High bit rate Digital Subscriber Line

Syndicate

Capt Tanveer Hussain
GC Kashif Farooq
GC Moonis Ali (Project Leader)
GC Altan Hafeez
PC Armughan Ahmed

Supervisor

Major Imtiaz Khokhar

Dissertation to be presented as partial requirement for the award of
B.E. Degree in Telecommunication Engineering

Department of Electrical Engineering
Military College of Signals
National University of Sciences & Technology
Rawalpindi
May 2003

ABSTRACT

Digital Subscribe Line (DSL) technology allows transmission of digital data at speed of 1-5 Mbps over normal, unconditioned copper phone lines. This technology is used to host broadband communication and among this xDSL technology is a mature High bit rate DSL (HDSL) technology. HDSL allows transmission of data of 1.544/2.048 Mbps up to a distance of 5-6 km over copper phone line.

The goal of the project is to develop a basic platform and hardware design for further research work. Modular approach is adopted for designing because it is easy to implement and understand. The modules are designed and simulated considering the practical requirements yet there are limitations of the software used.

The dissertation discusses E1 being decoded into bitstream and than converted to two-bit one-quaternary (2B1Q) line code to enable E1 to pass on copper pair at the receiver end the signal is decoded from 2B1Q format and than encoded back to E1 standard high density bipolar order 3 (HDB3) format.

Our main project report will reflect the major design portion in the form of HDB3 Codec and 2B1Q Codec but it also covers the simulations, PCB designs of the modules and auxiliary circuits required during the course of project.

DECLARATION

It is declared that this petition either in a whole or in parts is not presented for award of any degree at any institution and it is further declared that the project is outcome of the futile efforts put in by the complete syndicate as a team.

Table of contents

Chapter 1 HDSL Overview

Chapter 2 Software Employed

Chapter 3 HDB3 Codec

Chapter 4 2B1Q Codec

Chapter 5 Auxiliary Circuits

Chapter 6 PCB Designing

CHAPTER 1

HDSL OVERVIEW

1.1 INTRODUCTION

High Bit-Rate Digital Subscriber Line (HDSL) Modem allows telecommunications service providers and private network operators to obtain maximum performance from existing copper facilities. Using the HDSL Modem, network operators can provide E1/T1 service on copper pair, over distances up to 6-km [1]. The HDSL Modem is the solution to customer demand for more bandwidth, without having to upgrade facilities.

Standard E1 G.703/704 communications links make use of coaxial cable, which is expensive and hard to install. As an alternative a system with an HDSL primary rate interface module installed provides bi-directional (full-duplex) 2.048 Mb/s transmission capability over two pairs of unconditioned copper wires [2]. This allows service providers and private network users to take advantage of their installed base of copper facilities.

The modem can be used alone to provide a G.703 (CCITT/ITU E1) interface; or, it may be coupled with modem’s E1 primary rate Voice/Data Multiplexer to provide channelized voice and data interfaces for a wide variety of user applications. The HDSL Modem supports E1/T1 termination, drop/insert and branching configurations offering a full range of network architecture [2].

1.1.1 NYQUIST CRITERIA

A decreased frequency of operation and several enhancements that corrects common problems inherent in standard E1/T1 transmission, helps HDSL to provide superior service. Some of these problems are repeater reliability, pulse density, bridged taps, and degrading splices [5].

The conversion process from T1 to HDSL centers on a basic transmission theory that relates a signal’s frequency to the distance of transmission: the lower the signal’s Nyquist (center) frequency, the greater the distance it can travel.

For example, a 56 kb/s signal, having a Nyquist frequency of 28 kHz, can travel much further than a T1 signal having a Nyquist frequency of 772 kHz.

HDSL splits the incoming T1 into two separate data streams reducing the overall bit rate from 1.544 Mb/s to 784 kb/s per loop. Decreasing the overall bit rate –

and encoding with 2B1Q – combine to reduce the Nyquist frequency of HDSL transmission from the 772 kHz of standard T1 to196 kHz.

The lower Nyquist frequency enables HDSL to travel up to 12,000 ft. as opposed to 6,500 ft. with standard T1. This additional range enables service providers to reach approximately 80% of their customers without deploying line repeaters.

The lower frequency of HDSL transmission provides additional benefit by being less susceptible to cable impairments. In general, the higher a signal's frequency, the more vulnerable it is to cable faults.

As analog signal suffers less attenuation compared to a digital signal on cable so longer distances can be achieved by increasing the amplitude levels. Hence levels are selected as per system requirement.

1.1.2 BIT RATES

T1 rates specify 1.544 Mbps. This translates to a baud rate of 784 kbaud (symbols/second). The baud rate is not exactly a factor of 2 because of error correction bits added to data. The error correction supports adaptive equalization of the transmission line as well as 2nd order echo cancellation. Adaptive equalization is a DSP process that continuously tunes the digital filters to match conditions on the line. For two pair lines the symbols are transmitted over each twisted pair at equal rates; therefore the line transceiver circuitry must support 1/2 the modem baud rate [3].

1.1.3 Applications

The proliferation of ISDN (Integrated Services Digital Network) demonstrates the demand for high-speed modems for transmission of data. HDSL (High Speed Digital Subscriber Line) is a transmission scheme that fills the gap between the 128 kbps ISDN and ADSL (Asynchronous Digital Subscriber Line). Although HDSL cannot support this data rate, like ADSL, HDSL will provide high-speed data transmission over existing twisted pair telephone lines. Unlike ADSL, HDSL transmits at the same rate upstream and downstream [3].

As we know that T1 rates (US) specifies 1.544 Mbps over one or two pairs of copper line while E1 rates (Eur) call for 2.048 Mbps. HDSL modems available today support data transmission at these rates over two pair of copper wire, with equal data rates on both pairs.

HDSL modems, support data transmission at rates over two pair of copper, with equal data rates on both pairs (i.e., 784 kbps on each pair for a T1 link; 1168 kbps for E1) and single pair systems (This has a more limited range and requires much more sophisticated DSP). HDSL systems do not require that the lines be matched; they can still transmit accurately with stubs present on the transmission line [1].

A typical HDSL application might be the T1/E1 link between a cellular phone system basestation transceiver and the basestation controller, as well as a high-speed modem for Internet access and replacement for E1/T1 lines.

1.1.4 ADVANTAGES

a) The primary advantage of HDSL is that it is a mature and proven technology.
b) Easy and economical to install.
c) Its other main advantage is that it has decent transfer rate in both directions at 1.544 Mbps (784 kbps x 2) for T1 and 2.048Mbps(1168kbps x x 2) for E1.
d) Maximizes revenue and protects investment in existing copper facilities.
e) Delivers full E1 service up to 6 km using 2 twisted 0.9 mm copper pairs.
f) Greater distances can be achieved with larger diameter wire.
g) Increases end-user capacity without upgrading facilities.
h) Allows fast and cost-effective deployment of equipment and services.

1.1.5 DISADVANTAGES

a) The primary disadvantage of HDSL was that it requires two twisted pairs of wires to operate, which increases the deployment cost for service providers. The problem has been overcome by employing better dsp techniques [4].
b) HDSL does not support Plain Old Telephone Service (POTS) on the same lines as the data, so separate phone service is needed if the consumer wants to speak on the telephone.
c) Another disadvantage is that HDSL is slightly slower than some other forms of DSL, but it is still far superior to analog transmission.

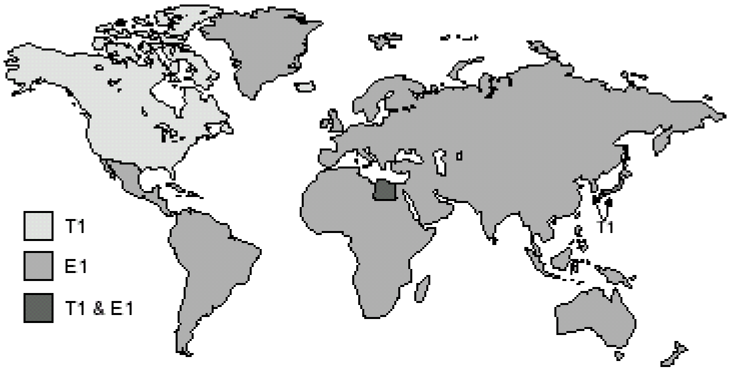
1.1.6 PRACTICAL IMPLICATIONS

a) High speed connectivity for last mile solutions using existing copper facilities.
b) Voice, data and digitized video delivery.
c) Spur lines off microwave or fibre-optic backbone networks.

1.2 E1 / 2.048 Mbps

1.2.1 BASICS

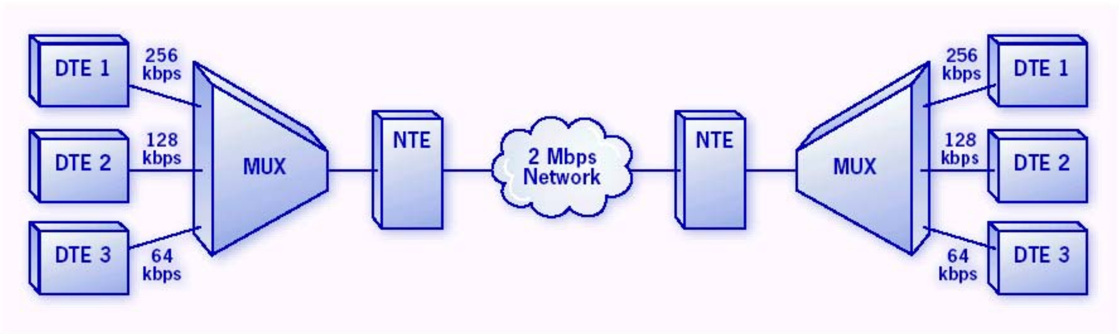
A 2.048 Mbps circuit provides high speed, digital transmission for voice, data, and video signals at 2.048 Mbps. 2.048 Mbps transmission systems are based on the ITU-T specifications G.703, G.732 and G.704, and are predominant in Europe, Australia, Africa, South America, and regions of Asia. Due to an increase in demand for global communications in recent years, 2.048 Mbps installations in



North America have risen sharply, and exist alongside the standard T-Carrier systems [6].

Figure* 1.1 Worldwide *Coverage ***source: NETS Lahore***

The 2.048 Mbps standards are now firmly established for transmission systems and are used by telecommunications network suppliers, international carriers and end users. The primary use of the 2.048 Mbps is in conjunction with multiplexers for the transmission of multiple low speed voice and data signals over one



communication path rather then over multiple paths.

Figure 1.2 A typical 2.048 Mbps transmission system
Source: TTC Networks

1.2.2 Line Code

The most common line code used to transmit the 2.048 Mbps signal is known as HDB3 (High Density Bipolar 3) which is a bipolar code with a specific zero suppression scheme where no more then three consecutive zeros are allowed to occur. The HDB3 line code is recommended for 2.048 Mbps signals by ITU-T Recommendations G.703 [6].

In some instances straightforward bipolar AMI (Alternate Mark Inversion) coding with no zero suppression is also encountered. In the following paragraphs, we will first review the AMI coding format, which represents the simplest version of bipolar line code. We will then move on to explaining the 2.048 Mbps HDB3 line code, which essentially is a variation of AMI where a high density of pulses is ensured by applying a zero suppression algorithm.

AMI or Bipolar Line Code In the AMI coding format, a binary one (mark) is represented by a square pulse with a 50% duty cycle and a binary zero (space) is represented by the lack of pulse, i.e., 0 Volts. Since successive pulses (i.e., marks) alternate in polarity the line code is termed AMI (Alternate Mark Inversion).

HDB3 Line Code Despite its numerous advantages, AMI coding has one very significant shortcoming. Since signal transition are the only way for 2.048 Mbps equipment to recover the timing information, long strings of zeros with no pulse transition in the data stream may cause the equipment to lose timing. Hence AMI coding puts strict limitations on the zero content of the data transmission in the 2.048 Mbps system. One solution to this problem is to use a coding scheme that suppresses long string of zeros by replacing them with a specific sequence of pulses, which can be recognised and decoded as zeros by 2.048 Mbps equipment. HDB3 is one such coding scheme upon which the 2.048 Mbps industry standardised [6].

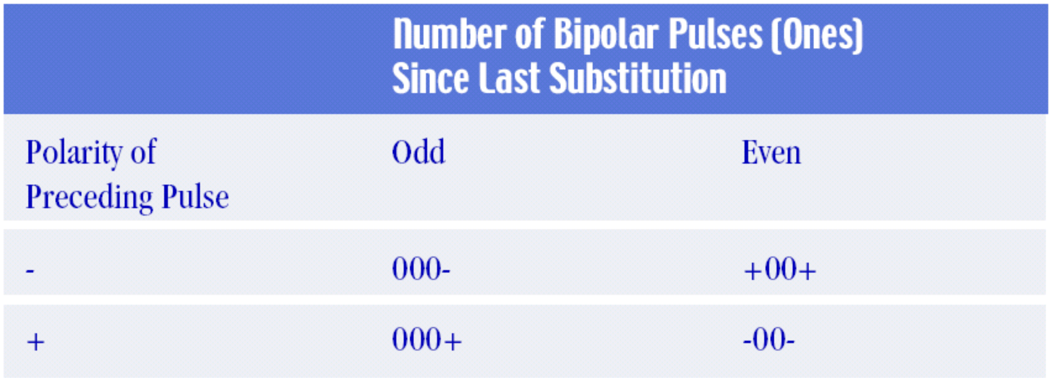
1.2.3 HDB3 CHARACTERISTICS

The HDB3 signal is a bipolar signal, where sets of 4 consecutive zeros are replaced by a specific sequence of pulses and the last pulse is coded as a violation. This ensures that the 2.048 Mbps signal has a high density of pulses and no more then 3 consecutive zeros. *Table 1.1* shows the rules for zero substitution using the HDB3 coding scheme. An example of how these rules are applied to an AMI signal is shown in *Figure 1.3*. It is important to note that:

1. The 4th zero is always coded as a violation pulse.

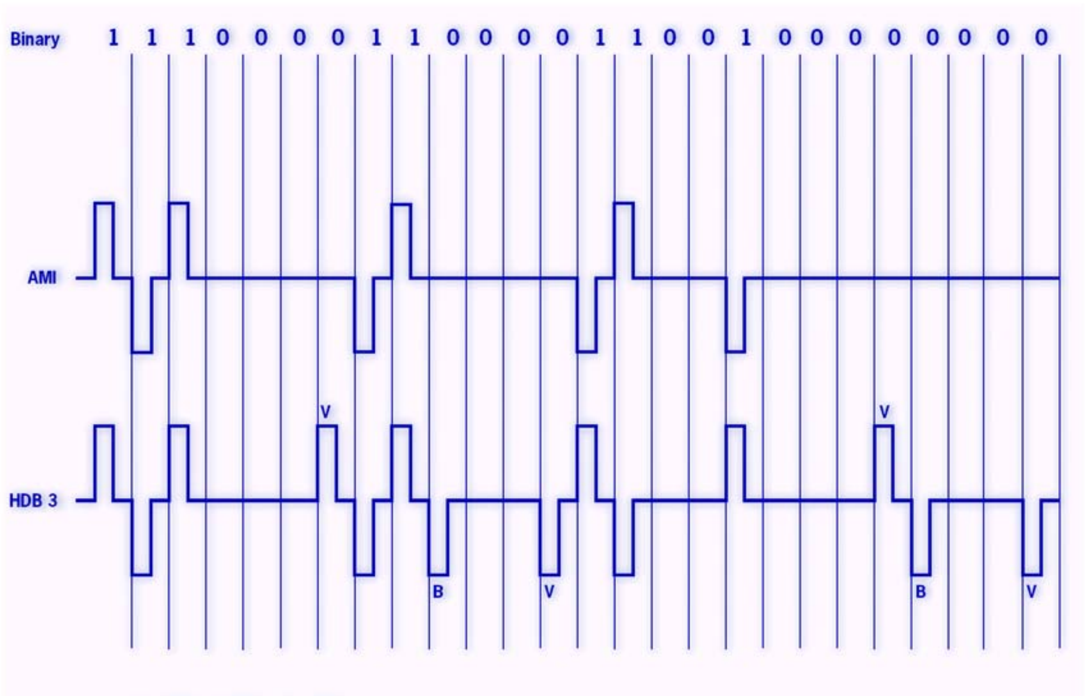
2. The 1st zero may be coded as a “balancing” pulse to ensure that successive HDB3 violation pulses are of opposite polarity, so that the net DC component of the signal remains zero.

Hence the HDB3 code eliminates all the limitations on the zero content of the signal transmitted in the 2.048 Mbps system, while preserving all the advantages of AMI coding. Successive HDB3 violation pulses are of opposite polarity, so that the net DC component of the signal remains zero



	Number of Bipolar Pulses (Ones) Since Last Substitution	
Polarity of Preceding Pulse	Odd	Even
-	000-	+00+
+	000+	-00-

Table 1.1 HDB3 substitution rules   Source: TTC Networks



V = Pulse violating the AMI sequence
B = Additional pulse ensuring that the consecutive V pulses are of opposite polarity

Figure 1.3 Example of a HDB3 signal   *Source: TTC Networks*

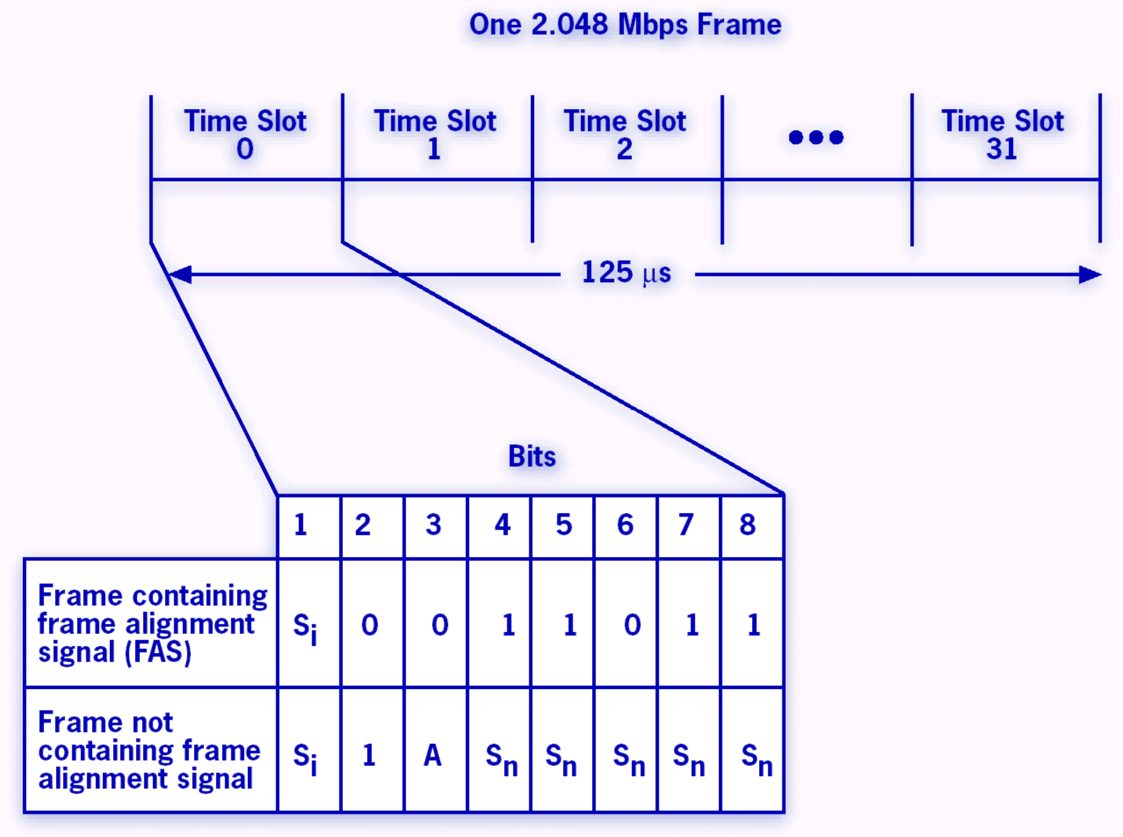
1.2.4 The 2.048 Mbps Framing Format

The 2.048 Mbps signal typically consists of multiplexed data and/or voice which requires a framing structure for receiving equipment to properly associate the appropriate bits in the incoming signal with their corresponding channels. *Figure1.4* shows the framing for the 2.048 Mbps signal as defined in ITU-T Recommendation G.704.

