0.35um implementation of an experimental mixed signal image compression circuit

Reema Divatia

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
0.35um Implementation of an Experimental Mixed Signal Image Compression Circuit

by

Reema Manojkumar Divatia

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

Supervised by

Dr. Marcin Lukowiak
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
September 2006

Approved By:

Dr. Marcin Lukowiak
Assistant Professor
Department of Computer Engineering, RIT
Primary Adviser

Dr. Pratapa Reddy
Professor
Department of Computer Engineering, RIT

Dr. Muhammad Shaaban
Associate Professor
Department of Computer Engineering, RIT
Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title: 0.35um Implementation of an Experimental Mixed Signal Image Compression Circuit

I, Reema Manojkumar Divatia, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Reema Manojkumar Divatia

9/8/06
Dedication

To my parents Sushama and Manoj Divatia, for their constant love, trust and motivation, without which this would not have been possible.
Acknowledgments

I would like to take this opportunity to thank several people who helped me during this thesis. Dr. Marcin Lukowiak, for his support, guidance and patience that helped me complete this thesis. Dr. Pratapa Reddy and Dr. Muhammad Shaaban for the advice and suggestions they gave me as members of my committee. Mr. Paul Mezzanini and Mr. Richard Tolleson for giving me constant help with lab issues. The Staff and Faculty of the Computer Engineering Department at RIT, for their everyday help. Sreeram for all his help and support.

I would also like to extend my gratitude to Mr. Manny and Mrs. Nancy Marcano, along with Mr. Walt Pyska, for their patience, understanding and invaluable support. Finally I would like to thank all my family and friends who have patiently supported me all these years.
Abstract

Switched-current is an analog, discrete in time, signal processing technique that is fully compatible with any digital CMOS technology. This means that analog circuits can be realized together with digital components on a single chip without any additional technological processes. In designs implemented using the switched-current technique, the individual circuit elements interact by the means of currents, which allows to reduce voltage swings and thus power consumption.

This work investigated the implementation of a low power mixed signal image compression system in TSMC 0.35um technology. The major components of this system were two dimensional discrete cosine transform processor, analog to digital converter, quantizer and entropy encoder. The discrete cosine transform section was implemented using switched-current technique. The digital part consisting of the quantizer, entropy encoder and control unit was modelled using VHDL and then synthesized into standard cells.
Contents

Dedication .......................................................... iii

Acknowledgments ..................................................... iv

Abstract ............................................................ v

1 Introduction ....................................................... 1

2 Switched-Current Circuits ........................................ 5
   2.1 Modified Switched-Current Memory Cell ..................... 6
      2.1.1 Design ......................................................... 6
      2.1.2 Simulation .................................................... 8

3 Digital to Analog Converter ..................................... 13
   3.1 Design ............................................................ 13
   3.2 Simulation ....................................................... 15

4 Two Dimensional Discrete Cosine Transform .................. 17
   4.1 One Dimensional Discrete Cosine Transform ................. 19
      4.1.1 Design ........................................................ 19
      4.1.2 Simulation .................................................... 21
   4.2 Switched-Current Memory Block ............................... 22
      4.2.1 Delay Cell .................................................... 22
      4.2.2 Design of Switched-Current Memory Block ................ 25
      4.2.3 Simulation of Switched-Current Memory Block ............ 27
   4.3 Differential to Single Ended Current Converter ............. 30
   4.4 Design and Simulation of 2D-DCT .............................. 32

5 Analog to Digital Converter .................................... 33
   5.1 Comparator ...................................................... 33
List of Figures

1.1 Block diagram of the experimental mixed signal image compression circuit 2

2.1 Implementation of adder and multiplier using current source .................. 5
2.2 Delay element ......................................................................................... 6
2.3 Modified switched-current memory cell .................................................. 7
2.4 Simulation results for switched-current memory cell ............................... 10
2.5 Simulation setup for switched-current memory cell ............................... 11
2.6 Switch used in switched-current memory cell .......................................... 12
2.7 Inverter used for the switches ................................................................. 12

3.1 8-bit DAC ............................................................................................... 14
3.2 Simulation result for DAC ........................................................................ 16

4.1 Block diagram of 2D-DCT ....................................................................... 19
4.2 1D-DCT .................................................................................................... 20
4.3 Delay cell .................................................................................................. 23
4.4 Simulation setup for delay element ............................................................ 24
4.5 Simulation results for delay element .......................................................... 24
4.6 Block diagram of SIM .............................................................................. 25
4.7 Simulation setup for a 2x2 Memory block ................................................ 26
4.8 2x2 Memory block implemented using delay cells .................................. 27
4.9 Output from 2x2 memory block ............................................................... 28
4.10 Output from memory block with first set of inputs ............................... 29
4.11 Differential to single ended current converter ........................................ 31

5.1 Comparator used for analog to digital converter ...................................... 34
5.2 Difference current generator used for the comparator ............................ 34
5.3 5-bit current mode ADC .......................................................................... 35
5.4 Simulation results for 5-bit ADC ............................................................. 37

6.1 Block diagram for digital section ............................................................. 38
6.2 Flow chart for digital section .................................. 40
6.3 Simulation results for register with 5-bit input and 8-bit output ........................................ 40
6.4 Simulation results for a register with 8-bit input and 8-bit output ................................ 41
6.5 Flow chart for zig-zag ordering ........................................... 42
6.6 Output file from zig-zag ordering simulation ............................................. 43
6.7 Flow chart for quantization .................................................. 46
6.8 Input text file to test quantization ............................................. 47
6.9 Simulation results after quantization ............................................ 50
6.10 Flow chart for Huffman encoding ........................................... 51
6.11 Simulation results from Huffman encoding ....................................... 52
6.12 Simulation results for control block ............................................ 54
6.13 Simulation results for digital section ............................................ 55

7.1 Experimental mixed signal image compression circuit ........................................ 58
7.2 Design flow ................................................................. 60
7.3 Layout for the experimental image compression circuit ........................................ 62
List of Tables

2.1 Transistor sizes for switched-current memory cell .................. 9
3.1 Transistor sizes for DAC ................................................. 15
3.2 Simulation results for DAC for an input reference current of 200uA . . . . 16
4.1 Transistor sizes for 1D-DCT ............................................. 20
4.2 Simulation results for 1D-DCT with 100uA constant current input .... 21
4.3 Simulation results for 1D-DCT ........................................... 21
4.4 Transistor sizes for delay cell ......................................... 22
4.5 Simulation results for 2x2 memory block .............................. 27
4.6 Simulation results for memory block with first set of inputs .......... 28
4.7 Simulation results for memory block with second set of inputs ...... 29
4.8 Results of differential to single ended current converter ............ 30
4.9 Simulation results of 2D-DCT ........................................... 32
5.1 Simulation results for ADC ............................................. 36
6.1 Difference magnitude categories and typical Huffman table [14] ....... 48
6.2 AC coefficient coding [12] ............................................... 49
6.3 Magnitude categories for encoding AC ................................ 50
6.4 Magnitude categories for encoding AC ................................ 56
7.1 Simulation results of 2D-DCT and ADC ............................... 61
8.1 Comparison of power consumption of different 2D-DCT structures ... 63
Chapter 1

Introduction

In today’s world, digital circuits are commonly used among various industries for signal processing. These digital circuits however have to interface with the analog world in order to acquire and export the data, thereby creating need for analog circuits like sensors, amplifiers, pre-processing filters, analog to digital converters and digital to analog converters.

Another very important reason why analog circuits are used today is due to their lower power consumption in comparison to digital circuits.

Switched-current is a current mode, discrete in time, analog signal processing technique. For implementing mixed signal integrated circuits, it might be advantageous to use switched-current technique instead of switched-capacitor method because unlike the switched-capacitor method which requires linear capacitors and hence extra fabrication processes in order to design analog circuits, the switched-current method uses only transistors [2].

The objective of this thesis was to investigate the implementation of an experimental mixed signal image compression circuit. The block diagram of the entire system that was implemented is shown in the figure 1.1.

The major components for the analog section of this mixed signal system were digital to analog converter (DAC), two dimensional discrete cosine transform (2D-DCT) and analog to digital converter (ADC).

As shown in figure 1.1, an 8-bit digital input was given to a DAC. The output of this converter was the analog current input to 2D-DCT. The DAC block was used at the input
Figure 1.1: Block diagram of the experimental mixed signal image compression circuit
for the purpose of simplification during testing of the design at a later stage.

