A cable/wireless modulation/demodulation frontend module for high speed/broadband datalink

Pay Peng Clarice Chong

*Edith Cowan University*

Follow this and additional works at: [https://ro.ecu.edu.au/theses_hons](https://ro.ecu.edu.au/theses_hons)

Part of the *Signal Processing Commons*

**Recommended Citation**

Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

• Copyright owners are entitled to take legal action against persons who infringe their copyright.

• A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author’s moral rights contained in Part IX of the Copyright Act 1968 (Cth).

• Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.
A CABLE/WIRELESS
MODULATION/DEMODULATION FRONTEEND
MODULE FOR HIGH SPEED/BROADBAND
DATALINK

Submitted
By

Chong Pay Peng Clarice

In partial fulfilment of the requirements for the
Bachelor of Engineering (Communication Systems) with honours
School of Engineering and Mathematics
February 9th, 2001
DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

i incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

ii contain any material previously published or written by another person except where due reference is made in the text; or

iii contain any defamatory material.

Signature: ____________________________

Date: 21/04/2001
ABSTRACT

What are the modulation/demodulation schemes that are suitable for cable or wireless high speed or broadband data link, between 1 Mbps to 30 Mbps? The recent adoption of modulation schemes such as OFDM, 8-VSB, QAM, for Digital Television broadcast (DTV), Asymmetrical Digital Subscriber Lines (ADSL), and Cable Modem (CM) for Internet access, is possible because of the increase bit/Hz in these modulations over conventional modulation schemes. These modulation/demodulation schemes could also offer similar advantages to dedicated data link applications such as home-networking using cable or wireless medium.

This report document the project implementation of Quadrature Amplitude Modulation as the modulation / demodulation scheme for a Cable/Wireless Modulation/Demodulation Frontend Module For High Speed/Broadband Data Link. A hardware prototype was constructed with a 10Base-T Module interface to the QAM modulator/demodulator module through a QAM-16 symbol mapper/decoder.
ACKNOWLEDGEMENTS

It is with gratitude that the author would like to thank Mr Huang HK whose most generous response to the author's request has provided invaluable technical help, guidance and encouragement in the course of the project, and thanks to whom the quality of the project was decidedly enhanced.
REPORT ORGANISATION

This thesis is divided into 3 parts, with each corresponds to the respective stages of the project.

Part 1 is the progress report generated as a joint-effort with Mr Loh Kee Kiang, Vincent as fulfilment of the requirements of Project 1 ENS4141. It gives the literature review of the subject whereby different approaches to the main objectives of the project are compared. This part of the thesis runs from chapter 1 to 8 and has 2 appendices, A and B.

Part 2 is the final year project report presented as fulfilment of the requirements of Project 2 (ENS4241). It constitutes the main focus of the whole project with details on implementation and verification of the ideas and concepts. This part of the thesis is divided into 6 chapters (chapter 9 to 14) and has 4 appendices (C to F).

Part 3 is an annex to Part 2 where it includes the work done on the project as part of the future improvement and development as mentioned in the Part 2 of this thesis and in preparation for the seminar presentation of the project. It consists of 5 sub parts.
# PART 1 PROGRESS REPORT

1. **INTRODUCTION** .......................................................... 1
   1.1 Objectives ................................................................. 2
2. **AN OVERVIEW OF MOD/DEMOD SCHEMES** .............. 4
   2.1 Orthogonal Frequency Division Multiplexing ................. 6
   2.2 8-VSB ................................................................. 6
   2.3 QAM ................................................................. 7
3. **THE PRINCIPLES OF OFDM** .................................. 10
   3.1 The general problem: Data transmission over multipath .... 10
   3.2 Single carrier approach ............................................. 11
   3.3 Multi carrier approach ............................................. 13
   3.4 Orthogonal Frequency Division Multiplexing (OFDM) .... 14
4. **PRINCIPLES OF 8-VSB** ........................................ 21
   4.1 System Overview .................................................... 21
   4.2 Nyquist Filter ........................................................ 25
   4.3 8-VSB Eye Diagram ............................................... 27
   4.4 8-VSB Signal Constellation ..................................... 28
   4.5 Advantages of 8-VSB ............................................. 29
5. **The PRINCIPLES OF QAM** .................................... 30
   5.1 QAM Generation and Detection ................................ 34
6. **EYE DIAGRAMS** .................................................. 36
   6.1 Diagnosis using the Eye Diagrams ............................. 37
   6.2 Example of a Complex Eye Diagram ........................... 38
   6.3 QAM Constellation Analysis ................................... 39
7. **FUTURE DEVELOPMENT WORK** ............................... 42
8. **RECOMMENDATION** ............................................. 43
9. **REFERENCES FOR PART 1** .................................... 44

# PART 2 PROJECT REPORT

9. **INTRODUCTION** ...................................................... 45
   9.1 Motivation ........................................................... 46
   9.2 Project Objectives .................................................. 47
   9.3 Project Report Overview ......................................... 48
10. **TARGET APPLICATION** ......................................... 49
   10.1 Modulation Schemes ............................................. 49
   10.2 Why QAM .......................................................... 50
   10.3 System Description ............................................... 52
   10.4 Hardware Architecture/Platform Selection ................. 54
LIST OF FIGURES

Figure 2.1 I-Q diagram or constellation for 16-QAM scheme ........................................... 9
Figure 3.1 Multi-Transmission in a broadcasting application ............................................. 10
Figure 3.2 Effective length of cir .................................................................................. 11
Figure 3.3 Basic structure of a single carrier system ....................................................... 12
Figure 3.4 Basic structure of a multi-carrier system ....................................................... 13
Figure 3.5 OFDM and the orthogonal principal .............................................................. 14
Figure 3.6 Setup of a basic OFDM transmitter and receiver .......................................... 17
Figure 3.7 Section of an OFDM signal showing 5 symbols, using a guard period, which is half a cyclic extension of the symbol, and half a zero amplitude signal (for a signal using a 2048 point FFT and 512-sample total guard period). ................................................................. 19
Figure 4.1 VSB transmitter ......................................................................................... 22
Figure 4.2 VSB data frame ......................................................................................... 23
Figure 4.3 VSB channel occupancy (nominal) ................................................................. 24
Figure 4.4 8-VSB RF Frequency Spectrum .................................................................... 25
Figure 4.5 Impossible 8-VSB RF Envelope ..................................................................... 26
Figure 4.6 The 8-VSB Eye Diagram ............................................................................... 27
Figure 4.7 8-VSB signal constellation vs QAM .............................................................. 29
Figure 5.1 QAM modulator ......................................................................................... 31
Figure 5.2 QAM Demodulators, Correlator Type (top), Low-pass Filter Type (bottom) ................................................................................................................................. 32
Figure 5.3 16-QAM generation and detection ................................................................. 35
Figure 6.1 Use of Eye Diagrams for System Characterization ........................................ 37
Figure 6.2 Diagnosis using the eye diagram .................................................................. 38
Figure 6.3 Example of Complex Eye Diagram ............................................................... 39
Figure 6.4 Constellation analysis for a good signal ....................................................... 40
Figure 6.5 Constellation analysis for a poor signal ....................................................... 40
Figure 6.6 Constellation analysis for a lost lock signal ................................................ 41
Figure 10.1 I-Q diagram or constellation for 16-QAM scheme ..................................... 51
Figure 10.2 Typical Target Application .......................................................................... 52
Figure 11.1 Schematic for 10Base-T Interface Module ................................................... 58
Figure 11.2 Block diagram of QAM Symbol Mapper ..................................................... 59
Figure 11.3 Block diagram for QAM modulator/demodulator Module ....................... 64
Figure 11.4 Schematic for QAM modulator/demodulator Module ................................ 68
Figure 11.5 Block diagram of QAM Symbol Decoder ................................................... 69
Figure 11.6 Complete Schematic of project .................................................................. 72
Figure 12.1 Measured transmitter timings ..................................................................... 74
Figure 12.2 bxDac IQ signal output ............................................................................... 76
Figure 12.3 Rx ADC and symbol decoding timings ....................................................... 79
Figure 12.4 Measured Receiver Parallel to Serial Conversion Timings ....................... 80
List of Tables

Table 1: Manchester Coded Data .................................................. 62
PART 1 PROGRESS REPORT
1. INTRODUCTION

Today's network devices, computing peripherals, internet appliance, home entertainment electronics comes with a varied form of physical interface (USB, 10Base-T, 10Base-2 etc.) and supported media (UTPs, coaxial, wireless etc). It will be interesting to explore the concept of allowing a number of network appliances with different network interface to be hooked on to a single shared cable/wireless medium through a class of 'Medium-Transceiver-Interface (MTI*)'. This would offer the user, say in a home environment, easy access to the resultant networked resources and make inter-operation between various devices/appliance/appliance possible (via a protocol translator/server).

The heart of such a class of MTI (other than data control and medium/channel arbitration unit) is the modulation/demodulator module, which has to provide a reliable, bandwidth efficient, digital/data modulation and demodulation for transmission and reception of data. The objective of this project is firstly, a detail study of the digital/data modulation/demodulation suitable for such a class of MTI, evaluating its technical feasibility and constraints. Modulation/demodulation schemes in consideration include OFDM, 8-VSB and QAM. Secondly, implementation of a suitable modulation and demodulation scheme for the MTI.

* This project uses, with permission, in some parts, intellectual property from unpublished works of Huang HK. Permission is granted for use only for the purpose of this project.
1.1 Objectives

The objective of the report is to give an update on the progress attained so far mainly in the research work for the project.

Firstly, an overview of the three different modulation/demodulation schemes in consideration, namely, OFDM, 8-VSB and QAM is briefly described and defined. This is then followed by a discussion on the principles of each scheme, their typical generation method and some of their advantages and disadvantages. The report also includes a short introduction to the eye diagram and constellation analysis as powerful visual tools for observing and diagnosing problems within the modulation/demodulation portion of a digital communication link.

In OFDM, the required amplitude and phase of a carrier can be calculated based on a QAM modulation scheme, while in 8-VSB, the data is transmitted exclusively in the amplitude of the RF envelope and not in the phase. A comparison 8-VSB to a digital modulation format such as QAM will give insight into the modulation scheme. To gain a good understanding of QAM, which is crucial for this project, a simulation of a typical 16-QAM using Matlab is carried out.

This report concludes by giving a recommendation for the future development of the project, which is to be completed by next semester in unit ENS4241 Engineering Project 2.
2. AN OVERVIEW OF MOD/DEMOD SCHEMES

There are many types of modulation schemes available today. Analog modulation processes perform their magic by changing one or more of the three characteristics of a sine wave: amplitude, frequency, and phase.

Digital modulation applies a digital data stream to the carrier and makes the data stream compatible with the RF communications channel. Each wave state generated in this way represents one symbol of data (each symbol is an N-bit word where N is a power of two from 1 to 8, depending on the technology used). The resulting modulations schemes are called amplitude shift keying, frequency shift keying, and phase shift keying. In RF communications however, the two main approaches are phase shift (constant amplitude) and amplitude shift. The number of symbols per second transmitted is known as the baud rate. The number of bits per second equals (symbols per second) multiplied by (bits per symbol).

When it comes to modulations, the lack of standards becomes quite evident. A hodgepodge of modulation techniques with a range of price/performance features are in use today, although the cable modem industry seems to be settling on a de-facto turn to a 64-QAM (N=6) or 256-QAM (N=8) delivery model for downstream data, and QPSK for a moderate bit-rate return path.

All modulation schemes can be judged by their spectral efficiency and by their error rates. Spectral efficiency is the input digital rate divided by the allocated RF channel bandwidth. The unit of measure is "bits/Hz." The error rate (usually
failed bits per million or bits per billion of "good" bits) is a function of several factors, including susceptibility to noise and interference, susceptibility to fading, and non-linearities, which can arise due to dependencies on signal frequency and amplitude. In general, as spectral efficiency increases, so unfortunately does the error rate, which means a higher signal-to-noise ratio might be needed to achieve acceptable error rates.

There are several ways to achieve more than one bit/Hz of throughput. Instead of simple binary encoding, the system can define four different voltages or phases for a single wave cycle, allowing one cycle to represent a two-bit symbol. If both phase and amplitude can vary simultaneously over four values, then one cycle can represent one of 16 discrete logical states. This squeezes 4 bits of data into a single wave cycle, or 4 bits/Hz. Much of the work on modulation techniques currently benefiting the cable modem industry stems from interest in digital video and from technology developed for telephony. While MPEG-2 is becoming the digital video standard for the broadcast industry, digital modulation techniques allow vendors to squeeze 6 or more digital channels into the 6 MHz space normally used for a single analog channel. This is one of digital video's major benefits (and is the basis for much of the talk about 500 cable channels in the future).

The set of available transmission symbols in a particular modulation scheme is known as its alphabet while a graph of the alphabet on a complex plane is known as the constellation. The constellation diagram is drawn by plotting the real part, I, and the complex (or imaginary) part, Q, on a 2-D map after symbols have been formed and converted to complex numbers.
2.1 Orthogonal Frequency Division Multiplexing

The orthogonal frequency division multiplexing (OFDM)\(^2\) is a form of multi-carrier modulation technique that was first introduced more than three decades ago. The advances in digital signal processing (DSP) and very large scale integrated circuit (VLSI) technologies have paved the way for the massive implementation of OFDM techniques in the consumer electronics field. One successful implementation of OFDM is in digital audio broadcasting (DAB), which was developed in Europe for terrestrial and satellite broadcasting of multiple digital audio programs to mobile receivers. Another implementation is in asymmetric digital subscriber line (ADSL) technology that has been selected by ANSI for transmission of digitally compressed video signals over telephone lines.

2.2 8-VSB

8 VSB: Vestigial sideband modulation with 8 discrete amplitude levels. 8-VSB \(^3\) is the RF modulation format utilized by the DTV (ATSC) digital television standard to transmit digital bits over the airwaves to the home consumer. Since any terrestrial TV system must overcome numerous channel impairments such as ghosts, noise bursts, signal fades, and interference in order to reach the home viewer, the selection of the right RF modulation format is critical. The 8-VSB format is the cornerstone upon which the DTV standard is based; developing a basic understanding of 8-VSB is imperative for those who will be working around DTV in the future.
The ATSC 8-VSB system uses a layered digital system architecture consisting of:

- picture layer that supports a number of different video formats;
- compression layer that transforms the raw video and audio samples into a coded bit stream;
- transport layer that "packetizes" data; and
- radio frequency (RF) transmission layer.

The ATSC 8-VSB transmission system is a single carrier frequency technology that employs vestigial sideband (VSB) modulation similar to that used by conventional analog television. The transmission layer modulates a serial bit stream into a signal that can be transmitted over a 6 MHz television channel. The ATSC 8-VSB system transmits data in a method that uses trellis-coding with 8 discrete levels of signal amplitude. Receivers provide a pilot tone to facilitate rapid acquisition of the signal. Complex coding techniques and adaptive equalization are used to make reception more robust to propagation impairments such as multipath, noise, and interference. The 6 MHz ATSC 8-VSB system transmits data at a rate of 19.4 Mbps. More detail will be discussed at the later section of the report.

2.3 QAM

In many applications, bandwidth is an important commodity. A widely used bandwidth-efficient modulation technique called quadrature amplitude modulation (QAM) allows two message signals to be transmitted in the same frequency band without mutual interference.
QAM systems [6] combine PSK and ASK to increase the number of states per symbol. QAM is a proven technique for the transmission of digital data over a wide range of channels from voice band modems at 9600 bps to microwave links transmitting hundreds of Mbps.

QAM is also the modulation technique used by V.34 modems. Each symbol value represents multiple bits. 16-QAM carries 4 bits per symbol while 256-QAM carries 8 bits per symbol. The signal-to-noise ratio at the receiver determines the QAM level that can be used reliably on a given transmission channel. Typical terrestrial and cable channels allow 16-QAM and 256-QAM, leading to digital data rates of approximately 20 and 40 Mbps, respectively.

In a typical cable TV application, 64-QAM can squeeze a 30 Mbps data stream into a 6 MHz (bandwidth) TV channel. QAM is also of use for digital video broadcast. Note that for video-on-demand or MPEG-based broadcast video delivery, 64-QAM allows five channels of 6 Mbps video for each analog channel allocation. For telephony, which uses 64 kbps data streams, a single "video channel" could handle over 450 downstream phone calls, which would be time-division multiplexed within the datastream. Hence, in 750 MHz cable systems, the upper 240 MHz can contain up to 1000 3-Mbps datastreams, each carrying a unique digital address that directs it to a particular set-top box or cable modem (used for video on demand, or VOD). QAM is used in some upstream traffic designs, but is less noise resistant, though more bit-efficient, than QPSK.

It is easier to visualize QAM by looking at 16-QAM. QAM separates points widely and is hence fairly noise immune. The system for 16-QAM combines 4 input bits to produce 1 signal burst. Both phase and amplitude are modulated. Odd-numbered bits in the input stream are combined in pairs to form one of 4 levels, which modulate the sine term.
Even-numbered bits are similarly combined to modify the cosine term. Sine and cosine terms are then combined.

\[ V(t) = x(t) \cos \frac{1}{2}t + y(t) \sin \frac{1}{2}t \]

![Diagram of I-Q constellation for 16-QAM scheme](image)

*Figure 2.1 I-Q diagram or constellation for 16-QAM scheme*

16-QAM has better spectral efficiency than 8-PSK and is less sensitive to noise than 16-PSK because the spacings between symbols are larger (see diagrams below). This is true because the symbols are not all on the same circle; the resultant signals are not all of the same amplitude.
3. THE PRINCIPLES OF OFDM [7]

3.1 The general problem: Data transmission over multipath

Differently from satellite communication where we have one single direct path transmitter to receiver in the classical terrestrial broadcasting scenario we have to deal with a multi-path – channel: The transmitted signal arrives at the receiver in various paths (See figure 3.1) of different length. Since multiple versions of the signal interface of the signal interfere with each other (Inter Symbol Interference (ISI)) it becomes very difficult to extract the original information.

Figure 3.1 Multi-Transmission in a broadcasting application.
The common representation of the multi-path channel is the channel impulse response (cir) of the channel, which is the signal at the receiver if a single pulse, is transmitted (Figure 3.2).

![Figure 3.2 Effective length of cir.](image)

Let's assume a system transmitting discrete information in time intervals, $T$. The critical measure concerning the multipath-channel is the delay $\tau_{\text{max}}/T$ previous symbols. This influence has to be estimated and compensated for in the receiver, a task that may become very challenging.

### 3.2 Single carrier approach

In figure 3.3, the general structure of a single carrier transmission system is depicted. The transmitted symbols are pulse formed by a transmitter filter. After passing the multipath channel in the receiver a filter matched to the channel is used to maximize signal to noise ratio of the device used to extract the data.
The scenario we are dealing with in DVB-T is characterized by the following conditions:

- Transmission Rate: \( R = \frac{1}{T} = 7.4 \text{Msym/s} \)
- Maximum channel delay: \( \tau \text{max} = 224\mu\text{s} \)

For the single carrier system, this results in an ISI of:

\[ \tau \text{max}/T \approx 1600 \]

The complexity involved in removing this interference in the receiver is tremendous. In the scenario under consideration here, using such an approach will only lead to sub-optimal results. This is the main reason why the multi carrier approach presented in the next section has become so popular.
3.3 Multi carrier approach

Figure 3.4 shows the general structure of a multi carrier system.

![Diagram of a multi-carrier system](image.png)

The original data stream of rate $R$ is multiplexed into $N$ parallel data streams of rate

$$R_{mc} = 1/T_{mc} = R/N$$

Each of the data streams is modulated with a different frequency and the resulting signals are transmitted together in the same band. Correspondingly the receiver consists of $N$ parallel receiver paths. Due to the prolonged distance in between transmitted symbols the ISI for each sub system reduces to

$$\frac{T_{\text{max}}}{T_{mc}} = \frac{T_{\text{max}}}{N \cdot T}$$

In the case of DVB-T we have $N=8192$ leading to an ISI of

$$\frac{T_{\text{max}}}{T_{mc}} = 0.2$$
Such little ISI can often be tolerated and no extra counter measure such as an equalizer is needed. Alas as far as the complexity of a receiver is concerned a system with 8192 parallel paths still isn't feasible. This asks for a slight modification of the approach that leads us to the concept of OFDM.

3.4 Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a multi carrier transmission technique, which divides the available spectrum into many carriers, each one being modulated by a low rate data stream. In OFDM the subcarrier pulse used for transmission is chosen to be rectangular. This has the advantage that the task of pulse forming and modulation can be performed by a simple Inverse Discrete Fourier Transform (IDFT) which can be implemented very efficiently as a Fast Fourier Transform (FFT). Accordingly in the receiver we only need a FFT to reverse this operation. According to the theorems of the Fourier Transform the rectangular pulse shape will lead to a $sin(x)/x$ type of spectrum of the subcarriers (see figure 3.5).

Figure 3.5 OFDM and the orthogonal principal
Obviously the spectrums of the subcarriers are not separated but overlap. The reason why the information transmitted over the carriers can still be separated is the so called orthogonality relation giving the method its name. By using an IFFT for modulation we implicitly chose the spacing of the subcarriers in such a way that at the frequency where we evaluate the received signal (indicated as arrows) all other signals are zero. In order for this orthogonality to be preserved the following must be true:

1. The receiver and the transmitter must be perfectly synchronized. This means they both must assume exactly the same modulation frequency and the same time-scale for transmission (which usually is not the case).

2. The analog components, part of transmitter and receiver, must be of very high quality.

3. There should be no multi-path channel.

In particular the last point is quite a pity, since we have chosen this approach to combat the multipath channel. Fortunately there's an easy solution for this problem: The OFDM symbols are artificially prolonged by periodically repeating the 'tail' of the symbol and precede the symbol with it (see figure 3.5). At the receiver this so called guard interval is removed again. As long as the length of this interval $\Delta$ is longer than the maximum channel delay $\tau_{\text{max}}$ all reflections of previous symbols are removed and the orthogonality is preserved. Of course this is not for free, since by preceding the useful part of length $T_u$ by the guard interval we lose some parts of the signal that cannot be used for transmitting information. Taking all this into account the signal model for the OFDM transmission over a multi-path channel becomes very simple: The transmitted symbols at time-slot $I$ and subcarrier $k$ are only disturbed by a factor $H_{l,k}$ which is the channel transfer function (the fourier transform of the cir) at the subcarrier frequency, an by additional white Gaussian noise $n$.
\[ z_{i,k} = r_{i,k} \cdot H_{i,k} + n \]

The influence of the channel can easily be removed dividing by \( H_{i,k} \).

As far as the analog components are concerned experience has shown that in the broadcasting applications under consideration here, they are not so critical. What remains is to establish 'perfect' synchronization. This requires a very sophisticated receiver.

3.4.1 Generation of OFDM signal

In order to generate OFDM signal successfully, the relationship between all the carriers must be carefully controlled to maintain the orthogonality of the carriers. First of all, we have to choose the spectrum required which is based on the input data, and modulation scheme used. Each carrier to be produced is assigned some data to transmit. The required amplitude and phase of the carrier is then calculated based on the modulation scheme (differential BPSK, QPSK, or QAM). The required spectrum is then converted back to its time domain signal using an Inverse Fourier Transform. In most applications, an Inverse Fast Fourier Transform (IFFT) is used. The IFFT performs the transformation very efficiently, and provides a simple way of ensuring the carrier signals produced are orthogonal.

The Fast Fourier Transform (FFT) transforms a cyclic time domain signal into its equivalent frequency spectrum. This is done by finding the equivalent waveform, generated by a sum of orthogonal sinusoidal components. The amplitude and phase of the sinusoidal components represent the frequency spectrum of the time domain signal. The IFFT performs the reverse process, transforming a spectrum (amplitude and phase of each component) into a time domain signal. An IFFT converts a number of complex data point in frequency spectrum used for an FFT or IFFT is called a bin. The orthogonal carriers required for the OFDM
signal can be easily generated by setting the amplitude and phase of each bin, then performing the IFFT as shown in figure 3.6.

**Transmitter**

![Transmitter Diagram](image)

**Receiver**

![Receiver Diagram](image)

*Figure 3.6 Setup of a basic OFDM transmitter and receiver.*

The signal generated is a base band, thus the signal is filtered, then stepped up in frequency before transmitting the signal.

### 3.4.2 Adding a Guard Period to OFDM

One of the most important properties of OFDM transmissions is the robustness against multi-path delay spread. This is achieved by having a long symbol period, which minimizes the inter-symbol interference. The level of robustness can in fact be increased even more by the addition of a guard period between transmitted symbols. The guard period allows time for multipath signals from the previous symbol to die away before the information from the current symbol is gathered. The most effective guard period to use is a cyclic extension of the symbol. If a mirror in time, of the end of the symbol waveform is put at the start of the symbol as the guard period, this effectively extends the length of the
symbol, while maintaining the orthogonality of the waveform. Using this cyclic extended symbol the samples required for performing the FFT (to decode the symbol) can be taken anywhere over the length of the symbol. This provides multi-path immunity as well as symbol time synchronization tolerance.

As long as the multi-path delay echoes stay within the guard period duration, there is strictly no limitation regarding the signal level of the echoes: they may even exceed the signal level of the shorter path! The signal energy from all paths just adds at the input to the receiver, and since the FFT is energy conservative, the whole available power feeds the decoder. If the delay spread is longer then the guard interval then they begin to cause inter-symbol interference. However, provided the echoes are sufficiently small they do not cause significant problems. This is true most of the time, as multi-path echoes delayed longer than the guard period will have been reflected of very distant objects.

Other variations of guard periods are possible. One possible variation is to have half the guard period a cyclic extension of the symbol, as above, and the other half a zero amplitude signal. This will result in a signal as shown in Figure 3.7. Using this method the symbols can be easily identified. This possibly allows for symbol timing to be recovered from the signal, simply by applying envelop detection. The disadvantage of using this guard period method is that the zero period does not give any multi-path tolerance, thus the effective active guard period is halved in length. It is interesting to note that this guard period method has not been mentioned in any of the research papers read, and it is still not clear whether symbol timing needs to be recovered using this method.
Figure 3.7 Section of an OFDM signal showing 5 symbols, using a guard period, which is half a cyclic extension of the symbol, and half a zero amplitude signal (for a signal using a 2048 point FFT and 512-sample total guard period.)
3.4.3 The Advantages and Disadvantages of OFDM

The advantages [9] of OFDM are:

- Spectrum efficiency, meaning more bps/Hz than conventional transmission schemes.
- Spectral efficiency is further enhanced as the spectrum can be made to look like a rectangular window, meaning all frequencies are utilized similarly.
- OFDM is less sensitive to timing errors. A timing error is simply translated to a phase offset in the frequency domain.

As mentioned earlier in Section 3.4, the disadvantages of OFDM are:

- The receiver and transmitter must be perfectly synchronized. This means that both the transmitter and receiver must assume exactly the same modulation frequency and the same time-scale for transmission (which usually is not the case).
- The analog components, part of transmitter and receiver. Must be of very high quality.
- There should be no multi-path channel.
4. PRINCIPLES OF 8-VSB

4.1 System Overview

8-VSB, the terrestrial broadcast mode support a payload data rate of 19.28 Mbps in a 6 MHz channel. A functional block diagram of a representative 8-VSB terrestrial broadcast transmitter is shown in Figure 4.1. The input to the transmission subsystem from the transport subsystem is a 19.39 Mbps serial data stream comprised of 188-byte MPEG-compatible data packets (including a sync byte and 187 bytes of data that represent a payload data rate of 19.28 Mbps). The incoming data is randomized and then processed for forward error correction (FEC) in the form of Reed-Solomon (RS) coding (20 RS parity bytes are added to each packet), 1/6 data field interleaving and 2/3 rate trellis coding. The randomization and FEC processes are not applied to the sync byte of the transport packet, which is represented in transmission by a Data Segment Sync signal as described below. Following randomization and forward error correction processing, the data packets are formatted into Data Frames for transmission and Data Segment Sync and Data Field Sync are added.
Figure 4.1 VSB transmitter

Figure 4.2 shows how the data are organized for transmission. Each Data Frame consists of two Data Fields, each containing 313 Data Segments. The first Data Segment of each Data Field is a unique synchronizing signal (Data Field Sync) and includes the training sequence used by the equalizer in the receiver. The remaining 312 Data Segments each carry the equivalent of the data from one 188-byte transport packet plus its associated FEC overhead. The actual data in each Data Segment comes from several transport packets because of data interleaving. Each Data Segment consists of 832 symbols. The first 4 symbols are transmitted in binary form and provide segment synchronization. This Data Segment Sync signal also represents the sync byte of the 188-byte MPEG-compatible transport packet. The remaining 828 symbols of each Data Segment carry data equivalent to the remaining 187 bytes of a transport packet and its associated FEC overhead. These 828 symbols are transmitted as 8-level signals and therefore carry three bits per symbol. Thus, 828 x 3 = 2484 bits of data are carried in each Data Segment, which exactly matches the requirement to send a protected transport packet:
Figure 4.2 VSB data frame

87 data bytes + 20 RS parity bytes = 207 bytes

207 bytes x 8 bits/byte = 1656 bits

2/3 rate trellis coding requires 3/2 x 1656 bits = 2484 bits.

The exact symbol rate is given by equation 1 below:

(1) $S_r (MHz) = \frac{4.5}{286} \times 684 = 10.76... MHz$
The frequency of a Data Segment is given in equation 2 below:

\[
(2) \quad f_{\text{seg}} = \frac{S_r}{832} = 12.94\ldots \times 10^3 \text{ Data Segments/s.}
\]

The Data Frame rate is given by equation (3) below:

\[
(3) \quad f_{\text{frame}} = \frac{f_{\text{seg}}}{626} = 20.66\ldots \text{ frames/s.}
\]

The symbol rate $S_r$ and the transport rate $T_r$ shall be locked to each other in frequency.

The 8-level symbols combined with the binary Data Segment Sync and Data Field Sync signals shall be used to suppress-carrier modulate a single carrier. Before transmission, however, most of the lower sideband shall be removed. The resulting spectrum is flat, except for the band edges where a nominal square root raised cosine response results in 620 kHz transition regions. The nominal VSB transmission spectrum is shown in Figure 4.3.

At the suppressed-carrier frequency, 310 kHz from the lower band edge, a small pilot shall be added to the signal.

![Figure 4.3 VSB channel occupancy (nominal).](image-url)
4.2 Nyquist Filter

[11] Like the traditional NTSC, the 8-VSB format utilizes a vestigial sideband approach in the interest of conserving spectrum space. The resulting frequency response after the Nyquist VSB filter is shown in figure 4.4.

![Figure 4.4 8-VSB RF Frequency Spectrum](image)

Note: the presence of ATSC pilot at lower edge of channel. The lower sideband (below pilot frequency) is almost completely removed.

This virtual elimination of the lower sideband, along with the narrowband filtering of the upper sideband, creates very significant changes in the RF waveform that is ultimately transmitted. For the
NTSC-hardened veteran, there is a great temptation to imagine the 8-VSB RF waveform as being a sort of "8-step luminance stair step" signal transmitting the eight levels of 8-VSB. Unfortunately, there is a fundamental flaw with this notion. As figure 4.5 illustrates, such a crisp stair step signal with "squared off" abrupt transitions would generate a frequency spectrum that is far too wide for the single 6 MHz channel. A "square symbol pulse" -type signal generates a rich spectrum of frequency side lobes that would interfere with adjacent channels.

![Figure 4.5 Impossible 8-VSB RF Envelope](image)

However, the 8-level information is only recognized during the precise instant of sampling in the receiver. At all other times, the symbol pulse amplitude is unimportant and can be modified in any way we please - so long as the amplitude at the precise instant of sampling still assumes one of the required eight amplitude levels.

If the narrowband frequency filtering is done correctly according to the Nyquist Theorem, the resulting train of symbol pulses will be orthogonal. This means that at each precise instant of sampling, only one symbol pulse will contribute to the final RF envelope waveform; all preceding and following symbol pulses will be experiencing a zero crossing in their amplitude. In this way, when the RF waveform is sampled by the receiver clock, the recovered voltage will represent only the current symbol's amplitude (one of the eight possible levels).
At any given sampling time (vertical line), only one symbol pulse contributes to total signal amplitude, all other pulses experience a zero crossing. The resulting RF envelope corresponds to the eight digital levels only during the precise instant of sampling.

4.3 8-VSB Eye Diagram

One popular representation of the 8-VSB signal that emphasizes the principals just discussed is the 8-VSB eye diagram. This diagram turns up in the pages of many articles on 8-VSB and on the screens of many pieces of 8-VSB test equipment. The eye diagram is the overlay of many traces of the received RF signal amplitude at the instant of sampling. Since the RF signal must attain one of eight possible levels whenever a sampling time occurs (somewhat like when the music stops in the game of musical chairs), the convergence of the many signal traces forms seven "eyes" that coincide with the occurrence of clock pulses in the receiver. This is shown in Figure 4.6.
At each sampling time, the demodulated RF amplitude assumes one of eight possible levels. The resulting display creates eight vertical "eyes." If the 8-VSB signal is corrupted during transmission, these "eyes" will close up and disappear as the RF signal will no longer possess the correct amplitude at the right instant.