As can be seen in *Figure 1.4*, the 2.048 Mbps frame is broken up into 32 timeslots numbered 0-31. Each timeslot contains 8 bits in a frame, and since there are 8000 frames per second, each time slot corresponds to a bandwidth of 8 x 8000 = 64 kbps. Time slot 0 is allocated entirely to the frame alignment signal (FAS) pattern, a remote alarm (FAS Distant Alarm) indication bit, and other spare bits for international and national use [6].

The FAS pattern (0011011) takes up 7 bits (bits 2-8) in timeslot 0 of every other frame. In those frames not containing the FAS pattern, bit 3 is reserved for remote alarm indication (FAS Distant Alarm) which indicates loss of frame alignment when it is set to 1. The remaining bits in timeslot 0 are allocated as shown in *Figure 1.5*.

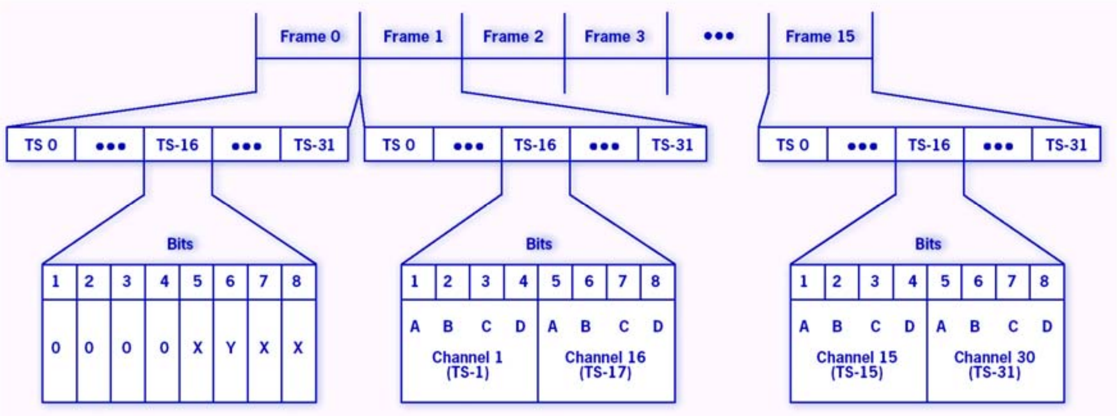
If the 2.048 Mbps signal carries no voice channels, there is no need to allocate additional bandwidth to accommodate signalling. Hence, time slot 1-31 are available to transmit data with an aggregate bandwidth of 2.048 Mbps - 64 kbps (TSO) = 1.984 Mbps.



	1	2	3	4	5	6	7	8
Frame containing frame alignment signal (FAS)	S_i	0	0	1	1	0	1	1
Frame not containing frame alignment signal	S_i	1	A	S_n	S_n	S_n	S_n	S_n

Frame Alignment Signal (FAS) pattern - 0 0 1 1 0 1 1
Si = Reserved for international use (Bit 1)
Sn = Reserved for notational use
A = Remote (FAS Distant) Alarm - set to 1 to indicate alarm condition

Figure 1.4 The 2.048 Mbps framing format Source: TTC Networks



Multiframe Alignment Signal (MFAS) pattern - 0 0 0 0
X = Spare parts (set to 1 if not used)
Y = Remote Alarm (set to 1 to indicate loss of multiframe alignment)

A B C D = Signaling bits

NOTE: Even numbered frames contain the FAS pattern in time slot 0

Figure 1.5 2.048 Mbps TS-16 multiframe format Source: TTC Networks

If there are voice channels on the 2.048 Mbps signal, it is necessary to take up additional bandwidth to transmit the signalling information. ITU-T Recommendation G.704 allocates time slot 16 for the transmission of the channel-associated signalling information. This is explained in the next section [6].

1.2.5 The Signaling Mechanism

Signaling mechanisms provide a wide range of functions and their protocol is application specific. Two modes of signaling are optional:

a) CCS – Common Channel Signaling. In this mode of operation one or more channels of 64 kbit/s are dedicated for signaling and the information carried in them asynchronously serves for all other channels. TS16 is usually used for this purpose.

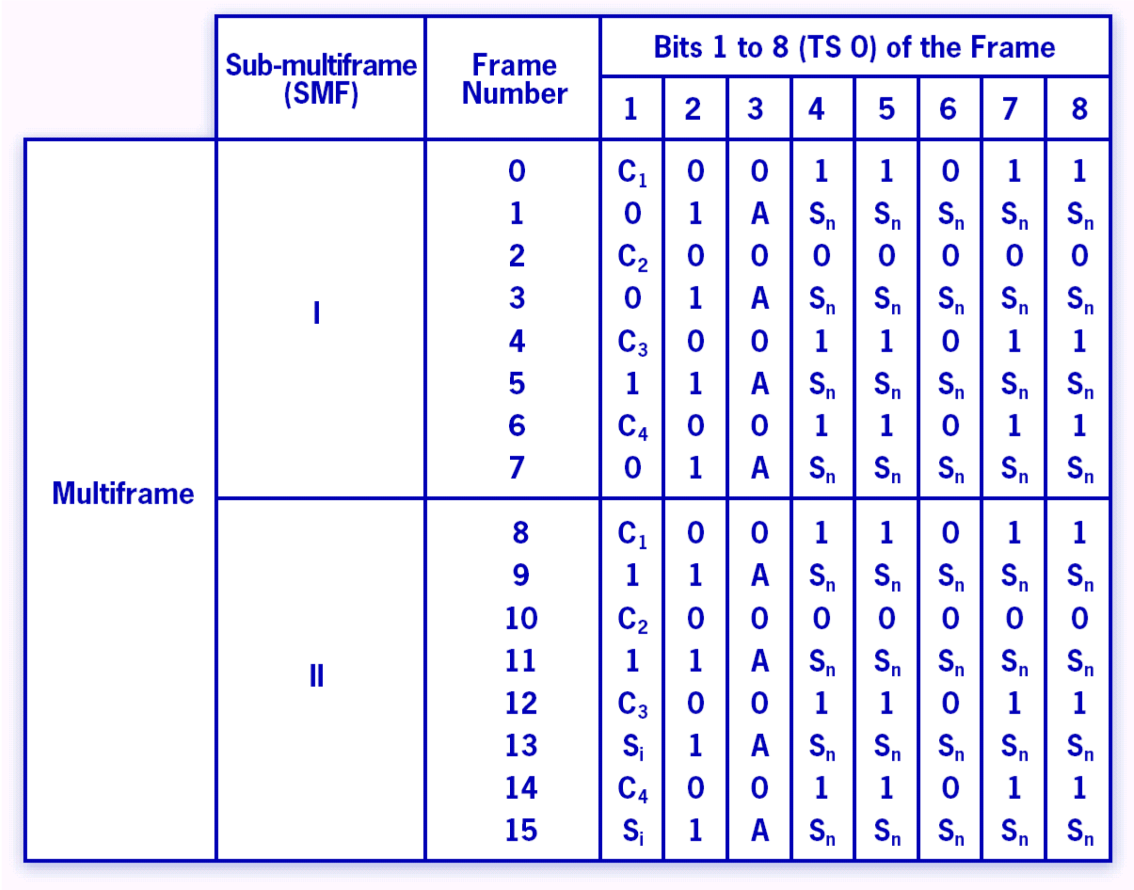
b) CAS – Channel Associated Signaling. In each Multiframe, each channel has a predetermined frame. In this frame half of TS16 is dedicated for this channel signaling information. The use of TS16 in each specific frame is described in next section.

1.2.6 CHANNEL-ASSOCIATED SIGNALLING (CAS)

CAS is a framing pattern on the E1 link. In CAS framing, timeslot 16 is used for channel-associated signalling and is not user-configurable. Timeslot 0 is used for framing. The rest of the timeslots are used for information. Whereas in CCS, timeslot 16 is used for circuit 31, which can be configured as an additional 64 kb/s data channel (typically for transport of CCS messages). Timeslot 0 is used for framing. The rest of the timeslots are used as in CAS.

Time slot TS-16 Multiframe Format of the 2.048 Mbps can carry up to thirty 64 kbps voice channels in time slot 1-15 and 17-31. Voice channels are numbered 1-30; voice channels 16-30 are carried in time slot 17-31.However, the 8 bits in time slot 16 are not sufficient for all 30 channels to signal in one frame. Therefore, a multiframe structure is required where channels can take turns using time slot 16 [6].

Since two channels can send their ABCD signalling bits in each frame, a total of 15 frames are required to cycle through all of the 30 voice channels. One additional frame is required to transmit the multiframe alignment signal (MFAS) pattern, which allows receiving equipment to align the appropriate ABCD signalling bits with their corresponding voice channels. This result in the TS-16 multiframe structure where each multiframe contains a total of 16 2.048 Mbps, numbered 0-15. *Figure 1.5* on the previous page shows the TS-16 multiframe format for the 2.048 Mbps signal as defined by the ITU-T Recommendation G.704. As can be seen in *Figure 1.5*, time slot 16 of frame 0 contains the 4-bit long multiframe alignment signal (MFAS) pattern (0000) in bits 1-4. The "Y" bit is reserved for the remote alarm (MFAS Distant Alarm) which indicates loss of multiframe alignment when it is set to 1 [6].



	Sub-multiframe (SMF)	Frame Number	Bits 1 to 8 (TS 0) of the Frame							
			1	2	3	4	5	6	7	8
Multiframe	I	0	C_1	0	0	1	1	0	1	1
		1	0	1	A	S_n	S_n	S_n	S_n	S_n
		2	C_2	0	0	0	0	0	0	0
		3	0	1	A	S_n	S_n	S_n	S_n	S_n
		4	C_3	0	0	1	1	0	1	1
		5	1	1	A	S_n	S_n	S_n	S_n	S_n
		6	C_4	0	0	1	1	0	1	1
		7	0	1	A	S_n	S_n	S_n	S_n	S_n
	II	8	C_1	0	0	1	1	0	1	1
		9	1	1	A	S_n	S_n	S_n	S_n	S_n
		10	C_2	0	0	0	0	0	0	0
		11	1	1	A	S_n	S_n	S_n	S_n	S_n
		12	C_3	0	0	1	1	0	1	1
		13	S_i	1	A	S_n	S_n	S_n	S_n	S_n
		14	C_4	0	0	1	1	0	1	1
		15	S_i	1	A	S_n	S_n	S_n	S_n	S_n

C1, C2, C2, and C4 = Cyclic Redundancy Check Bits
CRC Multiframe Alignment Signal 0 0 1 0 1 1
Sn = Reserved for notational use
A = Remote (FAS Distant) Alarm - set to 1 to indicate alarm condition

Figure 1.6 The 2.048 Mbps CRC multiframe format Source: TTC Networks

Time slot 16 of frames 1-15 contains the ABCD signalling bits of the voice channels. Time slot 16 of the nth frame carries the signalling bits of the nth and (n+15)th voice channels.

For example, frame 1 carries the signalling bits of voice channels 1 and 16, frame 2 carries the signalling bits of channels 2 and 17 etc. It is also important to note that the frame alignment signal (FAS) is transmitted in time slot 0 of the even numbered frames.

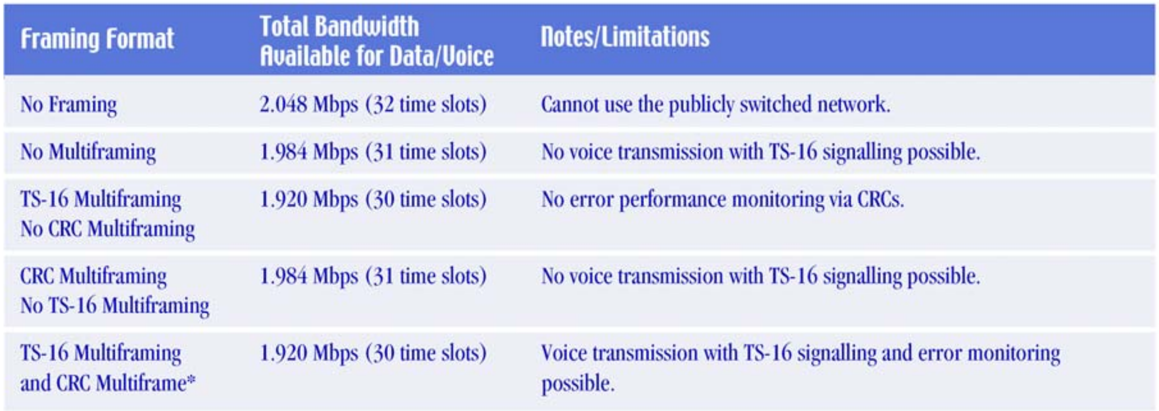
We have thus explained how frame alignment and channel associated signalling are achieved in 2.048 Mbps transmission. (Alternatively, time slot 16 may also be used for common channel signalling applications such as primary rate ISDN) [6].

It must be noted, however, that the 2.048 Mbps framing and TS-16 multiframing structures discussed so far do not provide any built in error detection capabilities, which could be used to determine the error performance of the 2.048 Mbps system on an in-service basis. This capability is provided by the CRC (Cyclic Redundancy Check) multiframe structure as explained in the next section.

1.2.7 CYCLIC REDUNDANCY CHECK (CRC) MULTIFRAME FORMAT

This section describes the specifics of the 2.048 Mbps CRC Multiframe format. The 2.048 Mbps CRC Multiframe structure as defined by ITU-T Recommendation G.704 is shown in *Figure 5*

The CRC Multiframe consists of 16 frames (numbered 0-15) which are divided into two sub-multiframes (SMF-1 and SMF-11) of 8 frames each. The 4-bit long CRC word associated with each submultiframe, SMF (N) is inserted into the next sub-multiframe, SMF (N+1). The CRC bits take up the 1st bit of time slot 0s containing the 7-bit FAS (Frame Alignment Signal) pattern [6].



Framing Format	Total Bandwidth Available for Data/Voice	Notes/Limitations
No Framing	2.048 Mbps (32 time slots)	Cannot use the publicly switched network.
No Multiframing	1.984 Mbps (31 time slots)	No voice transmission with TS-16 signalling possible.
TS-16 Multiframing No CRC Multiframing	1.920 Mbps (30 time slots)	No error performance monitoring via CRCs.
CRC Multiframing No TS-16 Multiframing	1.984 Mbps (31 time slots)	No voice transmission with TS-16 signalling possible.
TS-16 Multiframing and CRC Multiframe*	1.920 Mbps (30 time slots)	Voice transmission with TS-16 signalling and error monitoring possible.

Table 1.2 Various 2.048 Mbps frame and multiframe formats Source: TTC Networks

The two multiframe structures are not related, and need not be aligned with each other in any way. Alignment signal uses the 1st bit of time slot 0s not containing the FAS pattern (*Figure 1.6*). Combining the TS-16 and CRC Multiframe Structures, a 2.048 Mbps signal may come in a number of different formats, depending on which of the above frame and multiframe structures are implemented in the 2.048 Mbps system. *Table 1.2* gives a comparison of the possible variations of a 2.048 Mbps signal.

1.2.8 Analysis of Slips

A pattern slip is the insertion of data bits into or from the data stream [6]. Based on the source of the slip and its effect on the network, all slips can be placed on any of the following categories.

1. **Controlled Slips**: Controlled Slips are bit additions or deletions, which do not disrupt frame synchronisation. These slips are typically caused by synchronisation impairments in digital cross-connect (DCS) equipment. DCS equipment handles buffer overflows or underflows by deleting or repeating entire frames of data. Since data is added or deleted by entire frames, frame synchronisation is not disrupted.

2. **Uncontrolled Slips**: Uncontrolled slips are bit additions or deletions that cause both data and framing bits to be displaced. The misalignment of framing bits typically results in frame synchronisation loss. Uncontrolled slips are typically from synchronisation problems in equipment which buffer the entire bit stream such as satellite down link receivers. Since the buffer in this equipment does not distinguish between framing and data bits, buffer underflows or overflows result in the addition and deletion of arbitrary blocks of data. It should be noted that slips could also result from impairments unrelated to network synchronisation. Low signal level, noise, and excessive jitter can also cause slips.

1.3 MODULATION METHOD

Data transmitted on HDSL lines are coded with 2B1Q (2 Bits 1 Quaternary) coding. This PAM modulation method groups data bits into 2 bits per symbol, with each symbol represented by one of four power levels. The symbol rate is therefore 1/2 the bit rate. A sample 2B1Q waveform is shown in Figure 1.7.

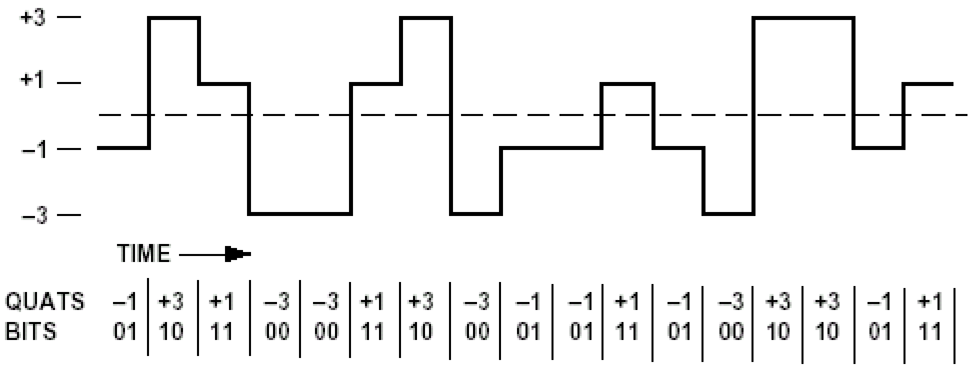

+3
+1
–1
–3
TIME
QUATS –1 +3 +1 –3 –3 +1 +3 –3 –1 –1 +1 –1 –3 +3 +3 –1 +1
BITS 01 10 11 00 00 11 10 00 01 01 11 01 00 10 10 01 11

FIGURE 1.7 An example of 2B1Q Symbols **Source:Zarlink Semiconductor**

1.3.1 2B1Q CHARACTERISTICS

The ISDN Physical Layer, is specified by the ITU I-series and G-series documents. The U interface provided for BRI is a 2-wire, 160 Kbps digital connection. Echo cancellation is used to reduce noise, and data encoding schemes (2B1Q in North America and Europe, 4B3T in Germany) permit this relatively high data rate over ordinary single-pair local loops [7].