The 2D-DCT is the basis for image compression standards. The image in spatial domain is divided into blocks and then is transformed to the frequency domain using 2D-DCT. In this work, the 2D-DCT was realized using switched current technique.

The output from the 2D-DCT, is converted to digital signal with the help of ADC. The ADC used in this work was a current-mode algorithmic ADC implemented using current mirrors, switches and current references [16].

The major components of the digital section are quantization block, entropy encoder and the control block. The JPEG compression standard was followed in this section cite-

pennebaker.

The digital data is quantized and then encoded by the entropy coding section to give additional compression by making use of the redundancy of data. The inputs are encoded with the minimum number of bits required to represent them. Some of the popular entropy coding schemes are Huffman Coding and Arithmetic Coding. A control block was designed to provide clock signals for the swtiched-current memory cell and also to generate control signals for the digital section. For this work, these digital blocks were modelled using VHDL and then synthezed for the TSMC 0.35um target technology.

The analog and the digital sections were combined to form the entire mixed signal circuit. The system was designed for TSMC 0.35u fabrication process available from MOSIS. The BSIM3 v3.1 transistor models were used for all analog circuit simulations.

This document is organized as follows. The background information about switched-current technique and the memory cell, used in the design of 2D-DCT, is provided in chapter 2. Chapter 3 shows in detail the design and simulation of the DAC used to provide current input to the 2D-DCT. The structure and working of the 2D-DCT is explained in Chapter 4. This chapter also provides detailed analysis of the design of each block used to create the 2D-DCT and discusses the verification by providing relevant simulation results for each individual block. Chapter 5 describes the implementation of the analog to digital converter. The digital section of the system consisting of the storage blocks, quantizer and
entropy encoder is covered in Chapter 6. This chapter also discusses the design of the control block for the entire system. Finally in chapter 7, the integration of all these individual blocks to form the entire circuit is described. The final simulation results for the circuit are discussed. Chapter 8 summarizes what was accomplished from this work and directions for future efforts.
Chapter 2

Switched-Current Circuits

Linear, discrete time systems can be implemented using three basic elements; adders, multipliers (scaling elements) and delay elements [7]. These elements can be realized in a simple way when current is used as the means of interaction between different circuit elements.

![Figure 2.1: Implementation of adder and multiplier using current source](image)

An adder is simply a node at which two or more currents add up based on the Kirchoff’s current law. A multiplier (scaling element) with inversion is obtained from the output of the current source as shown in the figure 2.1. A delay element is implemented using a switched-current (SI) memory cell. Figure 2.2 presents the first generation memory cell implemented using a current mirror and a switch.

In order to reduce non-idealities such as transistor parameter mismatch, finite input-output conductance, output current settling, clock feed-through (charge injection), drain to
gate capacitive coupling, noise etc, various structures for memory cell have been developed [2]. In this work the modified switched-current memory cell [6] was used as the basic block in implementation of the 2D-DCT.

2.1 Modified Switched-Current Memory Cell

2.1.1 Design

The memory cell used for this work is shown in figure 2.3. This memory cell works in two stages, memorizing and reading. In the memorizing phase $\Phi_1$, the switches S1 and S2 are closed so input current $I_{in}$ is given to the two memory transistors $M_p$ and $M_n$. These transistors are now diode connected. $I_{in}$ charges the parasitic gate capacitance $C_{SGp}$ and $C_{GSn}$ of the transistors $M_p$ and $M_n$ to a voltage equal to their gate-source voltages $V_{SGp}$ and $V_{GSn}$ respectively.

In the reading phase $\Phi_2$, the output switch S3 is closed and the input switches are open. The capacitances $C_{SGp}$ and $C_{GSn}$ retain their charge and maintain a gate to source voltage of $V_{SGp}$ and $V_{GSn}$ respectively for $M_p$ and $M_n$ hence causing an output current in the reading stage same as the input current of the memorizing stage.

The drain currents, $I_{Dp}$ and $I_{Dn}$ for transistors operating in the saturation region are as
Figure 2.3: Modified switched-current memory cell

shown in equation 2.1.

\[
\begin{align*}
I_{D_p} &= \frac{\beta_p}{2} (V_{SG_p} + V_{Tp})^2 \\
I_{D_n} &= \frac{\beta_n}{2} (V_{GS_n} - V_{Tn})^2
\end{align*}
\tag{2.1}
\]

Assuming that, \( V_{GS} \) is the gate to source voltage and \( V_T \) is the threshold voltage for the transistors, \( V_{SG_p} = V_{S_p} - V_{G_p} \) and \( V_{GS_n} = V_{G_n} - V_{S_n} \) hence,

\[
\begin{align*}
I_{D_p} &= \frac{\beta_p}{2} (V_{S_p} - V_{G_p} + V_{Tp})^2 \\
I_{D_n} &= \frac{\beta_n}{2} (V_{G_n} - V_{S_n} - V_{Tn})^2
\end{align*}
\tag{2.2}
\]

Source of \( M_p = V_{DD} \) and \( M_n = \text{GND} (=0V) \)

\[
\begin{align*}
I_{D_p} &= \frac{\beta_p}{2} (V_{DD} - V_{G_p} + V_{Tp})^2 \\
I_{D_n} &= \frac{\beta_n}{2} (V_{G_n} - V_{Tn})^2
\end{align*}
\tag{2.3}
\]
The gates of the transistors $M_p$ and $M_n$ are connected so $V_{G_p} = V_{G_n} = V_G$

$$I_{D_p} = \frac{\beta_p}{2} (V_{DD} - V_G + V_{T_p})^2$$
$$I_{D_n} = \frac{\beta_n}{2} (V_G - V_{T_n})^2$$  \hspace{1cm} (2.4)

Now the output current $I_{out}$ can be expressed as

$$I_{out} = I_{D_n} - I_{D_p}$$

$$= \frac{\beta_n}{2} (V_G - V_{T_n})^2 - \frac{\beta_p}{2} (V_{DD} - V_G + V_{T_p})^2$$

$$= \frac{1}{2} ((\beta_n - \beta_p)V_G^2 + [\beta_n(V_{DD} - V_{T_n}) + \beta_p(V_{DD} + V_{T_p})]V_G)$$

$$+ \frac{1}{2} [\beta_n(V_{DD} - V_{T_n})^2 - \beta_p(V_{DD} + V_{T_p})^2]$$  \hspace{1cm} (2.5)

In the output current equation 2.5, there are three terms. The first term, gives the distortion in the output. The second term, gives the linear dependence of output current on $V_G$. While the last term specifies the DC offset.

By making $\beta_n = \beta_p$, the distortion can be eliminated. This can be achieved by adjusting the length and width of the transistors as shown in equation 2.6.

$$\beta_n = \beta_p$$

$$\mu_n C_{ox} (W/L)_n = \mu_p C_{ox} (W/L)_p$$  \hspace{1cm} (2.6)

For $\beta_n = \beta_p$, same level of voltages can be obtained at the input and the output and this helps in minimizing the transmission error in case of cascaded memory cells. The dc offset can be removed by using a balanced structure. In this structure a differential input is used and since both outputs have same dc offset value, it gets cancelled out in the combined output.

### 2.1.2 Simulation

The switched-current memory cell shown in figure 2.3 was designed in OrCAD Capture as shown in the figure 2.5. The schematic was simulated using PSpice. A triangular input
with a peak value of 200uA was given as an input to the memory cell. The switches in the memory cell were designed as shown in figure 2.6 and the inverter used in these switches is shown in the figure 2.7. The output load was a diode connected complementary pair. The transistor sizes are as shown in the table 2.1. The simulation results are shown in the figure 2.4.

<table>
<thead>
<tr>
<th>Transistor</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>2u</td>
<td>12u</td>
</tr>
<tr>
<td>U2</td>
<td>2u</td>
<td>4u</td>
</tr>
<tr>
<td>SW:U5</td>
<td>1.2u</td>
<td>7.2u</td>
</tr>
<tr>
<td>SW:U6</td>
<td>1.2u</td>
<td>2.4u</td>
</tr>
<tr>
<td>INV:U7</td>
<td>0.4u</td>
<td>0.8u</td>
</tr>
<tr>
<td>INV:U8</td>
<td>0.4u</td>
<td>2.4u</td>
</tr>
<tr>
<td>U3</td>
<td>4u</td>
<td>36u</td>
</tr>
<tr>
<td>U4</td>
<td>4u</td>
<td>12u</td>
</tr>
</tbody>
</table>

Table 2.1: Transistor sizes for switched-current memory cell

In figure 2.4, the voltage source V2 forms clock PHI1 and the voltage source V3 forms PHI2. When the PHI1 phase was high, the input current of 200uA was memorized and then during the PHI2 phase the current observed at the output was 193.35uA. This was almost same as the input current of the memorizing stage. Hence these results show that the memory cell operated as desired.
Figure 2.4: Simulation results for switched-current memory cell
Figure 2.5: Simulation setup for switched-current memory cell
Figure 2.6: Switch used in switched-current memory cell

Figure 2.7: Inverter used for the switches
Chapter 3

Digital to Analog Converter

The input to the image compression system is given through an 8-bit digital to analog converter (DAC) for the purpose of simplification during testing of the combined circuit at a later stage. The output of DAC forms the input for the 2D-DCT block which is a differential, current mode circuit. So a current mode DAC was designed.