4.4 8-VSB Signal Constellation

In 8-VSB, the digital information is transmitted exclusively in the amplitude of the RF envelope and not in the phase. This is unlike other digital modulation format such as QAM, where each point in the signal constellation is a certain vector combination of carrier amplitude and phase. This is not possible in 8-VSB since the carrier phase is no longer an independent variable under our control, but is rather "consumed" in suppressing the vestigial lower sideband.

The resulting 8-VSB signal constellation, as compared to 64-QAM, is shown in figure 4.7. The 8 levels are recovered by sampling an in-phase (I channel) synchronous detector. Nothing would be gained by sampling a quadrature channel detector since no useful information is contained in this channel. Our signal constellation diagram is therefore a series of eight vertical lines that correspond to the eight transmitted amplitude levels. By eliminating any dependence on the Q-channel, the 8-VSB receiver need only process the I channel, thereby cutting in half the number of DSP circuits required in certain stages. The result is greater simplicity, and ultimately cost savings, in the receiver design.
8-VSB signal constellation is a series of eight vertical lines on the I (in-phase) axis. The Q (quadrature) axis is not used to convey information. When the 8-VSB RF signal is corrupted, the eight vertical lines become blurred and errors are received.

4.5 Advantages of 8-VSB\textsuperscript{[12]}

The 8-VSB system, in general, has better threshold or carrier-to-noise (C/N) performance, has a higher data rate capability, requires less transmitter power for equivalent coverage, and is more robust to impulse and phase noise.
5. The PRINCIPLES OF QAM

QAM \(^{13}\) utilizes one communication channel to contain two mutually independent orthogonal signals, namely I-channel modulated with a cosine wave and Q-channel modulated with a sine wave. This pair of I- and Q-channel can be thought as a complex signal. In QAM, two components (real and imaginary) of the complex signal is merged into one real signal at the modulator and then transmitted. At the receiver, the demodulator separates the merged two channels back to I- and Q-channel by using the orthogonal properties of sinusoidal waves:

\[
\int_{T_i} \sin \omega_c t_c \cos \omega_c t_c \, dt_c = 0
\]

\[
\int_{T_i} \sin \omega_c t_c \sin \omega_c t_c \, dt_c = \frac{T_i}{2}
\]

\[
\int_{T_i} \cos \omega_c t_c \cos \omega_c t_c \, dt_c = \frac{T_i}{2}
\]

The mathematical description of a QAM signal is given by

\[f(t) = l_i \cos \omega_c t + Q_i \sin \omega_c t\]

Where, \(\omega_c\) is a carrier frequency. The QAM signal \(f(t)\) is defined for every symbol period \(0 \leq t \leq T_s\). The period of the carrier signal, i.e. \(T_c = 2\pi/\omega_c\), and the symbol period \(T_s\) may have a relationship \(T_s = kT_c\). The integer \(k\) tells how many carrier cycles occupy one symbol period. \(l_i\) and \(Q_i\) are the magnitude of
cosine and sine wave, respectively. If we define a number of QAM levels being, \( M = 2^N \),

\[ I_i = Q_i = i - \frac{M + 1}{2} \quad \text{for } i = 1, \ldots, M \]

\( I_i \) and \( Q_i \) are independent of each other.

The QAM is standardized in terms of the number of the states determinable by both I- and Q-channel, which is \( M^2 \) if both channels use the same number of amplitude levels. If the power \( N \) of \( 2^N \) is 1, the QAM is QAM-4. For \( N = 2, 3 \) and \( 4 \), the corresponding QAM's are QAM-16, QAM-64 and QAM-256, respectively. Fig. 1 shows the block diagram of a QAM modulator. The \textit{Formatter} in Fig. 5.1 takes a bit stream of digital data as input, and then segments the bit stream into blocks of \( N \) bits. Those segmented blocks are alternately given to I-and Q-channel to determine their amplitudes.
Two types of QAM demodulators are shown in Fig.5.2. The modulated I- and Q-channel cannot be individually separated once the QAM modulator merges those. However, the amplitudes of I- and Q-channel can be obtained by taking the correlation between the received QAM signal and the carrier sine or cosine signal. The multiplications of a QAM signal and the carrier signals $\sin\omega_c t$ and $\cos\omega_c t$ and the integrators in Fig.5.2 are for the correlation calculations. As a result of the demodulation, the amplitudes of I- and Q-channel are obtained. A pair of the two amplitudes is usually mapped into a constellation map, which shows how the points taking the same state stay closely, allowing correct bit
stream reconstruction. The Reformatter performs logical operations in order to determine which bit pattern has arrived for I- and Q-channel from the state in the constellation map.

The demodulation scheme described above uses QAM signal's correlations to cos{w_c + 0} and sin{w_c - 0} measured for a symbol time. Another method to demodulate QAM signals uses a low-pass filter for each of I- and Q-channel. Assuming a single frequency component of cos (ω_c + ω_1) in I-channel and cos (ω_c - ω_1) in Q-channel, the QAM signal after merging I- and Q-channel is given by

\[ f(t) = \cos(\omega_c t + \theta_I) \cos \omega_c t + \cos(\omega_c t - \theta_Q) \sin \omega_c t \]

The received QAM signal f(t) is multiplied by the carrier signals cosω_c and sinω_c similarly as the previous scheme. This operation produces,

\[ f(t) = \cos \omega_c t = \frac{1}{4} \left\{ 2 \cos(\omega_c t + \theta_I) \right. \]
\[ + \cos((2\omega_c + \omega_1)t + \theta_Q) \]
\[ + \cos((2\omega_c - \omega_1)t - \theta_Q) \]
\[ - \sin((2\omega_c + \omega_1)t + \theta_I) \]
\[ - \sin((2\omega_c - \omega_1)t - \theta_I) \]

\[ f(t) = \sin \omega_c t = \frac{1}{4} \left\{ 2 \cos(\omega_c t + \theta_I) \right. \]
\[ + \sin((2\omega_c + \omega_1)t + \theta_Q) \]
\[ + \sin((2\omega_c - \omega_1)t - \theta_Q) \]
\[ - \cos((2\omega_c + \omega_1)t + \theta_I) \]
\[ - \cos((2\omega_c - \omega_1)t - \theta_I) \]

Since the frequencies, ω_1 and ω_0 are much less than the carrier frequency ω_c, it is possible to recover the frequency components cosω_c and sinω_c by eliminating the frequency components at 2ω_c ± ω_1 and 2ω_c ± ω_0. A low-pass filter having a cut-off frequency of can extract the original I- and Q-channel signal. This method, however, does not segment those I- and Q-channel signals into individual symbol intervals. Depending on the binary bit stream
representation, such as RZ, NRZ or biphasic, suitable edge detection may be used to find the symbol-to-symbol boundaries.

5.1 QAM Generation and Detection\textsuperscript{[14]}

The simplest form of QAM is in fact the QPSK symbol set, which can be viewed as two quadrature amplitude modulated carriers, with amplitude levels of and $-A$. Increasing the number of amplitude levels on each carrier to 4, for example, $A$, $3A$, gives 16 possible combinations of symbols at the transmitter output, each equally spaced on the constellation diagram, and each represented by a unique amplitude and phase.

Pulse shaping is performed by filtering the multi-level base band input symbol streams in exactly the same manner as would be used for a binary ASK waveform.

The modulator is again making use of orthogonality of the sine and cosine carriers to allow independent detection of the two M-ary ASK data streams at the receiver. The two figures below showing a 16-QAM generation and detection block.
A simulation of a typical 16-QAM using Matlab is carried out. The results are discussed and details are attached in the appendices.
6.  EYE DIAGRAMS

In order to characterize communication system performance in cases where the channel is not infinite in bandwidth or may be nonlinear, it is convenient to utilize an eye diagram or eye pattern. Figure 6.1 (a) illustrates an experimental setup for measuring an eye pattern, with a typical eye pattern shown in figures 6.1 b & c for the ideal infinite-bandwidth and the non-ideal, finite-bandwidth cases, respectively. For the non-ideal case, the eye is open only three-fourths of what it is for the ideal case, which implies a maximum signal-to-noise ratio degradation of \(-20\log_{10} \frac{3}{4} = 2.5\text{dB}\) (the samples at the detector output are smaller for the non-ideal case and will therefore impose a higher error probability). That is, for the bits giving the minimum eye opening \(\frac{3}{4}\) of the ideal, a signal power of 2.5dB greater than that used in the ideal case is required to achieve a given error probability on these bits. As in figure c suggests, not all bits suffer this minimum eye opening; this some bits give a lower error probability per bit over a long string of bits to get an average error probability.

This discussion shows that an eye diagram is a convenient tool for characterizing the performance of a system experimentally. It is also a convenient tool for a system through computer simulation.
6.1 Diagnosis using the Eye Diagrams

From the eye diagram \[^{14}\] it is possible to make an engineering judgment on the likely performance and sources of degradation in a data communications link. Shown here are examples of eye diagrams for various types of distortion – each having a unique identifiable effect on the appearance of the 'eye opening'.

Figure 6.1 Use of Eye Diagrams for System Characterization
The effect of timing error is seen as a skewing of the eye diagram and a closing of the eye due to the received symbol stream no longer being sampled at the point of zero ISI. The addition of noise affects the timing recovery circuitry and also causes a general closing of the eye until eventually the noise occasionally causes full 'eye-closure' and errors occur.

6.2 Example of a Complex Eye Diagram

Shown here are the eye diagrams for a modulation scheme with four states and 16 states in the demodulated signal. This would be typical for one branch of a 16-QAM and 256-QAM modem. The eye diagram as a diagnosis tool comes into its own here, showing clearly the individual 'eyes' between the discrete states, and also illustrating how critical the sample timing must be to detect the symbol at the maximum eye opening.
It also serves to show how much more susceptible higher-order modulation formats are to noise and distortion when compared with a binary system of equivalent energy per bit. The eye opening for the four-level system is so much narrower and the spacing between decision boundaries so much smaller than those in the previous two-state examples and this gets progressively worse as the number of symbol states rises. Great care should be exercised when using the eye diagram for diagnosis to ensure that the observation is made after all the filtering within the system.

6.3 QAM Constellation Analysis $^{[16]}

QAM constellation analysis represents the next generation of digital testing equipment. It is a method by which the installation engineer may visually foresee potentially serious problems in the cable network. By measuring the QAM signal and displaying it diagrammatically and
numerically, the MER (modulation error ratio) can be given a new, visual form capable of displaying a large amount of information. The following demonstration gives an accurate picture (no pun intended) of how QAM appears.

Figure 6.4 Constellation analysis for a good signal

Figure 6.4 shows a good signal with an MER of 30 dB. The smaller the constellation points the better the signal.

Figure 6.5 Constellation analysis for a poor signal

With reference to figure 6.5, this signal has a very poor signal to noise ratio, the picture from the set top box would be perfect, however another 2 dB of degradation and the signal would fail.
Figure 6.6 Constellation analysis for a lost lock signal

In figure 6.6 shows the signal has lost lock, hence will not be expecting to see any pictures on the TV.
7. **FUTURE DEVELOPMENT WORK**

With the theory research work attained so far during these few weeks, part 2 for this project will be preceded as follows:

- Target application description and requirements
- Feasibility on implementation, choice of modulation scheme
- Identify platform for implementation
- Select implementation of parts/concept to be verified
- Implementation with hardware, software, integration
- Evaluation and performance verification
- Final project report
8. RECOMMENDATION

Modulation scheme such as OFDM, 8-VSB and QAM increases the number of bit/Hz or bits per symbol of a data communication link when compared to conventional schemes. In an AWGN channel of 6 to 8 MHz bandwidth, these modulation schemes can offer similar bit rates, and spectral efficiency.

However, differences lie in their respective ability to handle performance impairments due to multi-path interference, fading, timing delays, noise etc.. OFDM is less prone to timing errors, while 8-VSB offers better threshold or carrier-to-noise C/N performance. Quadrature Amplitude Modulation, QAM, is susceptible to noise and more suitable for cable than for wireless. Despite the susceptibility to noise, Rayleigh fading and multipath interference, QAM has evolved from its original implementation to digital QAM with added defence such as carrier recovery, AGC and channel equalization to overcome such weakness.

The modulation scheme will determine the complexity of hardware platform suitable for implementation in the target application. The choice of a suitable modulation scheme will require understanding of the target application and its requirements. This next phase of this project will commence with a description and requirements of the target application, and choice of a suitable modulation scheme. This is followed by feasibility study on implementation; identification of a suitable platform; implementation; and evaluation and performance verification.
REFERENCES FOR PART 1

PART 2  PROJECT REPORT
CHAPTER 9

9. INTRODUCTION

This chapter presents a brief introduction on the Final Year Project titled - A cable/Wireless Modulation/Demodulation Frontend Module For High Speed/Broadband Data Link. The objectives of this project are first introduced, then followed by the report organization.

9.1 Motivation

Today's network devices, computing peripherals, internet appliance, home entertainment electronics comes with a varied form of physical interface (USB, Firewire, 10Base-T, 10Base-2, etc...) and supported media (UTPs, coaxial, wireless, etc). It will be interesting to explore the concept of allowing a number of network appliances with different network interface to be hooked on to a single shared cable/wireless medium through a class of 'Medium-Transceiver-Interface (MTI)'. This would offer the user, say in a home environment, easy access to the resultant networked resources and make inter-operation between various devices/appliance/appliance possible (via a protocol translator/server).
The heart of such a class of MTI (other than data control and medium/channel arbitration unit) is the modulation/demodulator module, which has to provide a reliable, bandwidth efficient, digital/data modulation and demodulation for transmission and reception of data. The objective of this project is firstly, a detail study of the digital/data modulation/demodulation suitable for such a class of MTI, evaluating its technical feasibility and constraints. Modulation/demodulation schemes in consideration include OFDM, 8-VSB and QAM. Secondly, implementation of a suitable modulation and demodulation scheme for the MTI. The former is treated in the theory research attained in the part 1 for this project as presented (1).

9.2 Project Objectives

With the theory research attained in the part 1 for this project, (that is in the course of Engineering Project 1), the objectives that carried forward to this implementation phase are:

- Target application description and requirements
- Feasibility on implementation, choice of modulation scheme
- Identify platform for implementation
- Select implementation of parts/concepts to be verified
- Implementation with hardware, firmware, integration
- Evaluation and performance verification
Chapter 10 presents the description of the target application and requirements; its feasibility and choice of modulation scheme. It also identifies the platform for implementation. It also specifies what is not expected in the project.

Chapter 11 discusses the implementation of the various project modules. It provides the design, approaches to show the problems encountered and schematic of the hardware.

Chapter 12 provides the verification of hardware and firmware.

Chapter 13 lists the problems encountered and the possible improvement to the design.

Chapter 14 concludes the report.
10. TARGET APPLICATION

10.1 Modulation Schemes

Modulations are the techniques to carry digital data over analog waveforms. This rather arcane subject has been brought to the forefront of the DSP, EE, and telecommunications worlds by the ongoing interest in broadband communications, specifically, the great opportunity represented by bringing low cost broadband communications to the home.[2]

The modulation process places (analog or digital) signal information onto sine wave carriers while demodulation reverses the process at the receiving end. Modulation schemes are very much in the news today because newer algorithms that take advantage of newer and more powerful DSP architectures make possible faster and more reliable communications than was possible before.

There are many types of modulation schemes available today. However, in this project QAM has been adopted as the modulation scheme out of the many.

QAM systems combine PSK and ASK to increase the number of states per symbol. QAM is a proven technique for the transmission of digital
data over a wide range of channels from voice band modems at 9600 bps to microwave links transmitting hundreds of Mbps. Each symbol value in QAM modulation technique represents multiple bits. 16-QAM carries 4 bits per symbol while 256-QAM carries 8 bits per symbol. The signal-to-noise ratio at the receiver determines the QAM level that can be used reliably on a given transmission channel.

10.2 Why QAM

Quadrature amplitude modulation is a widely used form of digital modulation in systems where a high data rate is required and where the bandwidth is limited. The channel must have good short term amplitude characteristics along with reasonably linear delay characteristics and a good signal-to-noise ratio.\[3\]

QAM separates points widely and is hence fairly noise immune. The system for 16-QAM combines 4 inputs bits to produce 1 signal burst. Both phase and amplitude are modulated. Odd-numbered bits in the input stream are combined in pairs to form one of 4 levels, which modulate the sine term. Even-numbered bits are similarly combined to modify the cosine term. Sine and cosine terms are then combined.
\[ V(t) = x(t) \cos \frac{1}{2} t + y(t) \sin \frac{1}{2} t \]

![I-Q diagram or constellation for 16-QAM scheme](image)

16-QAM has better spectral efficiency than 8-PSK and is less sensitive to noise than 16-PSK because the spacing between symbols are larger. This is true because the symbols are not all on the same circle; the resultant signals are not all of the same amplitude. QAM is also used in some upstream traffic designs, but is less noise resistant, though more bit-efficient, than QPSK, for example in today's common network such as cable modem. Typical terrestrial and cable channels allow 16-QAM and 256-QAM, leading data rates of approximately 20 and 40 Mbps, respectively. With this scheme in this project, it could be used to verify whether this can be used over the power-line at home over short distance.
10.3 System Description

Fig 10.2 illustrating the block diagram of a typical target application. As mentioned in chapter 9, the heart of the MTI is the modulation / demodulation for transmission and reception of data. The medium will be a shared cable or wireless medium.

Figure 10.2 Typical Target Application
This project implements a 10Base-T MTI for such a target application.

<table>
<thead>
<tr>
<th>(4) Media transceiver</th>
<th>(3) QAM Modulator/ Demodulator</th>
<th>(2) Symbol Mapper/ decoder</th>
<th>(1) 10Base-T interface</th>
</tr>
</thead>
</table>

This project covers modules (1)-(3) in order to have a meaningful coverage and the means to test and verify the implementation making use of a 10Base-T test data frame bitstream.

The project does not implement the media transceiver.
10.4 Hardware Architecture/Platform Selection

The hardware platform chosen for this project depends on the following:

- Availability
- Delivery
- Cost

For 10Base-T interface module, it was determined that a design based on a 10Base-T transceiver IC with digital output will suit well.

The choice to implement an analog or digital QAM modulator/demodulation becomes dependent on the availability of platform off-the-shelf and time constraints. It was decided that the prudent first approach is to implement an analog QAM modulation/modulation.

This could follow by implementation of digital QAM, depending on availability of time and hardware.

As for the symbol mapper/decoder, the debate is whether to implement using discrete ICs or using PCA, PLD, CPLD or FPGA. To provide the possibility to implement digital QAM, and take advantage of the flexibility and success rate in design, it was determined to adopt the use of a FPGA in the symbol mapper/decoder.
Advantages of using FPGA:

Digital Design can be implemented easily in FPGAs. The advantages of using FPGA for digital system prototyping are summarized as below.

- **High success rate** – FPGAs are arrays of digital gates fabricated on a single substrate that can be configured to perform specific functions. Hence, a design which has its functionality verified on the FPGAs has a very high chance of functioning on the actual Integrated Circuits (IC).

- **Low cost** – Since a FPGA can be reconfigured an unlimited number of times, the same FPGA can be reused and hence this can reduce the prototyping cost.

- **Easy design exploration** – FPGA provides the designer the ability to explore different design architectures and to choose the best implementation for fabrication.

The use of FPGA warrants the use of VHDL in the development of the symbol mapper/decoder and any future improvement.
11. IMPLEMENTATION

11.1 10BASE-T Interface Module

This module is designed around Motorola's MC34055 10 Base-T transceiver. This 10 Base-T transceiver is designed to comply with IEEE 802.3 specification. It was primarily designed for use in repeater or hub applications, and thus a digital interface is provided, which suits this project well. This IC has an on chip oscillator, capable of receiving a clock input or operation under crystal control, is provided for internal timing and driving a buffered clock out for external use. For a detail functional description, please refer to MC34055's datasheets in Appendix F. The completed design is as shown in Figure 11.1. The circuit is layout on a 65x80mm printed circuit board.
Key signals that are provided to the symbol-mapper are digital out 10Base-T received signal (RxdataA), recovered clock (rxcclk) - recovered from the Manchester coded 10Mbps data, and received data enable. For symbol decoder, 10MHz, 20MHz (derived from 10MHz) is provided for its timing generation. TxDataA (Manchester coded 10Mbps data) and Tx enable from the symbol decode is required for data transmission to the twisted-pair.
Figure 11.1 Schematic for 10Base-T Interface Module
11.2 QAM Symbol Mapper

The block diagram of the QAM symbol mapper for implementation is as shown in Figure 11.2. The complete block is implemented in a Atmel AT4K20 FPGA. The block is developed using VHDL, synthesized, compiled, its bit-stream download to the FPGA, and then tested, function by function, starting from the simple flip-flops in a layered manner as reflected in the complete VHDL listing for the symbol mapper. Complete VHDL listing, block diagram, data flow, and flow charts for the symbol decoder are given in Appendix D.

![Figure 11.2 Block diagram of QAM Symbol Mapper](image)
To provide timing for a 4 bit data for both I and Q symbols, the timing
generator consist of two parts, a 4 bit positive-edge-triggered counter,
which is resets on every 4\textsuperscript{th} falling-edge clock, and the reset pulse
generator which is a negative-edged-triggered 4 bit counter that resets
itself and the 4 bit positive-edge-triggered counter. While the reset pulse
generator resets itself on the 4\textsuperscript{th} falling-edge clock, it is also resets itself
and the 4 bit positive-edge-triggered counter at the beginning of every
data frame (provided by \textit{rxen}, the received data enable from the
10Base-T transceiver and the first clock cycle of the data frame). On
detecting the rising-edge of \textit{rxen}, flip-flop 1 setup the negative-edged-
triggered 4-bit counter to reset (synchronous) on the first clock. Flip-flop
resets itself on the rising-edge of second clock.

The main clock used is \textit{rxclk}, which is the 20MHz clock recovered from
the 10Mbps Manchester-coded data (\textit{rxDataA}) in the 10Base-T
interface module. Since this clock is derived from \textit{rxDataA}, it is
synchronized with \textit{rxDataA} to ensure correct symbol capture.
Preliminary investigations shows that \textit{rxclk}'s positive width varies
considerably, so it is re-timed by detecting it rising edge and the flip-flop
chain resets when positive width reaches 25 ns (\textit{rxclk1}). This will
provide improved accuracy for the timing generator. The output of this
timing generator are \textit{dacSelect}, \textit{dacWrite}, \textit{dacClk} for the transmit QAM
dac. It also generates the symbol data latch (\textit{symdatalatch}) pulse to
clock in the symbols from the serial-to-parallel data converter.
The serial to parallel data converter is constructed from 4 cascade D flip-flop, resets by rxen at the end of every data frame. Data $Rx_{dataA_d}$ is buffered $rxd_{dataA}$, where it’s rising and falling times are reduced. Also, the clock used is a delayed clock pulses $(rxclk2)$ derived from the positive edge of $rxclk$. The outputs are clocked into 4 D-flip-flop at each rising-edge pulse of $symd_{atalatch}$ to provide the 4-bit symbol in the symbol capture block. Output from the symbol capture block is fed to a encoding block.

It should be noted, that, though the data is Manchester coded, for symbol capture, it is treated as a 20Mbps NRZ data. The useful bandwidth of a random Manchester signal is double that of the data rate of the signal, as a Manchester signal contains no DC component, and is a passband signal. However, for this project, it is decided not to decode the Manchester data stream, firstly, to avoid the complexity in implementing a good Manchester data decoder, which is not the primary objective of this project. Secondly, the use of Manchester coded data is not necessary a disadvantage since encoding multiple bits into several voltage levels on the transmit signal or using orthogonal modulation techniques, pulse shaping will improve the bandwidth efficiency.

The encoding scheme takes advantage of the fact a 10Mbps 4bit Manchester coded data will only result in 16 unique states when treated as a 20Mbps NRZ for every 8 bits (4 for I channel, 4 for Q channel). This is illustrated in the Table 1.
The reduced number of states instead of 256 provides better margins against signal impairment due to noise in the medium and receiver LNA, gain control in variable gain amplifier (VGA) in preceding RF stages receiver ADC inaccuracy etc..

### Table 1: Manchester Coded Data

<table>
<thead>
<tr>
<th>Actual Data</th>
<th>Manchester Coded Data</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0101 0101</td>
<td>55</td>
</tr>
<tr>
<td>0001</td>
<td>0101 0110</td>
<td>56</td>
</tr>
<tr>
<td>0010</td>
<td>0101 1001</td>
<td>59</td>
</tr>
<tr>
<td>0011</td>
<td>0101 1010</td>
<td>5A</td>
</tr>
<tr>
<td>0100</td>
<td>0110 0101</td>
<td>65</td>
</tr>
<tr>
<td>0101</td>
<td>0110 0110</td>
<td>66</td>
</tr>
<tr>
<td>0110</td>
<td>0110 1001</td>
<td>69</td>
</tr>
<tr>
<td>0111</td>
<td>0110 1010</td>
<td>6A</td>
</tr>
<tr>
<td>1000</td>
<td>1001 0101</td>
<td>95</td>
</tr>
<tr>
<td>1001</td>
<td>1001 0110</td>
<td>96</td>
</tr>
<tr>
<td>1010</td>
<td>1001 1001</td>
<td>99</td>
</tr>
<tr>
<td>1011</td>
<td>1001 1010</td>
<td>9A</td>
</tr>
<tr>
<td>1100</td>
<td>1010 0101</td>
<td>A5</td>
</tr>
<tr>
<td>1101</td>
<td>1010 0110</td>
<td>A6</td>
</tr>
<tr>
<td>1110</td>
<td>1010 1001</td>
<td>A9</td>
</tr>
<tr>
<td>1111</td>
<td>1010 1010</td>
<td>AA</td>
</tr>
</tbody>
</table>

The symbol mapper implements the encoding function using the four unique 4-bit symbol, that is 5, 6, 9 and A, resulting in 16 possible state, meaning QAM16. The resultant bandwidth of QAM16 for this project on a 10Mbps Manchester coding is $10/4 = 2.5$ MHz, which falls well within bandwidth restriction on broadband data link applications.

To make use of the full 10-bit dynamic range of the transmitting DAC, the mapping scheme uses all 10 bits, each inter-symbol distance of 256 steps.
The mapping scheme for txDac data input is as follows:

When symbol is "0101" => "0010000000" (dec 128)
When symbol is "0110" => "0110000000" (dec 384)
When symbol is "1001" => "1010000000" (dec 640)
When symbol is "1010" => "1110000000" (dec 896)
When symbol is others => "0000000000" and tx_err_flag is raised.

The symbol mapper in fact, groups the symbols into Mbits before being split into two separate parallel I and Q data paths, consisting of M/2 bits. In essence, the symbol mapper is simply a serial-to-parallel shift register. The I and Q data from the symbol mapper are fed into their respective transmit DACs. The inputs constructed into their respective analog signal having discrete levels.
11.3 QAM Transmit DAC and IQ Modulator

An Analog Device's, AD9761, is adopted for implementation in this project mainly because it is complete dual channel, high speed, 10bit CMOS DAC. It was developed specifically for use in wide bandwidth communication applications where I and Q information is being processed during its transmit operations. It integrates two 10bit, 40 MSPS DACs, dual 2x interpolation filters, a voltage reference and digital input circuitry. A complete functional description of this IC can be found in Appendix F.

Figure 11.3 Block diagram for QAM modulator/demodulator Module
Output data from the symbol mapper is interleaved as I and Q signal by dacSelect signal, and is presented to the digital interface circuitry, which consists of I and Q latches. The data is de-interleaved back into its original I and Q data paring. The data output from each latch is then processed by a 2x interpolation filter. The use of an interpolating filter typically eases the implementation and complexity of the analog filter which can be significant contributor to mismatches in gain and phase between the two baseband channels. The interpolated output of each filter serves as the input to their respective 10-bit DACs.

The DAC input registers are latched on the rising-edge of dacWrite signal. The I and Q DACs are simultaneously updated on the rising-edge of dacClock with digital data from their respective 2x digital interpolation filters.

The DACs' outputs are applied to symmetrical nyquist low-pass filters, used to spectrally band-limit the DACs' output by suppressing inherent images. The matching Nyquist filters shapes and limits each component's spectral envelope while minimizing intersymbol interference and the output from each filters serves as input to the IQ modulator.
Since QAM scheme is adopted for this project, IQ modulator is employed in the implementation. The IQ modulator use for this project is Maxim's MAX2450, a ultra-low-power Quadrature modulator/demodulator. The output of each Nyquist filters of the preceding transmit DAC stage amplitude-modulates (via the two mixers) two different RF carriers having the same frequency but are in phase quadrature (i.e., sine and cosine waves having a fixed 90 degree phase offset). The two resulting amplitude-modulated carriers are summed together to produce a final QAM signal, essentially a carrier that is modulated in both amplitude and phase. In this project, the QAM modulation is performed in the analog domain. The MAX2450 combines a quadrature modulator and demodulator with a supporting oscillator. The modulator accepts differential I and Q signals with amplitudes up to 1.35Vp-p and bandwidth of 15MHz.

11.4 QAM Receiveing Demodulator and IQ Demodulator

IQ demodulation is performed in the same IC as for modulation, that is, using Maxim's MAX2450. Functional details are as described in it's datasheets in Appendix F. The modulator contains a single-ended-to-differential converter, two Gilbert-cell multipliers and two fixed gain stages with gain of 14 dB. The amplified IF signal is fed into the I and Q mixers for demodulation by mixing with the quadrature LO signals, resulting in baseband I and Q signals.
Demodulation is the inverse process of modulation, however, demodulation is not so simple in reality. The major difference between the modulation and demodulation diagrams lies in the data recovery and timing recovery circuits. Noise, multipath, fading impairments added to the signal as it transverse the medium, noise introduced receiver RF chain, critical gain control in variable gain amplifier (VGA) in RF stages ADC inaccuracy etc. work against data and timing recovery. However, this subject is beyond the scope of this project.

QAM demodulation is implement using Analog Device’s AD9201. The AD9201 is a complete dual channel 20MSPS, 10-bit CMOS ADC. It is optimized specifically for applications where close matching between two ADCs is required. Each ADC incorporates a simultaneous sampling. Output bits (D0 – D9) are de-interleaved by adcSelect channel, and could be latched into the symbol decoder. Details specification can be found in Appendix F.

The complete QAM transmitting DAC, modulator, demodulator and receiving ADC are layout on one 65x88mm printed circuit board. The schematic for this module is as shown in Figure 11.4.
11.5 QAM – Symbol Decoder

The block diagram of the QAM symbol decoder for implementation is as show in Figure 11.5. Just like the symbol mapper, the complete block is implemented in an Atmel AT4K20 FPGA, and is developed in the similar approach. Complete VHDL listing, block diagram, data flow, and flow charts for the symbol decoder are in Appendix C.

Figure 11.5 Block diagram of QAM Symbol Decoder
The receiver timing generator consists of two 4bit counter, reset on the 4th rising-edge clock, one driven by a 10MHz clock, the other 20 MHz. The clock signals are clock pulse (clk10_1 and clk20_1 respectively) derived from 10MHz and 20 MHz clock signal from the 10Base-T interface module. The timing generator provides the necessary clocking signals for the ADC (adcClk and adcSelect) andadcDlatch signal for clocking in the interleaved data from the ADC.

Output data bits rxadcData (0 – 9) are latch into the symbol decoder on the rising edge of adcDlatch signal, and stored as rx_bits (0 – 9). The decoding circuitry takes the 4 most significant bits to decode as follows:

```
when "1110" => "0101";    rxdatadecoded = '1'
when "1101" => "0101";    rxdatadecoded = '1'
when "1010" => "1001";    rxdatadecoded = '1'
when "1001" => "1001";    rxdatadecoded = '1'
when "0110" => "0110";    rxdatadecoded = '1'
when "0101" => "0110";    rxdatadecoded = '1'
when "0010" => "1010";    rxdatadecoded = '1'
when "0001" => "1010";    rxdatadecoded = '1'
when others => "0000";    rxdatadecoded = '0'
```
Only the 4 MSB are use because the 6 LSB are ignored to provide a tolerance of ±63 steps for the decoding the four unique symbols. The status of the decode is provided by the rxdatadecoded flag. The decoded symbols are converted to serial data by the parallel to serial converter. A parallel load enable (pL) and data in clk (cc(0)) signals are generated from adcDlatch with sufficient delay after adcDlatch rising edge. Data in clk cc(0) is OR with clk20_1 clock to provide the required clock for parallel load and shifting all 4 bits of data out before the next adcDlatch rising-edge event.

The complete schematic is as shown in Figure 11.6.
12. TEST AND VERIFICATION

Total of three printed circuit boards based on the schematic was fabricated, and populated with the required components and debugged. The printed circuit boards, namely the 10BaseT interface module, QAM module and a test board where, the two modules and the commercially available FPGA module are plugged into for tests. The test board consists of a regulated 5Vdc supply and provides the interconnections between the three modules. See Appendix C for photos of fabricated prototypes.