2B1Q (2 Binary 1 Quaternary) is the most common signalling method on U interfaces. This coding rule requires that binary data bits are grouped in pairs, and each pair is transmitted as a symbol, the magnitude of which may be 1 out of 4 equally spaced voltage levels (a ``Quat'') [8].

There is no symbol value at 0V in this code, the relative quat magnitudes being g1 (the ``inner'' levels) and g3 (the ``outer'' levels). No redundacy is included in this code, and in the limit there is no bound to the RDS (RDS is the Running Digital Sum, which is the algebraic summation of all symbol values in a transmission session).

In summary, 2B1Q provides:

a) Two bits per baud
b) 80 kbaud per second
c) Transfer rate of 160 kbps

Bits	Voltage Level

00	-5
01	-3
10	+5
11	+3

TABLE 1.3 Symbol voltage levels for respective dibit

This means that the input voltage level can be one of 4 distinct levels (note: 0 Volts is not a valid voltage under this scheme). These levels are called Quaternaries. Each quaternary represents 2 data bits, since there are 4 possible ways to represent 2 bits, as in the table above.

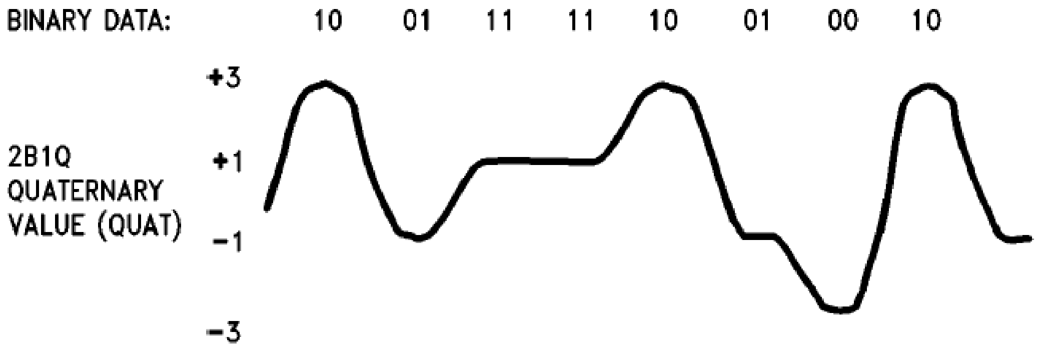


FIGURE 1.8 2B1Q Line-Coding Rule *Source : NS Conductors*

1.3.2 Frame Format

Each U interface frame is 240 bits long. At the prescribed data rate of 160 kbps, each frame is therefore 1.5 msec long. Each frame consists of:

a) Frame overhead - 16 kbps
b) D channel - 16 kbps
c) 2 B channels at 64 kbps - 128 kbps

Sync 18 bits	12 * (B_1 + B_2 + D) 216 bits	Maintenance 6 bits

d) The Sync field consists of 9 Quaternaries (2 bits each) in the pattern +3 +3 -3 -3 -3 +3 -3 +3 -3.
e) (B_1 + B_2 + D) is 18 bits of data consisting of 8 bits from the first B channel, 8 bits from the second B channel, and 2 bits of D channel data [8].

f) The Maintenance field contains CRC information, block error detection flags, and "embedded operator commands" used for loopback testing without disrupting user data.

Data is transmitted in a superframe consisting of 8 240-bit frames for a total of 1920 bits (240 octets). The sync field of the first frame in the superframe is inverted (i.e. -3 -3 +3 +3 +3 -3 +3 -3 +3). Inversion of the syncword defines the superframe boundary [8].

CHAPTER 2

SOFTWARE EMPLOYED

2.1 Electronics Workbench

Project simulations are carried out in circuit simulation software called Electronics Workbench (EWB), which can be viewed as an electronics lab in a computer.

Electronics Workbench has much tighter convergence criteria to guarantee accurate simulation. In EWB Version 5, three error tolerance quantities (relative tolerance RELTOL, absolute current tolerance ABSTOL and absolute voltage tolerance VNTOL) are combined together as the convergence criteria [9].

The software uses more accurate semiconductor device models to meet real time environment requirements. As device models model more nonlinear effects of the semiconductor devices, they could cause more numerical problems than a simpler device model [9]. This is particularly true when you simulate a circuit that employs both detailed semiconductor device models and simple ideal switch models.

Electronics Workbench runs only on Windows 95 and NT 3.51, not on Windows 3.1, 3.11 or NT 4.

2.1.1 The Digital Instruments

Each instrument can be placed in the circuit window with a click/drag operation of its icon from the instrument toolbar. Once an instrument icon is in the circuit window, connections to a simulated circuit are made on the icon. The detailed instrument provides controls that can be selected and changed just as on a real instrument.

The instruments offered by the software are as following

1. ***Multimeter*** The multimeter is used to measure voltage (ac or dc), current (ac or dc), resistance, and decibel loss.

2. ***Function Generator*** This instrument is a source of sinusoidal, triangular, or pulse waveforms whose frequency, amplitude, and duty cycle (for triangular and pulse waveforms) can be varied.

3. ***Oscilloscope*** The dual channel oscilloscope displays waveforms for measurement of amplitude and period. It has all the basic controls of an actual oscilloscope.

4. ***Word Generator*** This is a specialized EWB instrument that is used to send digital codes or patterns of bits to a circuit for testing it. As you will learn later, a word is a group of bits. Up to eight bits can be produced at a time in sixteen different patterns.

5. ***Logic Analyzer*** The logic analyzer can display up to sixteen digital waveforms in a timing diagram format.

6. ***Logic Converter*** This is another specialized EWB instrument used for making several types of conversions.

2.1.2 Simulation

The software has limitation that initial conditions can be set for nodes only, not for components [9]. During simulation subcircuits are treated as a unit. As a result we cannot directly access nodes that are in subcircuits.

Secondly the components and nodes within subcircuits cannot be specified individually for analyses. The software has option of exporting the schematic files in format compatible with other simulation softwares.

2.2 MULTISIM 2001

Multisim 2001 is a complete system design tool that offers a large component database, schematic entry, full analog/digital SPICE simulation, VHDL/Verilog HDL design entry/simulation, FPGA/CPLD synthesis, RF capabilities, Postprocessing features and seamless transfer to PCB layout packages such as Ultiboard, also from Electronics Workbench. It offers a single, easy-to-use graphical interface for all your design needs [10].

Multisim 2001 provides all the advanced functionality you need to take designs from specification to production. And because the program tightly integrates schematic capture, simulation, Printed Circuit Board (PCB) layout and programmable logic, one can design with confidence, knowing that the software is free from the integration issues often found when exchanging data between applications from different vendors.

2.2.1 Circuit Design Process

Multisim 2001 supports every step of the overall circuit design process [10], which typically includes the following phases:

1. Entering the design (using schematic capture, behavioral language formats or other methods) into the software tool being used.

2. Verifying that the behavior of the circuit matches expectations. This step is performed using simulation, and analysis.
3. Modifying the circuit design if the behavior does meet expectations, and returning to step 2 as often as necessary.
4. Depending on how the circuit is to be physically implemented, passing the design through the appropriate process. For example, if it is to be placed on a PCB, the next step is to use a PCB layout program such as Electronics Workbench's Ultiboard product. If it is to be placed on a programmable logic device (PLD, CPLD, FPGA, etc.), the next step is to use a synthesis tool such as that available from Electronics Workbench.

2.2.2 Multisim 2001 Features Summary

The software includes 9 Virtual Instruments like voltmeter, logic analyzer etc. Component Database and Editor has 6,000 parts standard, with 12,000 advanced parts. Another important feature is of design/debug and simulation facility for VHDL, also the Verilog HDL Synthesis is optional [10].

PSpice Import in the form of schematic and netlist file in other softwares is another useful aspect of Mulltisim.

2.2.3 Schematic Capture

Schematic capture is the first stage in developing your circuit. In this stage you choose the components you want to use, place them on the circuit window in the desired position and orientation, wire them together, and otherwise prepare your design. Multisim also allows you to modify component properties, orient your circuit on a grid, add text and a title block, add subcircuits and buses, and control the color of the circuit window background, components and wires.

2.2.4 Simulation

Simulation is a mathematical way of emulating the behavior of a circuit. With simulation, you can determine a circuit's performance without physically constructing the circuit or using actual test instruments. Multisim offers multiple simulators, optimized to meet the needs of various types of circuit designs and implementation. These simulators include [10]:

a) SPICE (including specialized RF simulation)
b) VHDL
c) Verilog HDL
d) Co-simulation of all three together

When designing with programmable logic devices such as Field Programmable Gate Arrays (FPGAs) or Complex Programmable Logic Devices (CPLDs), VHDL or Verilog HDL simulators have traditionally been used separately from each other and from SPICE. With Multisim, these simulators can be also used in combination. For example, to simulate a PCB designed using Multisim's schematic capture front end, Multisim uses SPICE for most of the simulation (that is, components will use SPICE models) and VHDL or Verilog HDL for modeling the most complex digital parts (including programmable devices), all brought together in the co-simulation mode. Multisim's simulation engine checks which type of model (SPICE, VHDL, Verilog HDL, etc.) is used, as indicated in the component database, and calls the appropriate simulator. It then controls the passing of information between these various simulators, all without requiring your intervention.

The type of simulation that is appropriate for a circuit depends on the type of circuit and how you plan to physically implement it. For example, analog, digital and mixed analog/digital circuits to be built as a PCB are, in general, best simulated with a SPICE simulation. Digital circuits to be implemented in Programmable Logic Devices are usually simulated at the behavioral language level, most commonly with VHDL or Verilog HDL. (See HDLs and Programmable Logic for details.)

For very complex digital devices (LSI or VLSI chips) such as microprocessors or memory, SPICE models are not usually practical, and in these cases, VHDL or Verilog HDL is the preferred solution [10].

2.3 Protel 99 SE

Protel 99 SE is designed as a "client – server" application, i.e. the main application program of Protel 99 SE, called Client99.exe, provides the basic infrastructure and user interface for Protel 99 SE, while specific services, such as editing a schematic or PCB are provided by a series of plug-in "servers" [11].

When you click on the Protel 99 SE icon in the Windows Start menu, Client99.exe is the application that is started. You do not then need to launch a schematic editor, PCB editor, etc. as separated programs - all your Protel EDA tools are available from within the Protel 99 SE desktop.

The use of a "client – server" architecture for Protel 99 SE means that you can easily expand the capabilities of the software. As well as the servers supplied with Protel 99 SE, various add-on servers are available from both Protel and numerous third-party vendors [11].

To use Protel 99 SE it is not necessary to understand how the client – server architecture works, however a basic knowledge of servers will help you get the most from Protel 99 SE.

2.3.1 Schematic Documents

A schematic is a diagrammatic representation of an electronic circuit, and schematic capture is the process of capturing a design as a schematic in a computer-aided design environment. A computer-based schematic is more than a simple drawing of the circuit. It also contains information about the connectivity of the circuit and the parts that make up the circuit.

In Protel, the basic workspace for capturing a schematic is called a schematic sheet. Electrical, drawing and directive objects are placed on a schematic sheet to design the circuit and produce working schematic drawings. A complete circuit design can use just a single sheet, or it can comprise a number of electrically linked sheets [11].

Protel allows you to create complex hierarchical and modular designs by linking any number of sheets to form a complete project.

2.3.2 PCB documents

A printed circuit board (PCB), sometimes referred to as printed wiring board (PWB), is the foundation of circuit construction. Components are soldered onto the PCB, and the PCB provides the electrical connection pathways between components to form the physical circuit. Connections are made using copper tracks etched onto the various layers of the PCB. A PCB document is displayed as a set of superimposed layers, with each layer corresponding to an individual "phototool" used to fabricate the board [11].

In general, a PCB is derived from a schematic representation of the circuit. When a schematic is loaded into a PCB document, schematic part symbols are translated to corresponding board component footprints, and the connectivity of the schematic is preserved and displayed as connection lines in the PCB document.

2.3.3 *Synchronizing SChematic & PCB Documents*

Protel99 includes a powerful design synchronization tool, which makes it very easy to transfer design information from the schematic to the PCB (and back again).

Synchronizer is used to initially transfer your design from the schematic documents to a PCB design document, and also to synchronize design changes made in either the PCB or the schematic documents.

The Synchronizer automatically extracts the component and connectivity information from the schematic, locate the required footprints in the PCB libraries and place them in the PCB workspace, then add the connection lines between connected component pins [11].

2.3.4 PCB Design Rules

PCB is designed by placing components, tracks, vias and other design objects. These objects must be placed in the workspace with regard to each other. Components must not overlap, nets must not short, power nets must be kept clear of signal nets, etc [11].

To allow an individual to remain focused on the task of designing the board, Protel 99 SE can monitor these design requirements for one. We can instruct the PCB editor of our requirements by setting up a series of design rules. These design rules are monitored as our layout PCB. As soon as an object is placed in violation of a design rule it is highlighted. Also, during the board verification process we can run the integrated Design Rule Checker, which will generate a report of any design rule violations in you PCB.

Protel allows a wide range of design rules to be defined for a PCB. These include clearances, object geometry, parallelism, impedance control, routing priority and topology, placement rules, and signal integrity rules. Each rule has a Rule Scope that defines how it is applied. The scope allows you to apply a rule to objects, nets, net classes, components, component classes, layers, regions, through to the whole board [11].

2.3.5 Autorouting a PCB

Protel 99 SE provides an easy to use, powerful, high-quality, shape-based autorouter, tightly integrated with the PCB design editor. When you run the autorouter, the board is routed directly in the PCB window, and adheres to relevant design rules set for the board.

In general, the autorouter does not need the set up of any options, as it will analyze the current PCB design and automatically select the most appropriate autorouting strategy. It is important, however, to ensure that any relevant design rules are set up prior to running the autorouter [11].

2.3.6 Verifying the PCB Design

Design verification is the process of investigating the PCB design for errors or potential problems before proceeding to manufacturing.

Protel is a rules-driven PCB layout environment and includes comprehensive design rule checking. Many of the design rules can be monitored and enforced on-line as you work, helping to prevent design rule violations during placement and routing. One can also manually run a design rule check (DRC) in batch mode at any time in the design process to check for conformance to specific rules [11].

To help solve signal integrity rule violations, Protel 99 SE includes a powerful Signal Integrity Analyzer that allows you to perform reflection and crosstalk simulations on your PCB [11].

2.3.7 Generating the Manufacturing Files

Completing the PCB layout is the first part of the process that culminates in the fabrication and assembly of your PCB. The link between your design and the finished board are the print, Gerber and NC drill fabrication files, as well as the Bill of Materials, testpoint report, and pick and place assembly files. Apart from printed output, all the PCB manufacturing output files are generated by the CAM Manager [11].

2.3.8 Interfacing A PCB Document with Third-Party Tools

The main method of interfacing from a PCB document to third party tools is the netlist. Netlists come in many different formats, but are usually generated as ASCII text files which carry at least two types of information:

1. Descriptions of the components in the design.

2. A list of all pin-to-pin connections in the design.

Some netlist formats combine both sets of data in a single description, others, including Protel, separate the two data into separate sections.

As straightforward text files, netlists are readily translated into other formats using a simple, user-written program. Netlists can also be created (or modified) manually using a simple text editor or word processor [11]. The PCB Editor can load Protel, Protel2 and Tango format netlists.

Another feature of Protel is that if both the schematic and PCB are being designed in Protel99 you do not need to use a netlist to transfer the design information. Protel99 includes a powerful design synchronization tool that automates the transfer of design information from schematic-to-PCB, and from PCB-to- schematic.

CHAPTER 3

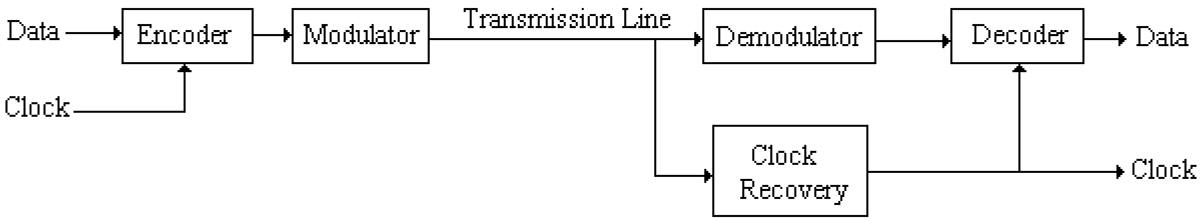
HDB3 CODEC

3.1 Introduction

This section covers the design and realization of HDB3 codec pair using modular approach. The design of other circuits like clock, clock recovery are necessary for system operation.

The sub-systems necessary for a functional communication system are illustrated below.

Figure 3.1 A Typical Communication System



The design and simulation of each of these sub-systems will be covered in the following sections.

3.2 Design Constraints

The encoder and decoder must be implemented on the Electronic Workbench 5.0a and Multisim 2001, which has limited inputs, outputs and internal logic complexity.

The design may face problems in real time hardware implementation but to be useful the encoder and decoder must operate at a minimum clock speed of 2 MHz.

3.3 Design Report

3.3.1 Design Methodology

Modular design approach will be adopted which is likely to produce optimal results. Also, modular design is creative and easy to understand. The modules are simulated and outputs analyzed to get optimum results. The two fundamental portions are further broken down into small modules.

3.3.2 Codec Pair

The Codec pair consists of HDB3 encoder and decoder. Before beginning to design the encoder and decoder some assumptions were made, these are:

a) Codec output to be full width (allows maximum flexibility of line modulation).

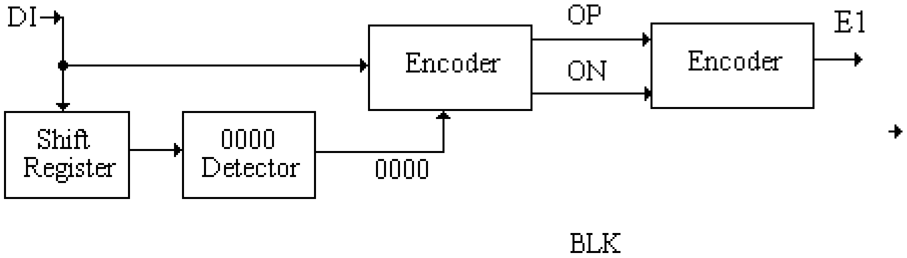
b) Received signal will be identical to the transmitted signal (an ideal transmission medium is error free).

3.3.2.1 HDB3 Encoder

To simplify the encoder design it was split up as follows:

Figure 3.2 HDB3 Encoder Sub Sections

3.3.2.2 HDB3 Decoder



The decoder is also broken down for simplicity into the following units:

Figure 3.3 HDB3 Decoder Sub Sections

When BLK is set high the "Block Output" section will set Z to low, if BLK is low Z follows the output of the last shift register sage.

3.4 Functional Decomposition

3.4.1 Encoder

For simplicity of design a slightly different form of coding will be used within the encoder, which can then be translated to the necessary OP & ON code. The code will consist of 2 bits, MK and POL.

When, MK=1 corresponds to a non-zero output and
POL to the polarity of the pulse produced (POL=1 positive pulse, POL=0 negative pulse).

