried out in two different steps in the same subroutine. First, a subroutine is called that partitions the macroblock and carries out all calculations. This includes calculating the residual texture for the macroblock, as well as the sum of all the blocks' texture. The code then enters a loop, which is executed 6 times, once for each block in the macroblock, where the transform size is selected and the block is transformed. The subroutine where the transform size is selected does not involve any calculations like the standard. It simply gets a pointer to the residual texture results and performs step 6. Not performing any calculations at the block level allows for greater speed.

4. Results

The motion estimation techniques and modified adaptive block size transform were implemented on the VC-1 sample encoder. A variety of video clips were chosen to make sure that both changes work on different types of video sequences. Some of the sequences have low detail and low movement, while others have medium detail and low movement or medium movement and low detail. All sequences are run 30 fps, and can be seen in **Table 1**.

The cases covered in **Table 1** give varieties of video clips that range from low to high movement and low to high details. More frames are needed for higher movement to cover information depth between frames. **Table 2** shows the key configurations we used for the encoder. It is important to note that the modified adaptive block size transform will run only if the option 'BlockType' is set to 'Any'. If 'random' or 'iterated' is set in the option file, then the subroutine is never called by the encoder due to the transform size being set. We tested all sequences with a search size of 8 and 14, hence both numbers below. These were two separate tests and will be noted later. All of the other important parameters were set to 'random' or 'iterated'.

All sequences were tested to determine the time taken for the motion estimation and adaptive block size transform, the signal-to-noise ratio (SNR) for the luminance and chrominance components, and the bitrate. The re-

sults of the motion estimation changes are compared against those of the standard. **Table 3** summarizes the results with a search size of 8, while **Table 4** summarizes the results with a search size of 14. Both tables use percentages to show how much faster the proposed algorithms are than the standard. It is noted that a speed improvement of up to 38% for search size equals to 14 and 18% for search size equals to 8.

Table 1. Video sequences tested.

Index	Sequence	Frames
1	Akiyo (CIF)	300
2	Coastguard (CIF)	300
3	Foreman (QCIF)	300
4	Stefan (CIF)	300
5	Flower (CIF)	250
6	Carphone (QCIF)	90
7	Football (CIF)	90
8	Erik (CIF)	50

Table 2. Key encoder options.

Parameter	Value
Profile:	Advanced
Level:	L0
Frame Pattern:	I:P:P:P
SearchSize:	8 or 14
DQuant:	1
MVRange:	128×64
BlockType:	Any
Quantizer:	Explicit

Table 3. Time results with a search size of 8.

Sequence	Percent Saving (%)			
	Fast	TSS	Diamond	2DLog
1	12.31	12.38	13.46	13.04
2	23.88	24.55	25.73	24.68
3	15.04	15.42	16.11	15.48
4	21.78	22.22	22.88	21.72
5	19.75	20.80	21.97	21.08
6	10.03	11.24	10.67	10.97
7	21.81	22.84	23.82	21.95
8	13.86	13.77	14.54	14.02
Average	17.31	17.90	18.64	17.87

Table 4. Time results with a search size of 14.

Sequence	Percent Saving (%)			
	Fast	TSS	Diamond	2DLog
1	32.75	33.53	34.45	34.39
2	44.10	45.60	46.68	46.35
3	33.17	34.23	35.04	34.63
4	42.05	43.30	44.23	43.56
5	40.67	42.68	43.58	43.07
6	27.64	28.55	28.87	29.30
7	43.28	44.74	45.87	45.03
8	32.12	32.85	33.98	33.53
Average	36.97	38.18	39.09	38.73

Tables 3 and 4 show that the diamond search is the strongest performer followed closely by the 2D log search and three step searches. The fast motion estimation is the worst performer of the group. So strictly from a speed aspect, the diamond search is the best choice.

Next, we will examine the signal-to-noise ratio (SNR) results for search sizes of 8 and 14. **Table 5** shows the SNR for the reference software. **Tables 6 and 7** show the percent degradation in the implemented algorithm for search sizes of 8 and 14 respectively. The negative values in the tables show that the SNR is better in the implemented algorithm.

The VC-1 encoder in its present state has an overflow issue with some of the sequences. The most reliable data comes from sequences 1, 3, and 6 due to little or no overflow occurring. Sequence 1 is the most reliable, as no overflow occurs during the encoding of this sequence. Therefore, using sequence 1 as the baseline we see that there is a very minimal SNR degradation. But, on average all searches see an improvement in SNR.

Lastly, we will look at the bitrates for all of the sequences. The bitrate numbers can get rather large so we examine the percent degradation numbers. Once again if the number is negative that means the proposed algorithm is better than the standard. **Tables 8 and 9** show the results below.

Aside from sequence 6 there is a very minimal effect on the bitrate. There is a range of both small increases from the standard as well as small decreases, neither of which are above 1% (excluding the 1.79% in sequence 4). As for sequence 6, the bitrate is 9% larger than the standard. The reason of such result will be studied in future work.

5. Conclusions

In this paper, we presented four different motion estimation techniques and a fast block size selection tool for

Table 5. SNR results for the standard with both search sizes.

Sequence	SNR (dB)					
	Search Size 8			Search Size 14		
	Y	U	V	Y	U	V
1	21.31	20.78	17.07	21.31	20.78	17.07
2	19.30	25.72	24.88	19.29	25.71	24.90
3	15.39	23.81	24.10	15.39	23.81	24.10
4	24.89	35.04	36.05	24.84	35.07	35.91
5	28.11	22.78	21.37	28.09	23.45	22.65
6	10.06	18.51	16.24	10.06	18.52	16.25
7	22.34	26.75	26.06	22.34	26.95	26.08
8	18.64	20.18	21.50	18.64	20.18	21.52

Table 6. SNR results with a search size of 8.

Sequence	Percent Degradation (%)											
	Fast			TSS			Diamond			2DLog		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
1	0.28	-0.27	-0.16	0.16	-0.11	-0.09	0.20	-0.17	-0.16	0.19	-0.16	-0.23
2	-2.48	-2.10	0.76	-2.56	-1.04	1.36	-3.30	-1.20	1.59	-3.52	-1.71	-1.01
3	0.48	-0.71	-1.17	0.20	-0.46	-0.83	0.34	-0.48	-0.85	0.33	-0.52	-0.90
4	-5.70	-4.51	-3.79	-6.77	-3.92	-3.93	-6.13	-5.02	-4.30	-5.83	-4.54	-2.91
5	-2.82	0.50	-2.00	-2.92	0.68	0.75	-2.94	0.25	0.73	-2.91	0.80	0.83
6	0.55	0.31	-0.98	0.04	0.70	-0.60	0.13	1.04	-0.96	0.17	0.52	-0.53
7	-0.58	2.15	2.02	-0.88	3.11	-1.43	-0.65	1.74	-1.61	-0.24	0.96	0.27
8	-4.08	-1.46	-3.91	-3.99	-1.47	-3.56	-3.74	-0.37	-2.69	-3.81	0.11	-2.46
Average	-1.79	-0.76	-1.15	-2.09	-0.31	-1.04	-2.01	-0.53	-1.03	-1.95	-0.57	-0.87

Table 7. SNR results with a search size of 14.

Sequence	Percent Degradation (%)											
	Fast			TSS			Diamond			2DLog		
	Y	U	V	Y	U	V	Y	U	V	Y	U	V
1	0.27	-0.13	-0.28	0.17	-0.02	0.15	0.20	-0.20	-0.20	0.20	-0.47	-0.28
2	-4.63	-0.98	0.82	-2.91	-1.37	1.40	-2.91	-1.07	1.44	-2.86	-1.18	1.10
3	0.42	-0.52	36.36	0.21	-0.50	-0.83	0.33	-0.53	-0.85	0.42	-0.52	36.36
4	-7.45	-3.90	26.66	-6.87	-3.75	-4.28	-4.94	-4.44	-3.56	-7.45	-3.90	26.66
5	-3.24	3.08	-27.7	-2.99	3.66	6.33	-3.01	3.32	6.17	-3.24	3.08	-27.7
6	0.29	0.58	38.17	0.21	0.65	-0.76	0.32	0.55	-0.90	0.29	0.58	38.17
7	-0.11	1.81	14.13	-0.50	1.83	0.43	-0.76	2.68	-1.20	-0.11	1.81	14.13
8	-3.87	-0.08	10.08	-4.30	-0.88	-3.54	-3.93	-1.55	-3.32	-3.87	-0.08	10.08
Average	-2.29	-0.02	12.28	-2.12	-0.05	-0.18	-1.84	-0.16	-0.30	-2.08	-0.09	12.32

Table 8. Bitrate results with a search size of 8.

Sequence	Percent Degradation (%)			
	Fast	TSS	Diamond	2DLog
1	0.34	0.20	0.09	0.11
2	-0.24	-0.25	-0.25	-0.29
3	-0.01	-0.01	0	0.01
4	0.01	-0.07	0	0
5	-0.27	-1.47	-0.03	0.26
6	9.68	10.15	8.81	8.37
7	0.09	0.05	0.26	0.06
8	0.49	0.49	0.37	0.48
Average	1.26	1.14	1.16	1.13

Table 9. Bitrate results with a search size of 14.

Sequence	Percent Degradation (%)			
	Fast	TSS	Diamond	2DLog
1	0.33	0.22	0.12	0.13
2	-0.06	-0.08	-0.08	0.16
3	-0.01	0	-0.01	-0.01
4	0.21	-0.47	-0.48	1.79
5	-0.51	-1.70	0.02	0.02
6	9.90	10.11	8.77	8.02
7	-0.28	-0.31	-0.29	-0.28
8	0.40	0.40	0.20	0.18
Average	1.25	1.02	1.03	1.25

VC-1. Finding an accurate and fast set of motion vectors is very important to any video codec. Also, adaptive block size transforms are a vital part of any next generation video standard. Overall, the best performer was the diamond search which was followed closely by the three and four step search. Both were fast with the diamond having an 18.64% and 39.09% increase in speed with a 0.98% increase in the SNR. The three and four step search had a 17.90% and 38.18% increase in speed with a 0.96% increase in the SNR. Both exhibited an increase in bitrate around 1.10%, which is mainly due to sequence 6. Otherwise they would be much closer to 0.