Key equipment used for the test:

- PC or notebook with 10Base-T NIC running a continuous test program that sends out a repeated 57.6μs data frame
- A 100Mz Tektronic TDS220 scope.
12.1 QAM Symbol Mapper Verification

Figure 12.1 Measured transmitter timings
Figure 12.1 Measured transmitter timings (cont’d)

Figure 12.1 shows the timings for the QAM symbol mapper. The measured timings are as expected, however, it is noted that glitches are found on the symbol capture_err_flag and output bits to the DAC, This occurs when data input to the mapper encoder block are in transition state. However, since no data latch operates during this transition, it does not pose any problem to the operation of the QAM symbol Mapper.
12.2 QAM Modulator

The two following figures show the $I$ and $Q$ output from the transmitting DACs. First figure shows the first 50 $\mu$s of 10Base-T test data frame. The second shows a more close-up snap shot of 250 ns window, at pos19.5$\mu$s.

Figure 12.2 $txDac$ IQ signal output
Figure 12.2 (cont'd) More detailed IQ signal wrt 10Mbps Manchester coded data and symbol data latch
12.3 IQ Modulator, Demodulator, QAM Demodulator

Due to time constraints, the above functions are not evaluated at the time of writing this report. It will be completed and presented during the final presentation of this project.
12.4 QAM Symbol Decoder

Figure 12.3 Rx ADC and symbol decoding timings
Figure 12.4 Measured Receiver Parallel to Serial Conversion Timings

Measured QAM symbol decoder timings are as show in Figure 12.3 and 12.4. Works as it should be.
13. RECOMMENDATION/FUTURE IMPROVEMENT

- Synchronized 20MHz clock for data stream from the 10Base-T module has varied negative width. The existing design used three Dual Retriggerable Monostable Multivibrator with reset IC (74 VHC 123). The first is used to detect the rising and falling edge of the Manchester coded data. The positive width of the output is set near to 50ns. The second and third IC is used to detect the rising and falling edge of the outputs of the first IC.

However due to different delays in the second and third IC, the resultant 20MHz after adding (OR) the four output have varied negative width. This could give rise to erroneous symbol capture. With careful re-time and delay of the clock and use of the positive edge clock pulses, the problem is under control but not elegant.

The same concept could be implemented in the FPGA where all gates are on the same die, would give more predictable delays.
Alternative to a non-oversampling clock recovery is to use two matched gated square-wave oscillators that are start and stopped by the data signals, however, implementation will define longer time.

- The symbol mapper/decoder does not implement the traditional sense of QAM mapping where odd-bits in the bit stream are combined to form a symbol for I channel, and the even-bits to form a symbol for the Q channel.

To implement this means not making use of the advantage of a Manchester coded bitstream. It may not be a disadvantage after all for a Manchester signal compared to an NRZ signal.

- VHDL — It is important to appreciate that synthesis tools sometime dumb. There are no reserved words in VHDL to specify whether a model is conditional or sequential and whatever the sequential logic is synchronous or asynchronous. Thus as a beginner in VHDL, one of the fundamental problem with synthesizing VHDL modelling is to ensure that the hardware produce by the synthesis system is what really desired. Example of good synthesis rules are given in [5].

- In an actual hardware implementation, any mismatch in gain, phase, and offset between the I and Q channels—in both the modulator and demodulator—can result in errors in the recovery process. The filters preceding the ADC may need to provide adaptive equalization to compensate for various channel impairments.
Filtering in the digital domain to ensure perfect matching characteristics between the I and Q channel, as well as phase alignment of the quadrature carriers would be a good approach. A high-speed performance ADC and DAC must then be used to reconstruct and sample the QAM IF signal. Hence, present time constraint does not permit this implementation.
14. CONCLUSION

QAM is used in many digital communications applications since it provides very efficient use of frequency spectrum. In this project, the choice of a suitable modulation scheme, that is, Quadrature Amplitude Modulation, QAM, has indeed able to fulfill the target application and its requirement. Though QAM, is susceptible to noise and more suitable for cable than for wireless, the merit gained most by QAM is the number of bits that can be transmitted within a symbol period. For most of the digital communication schemes, this is 1 bit/symbol period. In designing a QAM based digital communication system, bit rate is the key design factor. The discrete levels represented by assigned to I- and Q-channel must then decided

A major challenge for this project is to realize the implementation of the modulation / demodulation scheme with hardware and firmware integration.

With such a short project lead-time (less than 10 weeks), it is a major challenge to have hardware fabricate firmware written, debug, test and evaluate and write this report. This project was carried with a very short lead-time leading to many ad hoc adaptations to overcome unforeseen circumstance.
Due to time constraints, the section in the evaluation of the QAM was not able to complete at the time of writing this report. Therefore, it is targeted to continue the work after the completion of this report. And the results will be then presented during the presentation seminar on the 24th February 2001.

Hence, in conclusion, the objectives of the project have more or less been met. Most importantly is the valuable knowledge gained during the design cycle.
REFERENCES FOR PART 2

[1] Engineering project 1 progress report


PART 3 ANNEX TO PART 2
ANNEX 1 Symbol Mapper – Decoder Loop-back tests

**Equipment:**

- Test data frame: notebook running 10Base-T NIC test program
- Tektronix TDS224 oscilloscope with WSTRO Wavestar Software for waveform capture

**Preparation:**

To facilitate this loop-back test, the main clock for the symbol decoder now uses the same clock as the symbol mapper to enable oscilloscope triggering of both mapper and decoder signals. Changes are made in the VHDL files, after integrating both mapper and decoder functions, clk20 is replaced by rxclk, and clk10 is now generated internally. Other changes are adapting rxtimegen block so that it can be reset at the beginning of every data frame by signal rxen. After compilation, bit-stream is downloaded to the FPGA. A physical data bridge for the four MSB of txDACdata to rxADCdata is made, and tests is conducted.
Results:

Figure A1.1 Comparing Symbol Mapper input and Symbol decoder output

Checking at the 'critical' points in the test data frame, it was found that at the first instance when the Manchester coded data changes from 0101010101.... to 010110000.... data error was found in the time between cursor 1 and 2 of Figure A1.1.

The reason for the error was due to the problem of varying negative width of the 20MHz clock (rxclk) derived from the Manchester Coded 10Mbps bit-stream as discussed in Chapter 13. This caused a skewed parallel load (pL) and clk20_1 signals in the parallel-to-serial converter clock resulting in one bit less data output. This problem is depicted in Figure A1.2a and A1.2b.
Narrower, less than normal negative width

Figure A1.2a Varying negative width of rxclk

Figure A1.2b Skewed pL and clk20_1
Modifications:

To solve the skewed parallel-to-serial converter clock, it is decided to implement the clock recovery from the 10Mbps Manchester coded bit-stream in the FPGA than to rely on the existing unreliable external implementation. There are two approach, first, using the same as the existing external implementation, that is using both rising and falling-edge detections as described in the main report, second, a non-over sampling clock-recovery method by implementing two 'near-matched' gated square-wave oscillators similar to [4].

The second approach is taken, with the two near match oscillator implemented in the same FPGA (entity clksync). Apart from this change, to remove ambiguity in the behavioral parallel-to-serial converter, a structural parallel-to-serial converter is use in the VHDL coding and without using an extra clock (clk20plus1) for clocking as in a normal parallel-to-serial shift registers. Its reset value is changed to '1111' instead of '0000'.

Other changes includes T-flip-flop's variable state is replaced by signal state because the former reset part was not synthesized; update rxtimegen block using T-flip-flops, implement reset using signal rssi and assign rssi to rxen for this test program; change symbol decoder's recovered symbol rxsymbol value when correct data is not decoded to "1111" instead of "0000" ; add entity rxen_extended generation to extend the oscillator for another eight cycles after rxen has transition from high to low, to enable clocking of the last data symbol at the end a data frame and to reset the parallel-to-serial converter (though not very elegantly nor efficiently implemented with use of 30 self-reset flip-flops cascade); various changes to correct timings e.g. parallel-load signal (pL); add signal dacReset to enable reset of tx DAC at the beginning of each data frame.
The updated block diagram of the symbol mapper/decoder module for test and verification implement in the FPGA is as shown in Figure A1.3.
Figure A1.3. Updated Block Diagram of Symbol Mapper/Decoder
Results:

After synthesis and compilation, the generated bitstream is downloaded to the FPGA. Measured result is as shown in Figure A1.4, which shows that the problem is resolved.

![Figure A1.4 Received Data bitstream after modifications](image)

**Figure A1.4** Received Data bitstream after modifications

Notes and observations:

It was found though the design use only 16% of the AT40K20, problems begin to appear in placement optimization and route optimization when using Atmel's IDS6.0 (Figaro) for generation of the bitstream, slowing done project progress considerably. This may be due to the use of quite a number of derived clocks and cascaded flip-flops delays in the design. Alternative is to find a better synthesis tool and compiler.
ANNEX 2 Improved 20MHz Clock Recovery

A new clock recovery from the 10Mbps Manchester Coded data is implemented in the FPGA. Its concept is based on a non-oversampling clock-recovery method by implementing two ‘near-matched’ gated square-wave oscillators similar to [4]. The implemented clock recovery block diagram is as shown below.

![Clock Recovery Block Diagram](image)

**Figure A2.1 Clock Recovery Block Diagram**
Two 'near-matched' gated square wave oscillators (A and B) are started and stopped successively by the falling-edge of the data signal. The recovered clock is obtained by combining their outputs. This solution is adopted because it is memory-less and any accumulating phase errors due to the oscillators frequency mismatch is discarded every time the oscillators are stopped.

Each of the gated oscillators is design using a cascade of flip-flop delays.

![Gated Oscillator Diagram]

**Figure A2.2 Gated Oscillator**

This design is a fully integrated clock recovery circuit that locks instantaneously to the first arriving data of a burst-mode data frame. Though data is Manchester coded here, it is treated as a 20Mbps NRZ bit stream. There is a 1-bit delay in re-timed data and recovered clock signal does inherit data transition jitter.
Figure A2.3 Typical Measured Waveform at 20Mbps
ANNEX 3 QAM, IQ Mod/Demod Test and Verification

It was found when testing the transmit DAC performance, the DAC outputs are correct only after approximately 23μs of the 57.6μs data frame (see Figure A3.1). It seems that the DAC finite state machine that controls the generation of the divided clock and pairing of I and Q data inputs has difficulties in getting into the correct state for it to work properly, despite all SELECT, WRITE, CLOCK and RESET signals are provided accordingly. Investigation is going on to find the real cause of this problem. As such, functional blocks that follows the modulator, namely the Nyquist filters, IQ modulation are not tested nor verified at the moment. So are the IQ demodulation, and QAM demodulation.

![Figure A3.1 DAC output error](image)

**Figure A3.1 DAC output error**
ANNEX 4 Nyquist Filter Design

The Nyquist filter intended for implementation is as shown in Figure A4.1. The design is simulated using SIMetrix (spice simulator). With a symbol rate of 2.5MSPS, \( \alpha = 0.5 \), it means the filter corner frequency is at 1.25MHz. The final simulated result is as shown in Figure A4.2. 8 dB of voltage insertion loss is expected from this design. It should be noted that the 3dB frequency is at 1.32MHz instead of 1.25MHz with the use of standard value components.

Figure A4.1 Schematic for Nyquist Filter
Figure A4.2 Nyquist Filter Frequency Response
ANNEX 5 Updated VHDL code for Symbol Mapper/Decoder
-- entity declaration of QAM16 symbol mapper and decoder for 10Mbps Manchester coded data
-- hardware test version clk10, clk20 generated from rclk to latch scope triggering
-- during simulations rx and tx tests

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY QAM16_mapperdecoder IS
  PORT(
    -- tx ports
    clk10, clk20, rssi : in std_logic;
    adck1, adck2, adclatch, clk10_1, clk20_1 : out std_logic;
    ndatadecoded, ndataA, b, cren, clk20plus1 : out std_logic;
    pl : out std_logic;
    cc : out std_logic_vector(1 downto 0);
    rxdataA, rxen : in std_logic_vector(9 downto 0);
    rxdataA_a : out std_logic_vector(9 downto 0);
    -- rx ports
    rxdatal, rxen, rxdalk : in std_logic;
    rxselect, rxdacWrite, rxdacClk, rxdacReset : out std_logic;
    symB0, symB1, symB2, symB3, symB4, symB5 : out std_logic;
    symbol : out std_logic_vector(3 downto 0);
    rxdacData : out std_logic_vector(9 downto 0);
  );
END QAM16_mapperdecoder;
ARCHITECTURE QAM16_mapperdecoderArch OF QAM16_mapperdecoder IS
  COMPONENT rx_symbol_decoder
    PORT ( clk10, clk20, rssi : in std_logic;
      adck1, adck2, adclatch, clk10_1, clk20_1 : out std_logic;
      ndatadecoded, ndataA, b, cren, clk20plus1 : out std_logic;
      pl : out std_logic;
      cc : out std_logic_vector(1 downto 0);
      rxdataA, rxen : in std_logic_vector(9 downto 0));
  END COMPONENT;

  COMPONENT symbolmapper
    PORT ( ndataA, rxdal, rxdal1, rxen, rxdacWrite, rxdacClk, rxdacReset : out std_logic;
      symB0, symB1, symB2, symB3, symB4, symB5 : out std_logic;
      symbol : out std_logic_vector(3 downto 0);
      rxdacData : out std_logic_vector(9 downto 0));
  END COMPONENT;

Author: Chong Pay Peng, Clarice
end component;

component cksync
PORT ( 
    ndataA_d : in std_logic;
    ndck : out std_logic;
    ndataA_retimed : out std_logic;
    rxen : in std_logic
 );
end component;

component d1f3
PORT ( 
    d, r, clk : in std_logic;
    q : out std_logic
 );
end component;

component d1f5
PORT ( 
    l, r, clk : in std_logic;
    q : out std_logic
 );
end component;

signal clk10, clk20, rssi, rxdala : std_logic;
signal rxen_extended : std_logic;

begin
    for ndataA
dataA_d <= ndataA;
    end generating
    rxen: rxen_extended
end: rxen_ext_port_map (rxen, rxen_extended);

--
rssi <= rxen_extended;

p0: rx_symbol_decoder Port Map ( 
    ndata10, ndata102, rssi, adcSelect, adcData, clk10, clk20, clk10_1, 
    clk20_1, rxdala rekl, rxen, rxen_3, rxen_4, rxen_5, dacSelect, dacWrite, dacCk, 
    symdatalatch, sym_b1, sym_b2, sym_b3, sym_b4, sym_b5, sym_b6, sym_b7, symbolSelector, symbolSelector);

p1: symbolmapper Port Map ( 
    ndataA_retimed, rxdala, ndck, rxdala_1, rxdala_2, dacSelect, dacWrite, dacCk, 
    symdatalatch, sym_b1, sym_b2, sym_b3, sym_b4, sym_b5, symbolSelector);

p2: cksync port map (ndataA_d, ndck, ndataA_retimed, rxen_extended):

--generate clk10 and clk20
genclk10: d1f5 Port Map (T, rxen_extended, ndck, clk10);
clk20 <= ndck;

--generate dacReset
gendacReset:
p3: d1f3 port map (T, symdatalatch, rxen, dacReset);
end QAM16_mapperdecoderArch;

--entity

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY rxen_ext IS
PORT ( 
    rxen : in std_logic;
    rxen_extended : out std_logic
 );
END rxen_ext;
ARCHITECTURE rxen_extArch of rxen_ext IS

component d1f2
PORT ( 
    d, r, clk : in std_logic;
    q : out std_logic
 );
end component;

component d1f3
PORT ( 
    d, r, clk : in std_logic;
    q : out std_logic
 );
end component;

Author: Chong Pay Peng, Clarice
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY clksync IS
PORT (txdataA_d: in std_logic; rxdata : out std_logic; l2dataA_re: out std_logic; sJgnal11,12,12_bar: std_logic; pO, p1, p2: std_logic; p10, pfl, p12: std_logic; ppO: out std_logic; ppiO: out std_logic; nO, ttl: std_logic; nO_bar <= not nO); signal q: out std_logic; end clksync;
ARCHITECTURE clksyncArch of clksync IS
begin

Author: Chong Pay Peng, Clarice
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY rx_symbol_decoder IS
    PORT(
        dk10, clk20, rssi : in std_logic;
        addc, addSelect, addData : in std_logic;
        addcLatch : out std_logic;
        rxadc0ata : in std_logic_vector(9 downto 0),
        rxsymbol : out std_logic_vector(9 downto 0),
        rx_bits : out std_logic_vector(9 downto 0)
    );
END rx_symbol_decoder,

ARCHITECTURE rx_symbol_decoderArch of rx_symbol_decoder IS
begin
    component rxtimegen
        PORT(
            dk10, clk20, rssi : in std_logic;
            addc, addSelect, addData : in std_logic;
            addcLatch : out std_logic;
            rxadc0ata : in std_logic_vector(9 downto 0),
            rxsymbol : out std_logic_vector(9 downto 0),
            rx_bits : out std_logic_vector(9 downto 0)
        );
    end component

    component symdecoder
        PORT(
            addcLatch : in std_logic;
            rxadc0ata : in std_logic_vector(9 downto 0),
            rx_bits : out std_logic_vector(9 downto 0),
            rxsymbol : out std_logic_vector(9 downto 0)
        );
    end component

    component P2S
        PORT(
            clk1, clk : in std_logic;
            on : std_logic_vector(3 downto 0);
            Set: out std_logic
        );
    end component
end ARCHITECTURE;
component dfl3
  PORT (d, r, clk : in std_logic;
  q : out std_logic)
end component;

signal q0, q1, q2, q3, q4, q5, q6, q7 std_logic;
signal SerialOut std_logic;

begin
  process(clk, d, r)
  begin
    if clk'event and clk = '1' then
      q0 <= d;
      q1 <= r;
      q2 <= d and q1 or r;
      q3 <= d and q2 and r;
      q4 <= d and q3 and r;
      q5 <= d and q4 and r;
      q6 <= d and q5 and r;
      q7 <= d and q6 and r;
      SerialOut <= std_logic_vector(q0 to q7);
    end if;
  end process;
end process;

end symbol_decoderArch.

--entity declaration of symbol decoder
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY symbol_decoder IS
  PORT (adcOlatch : in std_logic;
  rxadc0ata : in std_logic_vector(9 downto 0);
  rx_bits : in std_logic_vector(9 downto 0);
  rxdatadecoded : out std_logic_vector(9 downto 0);
  rxtimegen : out std_logic_vector(9 downto 0));
END symbol_decoder;

ARCHITECTURE symdecoderArch of symbol_decoder IS

component dfl3
  PORT (d, r, clk : in std_logic;
  q : out std_logic)
end component;

signal b9downb6 : std_logic_vector(3 downto 0);

begin
  --clk in adc data
  dflch8 : dfl3 Port Map (nadcData(8), '0', adcOlatch, rx_bits(8));
  dflch6 : dfl3 Port Map (nadcData(6), '0', adcOlatch, rx_bits(6));
  dflch4 : dfl3 Port Map (nadcData(4), '0', adcOlatch, rx_bits(4));
  dflch2 : dfl3 Port Map (nadcData(2), '0', adcOlatch, rx_bits(2));
  dflch1 : dfl3 Port Map (nadcData(1), '0', adcOlatch, rx_bits(1));
  dflch0 : dfl3 Port Map (nadcData(0), '0', adcOlatch, rx_bits(0));
  b9downb6(0) <= rx_bits(0);
  b9downb6(2) <= rx Bits(8);
  b9downb6(1) <= rx Bits(7);
b9downlo6(0) <= rx_bits(6);
--symbol decode
process (b9downlo6)
begin
    nnsymdecencoded <= '0';
    msb <= '0000';
    when '1110' => nnsymbol <= '0101', --!01 0101. LSb righl mosl11!
        r.datadecoded <= '1';
    when '1010' => nnsymbol <= '0110', --!or 0110, LSB right mosl! M
        r.datadecoded <= '1';
    when '0110' => nnsymbol <= '1001', --!or 1001. LSB nght mosl! M
        r.datadecoded <= '1';
    when '0010' => nnsymbol <= '1010', --!or 1010. LSB nght mosl! M
        r.datadecoded <= '1';
    when '0001' => nnsymbol <= '1111', nnsymdecencoded <= '0';
end case.
end process;
end symdecoderArch.

--entity declaration of n timing generator
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY ntimemgen IS
    PORT ( 
        clk10_1, clk20, nssi : in std_logic;
        adcdk, adcDlatch, clk10_1, clk20_1 : out std_logic 
    );
END ntimemgen;
ARCHITECTURE ntimemgenArch of ntimemgen IS
    component df13
        PORT ( 
            d, r, clk : in std_logic;
            q, : out std_logic
        );
    end component;
    component df15
        PORT ( 
            d, r, clk : in std_logic;
            q, : out std_logic
        );
    end component;
signal q0, q1, q2, q3, q4, q5 : std_logic;
BEGIN
    --generate 10 MHz and 20 MHz clock pulses
    ffP0: df13 Port Map ('1', clk10_1, clk10, clk10_1);
    ffP1: df13 Port Map ('1', clk20_1, clk20, clk20_1);
    --T-flip flop counter block
    ffT0: df13 Port Map ('1', nssi, clk20_1, q0);
    ffT1: df15 Port Map ('1', nssi, q0, q1);
    ffT2: df15 Port Map ('1', nssi, q1, q2);
    --generate adcdk
    ffP10: df13 Port Map ('1', q3, q2, adcdk);
    ffP11: df13 Port Map ('1', adcdk, q1);
    ffP12: df13 Port Map ('1', q2, q1, q2);
    ffP13: df13 Port Map ('1', q3, q2, q3);
    --generate adcDlatch
    qpp_dk <= nnsymdecencoded;
    ffP2: df13 Port Map ('1', adcdk, qpp_dk, adcDlatch);
    --generate adcSelect
    adcSelect <= nnsymdecencoded;
END ntimemgenArch;

--entity declaration of parallel to serial converter +ve edge triggered
--active low nresl
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY P2S IS
    PORT ( 
        clk1, pl : in std_logic;

Author: Chong Pay Peng, Clarice
ARCHITECTURE P2S_arch of P2S IS

BEGIN

END P2S_arch,

ARCHITECTURE P2S_arch of P2S IS

component dt4

PORT ( d1: in std_logic;
q : out std_logic

end component,

signal dl, d2, d3: std_logic;
signal d1_bar, d2_bar, d3_bar: std_logic;
signal s0, s1, s2: std_logic;
signal s3: std_logic
BEGIN

dl_bar <= not dl;
d2 <= D;
d3 <= D;
d1_bar <= not dl;
d2_bar <= not dl;
d3_bar <= not dl;
r1 <= pl and d3_bar;
r2 <= pl and d2_bar;
r3 <= pl and d1_bar;
r0 <= pl and d2_bar;
s3 <= pl and d3;

END component dt4

component symbolmapper

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

process (sympol)

begin

end process

END symbolmapper

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.

--entity declaration of tx_ndata symbol mapper

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY symbolmapper IS

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

case symbol is
when "0101" =>

when "1001" =>

when "0110" =>

when "1010" =>

end case

END symbolmapper;

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.

--entity declaration of tx_ndata symbol mapper

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY symbolmapper IS

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

case symbol is
when "0101" =>

when "1001" =>

when "0110" =>

when "1010" =>

end case

END symbolmapper;

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.

--entity declaration of tx_ndata symbol mapper

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY symbolmapper IS

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

case symbol is
when "0101" =>

when "1001" =>

when "0110" =>

when "1010" =>

end case

END symbolmapper;

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.

--entity declaration of tx_ndata symbol mapper

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY symbolmapper IS

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

case symbol is
when "0101" =>

when "1001" =>

when "0110" =>

when "1010" =>

end case

END symbolmapper;

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.

--entity declaration of tx_ndata symbol mapper

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY symbolmapper IS

PORT ( rxdata,rxdk, rxen: in std_logic;
rxclk, rxclkd, dacSel, dacWnt, dacCk: out std_logic;
symData: out std_logic_vector(3 downto 0);
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

begin

case symbol is
when "0101" =>

when "1001" =>

when "0110" =>

when "1010" =>

end case

END symbolmapper;

ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbollookup

PORT ( ndata, ndx1, ndx2, dacSel, dacWnt, dacCk, symData, tx_en_flag: in std_logic;
sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
symbol: out std_logic_vector(3 downto 0);

end component

begin

process (symbol)

begin

END P2S_arch.
when others => txdata <= "0000000000"; tx_err_flag <= '1';

end case;

end process;
scl· symboiCJp Port Map
  (ndata, nclk, nen, nclk1, nclk2, dacSelect, dacWrite, dacClk, syndatalatch, sym_b0, sym_b1, sym_b2, sym_b3, symbol);
end symbolcaptureArch;

--entity declaration of symbol capture
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY symbolcap IS
  PORT ( ndata, nclk, nen, nclk1, nclk2, dacSelect, dacWrite, dacClk, syndatalatch out std_logic, sym_b0, sym_b1, sym_b2, sym_b3 out std_logic, symbol out std_logic_vector (3 downto 0) );
END symbolcap;

ARCHITECTURE symbolcapArch of symbolcap IS

component btmugen
  PORT ( nclk, nen, nclk1, nclk2, dacSelect, dacWrite, dacClk, syndatalatch, out std_logic );
end component;

component df3
  PORT ( d, r, cK in std_logic, q out std_logic );
end component;

component df5 IS
  PORT ( d, r, cK, in std_logic, q out std_logic );
end component;

begin
  -- beginning
  p3: btmugen Port Map ( nclk, nen, nclk1, nclk2, dacSelect, dacWrite, dacClk, syndatalatch);
  -- symbol capture
  f3sym0: df3 Port Map ( ndata, nen, nclk2, sym_b3);
  f3sym1: df3 Port Map ( sym_b3, nen, nclk2, sym_b2);
  f3sym2: df3 Port Map ( sym_b2, nen, nclk2, sym_b1);
  f3sym3: df3 Port Map ( sym_b1, nen, nclk2, sym_b0);
  df3rd1: df3 Port Map ( sym_b0, V, syndatalatch, symbol(0));
  df3rd2: df3 Port Map ( sym_b0, V, syndatalatch, symbol(1));
  df3rd3: df3 Port Map ( sym_b0, V, syndatalatch, symbol(2));
  df3rd4: df3 Port Map ( sym_b0, V, syndatalatch, symbol(3));
end symbolcaptureArch;

--entity declaration of a timing generator
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY btmugen IS
  PORT ( nclk, nen, in std_logic, nclk1, nclk2, dacSelect, dacWrite, dacClk, syndatalatch: out std_logic );
END btmugen;

ARCHITECTURE btmugenArch of btmugen IS

component brxRst1
  PORT ( d, cK : in std_logic, rst out std_logic );
end component;

component counter64ps
  PORT ( d, cK, rst : in std_logic, q0, q1, q2, q3 : out std_logic );
end component;

component df1
  PORT ( r, cK : in std_logic);
end component;
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY bi4Rst IS
  PORT ( d, clk : in std_logic;
         out std_logic;
         rst : out std_logic
  );
END bi4Rst;

ARCHITECTURE bi4RstArch OF bi4Rst IS

component df3
  PORT ( d, clk : in std_logic;
         q : out std_logic
  );
end component;

component df4
  PORT ( d, r, s, clk : in std_logic;
         q : out std_logic
  );
end component;

signal r1, r2, q0, q1, q2, q3, q4 std_logic;
BEGIN
  f3f0: df3 port Map (9'tb"0v", d, q0);
  f3f1: df3 port Map (9'tb"0v", d, q1);
  f3f2: df3 port Map (9'tb"0v", d, q2);
  f3f3: df3 port Map (9'tb"0v", d, q3);
  f3f4: df3 port Map (9'tb"0v", d, q4);
  r0 <= q0;
END bi4RstArch;

--entity declaration of 4 bit reset generator edge
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY counter4pos IS
  PORT ( d, clk, rst : in std_logic;
         out std_logic;
         q0, q1, q2, q3 : out std_logic
  );
END counter4pos;

ARCHITECTURE counter4posArch OF counter4pos IS

component df3
  PORT ( d, clk, rst : in std_logic;
         q : out std_logic
  );
end component;

component df4
  PORT ( d, r, s, clk : in std_logic;
         q : out std_logic
  );
end component;

signal r1, r2, q0, q1, q2, q3, q4 std_logic;
BEGIN
  f3f0: df3 port Map (9'tb"0v", d, q0);
  f3f1: df3 port Map (9'tb"0v", d, q1);
  f3f2: df3 port Map (9'tb"0v", d, q2);
  f3f3: df3 port Map (9'tb"0v", d, q3);
  f3f4: df3 port Map (9'tb"0v", d, q4);
  r0 <= q0;
END counter4posArch;

--entity declaration of 4 bit counter edge
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY counter4pos IS
  PORT ( d, clk, rst : in std_logic;
         out std_logic;
         q0, q1, q2, q3 : out std_logic
  );
END counter4pos;

ARCHITECTURE counter4posArch OF counter4pos IS

component df3
  PORT ( d, clk, rst : in std_logic;
         q : out std_logic
  );
end component;

component df4
  PORT ( d, r, s, clk : in std_logic;
         q : out std_logic
  );
end component;

signal r1, r2, q0, q1, q2, q3, q4 std_logic;
BEGIN
  f3f0: df3 port Map (9'tb"0v", d, q0);
  f3f1: df3 port Map (9'tb"0v", d, q1);
  f3f2: df3 port Map (9'tb"0v", d, q2);
  f3f3: df3 port Map (9'tb"0v", d, q3);
  f3f4: df3 port Map (9'tb"0v", d, q4);
  r0 <= q0;
END counter4posArch;
END counter4pos.
ARCHITECTURE counter4posArch of counter4pos IS
---
component dff3
PORT ( d, r, clk : in std_logic;
q : out std_logic
); end component.
---
BEGIN
ifp0. dff3 Port Map (d, r, clk, q0);
ifp1. dff3 Port Map (q0, r, clk, q1);
ifp2. dff3 Port Map (q1, r, clk, q2);
ifp3. dff3 Port Map (q2, r, clk, q3);
END counter4posArch.
---

--entity declaration of dff1 with reset +ve edge triggered
--active high reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff1 IS
PORT ( r, d, clk ; in std_logic;
q : out std_logic
); end ENTITY dff1.
ARCHITECTURE dff1Arch of dff1 IS
BEGIN
process (r, clk)
begin
if (r='1') then
q <= '0';
elsif (d='1' and clk'event) then
q <= '1';
end if.
end process.
END dff1Arch.
---

--entity declaration of dff2 with reset +ve edge triggered
--active high reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff2 IS
PORT ( d, r, clk : in std_logic;
q : out std_logic
); end ENTITY dff2.
ARCHITECTURE dff2Arch of dff2 IS
BEGIN
process (r, clk)
begin
if (r='1') then
q <= '0';
elsif (clk='0' and clk'event) then
q <= d;
end if.
end process.
END dff2Arch.
---

--entity declaration of dff3 with reset +ve edge triggered
--active high reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff3 IS
PORT ( d, r, clk : in std_logic;
q : out std_logic
); end ENTITY dff3.
END dff3.
ARCHITECTURE dff_arch of dff IS
BEGIN
process (r, s, dk)
begin
  if (r-'1') then
    q <= '0';
  elsif (s='1' and dk'event) then
    q <= s;
  else
    q <= d;
  end if;
end process;
END dff_arch;

ARCHITECTURE dff5arch of dff5 IS
BEGIN
  process (t, r, s, clk)
  begin
    variable state : std_logic;
    begin
      if (t='0') then
        state <= '0';
      elsif (clk='1' and clk'event) then
        if (t='1') then
          state <= not state;
        else
          q <= state;
        end if;
      end if;
    end process;
  end process;
END dff5arch;

ARCHITECTURE dff6arch of dff6 IS
BEGIN
  process (d, r, s, clk)
  begin
    if (d='0') then
      q <= '0';
    elseif (clk='1' and clk'event) then
      if (d='1') then
        q <= '1';
      else
        q <= s;
      end if;
    end if;
  end process;
END dff6arch;
process (r, clk)
begin
  if (r = '0') then
    q <= '0';
  else (clk = '1' and clk'delay) then
    q <= d;
  end if;
end process.

END architecture.
A randomly generated bit stream is divided into a pair of 2 bit words, one for I-channel and the other for Q-channel. The QAM modulation routine of the program generated a QAM signal. Figure App.1 below shows a close-up view of the QAM signal without noise. There are three distinct amplitude levels, which correspond to the three possible distances in the constellation map from the origin to stars, are observed. The QAM wave is non-coherent in spite of the coherent carrier used as a source of QAM modulation.

![QAM Waveform](image)

**Figure App.1**  A waveform of QAM signal without noise
A QAM signal observed for a longer period of time with added uniform random noise is shown in Fig App.2.

Figure App.2  A waveform of QAM signal with noise

The demodulator routine continues to integrate the product of the QAM signal and sine/cosine carrier signal for every symbol time. The frequency of the carrier signal is assumed to be known. At first, no active carrier signal recovery is considered. Carrier signal phase is, however, an important parameter to maintain the constellation map upright so that the reformatter in the demodulator can be a simple discriminator based on thresholds. If the phase is known, the constellation map can be rotated as accordingly to the phase angle. Figure App.3 shows the 16QAM constellation map without noise added, while that of figure App.4 shows the constellation map with noise added.
Figure App.3  Constellation Map of QAM – 16 without noise

Figure App.4  Constellation Map of QAM – 16 with noise added

The dispersion seen in each star position is due to the inserted noise.