3.1 Design

A current mode 8-bit DAC was implemented by using the structure shown in figure 3.1. A reference current was given as an input to a series of current mirrors. The transistors for these current mirrors were varying in their size by \( \frac{1}{2^n} \) where \( n=(0,1...7) \). The output of these current mirrors was passed through switches which were controlled by the corresponding digital inputs \( D_n \). The outputs of the switches were then combined to obtain the final output current.

A negative reference current was also passed through a similar structure to give a negative output. Eight such DAC were used to give eight differential inputs to the 2D-DCT block. The transistor sizes used in the DAC design are shown in table 3.1. In order to avoid using transistors with large widths, the length of some transistors were modified in order to achieve the desired W/L ratio.
Figure 3.1: 8-bit DAC
<table>
<thead>
<tr>
<th>Transistor</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>U5, U6</td>
<td>2u</td>
<td>38.4u</td>
</tr>
<tr>
<td>U7</td>
<td>4u</td>
<td>38.4u</td>
</tr>
<tr>
<td>U8</td>
<td>4u</td>
<td>19.2u</td>
</tr>
<tr>
<td>U9</td>
<td>4u</td>
<td>9.6u</td>
</tr>
<tr>
<td>U10</td>
<td>8u</td>
<td>9.6u</td>
</tr>
<tr>
<td>U11</td>
<td>8u</td>
<td>4.8u</td>
</tr>
<tr>
<td>U12</td>
<td>8u</td>
<td>2.4u</td>
</tr>
<tr>
<td>U13</td>
<td>8u</td>
<td>1.2u</td>
</tr>
<tr>
<td>U14, U15</td>
<td>2u</td>
<td>12.8u</td>
</tr>
<tr>
<td>U16</td>
<td>4u</td>
<td>12.8u</td>
</tr>
<tr>
<td>U17</td>
<td>4u</td>
<td>6.4u</td>
</tr>
<tr>
<td>U18</td>
<td>4u</td>
<td>3.2u</td>
</tr>
<tr>
<td>U19</td>
<td>8u</td>
<td>3.2u</td>
</tr>
<tr>
<td>U20</td>
<td>8u</td>
<td>1.6u</td>
</tr>
<tr>
<td>U21</td>
<td>8u</td>
<td>0.8u</td>
</tr>
<tr>
<td>U22</td>
<td>8u</td>
<td>0.4u</td>
</tr>
<tr>
<td>SW: U67</td>
<td>4u</td>
<td>24u</td>
</tr>
<tr>
<td>SW: U68</td>
<td>4u</td>
<td>8u</td>
</tr>
<tr>
<td>INV: U41</td>
<td>4u</td>
<td>24u</td>
</tr>
<tr>
<td>INV: U42</td>
<td>4u</td>
<td>8u</td>
</tr>
</tbody>
</table>

Table 3.1: Transistor sizes for DAC

### 3.2 Simulation

DAC was given an input reference current of 200uA. Different values of 8-bit digital inputs were given as control inputs for the switches. The outputs of these switches were connected together to get the corresponding analog current output. This output current was then passed through a load which was formed by a diode connected transistor pair. The transistor sizes for the load were $L_p = L_n = 4u$, $W_p = 36u$ and $W_n = 12u$.

Table 3.2 presents the results after simulating the DAC for different sets of digital inputs and then compares this result with theoretical calculations. For eg. an eight bit digital input of 00100101 was given to the DAC circuit. The theoretical output current was calculated first as shown by the equation 5.1

$$I_{out} = 200 \times \left\{ \left[ \frac{1}{2^0} \times 0 \right] + \left[ \frac{1}{2^1} \times 0 \right] + \left[ \frac{1}{2^2} \times 1 \right] + \left[ \frac{1}{2^3} \times 0 \right] + \left[ \frac{1}{2^4} \times 0 \right] \right\}$$
\[ +\left[ \frac{1}{25} \times 1 \right] + \left[ \frac{1}{36} \times 0 \right] + \left[ \frac{1}{27} \times 1 \right] \] \quad \text{uA} \quad (3.1)

\[ = \quad 200 \times \left\{ 0 + 0 + \frac{1}{4} + 0 + 0 + \frac{1}{32} + 0 + \frac{1}{128} \right\} \] \quad \text{uA}

\[ = \quad 57.81 \text{uA} \]

The results in table 3.2 show that the output obtained after simulation of the DAC design had negligible difference when compared to theoretical output value. The output waveforms corresponding to an 8-bit digital input of 00100101 and a reference current of 200uA are shown in figure 3.2.

<table>
<thead>
<tr>
<th>Digital Input</th>
<th>( I_{\text{out\ (theoretical)}} )</th>
<th>( +I_{\text{out\ (actual)}} )</th>
<th>( -I_{\text{out\ (actual)}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>0u</td>
<td>0u</td>
<td>0u</td>
</tr>
<tr>
<td>01000000</td>
<td>100u</td>
<td>98u</td>
<td>-107.6u</td>
</tr>
<tr>
<td>11111111</td>
<td>398.43u</td>
<td>360.5u</td>
<td>-363.8u</td>
</tr>
<tr>
<td>00100101</td>
<td>57.81u</td>
<td>56.5u</td>
<td>-57.5u</td>
</tr>
<tr>
<td>10000000</td>
<td>200u</td>
<td>184u</td>
<td>-184.7u</td>
</tr>
</tbody>
</table>

Table 3.2: Simulation results for DAC for an input reference current of 200uA

![Figure 3.2: Simulation result for DAC](image)
Chapter 4

Two Dimensional Discrete Cosine Transform

The 2D-DCT, $Y$, for an input image $X$ of size $N \times N$ pixels is given as shown in equation 4.1 [9].

$$ Y(k, l) = \frac{2}{N} C(k)C(l) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X(i, j) \cos \left( \frac{(2i + 1)k\pi}{2N} \right) \cos \left( \frac{(2j + 1)l\pi}{2N} \right), $$

(4.1)

where $k, l = 0, 1, 2, \ldots, N-1$; and

$$ C(k), C(l) = \begin{cases} 
\frac{1}{\sqrt{2}}, & \text{for } k, l = 0; \\
0, & \text{otherwise} 
\end{cases} \quad (4.2) $$

The matrix form of 2D-DCT is as shown in equation 4.3 where $Y$ is the output 2D-DCT matrix and $X$ is the $N \times N$ input matrix. The coefficients of matrix $C$ are represented as shown in equation 4.4

$$ Y = CXC^t, $$

(4.3)
\[ C = \frac{1}{2} \begin{bmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{bmatrix} \] (4.4)

\[ a = \cos(\pi/16), \quad b = \cos(2\pi/16), \quad c = \cos(3\pi/16), \quad d = \cos(4\pi/16), \quad e = \cos(5\pi/16), \quad f = \cos(6\pi/16), \quad g = \cos(7\pi/16) \]

The 2D-DCT can be easily implemented from two one dimensional discrete cosine transform (1D-DCT) by using the property of separability as shown in equation 4.5

\[ Y = Z^tC^t, \quad Z = X^tC^t \] (4.5)

\[ X \] is the input 8x8 block and \( X^t \) indicates that it is transposed. \( Z \) is the intermediate matrix, which is obtained from 1D-DCT and is saved in a memory array. These intermediate results are processed along with the input matrix \( X \) and the output signal matrix \( Y \) is obtained.

