EIGHT BITS PCM SPEECH CODER AND DECODER

Prepared by

Imee Ezainee Bt Ibrahim

This dissertation is presented to

UNIVERSITY SCIENCE OF MALAYSIA

Thesis submitted in fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING (HONS.)
(ELECTRICAL & ELECTRONIC ENGINEERING)

School Of Electrical & Electronic Engineering
University Science Of Malaysia **May 2006**

ABSTRACT (ENGLISH)

The scope of this project is divided into three main parts. Part one involves to study and understand the operation of an eight bit PCM Coder and Decoder that used for speech coding. The second part is to study the architecture of TMS320C6416 DSP Processor. The third part is to implement the PCM Coder and Decoder in real time using the TMS320C6416 DSP Processor. The study of the PCM Coder and Decoder has been carried out by considering a three bit PCM system and its simulation is done by using MATLAB and Simulink. Finally, the eight bit PCM Speech Coder and Decoder was implemented in real time by using the TMS320C6416.

ABSTRAK (BAHASA MELAYU)

Skop projek ini boleh dibahagikan kepada tiga bahagian utama. Bahagian pertama melibatkan pengenalan dan pemahaman tentang operasi bagi Pengekod dan Penyahkod Suara Pemodulatan Kod Denyut (PCM) Lapan Bit yang digunakan dalam pengkodan isyarat suara. Bahagian kedua melibatkan pemahaman terhadap binaan Pemproses Isyarat Digit TMS320C6416. Bahagian ketiga melibatkan pelaksanaan Pengekod dan Penyahkod PCM Lapan Bit dengan menggunakan Pemproses Isyarat Digit TMS320C6416. Pemahaman tentang Pengekod dan Penyahkod PCM ini dilakukan dengan mengambil kira sistem Pemodulatan Kod Denyut (PCM) tiga bit dan simulasinya dilaksanakan dengan menggunakan MATLAB dan Simulink. Kemudian, Pengekod dan Penyahkod PCM Lapan Bit dilaksanakan dengan menggunakan Pemproses Isyarat Digit TMS320C6416.

ACKNOWLEDGEMENTS

First of all I want to express my gratitude to my supervisor, Prof. Farid Ghani for his invaluable guidance throughout the course of this project. He fostered and helped to shape the ideas and concepts presented here.

I would like to thank to Mr. Suardi for his help when I got trouble in this project. This project could not have been possible without their technical expertise, critical insight and enlightening suggestions.

I would like to acknowledge all my fellow friends for their encouragement. Special thanks go to Digital Signal Processing Laboratory technicians, Mr. Amir and who gave me technical and non-technical advise. The friendly and supportive atmosphere that they provided was just as important as their technical help when I need it.

Finally, I thank my parents for their great love, support and many words of encouragement.

With all these great people around me, I hope I would do my very best to finish and succeed in this project. As the last word, I pray for everybody's happiness and success and may Allah bless us in our lives now and onwards.

Thank you.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

List Of Figures

LIST OF ABBREVIATIONS

PCM	-	Pulse Code Modulation
A/D	-	Analog to Digital Converter
D/A	-	Digital to Analog Converter
Vocoder	-	Voice Coder
Codec	-	Coder and Decoder
MOS	-	Mean Opinion Score
DPCM	-	Differential Pulse Code Modulation
DM	-	Delta Modulation
API	-	Application Program Interface
DSK	-	DSP Starter Kit
CCS	-	Code Composer Studio
RTDX^{TM}	-	Real Time Data Exchange
EMIF	-	External Memory Interface
CPLD	-	Complex Programmable Logic Device
ROM	-	Read-Only Memory
HPI	-	Host Port Interface
BLAS	-	Basic Linear Algebra Subroutines
DMA	-	Direct Memory Access
TLC	-	Target Language Compiler

CHAPTER 1

INTRODUCTION

1.1 Background

Digital Signal Processing (DSP) is the study of signals in a digital representation and the processing methods of these signals. DSP and analog signal processing are sub fields of signal processing. DSP has three major sub fields: audio signal processing, digital image processing and speech processing. DSP involves the manipulation of digital signals in order to extract useful information from them. Sampling, quantizing and coding an analog signal, forms a digital signal. A very general scheme to process analog signal digitally is shown in below.

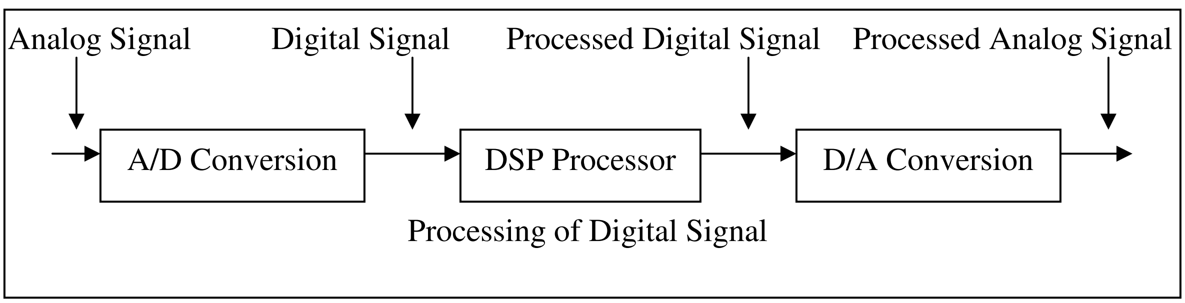


Figure 1.1: Generic DSP Scenario

The blocks marked **A/D** and **D/A** represents devices that convert **A**nalog signal into **D**igital one, and vice versa. These devices allow to translate signals in the physical world (analog) into sequences of numbers (digital) that computers can accept as input and process, and to convert sequences of numbers output by computers back into physical signals. DSP is truly superior to analog signal processing.