Author: Chong Pay Peng, Clarice
APPENDIX B 16-QAM MATLAB SIMULATION

print_key=input('Do you make eps files (y/n)? ','s');
noise_key=input('Do you include noise (y/n)? ','s');
phase_key=input('Do you add a phase-shift (y/n)? ','s');
if phase_key=='y'
    phase=pi/6;
else
    phase=0.0;
end;
dtamax=input('Specify I/Q data size (def=128) dtamax=');
if dtamax>256
    dtamax=256;
end;
if dtamax<=0
    dtamax=16;
end;
pilot_key=input('Do you include pilot signal of wc/2 (y/n)? ','s');
if pilot_key=='y'
adaptive_key=input('Do you adaptively adjust phase (y/n)? ','s');
else
    adaptive_key='n';
end;
% (1) Generate a random data sequence
dta=[];
% dtamax=128;
qlevel=1/4;
for k=1:dtamax*2;
    dta=[dta, fix(rand/qlevel)];
end;
% (2) Generate QAM16 signal
T=1; % sampling time
M=16; % ratio of sampling freq/carrier freq
N=4; % ratio of carrier freq/symbol freq
Tc=M*T; % carrier time interval
Ts=N*Tc; % symbol time interval

Author: Chong Pay Peng, Clarice
\[ W_c = 2\pi \frac{f_c}{T_c}; \quad \% \text{carrier frequency} \]
\[ t = \text{time range: } d_{\max} \cdot N \cdot M; \]
\[ s = []; \quad \% \text{QAM16 Signal} \]
for \( k = 1 : d_{\max}; \)
\[ I_{\text{data}} = d_{\text{data}}(2^k - 1) - 1.5; \]
\[ Q_{\text{data}} = d_{\text{data}}(2^k) - 1.5; \]
for \( n = 1 : N; \)
\[ I_{\text{data}} = d_{\text{data}}(2^k - 1) - 1.5; \]
for \( m = 1 : M; \)
\[ \text{time} = T_s \cdot (k - 1) + T_c \cdot (n - 1) + T \cdot (m - 1); \]
if pilot_key == 'y'
\[ \text{pilot} = \sin(W_c \cdot \text{time} / 2 + \text{phase} / 2); \]
else
\[ \text{pilot} = 0.0; \]
end;
if noise_key == 'y'
\[ \text{noise} = 1.0 \cdot (\text{rand} - 0.5); \]
s = [s, I_{\text{data}} \cdot \cos(W_c \cdot \text{time} + \text{phase}) + Q_{\text{data}} \cdot \sin(W_c \cdot \text{time} + \text{phase}) + \text{noise} + \text{pilot}];
else
\[ s = [s, I_{\text{data}} \cdot \cos(W_c \cdot \text{time} + \text{phase}) + Q_{\text{data}} \cdot \sin(W_c \cdot \text{time} + \text{phase}) + \text{pilot}]; \]
end;
end;
end;
figure(1);
plot(s(1:d_{\text{max}} \cdot M \cdot N));
title('QAM16 Waveform');
xlabel('Time (units: sampling time T)');
ylabel('Amplitude');
if print_key == 'y'
if noise_key == 'y'
\[ \text{print -deps qam16n.eps}; \]
else
\[ \text{print -deps qam16.eps}; \]
end;
end;
\% (3) Demodulate QAM16 Signal
\[ \theta = 0.0; \]
\[ \text{Angle} = []; \]
\[ \text{AIinteg} = []; \quad \text{AQinteg} = []; \]
\[ \text{Iraw} = []; \quad \text{Qraw} = []; \]
for \( k = 1 : d_{\text{max}}; \)
\[ \text{AQinteg} = 0; \]
\[ \text{AIinteg} = 0; \]
\[ \text{Iinteg} = 0; \]
\[ \text{Qinteg} = 0; \]
for \( n = 1 : N; \)
for \( m = 1 : M; \)
\[ \text{time} = T_s \cdot (k - 1) + T_c \cdot (n - 1) + T \cdot (m - 1); \]
\[ \text{received} = s(N \cdot M \cdot (k - 1) + M \cdot (n - 1) + m); \]
if adaptive_key=='y'
    A = A + received * cos(\(\omega_c \cdot \text{time}/2\)) \cdot T;
    AQ = AQ + received * sin(\(\omega_c \cdot \text{time}/2\)) \cdot T;
end;
    Integ = Integ + received * cos(\(\omega_c \cdot \text{time} + \theta\)) \cdot T;
    QInteg = QInteg + received * sin(\(\omega_c \cdot \text{time} + \theta\)) \cdot T;
end;
end;

Iraw = [Iraw, Integ];
Qraw = [Qraw, QInteg];
A = [A, AQ];
A = [A, A];
if adaptive_key=='y'
    AvgTo = size(AQ, 2);
    AvgFrom = AvgTo - 16 + 1;
    if AvgFrom<1
        AvgFrom = 1;
    end;
    AQ = 0; A = 0;
    for Aindex = AvgFrom:AvgTo
        A = A + AQ(Aindex);
        A = A + A(Aindex);
    end;
    theta = 2 * atan2(A, AQ);
end;

Angl = [Angl, theta];
end;
end;

figure(2);
plot(Iraw, Qraw, 'rx');
title('QAM16 Constellation Map');
xlabel('I-Axis');
ylabel('Q-Axis');
if print_key=='y'
    if noise_key=='y'
        print -deps constln2.eps;
    else
        print -deps constl2.eps;
    end;
end;
end;
figure(3);
plot(abs(fft(s)));
title('QAM16 Spectrum');
print -deps spcnl.eps
if print_key=='y'
    if noise_key=='y'
        print -deps spcn2.eps;
    else
        print -deps spcn2l.eps;
    end;
end;
end;

Author: Chong Pay Peng, Clarice 119
figure(4);
plot(180*Angl/pi);
title('Adjusted Angle');
if print_key=='y'
if noise_key=='y'
   print -deps angln.eps;
else
   print -deps angl.ep
end;
end;
figure(5);
plot(AQintg);
title('Q-Pilot Wave Correlation');
if print_key=='y'
if noise_key=='y'
   print -deps qpiilotn.ep
else
   print -deps qpiilot.ep
end;
end;
figure(6);
plot(AIintg);
title('I-Pilot Wave Correlation');
if print_key=='y'
if noise_key=='y'
   print -deps ipilotn.ep
else
   print -deps ipilot.ep
end;
end;
APPENDIX C

PHOTOS OF PROTOTYPES
View 1  The Whole System
View 2  10Base-T Interface Module
View 3 The Symbol Mapper / Decoder
View 4 The QAM Modulator / Demodulator
APPENDIX D

APPENDIX D.1 VHDL CODE FOR SYMBOL MAPPER
--entity declaration of tx_rxdata symbol mapper-----

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY symbolmapper IS
  PORT ( rxdataA, rxclk, rxen in std_logic;
         rxdataA_d, rxclk1, rxclk2, dSelect, dWrite, dclk, symdata latch: out std_logic;
         sym_b0, sym_b1, sym_b2, sym_b3, tx_err_flag: out std_logic;
         symbol: txdata
  );
END symbolmapper;
ARCHITECTURE symbolmapperArch of symbolmapper IS

component symbolcap
  PORT ( rxdataA, rxclk, rxen in std_logic;
         rxdataA_d, rxclk1, rxclk2, dSelect, dWrite, dclk, symdata latch: out std_logic;
         sym_b0, sym_b1, sym_b2, sym_b3: out std_logic;
         symbol: out std_logic_vector (3 downto 0)
  );
end component;

begin
  process (symbol)
  begin
    txdacData <= "0000000000";
    tx_err_flag <= '0',
    case symbol is
      when "0101" => txdacData <= "1110000000", --for 0101, LSB last
      when "1001" => txdacData <= "0100000000", --for 1001, LSB last
      when "0110" => txdacData <= "0010000000", --for 0110, LSB last
      when "1010" => txdacData <= "0000000000", --for 1010, LSB last
      when others =>
        txdacData <= "0000000000", tx_err_flag <= '1';
    end case;
  end process;
end symbolmapperArch;

--entity declaration of symbol capture
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY symbolcap IS
  PORT ( rxdataA, rxclk, rxen, rxdataA_d, rxclk1, rxclk2,
         dSelect, dWrite, dclk, symdata latch,
         sym_b0, sym_b1, sym_b2, sym_b3, symbol);
END symbolcap;
ARCHITECTURE symbolcapArch of symbolcap IS

component tximegen
  PORT ( rxclk, rxen : in std_logic;
         rxclk1, rxclk2, dSelect, dWrite, dclk, symdata latch: out std_logic
  );
end component;

component dff3
  PORT ( d, r, clk : in std_logic;
         q: out std_logic
  );
end component;

component dff6
  PORT ( d, r, elk: in std logic;
         q: out std_logic
  );
end component;

component buf_dcl
  PORT ( a_in : in std_logic;
         a_out: out std_logic
  );
end component;

Author: Chong Pay Peng, Clarice [**********]
--begin
--buffer for rxdataA-----------------------------------------------
buf0: buf_del Port Map (rxdataA, rxdataA_d),
--ex timing generator---------------------------------------------
p0: txtimegen Port Map (rxclk, rxen, rxclk1, rxclk2, dacSelect, dacWrite, dacClk, symdatalatch),
--symbol eqn()-------------------------------------------------
flipsym0 : dffl Port Map (rxdataA d, rxen, rxclk2, sym b3),
flipsym1 : dffl Port Map (sym b3, rxen, rxclk2, sym b2),
flipsym2 : dffl Port Map (sym b2, rxen, rxclk2, sym b1),
flipsym3 : dffl Port Map (sym b1, rxen, rxclk2, sym b0),
dich1 : dffl Port Map (sym b3, 'W', symdatalatch, symbol(3)),
dich2 : dffl Port Map (sym b2, 'W', symdatalatch, symbol(2)),
dich3 : dffl Port Map (sym b1, 'W', symdatalatch, symbol(1)),
dich4 : dffl Port Map (sym b0, 'W', symdatalatch, symbol(0));
end symbolcapArch;

--entity declaration of ex timing generator
LIBRARY ieee;
USE icox.std_logic_all.
ENTITY txtimegen IS
  PORT ( rxclk, rxen : in std_logic;
     rxclk1, rxclk2 : dacSelect, dacWrite, dacClk, symdatalatch: out std_logic);
END txtimegen;
ARCHITECTURE txtimegenArch of txtimegen IS
-----------------------------------------------
component bit4R5t
  PORT ( d, clk : in std_logic;
     rst: out std_logic);
end component;
-----------------------------------------------
component counter4pos
  PORT ( d, clk, rst : in std_logic;
     q0, q1, q2, q3 : out std_logic);
end component;
-----------------------------------------------
component df 1
  PORT ( r, clk : in std_logic;
     q: out std_logic);
end component;
-----------------------------------------------
component df 2
  PORT ( r, clk : in std_logic;
     q: out std_logic);
end component;
===============================================
signal rxclk, q0, q1, q2, q3, dpl, dp2 : std_logic;
BEGIN
--rcxclk re-time----------------------------------------------
clk0: df 1 Port Map (c4, rxclk, rxclk1);
clk1: df 1 Port Map (c4, rxclk1, c2);
clk2: df 1 Port Map (c4, c2, c3);
clk3: df 1 Port Map (c4, c3, c4);
--rcxclk2, +ve edge pulse of rxclk-----------------------------
--end delay for 12 ns----------------------------------------
clk4: df 2 Port Map (c5, rxclk, c5);
clk5: df 2 Port Map (rxclk2, c5, rxclk2);
--timing generation------------------------------------------
p0: bit4R5t Port Map (rxen, rxclk1, rst);
p1: counter4pos Port Map (rxen, rxclk1, rst, q0, dacWrite, dacClk, q3);
--dacselect generation----------------------------------------
tf1: df 2 Port Map (rxen, rxen, q0, dacSelect);
--symdatalatch generation------------------------------------
ffp1: df1 Port Map (symdatalatch, q3, dpl);
ffp2: df1 Port Map (symdatalatch, dpl, dp2);
ffp3: df1 Port Map (symdatalatch, dp2, symdatalatch);
END txtimegenArch;

Author: Chong Pay Peng, Clarice
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY bit4Rst IS
PORT ( d, clk, rst : in std_logic;
       q : out std_logic
     );
END bit4Rst;
ARCHITECTURE bit4RstArch of bit4Rst IS

---entity declaration of 4 bit reset generator +ve edge ----
-- ok, works good, with global reset!!

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY counter4pos IS
PORT ( d, elk, rst : in std_logic;
       q0, q1, q2, q3 : out std_logic
     );
END counter4pos;
ARCHITECTURE counter4posArch of counter4pos IS

---entity declaration of 4 bit counter +ve edge -----
BEGIN
  ff0: dff3 port Map ('1', r2, d, q0);
  ff1: dff3 port Map (q0, r2, clk, r1);
  ff2: dff3 port Map (r1, r2, clk, r2);
  ff3: dff4 Port Map (q1, q2, q3, clk, q1);
  ff4: dff4 Port Map (q2, q3, clk, q2);
  ff5: dff4 Port Map (q3, clk, q3);
  rst<=<!4;
END counter4posArch;

---entity declaration of 4 bit counter +ve edge -----
BEGIN
  ff0: dff3 port Map ('1', r2, d, q0);
  ff1: dff3 port Map (q0, r2, clk, r1);
  ff2: dff3 port Map (r1, r2, clk, r2);
  ff3: dff4 Port Map (q1, q2, q3, clk, q1);
  ff4: dff4 Port Map (q2, q3, clk, q2);
  ff5: dff4 Port Map (q3, clk, q3);
  rst<=<!4;
END counter4posArch;

---entity declaration of dff with reset +ve edge triggered
---active high reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff IS
PORT ( r, elk : in std_logic;
       q : out std_logic
     );
END dff;
ARCHITECTURE dffArch of dff IS
BEGIN
  process (r, elk)
  begin
    if (r='1') then
      q<=q';
    elsif (elk='1' and clk'event) then
      q<=q;
    end if;
  end process;
END dffArch;
BEGIN process (r, clk) begin if (r='1') then q <= '0'; elsif (clk='0' and clk'event) then q <= d; end if; end process; END dff3arch;

--entity declaration of dff4 -ve edge triggered
--active high synchronous reset
--active high synchronous set
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff4 IS
PORT ( d, r, s, elk : in std_logic;
q : out std_logic );
END dff4;
ARCHITECTURE dff4arch of dff4 IS
BEGIN process (r, s, elk) variable state : std_logic; begin if (r='0') then state := '0'; elsif (clk='0' and clk'event) then if (s='1') then state := '1'; else q <= d; end if; end if; q <= state; end process;
END dff4arch;

--entity declaration of dff5 -ve edge Toggle flip-flop
--active low reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff5 IS
PORT ( t, r, clk : in std_logic;
q : out std_logic );
END dff5;
ARCHITECTURE dff5arch of dff5 IS
BEGIN process (t, r, clk) begin if (r='0') then state := '0'; elsif (clk='1' and clk'event) then if (t='1') then state := not state; end if; q <= state; end if; end process;
END dff5arch;

Author: Chong Pay Peng, Clarice
--entity declaration of dff with reset -ve edge triggered
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff IS
    PORT ( d, r, clk : in std_logic;
           q : out std_logic
        );
END dff;
ARCHITECTURE dffarch of dff IS
BEGIN
    process (r, clk)
    begin
        if (r='0') then
            q <= '0';
        elsif (clk='1' and clk'event) then
            q <= d;
        end if;
    end process;
END dffarch;

--entity declaration of buffer/delay of xx
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY buf_del IS
    PORT ( a_in : in std_logic;
           a_out : out std_logic
        );
END buf_del;
ARCHITECTURE buf_delarch of buf_del IS
BEGIN
    process (a_in)
    begin
        case a_in is
            when '1' =>
                state <= 1;
            when '0' =>
                state <= 0;
        end case;
    end process;
    a_out <= state;
END buf_delarch;

Author: Chong Pay Peng, Clarice
APPENDIX D

APPENDIX D.2 BLOCK DIAGRAMS: SYMBOL MAPPER
APPENDIX

APPENDIX D.3 FLOWCHARTS: SYMBOL MAPPER
Architecture Declarations

Concurrent Statements

Sensitivity List

Process Declarations

Title: QAM16_SymbolMapper/dff3arch
Path: QAM16_SymbolMapper/dff3arch
Edited by chongpp on 09 Feb 2001
APPENDIX E.1 VHDL CODE FOR SYMBOL DECODER
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY rx_symbol_decoder IS
  PORT ( clk10, clk20 : in std_logic;
         adcclk,adcSelect,adcDlatch,clk10_1,clk20_1 : out std_logic;
         rxdatadecoded,txdataA,txen,clk20plus : out std_logic;
         pl : out std_logic;
         cc : out std_logic_vector(1 downto 0);
         rxenData : in std_logic_vector(9 downto 0);
         rxsymbol : out std_logic_vector(3 downto 0);
         rx_bits : out std_logic_vector(9 downto 0)
  );
END rx_symbol_decoder;
ARCHITECTURE rx_symbol_decoderArch of rx_symbol_decoder IS

  component rxtimegen
    PORT ( clk10, clk20 : in std_logic;
           adcclk,adcSelect,adcDlatch,clk10_1,clk20_1 : out std_logic
    );
  end component;
  component symbolcoder
    PORT ( adcDlatch : in std_logic;
           rxdataDecoded : in std_logic_vector(9 downto 0);
           rx_bits : out std_logic_vector(9 downto 0);
           rxsymbol : out std_logic_vector(3 downto 0);
           rxdatadecoded : out std_logic
    );
  end component;
  component P2S
    PORT ( );
  end component;
  component diD
    PORT ( )
  end component;

BEGIN
   rxtimegen Port Map (clk10, clk20,adcclk,adcSelect,adcDlatch,clk10_1,clk20_1);
   symbolcoder Port Map (adcDlatch,rxdataDecoded,rx_bits,rxsymbol,rxdatadecoded);
   p2'S P2S Port Map ('1',clk20plus,pl,rxsymbol,SerialOut);
   rxdataA <= SerialOut;
   txen <= rxdatadecoded;
   generate pl
     fpl: diD Port Map ( rxdatadecoded, q4,adcDlatch, pl);
     fpl1: diD Port Map ('1', q11, pl, q1);
     fpl2: diD Port Map ('1', q11, q1, q2);
     fpl3: diD Port Map ('1', q11, q2, q3);
     fpl4: diD Port Map ('1', q11, q3, q4);
     fpl5: diD Port Map ('1', q11, q4, q5);
     fpl6: diD Port Map ('1', q11, q5, q6);
     fpl7: diD Port Map ('1', q11, q6, q7);
     fpl8: diD Port Map ('1', q11, q7, q8);
     fpl9: diD Port Map ('1', q11, q8, q9);
     fpl10: diD Port Map ('1', q11, q9, q10);
     fpl11: diD Port Map ('1', q11, q10, q11);
   generate clk20plus
     d1: diD Port Map ('1', q90,adcDlatch, q99);
     d11: diD Port Map ('1', q91, q90, q91);
     d12: diD Port Map ('1', q92, q91, q92);
     d2: diD Port Map ('1', q93, q92, q93);
     d3: diD Port Map ('1', q94, q93, q94);
     d4: diD Port Map ('1', q95, q94, q95);
     d5: diD Port Map ('1', q96, q95, q96);
     d6: diD Port Map ('1', q97, q96, q97);
     d7: diD Port Map ('1', q98, q97, q98);
     d8: diD Port Map ('1', q99, q98, q99);
   end generate.
end;

Author: Chong Pay Peng, Clarice
--generate delayed clk20_1

clk0:dff3 Port Map ('1', qqq0,clk20_1, qqq9);
clk1:dff3 Port Map ('1', qqq1, qqq0, qqq1);
clk2:dff3 Port Map ('1', qqq2, qqq1, qqq2);
clk3:dff3 Port Map ('1', qqq3, qqq1, qqq3);
clk4:dff3 Port Map ('1', qqq4, qqq3, qqq4);
clk5:dff3 Port Map ('1', qqq5, qqq4, qqq5);
clk6:dff3 Port Map ('1', qqq6, qqq5, qqq6);
clk7:dff3 Port Map ('1', qqq7, qqq6, qqq7);
clk8:dff3 Port Map ('1', qqq8, qqq7, qqq8);
clk9:dff3 Port Map ('1', qqq9, qqq8, qqq9);
cc(0) <= qqq2; cc(1) <= clk20_1;
process (cc)
begin
  case cc is
  when "01" => clk20plusl <= '1';
  when "10" => clk20plusl <= '1';
  when "11" => clk20plusl <= '1';
  when others => clk20plusl <= '0';
  end case;
end process;
end rx_symbol_decoderArch;

--entity declaration of symbol decoder
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY symdecodcr IS
  PORT ( adcDlatch : in std_logic;
         rxadcData    : in std_logic_vector (9 downto 0);
         rx_bits      : out std_logic_vector (9 downto 0);
         rxsymbol     : out std_logic_vector (3 downto 0);
         rxdatadccodcd: out std_logic);
END rxtimegen;
ARCHITECTURE symdecoderArch of symdecodcr IS

  component dff3
  PORT ( d, r, clk : in std_logic;
         q : out std_logic ;
         sign: out std_logic_vector(3 downto 0));
end component;

  signal b9downto6 : std_logic_vector(3 downto 0);

  begin
  --clk in adc data
  dltch9 :dff3 Port Map (rxadcData(9), '0', adcDlatch, rx_bits(9));
  dltch8 :dff3 Port Map (rxadcData(8), '0', adcDlatch, rx_bits(8));
  dltch7 :dff3 Port Map (rxadcData(7), '0', adcDlatch, rx_bits(7));
  dltch6 :dff3 Port Map (rxadcData(6), '0', adcDlatch, rx_bits(6));
  dltch5 :dff3 Port Map (rxadcData(5), '0', adcDlatch, rx_bits(5));
  dltch4 :dff3 Port Map (rxadcData(4), '0', adcDlatch, rx_bits(4));
  dltch3 :dff3 Port Map (rxadcData(3), '0', adcDlatch, rx_bits(3));
  dltch2 :dff3 Port Map (rxadcData(2), '0', adcDlatch, rx_bits(2));
  dltch1 :dff3 Port Map (rxadcData(1), '0', adcDlatch, rx_bits(1));
  dltch0 :dff3 Port Map (rxadcData(0), '0', adcDlatch, rx_bits(0));
  b9downto6(3) <= rx_bits(9);
  b9downto6(2) <= rx_bits(8);
  b9downto6(1) <= rx_bits(7);
  b9downto6(0) <= rx_bits(6);
  --symbol decode
  process (b9downto6)
  begin
  rxdatadccodcd <= '0';
  rxsymbol <= "0000";
  case b9downto6 is
  when "1110" => rxsymbol <= "0101"; --for 0101, LSB right most bit
  when "1101" => rxsymbol <= "0101";
  when "1110" => rxsymbol <= "0101";
  when others => rxsymbol <= "0101";
  end case;
  when "1110" => rxsymbol <= "0101"; --for 0101, LSB right most bit
  when "1110" => rxsymbol <= "0101";
  when others => rxsymbol <= "0101";
  end process;
end symdecoderArch;
end case;
end process;
end symdecodeArch;

--entity declaration of rx timing generator
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY rxtimegen IS
PORT ( clk10, clk20 : in std_logic;
adcclk, adcSelect, adcDlatch, clk10_1, clk20_1: out std_logic );
END rxtimegen;
ARCHITECTURE rxtimegenArch of rxtimegen IS

component dfT3 PORT ();
end component;

component dfT5 PORT ( t, r, clk : in std_logic;
q: out std_logic );
end component;

signal ql, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12: std_logic;
BEGIN
--generate 10 MHz and 20 MHz clock pulses
ffp0: dfT3 Port Map ('I', clki0_I, clki0, clki0_1);
ffp1: dfT3 Port Map ('I', clki0_1, clki0, clki0_2);
--10 MHz clock in, generate adcclk
ffp10: dfT3 Port Map ('I', q3, clki0_1, adcclk);
ffp11: dfT3 Port Map (adcclk, q3, clki0_1, q1);
ffp12: dfT3 Port Map (q1, clki0_1, q2);
ffp13: dfT3 Port Map (q2, clki0_1, q3);
--20 MHz clock in, adcDlatch
ffp20: dfT3 Port Map ('I', q3, clki0_1, q0);
ffp21: dfT3 Port Map (q0, q3, clki0_1, q1);
ffp22: dfT3 Port Map (q1, q0, q3, clki0_1, q2);
ffp23: dfT3 Port Map (q2, q0, q3, clki0_1, q3);
ffp3: dfT3 Port Map ('I', adcDlatch, q1, adcDlatch);
--generate adcSelect
ffp1: dfT5 Port Map ('I', 'I', q3, q4, q5);
adcSelect <= 'I';
END rxtimegenArch;

--entity declaration of parallel to serial converter +ve edge triggered
--active low reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY P2S IS
PORT ( rstn, clock, pi : in std_logic;
Din : std_logic_vector(3 downto 0);
SerialOut: out std_logic );
END P2S;
ARCHITECTURE P2Sarch of P2S IS
signal q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12: std_logic;
BEGIN
--generate 10 MHz and 20 MHz clock pulses
ffp0: dfT3 Port Map ('I', clki0_I, clki0, clki0_1);
ffp1: dfT3 Port Map ('I', clki0_1, clki0, clki0_2);
--10 MHz clock in, generate adcclk
ffp10: dfT3 Port Map ('I', q3, clki0_1, adcclk);
ffp11: dfT3 Port Map (adcclk, q3, clki0_1, q1);
ffp12: dfT3 Port Map (q1, clki0_1, q2);
ffp13: dfT3 Port Map (q2, clki0_1, q3);
--20 MHz clock in, adcDlatch
ffp20: dfT3 Port Map ('I', q3, clki0_1, q0);
ffp21: dfT3 Port Map (q0, q3, clki0_1, q1);
ffp22: dfT3 Port Map (q1, q0, q3, clki0_1, q2);
ffp23: dfT3 Port Map (q2, q0, q3, clki0_1, q3);
ffp3: dfT3 Port Map ('I', adcDlatch, q1, adcDlatch);
--generate adcSelect
ffp1: dfT5 Port Map ('I', 'I', q3, q4, q5);
adcSelect <= 'I';
END P2Sarch;

Author: Chong Pay Peng, Clarice
elseif (clock'event and clock='1') then
    if (pl='1') then
        diff <= Din;
    else
        start <= diff(0);
        diff <= parbit & diff(3 downto 1);
    end if;
end if;
end process;
SerialOut <= start;
END P3Search;

--entity declaration of dff3 with reset +ve edge triggered
--active high reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff3 IS
    PORT ( d, r, clk : in std_logic;
            q: out std_logic );
END dff3;
ARCHITECTURE dff3arch of dff3 IS
BEGIN
    process (r, clk)
    begin
        if (r='1') then
            q <= '0';
        elsif (clk='1' and clk'event) then
            q <= d;
        end if;
    end process;
END dff3arch;

--entity declaration of dff5 +ve edge Toggle flip-flop
--active low reset
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY dff5 IS
    PORT ( t, r, clk : in std_logic;
            q: out std_logic );
END dff5;
ARCHITECTURE dff5arch of dff5 IS
BEGIN
    process (t, r, clk)
    variable state : std_logic;
    begin
        if (r='0') then
            state := '0';
        elsif (clk='1' and clk'event) then
            if (t='1') then
                state := not state;
            end if;
        end if;
        q <= state;
    end process;
END dff5arch;
APPENDIX E

APPENDIX E.2 BLOCK DIAGRAMS: SYMBOL DECODER
APPENDIX E.3 FLOWCHARTS: SYMBOL DECODER
Architecture Declarations

Concurrent Statements

Sensitivity List
i, r, d, h

Process Declarations
variable state : std_logic;

Start

d0

T

F

(a=d)

T

F

(a=r)

T

F

(a=i)

F

T

a1

state := not state;

End

Title: <enter diagram title here>
Path: QAM16_symDecoder/office/1Search
Edited: by chongpp on 09 Feb 2001

Printed by chongpp on 02/09/2001 at 03:10:16
APPENDIX F

DATA SHEETS
Dual Channel, 20 MHz 10-Bit Resolution CMOS ADC

AD9201

FEATURES
- Complete Dual Matching ADCs
- Low Power Dissipation: 215 mW (±3 V Supply)
- Single Supply: 2.7 V to 5.5 V
- Differential Nonlinearity Error: ±0.4 LSB
- On-Chip Analog Input Buffers
- On-Chip Reference
- Signal-to-Noise Ratio: 57.0 dB
- Over Nine Effective Bits
- Spurious-Free Dynamic Range: -73 dB
- No Missing Codes Guaranteed
- 28-Lead SSOP

PRODUCT DESCRIPTION
The AD9201 is a complete dual channel, 20 MSPS, 10-bit CMOS ADC. The AD9201 is optimized specifically for applications where close matching between two ADCs is required (e.g., 1/0 channel in communications applications). The 20 MHz sampling rate and wide input bandwidth will cover both narrow-band and spread-spectrum channels. The AD9201 integrates two 10-bit, 20 MSPS ADCs, two input buffer amplifiers, an internal voltage reference and multiplexed digital output buffers. Each ADC incorporates a simultaneous sampling sample-and-hold amplifier at its input. The analog inputs are buffered; no external input buffer op amp will be required in most applications. The ADCs are implemented using a multistage pipeline architecture that offers accurate performance and guarantees no missing codes. The outputs of the ADCs are ported to a multiplexed digital output buffer. The AD9201 is manufactured on an advanced low cost CMOS process, operates from a single supply from 2.7 V to 5.5 V, and consumes 215 mW of power (±3.3 V supply). The AD9201 input structure accepts either single-ended or differential signals, providing excellent dynamic performance up to and beyond its 10 MHz Nyquist input frequencies.