The block diagram of the 2D-DCT implemented in this thesis is as shown in figure 4.1. The input to the 2D-DCT comes from the DAC. Since 8 differential inputs were required, 8 DAC blocks were used at the input. The main components of the 2D-DCT are the 1D-DCT and the switched current memory block (SIM). Since the final output from the 2D-DCT was differential but the ADC accepted only single ended inputs, an additional block was designed to convert the differential output of the 2D-DCT to a single ended output. The design and simulation of these components is explained in the following sections.
4.1 One Dimensional Discrete Cosine Transform

4.1.1 Design

The 1D-DCT block was implemented using 16 current mirrors with scaling factors of a, b, c, d, e and f corresponding to the elements of the matrix C shown in equation 4.4. The outputs of the current mirrors were connected to each other, in order to add the currents and realize the matrix relations. The balanced structure allowed to realize the negative elements of the matrix. The structure of the 1D-DCT is as shown in the figure 4.2 The transistor sizes for the current mirrors to realize the C matrix are as shown in the table 4.1. The length and widths were actually determined for $\frac{1}{2}C$. 

Figure 4.1: Block diagram of 2D-DCT
Figure 4.2: 1D-DCT

<table>
<thead>
<tr>
<th>PMOS</th>
<th>NMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>$W$</td>
</tr>
<tr>
<td>4u</td>
<td>36u</td>
</tr>
<tr>
<td>4u</td>
<td>18u</td>
</tr>
<tr>
<td>4u</td>
<td>16.6u</td>
</tr>
<tr>
<td>4u</td>
<td>15u</td>
</tr>
<tr>
<td>4u</td>
<td>12.6u</td>
</tr>
<tr>
<td>4u</td>
<td>10u</td>
</tr>
<tr>
<td>4u</td>
<td>7.2u</td>
</tr>
<tr>
<td>4u</td>
<td>3.6u</td>
</tr>
</tbody>
</table>

Table 4.1: Transistor sizes for 1D-DCT
4.1.2 Simulation

The 1D-DCT was simulated individually by giving constant current signals at its input. Simulation was done with a diode connected complementary pair as the output load. The transistor sizes for the load were same as the first complementary pair at the input of 1D-DCT, \( L_p = L_n = 4u \), \( W_p = 36u \) and \( W_n = 12u \). The corresponding output was then compared with results calculated using an inbuilt function for 1D-DCT within Matlab. The results for the output matrix for two different sets of inputs are shown in table 4.2 and table 4.3.

<table>
<thead>
<tr>
<th>Input (uA)</th>
<th>Matlab Output</th>
<th>Output(uA)</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+I_out</td>
<td>-I_out</td>
</tr>
<tr>
<td>100</td>
<td>282.84</td>
<td>253.6</td>
<td>-259.5</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-2.7</td>
<td>-2.7</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-6.61</td>
<td>-6.61</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-2.7</td>
<td>-2.7</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-3.35</td>
<td>-3.35</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-2.7</td>
<td>-2.7</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-6.61</td>
<td>-6.61</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>-2.7</td>
<td>-2.7</td>
</tr>
</tbody>
</table>

Table 4.2: Simulation results for 1D-DCT with 100uA constant current input

<table>
<thead>
<tr>
<th>Input (uA)</th>
<th>Matlab Output</th>
<th>Output(uA)</th>
<th>Actual Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+I_out</td>
<td>-I_out</td>
</tr>
<tr>
<td>100</td>
<td>212.13</td>
<td>189.6</td>
<td>-195.8</td>
</tr>
<tr>
<td>100</td>
<td>13.79</td>
<td>10.6</td>
<td>-16.2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-6.5</td>
<td>-6.5</td>
</tr>
<tr>
<td>100</td>
<td>39.28</td>
<td>36.5</td>
<td>-42</td>
</tr>
<tr>
<td>100</td>
<td>70.71</td>
<td>61</td>
<td>-67.6</td>
</tr>
<tr>
<td>100</td>
<td>-58.79</td>
<td>-57.6</td>
<td>52.2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-6.6</td>
<td>-6.6</td>
</tr>
<tr>
<td>100</td>
<td>-69.35</td>
<td>-66.9</td>
<td>61.4</td>
</tr>
</tbody>
</table>

Table 4.3: Simulation results for 1D-DCT
4.2 Switched-Current Memory Block

4.2.1 Delay Cell

The basic building block of a switch-current memory block is the delay cell. As shown in figure 4.3, the delay cell was designed using a balanced structure. This helped in compensating the parasitic effects like dc offset, crosstalk from digital signals, clock feed-through and charge injection [7].

The balanced structure of the delay cell was formed using four switched-current memory cells. The input was sampled in each half of the clock period. Figure 2.6 and figure 2.7 show the structure of the switches and the inverter. The transistor sizes for the delay cell are shown in the table 4.4

<table>
<thead>
<tr>
<th>Transistor</th>
<th>Length</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI: U1,U2,U3,U4</td>
<td>4u</td>
<td>36u</td>
</tr>
<tr>
<td>SI: U5,U6,U7,U8</td>
<td>4u</td>
<td>12u</td>
</tr>
<tr>
<td>SW: U1</td>
<td>1.2u</td>
<td>7.2u</td>
</tr>
<tr>
<td>SW: U2</td>
<td>1.2u</td>
<td>2.4u</td>
</tr>
<tr>
<td>SWI: U1</td>
<td>.4u</td>
<td>36u</td>
</tr>
<tr>
<td>SWI: U2</td>
<td>.4u</td>
<td>12u</td>
</tr>
<tr>
<td>SW2: U1</td>
<td>.4u</td>
<td>36u</td>
</tr>
<tr>
<td>SW2: U2</td>
<td>.4u</td>
<td>12u</td>
</tr>
<tr>
<td>INV: U1</td>
<td>.4u</td>
<td>.8u</td>
</tr>
<tr>
<td>INV: U2</td>
<td>.4u</td>
<td>2.4u</td>
</tr>
<tr>
<td>LOAD: U3,U5</td>
<td>4u</td>
<td>36u</td>
</tr>
<tr>
<td>LOAD: U4,U6</td>
<td>4u</td>
<td>12u</td>
</tr>
</tbody>
</table>

Table 4.4: Transistor sizes for delay cell

The delay cell was simulated with an input of 200uA. Both output currents were delivered to the load which was also a complementary pair in diode connection. The simulation setup is as shown in figure 4.4. The corresponding simulation waveforms are shown in figure 4.5. The voltage pulse V6 is given as clock to signals W1 and R2 while V7 is given as clock to W2 and R1. The input is memorized when V6 is high and is read out when V7 is high. The output is delayed by half clock cycle.
Figure 4.3: Delay cell
Figure 4.4: Simulation setup for delay element

Figure 4.5: Simulation results for delay element
4.2.2 Design of Switched-Current Memory Block

The memory block consists of an array of delay cells as shown in figure 4.6. The inputs

![Diagram of SIM](image)

Figure 4.6: Block diagram of SIM

from the 1D-DCT were written (i.e. stored) in the memory cells in the one half of the clock period and read out in the next half. These inputs were memorized row by row and given out column by column. The outputs from the memory block were again given as inputs to the second 1D-DCT which processes the results in the same way as the first 1D-DCT giving a 2D-DCT at its output. The clock signals for the memory block were generated in the control block. The control block was a digital component. The design of the control block is discussed in chapter 6. In order to test the memory block, a small test code was written for a 16 bit ring counter. The 16 outputs were used as clocks for the memory block.
A 2x2 memory block was created first and tested as shown in figure 4.7. The two inputs to this memory block were constant current sources of amplitude 100uA and 200uA. Each output was connected to a diode connected complementary pair acting as the load. The transistor sizes for the load were $L_p = L_n = 4u$, $W_p = 36u$ and $W_n = 12u$.

The internal structure of this memory block consisted of four delay cells connected as shown in figure 4.8. Each delay cell consists of four memory cells as shown in figure 4.3. During the first clock pulse PHI10, the inputs were written in the first memory cell of the delay cells D1 and D2 (the row cells) and at the same time the outputs were read out from the second memory cell of delay cells D1 and D3 (the column cells). Similarly, when the clock PHI20 went high, the inputs were written into the delay cells D3 and D4 and the outputs were read out from D2 and D4. The process was repeated again during PHI11 and the PHI21 clock signals as shown in the figure 4.8.