These relate to ON & OP as follows:

MK	POL	OP	ON
0	0	0	0
1	0	0	1
0*	1*	X	X
1	1	1	0

*This state will NOT occur and X is the "don’t care" state.

TABLE 1.1 MK & POL Relationship

Giving the simple logic relationships:

OP = MK . POL

ON = MK . POL*

The design is now split into following functional parts:

a) Detecting 0000

b) Ascertaining next/last mark polarity.

c) Number of valid pulses (odd or even) since last V pulse, not including first V pulse.

d) Summer network

3.4.1.1 Detecting 0000

Four samples of the input bitstream must be made, each separated by 1 clock cycle from the previous. By using a shift register it is possible to separate input bits by a single clock cycle for each flip-flop, as data is clocked on the rising edge of the clock. By using a 3-stage shift register and a direct feed from the input it is simple to look for 0000, as in the following circuit:

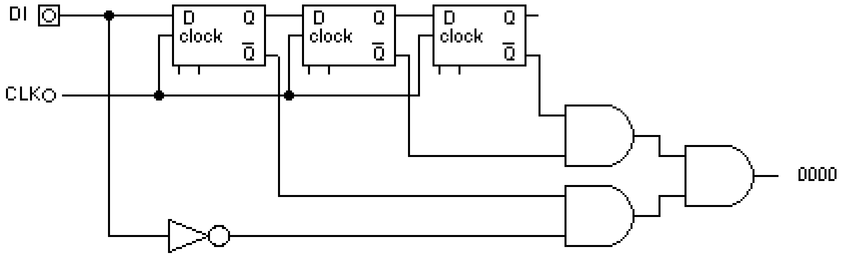


Figure 3.4 0000 Detector

The above circuit will detect any set of 0000, so a sequence of 00000 would cause two detections. To prevent this the detection of 0000 should be disabled until the shift register is clear of the original 4 0s.

It can also be seen that as soon as 4 0s have been detected it must be decided whether to issue a B pulse or not. It is known that a V pulse must be issued in the same slot as the most recent 0 (i.e. 3 clock pulses later). Both of these requirements can be simply achieved by using the 0000 condition to start a (non-resettable) counter to count 3 clock cycles.

This requires only a slight design effort, it will be a 2 bit counter as 4 states are required (quiescent and 3 count states). It was arbitrarily decided to design it as a standard up counter. The logic is as follows:

Current State			Next State	
0000 Detected	**Q1**	**Q0**	**D1**	**D0**
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

TABLE 3.2 Logic for 000 detection

This reduces to:

D1 = Q1 XOR Q0

D0 = (Q0* . 0000 Detected) + (Q1 . Q0*)

The timing for V pulse generation is simply generated by ANDing both flip-flop's outputs, and is shown below.

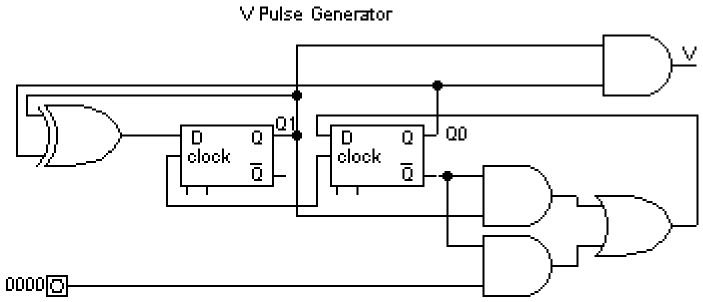


Figure 3.5 V Pulse Generator

3.4.1.2 Ascertaining next/last mark polarity

This part of HDB3 is identical to AMI so the selectable toggle flip-flop circuit can be used for this. This should only be toggled by valid data 1s or by B pulses, as in the next circuit.

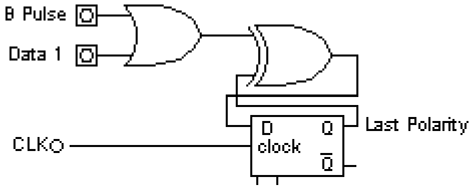


Figure 3.6 Last Polarity Tracker

3.4.1.3 Number of Valid Pulses since Last V Pulse

This can be split into two cases:

a) First V pulse will not have an associated B pulse.

b) If an even number of valid pulses have occurred since the last V pulse a B pulse will be issued.

To prevent the first V pulse triggering a B pulse it is necessary to have some sort of memory of whether a V pulse has previously occurred. As this is a one shot memory a latch can be used, whose logic table is:

Input (T)	Current State (Q)	Next State (D)
0	0	0
1	0	1
0	1	1
1	1	1

TABLE 3.3 Logic for Violation detection

Which reduces to

D = T + Q

To ascertain the odd/even number of valid pulses what is required is a 1 bit counter which a V pulse can reset. The necessary states are:

V Pulse	Valid Pulse (B OR 1)	Last State (Q)	Next State (D)
0	0	0	0
1	0	0	0
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	0
1	1	1	0

TABLE 3.4 Logic to ascertain the odd/even number of valid pulses

This reduces to:
D = V*. (Q XOR Valid Pulse)
Combining these two segments and ANDing both flip-flops outputs with 0000 detector gives the desired balance pulse, as in the following circuit.

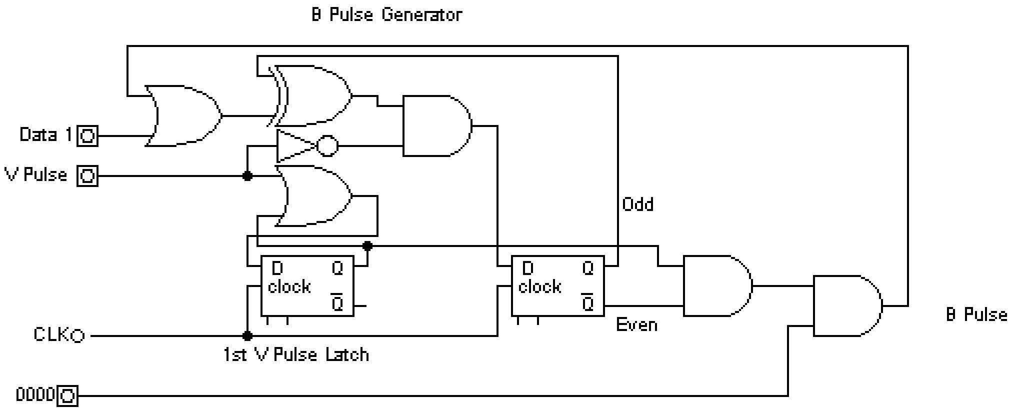


Figure 3.7 B Pulse Generator

3.4.1.4 Summing up OP and ON for generation of HDB3 signal

As, OP represents positive half of the output and ON the negative half, both need to be summed to give a synchronized output.

This is done by using a giving OP to an inverting amplifier ,ON to non inverting amplifier and finally giving output of both amplifiers to a difference amplifier.

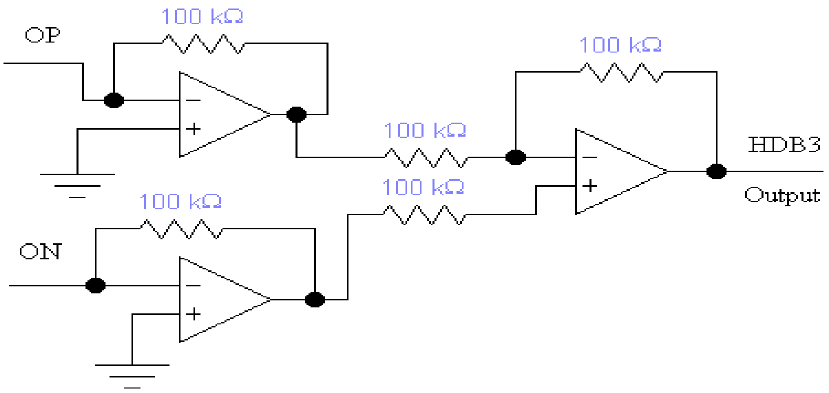


FIGURE 3.8 Summer network

3.4.1.5 Assembling the Sub-Circuits

As we wish to emit pulses in the following cases:

a) When a data 1 has reached the end of the shift register

b) When a B pulse is generated

c) When a V pulse is generated

To get the correct outputs in all three cases MK should be set, a simple OR operation.

The value of POL is determined by the case, if a data 1 or B pulse is encountered POL should be set to the opposite of the last POL issued i.e. Q* output. The toggle flip-flop should then be toggled. If a V pulse is encountered POL should be set to the same value as the previous POL and the flip-flop not toggled.

To allow selection of HDB3 operation through CS1 it is necessary to disable the production of B and V pulses. This can be achieved by de-gating the detection of the 0000 condition and the B pulse generator with AND gates.

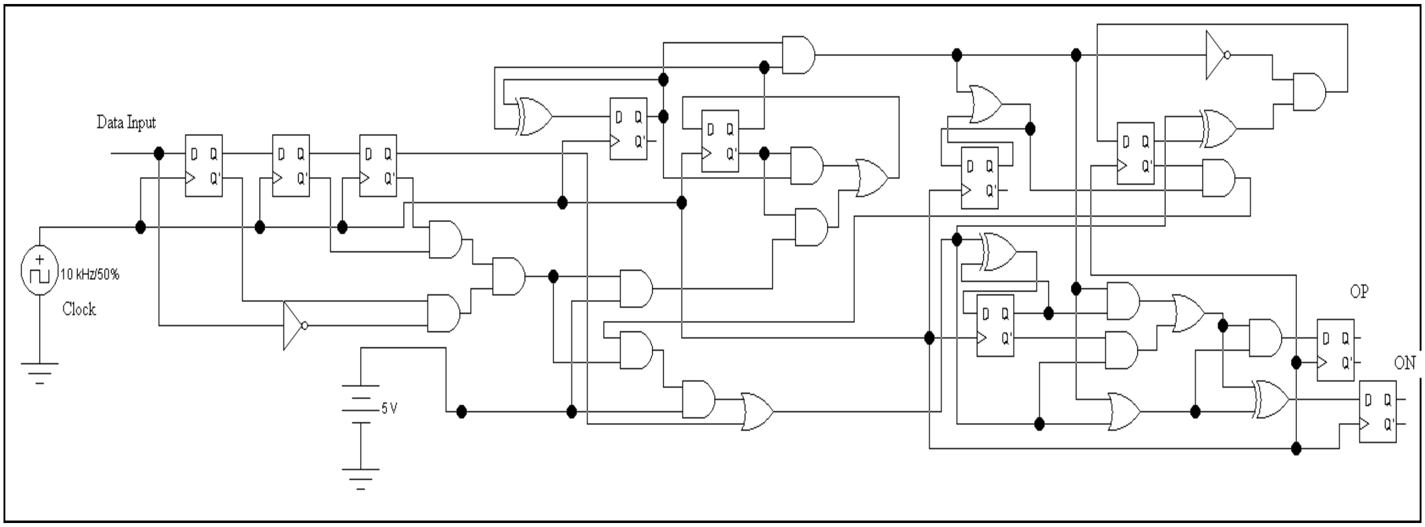


Figure 3.9 HDB3 Encoder

To provide the full-width output desired the ON and OP signals should be buffered by flip-flops. Alternatively it is also possible to AND these signals with the CLK to give half width output.

Implementing this additional circuitry with all the sub circuits gives the complete full width encoder circuit as shown above.

3.4.1.6 Issues Arising from Encoder Design

On a hard reset the shift register will be full of 0s so will add 000 to the start of the decoded bit-stream, and a delay of 3 clock cycles on link user data. When switching to HDB3 mode, after any previous use of HDB3 mode, the latch used to determine if any V pulses have occurred previously should be reset. However the only way to reset a flip-flop is to asynchronously reset *all* flip-flops, which will destroy the memory of the last pulse polarity and could cause erroneous operation. This would have the same effect as a hard reset, either of which would require the resetting or restarting of the decoder.

An alternative mode of operation is possible, but this does not meet exactly the HDB3 specification. If a B pulse is allowed for the first V pulse the circuit can be simplified, the latch for remembering the first V pulse can be removed. This would remove the need for resetting between uses of HDB3 mode.

The encoder can be altered to encode HDBn codes by setting the shift register length to n stages. The counter must also be changed to count 0 to n, with appropriate logic to produce a 1 on the count of n.

3.4.2 Decoder

Similarly to the encoder design the OP & ON code will be converted, at the input to the decoder, into MK and POL. In this case the conversion is very simple, MK = IP + IN and POL = IP.

3.4.2.1 Design Requirements

The decoder needs to again split the input into two levels as IP and IN so that the information carried in both the halves is interpreted. For this purpose class c amplifiers are used which split the positive and negative halves by operating only for half cycle i.e. 180 degrees.

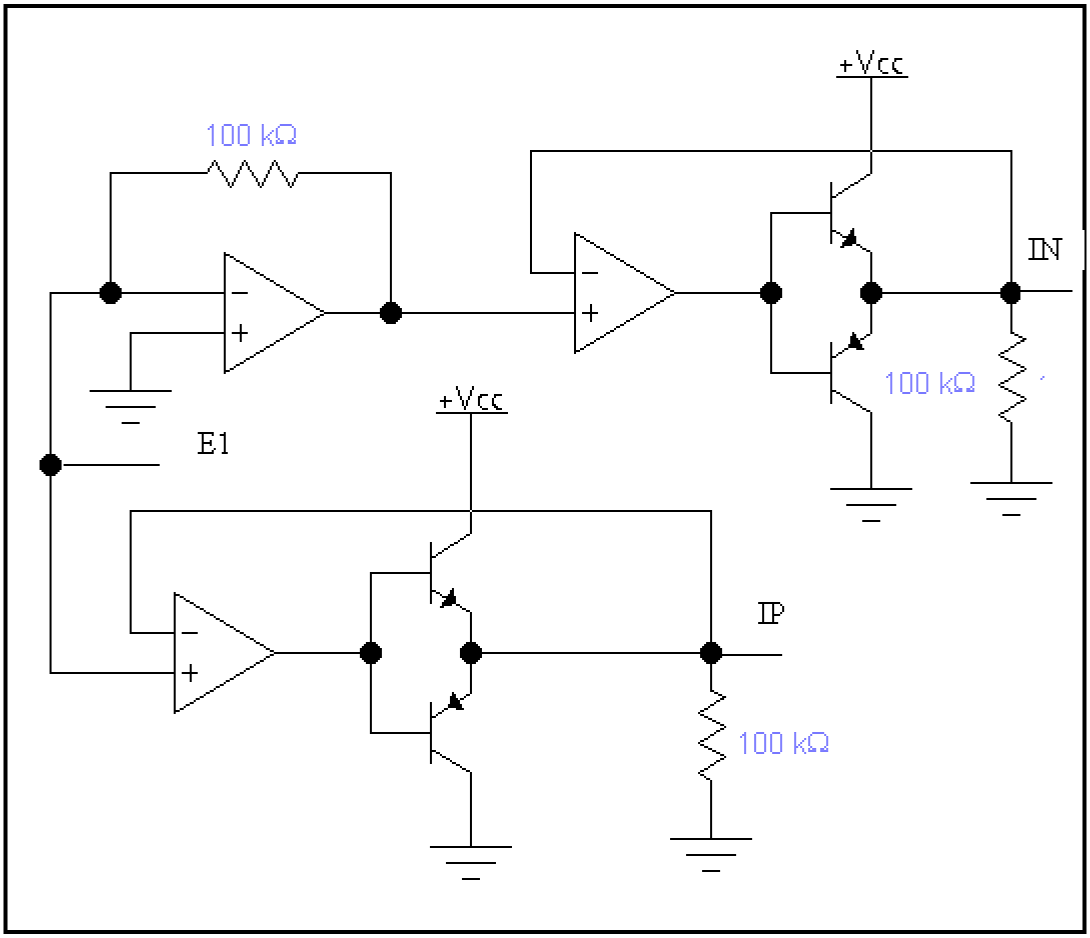


FIGURE 3.10 Extraction of IP and IN by Class C amplifiers

When decoding HDB3 the only guaranteed indicator of a run of 0s are V pulses, which can be easily detected by comparing the output of the toggle flip-flop with POL. It is known that three clock cycles before the V pulse a B pulse may have been emitted which will be decoded as a 1 but should be converted to a 0.

To allow removal of B pulses the output of the decoder should be stored for three clock cycles and 1s removed when V pulses received. This is easily accomplished by using a three-stage shift register and then de-gating the delayed output with the V pulse detector. The output is then buffered, to produce a glitch free output, giving the circuit as follows:

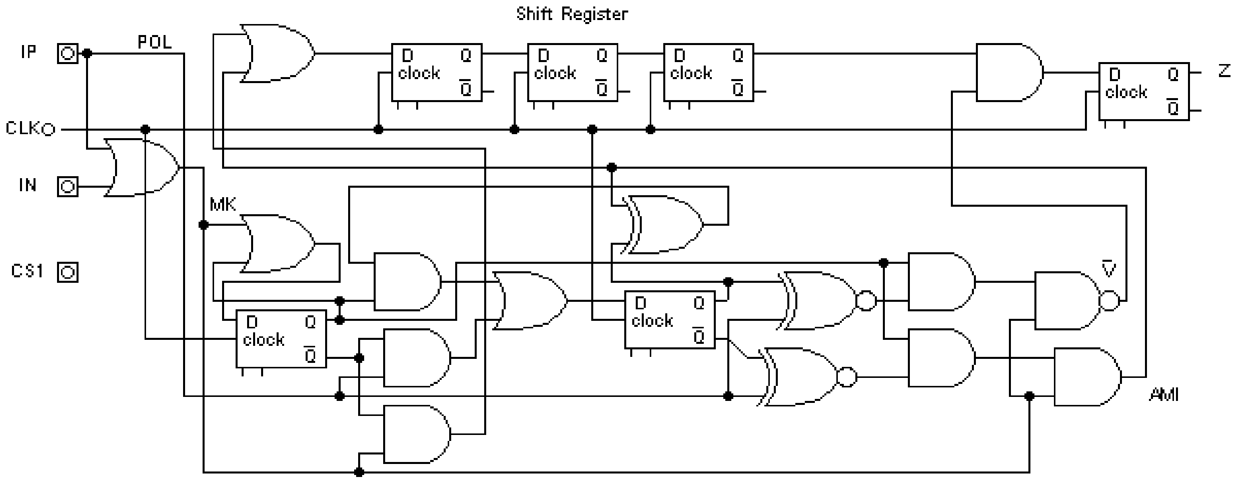


Figure 3.11 HDB3 Decoder

3.4.2.2 Issues Arising from Decoder Design

The decoder does not make use of the CS1 input as HDB3 codes are compatible, if there are no errors introduced between encoder and decoder.

The decoder described adds 000 to the start of the bit-stream, and delays the link user data by 3 clock cycles. As stated earlier the first non-zero input must be a valid pulse (B or data 1). If the encoder used is the one described above, in HDB3 mode, the first data input should be 1 (0 would lead to giving a V pulse as the first non-zero output). This would lead to a total link overhead of 7 bits (0000001) for HDB3.

If the encoder design is known the polarity of the first valid pulse would be known and the flip-flop used to latch onto the polarity disposed of.

The decoder can be altered to decode a HDBn code by ensuring that the shift register has n stages.