PRODUCT HIGHLIGHTS
1. Dual 10-Bit, 20 MSPS ADCs
   - A pair of high performance 20 MSPS ADCs that are optimized for spurious-free dynamic performance are provided for encoding of I and Q or diversity channel information.
2. Low Power
   - Complete CMOS Dual ADC function consumes a low 215 mW on a single supply (±3 V supply). The AD9201 operating on suitable voltages from 2.7 V to 5.5 V.
3. On-Chip Voltage Reference
   - The AD9201 includes an on-chip compensated bandgap voltage reference pin programmable for 1 V or 2 V.
4. On-chip analog input buffers eliminate the need for external op amps in most applications.
5. Single 10-Bit Digital Output Bus
   - The AD9201 ADC outputs are interleaved onto a single output bus saving board space and digital pin count.
6. Small Package
   - The AD9201 offers the complete integrated function in a compact 28-lead SSOP package.
7. Product Family
   - The AD9201 dual ADC is pin compatible with a dual 8-bit ADC (AD9281) and has a companion dual DAC product, the AD9781 dual DAC.
## AD9201 - Specifications

### Resolution

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOLUTION</td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td>Bits</td>
<td></td>
</tr>
</tbody>
</table>

### Conversion Rate

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Nonlinearity</td>
<td>DNL</td>
<td>±0.4</td>
<td>±1</td>
<td>±1</td>
<td>LSB</td>
<td>REFT = 1 V, REFB = 0 V</td>
</tr>
<tr>
<td>Integral Nonlinearity</td>
<td>INL</td>
<td>±1.3</td>
<td>±1.5</td>
<td>±2.5</td>
<td>LSB</td>
<td>REFT = 1 V, REFB = 0 V</td>
</tr>
<tr>
<td>Differential Nonlinearity (SE)</td>
<td>DNL</td>
<td>≤0.5</td>
<td>≤1</td>
<td>≤1</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Integral Nonlinearity (SE)</td>
<td>INL</td>
<td>≤1.5</td>
<td>≤2.5</td>
<td>≤5.4</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Zero-Scale Error, Offset Error</td>
<td>RE</td>
<td>±3.5</td>
<td>±5.4</td>
<td>±10</td>
<td>% FS</td>
<td></td>
</tr>
<tr>
<td>Gain Match</td>
<td>E,</td>
<td>±0.5</td>
<td>±1</td>
<td>±1</td>
<td>LSB</td>
<td></td>
</tr>
<tr>
<td>Offset Match</td>
<td></td>
<td>±5</td>
<td>±10</td>
<td>±20</td>
<td>LSB</td>
<td></td>
</tr>
</tbody>
</table>

### Analog Input

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage Range</td>
<td>AIN</td>
<td>-0.5</td>
<td>AVDD/2</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Capacitance</td>
<td>CN</td>
<td>2</td>
<td></td>
<td>pF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aperture Delay</td>
<td>tAP</td>
<td>4</td>
<td></td>
<td>ns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aperture Uncertainty (jitter)</td>
<td>tA0</td>
<td>2</td>
<td></td>
<td>ps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aperture Delay Match</td>
<td></td>
<td>2</td>
<td></td>
<td>ps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Bandwidth (-3 dB)</td>
<td>BW</td>
<td>240</td>
<td></td>
<td>MHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Power (0 dB)</td>
<td></td>
<td>245</td>
<td></td>
<td>MHz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Internal Reference

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Voltage (1 V Mode)</td>
<td>VRBF</td>
<td>1</td>
<td></td>
<td>V</td>
<td></td>
<td>REFSENSE = VREF</td>
</tr>
<tr>
<td>Output Voltage Tolerance (1 V Mode)</td>
<td>VRFB</td>
<td>2</td>
<td></td>
<td>mV</td>
<td></td>
<td>REFSENSE = GND</td>
</tr>
<tr>
<td>Output Voltage (2 V Mode)</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Voltage Tolerance (2 V Mode)</td>
<td></td>
<td>±15</td>
<td></td>
<td>mV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load Regulation (1 V Mode)</td>
<td></td>
<td>±10</td>
<td></td>
<td>% FS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load Regulation (2 V Mode)</td>
<td></td>
<td>±15</td>
<td></td>
<td>% FS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Power Supply

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>AVDD</td>
<td>2.7</td>
<td>3</td>
<td>3.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Supply Current</td>
<td>DRVDD</td>
<td>2.7</td>
<td>3</td>
<td>3.5</td>
<td>mA</td>
<td>AVDD = 3 V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>P0</td>
<td>215</td>
<td></td>
<td>245</td>
<td>mA</td>
<td>AVDD = 2.5 V</td>
</tr>
<tr>
<td>Power-Down</td>
<td></td>
<td>15.5</td>
<td></td>
<td>17.3</td>
<td>% FS</td>
<td></td>
</tr>
<tr>
<td>Power Supply Rejection</td>
<td>PSR</td>
<td>0.8</td>
<td></td>
<td>1.3</td>
<td>% FS</td>
<td></td>
</tr>
</tbody>
</table>

### Dynamic Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal-to-Noise and Distortion</td>
<td>SINAD</td>
<td>55.6</td>
<td>57.3</td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td></td>
<td>55.8</td>
<td></td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Harmonic Distortion</td>
<td>THD</td>
<td>55.9</td>
<td>57.4</td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spurious Free Dynamic Range</td>
<td>SFDR</td>
<td>56.2</td>
<td></td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two-Tone Intermodulation Distortion</td>
<td>IMD</td>
<td>-69</td>
<td>-63.3</td>
<td>dB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Phase</td>
<td>DP</td>
<td>0.1</td>
<td></td>
<td>% FS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Differential Gain</td>
<td>DG</td>
<td>0.6</td>
<td></td>
<td>% FS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cross-talk Rejection</td>
<td></td>
<td>68</td>
<td></td>
<td>dB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Notes

1. sinful = 3.58 MHz
2. IMD = 44.49 MHz and 45.52 MHz
3. NTSC 40 IRE Mod Ramp
4. REFT = 1 V, REFB = 0 V
5. REFT = 1 V, REFB = 0 V
6. 2 = 14.5 MHz
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNAMIC PERFORMANCE (SER)</td>
<td>SINAD</td>
<td>52.3</td>
<td>55.5</td>
<td>58</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Signal-to-Noise and Distortion</td>
<td>SNR</td>
<td>55.5</td>
<td></td>
<td></td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>f = 3.58 MHz</td>
<td>THD</td>
<td>-55</td>
<td></td>
<td></td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Signal-to-Noise</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f = 3.58 MHz</td>
<td>SFDR</td>
<td>-58</td>
<td></td>
<td></td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>Total Harmonic Distortion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f = 3.58 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spurious Free Dynamic Range</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>f = 3.58 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIGITAL INPUTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High Input Voltage</td>
<td>V_H</td>
<td>2.4</td>
<td>0.3</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Low Input Voltage</td>
<td>V_L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC Leakage Current</td>
<td>I_L</td>
<td>16</td>
<td></td>
<td></td>
<td>µA</td>
<td></td>
</tr>
<tr>
<td>Input Capacitance</td>
<td>C_IN</td>
<td>2</td>
<td></td>
<td></td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>LOGIC OUTPUT (with DVDD = 3 V)</td>
<td>V_OH</td>
<td>2.88</td>
<td></td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>High Level Output Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{LH} = 50 pA)</td>
<td>V_OL</td>
<td>0.095</td>
<td></td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Low Level Output Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOGIC OUTPUT (with DVDD = 5 V)</td>
<td>V_OH</td>
<td>4.5</td>
<td></td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>High Level Output Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{LH} = 50 pA)</td>
<td>V_OL</td>
<td>0.4</td>
<td></td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Low Level Output Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Valid Delay</td>
<td>t_{VD}</td>
<td>11</td>
<td></td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>MUX Select Delay</td>
<td>t_{MD}</td>
<td>7</td>
<td></td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Data Enable Delay</td>
<td>t_{ED}</td>
<td>13</td>
<td></td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Data High-Z Delay</td>
<td>t_{HZ}</td>
<td>13</td>
<td></td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>CLOCKING</td>
<td>t_{CH}</td>
<td>22.5</td>
<td></td>
<td></td>
<td>ns</td>
<td>C_{C} = 20 pF. Output Level to 90% of Final Value</td>
</tr>
<tr>
<td>Clock Pulsewidth High</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Pulsewidth Low</td>
<td>t_{CL}</td>
<td>22.5</td>
<td></td>
<td></td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>Pipeline Latency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOTES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 IMD: differential 2 V p-p, REFB = 1.5 V, REFH = -4.5 V.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 IMD: referred to lower of two input signals.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3S is single ended input, REFT = 1.5 V, REFB = -4.5 V.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifications subject to change without notice.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. ADC Timing
### AD9201

**ABSOLUTE MAXIMUM RATINGS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>With Respect to</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVDD</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>DVDD</td>
<td>DVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>AVSS</td>
<td>AVDD</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>AVDD</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>CLK</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>Digital Outputs</td>
<td>DVDD</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>AIN, AINB</td>
<td>AVSS</td>
<td>-1.0</td>
<td>0.3</td>
<td>V</td>
</tr>
<tr>
<td>VREF</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>REFSENSE</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>REFT, REFB</td>
<td>AVSS</td>
<td>-0.3</td>
<td>6.5</td>
<td>V</td>
</tr>
<tr>
<td>Junction Temperature</td>
<td></td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td></td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Lead Temperature 10 sec</td>
<td></td>
<td>+300</td>
<td></td>
<td>°C</td>
</tr>
</tbody>
</table>

*Some above values listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

### ORDERING GUIDE

<table>
<thead>
<tr>
<th>Model</th>
<th>Temperature Range</th>
<th>Package Description</th>
<th>Package Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD9201AR5</td>
<td>-40°C to +85°C</td>
<td>28-Lead SSOP</td>
<td>RS-38 Evaluation Board</td>
</tr>
<tr>
<td>AD9201-EVAL</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*RS = Stacked Small Outline.

### PIN CONFIGURATION

- **DVSS**
- **DVDD**
- **AVSS**
- **AVDD**
- **CLK**
- **Digital Outputs**
- **AIN, AINB**
- **VREF**
- **REFSENSE**
- **REFT, REFB**
- **Junction Temperature**
- **Storage Temperature**
- **Lead Temperature 10 sec**

### PIN FUNCTION DESCRIPTIONS

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DVSS</td>
<td>Digital Ground</td>
</tr>
<tr>
<td>2</td>
<td>DVDD</td>
<td>Digital Supply</td>
</tr>
<tr>
<td>3</td>
<td>AVSS</td>
<td>Bit (LSB)</td>
</tr>
<tr>
<td>4</td>
<td>AVDD</td>
<td>Bit 1</td>
</tr>
<tr>
<td>5</td>
<td>AIN</td>
<td>Bit 2</td>
</tr>
<tr>
<td>6</td>
<td>AOUT</td>
<td>Bit 3</td>
</tr>
<tr>
<td>7</td>
<td>AOUT</td>
<td>Bit 4</td>
</tr>
<tr>
<td>8</td>
<td>AOUT</td>
<td>Bit 5</td>
</tr>
<tr>
<td>9</td>
<td>AOUT</td>
<td>Bit 6</td>
</tr>
<tr>
<td>10</td>
<td>AOUT</td>
<td>Bit 7</td>
</tr>
<tr>
<td>11</td>
<td>AOUT</td>
<td>Bit 8</td>
</tr>
<tr>
<td>12</td>
<td>AOUT</td>
<td>Bit 9 (MSB)</td>
</tr>
<tr>
<td>13</td>
<td>SELECT</td>
<td>HI Channel Out, Lo Q Channel Out</td>
</tr>
<tr>
<td>14</td>
<td>CLOCK</td>
<td>Clock</td>
</tr>
<tr>
<td>15</td>
<td>SLEEP</td>
<td>HI Power Down, Lo Normal Operation</td>
</tr>
<tr>
<td>16</td>
<td>INA-I</td>
<td>Channel A, Input</td>
</tr>
<tr>
<td>17</td>
<td>INB-I</td>
<td>Channel B, Input</td>
</tr>
<tr>
<td>18</td>
<td>REFT-I</td>
<td>Top Reference Decoupling, I Channel</td>
</tr>
<tr>
<td>19</td>
<td>REFB-I</td>
<td>Bottom Reference Decoupling, I Channel</td>
</tr>
<tr>
<td>20</td>
<td>AVSS</td>
<td>Analog Ground</td>
</tr>
<tr>
<td>21</td>
<td>REFSENSE</td>
<td>Reference Select</td>
</tr>
<tr>
<td>22</td>
<td>VREF</td>
<td>Internal Reference Output</td>
</tr>
<tr>
<td>23</td>
<td>AVDD</td>
<td>Analog Supply</td>
</tr>
<tr>
<td>24</td>
<td>REFT-Q</td>
<td>Bottom Reference Decoupling, Q Channel</td>
</tr>
<tr>
<td>25</td>
<td>REFB-Q</td>
<td>Top Reference Decoupling, Q Channel</td>
</tr>
<tr>
<td>26</td>
<td>INB-Q</td>
<td>Q Channel, B Input</td>
</tr>
<tr>
<td>27</td>
<td>INA-Q</td>
<td>Q Channel, A Input</td>
</tr>
<tr>
<td>28</td>
<td>CHIP-SELECT</td>
<td>Hi-High Impedance, Lo-Normal Operation</td>
</tr>
</tbody>
</table>

### DEFINITIONS OF SPECIFICATIONS

**INTEGRAL NONLINEARITY (INL)**
Integral nonlinearity refers to the deviation from a line that is drawn through "zero" through "full scale." The point used as "zero" occurs 1/2 LSB before the first code transition. "Full scale" is defined as a level 1/2 LSBs beyond the last code transition. The deviation is measured from the center of each particular code to the true straight line.

**DIFFERENTIAL NONLINEARITY (DNL, NO MISSING CODES)**
An ideal ADC exhibits code transitions that are exactly 1 LSB apart. DNL is the deviation from this ideal value. It is often specified in terms of the resolution for which no missing codes (NMC) are guaranteed.

### CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD9201 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

---

WARNING:

Electrostatic Discharge (ESD) Sensitive Device.
OFFSET ERROR
The first transition should occur at a level 1 LSB above "zero." Offset is defined as the deviation of the actual first code transition from that point.

OFFSET MATCH
The change in offset error between I and Q channels.

EFFECTIVE NUMBER OF BITS (ENOB)
For a sine wave, SINAD can be expressed in terms of the number of bits. Using the following formula,
\[ N = \left(\frac{\text{SINAD}}{1.76}\right) \]
It is possible to get a measure of performance expressed as N, the effective number of bits.
Thus, effective number of bits for a device for sine wave inputs at a given input frequency can be calculated directly from its measured SINAD.

TOTAL HARMONIC DISTORTION (THD)
THD is the ratio of the rms sum of the first six harmonic components to the rms value of the measured input signal and is expressed as a percentage or in decibels.

SIGNAL-TO-NOISE RATIO (SNR)
SNR is the ratio of the rms value of the measured input signal to the rms sum of all other spectral components below the Nyquist frequency, excluding the first six harmonics and dc. The value for SNR is expressed in decibels.

SPURIOUS FREE DYNAMIC RANGE (SFDR)
The difference in dB between the rms amplitude of the input signal and the peak spurious signal.

GAIN ERROR
The first code transition should occur for an analog value 1 LSB above nominal negative full scale. The last transition should occur for an analog value 1 LSB below the nominal positive full scale. Gain error is the deviation of the actual difference between first and last code transitions and the ideal difference between the first and last code transitions.

GAIN MATCH
The change in gain error between I and Q channels.

PIPELINE DELAY (LATENCY)
The number of clock cycles between conversion initiation and the associated output data being made available. New output data is provided every rising clock edge.

MUX SELECT DELAY
The delay between the change in SELECT pin data level and valid data on output pins.

POWER SUPPLY REJECTION
The specification shows the maximum change in full scale from the value with the supply at its minimum limit to the value with the supply at its maximum limit.

APERTURE JITTER
Aperture jitter is the variation in aperture delay for successive samples and is manifested as noise on the input to the A/D.

APERTURE DELAY
Aperture delay is a measure of the Sample-and-Hold Amplifier (SHA) performance and is measured from the rising edge of the clock input to when the input signal is held for conversion.

SIGNAL-TO-NOISE AND DISTORTION (SN+D, SINAD) RATIO
SN+D is the ratio of the rms value of the measured input signal to the rms sum of all other spectral components below the Nyquist frequency, including harmonics but excluding dc. The value for SN+D is expressed in decibels.
AD9201—Typical Characteristic Curves

(AvDD = +3 V, Vinh = +3 V, f1 = 20 MHz (50% duty cycle), 2 V input span from –0.5 V to +1.5 V, 2 V internal reference unless otherwise noted)

Figure 3. Typical INL (1 V internal Reference)

Figure 4. Typical DNL (1 V internal Reference)

Figure 5. Input Bias Current vs. Input Voltage

Figure 6. SNR vs. Input Frequency

Figure 7. SINAD vs. Input Frequency

Figure 8. THD vs. Input Frequency
Figure 9. THD vs. Clock Frequency (f₀ = 1 MHz)

Figure 10. Voltage Reference Error vs. Temperature

Figure 11. Power Consumption vs. Clock Frequency

Figure 12. Grounded Input Histogram

Figure 13. Full Power Bandwidth

Figure 14. SNR vs. Input Frequency (Single Ended)
THEORY OF OPERATION

The AD9201 integrates two A/D converters, two analog input buffers, an internal reference and reference buffer, and an output multiplexer. For clarity, this data sheet refers to the two converters as 'I' and 'Q'. The two A/D converters simultaneously sample their respective inputs on the rising edge of the input clock. The two converters distribute the conversion operation over several smaller A/D subblocks, refining the conversion with progressively higher accuracy as it passes the result from stage to stage. As a consequence of the distributed conversion, each converter requires a small fraction of the 1023 comparators used in a traditional flash-type 10-bit ADC. A sample-and-hold function within each of the stages permits the first stage to operate on a new input sample while the following stages continue to process previous samples. This results in a "pipeline processing" latency of three clock periods between when an input sample is taken and when the corresponding ADC output is updated into the output registers.

The AD9201 integrates input buffer amplifiers to drive the analog inputs of the converters. In most applications, these input amplifiers eliminate the need for external op amps for the input signals. The input structure is fully differential, but the SHA common-mode response has been designed to allow the converter to readily accommodate either single-ended or differential input signals. This differential structure makes the part capable of accommodating a wide range of input signals.

The AD9201 also includes an on-chip bandgap reference and reference buffer. The reference buffer shifts the ground-referred reference to levels more suitable for use by the internal circuits of the converter. Both converters share the same reference and reference buffer. This scheme provides for the best possible gain match between the converters while simultaneously minimizing the channel-to-channel crosstalk. (See Figure 16.)

Each A/D converter has its own output latch, which updates on the rising edge of the input clock. A logic multiplexer, controlled through the SELECT pin, determines which channel is passed to the digital output pins. The output drivers have their own supply (DVDD), allowing the part to be interfaced to a variety of logic families. The outputs can be placed in a high impedance state using the CHIP SELECT pin.

The AD9201 has great flexibility in its supply voltage. The analog and digital supplies may be operated from 2.7 V to 5.5 V, independently of one another.

ANALOG INPUT

Figure 16 shows an equivalent circuit structure for the analog input of one of the A/D converters. PMOS source-followers buffer the analog input pins from the charge injection problems normally associated with switched capacitor ADC input structures. This produces a very high input impedance on the part, allowing it to be effectively driven from high impedance sources. This means that the AD9201 could even be driven directly by a passive antialias filter.

![Figure 16: Equivalent Circuit for AD9201 Analog Inputs](image)
AD9201

REFERENCE AND REFERENCE BUFFER

The reference and buffer circuitry on the AD9201 is configured for maximum convenience and flexibility. An illustration of the equivalent reference circuit is shown in Figure 24. The user can select from five different reference modes through appropriate pin-strapping (see Table I below). These pin strapping options cause the internal circuitry to reconfigure itself for the appropriate operating mode.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Input Span</th>
<th>REFSENSE Pin</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 V</td>
<td>1 V</td>
<td>VREF</td>
<td>22</td>
</tr>
<tr>
<td>2 V</td>
<td>2 V</td>
<td>AGND</td>
<td>23</td>
</tr>
<tr>
<td>Programmable</td>
<td>1 + (R1/R2)</td>
<td>See Figure 26</td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>= External Ref</td>
<td>AVDD</td>
<td>25</td>
</tr>
</tbody>
</table>

1 V Mode (Figure 22)—provides a 1 V reference and 1 V input full scale. Recommended for applications wishing to optimize high frequency performance, or any circuit on a supply voltage of less than 4 V. The part is placed in this mode by shorting the REFSENSE pin to the VREF pin.

3 V Mode (Figure 23)—provides a 2 V reference and 2 V input full scale. Recommended for noise sensitive applications on 5 V supplies. The part is placed in 2 V reference mode by grounding (shorting to AVSS) the REFSENSE pin.

Externally Set Voltage Mode (Figure 24)—this mode uses the on-chip reference, but scales the exact reference level through the use of an external resistor divider network. VREF is wired to the top of the network, with the REFSENSE wired to the tap point in the resistor divider. The reference level (and input full scale) will be equal to 1 V x (R1 + R2)/R1. This method can be used for voltage levels from 0.7 V to 2.5 V.

Reference Buffer—The reference buffer structure takes the voltage on the VREF pin and level-shifts and buffers it for use by various subblocks within the two A/D converters. The two converters share the same reference buffer amplifier to maintain the best possible gain match between the two converters. In the interest of minimizing high frequency crosstalk, the buffered references for the two converters are separately decoupled on the IREFB, IREFT, QREFB and QREFT pins, as illustrated in Figure 26.
AD9201

COMMON-MODE PERFORMANCE

Attention to the common-mode point of the analog input voltage can improve the performance of the AD9201. Figure 29 illustrates THD as a function of common-mode voltage (center point of the analog input span) and power supply.

Inspection of the curves will yield the following conclusions:

1. An AD9201 running with AVDD = 5 V is the easiest to drive.
2. Differential inputs are the most insensitive to common-mode voltage.
3. An AD9201 powered by AVDD = 3 V and a single ended input, should have a 1 V span with a common-mode voltage of 0.75 V.

Figure 29. THD vs. CML Input Span and Power Supply (Analog Input = 1 MHz)
DIGITAL INPUTS AND OUTPUTS
Each of the AD9201 digital control inputs, CHIP SELECT, CLOCK, SELECT and SLEEP are referenced to AVDD and AVSS. Switching thresholds will be AVDD/2.
The format of the digital inputs is straight binary. A low power mode feature is provided such that for STBY = HIGH and the clock disabled, the static power of the AD9201 will drop below 22 mW.

CLOCK INPUT
The AD9201 clock input is internally buffered with an inverter powered from the AVDD pin. This feature allows the AD9201 to accommodate either +5 V or +3.3 V CMOS logic input signal swings with the input threshold for the CLK pin nominally at AVDD/2.
The pipelined architecture of the AD9201 operates on both rising and falling edges of the input clock. To minimize duty cycle variations the logic family recommended to drive the clock input is high speed or advanced CMOS (HC/HCT, AC/ACT) logic. CMOS logic provides both symmetrical voltage threshold levels and sufficient rise and fall times to support 20 MSPS operation. Running the part at slightly faster clock rates may be possible, although at reduced performance levels. Conversely, some slight performance improvements might be realized by clocking the AD9201 at slower clock rates.
The power dissipated by the output buffers is largely proportional to the clock frequency; running at reduced clock rates provides a reduction in power consumption.

DIGITAL OUTPUTS
Each of the on-chip buffers for the AD9201 output bits (D0-D9) is powered from the DVDD supply pin, separate from AVDD. The output drivers are sized to handle a variety of logic families while minimizing the amount of glitch energy generated. In all cases, a fan-out of one is recommended to keep the capacitive load on the output data bits below the specified 20 pF level.

For DVDD = 5 V, the AD9201 output swing is compatible with high speed CMOS and TTL logic families. For TTL, the AD9201 on-chip, output drivers were designed to support several of the high speed TTL families (F, AS, S). For applications where the clock rate is below 20 MSPS, other TTL families may be appropriate. For interfacing with lower voltage CMOS logic, the AD9201 sustains 20 MSPS operation with DVDD = 3 V. In all cases, check your logic family data sheets for compatibility with the AD9201’s specification table.
A 2 ns reduction in output delays can be achieved by limiting the logic load to 5 pF per output line.

THREE-STATE OUTPUTS
The digital outputs of the AD9201 can be placed in a high impedance state by setting the CHI SELECT pin to HIGH. This feature is provided to facilitate in-circuit testing or evaluation.

APPLICATIONS USING THE AD9201 FOR QAM DEMODULATION
QAM is one of the most widely used digital modulation schemes in digital communication systems. This modulation technique can be found in both FDMA as well as spread spectrum (i.e., CDMA) based systems. A QAM signal is a carrier frequency which is both modulated in amplitude (i.e., AM modulation) and in phase (i.e., PM modulation). At the transmitter, it can be generated by independently modulating two carriers of identical frequency but with a 90° phase difference. This results in an inphase (I) carrier component and a quadrature (Q) carrier component at a 90° phase shift with respect to the I component. The I and Q components are then summed to provide a QAM signal at the specified carrier or IF frequency. Figure 31 shows a typical analog implementation of a QAM modulator using a dual 10-bit DAC with 2x interpolation, the AD9761. A QAM signal can also be synthesized in the digital domain thus requiring a single DAC to reconstruct the QAM signal. The AD9935 is an example of a complete (i.e., DAC included) digital QAM modulator.

Figure 30. Typical De-Mux Connection

Figure 31. Typical Analog QAM Modulator Architecture
AD9201

At the receiver, the demodulation of a QAM signal back into its separate I and Q components is essentially the modulation process explained above but in the reverse order. A common and traditional implementation of a QAM demodulator is shown in Figure 32. In this example, the demodulation is performed in the analog domain using a dual, matched ADC and a quadrature demodulator to recover and digitize the I and Q baseband signals. The quadrature demodulator is typically a single IC containing two mixers and the appropriate circuitry to generate the necessary 90° phase shift between the I and Q mixers' local oscillators. Before being digitized by the ADCs, the mixed down baseband I and Q signals are filtered using matched analog filters. These filters, often referred to as Nyquist or Pulse-Shaping filters, remove images from the mixing process and any out-of-band. The characteristics of the matching Nyquist filters are well defined to provide optimum signal-to-noise (SNR) performance while minimizing intersymbol interference. The ADC's are typically simultaneously sampling their respective inputs at the QAM symbol rate or, most often, at a multiple of it if a digital filter follows the ADC. Oversampling and the use of digital filtering eases the implementation and complexity of the analog filter. It also allows for enhanced digital processing for both carrier and symbol recovery and tuning purposes. The use of a dual ADC such as the AD9201 ensures excellent gain, offset, and phase matching between the I and Q channels.

These characteristics result in both a reduction of electromagnetic interference (EMI) and an overall improvement in performance.

It is important to design a layout that prevents noise from coupling onto the input signal. Digital signals should not be run in parallel with the input signal traces and should be routed away from the input circuitry. Separate analog and digital grounds should be joined together directly under the AD9201 to a solid ground plane. The power and ground return currents must be carefully managed. A general rule of thumb for mixed signal layouts dictates that the return currents from digital circuitry should not pass through critical analog circuitry. Transients between AVSS and DVSS will seriously degrade performance of the ADC.

If the user cannot tie analog ground and digital ground together at the ADC, he should consider the configuration in Figure 33.

GROUNDING AND LAYOUT RULES

As is the case for any high performance device, proper grounding and layout techniques are essential in achieving optimal performance. The analog and digital grounds on the AD9201 have been separated to optimize the management of return currents in a system. Grounds should be connected near the ADC. It is recommended that a printed circuit board (PCB) of at least four layers, employing a ground plane and power planes, be used with the AD9201. The use of ground and power planes offers distinct advantages:

1. The minimization of the loop area encompassed by a signal and its return path.
2. The minimization of the impedance associated with ground and power paths.
3. The inherent distributed capacity formed by the power plane, PCB insulation and ground plane.

Figure 32. Typical Analog QAM Demodulator

Figure 33. Ground and Power Consideration

Another layout and ground technique is shown in Figure 34. A separate ground plane has been split for RF or hard to manage signals. These signals can be routed to the ADC differentially or single ended (i.e., both can either be connected to the driver or RF ground). The ADC will perform well with several hundred mV of noise or signals between the RF and ADC analog ground.

Figure 34. RF Ground Scheme

---14---
The AD9201 evaluation board is shipped "ready to run."

Power and signal generators should be connected as shown in Figure 35. Then the user can observe the performance of the Q channel. If the user wants to observe the I channel, then he should install a jumper at JP22 Pins 1 and 2. If the user wants to toggle between I and Q channels, then a CMOS level pulse train should be applied to the "strobe" jack after appropriate jumper connections.
Figure 38. Evaluation Board Ground Plane Layout

Figure 39. Evaluation Board Solder-Side Layout
Figure 40. Evaluation Board Component-Side Silkscreen

Figure 41. Evaluation Board Power Plane Layout
AD9201

OUTLINE DIMENSIONS
Dimensions shown in inches and (mm).

21-Lead Shrink Small Outline Package (SSOP) (8-S-31)

[Diagram of the 21-Lead Shrink Small Outline Package (SSOP) (8-S-31)]

Author: Chong Pay Peng, Clarice
Dual 10-Bit TxDAC+™
with 2x Interpolation Filters
AD9761

FEATURES
- Complete 10-Bit, 40 MSPS Dual Transmit DAC
- Excellent Gain and Offset Matching
- Differential Nonlinearity Error: 0.5 LSB
- Effective Number of Bits: 20
- Signal-to-Noise and Distortion Ratio: 59 dB
- Spurious-Free Dynamic Range: 71 dB
- 2x Interpolation Filters
- 20 MSPS/Channel Data Rate
- Single Supply: +2.7 V to +5.5 V
- Low Power Dissipation: 260 mW (3.3 V Supply)
- On-Chip Reference
- 28-Lead SSOP

PRODUCT DESCRIPTION
The AD9761 is a complete dual channel, high speed, 10-bit CMOS DAC. The AD9761 has been developed specifically for use in wide bandwidth communication applications (e.g., spread spectrum) where digital I and Q information is being processed during transmit operations. It integrates two 10-bit, 40 MSPS DACs, dual 2x interpolation filters, a voltage reference, and digital input interface circuitry. The AD9761 supports a 2x MS PS per channel input data rate that is then interpolated by 2x up to 40 MSPS before simultaneously updating each DAC.

The interleaved I and Q input data streams are presented to the digital interface circuitry, which consists of I and Q latches as well as some additional control logic. The data is de-interleaved back into its original I and Q data. An on-chip state machine ensures the proper pairing of I and Q data. The data output from each latch is then processed by a 2x digital interpolation filter that eases the reconstruction filter requirements. The interpolated output of each filter serves as the input of their respective 10-bit DAC.

The DACs utilize a segmented current source architecture combined with a proprietary switching technique to reduce glitch energy and to maximize dynamic accuracy. Each DAC provides differential current output, thus supporting single-ended or differential applications. Both DACs are simultaneously updated and provide a nominal full-scale current of 10 mA. Also, the full-scale currents between each DAC are matched to within 0.07 dB (i.e., 0.75%), thus eliminating the need for additional gain calibration circuitry.

The AD9761 is manufactured on an advanced low cost CMOS process. It operates from a single supply of 2.7 V to 5.5 V and consumes 260 mW of power. To make the AD9761 complete it also offers an internal 1.20 V temperature-compensated bandgap reference.

TxDAC+ is a trademark of Analog Devices, Inc.

REV. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.
### AD9761—SPECIFICATIONS

#### DC SPECIFICATIONS *(TMIN to TMAX, AVDD = +5 V, DVDD = +5 V, IREF = 10 mA, unless otherwise noted)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOLUTION</td>
<td>10</td>
<td></td>
<td></td>
<td>Bits</td>
</tr>
<tr>
<td>DC ACCURACY1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integral Linearity Error (IINL)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMIN to TMAX</td>
<td>-1.75</td>
<td>±0.5</td>
<td>1.75</td>
<td>LSB</td>
</tr>
<tr>
<td>Gain Error (without Internal Reference)</td>
<td>-5.5</td>
<td>±1.0</td>
<td>5.5</td>
<td>% of FSR</td>
</tr>
<tr>
<td>Gain Matching between DACs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full-Scale Output Current</td>
<td>-1.0</td>
<td>±0.5</td>
<td>1.0</td>
<td>mA</td>
</tr>
<tr>
<td>Output Compliance Range</td>
<td>-1.0</td>
<td>±0.5</td>
<td>1.0</td>
<td>LSB</td>
</tr>
<tr>
<td>Output Capacitance</td>
<td>5</td>
<td></td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>REFERENCE OUTPUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference Voltage</td>
<td>1.14</td>
<td>1.20</td>
<td>1.26</td>
<td>V</td>
</tr>
<tr>
<td>Reference Output Current1</td>
<td>100</td>
<td></td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>REFERENCE INPUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input Compliance Range</td>
<td>0.1</td>
<td>1.25</td>
<td>1.25</td>
<td>V</td>
</tr>
<tr>
<td>Reference Input Resistance</td>
<td>1</td>
<td></td>
<td></td>
<td>Ohm</td>
</tr>
<tr>
<td>TEMPERATURE COEFFICIENTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unipolar Offset Drift</td>
<td>0</td>
<td></td>
<td></td>
<td>ppm/°C</td>
</tr>
<tr>
<td>Gain Drift (without Internal Reference)</td>
<td>±50</td>
<td></td>
<td></td>
<td>ppm/°C</td>
</tr>
<tr>
<td>Gain Drift (with Internal Reference)</td>
<td>±140</td>
<td></td>
<td></td>
<td>ppm/°C</td>
</tr>
<tr>
<td>Gain Matching Drift (between DACs)</td>
<td>±35</td>
<td></td>
<td></td>
<td>ppm/°C</td>
</tr>
<tr>
<td>Reference Voltage Drift</td>
<td>±50</td>
<td></td>
<td></td>
<td>ppm/°C</td>
</tr>
<tr>
<td>POWER SUPPLY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVDD</td>
<td>2.7</td>
<td>5.0</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Analog Supply Current (IAVDD)</td>
<td>25</td>
<td>35</td>
<td>35</td>
<td>mA</td>
</tr>
<tr>
<td>DVDD</td>
<td>2.7</td>
<td>5.0</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Digital Supply Current at 5 V (ISUPPLY)</td>
<td>70</td>
<td>85</td>
<td>85</td>
<td>mA</td>
</tr>
<tr>
<td>Digital Supply Current at 1 V (ISUPPLY)</td>
<td>35</td>
<td></td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td>Nominal Power Dissipation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVDD and DVDD at 5 V</td>
<td>200</td>
<td>250</td>
<td>250</td>
<td>mW</td>
</tr>
<tr>
<td>AVDD and DVDD at 1 V</td>
<td>500</td>
<td>650</td>
<td>650</td>
<td>mW</td>
</tr>
<tr>
<td>Power Supply Rejection Ratio (FSRR)-AVDD</td>
<td>-0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>% of FSRV</td>
</tr>
<tr>
<td>Power Supply Rejection Ratio (FSRR)-DVDD</td>
<td>-0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>% of FSRV</td>
</tr>
<tr>
<td>OPERATING RANGE</td>
<td>-40</td>
<td>485</td>
<td>485</td>
<td>°C</td>
</tr>
</tbody>
</table>

**NOTES**

1. Measured at IOUTA and QOUTA, driving a virtual ground.
2. Nominal full-scale current, IREF, is like the IOUT current.
3. Use an external amplifier to drive any external load.
4. Measured at 50 MHz and IREF = 10 mA.
5. Measured at 40 MHz and IOUT = 10 mA.

Specifications subject to change without notice.