The corresponding output results for this memory block were as shown in table 4.5.
The outputs from the clock generator giving the four clock signals PHI10, PHI11, PHI20 and PHI21 for the memory block and the output currents $I_{out1}$ and $I_{out2}$ are shown in the figure 4.9.

\begin{center}
\begin{tabular}{|c|cc|}
\hline
Input (uA) & Output(uA)  \\
\hline
 & $+I_{out}$ & $-I_{out}$  \\
100 & -94.64 & 101.42  \\
200 & -198.78 & 200.45  \\
\hline
\end{tabular}
\end{center}

Table 4.5: Simulation results for 2x2 memory block

4.2.3 Simulation of Switched-Current Memory Block

The entire SIM was simulated for two different sets of the inputs. Both these sets of inputs were given directly using constant current sources The results for the first set of simulation are shown in table 4.6.
Figure 4.9: Output from 2x2 memory block

The second input was a constant current source 282uA for the first row of inputs and 0uA for the rest. This set of inputs was the same as the output coming from 1D-DCT for an input of 100uA. The SIM was tested with this input and the results are as shown in the table 4.7. The output waveforms corresponding to the first set of input simulation are shown in figure 4.10.

<table>
<thead>
<tr>
<th>Input (uA)</th>
<th>Output(uA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+I_{out}$</td>
<td>$-I_{out}$</td>
</tr>
<tr>
<td>100</td>
<td>-99.34</td>
</tr>
<tr>
<td>200</td>
<td>-195.61</td>
</tr>
<tr>
<td>300</td>
<td>-288.66</td>
</tr>
<tr>
<td>400</td>
<td>-367.07</td>
</tr>
<tr>
<td>50</td>
<td>-46.64</td>
</tr>
<tr>
<td>150</td>
<td>-145.78</td>
</tr>
<tr>
<td>250</td>
<td>-243.98</td>
</tr>
<tr>
<td>350</td>
<td>-331.38</td>
</tr>
</tbody>
</table>

Table 4.6: Simulation results for memory block with first set of inputs
Figure 4.10: Output from memory block with first set of inputs

<table>
<thead>
<tr>
<th>Input (uA)</th>
<th>Output(uA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$+I_{out}$</td>
</tr>
<tr>
<td>282</td>
<td>277.4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.7: Simulation results for memory block with second set of inputs
4.3 Differential to Single Ended Current Converter

The 2D-DCT gives out a differential output while the ADC takes in a single ended current input. So an additional block was designed in order to convert the 16 output currents of the 2D-DCT to 8 current inputs for the ADC. A current mirror structure was designed. The transistor sizes were defined as shown in figure 4.11.

The input \((I_{in}^+)\) was passed on through two stages of current mirrors. The transistor sizes were maintained so that after the first stage, the output obtained is \([- (I_{in}^+)/2]\). Finally at the output of the second stage, the current obtained is \([(I_{in}^+)/2]\).

Similarly, the input \((I_{in}^-)\) is passed on through one stage of current mirror to obtain \([- (I_{in}^-)/2]\). Both these currents are combined to get \([(I_{in}^+) - (I_{in}^-)]/2\) as the final output which is passed on to the ADC.

Eight such circuits were used at the output of the 2D-DCT to get 8 single ended input currents for the ADC.

The design shown in Fig 4.11 was simulated for different sets of inputs. The results of the simulations are shown in Table 4.8. These results are comparable to the theoretical values as obtained by using the formula \([(I_{in}^+) - (I_{in}^-)]/2\). The results in the table 4.8 show that the output obtained is comparable with the theoretical values. It was observed that if the input currents are of magnitude higher than 2mA, then the transistor sizes had to be changed in order to obtain output within negligible difference of the theoretical values.

<table>
<thead>
<tr>
<th>(I_{in}^+)</th>
<th>(I_{in}^-)</th>
<th>(I_{out})</th>
</tr>
</thead>
<tbody>
<tr>
<td>500uA</td>
<td>-500uA</td>
<td>529.8u</td>
</tr>
<tr>
<td>400u</td>
<td>300u</td>
<td>54.15u</td>
</tr>
<tr>
<td>-600u</td>
<td>-300u</td>
<td>-174.7u</td>
</tr>
<tr>
<td>-800u</td>
<td>500u</td>
<td>-672.6u</td>
</tr>
</tbody>
</table>

Table 4.8: Results of differential to single ended current converter
Figure 4.11: Differential to single ended current converter
4.4 Design and Simulation of 2D-DCT

The 1D-DCT and memory block were combined together to form the circuit for 2D-DCT. The output of the 2D-DCT block was passed through the differential to single ended current converter to get the final 8 outputs. The input to the circuit was given using 8 DAC blocks, each block taking a 200µA constant current source as the reference current and a digital input of 01000000 to give a +100µA and -100µA output. Diode connected transistors were used as load at the output. The transistor sizes for the load are \( L_p = L_n = 4u, W_p = 36u \) and \( W_n = 12u \). The 16 clock signals for SIM were coming from the control block.

The simulation results for the 2D-DCT are shown in table 4.9. These results were compared with results obtained from Matlab. The inbuilt function in Matlab to obtain 2D-DCT of an 8x8 input matrix was used. The same set of inputs were given as a matrix in Matlab and the outputs obtained was then compared with the simulation results. As shown in table 4.9, the outputs obtained from simulation have negligible difference when compared with the Matlab output and hence these results verify the functionality of the 2D-DCT designed using switched-current technique.

<table>
<thead>
<tr>
<th>Input (µA)</th>
<th>Output(µA)</th>
<th>Matlab output for 2D-DCT</th>
<th>( I_{out} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>800</td>
<td>747.76</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.9: Simulation results of 2D-DCT
Chapter 5

Analog to Digital Converter

The output current from the 2D-DCT was converted to a digital voltage using an analog to digital converter (ADC). This output was passed on to the digital section for further processing.

The basic component in the ADC design was a comparator. The design and simulation of the comparator is explained in the following section.

5.1 Comparator

For this work the comparator shown in figure 5.1 was used. The two currents to be compared were first given to a difference current generator. Here using current mirrors, the difference current was obtained as shown in figure 5.2. This difference current was then passed on to another stage of current mirror. The output voltage of this stage was high if the difference was positive and the output was low for a negative difference. This output was inverted to get the complimentary output. To obtain an absolute high or low value for the output voltage, another inverter stage was added. Finally both the outputs were passed on through a latch. The structure for the comparator and the difference current generator used in this work is shown in figure 5.1 and figure 5.2. The transistor sizes used for the comparator design were large. The reason for such large sizes was that the current coming out of the 2D-DCT was in milliampere range. So support such high current values, the transistors needed to have a large W/L ratio.
Figure 5.1: Comparator used for analog to digital converter

I_{out} = I_{in} - I_{ref}

Figure 5.2: Difference current generator used for the comparator
5.2 Design of ADC

The detailed structure for the 5-bit ADC designed in this work is shown in figure 5.3. The number of comparators used in the design for ADC was equal to the number of output bits.

Figure 5.3: 5-bit current mode ADC

The current output from the 2D-DCT was given to the comparators and each time the input current was compared against a different reference current. The MSB was determined by comparing the input current \(I_{in}\) with a reference current \(I_{ref}\). The MSB was used to control a switch which passes on a current \(I_{ref}/2\). If \(I_{in} < I_{ref}\), \(Q_5 = 0\) and \(Q^\bar{5} = 1\), then the reference current \(I_{ref}/2\) was added to the input current \(I_{in}\) and then compared again with \(I_{ref}\). In the opposite scenario where \(I_{in} > I_{ref}\), \(I_{ref}/2\) was added to \(I_{ref}\) and the sum was compared against \(I_{in}\). The result of this second comparison determined the next significant bit. It was a basically a pipelined structure where the previous bit determines the next bit and finally you get the 5-bit output from the ADC. Basically, an n-bit structure designed this way used n comparators, \(n(n-1)/2\) switches and n reference currents and current
mirrors. In this work in the current mirrors were initially used to generate 5 input currents from a single current input and same way the reference currents were also generated using current mirrors.

5.3 Simulation of ADC

The ADC design shown in figure 5.3 was simulated by giving constant currents at the input. The simulation results for different values of input and reference currents are shown in table 5.1. The actual output value was taking at the falling edge of the input signal since there was some ambiguity at the rising edge of the input signal because of the switches.

The output of the 2D-DCT for an input current of 100uA was in the range of 750uA to 820uA. For this range the digital output obtained from the ADC was 01100. The theoretical analog value of current corresponding to this digital output was calculated as shown in the equation ???. The reference current used was 1mA(=1000uA).

\[
I_{out} = 1000 \times \left\{ \left[ \frac{1}{2^0} \times 0 \right] + \left[ \frac{1}{2^1} \times 1 \right] + \left[ \frac{1}{2^2} \times 1 \right] + \left[ \frac{1}{2^3} \times 0 \right] + \left[ \frac{1}{2^4} \times 0 \right] \right\} \mu A \quad (5.1) \\
= 1000 \times \left\{ 0 + \frac{1}{2^1} + \frac{1}{2^2} + 0 + 0 \right\} \mu A \\
= 750 \mu A
\]

This theoretical value obtained is comparable to the input current value to the ADC. This shows that the desired output was obtained from the ADC.