6. References

- [1] S. Srinivasan and S. L. Regunathan, "An Overview of VC-1", *Proceedings of SPIE, VCIP*, Beijing, July 2005, pp. 720-728.
- [2] Proposed SMPTE 421M, "VC-1 Compressed Video Bitstream Format for Decoding Process," 2006. <http://www.smpte.org>
- [3] Proposed SMPTE RP 227, "VC-1 Bitstreams Transport Encoding," 2007. <http://www.smpte.org>
- [4] Proposed SMPTE RP 228, "VC-1 Decoder and Bitstreams Conformance," 2008. <http://www.smpte.org>
- [5] S. L. Regunathan, A. M. Rohaly, R. Crinon and P. Griffiths, "Quality and Compression: The Proposed SMPTE Video Compression Standard VC-1," *SMPTE Motion Imaging Journal*, Vol. 114, No. 5-6, 2005, pp. 194-201.
- [6] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," *IEEE Transactions on Communications*, Vol. 38, No. 7, 1990, pp. 950-953.
- [7] P. I. Hosur and K. K. Ma, "Motion Vector Field Adaptive Fast Motion Estimation," *Presented at the 2nd International Conference on Information, Communications and Signal Processing*, Singapore, December 1999, pp. 7-10.
- [8] R. Li, B. Zeng and M. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, No. 4, 1994, pp. 438-442.
- [9] L. K. Liu and E. Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 4, 1996, pp. 419-422.
- [10] L.-M. Po and W.-C. Ma, "A Novel Four-Step Search Algorithm for Fast Block Matching," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, No. 3, 1996, pp. 313-317.
- [11] S. T. Samant and M. El-Sharkawy, "Modified Motion Vector Searches for H.264/AVC," *International Conference on Computer Engineering & Systems (ICCES'06)*, Cairo, Egypt, November 2006, pp. 331-336.
- [12] Z. Wang, Q. Peng and Y. Zeng, "Residual Texture Based Fast Block-Size Selection for Inter-Frame Coding in H.264/AVC," *IEEE Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies*, Dalian, 2005, pp. 853-855.

FPGA Design of an Intra 16×16 Module for H.264/AVC Video Encoder

Hassen Loukil¹, Imen Werda¹, Nouri Masmoudi¹, Ahmed Ben Atitallah², Patrice Kadionik³

¹University of Sfax, National School of Engineering, Sfax, Tunisia

²University of Sfax, High Institute of Electronics and Communication, Sfax, Tunisia

³IMS laboratory-ENSEIRB-MATMECA-University Bordeaux I-CNRS UMR 5218, 351 Cours de la Libération, Talence Cedex, France

E-mail: Nouri.Masmoudi@enis.rnu.tn

Received May 16, 2010; revised June 18, 2010; accepted June 23, 2010

Abstract

In this paper, we propose novel hardware architecture for intra 16×16 module for the macroblock engine of a new video coding standard H.264. To reduce the cycle of intra prediction 16×16 , transform/quantization, and inverse quantization/inverse transform of H.264, an advanced method for different operation is proposed. This architecture can process one macroblock in 208 cycles for all cases of macroblock type by processing 4×4 Hadamard transform and quantization during 16×16 prediction. This module was designed using VHDL Hardware Description Language (HDL) and works with a 160 MHz frequency using ALTERA NIOS-II development board with Stratix II EP2S60F1020C3 FPGA. The system also includes software running on an NIOS-II processor in order to implementing the pre-processing and the post-processing functions. Finally, the execution time of our HW solution is decreased by 26% when compared with the previous work.

Keywords: Nios H.264, FPGA, Intra 16×16 , NIOS-II, SOPC Design

1. Introduction

Currently, video system development is generally based on embedded systems. Such systems need to find a compromise between computational complexity and timing execution constraints. On the other hand, the H.264/AVC standard for video compression [1-5], due to its high complexity, needed powerful processors and hardware acceleration in order to respect application requirements.

In order to take advantages of hardware acceleration, each functional module of the H.264 video encoder has been carefully studied in order to determine its computational complexity. Furthermore, the intra process presents one of the highest computational complexities in H.264/AVC encoder [6]. This process is based on the hybrid encoding scheme shown in **Figure 1** which uses the intra prediction, integer cosine transform and quantization. The intra process is used to remove spatial redundancy. There are two types of intra modes: intra 4×4

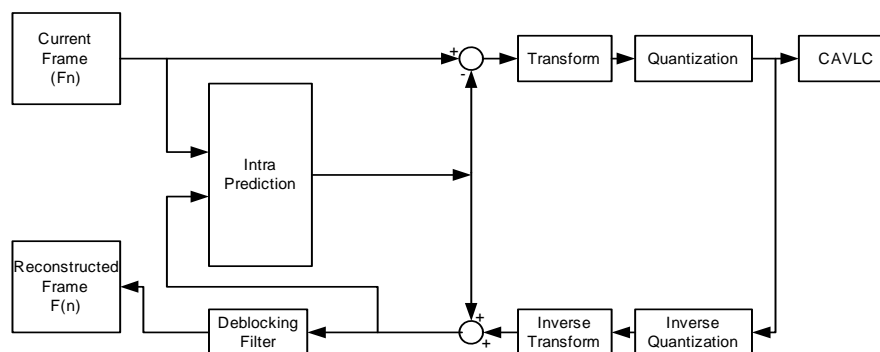


Figure 1. Hybrid encoder for video compression.

and intra 16×16 modes. The intra 16×16 is composed of intra 16×16 prediction (IP 16×16), integer cosine transform (ICT), quantization AC (QAC), inverse integer cosine transform (IICT), inverse quantization AC (IQ-AC), quantization DC (QDC), Hadamard transform (HT), inverse quantization DC (IQDC) and inverse Hadamard transform (IHT). Special hardware implementations of intra 16×16 for H.264 have been proposed [7,8]. They were shown that some of these parts can be optimized with parallel hardware structures implemented into the hardware system. These previous works have implemented the intra 16×16 algorithm with serial [7] and parallel [8] architectures directly into hardware device. But, our architecture uses both a parallel and pipelined structures in order to reduce the number of operations and the ability to achieve fast execution. Our design is described with VHDL (*VHSIC Hardware Description Language*) language and has been synthesized with the Altera NIOS II softcore processor for experimental validation into a single Altera Stratix II EP2S60 FPGA (Field Programmable Gate Array) device.

This paper is organized as follows: Section 2 presents an overview of intra 16×16 algorithm. In the next Section, we present the intra 16×16 architecture. The experiment results are shown in Section 4. Finally, Section 5 concludes the paper.

2. Overview of the Intra 16×16 Algorithm

The intra 16×16 algorithm is a critical component used in the H.264/AVC. There are eleven functional operations in this module: intra 16×16 prediction, residual calculation, integer transform, AC coefficient quantization, DC coefficient quantization, inverse AC coefficient quantization, inverse DC coefficient quantization, Hadamard transform, inverse Hadamard transform, inverse integer transform and pixel reconstruction. The 16×16 intra prediction mode is designed according to directions: vertical, horizontal, DC and plane modes are specified in the H.264 standard based on the reconstituted pixels from the previous macroblock (MB). **Figure 2** shows the intra 16×16 prediction mode.

For each MB, we compute the difference between the predicted pixel and the original pixel. After this step, we calculate the integer transform coefficients. In the H.264/AVC standard, the equation of the 4×4 integer transform is defined by [3,4].

$$I = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \times$$

$$\begin{bmatrix} X_0 & X_1 & X_2 & X_3 \\ X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} \\ X_{12} & X_{13} & X_{14} & X_{15} \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \quad (1)$$

“ X_i ” is the residual 4×4 block.

After this operation, we obtain two coefficients types: AC and DC coefficients. For the AC coefficients, we compute the quantization operation. In general the AC quantization operation is defined by [3,4].

$$Z_{ij} = \text{round}(I_{ij} \frac{PF}{QStep}) \quad (2)$$

We can write (5) as follows:

$$Z_{ij} = \text{round}(I_{ij} \frac{MF}{2^{qbits}}) \quad (3)$$

where:

$$\frac{MF}{2^{qbits}} = \frac{PF}{Qstep} \quad (4)$$

$$qbits = 15 + \text{floor}(QP/6) \quad (5)$$

I_{ij} is the uncalled coefficients after ICT for QAC. PF represents the scaling factor of the integer transform and QStep is the quantization step size. A total of 52 values of QStep are supported by the standard as shown in **Table 1** where QStep doubles in size for every 6 values of the step of quantization QP.

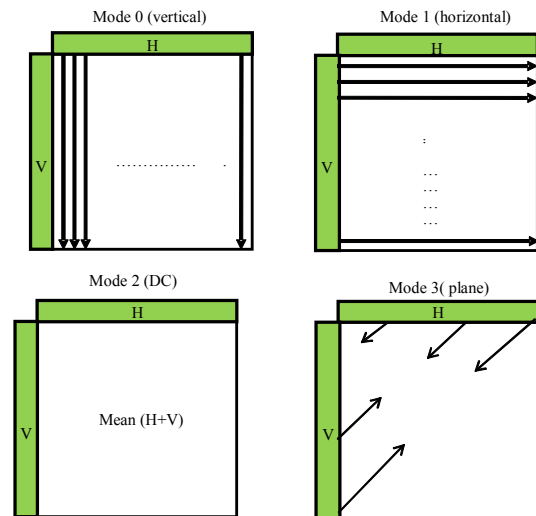


Figure 2. 16×16 intra prediction mode.

Table 1. Quantization step size in H.264/AVC.