---

**REV. A**
DYNAMIC SPECIFICATIONS

(Max to Min, AVDD = +5 V, DVDD = +5 V, I<sub>MIN</sub> = 10 mA, Differential Transformer Coupled Output, 50-ohm Doubly Terminated, unless otherwise noted)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DYNAMIC PERFORMANCE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum Output Update Rate</td>
<td>40</td>
<td>35</td>
<td></td>
<td>MSPS</td>
</tr>
<tr>
<td>Output Settling Time (t&lt;sub&gt;ST&lt;/sub&gt;) to 0.025%</td>
<td></td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Output Propagation Delay (t&lt;sub&gt;P&lt;/sub&gt;)</td>
<td>55</td>
<td></td>
<td></td>
<td>Input Clock Cycles</td>
</tr>
<tr>
<td>Glitch Impulse</td>
<td>5</td>
<td></td>
<td></td>
<td>pV-s</td>
</tr>
<tr>
<td>Output Rise Time (10% to 90%)</td>
<td>2.5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Output Fall Time (10% to 90%)</td>
<td>2.5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td><strong>AC LINEARITY TO NYQUIST</strong></td>
<td></td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Signal-to-Noise and Distortion (SINAD)</td>
<td>56</td>
<td>59</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Effective Number of Bits (ENOBs)</td>
<td>9.0</td>
<td>9.5</td>
<td></td>
<td>Bits</td>
</tr>
<tr>
<td>Total Harmonic Distortion (THD)</td>
<td>-68</td>
<td>-58</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Spectral-Free Dynamic Range (SFDR)</td>
<td>59</td>
<td>68</td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Channel Isolation</td>
<td></td>
<td></td>
<td></td>
<td>dB</td>
</tr>
</tbody>
</table>

DIGITAL SPECIFICATIONS

(Max to Min, AVDD = +5 V, DVDD = +5 V, I<sub>MIN</sub> = 10 mA unless otherwise noted)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DIGITAL INPUTS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic &quot;1&quot; Voltage @ DVDD = +5 V</td>
<td>3.5</td>
<td>5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Logic &quot;1&quot; Voltage @ DVDD = +3 V</td>
<td>2.4</td>
<td>3</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Logic &quot;0&quot; Voltage @ DVDD = +5 V</td>
<td>0</td>
<td>1.3</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Logic &quot;0&quot; Voltage @ DVDD = +3 V</td>
<td>0</td>
<td>0.9</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Logic &quot;1&quot; Current</td>
<td>-10</td>
<td></td>
<td>+10</td>
<td>pA</td>
</tr>
<tr>
<td>Logic &quot;0&quot; Current</td>
<td>-10</td>
<td></td>
<td>+10</td>
<td>pA</td>
</tr>
<tr>
<td>Input Capacitance</td>
<td>5</td>
<td></td>
<td></td>
<td>pF</td>
</tr>
<tr>
<td>Input Setup Time (t&lt;sub&gt;U&lt;/sub&gt;)</td>
<td>3</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Input Hold Time (t&lt;sub&gt;H&lt;/sub&gt;)</td>
<td>2</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CLOCK High</td>
<td>5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>CLOCK Low</td>
<td>5</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>Invalid CLOCK/ WRITE Window (t&lt;sub&gt;W&lt;/sub&gt;v)</td>
<td>1</td>
<td></td>
<td>5</td>
<td>ns</td>
</tr>
</tbody>
</table>

**NOTES**

1. The "1" data window of 4 ns duration beginning 1 ns AFTERT the leading edge of WRITE in which the leading edge of CLOCK MUST NOT occur.

Specifications subject to change without notice.

![Figure 1. Timing Diagram](image-url)

Author: Chong Pay Peng, Clarice
AD9761

DIGITAL FILTER SPECIFICATIONS

Parameter

MAXIMUM INPUT CLOCK RATE (f_cLOCK)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIGITAL FILTER CHARACTERISTICS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passband Width:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.005 dB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passband Width:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01 dB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passband Width:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1.0 dB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Phase (FIR implementation)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stopband Rejection: 0.3 f_cLOCK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Group Delay</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 Input clock cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impulse Response Duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 Input clock cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES

1 ESD protection of DAC.
2 Defined as the number of data clock cycles between input and peak of output response.
33 Input clock periods from input to DAC, 36 to Q DAC. Propagation delay is delay from data input to DAC output.

Table 1. Integer Filter Coefficients for 43-Tap Halfband FIR Filter

<table>
<thead>
<tr>
<th>Lower Coefficient</th>
<th>Upper Coefficient</th>
<th>Integer Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>H(1)</td>
<td>H(43)</td>
<td>1</td>
</tr>
<tr>
<td>H(2)</td>
<td>H(42)</td>
<td>0</td>
</tr>
<tr>
<td>H(3)</td>
<td>H(41)</td>
<td>-3</td>
</tr>
<tr>
<td>H(4)</td>
<td>H(40)</td>
<td>0</td>
</tr>
<tr>
<td>H(5)</td>
<td>H(39)</td>
<td>8</td>
</tr>
<tr>
<td>H(6)</td>
<td>H(38)</td>
<td>0</td>
</tr>
<tr>
<td>H(7)</td>
<td>H(37)</td>
<td>-16</td>
</tr>
<tr>
<td>H(8)</td>
<td>H(36)</td>
<td>0</td>
</tr>
<tr>
<td>H(9)</td>
<td>H(35)</td>
<td>28</td>
</tr>
<tr>
<td>H(10)</td>
<td>H(34)</td>
<td>0</td>
</tr>
<tr>
<td>H(11)</td>
<td>H(33)</td>
<td>-50</td>
</tr>
<tr>
<td>H(12)</td>
<td>H(32)</td>
<td>0</td>
</tr>
<tr>
<td>H(13)</td>
<td>H(31)</td>
<td>81</td>
</tr>
<tr>
<td>H(14)</td>
<td>H(30)</td>
<td>0</td>
</tr>
<tr>
<td>H(15)</td>
<td>H(29)</td>
<td>-191</td>
</tr>
<tr>
<td>H(16)</td>
<td>H(28)</td>
<td>0</td>
</tr>
<tr>
<td>H(17)</td>
<td>H(27)</td>
<td>216</td>
</tr>
<tr>
<td>H(18)</td>
<td>H(26)</td>
<td>0</td>
</tr>
<tr>
<td>H(19)</td>
<td>H(25)</td>
<td>-400</td>
</tr>
<tr>
<td>H(20)</td>
<td>H(24)</td>
<td>0</td>
</tr>
<tr>
<td>H(21)</td>
<td>H(23)</td>
<td>1264</td>
</tr>
<tr>
<td>H(22)</td>
<td></td>
<td>1998</td>
</tr>
</tbody>
</table>

Figure 2a. FIR Filter Frequency Response

Figure 2b. FIR Filter Impulse Response
AD9761

ORDERING GUIDE

<table>
<thead>
<tr>
<th>Model</th>
<th>Package Description</th>
<th>Package Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD9761ARS</td>
<td>28-Lead Shrink Small Outline (SSOP)</td>
<td>RS-28</td>
</tr>
<tr>
<td>AD9761-EB</td>
<td>Evaluation Board</td>
<td></td>
</tr>
</tbody>
</table>

THERMAL CHARACTERISTICS

Thermal Resistance
28-Lead SSOP
$R_{th}=109^\circ C/W$

ABSOLUTE MAXIMUM RATINGS*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>With Respect to</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVDD</td>
<td>ACOM</td>
<td>-0.3</td>
<td>+6.5</td>
<td>V</td>
</tr>
<tr>
<td>DVDD</td>
<td>DCOM</td>
<td>-0.3</td>
<td>+6.5</td>
<td>V</td>
</tr>
<tr>
<td>AGCOM</td>
<td>DCOM</td>
<td>-0.3</td>
<td>+0.3</td>
<td>V</td>
</tr>
<tr>
<td>AVDD</td>
<td>DVDD</td>
<td>-6.5</td>
<td>+6.5</td>
<td>V</td>
</tr>
<tr>
<td>CLOCK, WRITE</td>
<td>DCOM</td>
<td>-0.3</td>
<td>DVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>SELECT, SLEEP</td>
<td>DCOM</td>
<td>-0.3</td>
<td>DVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>Digital Inputs</td>
<td>DCOM</td>
<td>-0.3</td>
<td>DVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>IOUTA, IOUTB</td>
<td>ACOM</td>
<td>-1.0</td>
<td>AVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>GOUTA, GOUTB</td>
<td>ACOM</td>
<td>-1.0</td>
<td>AVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>COMP1, COMP2</td>
<td>ACOM</td>
<td>-0.3</td>
<td>AVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>COMP3</td>
<td>ACOM</td>
<td>-0.3</td>
<td>AVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>REPOF, FSADJ</td>
<td>ACOM</td>
<td>-0.3</td>
<td>AVDD+0.3</td>
<td>V</td>
</tr>
<tr>
<td>REFO</td>
<td>ACOM</td>
<td>-0.3</td>
<td>+0.3</td>
<td>V</td>
</tr>
<tr>
<td>Junction Temperature</td>
<td></td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td></td>
<td>-65</td>
<td>+150</td>
<td>°C</td>
</tr>
<tr>
<td>Lead Temperature (10 sec)</td>
<td></td>
<td>-65</td>
<td>+300</td>
<td>°C</td>
</tr>
</tbody>
</table>

*This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Figure 3. Basic AC Characterization Test Setup

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the AD9761 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

WARNING!

REV. A

Author: Chong Pay Peng, Clarice
PIN FUNCTION DESCRIPTIONS

<table>
<thead>
<tr>
<th>Pin No.</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DB9</td>
<td>Most Significant Data Bit (MSB).</td>
</tr>
<tr>
<td>2-9</td>
<td>DB8-DB1</td>
<td>Data Bits 1-8.</td>
</tr>
<tr>
<td>10</td>
<td>DB0</td>
<td>Least Significant Data Bit (LSB).</td>
</tr>
<tr>
<td>11</td>
<td>CLOCK</td>
<td>Clock Input. Both DACs’ outputs updated on positive edge of clock and digital filters read respective input registers.</td>
</tr>
<tr>
<td>12</td>
<td>WRITE</td>
<td>Write Input. DAC input registers latched on positive edge of write.</td>
</tr>
<tr>
<td>13</td>
<td>SELECT</td>
<td>Select Input. Select high routes input data to I DAC, select low routes data to Q DAC.</td>
</tr>
<tr>
<td>14</td>
<td>DVDD</td>
<td>Digital Supply Voltage (+2.7 V to +5.5 V).</td>
</tr>
<tr>
<td>15</td>
<td>DCOM</td>
<td>Digital Common.</td>
</tr>
<tr>
<td>16</td>
<td>COMP3</td>
<td>Internal Bias Node for Switch Driver Circuitry. Decouple to ACOM with 0.1 µF capacitor.</td>
</tr>
<tr>
<td>17</td>
<td>QOUTA</td>
<td>Q DAC Current Output. Full-scale current when all data bits are 1s.</td>
</tr>
<tr>
<td>18</td>
<td>QOUTB</td>
<td>Q DAC Complementary Current Output. Full-scale current when all data bits are 0s.</td>
</tr>
<tr>
<td>19</td>
<td>REFLO</td>
<td>Reference Ground when Internal 1.2 V Reference Used. Connect to AVDD to disable internal reference.</td>
</tr>
<tr>
<td>20</td>
<td>REFIO</td>
<td>Reference Input/Output. Serves as reference input when internal reference disabled. Serves as 1.2 V reference output when internal reference activated. Requires 0.1 µF capacitor to ACOM when internal reference activated.</td>
</tr>
<tr>
<td>21</td>
<td>FSADJ</td>
<td>Full-Scale Current Output Adjust. Resistance to ACOM acts full-scale output current.</td>
</tr>
<tr>
<td>23</td>
<td>AVDD</td>
<td>Analog Supply Voltage (+2.7 V to +5.5 V).</td>
</tr>
<tr>
<td>24</td>
<td>ACOM</td>
<td>Analog Common.</td>
</tr>
<tr>
<td>25</td>
<td>IOUTA</td>
<td>I DAC Current Output. Full-scale current when all data bits are 0s.</td>
</tr>
<tr>
<td>26</td>
<td>COMP1</td>
<td>Internal Bias Node for Switch Driver Circuitry. Decouple to AGND with 0.1 µF capacitor.</td>
</tr>
<tr>
<td>27</td>
<td>RESET/SLEEP</td>
<td>Power-Down control input if asserted for four clock cycles or longer. Reset control input if asserted for less than four clock cycles. Active high. Connect to DCOM if not used. Refer to RESET/SLEEP section.</td>
</tr>
</tbody>
</table>

PIN CONFIGURATION

![Pin Configuration Diagram]

Author: Chong Pay Peng, Clarice
DEFINITIONS OF SPECIFICATIONS

Linearity Error (Also Called Integral Nonlinearity or INL)
Linearity error is defined as the maximum deviation of the actual analog output from the ideal output, determined by a straight line drawn from zero to full scale.

Differential Nonlinearity (or DNL)
DNL is the measure of the variation in analog value, normalized to full scale, associated with a 1 LSB change in digital input code.

Monotonicity
A DA converter is monotonic if the output either increases or remains constant as the digital input increases.

Offset Error
The deviation of the output current from the ideal channel output current associated with all digital inputs set to Os. For IOUT, 0 mA output is expected when all inputs are set to 1s.

Gain Error
The difference between the actual and ideal output span. The actual span is determined by the output when all inputs are set to 1s minus the output when all inputs are set to Os.

Output Compliance Range
The range of allowable voltage at the output of a current-output DAC. Operation beyond the maximum compliance limits may cause either output stage saturation or breakdown, resulting in nonlinear performance.

Temperature Drift
Temperature drift is specified as the maximum change from the ambient (±25°C) value to the value at either TMIN or TMAX. For offset and gain drift, the drift is reported in ppm of full-scale range (FSR) per °C. For reference drift, the drift is reported in ppm per °C.

Power Supply Rejection
The maximum change in the full-scale output as the supplies are varied from nominal to minimum and maximum specified voltages.

Settling Time
The time required for the output to reach and remain within a specified error band after its final value, measured from the start of the output transition.

Glitch Impulse
Asymmetrical switching times in a DAC give rise to undesired output transients that are quantified by a glitch impulse. It is specified as the net area of the glitch in pV-s.

Channel Isolation
Channel Isolation is a measure of the level of crosstalk between channels. It is measured by producing a full-scale 8 MHz signal output for one channel and measuring the leakage into the other channel.

Spurious-Free Dynamic Range
The difference, in dB, between the rms amplitude of the output signal and the peak spurious signal over the specified bandwidth.

Total Harmonic Distortion
THD is the ratio of the sum of the rms value of the first six harmonic components to the rms value of the measured output signal. It is expressed as a percentage or in decibels (dB).

Signal-to-Noise and Distortion (SIN+D, SNR/NR) Ratio
S/N+D is the ratio of the rms value of the measured output signal to the rms sum of all other spectral components below the Nyquist frequency, including harmonics but excluding dc. The value for S/N+D is expressed in decibels.

Effective Number of Bits (ENOB)
For a sine wave, SINAD can be expressed in terms of the number of bits. Using the following formula,

\[
N = \frac{\text{SINAD}}{\text{SINAD} - 1.76)} \times 0.02
\]

it is possible to get a measure of performance expressed as N, the effective number of bits.

Thus, effective number of bits for a device for sine wave inputs at a given input frequency can be calculated directly from its measured SINAD.

Passband
Frequency band in which any input applied therein passes unattenuated to the DAC output.

Stopband Rejection
The amount of attenuation of a frequency outside the passband applied to the DAC, relative to a full-scale signal applied at the DAC input within the passband.

Group Delay
Number of input clocks between an impulse applied at the device input and peak DAC output current.

Impulse Response
Response of the device to an impulse applied to the input.
AD9761

Typical AC Characterization Curves @ +5 V Supplies

AVA = +5 V, BVDD = +5 V, 50 Ω Doubly Terminated Load, TA = +25°C, fDATA = 40 MSPS, unless otherwise noted, worst of I or Q output performance shown.

Figure 4. Single-Tone SFDR (DC to
2 fDATA, fCLOCK = 2 fDATA)

Figure 5. SINAD (ENOBs) vs. fOUT (DC to fDATA/2)

Figure 6. SFDR vs. fOUT (DC to fDATA/2)

Figure 7. "Out-of-Band" SFDR vs. fOUT (fDATA/2 to 3/2 fDATA)

Figure 8. SINAD vs. AOUT (DC to fDATA/2, Differential Output)

Figure 9. SINAD vs. AOUT (DC to fDATA/2, Single-Ended Output)

Figure 10. SINAD/SFDR vs. fOUTs (DC to fDATA/2, Differential Output)

Figure 11. SINAD/SFDR vs. fOUTs (DC to fDATA/2, Single-Ended Output)

Figure 12. Wideband Spread-Spectrum Spectral Plot (DC to fDATA)
Typical AC Characterization Curves @ +3 V Supplies

(AVDD = +3 V, DVDD = +3 V, 50 Ω Doubly Terminated Load, T_A = +25°C, f_CLOCK = 10 MSPS, unless otherwise noted, worst of 1 or Q output performance shown)

Figure 13. Single-Tone SFDR (DC to f_DATA/2, f_CLOCK = 2 f_DATA)

Figure 14. SINAD (ENOB) vs. f_OUT (DC to f_DATA/2)

Figure 15. SFDR vs. f_OUT (DC to f_DATA/2)

Figure 16. "Out-of-Band" SFDR vs. f_OUT (f_DATA/2 to 3 f_DATA)

Figure 17. SINAD vs. A_OUT (DC to f_DATA/2, Differential Output)

Figure 18. SINAD vs. A_OUT (DC to f_DATA/2, Single-Ended Output)

Figure 19. SINAD/SFDR vs. i_LIMs (DC to f_DATA/2, Differential Output)

Figure 20. SINAD/SFDR vs. i_LIMs (DC to f_DATA/2, Single-Ended Output)

Figure 21. Narrowband Spread-Spectrum Spectral Plot (DC to f_DATA)

REV. A
AD9761

FUNCTIONAL DESCRIPTION

Figure 22 shows a simplified block diagram of the AD9761. The AD9761 is a complete dual channel, high speed, 10-bit CMOS DAC capable of operating up to a 40 MHz clock rate. It has been optimized for the transmit section of wideband communication systems employing I and Q modulation schemes. Excellent matching characteristics between channels reduces the need for any external calibration circuitry. Dual matching 2x interpolation filters included in the I and Q data path simplify any post, bandlimiting filter requirements. The AD9761 interfaces with a single 10-bit digital input bus that supports interleaved I and Q input data.

![Block Diagram of AD9761](image)

Figure 22. Dual DAC Functional Block Diagram

Referring to Figure 22, the AD9761 consists of an analog section and a digital section. The analog section includes matched I and Q 10-bit DACs, ±1.22 V bandgap voltage reference and a reference control amplifier. The digital section includes: two 2x interpolation filters; segment decoding logic; and some additional digital input interface circuitry. The analog and digital sections of the AD9761 have separate power supply inputs (i.e., AVDD and DVDD) that can operate over a 2.7 V to 5.5 V range.

Each DAC consists of a large PMOS current source array capable of providing up to 10 mA of full-scale current, IOUT. Each array is divided into 15 equal currents that make up the four most significant bits (MSBs). The next four bits or middle bits consist of 15 equal current sources whose values are 1/16th of an MSB current source. The remaining LSBs are binary weighted fractions of the middle-bits current sources. All of these current sources are switched to one or the other of two output nodes (i.e., IOUTA or IOUTB) via PMOS differential current switches.

The full-scale output current, IOUTP, of each DAC is regulated from the same voltage reference and control amplifier, thus ensuring excellent gain matching and drift characteristics between DACs. IOUTP can be set from 1 mA to 10 mA via an external resistor, RREF. The external resistor in combination with both the reference control amplifier and voltage reference, VREF, sets the reference current, IREF, which is mirrored over to the segmented current sources with the proper scaling factor. IOUTP is exactly sixteen times the value of IREF.

The I and Q DACs are simultaneously updated on the rising edge of CLOCK with digital data from their respective 2x digital interpolation filters. The 2x interpolation filters essentially multiplies the input data rate of each DAC by a factor of two, relative to the original input data rate while simultaneously reducing the magnitude of first image associated with the DAC's original input data rate. Since the AD9761 supports a single 10-bit digital bus with interleaved I and Q input data, the original I and Q input data rate before interpolation is one-half the CLOCK rate. After interpolation, the data rate in each I and Q DAC becomes equal to the CLOCK rate.

The benefits of an interpolation filter are clearly seen in Figure 23, which shows an example of the frequency and time domain representation of a discrete time sine wave signal before and after it is applied to a digital interpolation filter. Images of the sine wave signal appear around multiples of the DAC's input data rate as predicted by the sampling theory. These undesired images will also appear at the output of a reconstruction DAC, although modified by the DAC's sin(x)/x response. In many bandlimited applications, these images must be suppressed by an analog filter following the DAC. The complexity of this analog filter is typically determined by the proximity of the desired fundamental to the first image and the required amount of image suppression.

![Time and Frequency Domain Example of Digital Interpolation Filter](image)

Figure 23. Time and Frequency Domain Example of Digital Interpolation Filter

-10-
Refering to Figure 23, the "new" first image associated with the DAC's higher data rate after interpolation is "pushed" out further relative to the input signal. The "old" first image associated with the lower DAC data rate before interpolation is suppressed by the digital filter. As a result, the resolution band for the analog reconstruction filter is increased that reducing the complexity of the analog filter.

The digital interpolation filters for I and Q paths are identical 43-tap halfband symmetrical FIR filters. Each filter section de-interleaves I or Q data from the digital input interface. The input CLOCK signal is internally divided by two to generate the filter clock. The filters are implemented with two parallel paths running at the filter clock rate. The output from each path is selected on opposite phases of the filter clock, thus producing interpolated filtered output data at the input clock rate. The frequency response and impulse response of these filters are shown in Figures 2a and 2b. Table I lists the idealized filter coefficients that correspond to the filter's impulse response.

The digital section of the AD9761 also includes an input interface section designed to support interleaved I and Q input data from a single 16-bit bus. This section de-interleaves the I and Q input data while ensuring its proper pairing for the 2x interpolation filters. A SLEEP/RESET input serves an dual function by providing a reset function for this section as well as providing power down functionality. Refer to the DIGITAL INPUT AND INTERFACE CONSIDERATIONS and SLEEP/RESET sections for a more detailed discussion.

**DAC TRANSFER FUNCTION**

Each I and Q DAC provides complementary current output pins: IOUT(A/B) and QOUT(A/B) respectively. Note, QOUTA and QOUTB operate individually for IOUTA and IOUTB. IOUTA will provide a near full-scale current output, IQOUTA, when all bits are high (i.e., DAC CODE = 1023) while IOUTB, the complementary output, provides no current. The current output of IOUTA and IOUTB are a function of both the input code and IQOUT and can be expressed as:

\[ I_{\text{OUT}} = (\text{DAC CODE}/1024) \times I_{\text{OUT}} \]

where:

\[ \text{DAC CODE} = 0 \text{ to } 1023 \] (i.e., Decimal Representation).

As previously mentioned, IQOUTA is a function of the reference current, [\text{REF}], which is nonlinearly set by a reference, [\text{VRFB}], and external resistor, [\text{R}\text{LOAD}]. It can be expressed as:

\[ I_{\text{OUT}} = 16 \times I_{\text{REF}} \]

where:

\[ I_{\text{REF}} = \text{VRFB} / R_{\text{LOAD}} \]

The two current outputs will typically drive a resistive load directly or via a transformer. If dc coupling is required, IOUTA and IOUTB should be directly connected to matching negative leads, R\text{LOAD}, which are tied to analog common, ACOM. Note, R\text{LOAD} represents the equivalent load resistance seen by IOUTA or IOUTB. The single-ended voltage output appearing at IOUTA and IOUTB pins is simply:

\[ V_{\text{IOUTA}} = I_{\text{IOUTA}} \times R_{\text{LOAD}} \]

\[ V_{\text{IOUTB}} = I_{\text{IOUTB}} \times R_{\text{LOAD}} \]

Note, the full-scale values of V\text{IOUTA} and V\text{IOUTB} should not exceed the specified output compliance range from REFIO to IOUT.

The differential voltages, V\text{DIFF}, appearing across IOUTA and IOUTB is:

\[ V_{\text{DIFF}} = V_{\text{IOUTA}} - V_{\text{IOUTB}} \times R_{\text{LOAD}} \]

Substituting the values of I\text{OUTA}, I\text{OUTB}, and \text{REFIO} V\text{DIFF} can be expressed as:

\[ V_{\text{DIFF}} = (12 \times \text{DAC CODE} - 1023) \times \frac{16 \times \text{RLOAD} \times \text{VRFB}}{2 \times 1023} \]

These two equations highlight some of the advantages of using the AD9761 differentially. First, differential operation will help cancel common-mode error sources associated with I\text{OUTA} and I\text{OUTB} such as noise and distortion. Second, the differential code dependent current and subsequent voltage, V\text{DIFF}, is twice the value of the single-ended voltage output (i.e., V\text{IOUTA} or V\text{IOUTB}) thus providing twice the signal power to the load.

**REFERENCE OPERATION**

The AD9761 contains an internal 1.20 V bandgap reference which can be easily disabled and overridden by an external reference. REFIO serves as either an input or output depending on whether the internal or an external reference is selected. If REFIO is tied to ACOM as shown in Figure 24, the internal reference is activated and REFIO provides a 1.20 V output. In this case, the internal reference must be filtered externally with a ceramic chip capacitor of 0.1 µF or greater (since the internal reference is disabled and the high input impedance (i.e., 1 MΩ) of REFIO minimizes any loading of the external reference.)

![Figure 24. Internal Reference Configuration](image)

The internal reference can also be disabled by connecting REFIO to AVDD. In this case, an external reference may then be applied to REFIO as shown in Figure 25. The external reference may provide either a fixed reference voltage to enhance accuracy and drift performance or a varying reference voltage for gain control. Note that the 0.1 µF compensation capacitor is not required since the internal reference is disabled and the high input impedance (i.e., 1 MΩ) of REFIO minimizes any loading of the external reference.)
The control amplifier allows a wide control range, by varying the REFD voltage. IREF can be varied for a fixed RFB by disabling the internal reference and varying the voltage of REFD over its compliance range of 1.25 V to 0.10 V. REFD can be driven by a single-supply amplifier or DAC thus allowing IREF to be varied for a fixed RFB. Since the input impedance of REFD is approximately 1 MΩ, a simple, low-cost R-2R ladder DAC configured in the voltage mode topology may be used to control the gain. This circuit is shown in Figure 26 using the AD7524 and an external 1.2 V reference, the AD1580.

ANALOG OUTPUTS
As previously stated, both the I and Q DACs produce two complementary current outputs which may be configured for single-end or differential operation. IOUTA and IOUTB can be converted into complementary single-ended voltage outputs, VOUTA and VOUTB, via a load resistor, RLOAD, as described in the DAC TRANSFER SECTION by Equations 5 through 8. The differential voltage, VOUT, existing between VOUTA and VOUTB can also be converted to a single-ended voltage via a transformer or differential amplifier configuration.

Figure 27 shows an equivalent circuit of the AD9761's I (or Q) DAC output. It consists of a parallel array of PMOS current sources in which each current source is switched to either IOUTA or IOUTB. As a result, the equivalent output impedance of IOUTA and IOUTB remains quite high (i.e., >100 kΩ and 5 pF).

**Figure 27. Equivalent Circuit of the AD9761 DAC Output**

IOUTA and IOUTB have a negative and positive voltage compliance range which must be adhered to achieve optimum performance. The negative output compliance range of -1 V is set by the breakdown limits of the CMOS process. Operation beyond this maximum limit may result in a breakdown of the output stage.
The positive output compliance range is slightly dependent on the full-scale output current, I_{OUT}. It degrades slightly from its nominal 2.5 V for an input of 10 mA to 1.09 V for an input of 5 mA. Applications requiring the AD9761's output (i.e., V_{OUT} and/or V_{INREF}) to extend to its output compliance range should scale V_{CLAMP} accordingly. Operation beyond this compliance range will adversely affect the AD9761's linearity performance and subsequently degrade its distortion performance. Note, the optimum distortion performance of the AD9761 is obtained by restricting its output(s) as seen at IO\_TOUT(A\_B) and QOUT(A\_B) to within ±0.5 V.

**DIGITAL INPUTS AND INTERLEAVED INTERFACE CONSIDERATIONS**

The AD9761 digital interface consists of 10 data input pins, a clock input pin, and three control pins. It is designed to support a clock rate up to 60 MSPS. The 10-bit parallel data inputs follow standard positive binary coding, where D9 is the most significant bit (MSB) and D0 is the least significant bit (LSB). IO\_TOUT (or QOUT) produces a full-scale output current, while all data bits are at Logic 1. IO\_TOUT (or QOUT) produces a complementary output, with the full-scale current split between the two outputs as a function of the input code.

![Figure 28. Block Diagram of Digital Interface](image)

The AD9761 integrates with a single 10-bit digital input that supports interleaved I and Q input data. Figure 28 shows a simplified block diagram of the digital interface circuitry consisting of two banks of edge triggered registers, two multiplexers, and a state machine. Interleaved I and Q input data is presented to the DATA input bus, where it is then latched into the selected I or Q input register on the rising edge of the WRITE input. The output of these input registers is transferred in pairs to their respective interpolator filters' registers after each Q write or the rising edge of the CLOCK input (refer to Timing Diagram in Figure 2). A state machine ensures the proper pairing of I and Q input data to the interpolation filters' inputs.

The SELECT signal at the time of the rising edge of the WRITE signal determines which input register latches the input data. If SELECT is high around the rising edge of WRITE the data is latched into the I register of the AD9761. If SELECT is low around the rising edge of WRITE, the data is latched into the Q register of the AD9761. If SELECT is kept in one state while data is repeatedly written to the AD9761, the data will be written into the selected filter register at half the input data rate since the data is always assumed to be interleaved.

The state machine controls the generation of the divided clock and hence pairing of I and Q data inputs. After the AD9761 is reset, the state machine keeps track of the paired I and Q data. The state transition diagram is shown in Figure 29, in which all the states are defined. A transition in state occurs upon the rising edge of CLOCK and is a function of the current state and whether the parallel output is selected or deselected. The state machine is reset on the first rising edge of CLOCK, and it is reinitialized from the initial state. The state machine will advance to either the SELECT or QOUT state. The state machine will advance to the ONE-I state upon writing I data followed by a clock.

![Figure 29. State Transition Diagram of AD9761 Digital Interface](image)

An example helps illustrate the digital timing and control requirements to ensure proper pairing of I and Q data. In this example, the AD9761 is assumed to interface with a host processor on a dedicated data bus and the state machine is reset by asserting a Logic Level "1" to the RESET/SLEEP input for a duration of one clock cycle. In the timing diagram shown in Figure 24, WRITE and CLOCK are tied together while SELECT is updated at the same instance as DATA. Since SELECT is high upon RESET returing low, I data is latched into the I register on the first rising WRITE. On the next rising WRITE edge, the Q data is latched into the Q input register and the outputs of both input registers are latched into their respective I and Q filter registers. The sequence of events is repeated on the next rising WRITE edge with the new I data being latched into the I input register.

The digital inputs are CMOS compatible with logic thresholds, V_{THREN} and V_{THREN} = V_{DD} (20%).

The internal digital circuitry of the AD9761 is capable of operating over a digital supply range of 2.7 V to 5.5 V. As a result, the digital inputs can also accommodate TTL levels when V_{DD} is set to accommodate the maximum high level voltage, V_{HLMAT} of the TTL drivers. A V_{DD} of 3 V to 5.5 V will typically...
ensure proper compatibility of most TTL logic families. Figure 31 shows the equivalent digital input circuit for the data, sleep and clock inputs.

**Figure 30. Timing Diagram**

![Timing Diagram](image)

**Figure 31. Equivalent Digital Input**

Since the AD9761 is capable of being updated up to 40 MSPS, the quality of the clock and data input signals are important in achieving the optimum performance. The drivers of the digital data interface circuitry should be specified to meet the minimum rise and high levels of the AD9761 as well as its required minimum input logic level thresholds. The external clock driver circuitry should provide the AD9761 with a low jitter clock input meeting the minimum logic level while providing fast edges. Fast clock edges will help minimize any jitter that can manifest itself as phase noise on a reconstructed waveform.

Digital signal paths should be kept short, and run lengths matched to avoid propagation delay mismatch. The insertion of low value resistor network (i.e., 20 Ω to 100 Ω) between the AD9761 digital inputs and driver outputs may be helpful in reducing any overshoot and ringing at the digital inputs, which contributes to data feedthrough. Operating the AD9761 with reduced logic swings and a corresponding digital supply (DVDD) will also reduce data feedthrough.

**RESET/SLEEP MODE OPERATION**

The RESET/SLEEP input can be used either to power-down the AD9761 or reset its internal digital interface logic. If the RESET/SLEEP input is asserted for greater than one clock cycle but under four clock cycles by applying a logic level “1,” the internal state machine will be reset. If the RESET/SLEEP input is asserted for four clock cycles or longer, the power-down function of the AD9761 will be initiated. The power-down function turns off the output current and reduces the supply current to less than 9 mA over the specified supply range of 2.7 V to 5.5 V and temperature range.

The power-up and power-down characteristics of the AD9761 is dependent upon the value of the compensation capacitor connected to COMP1 and COMP3. With a nominal value of 0.1 μF, the AD9761 takes less than 5 μs to power down and approximately 325 ms to power back up.