<table>
<thead>
<tr>
<th>( I_{in} )</th>
<th>( I_{ref} )</th>
<th>( Q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 62uA</td>
<td>1mA</td>
<td>00000</td>
</tr>
<tr>
<td>250uA - 310uA</td>
<td>1mA</td>
<td>00100</td>
</tr>
<tr>
<td>1.25mA - 1.3mA</td>
<td>1mA</td>
<td>10100</td>
</tr>
<tr>
<td>750uA - 820uA</td>
<td>1mA</td>
<td>01100</td>
</tr>
</tbody>
</table>

Table 5.1: Simulation results for ADC
Figure 5.4: Simulation results for 5-bit ADC
Chapter 6

Digital Section

The major components of the digital section are storage registers, zig-zag ordering block, quantization block, entropy encoder and the control block. For this thesis, the individual blocks of the digital section were modelled using VHDL. The JPEG compression standard was followed in this section [14]. The block diagram for this section is shown in figure 6.1.

![Block diagram for digital section]

Figure 6.1: Block diagram for digital section

The digital output from ADC was stored in the storage registers and once all the 64 bytes of data from one input block of 8x8 pixel, were stored, processing started for 64 inputs at a time. The inputs undergo a zig-zag scan and are stored as 1x64 vector. The
quantization matrix was also read out in a zig-zag format and then each input was quantized by its corresponding value in the quantization matrix. Finally the huffman code was taken from the lookup table for each quantized input value and formed the final compressed output. The matrix used for quantization and its corresponding huffman codes were taken from the luminance matrix [14]. The details of each section in the flow-chart shown in figure 6.2 are described in the following sections. The control block generated the control signals for all the blocks of the digital section. It also generated the 16 clock signals for the switched current memory block.

6.1 Storage Registers

The input to this experimental image compression system was given in the form of 8x8 pixel blocks. The processing of the data was done column after column. The digital section processed all the 64 data together. So before the digital output of the ADC could be passed on to the digital section, a storage unit had to be designed which stored the data till all the 64 data bytes were present.

The basic components of the storage block were 8-bit registers. The first column of the storage block consisted of registers which took in a 5-bit data coming from ADC and added three least significant zero bits to give an 8-bit output when the load signal became high. This component was modelled in VHDL. The VHDL code was then synthesized into TSMC 0.35u technology cells. The simulation results for this register, obtained from ModelSim are shown in figure 6.3.

The remaining columns of the storage block were made of 8-input 8-output registers. The simulation results for these registers are shown in figure 6.4.

Using these two register components, a VHDL code was written to create the storage block with 64 registers. The first column of registers in the storage unit stores the data coming from the ADC. It then shifts this data to the next column while the first column takes in the next set of data from ADC. This chain went on till all the 8 column data was
Set
8-bit inputs from ADC and store them in storage registers till all 64 bytes are processed

Zigzag scan of the inputs and storing them in a 1x64 vector in the memory

Reading inputs from quantization matrix in a zigzag format and then doing quantization for each element of the input matrix

Get the huffman code corresponding to each input quantized matrix coefficient

Get output in the form of huffman codes 8-bits at a time

Figure 6.2: Flow chart for digital section

Figure 6.3: Simulation results for register with 5-bit input and 8-bit output
stored in this block. The control signals for the registers in this block were coming from the control unit described in the following sections.

6.2 Zig-zag Ordering

Zig-zag ordering is arranged based on increasing spatial frequency. The DC coefficient is the average value of the 64 image samples. It contains a significant portion of the total energy of the 8x8 block of image samples. The AC coefficients represent frequency components of the input matrix. Statistically, the low frequency components contain more energy than the high frequency components. Also the human perceptual system is more sensitive to the low frequency components than the high frequency components. After quantization there is a high probability that the values of the higher frequency components will be zero. Hence to achieve the best coding efficiency, the DC coefficient should be encoded separately from the AC coefficients and the low frequency components should be placed before the high frequency components which are zeros [12]. This was achieved by zig-zag ordering.
of a matrix as shown in equation 6.1.

\[
\begin{bmatrix}
1 & 2 & 6 & 7 & 15 & 16 & 28 & 29 \\
3 & 5 & 8 & 14 & 17 & 27 & 30 & 43 \\
4 & 9 & 13 & 18 & 26 & 31 & 42 & 44 \\
10 & 12 & 19 & 25 & 32 & 41 & 45 & 54 \\
11 & 20 & 24 & 33 & 40 & 46 & 53 & 55 \\
21 & 23 & 34 & 39 & 47 & 52 & 56 & 61 \\
22 & 35 & 38 & 48 & 51 & 57 & 60 & 62 \\
36 & 37 & 49 & 50 & 58 & 59 & 63 & 64 \\
\end{bmatrix}
\]

(6.1)

The flow chart for implementing zig-zag ordering is shown in the figure 6.5. The 2D-DCT

![Flow chart for zig-zag ordering](image)

Figure 6.5: Flow chart for zig-zag ordering

output after being digitized in the ADC, is stored in the storage block. The data from the storage block is read in as input for the zig-zag ordering block. Each input is read in and stored in a matrix in the zig-zag order and the output is given to the quantization block.

In order to test the code for zig-zag ordering, a testbench was created to read 64 input data from a text file and after processing the data, the output was obtained in the zig-zag format. Different sets of inputs were tested. The input and output values for one such test are shown in equation 6.2 and figure 6.6. The input values are decimal numbers from 1 to
64 arranged in an 8x8 matrix. The simulation results show that the output is a 1x64 vector with the input values in a zig-zag format shown in equation 6.1. The values are displayed in decimal format in the simulation results shown in figure 6.6 for easier verification of results.

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\
25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 \\
33 & 34 & 35 & 36 & 37 & 38 & 39 & 40 \\
41 & 42 & 43 & 44 & 45 & 46 & 47 & 48 \\
49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 \\
57 & 58 & 59 & 60 & 61 & 62 & 63 & 64 \\
\end{bmatrix}
\]

(6.2)

Figure 6.6: Output file from zig-zag ordering simulation
6.3 Quantization

Quantization is a lossy compression process. Each co-efficient of the 2D-DCT is divided by a user defined quantizer. Hence in order to achieve quantization of the 64 coefficients, a matrix formed by 64 quantization elements is used. This quantization matrix is not fixed. It can be a user defined matrix depending on the type of application and the compression factor required for the particular application. After dividing each coefficient by its corresponding quantizer, the output is rounded to the nearest integer as described by the formula in equation 6.3 [12].

\[
Y'(k, l) = \text{round} \left( \frac{Y(k, l)}{Q(k, l)} \right)
\]

where,

- \text{round} = \text{function} to round a variable to its nearest integer
- \(Y(k, l)\) = 2D-DCT matrix
- \(Q(k, l)\) = Quantization matrix
- \(k, l = 0, 1, 2...7\)

The JPEG standard provides examples of the quantization matrix [17]. In this thesis, the luminance matrix shown in equation 6.4 was used as the quantization matrix. If a better compression ratio was required then the values in this matrix could be multiplied by 2. So basically depending on the compression ratio required the values for the \(Q\) matrix are determined.
Each component of the 2D-DCT matrix $Y$ was divided by the corresponding component of the matrix $Q$ and then rounded to its nearest integer value. The rounding was achieved by using the relation shown in equation 6.5 [13]. In the resultant matrix $Y'$, the higher frequency components were divided by larger values in $Q$ and hence these higher frequency components were not as important to the image quality.

$$Y'(k, l) = \begin{cases} 
\frac{Y(k, l) + Q(k, l)}{Q(k, l)}, & \text{if } Y(k, l) \geq 0; \\
\frac{Y(k, l) - Q(k, l)}{Q(k, l)}, & \text{if } Y(k, l) < 0. 
\end{cases} \quad (6.5)$$

The implementation of quantization was done using the flowchart shown in figure 6.7.

The 1x64 vector output obtained from zig-zag scan, was passed on as input to the quantization block. Corresponding to this input value, an address for the lookup table was generated in the input stage. This address was used to obtain the corresponding quantizer value from the quantization matrix. The quantizer values were also obtained in zig-zag format since the input values were in that format. In order to achieve this, the quantization values in the lookup table were stored in zig-zag format.