QP	0	1	2	3	4	5
QStep	0.625	0.6875	0.8125	0.875	1	1.125
QP	6	7	8	9	10	11
QStep	1.25	1.375	1.625	1.75	2	2.25
QP
QStep
QP	48	49	50	51		
QStep	160	224		

Hence, the shift operation can be greatly used in the quantization and rescaling stages. To simplify the arithmetic, the quantization stated in (6) can be rewritten as (9, 10) for the AC coefficients [3,4].

$$|Z_{ij}| = (|I_{ij}| \cdot MF + f) \gg \text{qbits} \quad (6)$$

$$\text{sign}(Z_{ij}) = \text{sign}(I_{ij}) \quad (7)$$

Z_{ij} is the uncalled coefficients after QAC operation. The first 6 values of MF used in the H.264 references are listed in **Table 2**.

The 2nd and 3rd columns are the different positions in the scaling matrix. QP%6 represents the QP division rest by 6.

After the calculation of QAC, we must compute the inverse AC quantization. This operation is defined as [3,4].

$$Y_{ij} = Z_{ij} \cdot Qstep \quad (8)$$

A constant equal to 64 is integrated in order to avoid rounding errors. The inverse quantization AC equation becomes therefore:

$$Y_{ij} = Z_{ij} \cdot Qstep \cdot PF \cdot 64 \quad (9)$$

Y_{ij} is the result of inverse AC quantization. It must be divided by 64 for recovering the exact value without factor of scaling. The H.264 draft standard doesn't precise Qstep or PF directly. It uses a parameter given by:

$$V = (Qstep \cdot PF \cdot 64) \quad (10)$$

The final equation for the inverse quantization is:

$$Y_{ij} = Z_{ij} \cdot V_{ij} \cdot 2^{\text{floor}(QP/6)} \quad (11)$$

The first 6 values of V used in the H.264 standard are listed in **Table 3**. The 2nd and 3rd columns are the different positions in the scaling matrix.

For the DC coefficients, Hadamard transform is applied. The equation of 4×4 hadamard transform is defined by [3,4].

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \times$$

$$\begin{bmatrix} D_0 & D_1 & D_2 & D_3 \\ D_4 & D_5 & D_6 & D_7 \\ D_8 & D_9 & D_{10} & D_{11} \\ D_{12} & D_{13} & D_{14} & D_{15} \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (12)$$

“ D_i ” is the DC coefficients.

In next step, we calculate the quantization operation for the DC coefficients. This operation is defined by [3, 4].

$$K_{ij} = (H_{ij} \cdot MF(0,0) + 2f) \gg (\text{qbits} + 1) \quad (13)$$

K_{ij} is the uncalled coefficients after QDC operation. MF(0, 0) is the multiplication factor for position (0, 0) in **Table 2**. After the calculation of QDC, we must compute the 4×4 inverse hadamard transform. This operation is defined by [3,4].

$$H' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} D'_0 & D'_1 & D'_2 & D'_3 \\ D'_4 & D'_5 & D'_6 & D'_7 \\ D'_8 & D'_9 & D'_{10} & D'_{11} \\ D'_{12} & D'_{13} & D'_{14} & D'_{15} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (14)$$

Table 2. Multiplication factor MF in H.264/AVC.

QP%6	Positions (0,0),(2,0), (0,2),(2,2)	Positions (1,1),(1,3), (3,1),(3,3)	Others positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

Table 3. Values of V used in the H.264 standard.

QP%6	Positions (0,0),(2,0), (0,2),(2,2)	Positions (1,1),(1,3), (3,1),(3,3)	Others positions
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

“D’_i” is the block 4×4 quantified DC.

The final step for the DC coefficient is the inverse DC quantization. This operation is defined by [3,4].

$$\begin{aligned} &\text{for}(\text{QP} \geq 12) \\ &W_{ij} = H'_{ij} \cdot V(0,0) \cdot 2^{\text{floor}(\text{QP}/6) - 2} \\ &\text{for}(\text{QP} < 12) \\ &W_{ij} = [H'_{ij} \cdot V(0,0) + 2^{1 - \text{floor}(\text{QP}/6)}] \gg (2 - \text{floor}(\text{QP}/6)) \end{aligned} \quad (15)$$

where $V(0,0)$ is the multiplication factor for position (0,0) in **Table 3**.

After all operations, we can combine the AC and the DC coefficients for compute the inverse integer transform. Equation (19) gives the equation of 4×4 inverse integer defined as [3,4].

$$I' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{pmatrix} \begin{pmatrix} X'_0 & X'_1 & X'_2 & X'_3 \\ X'_4 & X'_5 & X'_6 & X'_7 \\ X'_8 & X'_9 & X'_{10} & X'_{11} \\ X'_{12} & X'_{13} & X'_{14} & X'_{15} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{pmatrix} \quad (16)$$

“X’_i” is the block 4×4 after all operations (AC and DC coefficients).

3. Intra 16×16 Architecture

The intra 16×16 architecture partitions the MB into sixteen 4×4 blocks. The scanning order for one MB is shown in **Figure 3**. This order is scanned in the x direction first and then performs the scanning in the y direction. The scanning order is the label order from top to bottom, from left to right which is the actual processing order for one MB. The MB is partitioned into sixteen 4×4 small sub-blocks. The partitions between the 16×16 scanning order labels and the 4×4 scanning order labels are shown in **Figure 4**.

The 4×4 scanning order labels are shown in **Figure 5**.

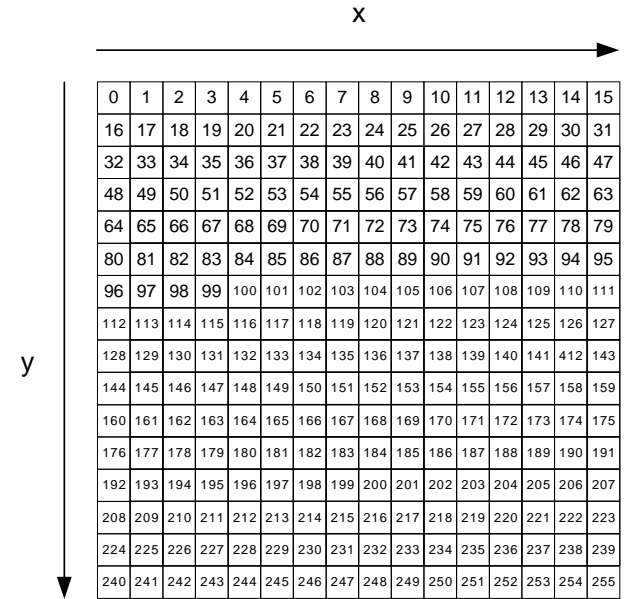


Figure 3. 16×16 scanning order labels.

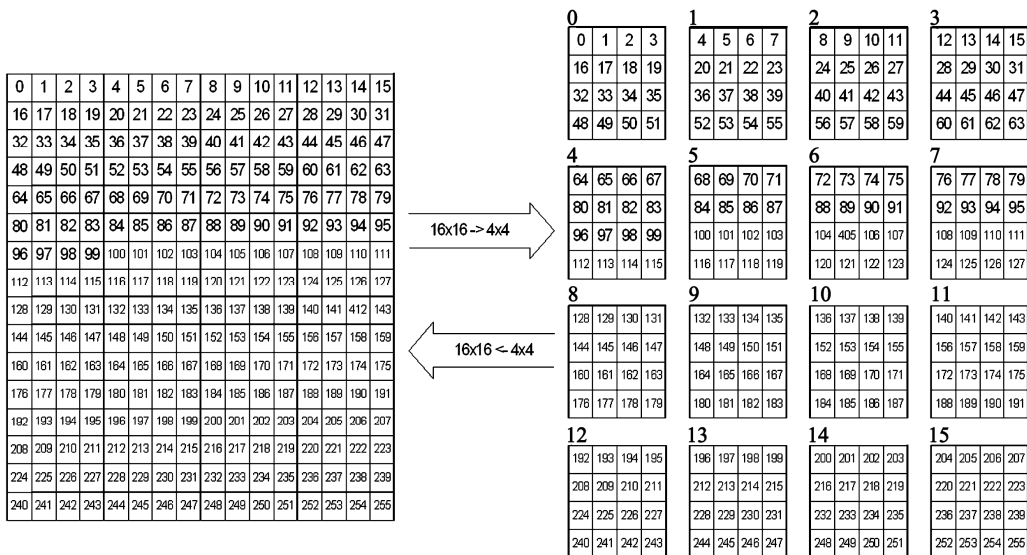


Figure 4. Relationship between 16×16 and 4×4 scanning order labels.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Figure 5. 4×4 scanning order labels.

Figure 6 shows the functional flow diagram of the intra 16×16 process.

In the first step, we compute the intra prediction 16×16 for all 4×4 blocks. After this, we calculate the residual, the integer transform, the AC quantization and the inverse AC quantization for each 4×4 block. During the

calculation of integer transform, we extract the DC coefficient for each 4×4 block. After obtain the 16 DC coefficients, we calculate the hadamard transform, the DC quantization, the inverse hadamard transform and the inverse DC quantization. Finally, we combine AC and DC coefficient for each 4×4 block to perform the inverse integer transform and the reconstruction pixels.

The intra 16×16 hardware architecture is composed by two modules. The first component contains the intra 16×16 prediction module and the residual module. The second component contains the coding chain module and the reconstruct module. The block diagram of the proposed hardware architecture for H.264 video coding is shown in Figure 7.