3.4.2.3 Error Detection

It is possible to detect errors in the input bit stream and either correct them or let the link user know there is an error present.

Several errors can be simply checked for in a HDB3 encoded stream, these include:

3.4.2.3.1 Greater than 3 Consecutive 0s

More than three consecutive 0s - ANDing the inverse outputs of the shift register with MK and the inverse of POL can detect four 0s, a clear violation of the HDB3 code.

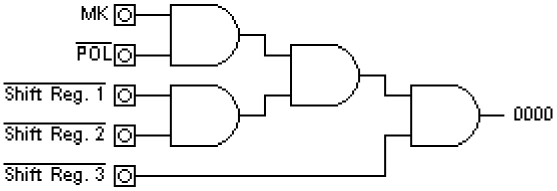


Figure 3.12 > 3 Consecutive 0s

3.4.2.3.2 Greater than 1 V Pulse

More than one V pulse in any four inputs – detectable by using the V pulse detector as an input to a three stage shift register and then detecting 2 or more cases.

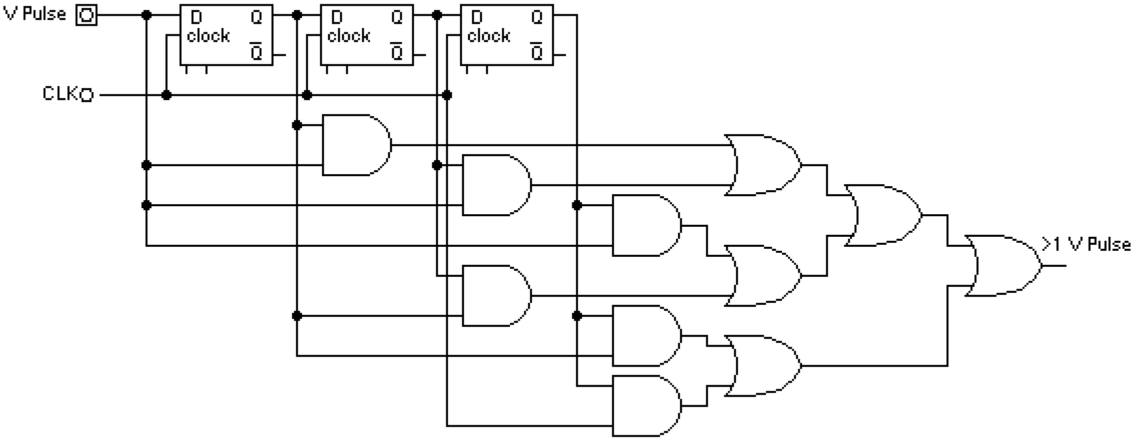


Figure 3.13 >1 V Pulse

3.4.2.3.3 Misplaced V pulse (XXV) Detector

Misplaced V pulse – a non 00V (i.e. XXV) sequence can be detected by ANDing the V pulse detector output with either of the inverse of the first 2 stages of the shift register.

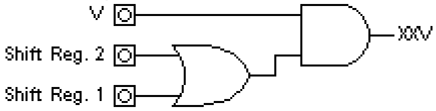


Figure 3.14 XXV Detector

The above error detection schemes must only produce an error output when HDB3 mode is in use so a simple ANDing of the sum of their outputs with CS1.

In all these cases the position of the error is not known, it could be at the start or end of the sequence checked. To adequately warn the link operator any error indicator should indicate that there might be an error over the output time span of the checked bits.

3.5 Realizing the Design

Having designed the circuits they must be validated, simulated and compiled. The main tool available for this is the Electronic Workbench (EWB 5.0a) environment.

3.5.1 Validation

Having entered the circuits into EWB a preliminary compilation was made to check that the logic used was not too complex for the software to implement.

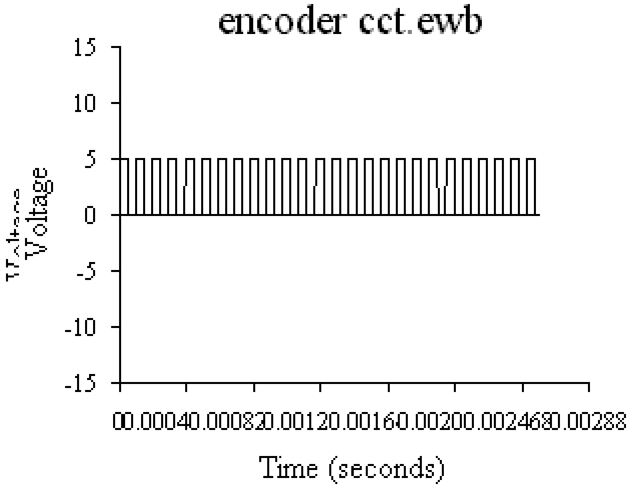
This proven not to be the case and the logic of both encoder and decoder were well within the capabilities of the EWB.

3.5.2 Simulation

To verify that the circuit would perform as desired it was necessary to simulate it. To test the operation fully requires a set of test vectors that cover all the necessary functionality.

For simplicity we will create a single comprehensive vector for the encoder and use it’s corresponding encoded output as the test vector for the decoder. The simulation will be carried out with a clock rate of 2 MHz, the basic E1 bit rate.

FIGURE 3.15 Data input (DI)



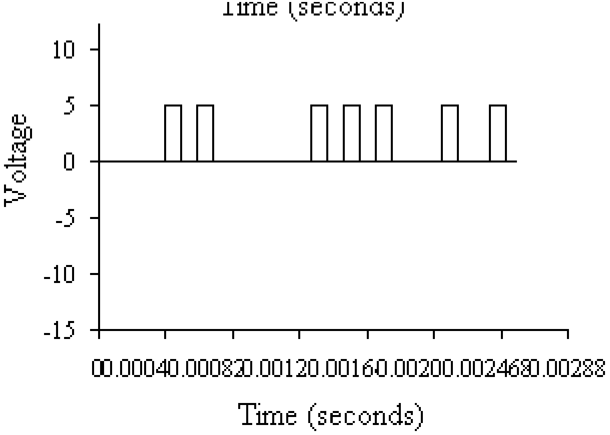


FIGURE 3.16 Clock signal (Clk)

FIGURE 3.17 Output at OP

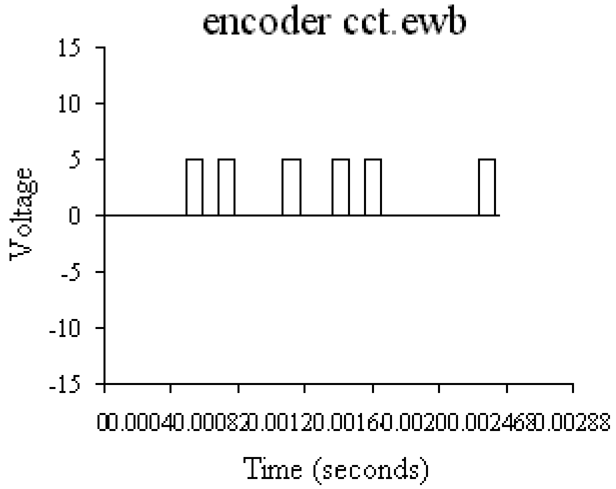


FIGURE 3.18 Output at ON

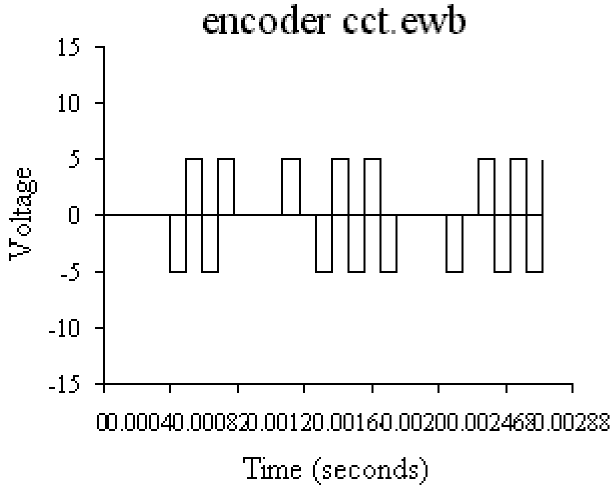


FIGURE 3.19 HDB3 output

The test vector for HDB3 is more complex and must test several cases. Several sets of 0000 must be included to check for no B pulse for first V pulse and B pulses issued with V pulses when an even number of valid pulses have occurred since the last V pulse (noting that 0 is an even number).

Injecting the same signals into the decoder gave the traces shown below:

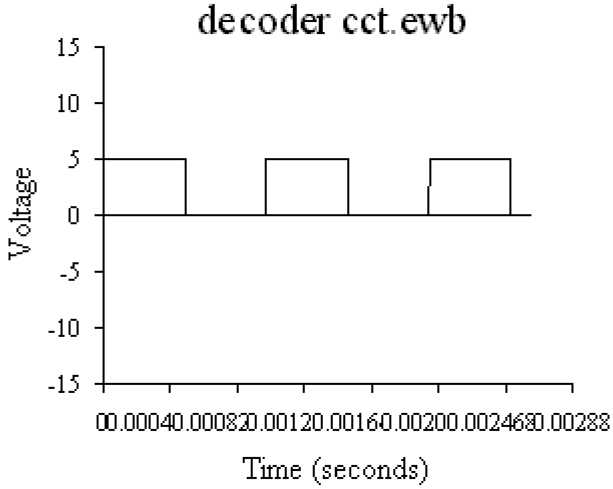


Figure 3.20 HDB3 Decoder Simulation

These demonstrate that the correct operation was attained as the original test vector is reconstituted.

3.5.3 Testing

The tests carried out in the Electronics Work Bench and Multisim include:

a) HDB3 all 0s.

b) HDB3 all 1s.

The encoder did not pass the HDB3 all 1s test on its first attempt, but a second test was successful. The encoder was then tested for speed of operation and reached a clock rate of 2 MHz (the maximum testable) with no problems.

The decoder passed all tests first time with the encoder described above and a pre-existing version. The decoder reached a maximum speed of operation of 2 MHz before errors occurred.

It may be worth noting that the encoder/decoder of project team erred at the same speed so it may be due to a hardware problem.

The input and output pins have been set as follows for the encoder:

Name	Description
CLK	Clock signal
CS1	AMI/HDB3 (0/1) mode select
DI	Data input
OP	Positive line voltage output
ON	Negative line voltage output

For the decoder:

Name	Description
CLK	Clock signal
CS1	AMI/HDB3 (0/1) mode select
IP	Positive voltage input
IN	Negative voltage input

Z	Data output

CHAPTER 4

2B1Q CODEC PAIR

4.1 Design Methodology

Modular approach is adopted for the circuit designing because it's creative and easy to understand. Simulations are carried out in *Electronic Workbench 5.0a* and *Multisim 2001* to get better understanding of the on going processes and design parameters. A main disadvantage remains that the simulations may not fit hardware and they may generate unwanted and spurious signals on assembly of modules

There are constraints on the amount of circuitry that can be realized *on Electronic Workbench 5.0a*, so some of the work is done in *Multisim 2001*. These may impinge upon the design and efforts are made to minimize the amount of circuitry used by each section design.

4.2 2B1Q Encoder

Before beginning to design encoder and decoder some assumptions are made, these are

a) deal conditions i.e. ideal error free transmission medium is available

b) Received signal will be identical to the transmitted signal

Following are the main functional parts required for the design

a) Serial to parallel converter

b) Logic code

c) Adjustment of Voltage Levels

d) Summer

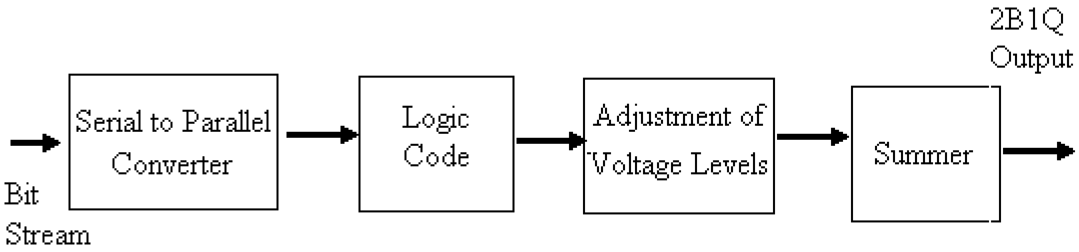


Figure 4.1 Block Diagram

4.2.1Serial to Parallel Converter

As the output of the HDB3 decoder is in the form of bit stream and to encode them we need to get them in the form of pair of bits. This is done by converting the serial bit stream pattern into two parallel bits. Another advantage is that serial transmission from one digital system to another reduces the number of wires in the transmission line.

The simplest solution is to use flip-flops, which shift the data in required order. So a pair of D flip-flops are used for the desired results with the flip-flop having two inputs Data input (D) and clock signal (C). The output at the Q of each flip-flop toggles with every falling edge of the clock.

Inputs		**Outputs**	
D	CLK	Q	Q*
0	0	NO CHANGE	NO CHANGE
0	1	0	1
1	1	1	0

TABLE 4.1 Logic Table

D latch has ability to hold data in its internal storage. The binary information present at the data input of d latch is transferred to the Q output when control input is enabled (logic 1). The output follows the changes in the data input as long as the control input is enabled. When control input is disabled logic (0), the binary information that was present at the data input at the time transition occurred is retained at Q output until the control input is enabled again.

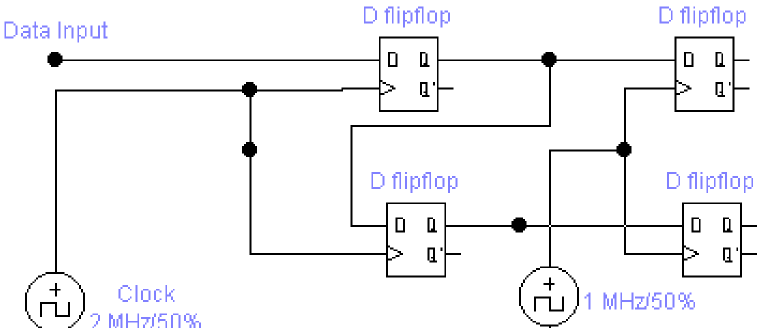


Figure 4.2 Serial to parallel converter

Same circuitry is used in making other D flip-flop as well, the only difference is that clock input given to this latch is inverted. The purpose is once we have a binary bitstream coming in serial form every latch keeps taking one bit and at the same time ignores the subsequent bit.

The rate of clock and data are set at the same speed so first latch is getting enabled on positive pulse and second flip-flop is enabled at the negative pulse. So, bitstream is available in the form of binary pair.

4.2.2Logic code

The 2B1Q logic specifies four distinct symbols having predefined voltage levels. So, to get four different voltage levels we need to split the bit stream on the basis of dibit as done in above step. The logic use to get the required distinguished levels is given below

INPUTS		**OUT PUTS**			
A	B	P	Q	R	S
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

TABLE 4.2 Logic Code

So, the table ends up giving a very simple logic

P = (A + B)*

Q = A*.B

R = A.B*

S = A.B

The logic diagram of the encoder module is shown as

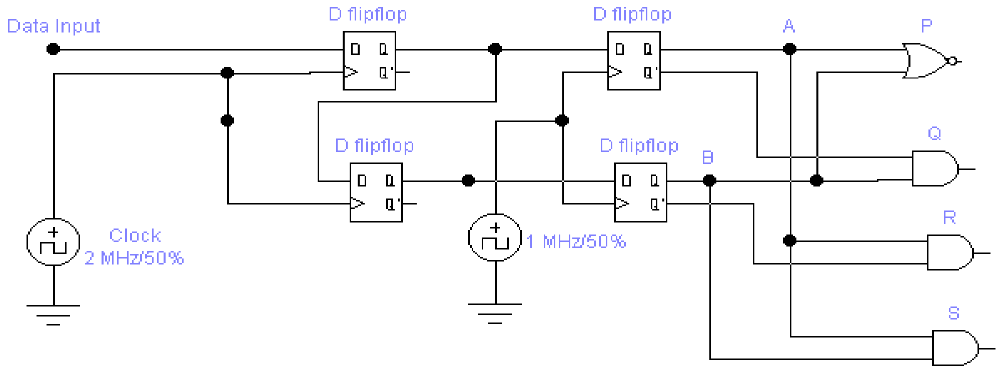


Figure 4.3 logical encoding diagram

4.2.3 Adjustment of Voltage Levels

After detection of the required logic sequence the information is to be transmitted in the form of a symbol i.e. predefined voltage level.

A module consisting of a combination of inverting and non-inverting amplifiers is used to assign the specified voltage level as per requirement, the gains of both inverting an non inverting amplifiers are adjusted to four distinct levels L1, L2, L3 and L4 by varying the feed back gain.

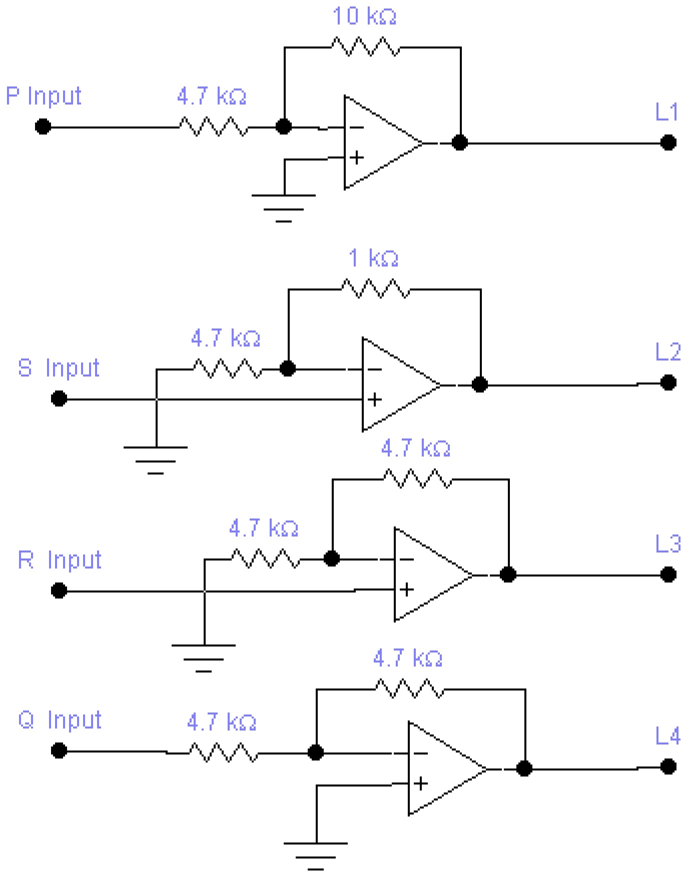


Figure 4.4 voltage assignment module

Gain for inverting amplifiers is given as

Vo / Vs = - R2 / R1

Similarly, for non-inverting amplifiers the relationship stands

Vo / Vs = 1 + R2 / R1

4.2.4 Summer Network

Until previous stage the processing on signal is carried out separately. Now to get the output in serial form again without any loss of information summer network is used. So the input voltage levels L1, L2, L3, L4 are summed up by the network keeping the ratio as per equation given below

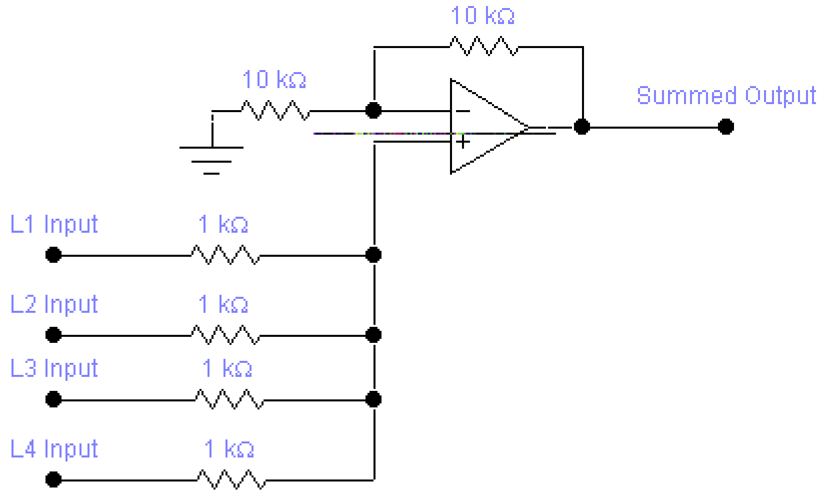


Figure 4.5 Summer Network

Vo = 8 (L1 + L2 + L3 + L4)

Note that each summing coefficient may be independently adjusted by changing the corresponding value of resistors.