Once the input and its corresponding quantizer value were obtained, the next step was to do rounding using equation 6.5. The quantizer value was divided by 2 and then added or subtracted from the input value depending on the conditions specified in the equation. This formed the rounding stage.

Once the input was adjusted by rounding, it was passed on to the dividing stage to
Output of the zigzag block is given as the input to the quantization block.

An address to the lookup table is generated corresponding to the input value.

Using the address generated for the lookup table, a corresponding quantizer value is chosen.

Rounding is done by adding or subtracting half of the quantizer value to the input value as shown in rounding equation.

9 stage diving block which takes the rounded input from the rounding stage and then divides it by corresponding quantizer from Q matrix. Each stage computes one bit of the final result.

Output of the dividing stage gives the quantized output where input is divided by quantizer and then rounded to the nearest integer.

Figure 6.7: Flow chart for quantization
obtain the final quantized output. The dividing was done using 9 pipelined stages. These 9 dividing stages were arranged in the order of most significant bit (MSB) to least significant bit (LSB). Each stage gave one bit of the result after performing division. The rounded input was now divided by the quantizer from the quantization matrix and the final quantized value was given as the output. This value was then passed on to the encoding stage.

The VHDL code for quantization was divided into four different stages as shown in the flowchart of figure 6.7. The input was read from a text file and was passed on to the zig-zag stage. The input text file given to the zig-zag stage for testing quantization is shown in figure 6.8. The simulation results for of the quantization component is shown in figure 6.9.

The simulation results shown in figure 6.9 are verified manually. The signal values in the waveforms are shown in decimal format so that the results can be verified easily.

![Input text file to test quantization](image)

### 6.4 Huffman Encoding

The quantized input data is passed on to the entropy encoding section. Huffman coding is one of the most popular entropy coding schemes. It produces the shortest possible average code length for a given set of inputs based on their associated probabilities. There is no fixed table for Huffman codes. In this thesis the Huffman table for the luminance matrix [14] was used.

The DC coefficient was encoded by using a DPCM (Differential Pulse Code Modulation) technique. This method is more efficient and it encodes the difference between the DC coefficient from the current 8x8 block and the coded DC coefficient from the previous...
8x8 block. In case of the first 8x8 block of the image the value for the previous block was set to 0 since there was no block before. This was a two step process. First the DPCM difference was taken and a corresponding category to which it belonged, was determined. Second, additional bits were assigned to specify the sign and the exact value of difference.

The difference magnitude categories [14] are shown in table 6.1. The SSSS value is a 4 bit value representing the size of the difference. The coding obtained from this table was

<table>
<thead>
<tr>
<th>Category (SSSS)</th>
<th>DPCM Difference DIFF Values</th>
<th>Code Length</th>
<th>Huffman Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td>1</td>
<td>-1,1</td>
<td>3</td>
<td>010</td>
</tr>
<tr>
<td>2</td>
<td>-3,-2,2,3</td>
<td>3</td>
<td>011</td>
</tr>
<tr>
<td>3</td>
<td>-7,-4,4,7</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>-15,-8,8,15</td>
<td>3</td>
<td>101</td>
</tr>
<tr>
<td>5</td>
<td>-31,-16,16,31</td>
<td>3</td>
<td>110</td>
</tr>
<tr>
<td>6</td>
<td>-63,-32,32,63</td>
<td>4</td>
<td>1110</td>
</tr>
<tr>
<td>7</td>
<td>-127,-64,64,127</td>
<td>5</td>
<td>11110</td>
</tr>
<tr>
<td>8</td>
<td>-255,-128,128,255</td>
<td>6</td>
<td>111110</td>
</tr>
</tbody>
</table>

Table 6.1: Difference magnitude categories and typical Huffman table [14]

encoded using Huffman tables which are primarily based on the probabilities of the values used. The more frequently used values have the shortest codes assigned to them.

The principle behind the AC coefficients is that the information may be concentrated into fewer low frequency transform coefficients and high frequency AC coefficients are likely to be zero after quantization. With zig-zag ordering of the DCT coefficients, most higher index elements of the zig-zag sequence are usually zero. The AC-coefficients were coded using an 8-bit value represented as RRRRSSSS. The run length, 4-bit RRRR value, is the number of zeros preceding a nonzero value using the zig-zag format of reading a matrix. The non-zero value was coded by size, 4-bit SSSS value, as was described for the difference magnitude [14]. Table 6.2 shows the possible combinations for the AC coefficient coding.

These RRRRSSSS values were encoded with a Huffman table as shown in table 6.3. The output was the Huffman code with the value, without the MSB, attached to the end of it. A part of the Huffman table used for encoding AC coefficients [14] is shown in
Table 6.2: AC coefficient coding [12]

<table>
<thead>
<tr>
<th>RRRR</th>
<th>SSSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>EOB</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
</tr>
</tbody>
</table>

The implementation of the entropy coding section using Huffman coding in VHDL was done as shown in figure 6.10.

The Huffman coding component took the quantized DCT coefficient values as input. These inputs were in the zig-zag format. The DC coefficient was saved in the input stage so that the difference magnitude could be determined using DPCM. The number of zeros preceding a non-zero value was also counted in order to determine the RRRR value for AC coefficient coding. The SSSS value was determined by determining the position of the MSB. This RRRRSSSSS value was used and corresponding address for the Huffman look up table was also generated in this stage. There are three look up tables for the Huffman codes. The first table gives the size of the Huffman code. The second table gives the upper 8 bits of the Huffman code and the third table gives the lower 8 bits. The Huffman code obtained from the tables is then shifted and left justified along with the input and finally both the input and the Huffman code for that input are merged together to form a 23 bit output. This merged output is given to the arbiter block and it again saved this merged input along with its 5 bit size and stores the entire 28 bits to the buffers. Finally in the output stage, the buffers give out 8 bits at a time in the output. These 8-bits are written to a file and this file forms the compressed data file.

The Huffman encoding section was also tested. The testbench took the quantized values as the input and also generated the clk, reset, hold_in, done_in and valid_in signals. The
Figure 6.9: Simulation results after quantization

<table>
<thead>
<tr>
<th>Category (SSSS)</th>
<th>AC Coefficients</th>
<th>DIFF Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>-1,1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>-3,-2,2,3</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>-7,-4,4,7</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>-15,-8,-8,15</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>-31,-16,16,31</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>-63,-32,32,63</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>-127,-64,64,127</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>-255,-128,128,255</td>
</tr>
</tbody>
</table>

Table 6.3: Magnitude categories for encoding AC
The quantized input is obtained. DC coefficient is stored to determine difference magnitude using DPCM.

Determine the number of zeros coming in and also the size of the non-zero value for AC coding.

Determine address to get the correct Huffman code from the look up table.

The input value and its corresponding huffman code are merged to form a 23 bit value.

Combine the merged value and the total bit count of the merged value and store this 28 bit value in a buffer.

Output the values from buffer 8-bit at a time.

Figure 6.10: Flow chart for Huffman encoding
simulation output from this stage is shown in figure 6.11. The Huffman encoding is done in five stages. In the input stage, the address for the Huffman look up tables is generated. This address is generated using from the SSSS for DC-components and RRRRSSSS for AC-coefficients. This address is then given to the look up table stage. Three look up tables were used. The first table gave the length of the Huffman code while the other two tables together gave the 16-bit Huffman code value corresponding to the address generated in the input stage. Once the input values and the corresponding Huffman codes were obtained, they were merged together to form a 23 bit value. Here the MSB of the input was left out since it was already coded in the Huffman code. The final output was formed after combining the merged output along with its bit count and was given to the output stage. The output stage gave the data 8-bit at a time.

Figure 6.11: Simulation results from Huffman encoding


6.5 Control Block

The control block was also modelled using VHDL. It takes in CLK, RESET and INIT signals as inputs. The CLK is the system clock. The RESET signal, when high resets the system. The INIT signal was used for initializing the 16 clock signals generated for the SI memory block. Using these inputs the control block generated signals which were used to control the SI memory block in the 2D-DCT, the storage register block, quantizer and entropy encoder.

A VHDL code was written within the control block for a 16-bit ring counter. The INIT signal initialized the counting when asserted high. The RESET signal, when high, resets the system count to zero. The resulting sixteen non overlapping output signals were used as clock signals for the SI memory block used in 2D-DCT.

Once the first set of data was processed in the analog section, it was passed on to the storage unit through the ADC. The LOAD signal became high and was given to the LOAD pin of the first column of the 8-bit registers. This column of registers took in the 5-bit output data from the ADC and gave it as an 8-bit output. When the next set of data was processed, the data stored in the first column was shifted to the next column of registers.