There are many reasons why one would want to process an analog signal in a digital fashion by converting it into a digital signal. The greater functionality derives from the possibility of implementing processes that would be extremely difficult and/or expensive to build in analog circuitry. Accuracy and reproducibility are characteristics of digital numbers in contrast to analog voltages. Precision is a function of the number of bits used in computation, and digital numbers can be protected against inaccuracy by error-correcting codes. The modularity and flexibility are byproducts of programmability. DSP code can readily be reused and modified. In recent years significant advances have been achieved in digital technology, including the development of smaller, faster, more power efficient and less expensive digital processors. Once the process can be performed digitally, it usually takes only a very short time until it is profitable to do so [5].

1.2 Speech Coding

There are many techniques for representing analog waveforms by sequences of binary digits or bits. Digital coding is the process or sequence of processes that leads to such digital representations. The benefits of digital representation are many and well known. A basic problem in waveform coding is to achieve the minimum possible distortion for a given encoding rate (bits per second or bits per waveform sample) or equivalently to achieve a given acceptable level of distortion with the least possible encoding rate. The distortion is specified either by an objective measure such as the signal-to-noise ratio or by a subjective measure such as the *mean opinion score* (MOS). Applications of digital representation are many and increasing. They include transmission and/or storage of speech, audio, video and graphics waveforms, satellite imagery and geophysical and medical data.

1.3 The Need for Speech to be Digitized

It is important to know that why speech needs to be digitized. The reasons for requiring speech to be in digital form are as below:

i. The most significant benefit is the fact that digital signal are less sensitive than analog signals to transmission noise. As a result, digitals signals have offered the possibility of making better use of interference and noise-limited communication media, although at the expense of greater bandwidth than that demand in analog communications.
ii. Digitized signals are easy to regenerate and store, to error-protect and encrypt and to multiplex, packetize and mix.

iii. Digitization of waveforms also enables an efficient unification of transmission and switching functions in communications.

iv. It is also permits an extensive application of digital signal processing.

1.4 Method of Coding

Speech coding techniques can be broadly divided into two classes; waveform coding and vocoders (voice coder). Waveform coding aims at reproducing the speech waveform as faithfully as possible. Vocoders preserve only the spectral properties of speech in the encoded signal. The waveform coders are able to produce high-quality speech at high enough bit rates while vocoders produce intelligible speech at much lower bit rates.

The three most common techniques used to encode a voice waveform are as follows:

- Pulse Code Modulation (PCM),
- Differential PCM (DPCM) and
- Delta Modulation (DM).

Except for special purposes, digital telephony uses these techniques [1].

1.5 Scope of the Project

The scope of this project can be divided into four main parts. Part one involves the study of the TMS320C6416 DSP Processor. The second part is study the Pulse Code Modulation (PCM) and its application to speech digitization. The third part is to design an eight bits PCM Coder and Decoder for speech. Initially MATLAB and SIMULINK were used to explain the process of PCM Coder and Decoder. Later, the sampling, quantizing and coding process are implemented in real time using the TMS320C6416 DSP Processor. This project concerned on the understanding of the analog signal, digital signal, process of PCM coder and decoder to the speech signal.

1.6 Objectives of the Project

The objectives of this project can be divided into two mains parts. The first part of this project is to simulate an eight bits PCM coder and decoder using SIMULINK and MATLAB. The quantization process is uniform which implies that it has a fixed quantization step size. The second part of this project is to implement PCM coder and decoder in real time by using the TMS320C6416 DSP Processor.

1.7 Methodology

The project has been carried out under the following main steps:

a) *Study of the TMS320C6416 DSP Processor and its architecture.*

   The TMS320C6416 DSP Starter Kit is available at the DSP Lab of the Electric & Electronic Department. There are three types or versions of the DSP Processor. First, for application that focuses on digital control, the TMS320C2000 platform offers the most controlled optimized DSPs in the world. For application that requires low power consumption the TMS320C5000 platform offers good power efficient performance. For high performance applications, the TMS320C6000 platform of DSPs is considered the highest performance DSPs in the world.

b) *Study of Pulse Code Modulation for Coding Speech Signals.*

   Pulse Code Modulation (PCM) is a common technique used in coding of analog signals because it provides excellent quality for all types of input signals (e.g., voice or data) at a moderate data rate (64 kb/s) and a moderate cost. PCM uses discrete sample times with analogue sample amplitudes to extract the information in a continuously varying analog signal. PCM system is inherently capable of encoding an arbitrarily random waveform whose maximum frequency component does not exceed one-half the sampling rate. Furthermore, the coded speech from an eight bits PCM system with a sampling frequency of 8kHz is generally used as the reference for comparing lower rate speech coders, as its performance is considered to be toll quality.

c) *The Process of PCM Coder and Decoder is modeled in SIMULINK and MATLAB.*

   In this case, three bits PCM Coder and Decoder is easy to model rather then eight bits to show the PCM process in speech coding technique. The sampling frequency for this model is chosen at 200Hz.

d) *Installation and Testing of the DSP Processor*

Installation and Testing of the TMS320C6416 DSP Processor was carried out at the DSP Lab of the Electric &Electronic Department.

e) *Real time realization of PCM Coder and Decoder and reconstruction process on the DSP Processor.*

A real-time process is a task, which needs to be performed within a specified time limit. Real-time implementation is very important from a cost point of view. Any speech coding algorithm can be implemented using available digital signal processor (DSP) chip technology, but the cost of that implementation will increase rapidly with the increase in the number of DSP chip used. The other important consideration in real-time implementation is the power consumption of the final product.