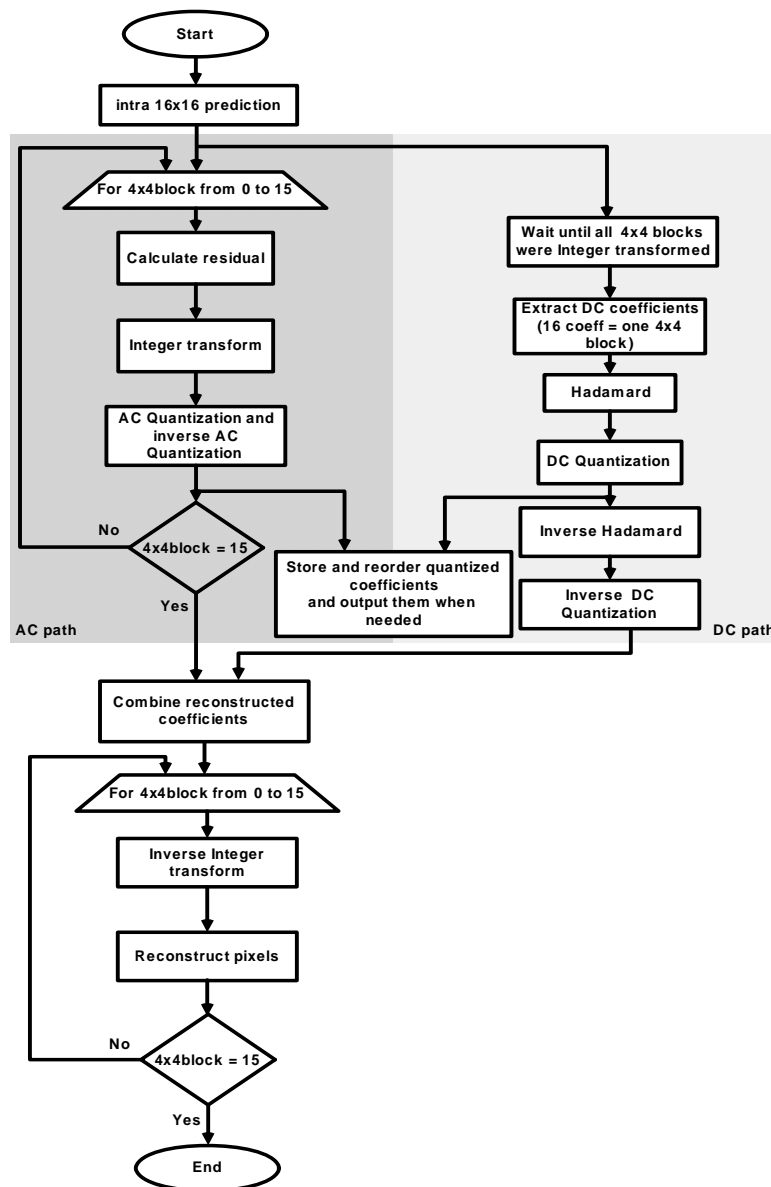


Figure 6. Intra 16×16 functional flow diagram.

3.1. Intra 16×16 Prediction

Different works have been proposed [9-13]. For our architecture, the MB pixels are loaded into a dual RAM (Random Access Memory) for reordering and then give (to the residual or reconstruction blocks) by sets of 16 pixels (4×4 block).

This block calculates the predicted pixels of MB for all 3 intra 16×16 prediction modes specified in the H.264 standard (horizontal, vertical and DC) in parallel based on the reconstituted pixels from the previous MB (planar mode is not used [14]). **Figure 8** presents the

intra prediction hardware architecture. These predicted pixels are stored into RAM for all modes. We also use a $SAD_{4 \times 4}$ block for calculating the SAD value for each mode. We accumulate this value 16 times in order to obtain the $SAD_{16 \times 16}$ for each mode. Those absolute values permit to give the sum of absolute differences (SAD) for each prediction mode. The comparator compares the SAD values for all prediction modes and picks the lowest value for determining which prediction mode will be used. After obtaining the best SAD (MIN_SAD), the best MB is given. The difference between the predicted pixels and the source pixels is then calculated for the best prediction mode for obtain the residual MB.

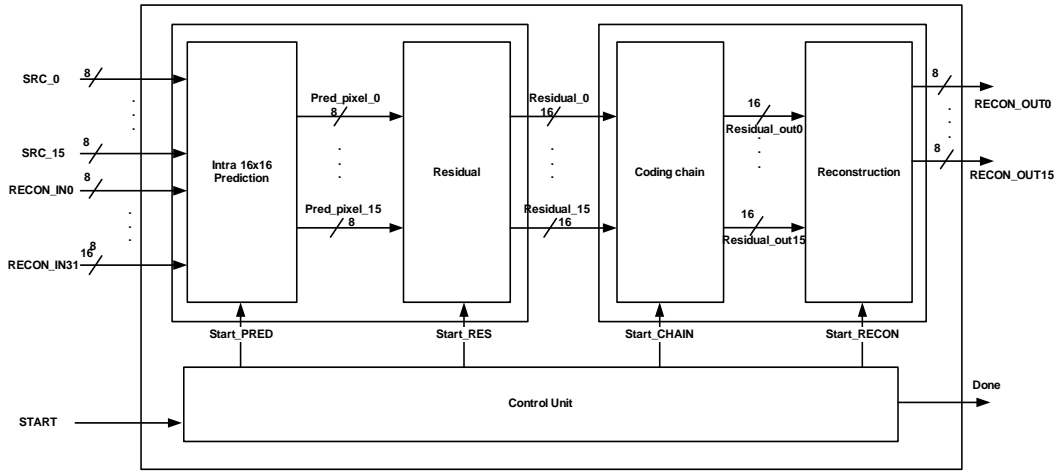


Figure 7. Intra 16×16 hardware architecture.

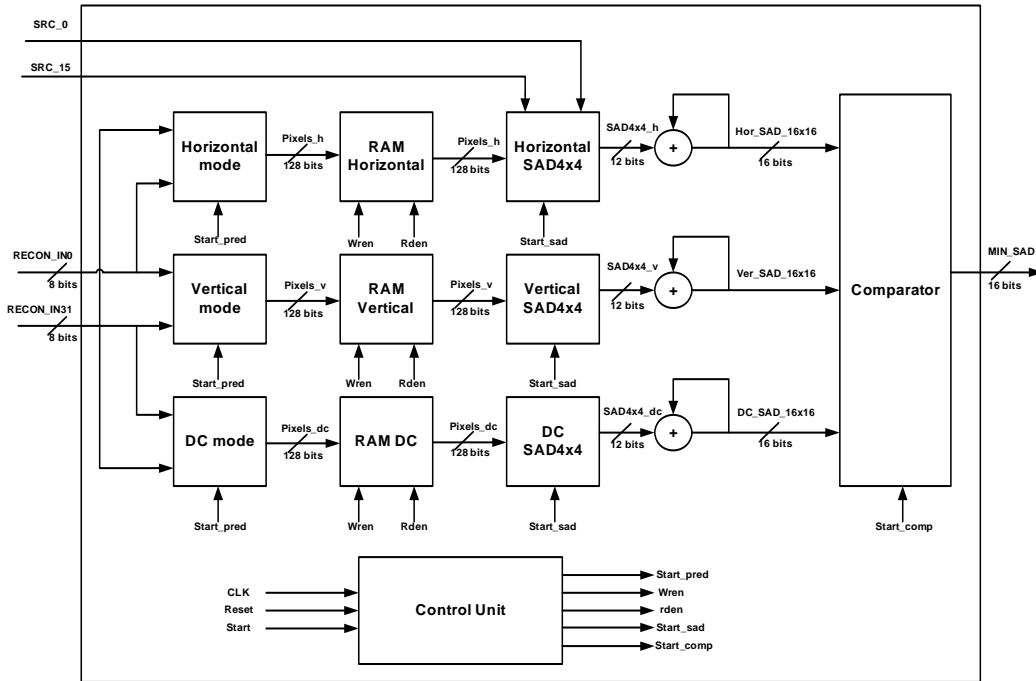


Figure 8. Intra 16×16 prediction hardware architecture.

3.2. ICT and HT Architectures

Different works have been published on the integer transform [15-19]. It is obvious that “I” shown in (1) or “H” shown in (12) can be implemented by a 1-D transform. **Figure 9** shows the fast implementation for the integer transform. The matrix contains only four coefficients: 1, -1, 2, and -2. It also can be implemented by using addition, subtraction and shift operations.

The Hadamard transform matrix is very similar to the integer transform matrices. The difference is that the coefficients of Hadamard transform are only 1 or -1. Therefore, the fast implementation for the Hadamard transform is shown in **Figure 10**.

The hardware implementation of 1-D ICT or HT is given in **Figure 11**. The input for this module is a 4×4 block. For full transform operation, we use two 1-D transforms in order to obtain the 2-D transform. **Figure 12** presents the architecture for the 2-D transform.

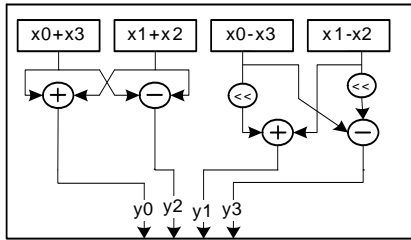


Figure 9. Fast implementations of H.264 integer transform.

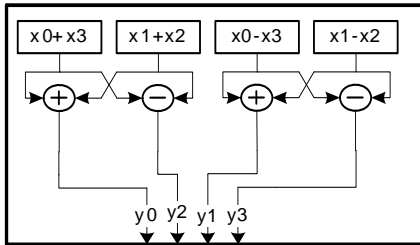


Figure 10. Fast implementations of H.264 Hadamard transform.

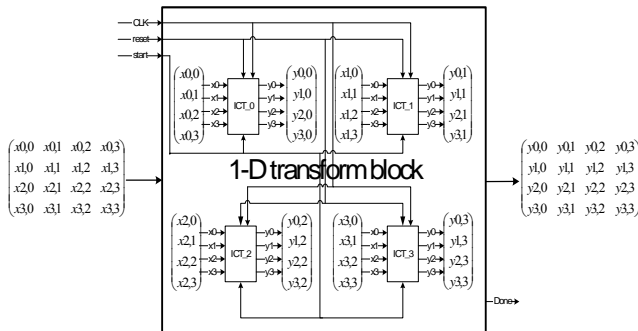


Figure 11. Fast implementations of H.264 1-D transform.

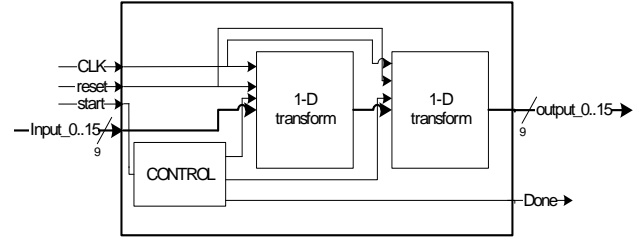


Figure 12. Fast implementations of H.264 2-D transform.