**POWER DISSIPATION**

The power dissipation of the AD9761 is dependent on several factors which include: (1) AVDD and DVDD, the power supply voltages; (2) I_{DCLOCK}, the full-scale current output; (3) f_{CLK}, the update rate; (4) and the reconstructed digital input waveform. The power dissipation is directly proportional to the supply voltage current, I_{DCOV}, and the digital supply current, I_{DD}. I_{DCOV} is directly proportional to I_{DCLOCK}, as shown in Figure 32 and is insensitive to f_{CLK}.

Conversely, I_{DD} is dependent on both the digital input waveform, f_{DCLOCK}, and digital supply DVDD. Figures 33 and 34 show I_{DD} as a function of a full-scale sine wave output ratio's I_{DCOV} vs. I_{DCLOCK} for various update rates with DVDD = 5 V and DVDD = 3 V respectively.

**Figure 32. I_{DCOV} vs. I_{DCLOCK}**

![I_{DCOV} vs. I_{DCLOCK}](image)

**Figure 33. I_{DD} vs. Ratio @ DVDD = 5 V**

![I_{DD} vs. Ratio](image)
APPLYING THE AD9761

OUTPUT CONFIGURATIONS

The following sections illustrate some typical output configurations for the AD9761. Unless otherwise noted, it is assumed that IOFF is set to a nominal 10 mA. For applications requiring the optimum dynamic performance, a differential output configuration is suggested. A differential output configuration may consist of either an RF transformer or a differential op amp configuration. The transformer configuration provides the optimum high-frequency performance and is recommended for any application allowing for ac coupling. The differential op amp configuration is suitable for applications requiring dc coupling, a bipolar output, signal gain, and/or level shifting.

A single-ended output is suitable for applications requiring a unipolar voltage output. A positive unipolar output voltage will result if IOUTA and IOUTB are connected to AC output. This configuration may be more suitable for a single-supply system requiring a dc-coupled, ground-referred output voltage. Alternatively, an amplifier could be configured as an inverting converter thus converting IOUTA or IOUTB into a negative unipolar voltage. This configuration provides the best dc linearity since IOUTA or IOUTB is maintained at a virtual ground.

DIFFERENTIAL COUPLING USING A TRANSFORMER

An RF transformer can be used to perform a differential-to-single-ended signal conversion as shown in Figure 35. A differentially coupled transformer output provides the optimum distortion performance for output signals whose spectral content lies within the transformers passband. An RF transformer such as the Mini Circuits T1-1T provides excellent rejection of common-mode distortion (i.e., even-order harmonics) and noise over a wide frequency range. It also provides electrical isolation and the ability to deliver twice the power to the load. Transformers with differential impedance ratios may also be used for impedance matching purposes. Note that the transformer provides an output only.

DIFFERENTIAL USING AN OP AMP

An op amp can also be used to perform a differential to single-ended conversion as shown in Figure 36. The AD9761 is configured with two equal load resistors, RLOAD, of 50 Ω. The differential voltage developed across IOUTA and IOUTB is connected to a single-ended signal via the differential op amp configuration. An optional capacitor can be inserted across IOUTA and IOUTB forming a real pole in a low-pass filter. The addition of this capacitor also enhances the op amp distortion performance by preventing the DAC's high slewing output from overloading the op amp's input.

Figure 35. Differential Output Using a Transformer

The center-tap on the primary side of the transformer must be connected to ACOM to provide the necessary dc output path for both IOUTA and IOUTB. The complementary voltages appearing at IOUTA and IOUTB (i.e., VOUTA and VOUTB) are summed symmetrically around ACOM and should be maintained within the specified output compliance range of the AD9761. A differential output voltage, VDIFF, may be inserted in applications in which the output of the transformer is connected to the load, RLOAD, via a passive reconstruction filter or cable requiring double termination. RDIFF is determined by the transformer's impedance ratio and provides the proper source termination which results in a low VSWR. Note that approximately half the signal power will be dissipated across RDIFF.

Figure 36. DC Differential Coupling Using an Op Amp

The common-mode rejection of this configuration is typically determined by the resistor matching. In this circuit, the differential op amp circuit using the AD8042 is configured to provide some additional signal gain. The op amp must operate from a dual supply since its input is approximately 11.0 V. A high-speed amplifier capable of preserving the differential performance of the AD9761 while meeting other system level objectives (i.e., cost, power) should be selected. The op amp differential gain, its gain setting resistor values, and full-scale output swing capabilities should all be considered when optimizing this circuit.
AD9761

The differential circuit shown in Figure 37 provides the necessary level-shifting required in a single supply system. In this case, AVDD, which is the positive analog supply for both the AD9761 and the op amp, is used to level-shift the differential output of the AD9761 to midsupply (i.e., AVDD/2).

Figure 37, Single-Supply DC Differential Coupled Circuit

SINGLE-ENDED UNBUFFERED VOLTAGE OUTPUT

Figure 38 shows the AD9761 configured to provide a unipolar output range of approximately 0 V to +0.5 V since the nominal full-scale current, IOUT, of 10 mA flows through an RLOAD of 50 Ω. In the case of a double-ended output, RLOAD represents the equivalent load resistance seen by IOUTA or IOUTB. The unused output (IOUTA or IOUTB) can be connected to AGND directly or via a matching RLOAD. Different values of IOUT and RLOAD can be selected as long as the positive compliance range is adhered to.

Figure 38, 0 V to +0.5 V Unbuffered Voltage Output

DIFFERENTIAL, DC COUPLED OUTPUT CONFIGURATION WITH LEVEL SHIFTING

Some applications may require the AD9761 differential outputs to interface to a single-supply quadrature upconverter. Although most of these devices provide differential inputs, their common-mode voltage range does not typically extend to ground. As a result, the ground-referred output signals shown in Figure 38 must be level shifted to within the specified common-mode range of the single-supply quadrature upconverter. Figure 39 shows the addition of a resistor pull-up network which provides the level shifting function. The use of matched resistor networks will maintain maximum gain matching and minimum offset performance between the I and Q channels. Note, the resistor pull-up network will introduce approximately 6 dB of signal attenuation.

POWER AND GROUNDING CONSIDERATIONS

In systems seeking to simultaneously achieve high speed and high performance, the implementation and construction of the printed circuit board design is often as important as the circuit design. Proper RF techniques must be used in device selection, placement and routing, and supply bypassing and grounding.

The evaluation board for the AD9761, which uses a four-layer PCB, serves as a good example for the above mentioned considerations. The evaluation board provides an illustration of the recommended printed circuit board ground, power and signal plane layout.

Proper grounding and decoupling should be a primary objective in any high-speed, high-resolution system. The AD9761 features separate analog and digital supply and ground pins to optimize the management of analog and digital ground currents in a system. In general, AVDD, the analog supply, should be decoupled to AGND, the analog common, as close to the chip as physically possible. Similarly, DVDD, the digital supply should be decoupled as closely as physically possible to DCM.

For those applications requiring a single ±5 V or ±3 V supply for both the analog and digital supply, a clean analog supply may be generated using the circuit shown in Figure 40. The circuit consists of a differential LC filter with separate power supply and return lines. Lower noise can be attained using low ESR type electrolytic and tantalum capacitors.

Figure 39, Differential, DC Coupled Output Configuration with Level-Shifting

Figure 40, Differential LC Filter for Single ±5 V or ±3 V Applications
Maintaining low noise on power supplies and ground is critical to obtaining optimum results from the AD9761. If properly implemented, ground planes can perform a host of functions on high-speed circuit boards: biasing, decoupling, current transport, etc. In mixed signal design, the analog and digital portions of the board should be distinct from each other, with the analog ground plane confined to the area covering the analog signal traces and the digital ground plane confined to areas covering the digital interconnects.

All analog ground pins of the DAC, reference and other analog components should be tied directly to the analog ground plane. The two ground planes should be connected by a path 1/8 to 1/4 inch wide underneath, or within 1/2 inch of the DAC to maintain optimum performance. Care should be taken to ensure that the ground plane is uninterrupted over crucial signal paths. On the digital side, this includes the digital input lines running to the DAC as well as any clock signals. On the analog side, this includes the DAC output signal, reference signal and the supply feeders.

The use of wide runs or planes in the routing of power lines is also recommended. This serves the dual role of providing a low series impedance power supply to the part, as well as providing some "free" capacitive decoupling to the appropriate ground plane. It is essential that care be taken in the layout of signal and power ground interconnects to avoid inducing extraneous voltage drops in the signal ground paths. It is recommended that all connections be short, direct and as physically close to the package as possible, in order to minimize the sharing of conductive paths between different currents. When runs exceed an inch in length, strip line techniques with proper termination resistor should be considered. The form and value of this resistor will be dependent upon the logic family used.

For a more detailed discussion of the implementation and construction of high speed, mixed signal printed circuit boards, refer to Analog Devices' application notes AN-380 and AN-133.

APPLICATIONS

Using the AD9761 for QAM Modulation

QAM is one of the most widely used digital modulation schemes in digital communication systems. This modulation technique can be found in both FDM as well as spread spectrum (i.e., CDMA) based systems. A QAM signal is a carrier frequency that is modulated both in amplitude (i.e., AM modulation) and in phase (i.e., FM modulation). It can be generated by independently modulating two carriers of identical frequency but with a 90° phase difference. This results in an in-phase (I) carrier component and a quadrature (Q) carrier component at a 90° phase shift with respect to the I component. The I and Q components are then summed to provide a QAM signal at the specified carrier frequency.

Figure 41. Typical Analog QAM Architecture

A common and traditional implementation of a QAM modulator is shown in Figure 41. The modulation is performed in the analog domain in which two DACs are used to generate the baseband I and Q components, respectively. Each component is then typically applied to a Nyquist filter before being applied to a quadrature mixer. The matching Nyquist filters shapes and limits each component's spectral envelope while minimizing intersymbol interference. The DAC is typically updated at the QAM symbol rate or possibly a multiple of it if an interpolating filter precedes the DAC. The use of an interpolating filter typically eases the implementation and complexity of the analog filter which can be a significant contributor to mismatches in gain and phase between the two baseband channels. A quadrature mixer modulates the I and Q components with in-phase and quadrature phase carrier frequency and then sums the two outputs to provide the QAM signal.

Figure 41. Typical Analog QAM Architecture

EVALUATION BOARD

The AD9761-EV is an evaluation board for the AD9761 dual 18-bit, 40 MSPS DAC. Careful attention to layout and circuit design along with prototyping area, allows the user to easily and effectively evaluate the AD9761. This board allows the user the flexibility to operate each of the AD9761 DACs in a single-ended or differential output configuration. Each of the DAC’s single-ended outputs are terminated in a 50 Ω resistor. Evaluation using a transformer coupled output can be accomplished simply by installing a Mini-circuit transformer (i.e., Model T5-17) into the available socket.

The digital inputs are designed to be driven directly from various word generators with the onboard option to add a resistor network for proper load termination. Separate 50 Ω terminated SMA connectors are also provided for the CLOCK, WRITE and SELECT inputs. Provisions are also made to operate the AD9761 with either the internal or an external reference as well as to exercise the power-down feature.
Figure 42a. Evaluation Board Schematic
Figure 42b. Evaluation Board Schematic
Figure 43. Silkscreen Layer—Top

Figure 44. Component Side PCB Layout (Layer 1)
Figure 45. Ground Plane PCB Layout (Layer 2)

Figure 46. Power Plane PCB Layout (Layer 3)
Figure 47. Solder Side PCB Layout (Layer 4)

Figure 48. Silkscreen Layer—Bottom

Author: Chong Pay Peng, Clarice

REV. A
OUTLINE DIMENSIONS

Dimensions shown in inches and (mm).

28-Lead Shrink Small Outline Package (SSOP)

(RS-23)

---

REV. A

23
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

General Description
The MAX2450 combines a quadrature modulator and quadrature demodulator with a supporting oscillator and divide-by-8 prescaler on a monolithic IC. It operates from a single +3V supply and draws only 5.9mA. The demodulator accepts an amplified and filtered IF signal in the 35MHz to 80MHz range, and demodulates it into I and Q baseband signals with 51dB of voltage conversion gain. The IF input is terminated with a 400Ω thin-film resistor for matching to an external IF filter. The baseband outputs are fully differential and have 1.2Vp-p signal swings. The modulator accepts differential I and Q baseband signals with amplitudes up to 1.35Vp-p and bandwidths to 15MHz, and produces a differential IF signal in the 35MHz to 80MHz range.

Pulling the CMOS-compatible ENABLE pin low shuts down the MAX2450 and reduces the supply current to less than 1μA. To minimize spurious feedback, the MAX2450's internal oscillator is set at twice the IF via external tuning components. The oscillator and associated phase shifters produce differential signals exhibiting low amplitude and phase imbalance, yielding modulator sideband rejection of 38dB. The MAX2450 comes in a CSOP package.

Features
- Combines Quadrature Modulator and Demodulator
- Integrated Quadrature Phase Shifters
- On-Chip Oscillator (Requires External Tuning Circuit)
- On-Chip Divide-by-8 Prescaler
- Modulator Input Bandwidth Up to 15MHz
- Demodulator Output Bandwidth Up to 9MHz
- 51dB Demodulator Voltage Conversion Gain
- CMOS-Compatible Enable
- 5.9mA Operating Supply Current
- 1μA Shutdown Supply Current

Applications
Digital Cordless Phones
GSM and North American Cellular Phones
Wireless LANs
Digital Communications
Two-Way Pagers

Pin Configuration

Ordering Information

<table>
<thead>
<tr>
<th>PART</th>
<th>TEMP. RANGE</th>
<th>PIN-PACKAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX2450CEP</td>
<td>0°C to +70°C</td>
<td>20 QSOP</td>
</tr>
</tbody>
</table>

Functional Diagram

For free samples & the latest literature: http://www.maxim-ic.com, or phone 1-800-998-8800.
For small orders, phone 1-800-835-8769.
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

ABSOLUTE MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage Range</td>
<td>VCC, LO_Vcc</td>
<td></td>
<td>-3.3V to +3.3V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supply Current</td>
<td>Icc (O)</td>
<td>ENABLE = 0.4V</td>
<td>2</td>
<td>20</td>
<td></td>
<td>µA</td>
</tr>
<tr>
<td>Shutdown Supply Current</td>
<td>Is (O)</td>
<td>ENABLE = 0.4V</td>
<td>10</td>
<td></td>
<td></td>
<td>µs</td>
</tr>
<tr>
<td>Enable/Disable Time</td>
<td>IENA</td>
<td>ENABLE = VCC</td>
<td>1</td>
<td>3</td>
<td></td>
<td>µA</td>
</tr>
<tr>
<td>ENABLE Bias Current</td>
<td>IEN</td>
<td>ENABLE = VCC</td>
<td>0.4</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>ENABLE Low Voltage</td>
<td>VENH</td>
<td>VCC - 0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALL- SIGNAL-LEVEL Voltage</td>
<td>V1_IN, Q_IN</td>
<td>VCC - IN</td>
<td>1.25</td>
<td>1.5</td>
<td>1.75</td>
<td>V</td>
</tr>
<tr>
<td>Demodulator Differential Input Impedance</td>
<td>Zd (IN)</td>
<td>Zd (OUT)</td>
<td>35</td>
<td>44</td>
<td></td>
<td>kΩ</td>
</tr>
<tr>
<td>Demodulator I and O Baseband DC Offset</td>
<td>Vq (IN)</td>
<td>VCC - 1.5</td>
<td>320</td>
<td>450</td>
<td>480</td>
<td>Ω</td>
</tr>
<tr>
<td>Demodulator I and O Baseband AC Offset</td>
<td>Vq (IN)</td>
<td>VCC - 1.5</td>
<td></td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>DEMODULATOR DC and O Voltage Swing</td>
<td>VSw</td>
<td>VCC - 0.1</td>
<td>1.35</td>
<td></td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>Voltage Conversion Gain</td>
<td>VCONV</td>
<td></td>
<td>51</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Allowable I and Q Voltage Swing</td>
<td>Vps</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Noise Figure</td>
<td>Vn</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>I and Q IM3 Level</td>
<td>IM3 (O)</td>
<td>(Note 2)</td>
<td>-44</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>I and Q IM5 Level</td>
<td>IM5 (O)</td>
<td>(Note 2)</td>
<td>-60</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>I and Q Signal 3dB Bandwidth</td>
<td>BW3 (O)</td>
<td></td>
<td>9</td>
<td></td>
<td></td>
<td>MHz</td>
</tr>
<tr>
<td>Oscillator Frequency Range</td>
<td>fLO</td>
<td>(Notes 1, 3)</td>
<td>70</td>
<td></td>
<td></td>
<td>MHz</td>
</tr>
<tr>
<td>LO Phase Noise</td>
<td>LO offset</td>
<td></td>
<td>80</td>
<td></td>
<td></td>
<td>dB/Hz</td>
</tr>
<tr>
<td>PRE_OUT Output Voltage</td>
<td>VOUT</td>
<td></td>
<td>0.35</td>
<td></td>
<td></td>
<td>Vps</td>
</tr>
<tr>
<td>PRE_OUT Slew Rate</td>
<td>SR (OUT)</td>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td>Vps</td>
</tr>
</tbody>
</table>

DC ELECTRICAL CHARACTERISTICS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Conditions</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMODULATOR I and O Amplitude Balance</td>
<td>A (I)</td>
<td></td>
<td>0.45</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>DEMODULATOR I and Q Phase Accuracy</td>
<td>Q (I)</td>
<td></td>
<td>±1.3</td>
<td></td>
<td></td>
<td>degrees</td>
</tr>
<tr>
<td>DEMODULATOR Voltage Conversion Gain</td>
<td>VCONV</td>
<td></td>
<td>51</td>
<td></td>
<td></td>
<td>dB</td>
</tr>
<tr>
<td>Oscillator Frequency Range</td>
<td>fLO</td>
<td>(Notes 1, 3)</td>
<td>70</td>
<td></td>
<td></td>
<td>MHz</td>
</tr>
<tr>
<td>LO Phase Noise</td>
<td>LO offset</td>
<td></td>
<td>80</td>
<td></td>
<td></td>
<td>dB/Hz</td>
</tr>
<tr>
<td>PRE_OUT Output Voltage</td>
<td>VOUT</td>
<td></td>
<td>0.35</td>
<td></td>
<td></td>
<td>Vps</td>
</tr>
<tr>
<td>PRE_OUT Slew Rate</td>
<td>SR (OUT)</td>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td>Vps</td>
</tr>
</tbody>
</table>

MAX2450

Author: Chong Pay Peng, Clarice
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

AC ELECTRICAL CHARACTERISTICS (continued)

(MAX2450 EV kit. VCC = 1.0 V, fO = 140MHz, fIF = 2 fC/MMIG = 1.2Vpp, fr_IN = 70.1MHz, Vp_IN = 2.82Vpp, TA = +25°C, unless otherwise noted.)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>CONDITIONS</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULATOR</td>
<td>Allowable Differential Input Voltage</td>
<td>Vp_IN/OUT</td>
<td>(Note 1)</td>
<td>1.35</td>
<td>Vpp</td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>Input Common-Mode Voltage Range</td>
<td>IN/OUT</td>
<td>1.25</td>
<td>1.75</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>IF Differential Output Voltage</td>
<td>Vp_OUT/OUT</td>
<td>Vp_IN/OUT, Vp_IN/OUT = 1.2Vpp, RL = 200Ω differential, Co = 1pF differential</td>
<td>65</td>
<td>mVpp</td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>IF Output IM3 Level</td>
<td>IM3L</td>
<td>Vp_IN/OUT = 1.35Vpp composite (Note 4)</td>
<td>-60</td>
<td>dBc</td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>IF Output IM5 Level</td>
<td>IM5L</td>
<td>Vp_IN/OUT = 1.35Vpp composite (Note 4)</td>
<td>-50</td>
<td>dBc</td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>Sideband Rejection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODULATOR</td>
<td>Carrier Suppression at Modulator Output</td>
<td></td>
<td></td>
<td>-35</td>
<td>dBc</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: Guaranteed by design, not tested.
Note 2: fr_IN = 2 tones at 70.10MHz and 70.11MHz. Vp_IN = 1.41mVpp per tone.
Note 3: The frequency range can be extended in either direction, but has not been characterized. At higher frequencies, the modulator IF output amplitude may decrease and distortions may increase.
Note 4: Vp_IN/OUT ports are terminated. Vp_IN/OUT = 2 tones at 550kHz and 600kHz.

Typical Operating Characteristics

(MAX2450 EV kit. VCC = 1.0 V, fO = 140MHz, fIF = 2 fC/MMIG = 1.2Vpp, fr_IN = 70.1MHz, Vp_IN = 2.82Vpp, TA = +25°C, unless otherwise noted.)

Author: Chong Pay Peng, Clarice
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

Typical Operating Characteristics (continued)

(MAX2450 EV Kit, VCC = 3.0V, VEE = ENABLE = 3.0V, LO = 140MHz, IF = VQIN/2 = 600kHz. VIN = 1.2Vp-p.

MODULATOR IF OUTPUT vs. SUPPLY VOLTAGE

MODULATOR IF OUTPUT vs. TEMPERATURE

MODULATOR SIDEHANDLE REJECTION vs. IF FREQUENCY

MODULATOR SIDEHANDLE REJECTION vs. TEMPERATURE

CARRIER SUPPRESSION vs. IF FREQUENCY

PRE-OUT WAVEFORM

MODULATOR OUTPUT SPECTRUM

Author: Chong Pay Peng, Clarice
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

Typical Operating Characteristics (continued)

(MAX2450 Evaluation Kit, VDD = 3.0V, LO = ENABLE = 3.0V, fosc = 140MHz, f1c = f2c = 500kHz, VIF = 2.82mVpeak-to-peak, 1.2Vp-p, fIF = 70.1MHz, VIF1 = 2.82mVp-p, TA = +25°C, unless otherwise noted.)

**Demodulator Voltage Conversion Gain vs. Temperature and Supply Voltage**

**Demodulator Voltage Conversion Gain vs. IF Frequency**

**Demodulator Intermodulation Power vs. Temperature**

**Demodulator IQ Phase and Amplitude Mismatch vs. Temperature**
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

<table>
<thead>
<tr>
<th>PIN</th>
<th>NAME</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IF_OUT</td>
<td>Modulator IF Output</td>
</tr>
<tr>
<td>2</td>
<td>IF_OUT</td>
<td>Modulator IF Inverting Output</td>
</tr>
<tr>
<td>3, 19</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>4</td>
<td>I_IN</td>
<td>Baseband Inphase Input</td>
</tr>
<tr>
<td>5</td>
<td>Q_IN</td>
<td>Baseband Inverting Input</td>
</tr>
<tr>
<td>6</td>
<td>Q_IN</td>
<td>Baseband Quadrature Input</td>
</tr>
<tr>
<td>7</td>
<td>I_IN</td>
<td>Baseband Quadrature Inverting Input</td>
</tr>
<tr>
<td>8</td>
<td>ENABLE</td>
<td>Enable Control, active High</td>
</tr>
<tr>
<td>9</td>
<td>PRE_OUT</td>
<td>Local-Oscillator, Divide-by-8, Prescaled Output</td>
</tr>
<tr>
<td>10</td>
<td>LO_Vcc</td>
<td>Local-Oscillator Supply. Bypass separately from Vcc</td>
</tr>
<tr>
<td>11</td>
<td>TANK</td>
<td>Local-Oscillator Resonant Tank Input (Figure 4)</td>
</tr>
<tr>
<td>12</td>
<td>TANK</td>
<td>Local-Oscillator Resonant Tank Inverting Input (Figure 4)</td>
</tr>
<tr>
<td>13</td>
<td>LO_GND</td>
<td>Local-Oscillator Ground</td>
</tr>
<tr>
<td>14</td>
<td>Q_OUT</td>
<td>Demodulator Quadrature Inverting Output</td>
</tr>
<tr>
<td>15</td>
<td>Q_OUT</td>
<td>Demodulator Quadrature Output</td>
</tr>
<tr>
<td>16</td>
<td>I_OUT</td>
<td>Demodulator Inphase Output</td>
</tr>
<tr>
<td>17</td>
<td>I_OUT</td>
<td>Demodulator Inphase Output</td>
</tr>
<tr>
<td>18</td>
<td>Vcc</td>
<td>Modulator and Demodulator Supply</td>
</tr>
<tr>
<td>19</td>
<td>IF_IN</td>
<td>Demodulator IF Input</td>
</tr>
</tbody>
</table>

Figure 1. Typical Application Block Diagram

Author: Chong Pay Peng, Clarice
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

Detailed Description

The following sections describe each of the functional blocks shown in the Functional Diagram. They also refer to the Typical Application Block Diagram (Figure 1).

Demodulator

The demodulator contains a single-ended to differential converter, two Gilbert-cell multipliers, and two fixed gain stages. The IF signal should be AC coupled into IF_IN, and provides a gain of 14dB. This amplified IF signal is fed into the I and Q mixers for demodulation. The multipliers mix the IF signal with the quadrature LO signals, resulting in baseband I and Q signals. The conversion gain of the multipliers is 15dB. These signals are further amplified by 21dB by the baseband amplifiers. The baseband I and Q amplifier chains are DC coupled.

Local Oscillator

The local-oscillator section is formed by an emitter-coupled differential pair. Figure 2 shows the equivalent local-oscillator circuit schematic. An external LC resonant tank determines the oscillation frequency, and the Q of this resonant tank affects the oscillator phase noise. The oscillation frequency is twice the IF frequency, so that the quadrature phase generator can use two latches to generate precise quadrature signals.

The oscillator may be driven by an external source. The source should be AC coupled into TANK/TANK, and should provide 200mVp-p levels. A choke (typically 2.2μH) is required between TANK and TANK. Differential input impedance at TANK/TANK is 10kΩ. For single-ended drive, connect an AC bypass capacitor (1000pF) from TANK to GND, and AC couple TANK to the source.

Quadrature Phase Generator

The quadrature phase generator uses two latches to divide the local-oscillator frequency by two, and generates two precise quadrature signals. Internal limiting amplifiers shape the signals to approximate square waves to drive the Gilbert-cell mixers. The inphase signal (at half the local-oscillator frequency) is further divided by four for the prescaler output.

Prescaler

The prescaler output, PRE_OUT, is buffered and swings typically 0.35Vp-p with a 10kΩ and 6pF load. It can be AC-coupled to the input of a frequency synthesizer.

Modulator

The modulator accepts I and Q differential baseband signals up to 1.35Vp-p with frequencies up to 15MHz, and upconverts them to the IF frequency. Since these inputs are biased internally at around 1.5V, I and Q signals should be capacitively coupled into these high-impedance ports (the differential input impedance is approximately 44kΩ). The self-bias design yields very low on-chip offset, resulting in excellent carrier sup-
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

MAX2450

Ultra-Low-Power Quadrature Modulator/Demodulator

Alternatively, a differential DAC may be connected without AC coupling, as long as a common-mode voltage range of 1.25V to 1.75V is maintained. For single-ended drive, connect UN and QIN via AC-coupling capacitors (0.1µF) to GND.

The IF output is designed to drive a high impedance (> 20kΩ), such as an IF buffer or an upconverter mixer. IF_OUT/IF_OUT must be AC coupled to the load. Impedances as low as 200Ω can be driven with a decrease in output amplitude (Figure 3). To drive a single-ended load, AC couple and terminate IF_OUT with a resistive load equal to the load at IF_OUT.

Master Bias

During normal operation, ENABLE should remain above VCC - 0.4V. Pulling the ENABLE input low shuts off the master bias and reduces the circuit current to less than 2mA. The master bias section includes a bandgap reference generator and a PTAT (Proportional To Absolute Temperature) current generator.

Applications Information

Figure 4 shows the implementation of a resonant tank circuit. The inductor, two capacitors, and a dual varactor form the oscillator's resonant circuit. In Figure 4, the oscillator frequency ranges from 130MHz to 160MHz.

To ensure reliable start-up, the inductor is directly connected across the local oscillator's tank ports. The two 33pF capacitors affect the Q of the resonant circuit. Other values may be chosen to meet individual application requirements. Use the following formula to determine the oscillation frequency:

\[ f_0 = \frac{2\pi}{2\sqrt{L_{EQ}C_{EQ}}} \]

where

\[ C_{EQ} = \frac{1}{C_1 + \frac{1}{C_2 + \frac{2}{C_{VAR}}}} + C_{STRAY} \]

and

\[ L_{EQ} = L + L_{STRAY} \]

where CSTRAY = parasitic capacitance and LSTRAY = parasitic inductance.

To alter the oscillation frequency range, change the inductance, the capacitance, or both. For best phase-noise performance keep the Q of the resonant tank as high as possible:

\[ Q = \frac{R_{EQ}}{L_{EQ}} \]

where \( R_{EQ} = 10kΩ \) (Figure 2).

The oscillation frequency can be changed by altering the control voltage, VCTRL.

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patents licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7800
© 1998 Maxim Integrated Products Printed USA MAXIM is a registered trademark of Maxim Integrated Products.
The Motorola 10BASE-T transceiver, designed to comply with the ISO 8802-3 [IEEE 802.3] 10BASE-T specification, will support a Medium Dependent Interface (MDI) in an embedded Media Attachment Unit (MAU)*. The interface supporting the Data Terminal Equipment (DTE) is TTL, CMOS, and raised ECL compatible, and the interface to the Twisted Pair (TP) media is supported through standard 10BASE-T filters and transformers. Differential data intended for the TP media is provided a 50 ns pre-emphasis and data at the TP receiver is screened by Smart Squelch circuitry for specific threshold, pulse width, and sequence requirements.

Other features of the MC34055 include: Collision and Jabber detection status outputs, select mode pins for forcing Loop Back and Full-Duplex operation, a Signal Quality Error pin for testing the collision detect circuitry without affecting the TP output, and a LED driver for Link integrity status. An on-chip oscillator, capable of receiving a clock input or operating under crystal control, is also provided for internal timing and driving a buffered clock output.

The MC34055 is manufactured on a BICMOS process and is packaged in a 24 pin SOIC.

- BICMOS Technology for Low Power Operation
- Standard 5.0 V ± 5% Voltage Supply
- Smart Squelch Enforcement of Threshold, Pulse Width, and Sequence Requirements
- Driver Pre-Emphasis for Output Data
- TTL, CMOS and Raised ECL Compatible
- Interfaces to TP Media with Standard 10BASE-T Filters and Transformers
- LED Capable Status Outputs for Collision, Jabber Detection, and Link Integrity
- Directly Driven or Crystal Controlled Clock Oscillator
- Selectable Full-Duplex Operation
- Signal Quality Error Test Pin
- Selectable Loop Back

MAXIMUM RATINGS (TA = 25°C, unless otherwise noted.)