1.8 Relevance of the Study

DSP has three major sub fields: audio signal processing, digital image processing and speech processing. Two unique features of Digital Signal processing as opposed to plain old ordinary digital processing:

- Signals come from the real world - this intimate connection with the real world leads to many unique needs such as the need to react in real time and a need to measure signals and convert them to digital numbers
- Signals are discrete - which means the information in between discrete samples is lost

Pulse code modulation (PCM) is the most widely used voice digitization approach. PCM is a common technique used to encode a voice waveform.

1.9 Organization of the Project

Chapter 1: Introduction of Digital Signal Processing

Chapter 2: Introduction of PCM and the operation involved: *sampling*, *quantizing* and *Coding*

Chapter 3: Introduction of SIMULINK and MATLAB and Development of SIMULINK Model

Chapter 4: Introduction to TMS320C6416 DSP Processor

Chapter 4: Discussing about the Architecture Of The TMS320C6416 DSP Processor

Chapter 5: Discussing about the Real-Time Implementation Of PCM Coder And Decoder Using TMS320C6416 Board

Chapter 6: Conclusions about the project

CHAPTER 2

PULSE CODE MODULATION (PCM)

2.1 Introduction

In a digital waveform coding, a train of pulses is modulated accordance with a digital incoming massage signal. The most common form of digital waveform coding is known as Pulse Code Modulation.

2.2 Pulse Amplitude Modulation (PAM)

The amplitude, duration, and timing of a series of pulses are controlled in PCM, which is relatively simple for digital data already in binary code. Analog signals need to be converted into a recognizable binary code (a pulse-amplitude modulated signal) by regular sampling of its amplitude. If an analog signal is to be transmitted, it must first be converted to a PAM signal by regular sampling of its amplitude. The value of the amplitude is then converted into a binary code for transmission on the carrier wave.

2.3 Pulse Code Modulation (PCM)

PCM is a waveform coding method defined in the ITU-T G. 711 specification. It is literally means modulating a signal by converting it to pulses and then coding these. It is true that most information signals that are to be transmitted are analog. These analog signals need to be converted to the digital format, and the most widely used technique is known as PCM.

PCM involved sampling and quantizing the information signal at regular time intervals, and coding the measured amplitude value into a sequence of pulses. The pulses became the transmitted signal and they conveyed an impression of the message as a series of binary numbers. At the receiver the binary numbers were used to reconstruct the original analog signal in a similar way.

PCM is dependent on three separate operations; *sampling*, *quantizing* and *coding*. PCM is also very demanding system in terms of bit error rate on the digital channel. PCM is the best established, the most implemented and the most applied of all digital coding systems. PCM is widely accepted as a standard against which to calibrate other approaches to waveform digitization.

2.3.1 Sampling

The first step to convert the signal from analog to digital is to filter out the higher frequency component of the signal. Most of the energy of spoken language is between 200 or 300 Hz and about 2700 or 2800 Hz. It means that, voice bandwidth is 200 to 3400 Hz. Roughly 3000 Hz bandwidth for standard speech and standard voice communication is established. Therefore, they do not have to have precise filters. A bandwidth of 4000 Hz is a band-limiting filter is used to prevent aliasing. This happens when the input analog voice signal is under sampled. It is defined by the Nyquist Criterion, $F_s \leq 2(BW)$. The sampling frequency F_s is less than the highest frequency of the input analog signal. This creates an overlap between the frequency spectrum of the samples and the input analog signal. This creation of a false signal when sampling is called aliasing.

The second step to convert an analog voice signal to a digital voice signal is to sample the filtered input signal at a constant sampling frequency. It is accomplished by using a process called pulse amplitude modulation (PAM). This step uses the original analog signal to modulate the amplitude of a pulse train that has a constant amplitude and frequency. The pulse train moves at a constant frequency called the sampling frequency. The Nyquist Criterion state that the sampling frequency is sampled at least twice the highest frequency of the original input analog voice signal.

$$F_s \geq 2(BW) \tag{2.1}$$

F_s = Sampling frequency

BW = Bandwidth of original analog voice signal

After filter and sample (using PAM) an input analog signal, the next step is to digitize these samples in preparing for transmission over a Telephony network. The process of digitizing analog voice signals is called PCM. PCM decodes each analog sample using binary code.

PCM has an analog-to-digital converter on the source side and a digital-to-analog converter on the destination side. PCM used a technique called quantization to encode these samples.

2.3.2 Quantizing

Quantization is the process of converting each analog sample value into a discrete value. As the input signal samples enter the quantization phase, they are assigned to a quantization interval. All quantization intervals are equally spaced (uniform quantization) throughout the dynamic range of the input analog signal. Each quantization interval is assigned a discrete value in the form of a binary code. If an input analog signal is sampled 8000 times per second and each sample is given a code that is 8 bits long, the maximum transmission bit rate using PCM is 64000 bits per second.

Each input sample is assigned a quantization interval that is closest to its amplitude height. If an input sample is not assigned a quantization interval that matches its actual height, an error is introduced into the PCM process. This error is called quantization noise. Quantization noise is equivalent to the random noise that impacts the signal-to-noise ratio (SNR) of a voice signal. SNR is a measure of signal strength relative to background noise. The ration is usually measured in decibels (dB). The higher the SNR, the better the voice quality. Quantization noise reduces the SNR of a signal. Therefore, an increase in quantization noise degrades the quality of a voice signal. For coding purpose, an N bit yields 2^N quantization levels.

One way to reduce quantization noise is to increase the amount of quantization intervals. The difference between the input signal amplitude height and the quantization interval decreases as the quantization intervals are increased (increases in the intervals decrease the quantization noise). However, the amounts of code also need to be increased in proportion to the increase in quantization intervals. This process introduces additional problems that deal with the capacity of a PCM system to handle more code.