Once all the 64 data bytes were processed and stored in the storage unit, the reset signal from the control unit becomes low and so the digital section took the input from the storage unit and processed it to give the final compressed data. The clock for the quantizer and entropy encoder was the same as the system clock. The control unit also generated the valid_in, done_in and hold_in signals for the quantizer and entropy encoder.

The control block was tested with various sets of inputs. The simulation results for the control block are shown in figure 6.12.

6.6 Simulation of Digital Section

The zig-zag ordering, quantization and Huffman encoding sections were tested individually before combining them with the storage block. Finally all the components were combined
together and the final compressed digital output was obtained. The simulation result for the combined digital section is shown in figure 6.13. The reset signal resets all the blocks. The inputs and outputs are 8-bit data values. The valid_in signal when high implies that the data at the input is valid data. The valid_out signal signifies the same for the output data. The done_in signal becomes high once the last byte of input data is entered whereas the done_out signal goes high once the last output byte is given. The hold_in signal is used to stall the system when no further data can be accepted at the input.
Figure 6.13: Simulation results for digital section.
<table>
<thead>
<tr>
<th>RRRR/SSSS</th>
<th>Code Length</th>
<th>Huffman Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/0</td>
<td>4</td>
<td>1010</td>
</tr>
<tr>
<td>0/1</td>
<td>2</td>
<td>00</td>
</tr>
<tr>
<td>0/2</td>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>0/3</td>
<td>3</td>
<td>100</td>
</tr>
<tr>
<td>0/4</td>
<td>4</td>
<td>1011</td>
</tr>
<tr>
<td>0/5</td>
<td>5</td>
<td>11010</td>
</tr>
<tr>
<td>0/6</td>
<td>7</td>
<td>1111000</td>
</tr>
<tr>
<td>0/7</td>
<td>8</td>
<td>11111000</td>
</tr>
<tr>
<td>0/8</td>
<td>10</td>
<td>1111110110</td>
</tr>
</tbody>
</table>

Table 6.4: Magnitude categories for encoding AC
Chapter 7

System Integration

Once the individual blocks of the experimental image compression system were designed and verified by simulation, the entire circuit was integrated by combining all these blocks. Figure 7.1 shows the entire system. This chapter will explain in detail about the design flow that was followed in order to create the entire circuit. The simulation results after system integration and the complete circuit layout are also discussed.

7.1 Design Flow

The experimental mixed signal image compression system design consists of the DAC, 2D-DCT, differential to single ended current converter and ADC, which form the analog components of the system and the storage unit, quantizer, entropy encoder and control block constituting the digital section. The analog section of this circuit was initially designed and tested using OrCAD Capture for schematic entry and PSpice for simulations. The analog circuit was also done using the Mentor Graphics tools in order to combine it with the digital portions.

The implementation of this mixed signal design utilized the following set of tools from Mentor Graphics

1. Design Architect: Schematic entry

2. Accusim: Circuit simulations
Figure 7.1: Experimental mixed signal image compression circuit
3. IC Station: Layout of analog and digital components
4. Calibre PEX: Extracting post layout netlist with parasites
5. Eldo: Post layout simulations
6. VHDL: Digital design
7. ModelSim: Digital design simulations
8. Leonardo Spectrum: Synthesis of the digital part designed using VHDL

The design flow is explained in the figure 7.2. The analog section was designed using Mentor Graphics schematic entry tool Design Architect. The schematic designs were simulated using Accusim. Once the designs were simulated and the functionality was verified, they were taken to layout. Mentor Graphics IC Station was used for layout. Finally after layout the parasitics were extracted and the post layout simulations were performed in Eldo.

For the digital portions, the designs were modelled using VHDL and the functionality was verified using ModelSim. Then these VHDL designs were synthesized into TSMC 0.35u target technology using the synthesis tool Leonardo Spectrum. After synthesis, the gate level verilog netlist was generated in Leonardo and imported into Design Architect IC. Once imported, these digital sections were combined with the analog section at the schematic level and the same flow as used for the analog sections was followed.

7.2 Simulation and Layout of the Circuit

The mixed signal circuit shown in figure 7.1 was simulated at the schematic level using Mentor Graphics tool Accusim. A reference current of 200μA and a digital input of 01000000 was given to the DAC. This would result in an input current of 100μA to the entire system. The reference current for the ADC blocks was given using a DC current
Design Modelling Using VHDL

Verification by Simulation (ModelSim)

Logic Synthesis for TSMC 0.35u target technology (Leonardo Spectrum)

Gate Level Verilog Netlist

Digital Section

Design by schematic entry (Design Architech IC)

Verification by Simulation (Accusim)

IC Layout (IC Station)

Layout versus Schematic (LVS) Design Rules Check (DRC) Parasitic Extraction (PEX) (Calibre)

Post Layout Simulation (ELDO)

Mixed Signal Section

Figure 7.2: Design flow
source of 1.5mA. An input clock of frequency 1MHz was used. The reset and init signals were set to zero. The output was checked after each stage and verified manually.

The output after the 2D-DCT was verified against the output from Matlab as shown in table 4.9. The ADC was connected at the output of the 2D-DCT The simulation results for this circuit are shown in table 7.1.

<table>
<thead>
<tr>
<th>Input (uA)</th>
<th>Output(uA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I_{out}</td>
<td>5-bit ADC Output</td>
</tr>
<tr>
<td>100</td>
<td>830.846</td>
</tr>
<tr>
<td>100</td>
<td>-7.34</td>
</tr>
<tr>
<td>100</td>
<td>-4.66</td>
</tr>
<tr>
<td>100</td>
<td>-4.73</td>
</tr>
<tr>
<td>100</td>
<td>-4.92</td>
</tr>
<tr>
<td>100</td>
<td>-7.68</td>
</tr>
<tr>
<td>100</td>
<td>-6.02</td>
</tr>
<tr>
<td>100</td>
<td>-4.44</td>
</tr>
</tbody>
</table>

Table 7.1: Simulation results of 2D-DCT and ADC

The complete analog circuit was combined with the digital section. This formed the complete experimental circuit shown in figure 7.1. This entire circuit was simulated and the results obtained were verified after each stage. Layout was done for all the components. Post layout simulations were done after extracting parasites for each individual blocks. These results were compared against the results of simulation at schematic level. The complete circuit layout was done by combining the individual layouts. The complete circuit layout is shown in figure 7.3.
Figure 7.3: Layout for the experimental image compression circuit
Chapter 8

Conclusion and Future Work

This work investigated the implementation of a low power mixed signal image compression system in TSMC 0.35um technology. The major components of this system, two dimensional discrete cosine transform processor, analog to digital converter, quantizer and entropy encoder were simulated and the functionality was verified.

The two dimensional discrete cosine transform section was implemented using switched-current technique. The sampling frequency used was 4MHz. However the results obtained for the 2D-DCT block were accurate even for sampling frequencies as high as 66MHz. The power consumption of the 2D-DCT implemented using switched current technique at a supply voltage of 3.3V was measured to be 2.1mW for a frequency of 66MHz. This was compared with results from previous work showing the digital implementation of a 2D-DCT. The table 8.1 shows a comparison.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Voltage</th>
<th>Frequency</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>[18]</td>
<td>0.6um</td>
<td>2V</td>
<td>100MHz</td>
</tr>
<tr>
<td>[19]</td>
<td>0.18um</td>
<td>1.8V</td>
<td>100MHz</td>
</tr>
<tr>
<td>[20]</td>
<td>1.2um</td>
<td>1.5V</td>
<td>20MHz</td>
</tr>
<tr>
<td>This thesis</td>
<td>0.35um</td>
<td>3.3V</td>
<td>66MHz</td>
</tr>
</tbody>
</table>

Table 8.1: Comparison of power consumption of different 2D-DCT structures

The ADC block designed in this thesis, used multiple switches. If the frequency of the input signal was higher than 4MHz, then the delay caused by switches resulted in an ambiguous outputs. This limited the frequency of operation of the system to 4MHz.
8.1 Future Work

There are some areas of this thesis which can be researched further to obtain better results.

1. The ADC can be implemented using modified switch-current memory cell as the basic cell. This can lead to a better overall frequency of operation for the system and also possibly lower power consumption.

2. The quantization block can be enhanced by adding the ability to be able to use different quantization tables based on the input values. Similarly, the capability to change between different huffman tables can also be beneficial.
Bibliography