4.2.5 Assembling the Sub-Circuits

Implementing the additional circuitry with all sub circuits gives the complete encoder circuit as shown below.

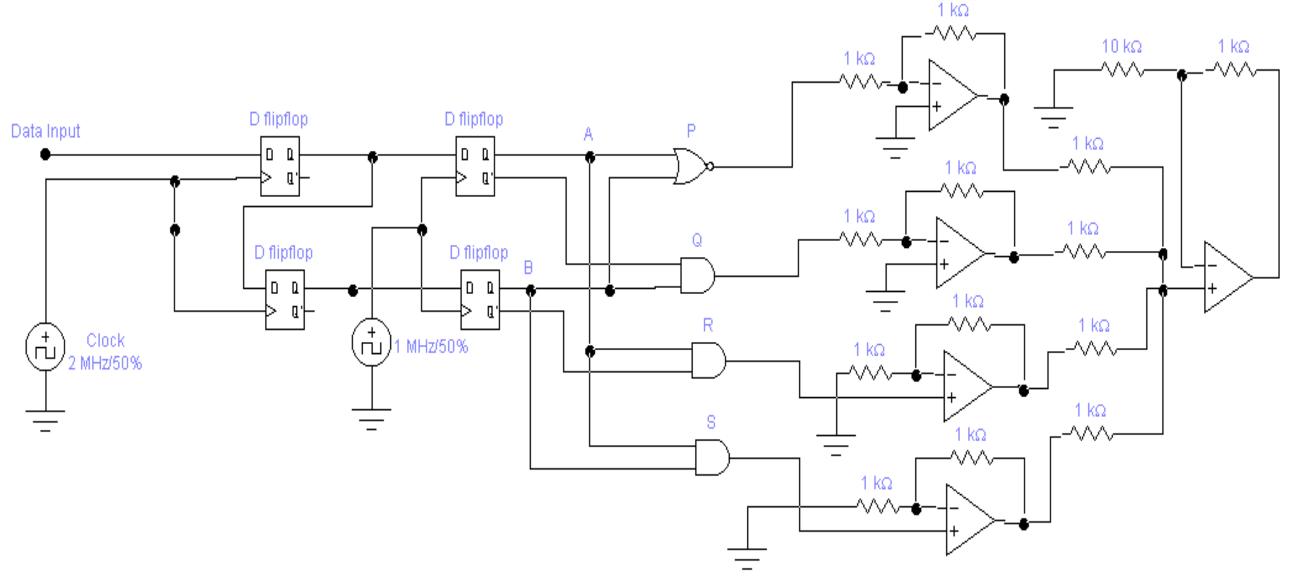


Figure 4.6 2b1q encoder diagram

4.2.6 Issues Arising from Encoder Design

As the shift register used will give a delay of two clock cycles at the start of decoded bit stream. So, the synchronization may get disturbed resulting in an erroneous interpretation at the other end.

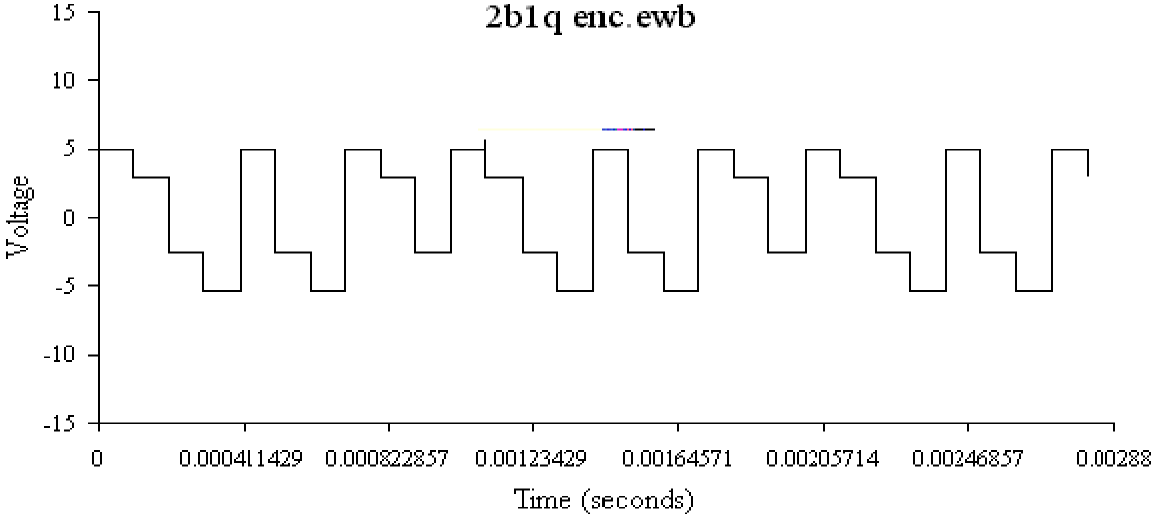


Figure 4.7 2B1Q Simulation Result

Secondly the inverting and non-inverting amplifiers used in the design can result in fluctuation of voltage levels. The error caused by them may cause the complete dibit or few dibits destroyed in the decoded bitstream.

4.3 2B1Q Decoder

The decoder module comprises of

a) Difference Amplifiers

b) Comparator

c) Logic code

d) Parallel to serial conversion

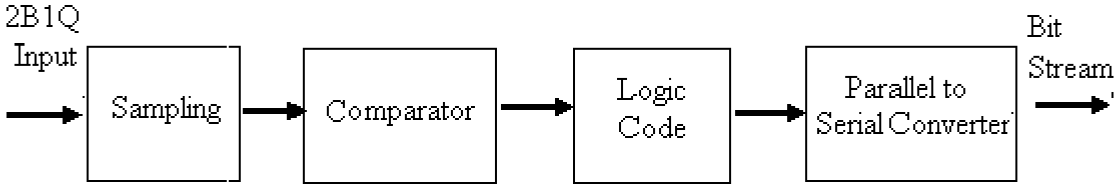
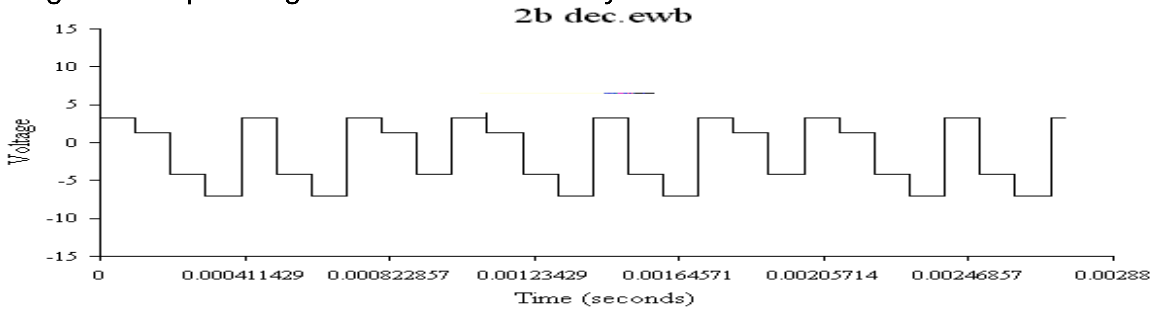


Figure 4.8 Block Diagram 2B1Q Decoder

4.3.1 Difference Amplifiers

The objective of this stage is to first identify received voltage levels from encoder stage and separating each level individually.



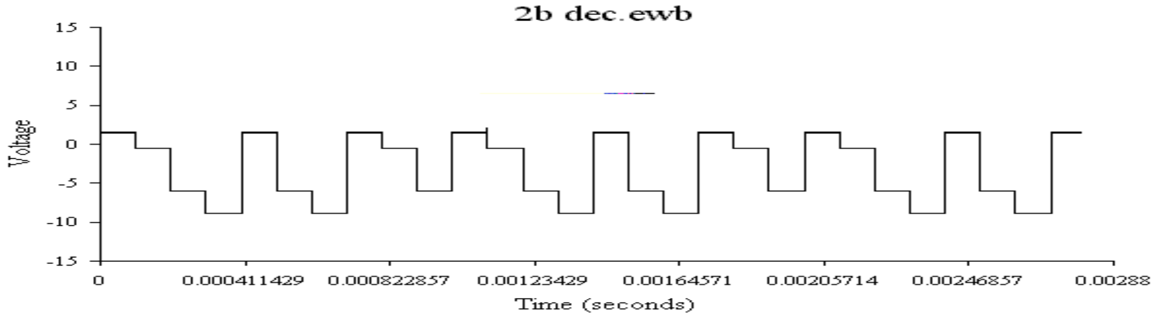


FIGURE 4.9 level obtained at "LEVEL 1"
FIGURE 4.10 level obtained at "LEVEL 2"
FIGURE 4.11 level obtained at "LEVEL 3"

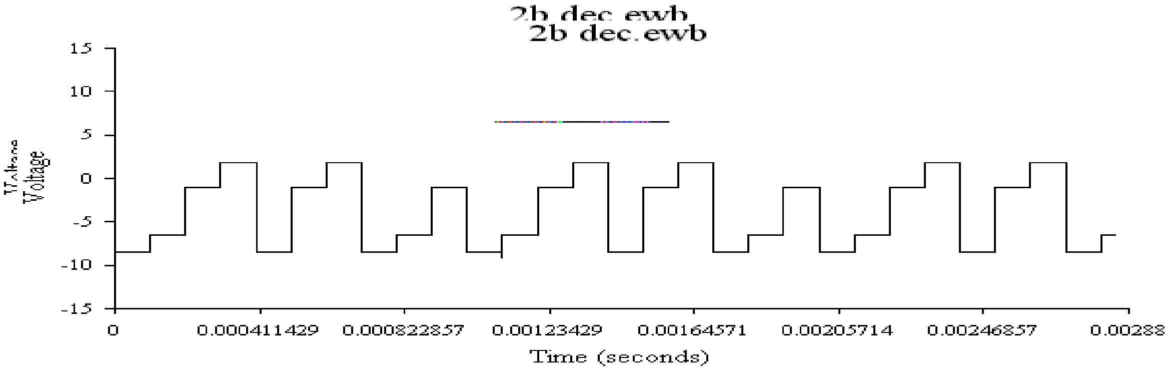


FIGURE 4.12 level obtained at "LEVEL 4"

The signals at receiver end can only be among four levels as defined in the encoder module. The received signal in the form of symbols are sampled according to a defined set of reference voltages.

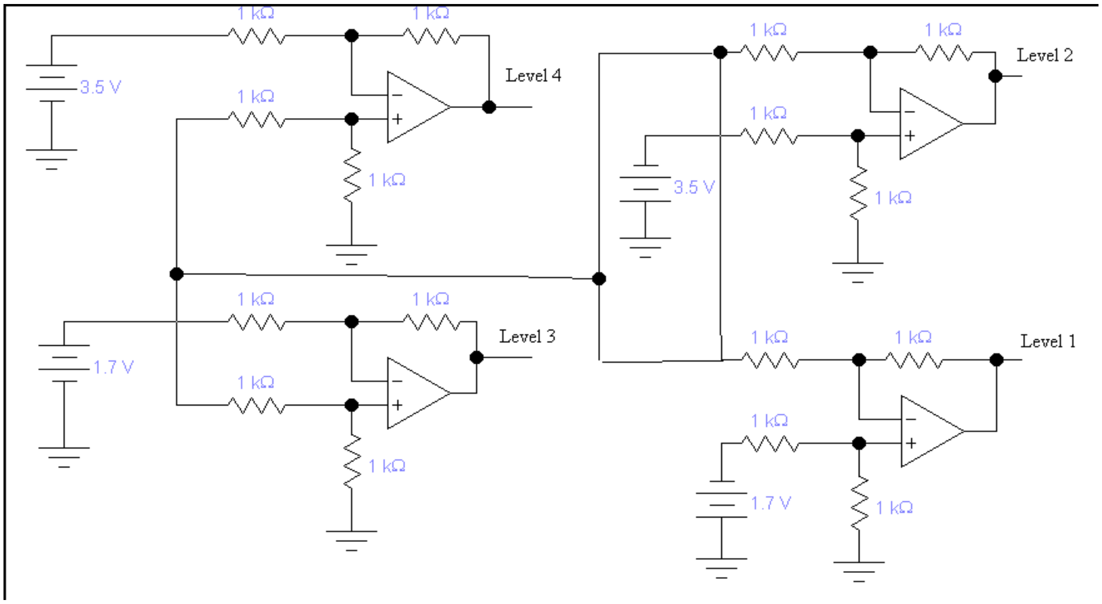


FIGURE 4.13 Different reference levels required for sampling the input

Four difference amplifiers for this purpose, with their gains adjusted with the value of resistance used. We kept four separate threshold levels for differentiating between the input d signal levels. These distinct levels are then given to next stage.

4.3.2 Comparator

The signal received in previous stage indicating the dibit pattern are not of significant numerical value. In order to make the detected signal to cross threshold level required to operate the logic gate ICs following arrangement is used.

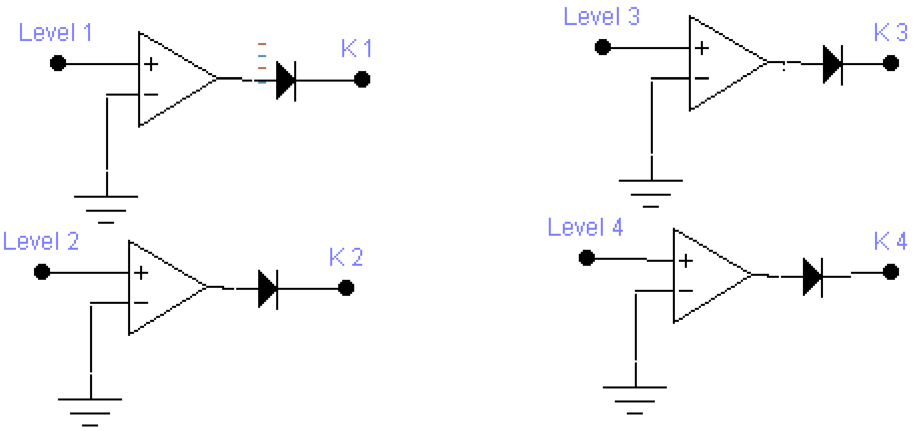


Figure 4.14 Set of Comparators

The comparators gives the signal required level and the diodes rectify i.e. eliminates the negative portion of the signal making it acceptable in the form of logic levels.