2.3.3 Coding

Waveform coding schemes are designed to reproduce the waveform output of the source at the destination with as small as a distortion as possible. Pulse Code Modulation is the simplest and oldest waveform-coding scheme. A pulse code modulator consists of three basics sections; a sampler, a quantizer and an encoder. A functional block diagram of a PCM system is shown below:

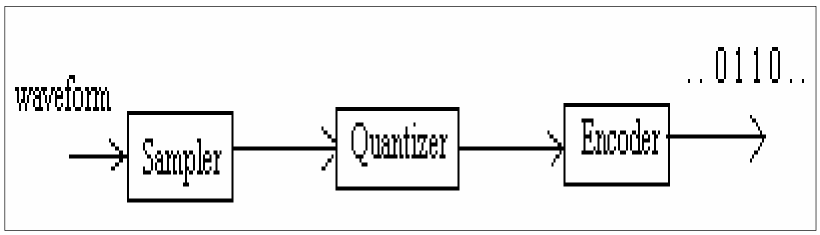


Figure 2.1: Block Diagram of PCM System

2.3.4 PCM Coder and Decoder

The most favored modulation scheme of waveform codecs (coder and decoder) is PCM codecs. The narrow-band speech is typically bandlimited to 4 kHz and sampled at 8 kHz. For coding the speech, it was found that with nonlinear quantization 8 bits per sample was sufficient for speech quality, which is almost indistinguishable from the original. In speech coding an approximation to a logarithmic quantizer often used. Such quantizers give a *signal to noise ratio* (SNR) which is almost constant over a wide range of input levels, and using 8 bits per sample gives a bit rate 64 kbps.

CHAPTER 3

MODELLING OF PCM CODER AND DECODER USING SIMULINK

3.1 Introduction to MATLAB and SIMULINK

MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including graphical user interface building

MATLAB (stands for *matrix laboratory*) is an interactive system whose basic data element is an array that does not require dimensioning. This allows users to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.
[MATLAB 7.0]

3.2 Development of SIMULINK Model

3.2.1 SIMULINK Model

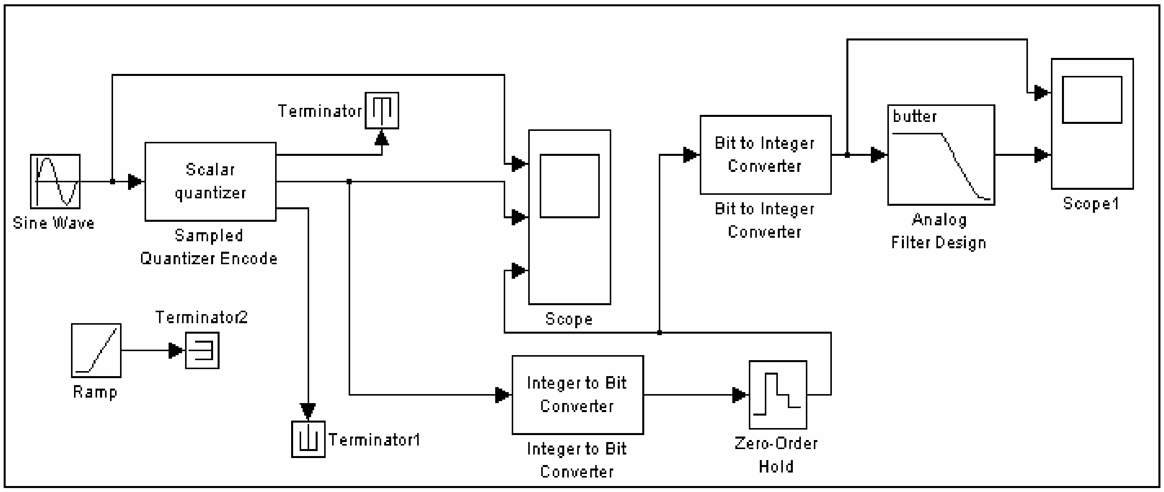


Figure 3.1: Process of PCM Coder and Decoder

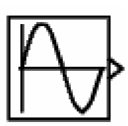
3.3 Block Description

This model is created by using SIMULINK and explains the process of PCM Coder and Decoder. Scopes are used to display the output. For this model, three bit PCM Coder and Decoder is used. It is easy to interpret by using three bit PCM Coder and Decoder rather then eight bit PCM Coder and Decoder from the scope. The quantization levels and the coding are easy to identify by using three bits PCM Coder and Decoder.

3.3.1 Sine Wave

Purpose: Generate a sine wave

Symbol:



Library: Sources

Description: The Sine Wave block provides a sinusoid. The block can operate in either time-based or sample-based mode.

Dialog Box:

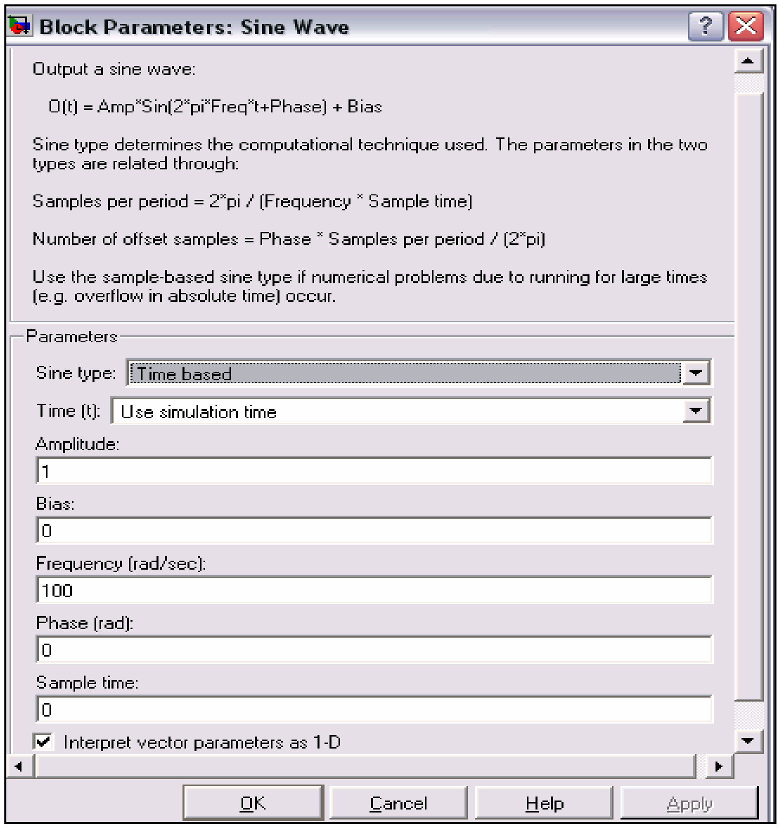


Figure 3.2: Block Parameters Dialog Box for Sine Wave

Sine type

The type of sine wave generated by this block, either time- or sample-based. For this model, time-based is used. Time-based mode has two sub modes: continuous mode or discrete mode. The value of the Sample time parameter determines whether the block operates in continuous mode or discrete mode:

- 0 (the default) causes the block to operate in continuous mode.
- >0 causes the block to operate in discrete mode.

Time

Specifies whether to use simulation time as the source of values for the sine wave's time variable or an external source.

Amplitude

The amplitude of the signal. The default is 1.

Bias

Constant value added to the sine to produce the output of this block.

Frequency

The frequency is in radians/second. For this block the frequency is 100 radians/second.

Phase

The phase shift is in radians. The default is 0 radians.

Sample time

This is the sample period. The default is 0.

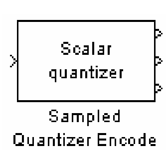
Interpret vector parameters as 1-D

If selected, column or row matrix values for the Sine Wave block's numeric parameters result in a vector output signal; otherwise, the block outputs a signal of the same dimensionality as the parameters. If this option is not selected, the block always outputs a signal of the same dimensionality as the block's numeric parameters.

3.3.2 Sampled Quantizer Encode

Purpose: Quantize an analog signal to digital signal, quantization value and distortion at every sample time point

Symbol:



Description: The input can be a scalar, a sample-based vector or a frame-based row vector. This block processes each vector element independently.

Dialog Box:

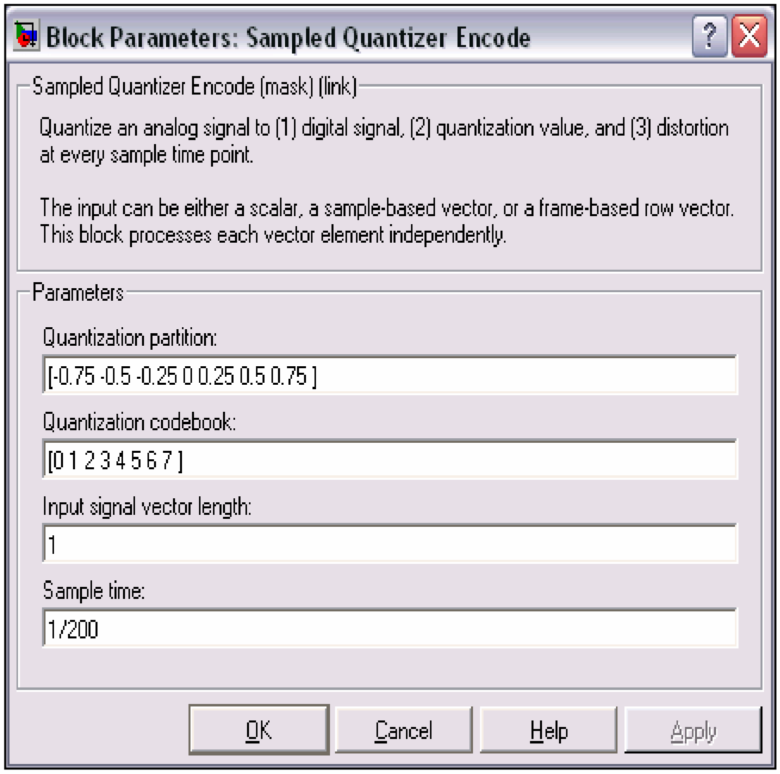


Figure 3.3: Block Parameters Dialog Box for Sampled Quantizer Encode

3.3.3 Integer to Bit Converter

Purpose: Map a vector of integers to a vector of bits

Symbol:

Library: Utility Blocks

Description:

The Integer to Bit Converter block maps each integer in the input vector to a group of bits in the output vector.

Dialog Box:

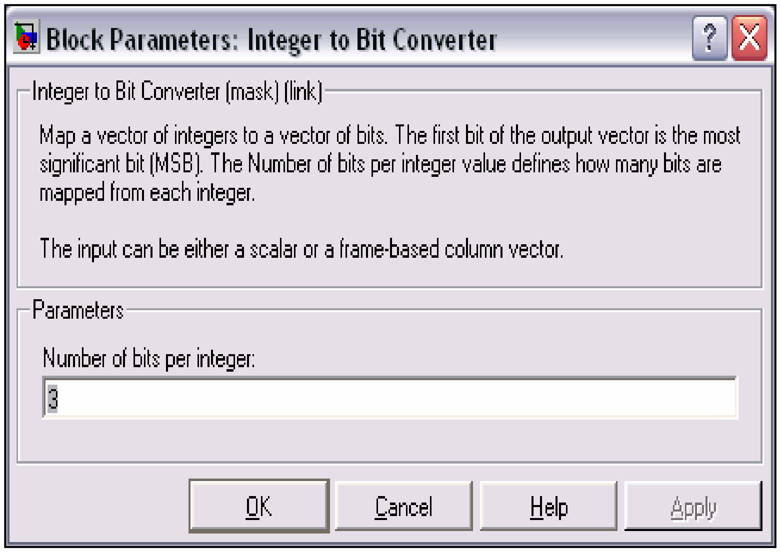


Figure 3.4: Block Parameters Dialog Box for Integer to Bit Converter

Number of bits per integer

The number of bits the block uses to represent each integer of the input. This parameter must be an integer between 1 and 31. For this model, 3 bits per integer are used.