3.3. QAC & QDC Architectures

The Quantization hardware architectures have been proposed in [8,20]. The architecture of DC quantization is similar to the AC quantization presented in **Figure 13**. The multiplication factors stated in **Table 1** are stored into ROM (Read Only Memory) and selected according to the $QP \% 6$ values. The correct factor is multiplied by the uncalled coefficient in the corresponding position. The shifter will shift the product to right with qbits.

The QAC or QDC modules will quantify at the same time 16 pixels according to QP factor. These modules are composed by a quantization block (noted 0...15), a memory for storing the input pixels (noted input_0..15) and two read-only memories for storing QE (equal to $QP \% 6$) and F values noted respectively ROM_QE and ROM_F. The AC and DC quantization blocks are constituted by three basic components presented in **Figure 14**.

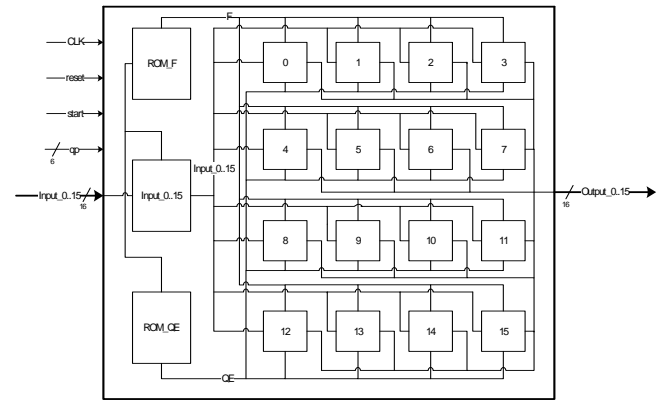


Figure 13. Quantization architecture.

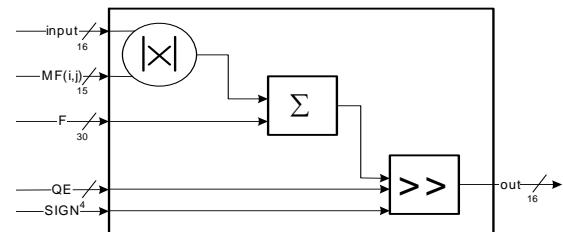


Figure 14. AC or DC quantification.

A multiplier deals perform the multiplication operation of AC coefficients with the corresponding MF (i, j) factor and gives the absolute value. An adder will perform the sum operation of values given by the multiplier with the F parameter given by the ROM memory. A shifter allows performing the shift operation the result from the adder by “qbits” (varies 15 to 23 according to the value of QP).

3.4. IQAC & IQDC Architectures

The IQAC or IQDC modules will quantify 16 pixels according to the QP factor. The architecture of these modules is similar to the QAC or QDC modules respectively presented by the **Figure 13**. The difference between quantization (AC or DC) and inverse quantization (AC or DC) is presented in the quantization block. For having the inverse AC quantization values, we use a multiplier to perform the multiplication operation between the QAC coefficients and the $V(i, j)$ values. We also use a shifter for shifting the result from the multiplier floor ($QP/6$). The architecture for this module is presented by the **Figure 15**.

For the DC coefficients, we use a multiplier to perform the multiplication operation between the QDC coefficients and the $V(0, 0)$ value. An adder will perform the sum of values given by the multiplier with $\{0, 1, 2\}$ (0 for $QP > 12$, 1 for $QP < 12$, 2 others parts). A shifter will perform the shift of result from the adder by floor $(QP/6) - 2$ for $QP \geq 12$ and by $(2 - \text{floor}(QP/6))$ for $QP < 12$. The architecture for this module is presented in **Figure 16**.

3.5. IICT and IHT Architectures

The IICT or IHT architectures are similar to the ICT or HT architectures respectively presented by the **Figures 12 and 13**. The inverse integer transform matrix contains only four coefficients: 1, -1 , $1/2$, and $-1/2$. **Figure 17** shows the fast implementation for the inverse integer transform. The inverse Hadamard transform matrix contains only two coefficients, 1 and -1 . **Figure 18** shows the fast implementation for the inverse Hadamard transform.

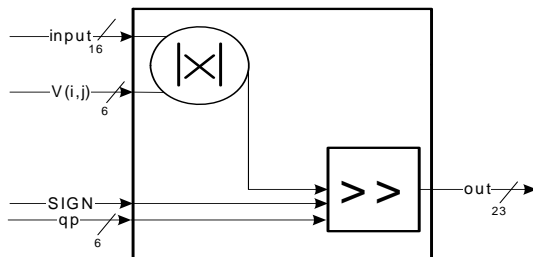


Figure 15. AC inverse quantification.

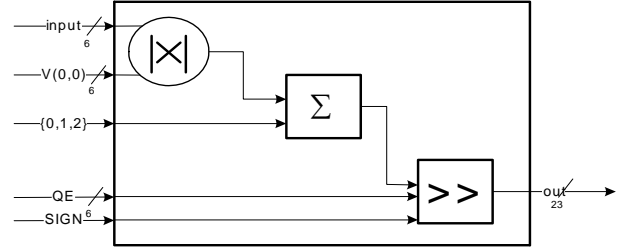


Figure 16. DC inverse quantification.

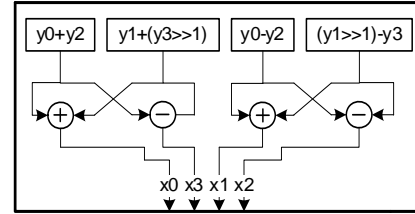


Figure 17. Fast implementations of H.264 inverse integer transform.

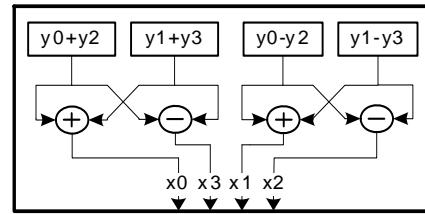


Figure 18. Fast implementation of H.264 inverse Hadamard transform.

3.6. Intra 16×16 Execution Time

The intra 16×16 execution time is presented in **Figure 19**. This figure is divided into two parts. The first part concerns the intra 16×16 prediction. This part takes 115 clock cycles for the best predicted MB [21]. The second part concerns the coding chain block that needs 77 clock cycles. In this part, we use a pipeline as shown in **Figure 19**. To get the reconstructed MB, we need 16 clock cycles. Finally, 208 clock cycles are necessary to achieve the intra 16×16 operations. Comparing with [7] and [8], the proposed architecture takes less clock cycles. Simulation of our proposed RTL design shows major improvements by reducing clock cycles for the intra 16×16 operation as shown in **Table 4**. Thus, our hardware implementation is optimized to achieve higher performances for the H.264 video encoder than the hardware architecture presented in [7-8].

4. Experimental Results

The whole design has been designed by using VHDL

(RTL level). The VHDL code of all modules was synthesized for an EP2S60F1020C3 Altera Stratix II FPGA circuit by using the Altera Quartus tool. **Table 5** shows the implementation results of the intra 16×16 module for the Stratix II EP2S60 FPGA circuit.

For experimental verification, we have developed a C language reference model of H.264 software. We have compared the output results of our C reference model with the JM 10.1 model [22] and we have confirmed the correctness of our model. We have also used the NIOS II softcore processor for sending data to the intra frame hardware coprocessor. The block diagram of the implemented H.264 intra frame encoder is shown in **Figure 20**. The design is composed by three parts: the NIOS II processor, the intra 16×16 frame module and the other peripherals connected to the Altera Avalon Bus. The Avalon bus has control, data and address signals and has its bus arbitration logic.

Our embedded system has been tested by using the Altera NIOS II development board. The heart of the target board is the Altera Stratix II EP2S60F1020C3 FPGA circuit. For all experiments, CIF test sequences are coded at 30 Hz. We have focussed on the following video test sequences: “Foreman”, “Paris”, “Mobile”, “Tb420” and “Akiyo”. These test sequences have different movement and camera particularities.

We have determined the processing time of intra 16×16 for the SW (software) solution. From the **Table 6**, we can conclude that a 35 time improvement for the processing speed compared to the software solution can be obtained by using our HW implementation.

Table 4. Comparison between different intra 16×16 architectures.

architectures	[7]	[8]	Proposed architecture
Number cycles/MB	3307	269	208
Frequency (Mhz)	71	54	160
Execution time/MB(ns)	46.57	4.98	1.3

Table 5. Implementation results for Stratix II FPGA.

Used Resources	
ALUTs	22,685/48,352 (47%)
Memory (KB)	27/2484 (1%)
Pins	526/719 (73%)
DSP block	124/288 (43%)

Table 6. Time comparison between SW and HW implementations.