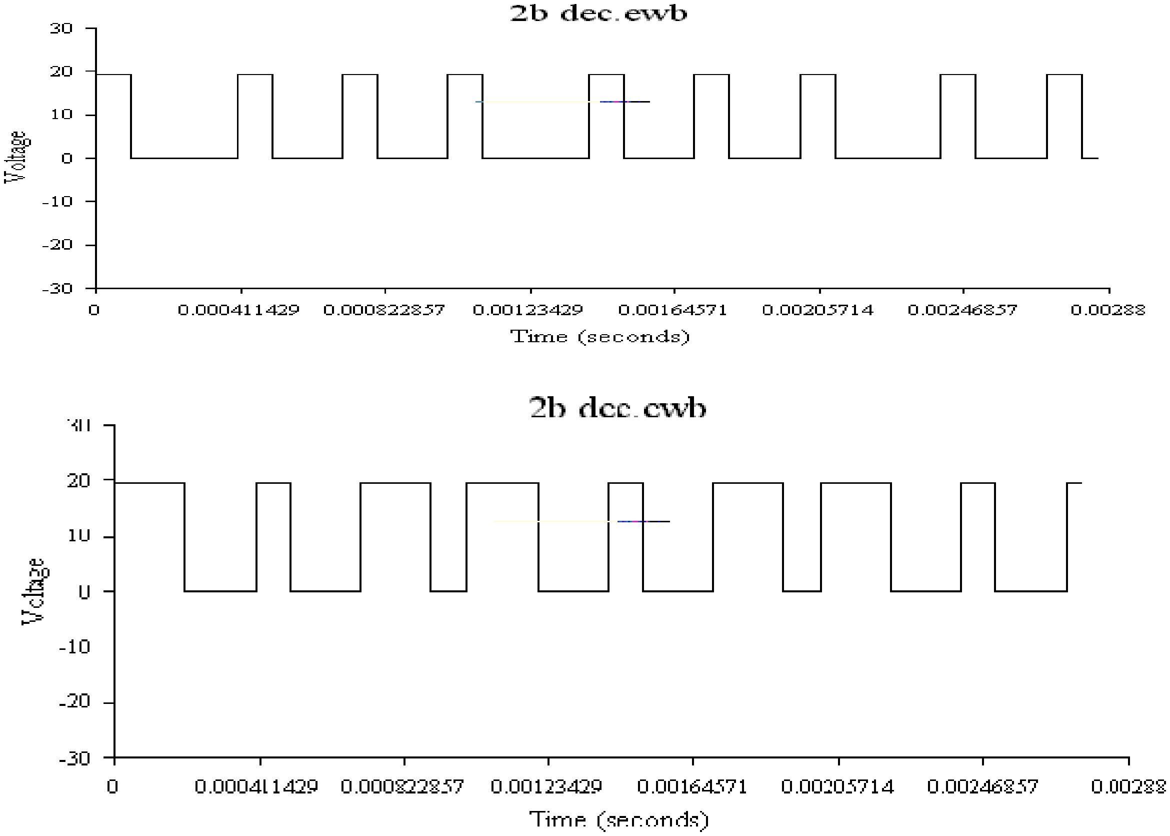


Figure 4.15 Comparator output at K1

Figure 4.16 Comparator output at K2

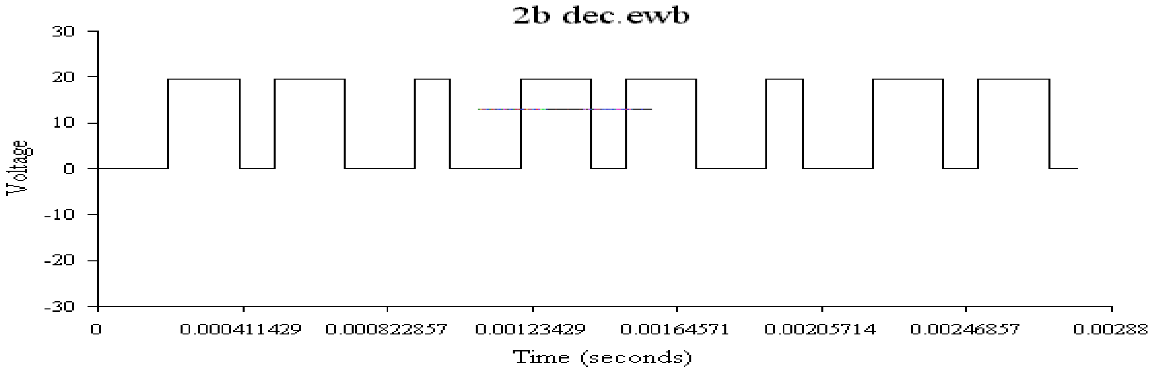


Figure4.17 comparator output at K3

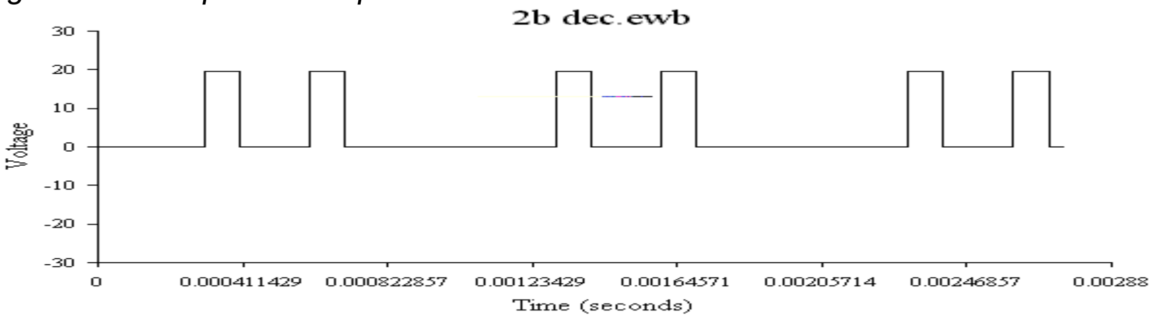


Figure 4.18 comparator output at K4

4.3.3 Logic code

Logical simplification of the levels is shown in the table below

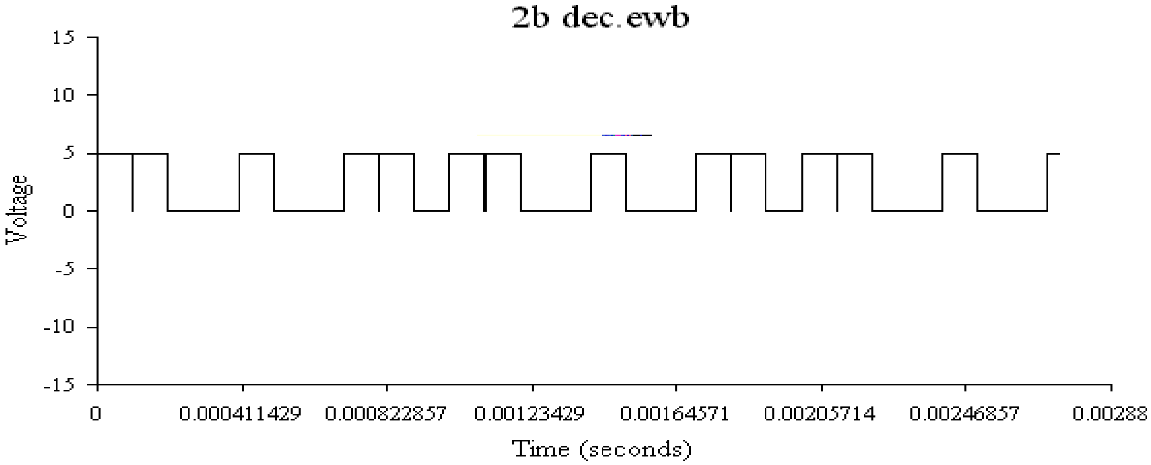
Inputs				**Outputs**	
K1	K2	K3	K4	A0	A1
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

TABLE 4.3 Logic code

The above truth table reduces to

A0 = K1 + K2

A1 = K1 + K3



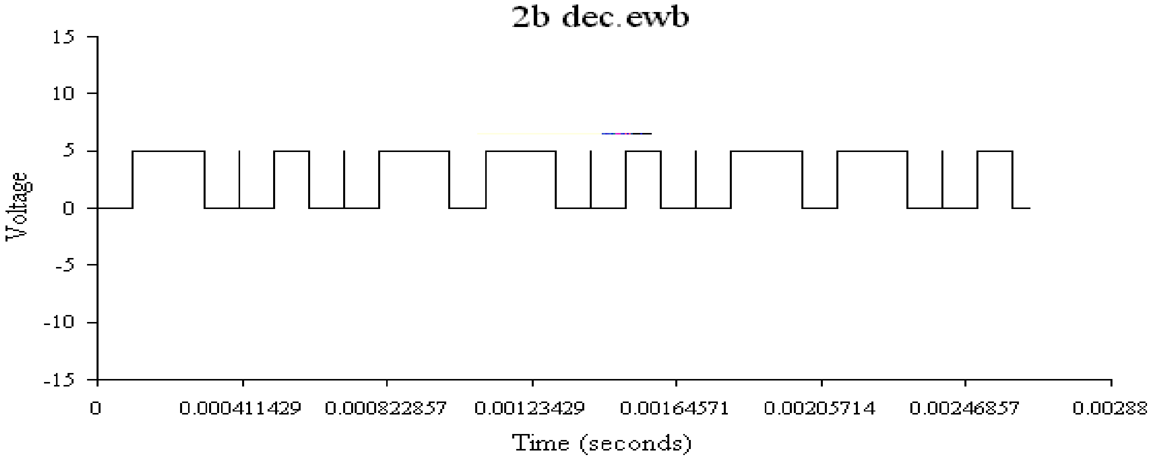


Figure 4.19 Output at A0

Figure 4.20 Output at A1

Whenever any of the input combination is given to this stage the output will be combination of two bits. Every input bit combination will have different two bit output combination.

4.3.4Parallel to serial converter

The aim of this particular stage is to convert the in coming 2 bit data into 1 bit serial data, which was originally sent from the transmitter end.

We carried out this design with the help of four NAND gates. The two input signal sources are the parallel data (A0 and A1) and clock signal (CLK) is used as control input. The clock signal controls which data source is to be selected

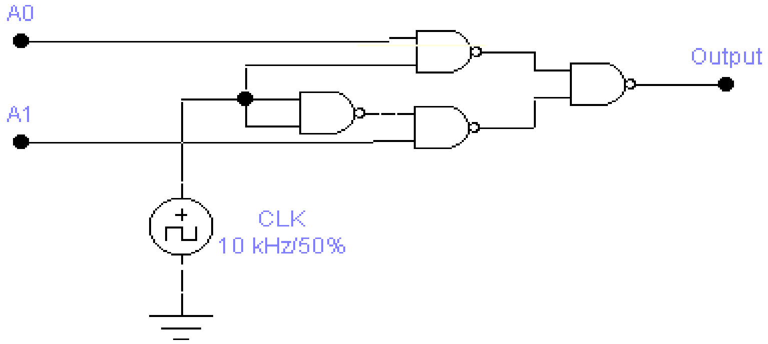


Figure 4.21 Nand gate parallel to serial converter

This circuit is logically equivalent for a simple single-pole double-throw (SPDT) switch.

4.3.5 Assembling the Sub Circuits

Implementing the additional circuitary with all sub circuits gives the complete decoder circuit as shown below in figure 4.22.

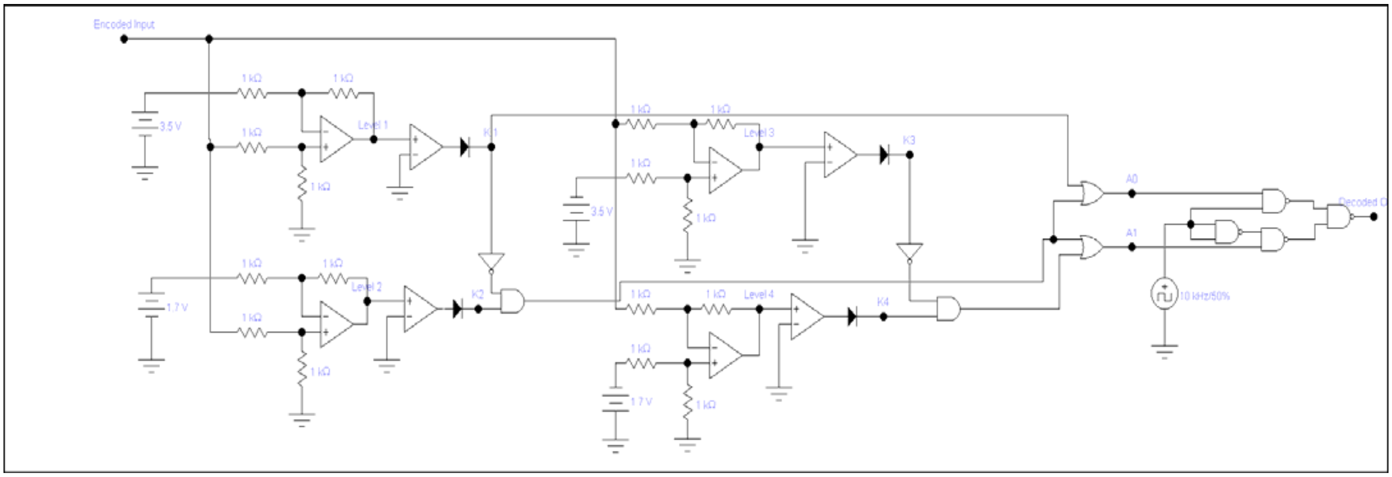


Figure 4.22 2B1Q Decoder Diagram

CHAPTER 5

AUXILIARY CIRCUITS

5.1 Transformerless Power Supply

As equipment attached to main usually consumes a lot of current, the circuits designed by us only require a few milliamps. Thus we require supply up to about 20ma at 5 volts.

The design uses capacitive reactance instead of resistance; and it doesn't generate very much heat. The circuit draws about 30ma AC.A fuse or a fusible resistor can be used to be on the safe side [12].

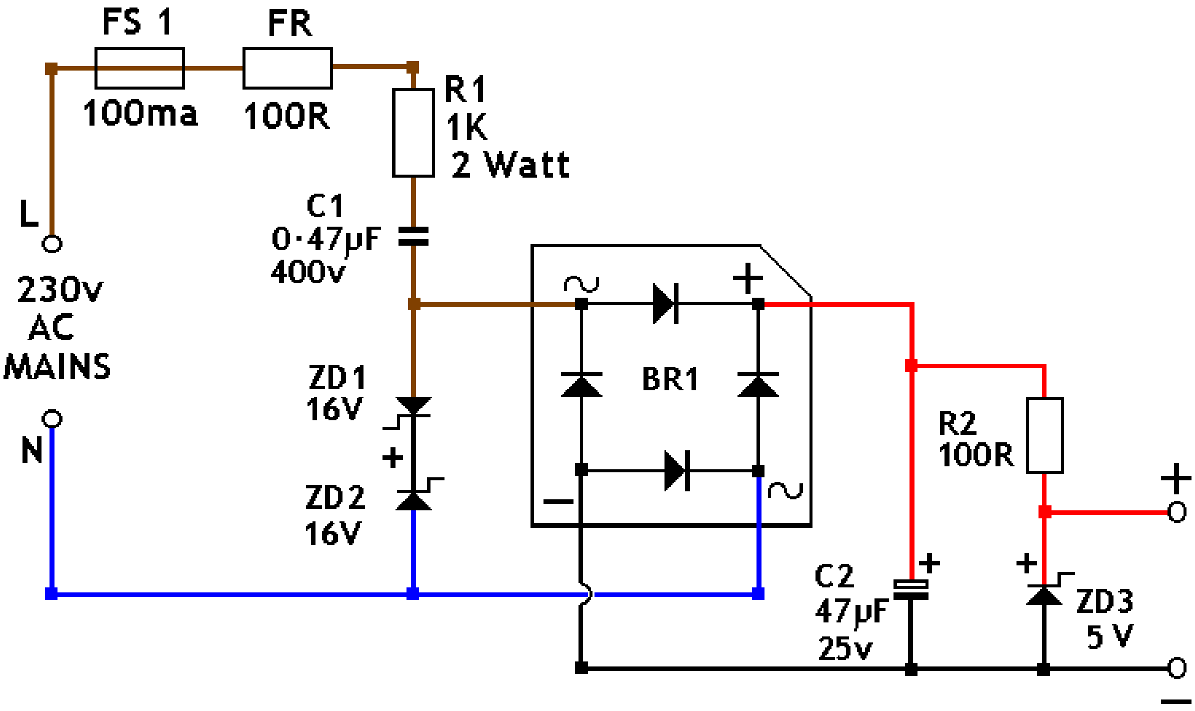


FIGURE 5.1 Transformerless Power Supply *Source: MIT*

In case of larger current requirements, a larger value capacitor can be used. ZD1 and ZD2 supply the low voltage ‘AC’. The bridge rectifier can be any of the small 'Round', 'In-line', or 'DIL' types; or four separate diodes could be used [12].

We can also replace R2 and ZD3 with a 78 Series regulator. For getting the desired reference voltage levels the value of the zener diodes are altered

The power supply cannot be modified to provide currents of anything up to 50 amps and above. The circuit is designed to provide a cheap compact power supply for Cmos logic circuits that require only a few milliamps.

For more than 20mA is required it is possible to increase C1 to 0.68uF or 1uF and thus obtain a current of up to about 40mA. But these suppresser capacitors are relatively big and more expensive than regular capacitors; and increasing the current means that higher wattage resistors and zener diodes are required [12].

5.2 Clock

Clock is generated using crystal for basic production of oscillations. Generator uses fundamental Not gate topology of circuit for the production of oscillations

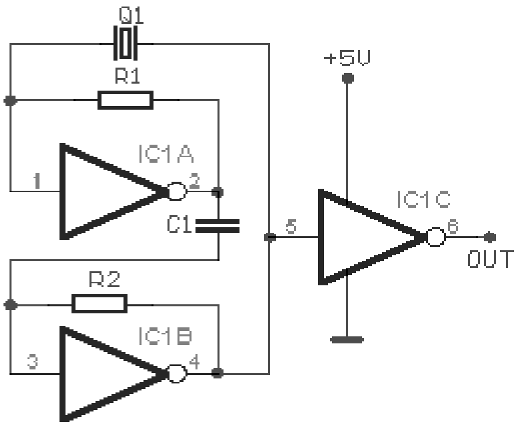


FIGURE 5.2 Logical Implementation
Source: SAM Electronic Circuits

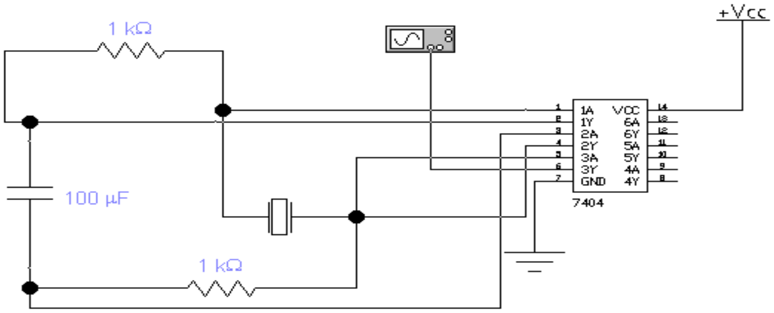


FIGURE 5.3 Clock Circuit Diagram

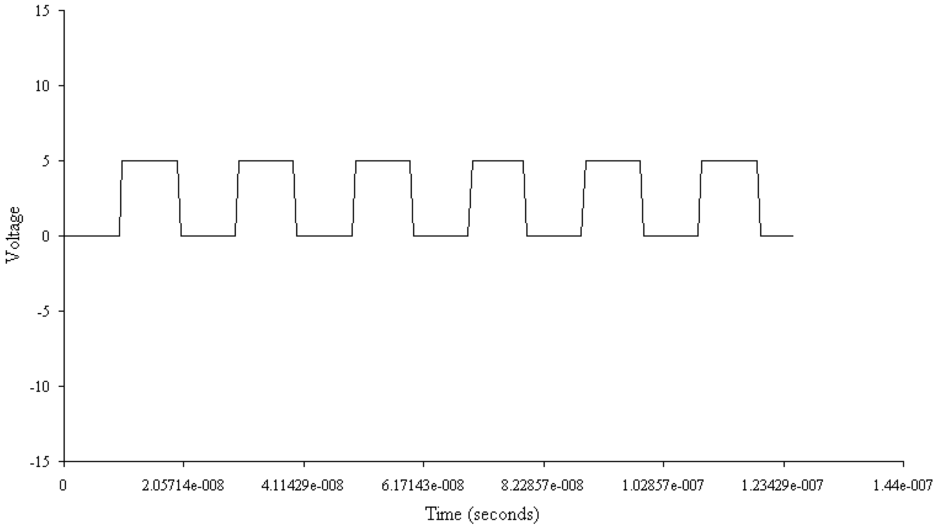


FIGURE 5.4 Clock Output

5.3 Extraction of Clock Signal

The clock recovery system is to recover the clock timing from the transmitted bipolar signal.

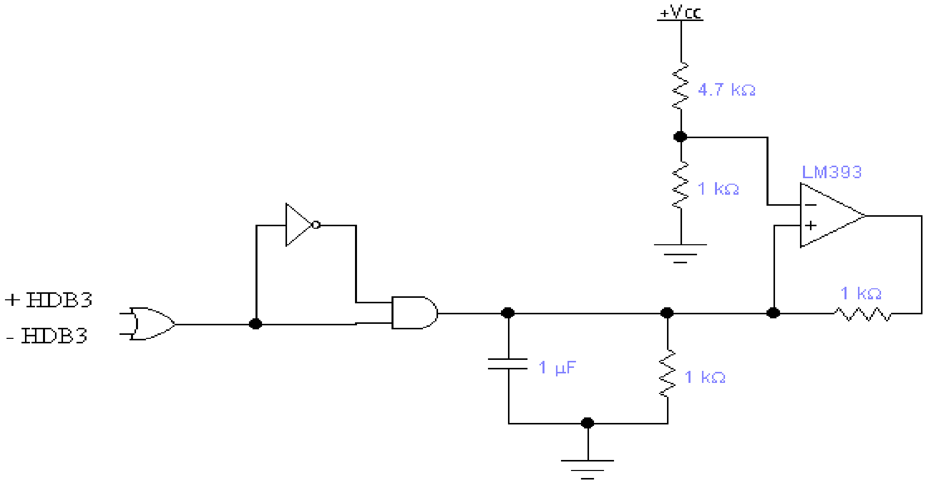
The received digital signal needs to be sampled at precise instants. This requires a clock signal at the receiver in synchronism with the clock signal at transmitter.

For extraction of clock signal from +HDB3 and -HDB3 pulses a logic circuit is used that creates a strobe for alignment of phase of clock with the incoming data and an astable multivibrator Both +HDB3 and -HDB-3 RZ signals are applied to an OR gate.