3.3.4 Bit to Integer Converter

Purpose: Map a vector of bits to a corresponding vector of integers

Symbol:

Library: Utility Blocks

Description:

The Bit to Integer Converter block maps groups of bits in the input vector to integers in the output vector.

Dialog Box:

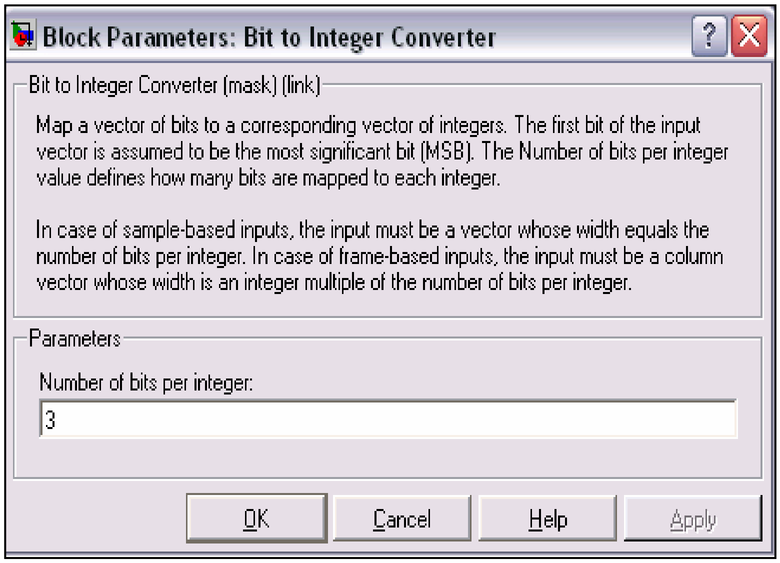


Figure 3.5: Block Parameters Dialog Box for Bit to Integer Converter

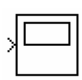
Number of bits per integer

The number of input bits that the block maps to each integer of the output. This parameter must be an integer between 1 and 31. For this model, 3 bits per integer are used.

3.3.5 Scope

Purpose: Display signals generated during a simulation

Symbol:



Library: Sinks

Description:

The Scope block displays its input with respect to simulation time. The Scope block can have multiple axes (one per port); all axes have a common time range with independent y-axes.

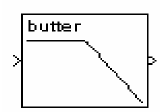
3.3.6 Analog Filter Design

a) Anti aliasing Filter

b) Reconstruction Filter

Purpose: Design and implement an analog filter

Symbol:



Library: Filtering / Filter Designs

Description:

The Analog Filter Design block designs and implements a Butterworth, Chebyshev type I, Chebyshev type II, or elliptic filter in a highpass, lowpass, bandpass, or bandstop configuration. The input must be a sample-based scalar signal. The design and band configuration of the filter are selected from the **Design method** and **Filter type** pop-up menus in the dialog box. [MATLAB 7.0]

i. **Anti aliasing Filter**

The anti aliasing filter should guarantee that no frequencies over Nyquist may pass. Assume that the highest frequency of importance in the signal to be sampled is $f_{\max}$. The sampling theorem allows us to sample at any frequency above the Nyquist frequency $f_N = 2f_{\max}$, but in practice we can only sample this way if there is absolutely nothing above $f_{\max}$. Thus in order to utilize realizable filter, we must sample at a frequency $2f_{\max}$, higher than the sampling theorem strictly requires.

ii. **Reconstruction Filter**

The reconstruction filter needs to smooth out the D/A output, which is properly defined only at the sampling instants, and recreate the proper behavior all the times. The reconstruction process begins by taking a sampled signal, which will be in discrete time, and performing a few operations in order to convert them into continuous-time (exact copy of the original signal).

When considering the reconstruction of a signal, it is important to be familiar with the idea of the Nyquist rate. If we sample at below the Nyquist rate, then the problem is known as *aliasing* will arise that will make perfect reconstruction impossible.

Dialog Box:

The dialog box for Anti aliasing filter and Reconstruction filter is the same for both.