<table>
<thead>
<tr>
<th>Rating</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply Voltage</td>
<td>VCC</td>
<td>0.5 to 7.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>Differential Voltage at RX&lt;-&gt;TX</td>
<td>VDD</td>
<td>-5.25 to 5.25</td>
<td>Vdc</td>
</tr>
<tr>
<td>Voltage Applied to Logic and Mode/Test Select Inputs</td>
<td>0.5 to 5.5 Vdc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage Applied to Logic Outputs and Output Status Pins</td>
<td>0.5 to 7.0 Vdc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ambient Operating Temperature Range</td>
<td>TA</td>
<td>0 to 70 °C</td>
<td></td>
</tr>
<tr>
<td>Junction Temperature</td>
<td>TJ</td>
<td>-55 to 150 °C</td>
<td></td>
</tr>
</tbody>
</table>

NOTE: Devices should not be operated at these limits. The "Recommended Operating Conditions" table provides for actual device operation.
MC34055
Simplified Block Diagram

**RECOMMENDED OPERATING CONDITIONS**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply Voltage</td>
<td>VCC</td>
<td>4.75</td>
<td>5.0</td>
<td>5.25</td>
<td>Vdc</td>
</tr>
<tr>
<td>Voltage Applied to Logic Inputs and Status Pins</td>
<td>VCC</td>
<td>0</td>
<td>0</td>
<td>5.25</td>
<td>Vdc</td>
</tr>
<tr>
<td>Differential Input Voltage</td>
<td>VCC</td>
<td>0.59</td>
<td></td>
<td>2.6</td>
<td>Vpp</td>
</tr>
<tr>
<td>Operating Ambient Temperature</td>
<td>TA</td>
<td>0</td>
<td></td>
<td>70</td>
<td>°C</td>
</tr>
</tbody>
</table>

**ELECTRICAL CHARACTERISTICS**

The device contains 8.875 active transistors.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Current (4.75 V ≤ VCC ≤ 5.25 V)</td>
<td>Ipcc</td>
<td>0.4</td>
<td>60</td>
<td>180</td>
<td>mA</td>
</tr>
<tr>
<td>Reset Circuit Threshold</td>
<td>Vpp</td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TWISTED PAIR TRANSMITTER**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Differential Voltage</td>
<td>V0</td>
<td>2.2</td>
<td>2.53</td>
<td>2.8</td>
<td>Vpp</td>
</tr>
<tr>
<td>Output Differential Voltage with Pre-Emphasis</td>
<td>V0</td>
<td>1.56</td>
<td>1.72</td>
<td>1.98</td>
<td></td>
</tr>
<tr>
<td>Common Mode Driver Impedance</td>
<td>ZcDM</td>
<td>6.0</td>
<td>8.5</td>
<td>14</td>
<td>Ω</td>
</tr>
<tr>
<td>Transmitter Differential Output Impedance</td>
<td>ZOD</td>
<td>8.0</td>
<td>15.5</td>
<td>29</td>
<td>Ω</td>
</tr>
</tbody>
</table>

**TX DATA A**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input High Voltage (4 mV + 20 mA)</td>
<td>Vih</td>
<td>3.15</td>
<td></td>
<td>5.25</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input Low Voltage (0 mA - 150 mA)</td>
<td>Vil</td>
<td>0</td>
<td></td>
<td>0.8</td>
<td></td>
</tr>
</tbody>
</table>

**TX DATA B**

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage (See Load Circuits: ECL Load Circuit)</td>
<td>Vih</td>
<td>0.984 VCC - 0.923</td>
<td>0.984 VCC - 0.703</td>
<td>Vdc</td>
<td></td>
</tr>
<tr>
<td>High:</td>
<td></td>
<td>0.984 VCC - 0.977</td>
<td>0.984 VCC - 0.727</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 0°C</td>
<td>0.894 VCC - 0.975</td>
<td>0.894 VCC - 0.644</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 25°C</td>
<td>0.894 VCC - 0.975</td>
<td>0.894 VCC - 0.644</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low:</td>
<td></td>
<td>0.750 VCC - 0.568</td>
<td>0.750 VCC - 0.281</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 0°C</td>
<td>0.750 VCC - 0.568</td>
<td>0.750 VCC - 0.281</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 25°C</td>
<td>0.750 VCC - 0.568</td>
<td>0.750 VCC - 0.281</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTICE:** All limits are not necessarily functional concurrently.
### MC34055

**ELECTRICAL CHARACTERISTICS** (0°C ≤ Ta ≤ 70°C, VCC = 5.0 V, unless otherwise noted.)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TX EN H</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input High Voltage (IQH = 200 µA)</td>
<td>VH</td>
<td>2.0</td>
<td>–</td>
<td>5.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>Input Low Voltage (IQL = –20 µA)</td>
<td>VL</td>
<td>0</td>
<td>–</td>
<td>0.8</td>
<td>Vdc</td>
</tr>
<tr>
<td><strong>RX DATA A/RX EN H/RXABBB HICITL H</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Voltage (See Load Circuits: CMOS Load Circuit)</td>
<td>VOH</td>
<td>3.7</td>
<td>–</td>
<td>–</td>
<td>Vdc</td>
</tr>
<tr>
<td>High (IQH = –12 mA)</td>
<td>VIL</td>
<td>–</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low (IQL = +16 mA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RX DATA B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Voltage (See Load Circuits: ECL Load Circuit)</td>
<td>VOH</td>
<td>0.984 VCC – 0.923</td>
<td>0.984 VCC – 0.763</td>
<td>Vdc</td>
<td></td>
</tr>
<tr>
<td>High: @ 0°C</td>
<td>VIL</td>
<td>0.984 VCC – 0.877</td>
<td>0.984 VCC – 0.727</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 25°C</td>
<td>VIL</td>
<td>0.984 VCC – 0.825</td>
<td>0.984 VCC – 0.664</td>
<td></td>
<td></td>
</tr>
<tr>
<td>@ 70°C</td>
<td>VIL</td>
<td>0.750 VCC – 0.569</td>
<td>0.750 VCC – 0.361</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SIGNAL QUALITY ERROR TEST ENABLE CONTROL (SQE EN L)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Control Voltage</td>
<td>VH</td>
<td>2.0</td>
<td>–</td>
<td>5.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>Test Disabled (Input High Voltage) (IQH = +20 µA Max.)</td>
<td>VIL</td>
<td>0</td>
<td>–</td>
<td>0.8</td>
<td>Vdc</td>
</tr>
<tr>
<td>Test Enabled (Input Low Voltage) (IQL = –50 µA &lt; IQL &lt; –150 µA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FULL DUPLEX MODE SELECT (FULLD L)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode Select Control Voltage</td>
<td>VIL</td>
<td>0.984 VCC – 0.877</td>
<td>0.984 VCC – 0.727</td>
<td>Vdc</td>
<td></td>
</tr>
<tr>
<td>Normal Operation (Input High) (IQH = +20 µA)</td>
<td>VH</td>
<td>2.0</td>
<td>–</td>
<td>5.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>Full Duplex (Input Low) (–50 µA &lt; IQL &lt; –150 µA)</td>
<td>VIL</td>
<td>0</td>
<td>–</td>
<td>0.8</td>
<td>Vdc</td>
</tr>
<tr>
<td><strong>LOOPBACK TEST MODE FUNCTION (LOOP L)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Control Voltage</td>
<td>VIL</td>
<td>0.984 VCC – 0.877</td>
<td>0.984 VCC – 0.727</td>
<td>Vdc</td>
<td></td>
</tr>
<tr>
<td>Test Disabled (Input High) (IQH = +20 µA)</td>
<td>VH</td>
<td>2.0</td>
<td>–</td>
<td>5.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>Test Enabled (Input Low) (IQL = –200 µA)</td>
<td>VIL</td>
<td>0</td>
<td>–</td>
<td>0.8</td>
<td>Vdc</td>
</tr>
<tr>
<td><strong>LINK FAIL STATUS (LINKFL H)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status Output Voltage (See Load Circuits: CMOS Load Circuit)</td>
<td>VOH</td>
<td>–</td>
<td>–</td>
<td>0.5</td>
<td>Vdc</td>
</tr>
<tr>
<td>Output Low Sink Current</td>
<td>VIL</td>
<td>–</td>
<td>20</td>
<td></td>
<td>mA</td>
</tr>
<tr>
<td><strong>CLOCK OSCILLATOR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clk+ Input Logic Threshold</td>
<td>VH</td>
<td>2.0</td>
<td>–</td>
<td>5.0</td>
<td>Vdc</td>
</tr>
<tr>
<td>High Level Input Voltage (IQH = +100 µA Max.)</td>
<td>VIL</td>
<td>–</td>
<td>–</td>
<td>0.8</td>
<td>µA</td>
</tr>
<tr>
<td>Logic Low Input Voltage (IQL = –200 µA Max.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clk Out Output Voltage (See Load Circuits: CMOS Load Circuit)</td>
<td>VOH</td>
<td>3.7</td>
<td>3.9</td>
<td>–</td>
<td>Vdc</td>
</tr>
<tr>
<td>Logic High (IQH = –12 mA)</td>
<td>VIL</td>
<td>–</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logic Low (IQL = +18 mA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Output Load Circuits

- **ECL Load Circuit**
- **TTL/CMOS Load Circuit**
- **Differential Load Circuit**

### MOTOROLA ANALOG IC DEVICE DATA

Author: Chong Pay Peng, Clarice
### MC34055 TIMING CHARACTERISTICS  (2.7V ≤ VCC ≤ 5.5V)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSMIT START TIMING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TX EN H to TX+/TX- Disable Time</td>
<td>TTXN</td>
<td>–</td>
<td>–</td>
<td>75</td>
<td>ns</td>
</tr>
<tr>
<td>TX Data A/B to TX+/TX- Enable Time</td>
<td>TTXD</td>
<td>–</td>
<td>–</td>
<td>75</td>
<td>ns</td>
</tr>
<tr>
<td>Steady State Propagation Delay of TX Data A/B to TX+/TX- Output</td>
<td>TXSS</td>
<td>–</td>
<td>–</td>
<td>75</td>
<td>ns</td>
</tr>
<tr>
<td>Pre-Emphasis Pulse Width</td>
<td>TRCM</td>
<td>45</td>
<td>–</td>
<td>55</td>
<td>ns</td>
</tr>
<tr>
<td>Transmitter Caused Edge Skew Between TX+ and TX-</td>
<td>TrSkewT</td>
<td>–</td>
<td>–</td>
<td>2.0</td>
<td>ns</td>
</tr>
<tr>
<td>Transmitter Added Edge Jitter to TX+/TX- from TX Data A/B</td>
<td>UiterT</td>
<td>–</td>
<td>–</td>
<td>4.0</td>
<td>ns</td>
</tr>
<tr>
<td>Steady-State Delay between the TX Data A/B Input to the RX Data A/B Outputs for Normal Operation</td>
<td>TXED</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>TX EN H Assert to RX EN H Assert Under Normal Operation</td>
<td>TDEL</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>TRANSMIT STOP TIMING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay between TX EN H Low and TX+/TX- High</td>
<td>TTXD</td>
<td>–</td>
<td>–</td>
<td>75</td>
<td>ns</td>
</tr>
<tr>
<td>TX EN H Assert/De-assert Delay from TX EN H to RX EN H Assert/De-assert</td>
<td>TTXRE</td>
<td>–</td>
<td>–</td>
<td>400</td>
<td>ns</td>
</tr>
<tr>
<td>End of Packet Hold Time from Last TX Data A/B Edge or TX EN H De-assert</td>
<td>TDDC</td>
<td>250</td>
<td>–</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>LINK BEAT PULSES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Link Test Pulse Width</td>
<td>LKPW</td>
<td>80</td>
<td>–</td>
<td>120</td>
<td>ns</td>
</tr>
<tr>
<td>Minimum Link Beat Pulse Duration on RX+/RX-</td>
<td>LDCY_A</td>
<td>80</td>
<td>–</td>
<td>162</td>
<td>ns</td>
</tr>
<tr>
<td>LOOP BACK MODE TIMING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay from Loop L Deassertion to RX EN H Driven from TX EN H Status</td>
<td>LTRA</td>
<td>–</td>
<td>–</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>TX EN H Assert/De-assert to RX EN H, Assert/De-assert when in Loop-Back Mode and Receiver Inactive</td>
<td>LTRX</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>Steady-State TX Data A/B to RX Data A/B when in Loop-Back Mode</td>
<td>LTRD</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>SMART SQUELCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interval Unit Squelch Deactivation</td>
<td>SQ</td>
<td>–</td>
<td>–</td>
<td>5.0</td>
<td>Bits Times</td>
</tr>
<tr>
<td>RECEIVE START TIMING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver-Added Edge Skew to RX Data A/B Signal</td>
<td>U_sawR</td>
<td>–</td>
<td>–</td>
<td>1.5</td>
<td>ns</td>
</tr>
<tr>
<td>Receiver-Added Edge Jitter to RX Data A/B Signal</td>
<td>Uiter R</td>
<td>–</td>
<td>–</td>
<td>1.3</td>
<td>ns</td>
</tr>
<tr>
<td>Start-Up Delay from RX+/RX- to RX Data A/B</td>
<td>TRXHE</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>Delay from RX EN H Assertion Unit RX Data A/B Valid</td>
<td>TRXAE</td>
<td>–10</td>
<td>–</td>
<td>+10</td>
<td>ns</td>
</tr>
<tr>
<td>Steady-State Propagation Delay from RX+/RX- Data A/B</td>
<td>TRXSS</td>
<td>–</td>
<td>–</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>RECEIVE SHUTDOWN TIMING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last received Data Edge until the RX EN H Output forces low</td>
<td>RXDE</td>
<td>155</td>
<td>–</td>
<td>250</td>
<td>ns</td>
</tr>
</tbody>
</table>

MOTOROLA ANALOG IC DEVICE DATA
MC34055

Figure 1. Start Up and Steady State Transmit Timing

Figure 2. Driver Shutdown Timing

Figure 3. Link Pulse Timing
MC34055

Figure 4. Loop Back Timing

Figure 5. Receive Startup Timing

Figure 6. Receive Shutdown Timing

MOTOROLA ANALOG IC DEVICE DATA
### MC34055

**PIN FUNCTION DESCRIPTION**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Clk Out</td>
<td>TTL/CMOS buffered 10 MHz clock output. This pin will source 40μA and sink 16 mA.</td>
</tr>
<tr>
<td>2</td>
<td>TX Data A</td>
<td>CMOS transmit input pin. Data input at this pin is output to the TP media. The input will source less than 175μA and sink less than 20μA.</td>
</tr>
<tr>
<td>3</td>
<td>TX Data B</td>
<td>Raised ECL transmit input pin. Data input at this pin is output to the TP media. The input can source 40μA for a high level input or 70μA for a low level input.</td>
</tr>
<tr>
<td>4</td>
<td>TX EN H</td>
<td>TTL/CMOS transmit enable pin. Transmit is enabled when asserted high. The input will source less than 175μA and sink less than 20μA.</td>
</tr>
<tr>
<td>5</td>
<td>Dig. Gnd</td>
<td>Digital ground</td>
</tr>
<tr>
<td>6</td>
<td>VCC(DigAna)</td>
<td>Digital and analog VCC. With the current consumed at this pin and Pin 18, the device will consume less than 180 mA at 5.0 Vdc.</td>
</tr>
<tr>
<td>7</td>
<td>Ana. Gnd</td>
<td>Analog ground</td>
</tr>
<tr>
<td>8</td>
<td>RX Data A</td>
<td>TTL/CMOS received data output pin. Data from the TP media is output at this pin. The output will source 12 mA and sink 16 mA.</td>
</tr>
<tr>
<td>9</td>
<td>RX Data B</td>
<td>Raised ECL received data output pin. Data from the TP media is output at this pin.</td>
</tr>
<tr>
<td>10</td>
<td>RX EN H</td>
<td>TTL/CMOS received data output enable pin. This pin is asserted after the Smart Squelch circuitry determines that there is valid data at the TP input pins and also when internal loop-back is occurring. The output will source 12 mA and sink 16 mA. The received data outputs are forced high when this pin is low.</td>
</tr>
<tr>
<td>11</td>
<td>Loop L</td>
<td>TTL/CMOS Loopback test select. Asserting this pin causes the transmit data to be looped to the receive circuit while the TP transmit driver sends a link pulse. The output will source less than 175μA and sink less than 20μA.</td>
</tr>
<tr>
<td>12</td>
<td>LNKFL H</td>
<td>This pin is driver high to indicate a link fail state. When low, the pin will sink 20 mA to light an LED. An unsquelched condition due to valid data on the receive circuit will cause the pin to transition high and low in 100 ms intervals.</td>
</tr>
<tr>
<td>13</td>
<td>JABB H</td>
<td>TTL/CMOS Jabber status pin. This pin is asserted when a Jabber condition is detected and will source 12 mA and sink 16 mA.</td>
</tr>
<tr>
<td>14</td>
<td>CTL H</td>
<td>TTL/CMOS status pin. This pin pulled high when Jabber or Collision conditions are detected. Also high for a time interval when a Signal Quality Error test is being performed. The pin will source 12 mA and sink 16 mA.</td>
</tr>
<tr>
<td>15</td>
<td>RX~</td>
<td>The inverting terminal of the TP differential receiver.</td>
</tr>
<tr>
<td>16</td>
<td>RX+</td>
<td>The noninverting terminal of the TP differential receiver.</td>
</tr>
<tr>
<td>17</td>
<td>FULLDL L</td>
<td>TTL/CMOS duplex mode select. When low, this pin forces the device to operate in full-duplex mode. The input will source less than 175μA and sink less than 20μA.</td>
</tr>
<tr>
<td>18</td>
<td>Pwr VCC</td>
<td>Power supply pin. With the current consumed at this pin and Pin 8, the device will consume less than 180 mA at 5.0 Vdc.</td>
</tr>
<tr>
<td>19</td>
<td>Pwr Gnd</td>
<td>Power ground pin.</td>
</tr>
<tr>
<td>20</td>
<td>TX~</td>
<td>The inverting terminal of the TP differential driver.</td>
</tr>
<tr>
<td>21</td>
<td>TX+</td>
<td>The noninverting terminal of the TP differential driver.</td>
</tr>
<tr>
<td>22</td>
<td>SQE EN L</td>
<td>TTL/CMOS Signal Quality Error test enable pin. Pulling this pin low allows test of the collision detect circuitry without affecting the twisted pair channel. The input will source less than 17μA and sink less than 20μA.</td>
</tr>
<tr>
<td>23</td>
<td>Clk~</td>
<td>TTL/CMOS clock oscillator pin. See Pin 24.</td>
</tr>
<tr>
<td>24</td>
<td>Clk+</td>
<td>TTL/CMOS clock oscillator pin. This pin is used with Pin 23 if the internal oscillator is to be free run with a crystal. The oscillator can also be directly driven with a TTL/CMOS clock signal at this pin. The oscillator frequency should be 10 MHz with a duty cycle of 50% 20%.</td>
</tr>
</tbody>
</table>

**MOTOROLA ANALOG IC DEVICE DATA**

---

Author: Chong Pay Peng, Clarice
MC34055

FUNCTIONAL DESCRIPTION

Introduction
The Motorola 10BASE-T transceiver, designed to comply with the ISO 8802-3 (IEEE 802.3) 10BASE-T specification, will support one Medium Dependent Interface (MDI) through standard 10BASE-T filters and transformers. Although the device is capable of being used in embedded or external Medium Attachment Units (MAU), it was primarily designed for use in repeater or hub applications. For this reason a digital interface is provided rather than an AUI interface. This interface is TTL, CMOS, and raised ECL compatible and allows for easy connection in hub applications.

Data Transmission
For data intended for the twisted pair, the MC34055 has two data inputs, TX Data A and TX Data B. TX Data A is CMOS compatible and TX Data B is raised ECL compatible. The inputs were not intended to be used simultaneously in a single application and are internally logically combined. The unused input should be disabled by connection to VCC.

When data transmission is intended, the MC34055 detects the first falling edge of the Manchester encoded frame at the input being used, synchronizes the on-chip oscillator (Pins 23 and 24) and asserts the twisted pair driver output to full differential amplitude within 25 ns if the driver enable pin (TX EN H) is previously asserted. Also, since twisted pair attenuates a 10 MHz signal more than a 5.0 MHz signal the 10BASE-T standard requires that data applied to the twisted pair receive pre-equallization. To fulfill this requirement the MC34055 provides an additional 730 mV for approximately 50 ns to output data. This is accomplished over the single pair of differential driver pins, TX+ and TX-, and effectively equalizes the power of all data components at the receiver. Figure 7A shows a 10 MHz waveform. Figure 7B shows the effect of pre-emphasis on a 5.0 MHz waveform. Manchester encoded data with the pattern shown in Figure 7A would represent a repeating pattern of zeros (0000000...). Figure 7B would represent an alternating pattern of ones and zeros (0101010...).

Figure 7A. 10 MHz Waveform on Differential Outputs

<table>
<thead>
<tr>
<th>Bit Pattern</th>
<th>TX+ Pin 21</th>
<th>1.25 Vp-p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 7B. 5.0 MHz Waveform on Differential Outputs

<table>
<thead>
<tr>
<th>Bit Pattern</th>
<th>TX+ Pin 21</th>
<th>0.50 Vp-p</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
<tr>
<td></td>
<td>0.50 Vp-p</td>
<td>0.888 Vp-p</td>
</tr>
</tbody>
</table>

MOTOROLA ANALOG IC DEVICE DATA

Author: Chong Pay Peng, Clarice
The figures show the voltage waveforms on the differential driver output pins. To actually meet the 10BASE-T specification requires bandpass filtering and a pulse transformer.

The output voltage waveform specifications of the IEEE 802.3 standard require that voltages impressed on the differential components in the driver output circuitry. When the twisted pair meets the voltage resistor pulls TX Data by choosing the appropriate low pass filter and a pulse transformer as shown in Figure 8, the resultant waveform is capable of meeting the voltage template.

Following the end-of-frame activity, an internal pull-up resistor pulls TX Data A/B high and causes the differential driver to maintain full differential output voltage for approximately 250 ns. The differential driver interprets the lack of transition activity as an end of frame and starts an idle timer. Should another frame intended for the twisted pair arrive before the idle timer expires (~250 ns), the idle timer will be reset, if not, the transmit driver function will begin the decay to idle process. During idle periods the differential driver must force the media to a minimal differential voltage unless a link beat is being produced. The transition to minimal voltage is subject to performance requirements in the IEEE specification and is met by the MC34055 when the appropriate filters and transformers are used to interface to the media.

The MC34055 differential driver generates link pulses (beats) during idle periods. The link pulses produced are singular positive (TX+ positive with respect to TX-) pulses applied to the media at 16 ms intervals and last approximately 100 ns. The link pulses allow the receiver at the other end of the link to verify the validity of the segment. There is the possibility, due to the two asynchronous sources, that one of the two input pins (TX Data A or TX Data B) will receive frame activity immediately after a link pulse is initiated. In this event, the transmit differential driver will remain active for the entire frame interval and the link pulse will not affect more than one bit interval.

The MC34055 also has Jabber function to detect and disable the twisted pair drivers in the event that a serial controller fails to transmit. Should any data source try to transmit longer than 20 ms minimum, the Jabber function will disable the differential driver output, the collision presence detector and the internal loopback function. Also, two status indicator pins, CTL H and JAB B H are asserted. The MC34055 will remain in the jabber state until the TX EN H pin is pulled low or the jabbering input ceases to toggle for a minimum of 500 ms. The status indicator pins, CTL H and JAB B H will also sink up to 20 mA and can therefore support external LEDs.

The driver also works with the receiver to provide loop-back. Under normal operating conditions (Loop = "1"), the data applied to the TX Data A/B pins is looped back internally to the RX Data A/B pins. This function is disabled when there is a collision condition or FULL L is low.

Data Reception

Data intended for the DTE proceeds from the twisted pair to the isolation transformer and bandpass filters before reaching the differential receiver terminals. Figure 9 shows the configuration of the external media receive circuitry. Once transitions at the receiver terminals (RX+ and RX-) are detected, the on-chip oscilator is synchronized and the received data is screened by smart squelch circuitry for validity. This qualification requires incoming data to meet amplitude and sequence requirements. If the data meets the Smart Squelch requirements, the receiver enters the unsquelched state and the data is forwarded to the RX Data A/B output pins provided Loop L is not low. Two data outputs are provided to increase design flexibility. RX Data A and RX Data B. RX Data A is CMOS/TTL compatible and RX Data B is raised ECL compatible.
The MC34055 powers up in a squelched and "link OK" state, after which minimum and maximum link test and maximum link fail timers are started. If valid data or a link pulse is received after the link test minimum timer but before the link fail maximum timer times out, the timers are reset and begin counting again. In the event of missing or incorrect link pulses, the MC34055 enters the link fail state whereby the LNKFL H status pin is asserted until valid data or link pulse activity appears at the receiver terminals.

Powering up in the squelched state assures that the data path to the data output pin (RX Data NB) is disabled, and prevents noise at the receiver terminals (RX+ or RX-), from being interpreted as valid input data. Once transitions appear at the receiver terminals, the smart squelch circuitry checks for the smart squelch requirements to unsquelch; an alternating sequence (1010... or 0101...) of pulses with amplitude of at least 525 mV. This requirement is met by the preamble of an IEEE 802.3 frame with good signal to noise ratio. 

After a pulse is received and checked for proper polarity and amplitude, the pulse width is checked for proper duration. If the duration is too short or too long the smart squelch circuitry resets and begins to look again for a proper sequence. By requiring the differential pulses to meet amplitude and sequence requirements, it is unlikely that pulses due to crosstalk from co-resident twisted pairs are capable of causing the receiver to unsquelch. If a positive pulse is received first and the differential driver is not transmitting, the receiver should unsquelch after three alternating pulses. If a negative pulse is received first, one additional pulse is required before unsquelch. If the differential driver is transmitting, three additional pulses are required to unsquelch.

After meeting the smart squelch requirements, the MC34055 will pull high the RX EN H pin and enable the path to the receiver data pin (RX Data NB) provided the MC34055 is not in the loop back test mode (Loop L low). If the receiver unsquelches, the receive enable pin remains high and the data path to the receiver data pin remains enabled until transitions cease to exist at the receiver terminals. Valid data reception is also indicated by high/low transitions of the LNKFL H pin at 100 ms intervals. When transitions at the differential terminals cease, marking the end of frame activity, the receiver re-enters the squelch state, pulls low on the RX EN H pin, and begins accepting valid link pulses until the start of the next 802.3 frame.

If the MC34055 is requested to begin transmitting (TX EN H asserted), and the receiver unsquelches simultaneously, there is a collision. Also, if the MC34055 driver enable pin is previously asserted and the receiver detects valid transition activity, the receiver Smart Squelch circuitry verifies the possibility of collision by requiring three extra transitions at the differential receiver before the unsquelch condition is reached. If unsquelch occurs, a collision condition exists. During all collision conditions the MC34055 asserts the CTL H status pin for the duration of the condition and for a time after the end of collision.

During a collision condition the receive and transmit paths are still both enabled allowing transparency to the media. Either the presence of simultaneous transmit and receive activity or the condition of the CTL H status pin can be used by the communications controller to acknowledge and react to the collision. In applications where a 10 MHz collision signal is required by an EIA, the combination of this status pin and the clock oscillator output can be logically combined to provide a 10 MHz output. If the DTE reacts to the collision and ceases transmitting, the MC34055 will decay to idle until a re-transmit is attempted.

Crystal Oscillator
The MC34055 has an on-chip clock oscillator used to provide a reliable and accurate time reference to all the internal timers. The oscillator can be run with a crystal or driven at Pin 24 from an external clock source. Also provided is a buffered clock output which is useful if the MC34055 is to be used in a repeater or concentrator application.

Table 1. The crystal used in the oscillator is subject to the following specifications.

<table>
<thead>
<tr>
<th>Crystal Operating Mode</th>
<th>Fundamental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crystal Cut Type</td>
<td>AT</td>
</tr>
<tr>
<td>Crystal External Shunt Capacitance</td>
<td>7.6 pF Max</td>
</tr>
<tr>
<td>Crystal Resonant Mode</td>
<td>Series</td>
</tr>
<tr>
<td>Crystal Accuracy</td>
<td>± 0.01% @ 25°C</td>
</tr>
<tr>
<td>Crystal Temperature Variance</td>
<td>0.005% from 0°C to 70°C</td>
</tr>
<tr>
<td>Crystal Series Resistance</td>
<td>25 Ω Max, 17 Ω Typical</td>
</tr>
<tr>
<td>Crystal Operating Temperature Range</td>
<td>0°C to 70°C</td>
</tr>
</tbody>
</table>

Author: Chong Pay Peng, Clarice
MC34055

**LOOP L Test Mode**

If the Loop L pin is low, the MC34055 is in a test mode whereby the data at the input pin (TX Data A/B) is being looped back internally to the receive data output pins (RX Data A/B). In this mode the data path from the differential receiver terminals to the receive data output pins is disconnected while the Smart Squelch functionality of the differential receiver is still operational. This test mode allows the DTE to test the MC34055 internal loop back circuitry since the data is looped back to the receive circuitry as close to the twisted pair interface as possible.

**Signal Quality Error Test**

The MC34055 also provides the ability to test the collision detect circuitry without disabling either of the data paths. By pulling the SQE EN L pin low, a collision test is provided to the collision detect circuitry immediately following the last edge of a transmitted 802.3 frame. The test verifies the operability of the collision detect circuitry, operability is announced by the assertion of the CTL H pin for a period following a valid data transmission.

**Jabber Detection**

The transmit circuitry of the MC34055 has the ability to monitor and shut down the differential driver in the event of a jabber condition. If transmission activity ever exceeds 20 ms minimum, the differential driver, the collision detect, and internal loop back circuits are disabled. To announce the presence of a jabber condition, both the CTL H and the JABB H status output pins are asserted. In order to end the jabber condition, the TX Data A/B input must stop toggling, or the TX EN H pin must be pulled low for a minimum of 500 ms. The status output pins have the ability to drive an external led and were added to facilitate network manageability. The jabber status outputs will not assert during power up or power down.

**Full Duplex Mode**

The MC34055 can be operated in a full-duplex mode if required. When the FULLD L pin is pulled low the device will enter the full duplex mode. This mode allows the transmitter and receiver to operate independently. Collision will not be announced and the internal loop back operation is disabled. The Signal Quality Error test, however, is still operational if enabled.

**Status Pins**

The MC34055 has three status indicator pins capable of sourcing or sinking enough current to support an external LED. Status pin levels ("1" or "0") report the condition of the transceiver. Table 2 shows the combinations and significance.

<table>
<thead>
<tr>
<th>Status Pin</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>JABB H</td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>CTL H</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>LNKFL H</td>
<td>X</td>
</tr>
<tr>
<td>&quot;0&quot;</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;1&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;0&quot;</td>
</tr>
<tr>
<td>X</td>
<td>&quot;1&quot;</td>
</tr>
</tbody>
</table>

**Test Select Pins**

The MC34055 has three operation mode test select pins, Loop L, SQE EN L and FULLD L. The level of the pin determines the mode of operation. Table 3 shows the levels and corresponding conditions of the status pins.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Status</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop L</td>
<td>&quot;1&quot;</td>
<td>Normal operating mode. Loop back occurs when the transmitter initiates and the receiver is receiving link pulses. The RX EN H pin follows the TX EN H pin and the transmit data appears on the RX Data A/B output pin being used.</td>
</tr>
<tr>
<td></td>
<td>&quot;0&quot;</td>
<td>Loop back test mode. The transmit circuit is looped back internally as close to the differential receive circuit as possible. In this mode the RX EN H pin follows the TX EN H pin and the transmit data appears on the RX Data A/B output pin being used. Any received data other than link pulses are ignored and the receiver will not unsquelch or announce collision.</td>
</tr>
<tr>
<td>SQE EN L</td>
<td>&quot;0&quot;</td>
<td>Normal operating mode. Concurrent transmit and receive activity results in a collision condition.</td>
</tr>
<tr>
<td></td>
<td>&quot;1&quot;</td>
<td>Test enabled. An internal test is run on the collision circuitry and the CTL H pin is asserted for a time window following the last positive packet edge. Data transmission and reception is undisturbed.</td>
</tr>
<tr>
<td>FULLD L</td>
<td>&quot;1&quot;</td>
<td>Internal loop-back is disabled and collision will not be announced. Signal Quality Error test is still operable.</td>
</tr>
<tr>
<td></td>
<td>&quot;0&quot;</td>
<td>Internal loop-back is disabled and collision will not be announced. Signal Quality Error test is still operable.</td>
</tr>
</tbody>
</table>
MC34055

APPLICATIONS INFORMATION

The MC34055 implements the physical layer of a 10BASE-T application of IEEE 802.3. It provides the physical connection to the media (twisted pair) and the services required by the MAC sublayer of the Data Link Layer. Two interfaces are defined in the IEEE 802.3 specification of the physical layer; one is the MDI providing connection to the twisted pair; and the other is the AUI providing connection to the encoder/decoder function of the Data Link Layer. While the MC34055 provides the connection to the twisted pair, a CMOS and raised ECL interface is provided instead of an AUI.

The MC34055 implements the twisted pair interface of the physical layer in a 802.3 10BASE-T application but circuitry must be added if an AUI is desired, (see Figure 10 for suggested schematic). For example, an external MAU application requires the AUI and a twisted pair interface. A chip capable of realizing the AUI interface is the Texas Instruments SN75ALS085. This IC has an AUI interface and another interface which is compatible with the MC34055. The differential input of the 75ALS085 can be used for the TX+/TX- terminals of the AUI. The differential drivers of the 75ALS085 can be used as the RX+/RX- and the COL+/COL- terminals of the AUI. The other interface of the 75ALS085 then will interface to the MC34055 by three paths shown in the application suggestion. The application accounts for all the inputs/outputs of an external MAU.

Embedded applications do not require a full AUI and a MC10116 can be used to interface between the raised ECL interfaces of the MC34055 and the AUI of existing encoder/decoder chips. The MC10116 is a MECL 10k Triple Line Receiver with typical propagation delay and rise and fall times (20% to 80%) of 2.0 ns. Figure 11 shows the use of the MC10116 with the MC34055 and the AMD 7992 SIA.

In a multi-port repeater, or hub, a port is required for each DTE connected to the IEEE 802.3 network. This port consists of two connections, one for the TX+TXX-pair and another for the RX+RX-pair. The repeater unit then multiplexes these lines so that all of the stations are capable of transmitting to or receiving from all the other stations on the network. This establishes the need for a transceiver without an AUI Interface. If an AUI is present with each 10BASE-T transceiver, chip count is increased because there is a requirement to convert from balanced to unbalanced lines before multiplexing.

An application suggestion for the use of the MC34055 used in a multiport repeater is shown in Figure 6. Here the receive and transmit lines for the 10BASE-T transceivers are multiplexed by the hub hardware.
Figure 12. 10BASE-T Concentrator Application
MC34055

OUTLINE DIMENSIONS

**DW SUFFIX**

PLASTIC PACKAGE

CASE 751E-04

(30-24L)

**NOTES:**


2. CONTROLLING DIMENSION: MILLIMETERS.

3. THERMAL SLOPE SPECIFIED WITHOUT TEST.

4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.

5. SHAPE OF IC CORNER WILL BE ALLOWED UP TO 0.10 (0.004) TO EXCEED 2 SIDES AT MAXIMUM MATERIAL CONDITION.

**Allowable protrusion shall be limited to 0.10 (0.004) in any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer must assure itself of the suitability of the product for its application and agrees to assume all risks of such use.**

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranties, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation incidental or consequential damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications shall not be considered as limits of tolerances. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are designed, oriented, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life. Motorola makes no representation that Motorola products are qualified for such applications, and Buyer must assure itself of the suitability of these products. Buyer’s application of Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Motorola logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.