Total time (ms)	Sequence	SW	HW
Time (ms)	Foreman	684.74	18.73
	Paris	688.21	18.88
	Mobile	689.40	18.72
	Tb420	685.78	19.08
	Akiyo	687.95	18.70

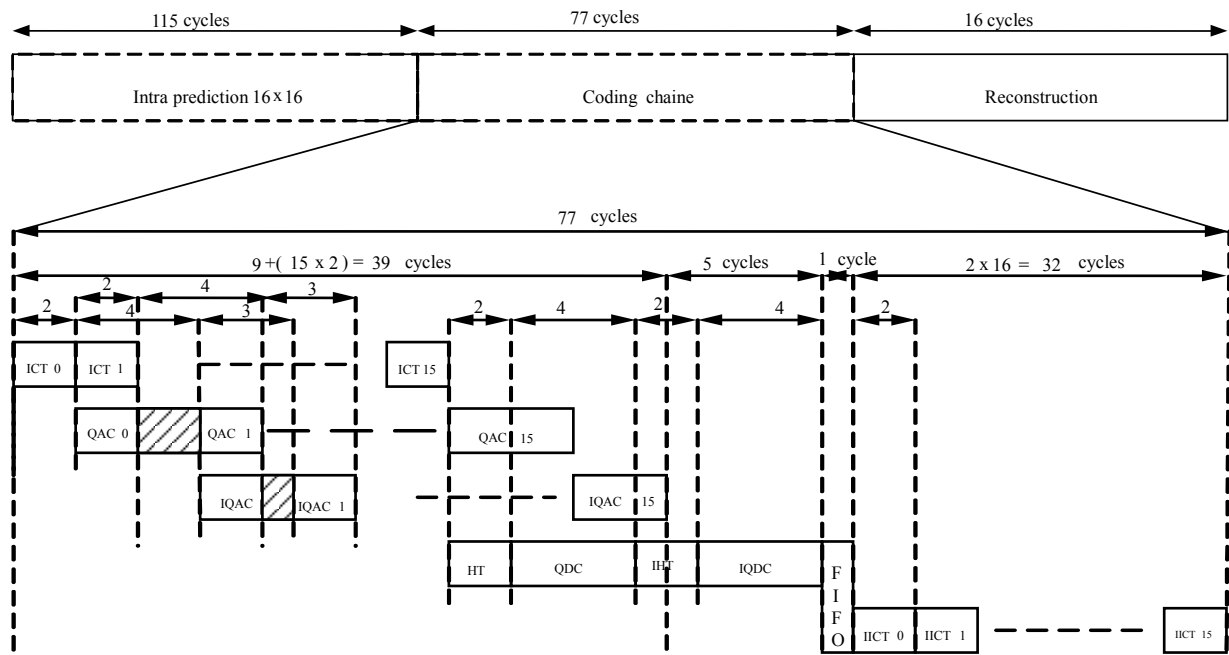


Figure 19. Intra 16×16 execution time.

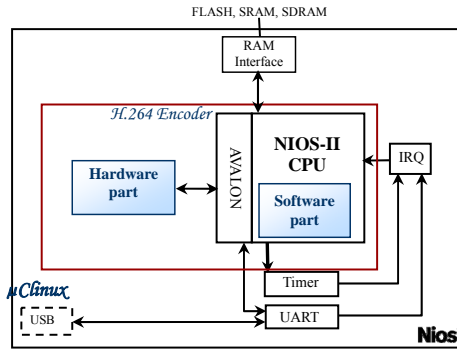


Figure 20. H.264 embedded system video encoder.

Table 7. PSNR comparison between SW and HW implementation.

PSNR	Sequence	SW	HW/SW
	Foreman	38.08	38.08
	Paris	37.15	37.15
	Mobile	36.37	36.37
	Tb420	37.04	37.04
	Akiyo	40.01	40.01

In order to evaluate the image quality given by this architecture, we have used the average peak signal-to-noise ratio (PSNR) which is here used as a measure of objective quality. The PSNR metric as shown as in **Table 7** has not detected any difference between the SW and HW solutions. Thus, the quality comparison confirms the correctness of the designed architecture.

The **Figure 21** presents the original and the two reconstructed (one from SW, the other from HW) of the 10th frame of the test video sequences.

5. Conclusions

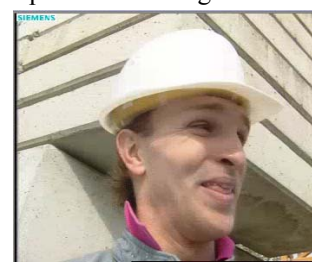
In this paper, we have described a new flexible and efficient HW architecture for H.264 video encoder. The hardware part has been implemented by using VHDL language. Comparing with [7] and [8], our proposed RTL implementation gives major improvements by reducing clock cycles for the intra 16×16 operation. The execution time is decreased by 26% even when compared with the best previous work for intra frame coding [8]. We have also designed an embedded system based on an Altera Stratix II FPGA platform running at 160 MHz in order to



Foreman sequence



PSNR – Y = 38.08 dB



PSNR – Y = 38.08 dB



Foreman Mobile



PSNR – Y = 36.37 dB



PSNR – Y = 36.37 dB



Paris sequence



PSNR – Y = 37.15 dB



PSNR – Y = 37.15 dB

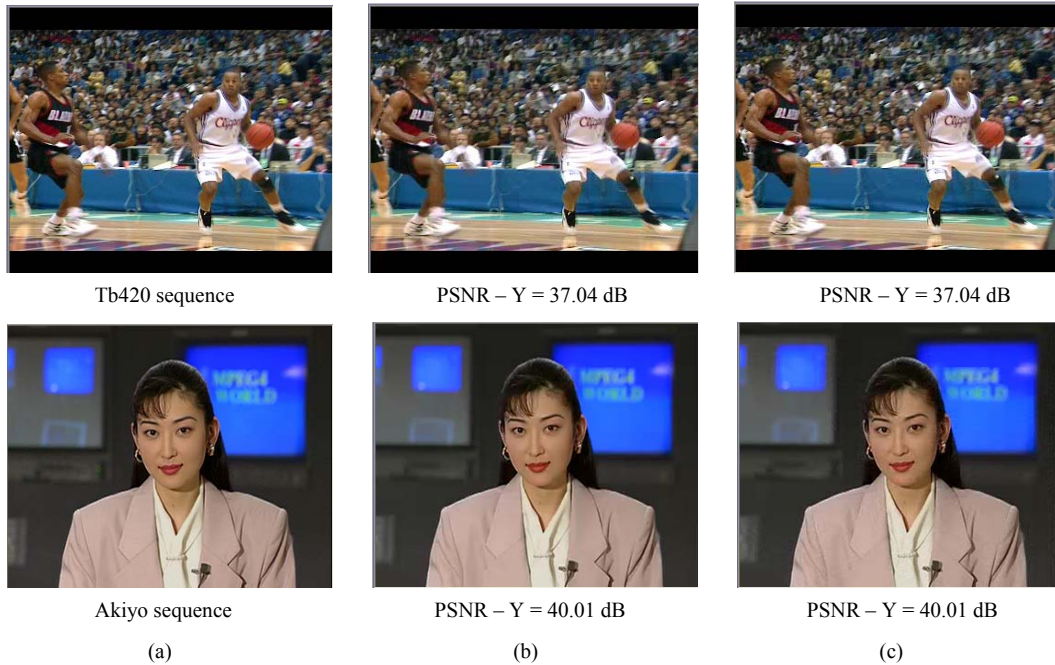


Figure 21. (a) Original, (b) Reconstructed from SW and (c) Reconstructed from HW/SW of the 10th frame of the test video sequences.

evaluate the performance of our design in HW/SW code-sign context. We have shown that our HW solution improves considerably the intra 16×16 process (35 times faster) compared to an all software solution with the same image quality.

6. References

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, 2003, pp. 560-576.
- [2] A. Luthra, G. J. Sullivan and T. Wiegand, "Introduction to the Special Issue on the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 7, 2003, pp. 557-559.
- [3] I. Richardson, "H.264 and MPEG-4 Video Compression," John Wiley and Sons Ltd., Chichester, 2003.
- [4] Joint Video Team (JVT) of ITU-T VCEG and ISO/IEC MPEG, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC)," May 2003.
- [5] G. J. Sullivan and T. Wiegand, "Video Compression—from Concepts to the H.264/AVC Standard," *Proceedings of the IEEE*, Vol. 93, No. 1, 2005, pp. 18-31.
- [6] Y.-W. Huang, B.-Y. Hsieh, T.-C. Chen and L. G. Chen, "Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder," *IEEE Transactions Circuit and Systems for Video Technology*, Vol. 15, No. 3, 2005, pp. 378-401.
- [7] İ. Hamzaoglu, Ö. Taşdizen and E. Şahin, "An Efficient H.264 Intra Frame Coder System Design," *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 4, 2008, pp. 1903-1911.
- [8] K. Suh, S. Park and H. Cho, "An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder," *ETRI Journal*, Vol. 27, No. 5, 2005, pp. 511-524.
- [9] B. Meng, O. C. Au, C.-W. Wong and H.-K. Lam, "Efficient Intra-Prediction Mode Selection for 4×4 Blocks in H.264," *Proceedings of International Conference on Multimedia and Expo*, Baltimore, 2003, pp. 521-524.
- [10] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu and S. Wu, "Fast Mode Decision Algorithm for Intra prediction in H.264/AVC Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 7, 2005, pp. 813-822.
- [11] B. Meng, O. C. Au, C. W. Wong and H. K. Lam, "Efficient Intra-Prediction Algorithm in H.264," *Proceedings of International Conference on Image Processing*, Barcelona, 2003, pp. 837-840.
- [12] S. S. Chun, J.-C. Yoon and S. Sull, "Efficient Intra Prediction Mode Decision for H.264 Video," *Lecture Notes in Computer Science*, Vol. 3767, 2005, pp. 168-178.
- [13] H. Loukil, A. Ben Atitallah and N. Masmoudi, "An Efficient FPGA Parallel Architecture for H.264/AVC Intra Prediction Algorithm," *Proceeding of International Conference on Embedded Systems and Critical Applications*, Gammarth, Tunisia, 2008, pp. 191-196.
- [14] A. Kessentini, B. Kaanich, I. Werda, A. Samet and N. Masmoudi, "Low Complexity Intra 16×16 Prediction for