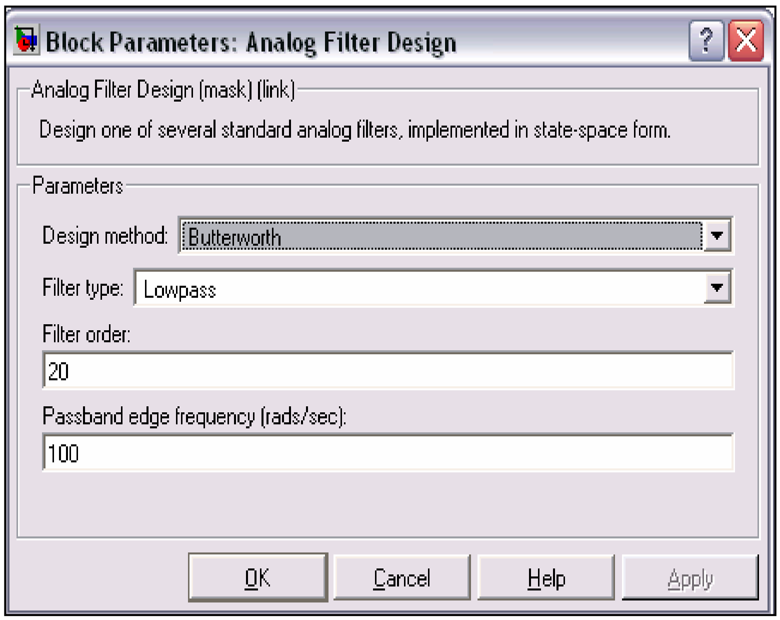


Figure 3.6: Block Parameters Dialog Box for Analog Filter Design method

The filter design method

`Butterworth`, `Chebyshev type I`, `Chebyshev type II`, or `Elliptic`. For this model, the filter design method chosen is `Butterworth`.

Filter type

The type of filter to design: `Lowpass`, `Highpass`, `Bandpass`, or `Bandstop`. This model uses a `lowpass` filter.

Filter order

The order of the filter, for lowpass and highpass configurations. For bandpass and bandstop configurations, the order of the final filter is twice this value.

Passband edge frequency

The passband edge frequency, in rad/s, for the highpass and lowpass configurations of the Butterworth, Chebyshev type I, and elliptic designs. The value for this model is 100Hz. This value is multiplied by 2π to transform into rad/s.

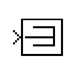
Sampling frequency, $f_s = 200Hz$

So,

Sampling frequency in rad/s $\omega_s = 2\pi f_s = (2*3.142*200)$ rad/s

3.3.7 Terminator

Purpose: Terminate an unconnected output port

Symbol: 

Library: Sinks

Description:

The Terminator block can be used to cap blocks whose output ports are not connected to other blocks. If you run a simulation with blocks having unconnected output ports, Simulink issues warning messages. Using Terminator blocks to cap those blocks avoids warning messages.

Dialog Box:

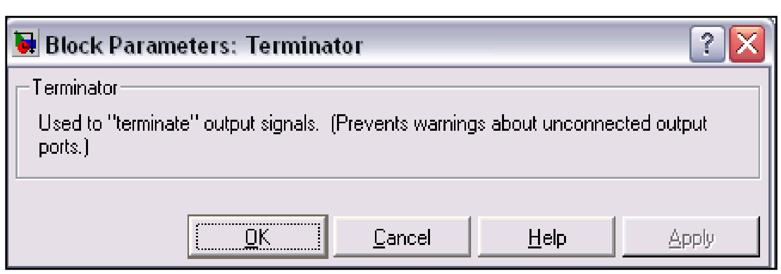


Figure 3.7: Block Parameters Dialog Box for Terminator

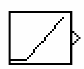
Data Type Support

The Terminator block accepts real or complex signals of any data type supported by Simulink, including fixed-point data types

3.3.8 Ramp

Purpose: Generate constantly increasing or decreasing signal

Symbol:



Library: Sources

Description: The Ramp block generates a signal that starts at a specified time and value and changes by a specified rate. The block's Slope, Start time, Duty Cycle, and Initial output parameters determine the characteristics of the output signal. All must have the same dimensions after scalar expansion.

Dialog Box:

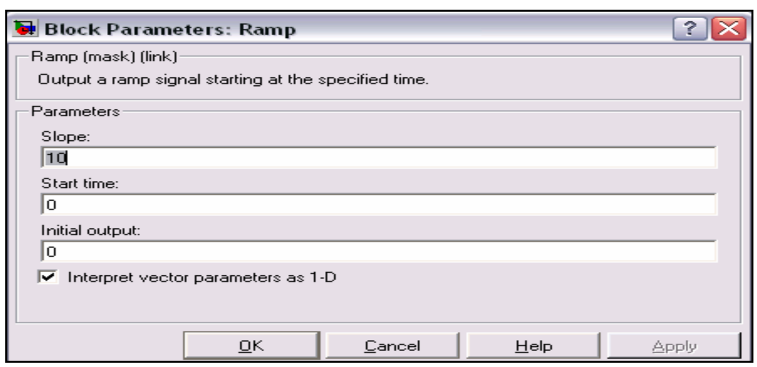


Figure 3.8: Block Parameters Dialog Box for Ramp

Slope

The rate of change of the generated signal. The default is 1.

Start time

The time at which the signal begins to be generated. The default is 0.

Initial output

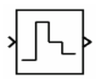
The initial value of the signal. The default is 0.

Interpret vector parameters as 1-D

If you select this option and the other parameters are one-row or one-column matrices, after scalar expansion, the block outputs a 1-D signal (vector). Otherwise, the output dimensionality is the same as that of the other parameters.

3.2.9 Zero-Order Hold

Purpose: Implement a zero-order hold of one sample period

Symbol: 

Library: Discrete

Description:

The Zero-Order Hold block samples and holds its input for the specified sample period. The block accepts one input and generates one output, both of which can be scalar or vector. If the input is a vector, all elements of the vector are held for the same sample period.

Dialog Box:

Figure 3.9: Block Parameters Dialog Box for Zero-Order Hold

Sample time (-1 for inherited)

Specify the time interval between samples. To inherit the sample time, set this parameter to. For this model, the sample time is 1/200.

3.4 Results

The performance of the three bit PCM Coder and Decoder can be viewed by the signal waveforms in scope (**Figure 3.10)** and scope 1 (**Figure 3.11**).

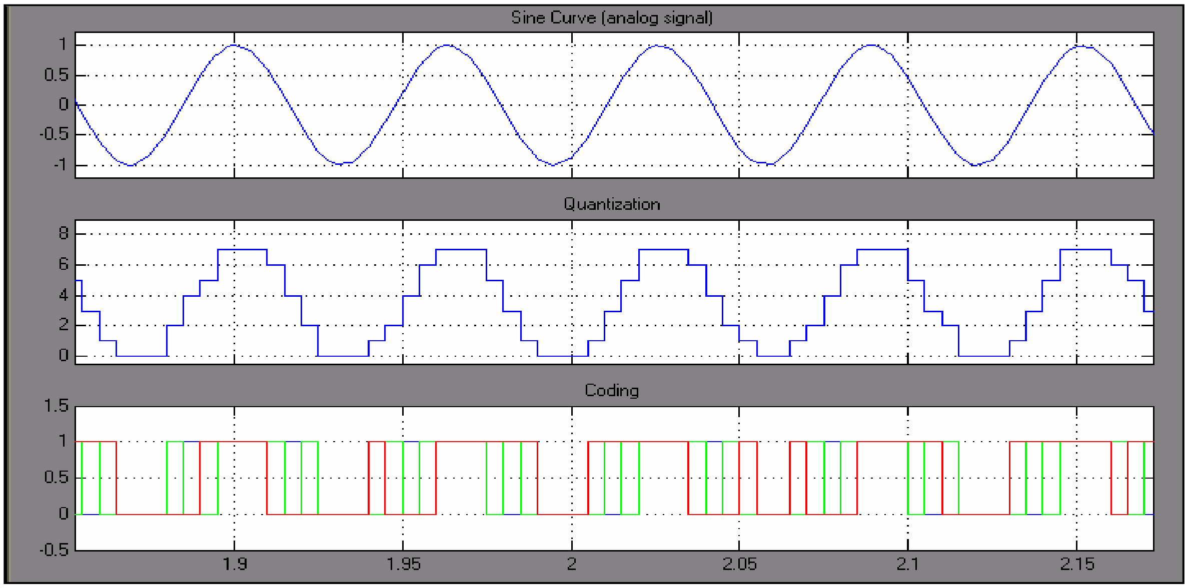


Figure 3.10: Output from Scope
Waveforms explains the Process of PCM Coder and Decoder (Scope)

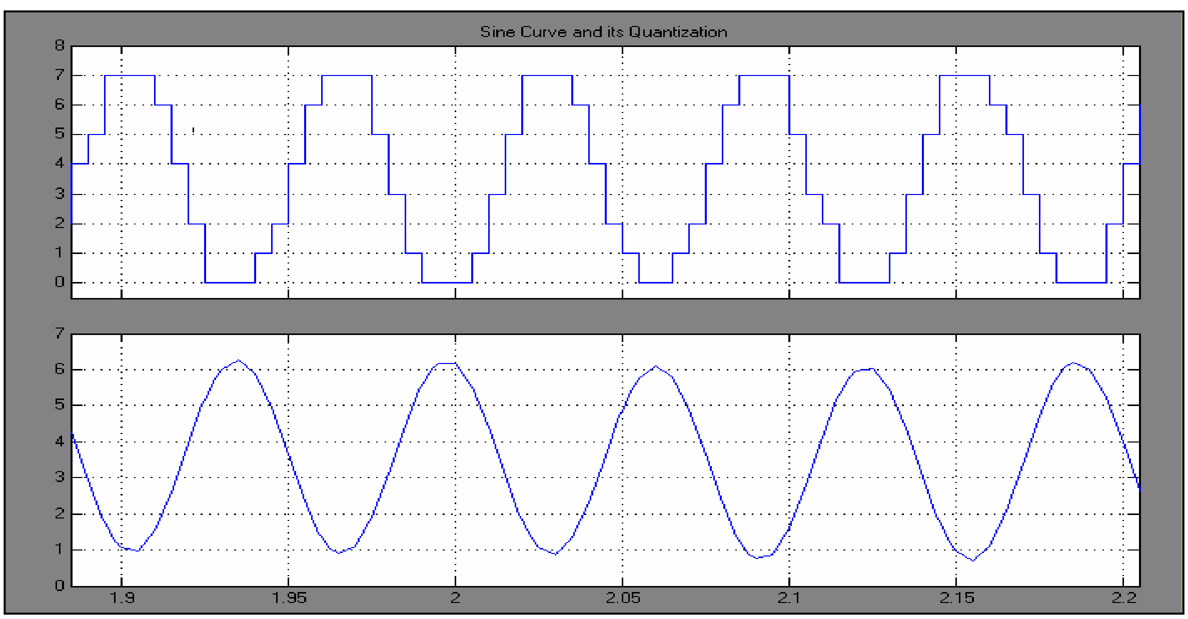


Figure 3.11: Output from Scope 1
Waveforms explains the Process of PCM Coder and Decoder

3.5 Discussion

This model is in SIMULINK and explained the process of PCM Coder and Decoder. The input signal generated by the 'Sine Wave' block. The Sampled Quantizer Encode quantized the analog signal to digital signal. After passed by the Sampled Quantizer Encode, the original signal would be quantized to digital signal as shown in **Figure 3.10** . The quantized signal passed by the Integer to Bit Converter. The Integer to Bit Converter block maps each integer in the input vector to a group of bits in the output vector. The Zero-Order Hold block samples and holds its input for the specified sample period. The block accepts one input and generates one output. After that, it would pass by the Bit to Integer Converter. The Bit to Integer Converter block maps groups of bits in the input vector to integers in the output vector. This signal would pass through the analog filter design. For this model, the type of filter to design is Lowpass Filter.

Pulse Code Modulation (PCM) is a common technique used in coding of analog signals because it provides excellent quality for all types of input signals (e.g., voice or data) at a moderate data rate (64 kb/s) and a moderate cost.