- H.264/AVC,” *Proceedings of International Conference on Embedded Systems & Critical Applications*, Tunis, Tunisia, 2008, pp. 197-201.
- [15] T.-C. Wang, Y.-W. Huang, H.-C. Fang and L.-G. Chen, “Parallel 4×4 2D Transform and Inverse Transform Architecture for MPEG-4 AVC/H.264,” *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems*, Bangkok, 2003, pp. 800-803.
- [16] L. Liu, Q. Lin, M. Rong and J. Li, “A 2-D Forward/Inverse Integer Transform Processor of H.264 Based on Highly-Parallel Architecture,” *Proceedings of the 4th IEEE International Workshop on System-on-Chip for Real-Time Applications*, Banff, July 19-21, 2004, pp. 158-161.
- [17] K.-H. Chen, J.-I. Guo and J.-S. Wang, “An Efficient Direct 2-D Transform Coding IP Design for MPEG-4 AVC/H.264,” *IEEE International Symposium on Circuits and Systems*, Kobe, May 23-26, 2005, pp. 4517-4520.
- [18] G. Raja, S. Khan and M. J. Mirza, “VLSI Architecture & Implementation of H.264 Integer Transform,” *The 17th International Conference on Microelectronics*, Islamabad, December 13-15, 2005, pp. 218-223.
- [19] C.-P. Fan, “Fast 2-Dimensional 4×4 Forward Integer Transform Implementation for H.264/AVC,” *IEEE Transactions on Circuits and Systems—II: Express Briefs*, Vol. 53, No. 3, 2006, pp. 174-177.
- [20] R. Kordasiewicz and S. Shirani, “Hardware Implementation of the Optimized Transform and Quantization Blocks of H.264,” *IEEE Canadian Conference on Electrical and Computer Engineering*, Canada, May 2-5, 2004, pp. 943-946.
- [21] H. Loukil, S. Arous, I. Werda, A. Ben Atitallah, P. Kadi-onik and N. Masmoudi, “Hardware Architecture for H.264/AVC INTRA 16×16 Frame Processing,” *IEEE International Multi-Conference on Systems, Signals and Devices*, Djerba, March 23-26, 2009, pp. 1-5.
- [22] “JVT H.264 Reference Software Version JM10.1,” http://iphome.hhi.de/suehring/tml/download/old_jm/

The Design of an Intelligent Security Access Control System Based on Fingerprint Sensor FPC1011C

Yan Wang^{1,2}, Hongli Liu¹, Jun Feng²

¹College of Electrical and Information Engineering, Hunan University, Changsha, China

²College of Electrical Engineering, Nanhua University, Hengyang, China

E-mail: wangyan5406@163.com

Received April 29, 2010; revised May 30, 2010; accepted June 5, 2010

Abstract

This paper deals with the design of an intelligent access control system based on the fingerprint sensor FPC1011C. The design uses the S3C2410 and TMS320VC5510A as the system processor. A fingerprint acquisition module and a wireless alarm module were designed by using the fingerprint sensor FPC1011C and GPRS module SIM100 respectively. The whole system was implemented wireless alarm through messages and GPRS-Internet in the GSM/GPRS web. In order to achieve the simple and high Real-time system, the μ C-Linux system migration was also implemented.

Keywords: Fingerprint Sensor, Security Access Control System, ARM, Wireless Alarm

1. Introduction

Generally, traditional fingerprint access control system is based on the computer or microcontroller. The Computer platform is hard to carry because of high cost and the MCU platform is more difficult or even impossible to achieve for its features such as data processing capabilities and storage capacity constraints. The circuit of wireless alarm is vulnerable and security is not high with high maintenance cost; special networks of wireless alarm are too costly. All those factors prevent the development of fingerprint access control system.

This design uses ARM + DSP microprocessor and advanced fingerprint sensor FPC1011C, and is also equipped with GPRS\GSM network for wireless alarm. In the Fingerprint access control system, it is lack of control using DSP microprocessor [1] and weak processing capability using ARM microprocessor [2]. And thus it may be an attractive approach to a more perfect framework for fingerprint access controlling by combining ARM and DSP in an integrative platform. Fingerprint sensor FPC1011C is suitable for both dry and wet fingers due to its adopted reflective detection technology. As a public network with wide coverage, GPRS/GSM supports real-time transmission, lower operation costs, SMS service and internet access [3]. So it can be widely used in wireless alarm of access control system.

2. System Principles

System principles are as shown in **Figure 1**. The system acquires fingerprint data from the fingerprint device. Then it will be sent to the DSP processor, where fingerprint data are processed and effective features are extracted. In ARM processors, the feature will be registered to the database or acts as data for triggering target event: if the fingerprint's feature and some sample in the database matches, access is permitted, or there will be alarm [4]. Thereinto, alarms are divided into local alarms and wireless ones. Local alarm is given by sound and light lamps. Wireless alarm send messages to the concerned mobile phone by GPRS module through which connect GPRS/GSM network controlled by ARM, and it is also sent to the PC alarm monitoring center by GPRS-internet network.

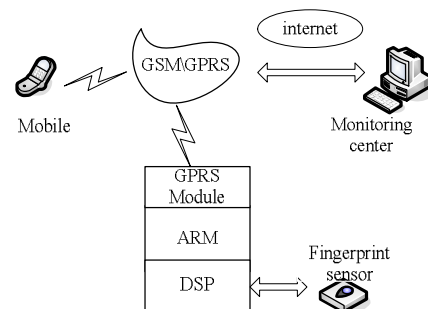


Figure 1. System principle.

3. Hardware Design

The system hardware includes: DSP, fingerprint sensor, ARM controller, GPRS module and its external power supply module, extended memory devices, LCD and keyboard.

3.1. DSP, FPC1011C and ARM

This design uses a DSP chip TMS320VC5510A as the processor and the most advanced fingerprint sensor capacitance FPC1011C. Control chip used ARM Samsung S3C2410, which is a 16/32-bit embedded RISC micro-processor [5] based on ARM920T core. The interface of ARM + DSP and FPC1011C is shown in **Figure 2**.

FPC1011C is a capacitive fingerprint sensor surface shape which is launched successfully by Finger Prints, a Swedish company [6]. It has high image quality, wear resistance, static and low power consumption. It uses the reflective detection technology which is a patent by Finger Prints. The electrical pulse signal is generated from the internal IC, and then passed to finger through the conductive ABS frame. Because the human body is a conductor, fingerprints will produce high and low voltage form Valley and Ridge when the pulse through the finger. The wafer body accepts and amplifies the signal, then exports digital signal of fingerprint after A/D conversion. Because of its unique detection technology, it has good applicability of the dry and wet fingers and longer life.

FPC1011C provides a high-speed SPI interface to communicate with the DSP processor TMS320VC5510A, so its hardware interface circuit design is simple with fast transfer rate. DSP's Mcbsp can simulate the SPI protocol, complete with a seamless interface to the fingerprint sensor. The useful pins of FPC1011C are CS_N, SO, SI, SCK and so on. CS_N is the chip select signal line, which is directly to ground, so that the fingerprint sensor has been in the selected state. SO and SI is the SPI data lines which connected with DSP-DR2, DX2 to transmit the serial data. SCK is the clock signal line and linked by DSP's CLKX2 to provide the sampling clock [7].

TMS320VC5510A host interface (HPI) is an interface designed by TI Company can greatly simplify the hardware design of DSP exchanging data with external device. Communication can be achieved by linking ARM's I/O port to the DSP's HPI. System maps the all HPI interface controllers, address registers, data registers which have unified address to the S3C2410's I/O memory space. As the basic HPI, HCS enables input signal, and controls HPI data transfer with HDS1, HDS2. Address line A1~A4 produces the required control signals by HPI.

3.2. GPRS Module

The design of GPRS module uses SIM100, and it has high performance with enhanced AT command set. The technical specifications are suitable for the development of GPRS-based wireless products, and it provides users with a fully functional system interface, saving users time and cost of development. Module supports an external SIM card, and you can connect directly with the SIM card. The module can automatically detect and adapt to SIM card type, which have GPRS service. Module supports GSM, SMS business and the GPRS access internet [8]. GPRS and SIM interface circuit diagram are shown in **Figure 3**.

4. System Software

4.1. μ CLinux Transplantation

Linux operating system as a derivative μ CLinux, followed the vast majority of linux features, overcame the shortage of cell-free MMU, and had powerful network processing capabilities, it also can support various file systems and has become mainstream in embedded operating system. This design transplants the ARM- μ CLinux, and the specific process includes the development of environment, preparation, revision of Bootloader, modification and compilation of the kernel source, transplantation of kernel image and the file system [9].

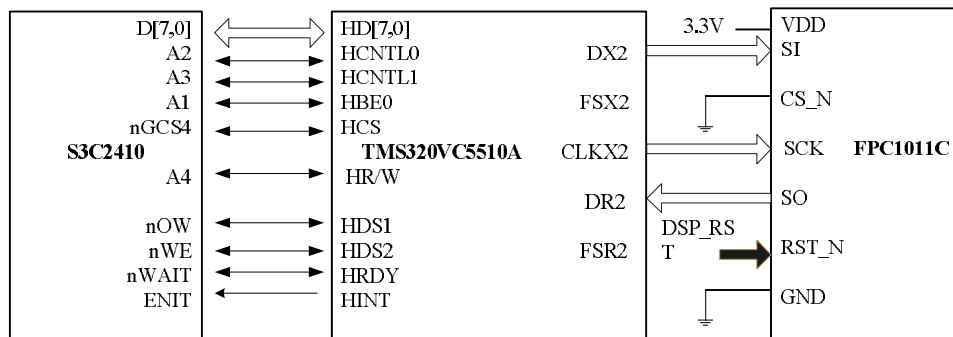


Figure 2. Interface of ARM + DSP and FPC1011C.

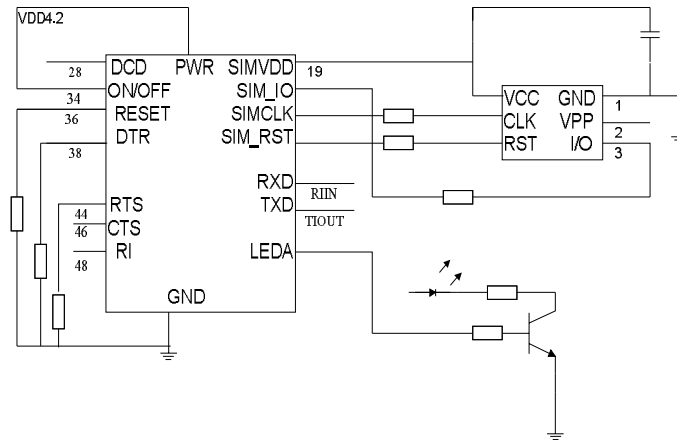


Figure 3. Interface of SIM100 with the SIM Card.

4.2. GPRS Driver Design

GPRS module driver includes initializing, opening, closing and sending or receiving data [10]. By calling GPRS_Init() function to achieve the basic GPRS module initialization, includes initializing serial port, testing signal quality, opening AT protocol and GPRS, setting IP address, port number and so on. It is concentrated by calling the AT command, AT + WIPDATA and AT + WIPCLOSE order to achieve the connection beginning and closing. S3C2410 communicates with GPRS module by AT command. GPRS_SendData() is used to send AT commands to the GPRS module, and GPRS_RecvData() accepts GPRS data module. GPRS_SendMsg() sends a short message, and GPRS_RecvMsg() parses text messages.

System is mainly divided into two parts: the fingerprint processing and ARM processor control.

1) The main program flow chart of fingerprint processing is shown in Figure 4. Firstly, initialization operation, includes the DSP system initialization and peripheral initialization; and then it starts to test whether it has fingerprints input on the fingerprint sensor. If there are fingerprints input on the fingerprint sensor, it identifies this signal. The fingerprint image starts preprocessing work after recognition. This fingerprint image quality evaluation is in front of the pretreatment on the identification of incoming, if the quality of fingerprint is good enough then continue the rest of the pretreatment. If image quality is poor, this fingerprint data is discarded and required to identify a fingerprint data again. After the extraction of high-quality fingerprint image feature, the judge is the registered fingerprint or fingerprint match. If it is to register the fingerprint, the fingerprint data is put into the database. If it is to match the fingerprint, system will compare the fingerprint with each sample in the library, which leads to a result means weather or not match successfully.

2) External LCD of ARM provides user interface and prompts information, the keyboard can input data [11]. Access Administrators use them to set the system to perform specific technical indicators [12], for example, opening access, allowing access to the total number, etc, accessing to personnel information prompted by operating the input fingerprint. The chart of ARM processor to control the main program flow is shown in Figure 5.

Firstly, initializing the module, and then determining whether there is an order input. If there is administrator command, prompting for a password, and the fingerprint verification follow-up after the administrator can operate. If there is ordinary user command, the user will be reminded to input the password and fingerprint, the access control will be lifted after authentication. Otherwise, the control alarm module will conduct wireless alarm.

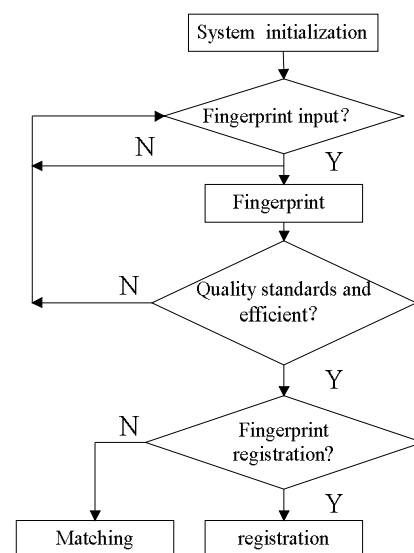


Figure 4. Flow of fingerprint procedures.

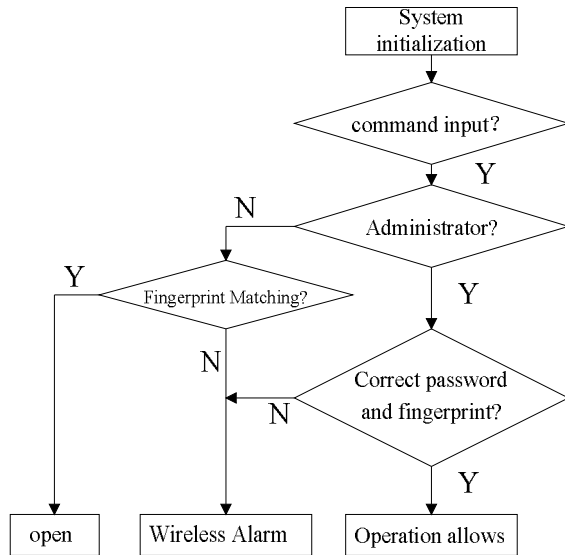


Figure 5. Flow of ARM control.

5. Conclusions

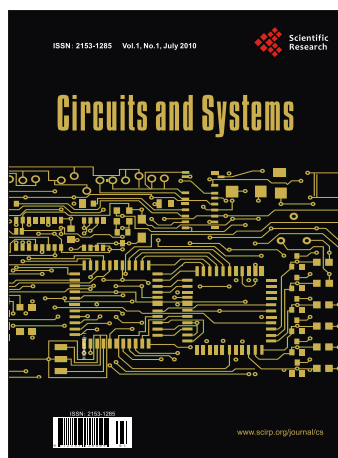
In this paper, we researched the embedded wireless alarm system and fingerprint access control, which use the ARM + DSP dual processors to improve system control and data processing capabilities, and it uses GPRS/GSM network to improve the speed and security of alarm and reduce the cost. After transplanting μ CLinux system, system function can be improved and extended cut, which make the system has good openness. This design is efficient, compact, and with low power cost and high safety level, which significantly meet the requirements of the current fingerprint access control system and increased the levels of Access Control System.

6. References

- [1] K. L. Zhou and Z. X. Lu, "Design of Vehicle Locks Based on DSP and Fingerprint Identification System,"

Journal of Shanxi University of Science & Technology, Vol. 27, No. 5, 2009, pp. 103-105.

- [2] C. G. Xie, "ARM-Based Automatic Fingerprint Identification System," *Microcomputer Information*, Vol. 25, No. 1-4, 2009, pp. 292-294.
- [3] H. Guo, Y. S. Guo and Y. Chen, "The Implementation of Remote Meter Reading System Based on Linux and GPRS," *Application of Electronic Technique*, Vol. 34, No. 11, 2008, pp. 82-84.
- [4] F. Ding, "ARM-Based Fingerprint Identification System Research and Implementation of [D]," South China University of Technology, Guangzhou, 2007.
- [5] Z. M. Ma and Y. H. Xu, "ARM Based Embedded Processor Architecture and Application," Beijing University of Aeronautics and Astronautics Press, Beijing, 2002.
- [6] J.-M. Nam, S.-M. Jung, D.-H. Yang and M.-K. Lee, "Design and Implementation of 160×192 Pixel Array Capacitive-Type Fingerprint Sensor," *Circuits, Systems & Signal Processing*, Vol. 24, No. 4, 2005, pp. 401-413.
- [7] L. Zhang, "Based on DSP and RF Card Embedded Fingerprint Identification System Design and Implementation of [D]," University of Electronic Science and Technology of China, Chengdu, 2009.
- [8] K. Al-Begain, I. Awan and D. D. Kouvatso, "Analysis of GSM/GPRS Cell with Multiple Data Service Classes," *Wireless Personal Communications*, Vol. 25, No. 1, 2003, pp. 41-57.
- [9] Y. Li, "MCLinux Based on ARM Embedded System Theory and Application," Tsinghua University Press, Beijing, 2009.
- [10] J. K. Zhang and X. Q. Zhang, "Embedded Linux System Development Technology Xiangjie—Based on ARM," Posts & Telecom Press, Beijing, 2006.
- [11] M. F. Du, "Development of Ethernet Interface for Smart Entry Controller Based on ARM & Linux Architecture," *Computer Engineering*, Vol. 33, No. 16, 2007, pp. 234-236.
- [12] C. J. Zhang, "System of Defending to Rob by Tailing Behind of Double Gate by Chip Microcomputer Control," *Computer Engineering and Applications*, Vol. 43, No. 5, 2007, pp. 79-81.



Call for Papers

Circuits and Systems

ISSN Print: 2153-1285 ISSN Online: 2153-1293

<http://www.scirp.org/journal/cs>

Circuits and Systems is an international journal dedicated to the latest advancement of theories, methods and applications in Electronics, Circuits and Systems. The goal of this journal is to provide a platform for scientists and academicians all over the world to promote, share, and discuss various new issues and developments in different areas of electronics, circuits and systems.

Subject Coverage

All manuscripts must be prepared in English, and are subject to a rigorous and fair peer-review process. Accepted papers will immediately appear online followed by printed hard copy. The journal publishes original papers including but not limited to the following fields:

- Analog Signal Processing
- Biomedical Circuits and Systems
- Blind Signal Processing
- Cellular Neural Networks and Array Computing
- Circuits and Systems for Communications
- Computer-Aided Network Design
- Digital Signal Processing
- Graph Theory and Computing
- Life-Science Systems and Applications
- Live Demonstrations of Circuits and Systems
- Multimedia Systems and Applications
- Nanoelectronics and Gigascale Systems
- Neural Systems and Applications
- Nonlinear Circuits and Systems
- Power Systems and Power Electronic Circuits
- Sensory Systems
- Visual Signal Processing and Communications
- VLSI Systems and Applications

We are also interested in: 1) Short Reports 2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data; 2) Book Reviews Comments and critiques.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-Mail

<http://www.scirp.org/journal/cs>

E-mail: cs@scirp.org

TABLE OF CONTENTS

Volume 1 Number 1

July 2010

Two Simple Analog Multiplier Based Linear VCOs Using a Single Current

Feedback Op-Amp

D. R. Bhaskar, R. Senani, A. K. Singh, S. S. Gupta..... 1

Voltage Mode Cascadable All-Pass Sections Using Single Active Element and

Grounded Passive Components

J. Mohan, S. Maheshwari, D. S. Chauhan..... 5

Fast Implementation of VC-1 with Modified Motion Estimation and Adaptive

Block Transform

M. Tammen, M. El-Sharkawy, H. Sliman, M. Rizkalla..... 12

FPGA Design of an Intra 16×16 Module for H.264/AVC Video Encoder

H. Loukil, I. Werda, N. Masmoudi, A. B. Atitallah, P. Kadionik..... 18

The Design of an Intelligent Security Access Control System Based on Fingerprint

Sensor FPC1011C

Y. Wang, H. L. Liu, J. Feng..... 30