The output of OR gate is in then applied to an INVERTER and it also act as one of the input to an AND gate. The output of INVERTER is applied to second input of AND gate. INVERTER part from inverting the signal also causes a delay of 3nseconds. Therefore at the input of AND gate, there is an overlap of three nano seconds. During this period, both the inputs to AND gate are high thus the output of AND gate is also high.

FIGURE 5.5 Clock recovery circuit diagram

In this way a strobe signal having time interval of 3- 11 nano second is applied directly to the inverting terminal of the comparator LM393. A voltage divider network consisting of R1 and R2 applies constant 0.8947 volt to non-inverting terminal of the comparator LM393.Capacitor C charges through R4 and discharges through R3 and R4.In this case there is a delay of 23 nano seconds from input to output. During this time capacitor charges towards Vo.



When output goes low the capacitor starts discharging and it discharges to a value of 0.8947 volts. Then input to capacitor reverses and after 23- nano seconds delay, output goes high.

During this delay capacitor discharges more and when output of comparator changes to high, capacitor again starts charging. In this way we get 8.196 MHz clock signal as shown in the figure. The output of INVERTER is applied to the clock input of a JK negative edge triggered flip-flop.

The input is kept at 1 always. Therefore its output Q toggles on each falling edge of clock signal. The output of this flip-flop is 4.096 MHz. This is then applied to the clock input of second negative edge triggered JK flip flop.

The input is kept always 1 therefore its output also toggles on each falling edge of clock signal. It therefore further divides the clock signal by 2 and finally at its output we get 2.048 MHz clock signal.

This is then applied to the HDB3 decoder. 2b1q to HDB3 & AMI encoders also use this extracted clock.

CHAPTER 6

Printed Circuit Boards (PCB's)

6.1 Introduction

There are two generally main methods for the small-scale production of Printed Circuit Boards (PCB's) [13]. The first, preferred by most people, is by laying down special etch resistant transfers onto clean copper board and then etching the board in a bath of ferrous chloride solution. The second is to produce the artwork (foils) for the PCB layout using a PC software application, and then to transfer the track pattern to the copper board using a technique similar to developing and printing a photograph.

Both methods are quite straightforward, but the latter method, which is more expensive but quicker, produces better, results and allows more dense population of the PCB. The summary of the steps involved in making the PCB by second method is described in the following sections.

6.2 Method

There are six main steps for making a PCB [13], which are shown in the graphic below.

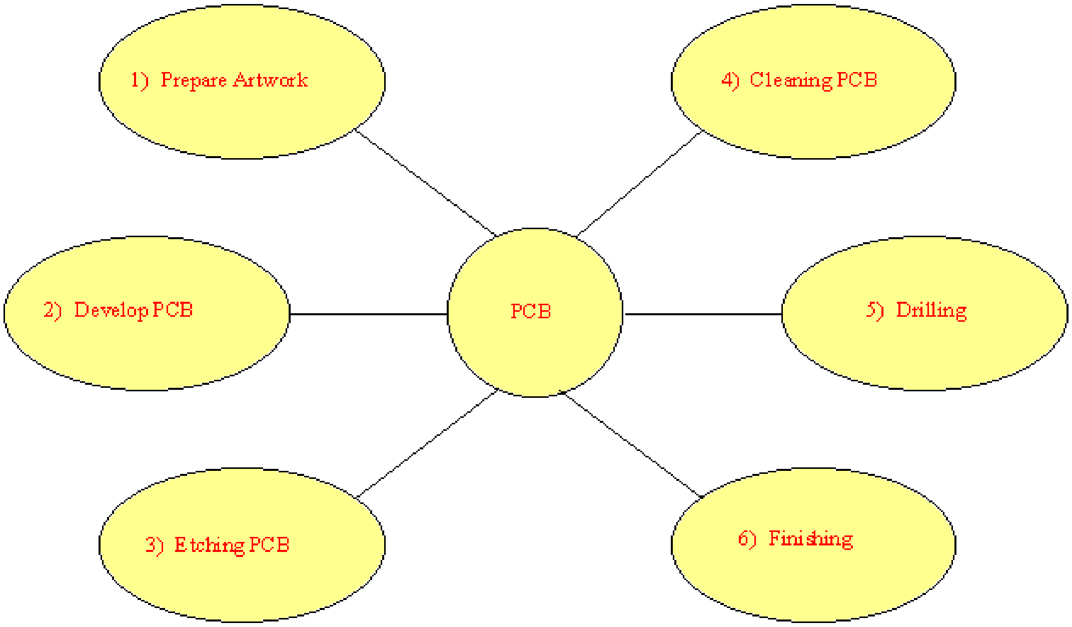


FIGURE 6.1 Steps for making a PCB [13]

6.2.1 Preparing the Artwork

6.2.1.1 Using PCB Layout Software

There are a large number of suppliers of PCB layout applications, which run on a PC like Protel, Tango, Eagle etc. Protel SE 99 has considerably a large library depending on the functions and complexity (i.e.: number of layers, pads and size of library), so it is the best available option for designing the lay out of the design.

To make an artwork open the layout application and using the library of packages provided, select all the component packages to be used in the layout (like 74LS08, TIP 31, resistors etc). These packages are then placed in their rough positions on the board area and their pins connected together as required. The screen shot below shows a Layout Software Tool in use.

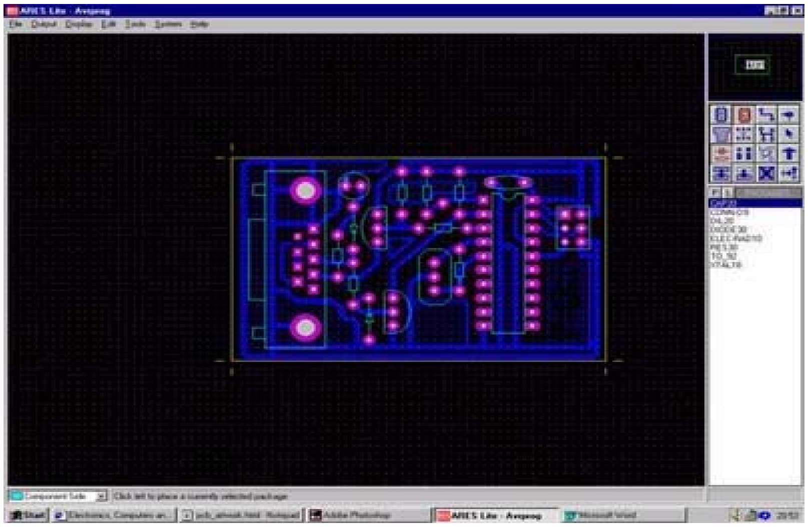


FIGURE 6.2 Layout software screen display

Making an artwork is a time consuming process, and one has to be very careful to connect the pins together correctly because there is no checking mechanism. Alternatively, the circuit can be entered in an accompanying schematic capture application and the PCB layout can be laid out automatically using the auto-router.

When the artwork is finished, the layers (usually top and bottom) are printed onto either acetate film or good quality tracing paper.

6.2.1.2 Using transfers

This is a slow method, which we adopted for our project design. We initially made the schematics in EWB 5.0a and the netlist files obtained were imported to layout software. Inorder to get required layouts the netlist files are loaded but we faced problems in respect of footnotes, so we switched over to first option.

6.2.2 Developing the Artwork

6.2.2.1 Pre-Sensitised boards

These are relatively expensive, but results can be excellent and quick. The boards are supplied with black plastic covering the surfaces to protect the Ultra-violet (UV) sensitive surfaces and this covering is removed immediately prior to using.

For the double-sided board, before removing the plastic, four pilot holes can be drilled to assist lining up the layers.

The bottom foil is not reversed when printing so that the printed sides of the artwork results close as possible to the copper surface. This will result in sharper and better resolution for thin tracks, because the UV light has less opportunity to 'spread' within the thickness of the plastic film or tracing paper used for the foil.

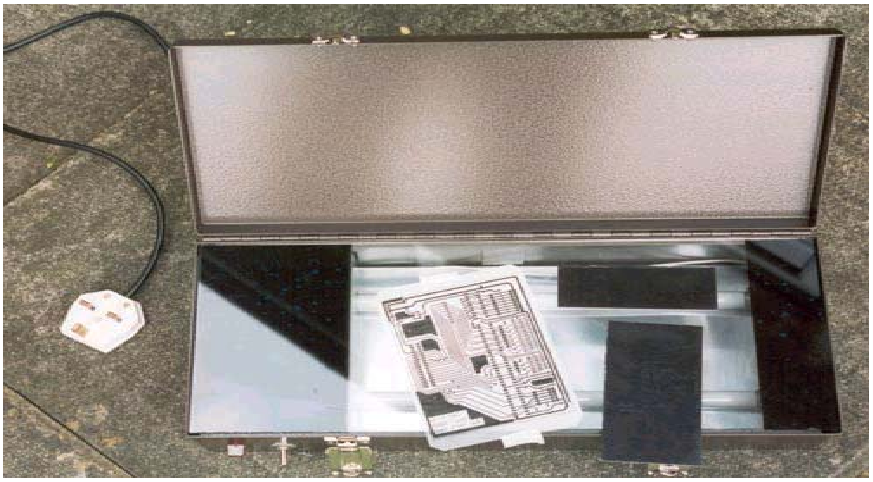


FIGURE 6.3 Light box being used for exposing [13]

The foils are affixed to the board with small pieces of adhesive tape. At this stage the artwork and PCB should be cut larger than the finished board. The board is then placed in the UV exposure box for an appropriate amount of time to allow the PCB pattern to be transferred to the board.

Each side of the board is usually exposed separately when using non-professional equipment. The photo' shows light box with the Development Board Foil ready to be used.

After exposure, the foils are carefully removed and the board is placed in a solution of developer for a couple of minutes. The tracks and pads will magically appear, similar to developing a photograph. Caustic Soda can be used with the pre-sensitized boards and this is available from most hardware stores for cleaning drains etc. It should be used in a well-ventilated area.

As soon as the developing is complete, the board is washed under cold running water with care taken to avoid damaging the etch resist on the board surfaces, which will be very soft at this stage. Etching should now be undertaken as soon as possible.

6.2.2.2 Coated Boards

A cheaper method is to use plain copper board and to apply an UV sensitive coating to it. A coating is applied from an aerosol spray under low light conditions.

This is a very hit and miss process, where good results are hard to obtain. If this method is used, it is important for the same manufacturer developer to be used throughout the process if the process is to work successfully.

6.2.3 Etching

This is the hazardous part and great care should be taken with the Ferric Chloride while preparing, using and disposing it off. This chemical (and to a lesser extent) the caustic soda developer solution, should be used in a well-ventilated area.

Before etching begins, the artwork on the PCB should be inspected for damaged tracks and hairline cracks, which should be corrected using an etch resist pen or similar. If this is necessary, the board should first be dried off, as soon as possible after developing.

FIGURE 6.4 Artwork dipped in Etching solution [13]

Practically it is found that etching is best completed with the chemical heated to a little above room temperature, using a hot water-bath. Etching should then take little more than 15 to 20 minutes with constant agitation of the board. Leaving the etching bath floating in the hot water-bath makes agitation easy but care is taken to avoid spilling of the chemical.

When the PCB looks ready, it should be carefully removed from the chemical, using plastic gloves and thoroughly rinsed in a cold water bath.

After inspection, if it is finished then it should be returned to the caustic soda solution, to soften the resist, which can then be removed with a soft abrasive (e.g. fine wet and dry paper). However, it is preferred to remove the resist at the end, after all other stages have been completed.

6.2.4 Cleaning the PCB

Cleaning the PCB, is perhaps easiest to do at this stage, as the etch resist is soft, but it is preferred to complete the drilling and cutting of the board to size, first. Otherwise, a further, final session of cleaning will be needed later.

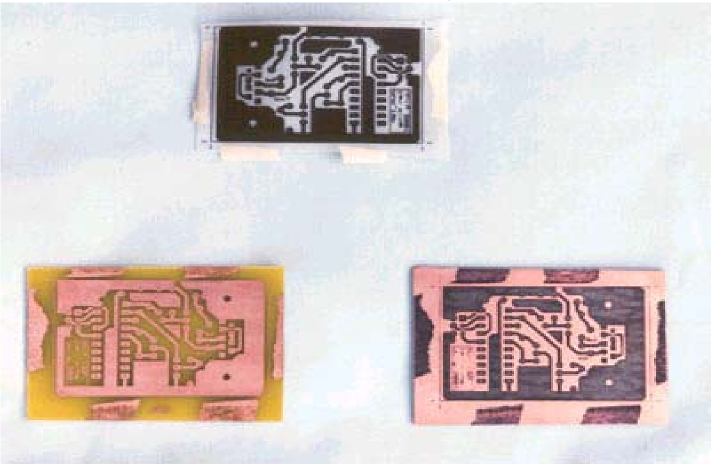


FIGURE 6.5 Etched PCB layout [13]

Transfers and etch resist is fairly easily removed with a medium density, waterproof, abrasive paper, that can be used under running water. Only light pressure is needed, to avoid damaging the thinner copper tracks. This can be followed by use of a very fine paper to give a better finish.

The picture, above, shows from top clockwise, the some artworks. Finally, the etched layout is ready for drilling and finishing.

6.2.5 Drilling the PCB

Most PCBs these days contain a few IC's as a minimum, and this can quickly multiply the number of holes that need to be drilled. It is important, especially with dual sided boards, that the holes are drilled with the drill 'upright' so that the holes are lined up in the middle of the pads on both sides. This is easy if you have a small bench drill, which will fit into a pillar stand.

A soft material should be placed under the board for the drill to pass into, such as a spare piece of cork or an old 'jiffy' bag! Whatever method is used, it is important not to allow any sideways movement of the PCB (or the drill) to avoid breakage. The drills used should be having a larger shank, as these will not blunt quickly [13].

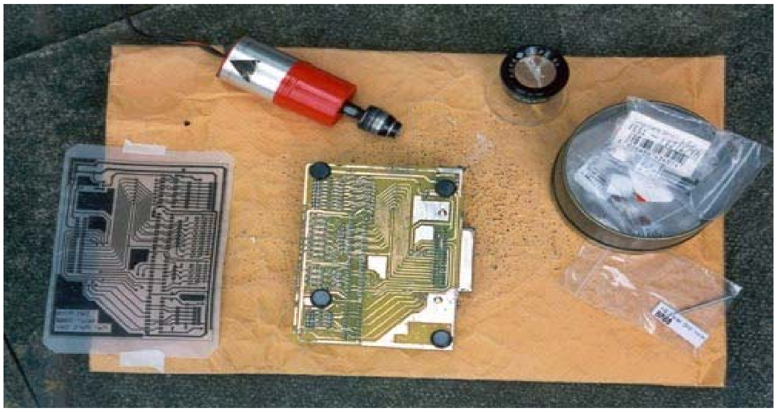


FIGURE 6.6 Drill process [13]

It is best to use a range of drill sizes - 0.8mm for IC pads and most other components, 1.0mm for thicker component leads (diodes and regulators) and 1.2mm for some larger components [13]. The normal practice of drilling a pilot hole and then the final size later. Therefore, drill each hole only once, with the correct sized drill.

6.2.6 Finishing the PCB

After the etching has been done, the holes have been drilled and the last task before soldering the components, is to finish the PCB so that it looks as professional as possible.

First, the oversize board can be cut to size, using a hacksaw, to allow for filing/smoothing of the rough-cut PCB edge. Take care not to rub fingers and hands against the rough PCB edges, as the glass fibres are so fine, they can enter the body. Similarly, do not breathe in dust generated when drilling, cutting or filing the board.

The board should now be cleaned as described in the earlier page, but if this has already been completed, then a light rub over with a fine, waterproof, abrasive paper should be carried out.

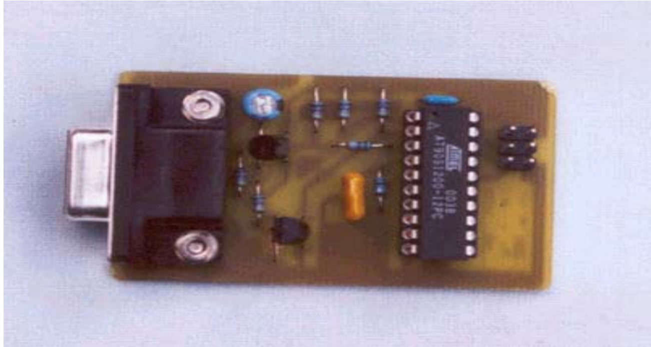


FIGURE 6.7 PCB in Final Shape [13]

The board, with shiny copper tracks, is now ready for assembly and soldering. After this has been completed, and basic functional testing carried out (to spot the mistakes), the bottom surface should be coated with a protective lacquer, to prevent oxidization of the tracks, over time. This should be done as soon as possible after component assembly.

A better approach is to 'tin' the copper tracks before component assembly. This takes some practice, if a messy result is to be avoided, but the key to success is heat and flux.

Smear a THIN layer of plumbers flux across the surface to be tinned, then using the soldering iron and the minimum possible solder, work the solder across the pads and along tracks as quickly as possible. Avoid using too much heat on thinner tracks to avoid damaging them. Finally, inspect the board for solder bridges between tracks and pads.

The flux is messy and this is best removed using cellulose thinners, in a well-ventilated area, followed by a wash with soapy water. A protective lacquer is not needed with tinned boards, but will enhance appearance, if applied to the finished board after components have been assembled and soldered.

The picture here shows a tinned PCB ready for cutting and finishing.

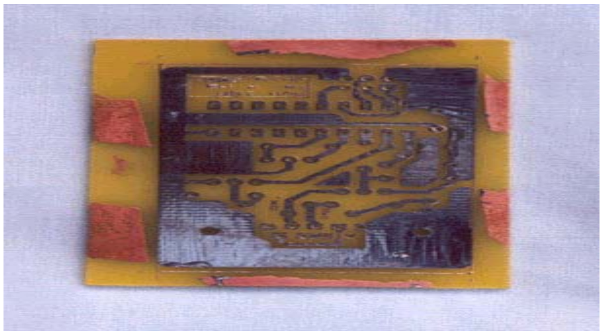


FIGURE 6.8 A Tinned PCB Layou [13]t

6.3 Potential Health Hazards

1. Inhalation of concentrated vapors or mist may cause irritation of the respiratory tract.

2. Ingestion may cause severe liver and/or kidney damage, and may be fatal.

3. Contact with liquid, mist, or vapor can cause immediate irritation or corrosive burns to all human tissue. Severity is generally determined by the concentration of the solution and duration of exposure.

4. Contact with eyes may cause irritation, tearing and eye tissue discoloration, and may result in permanent visual loss unless removed quickly by thorough irrigation with water

6.4 Safety Precautions

1. Handle under a fume hood whenever possible.

2. Work with the fume hood splash guard as low as possible. This improves the effectiveness of the fume hood ventilation as well as providing protection from splashed solution.

3. Wear rubber gloves and eye protection whenever handling solution.

4. Ferric Chloride will stain anything that it contacts. You may want to wear an apron or lab coat to protect clothing.

6.5 Project Modules PCB Designs

All the PCB layouts were prepared in Protel environment using the appropriate footprints positioning on the required board area.

6.5.1 Encoder Modules

6.5.1.1 HDB3 Encoder

a) Circuit Diagram

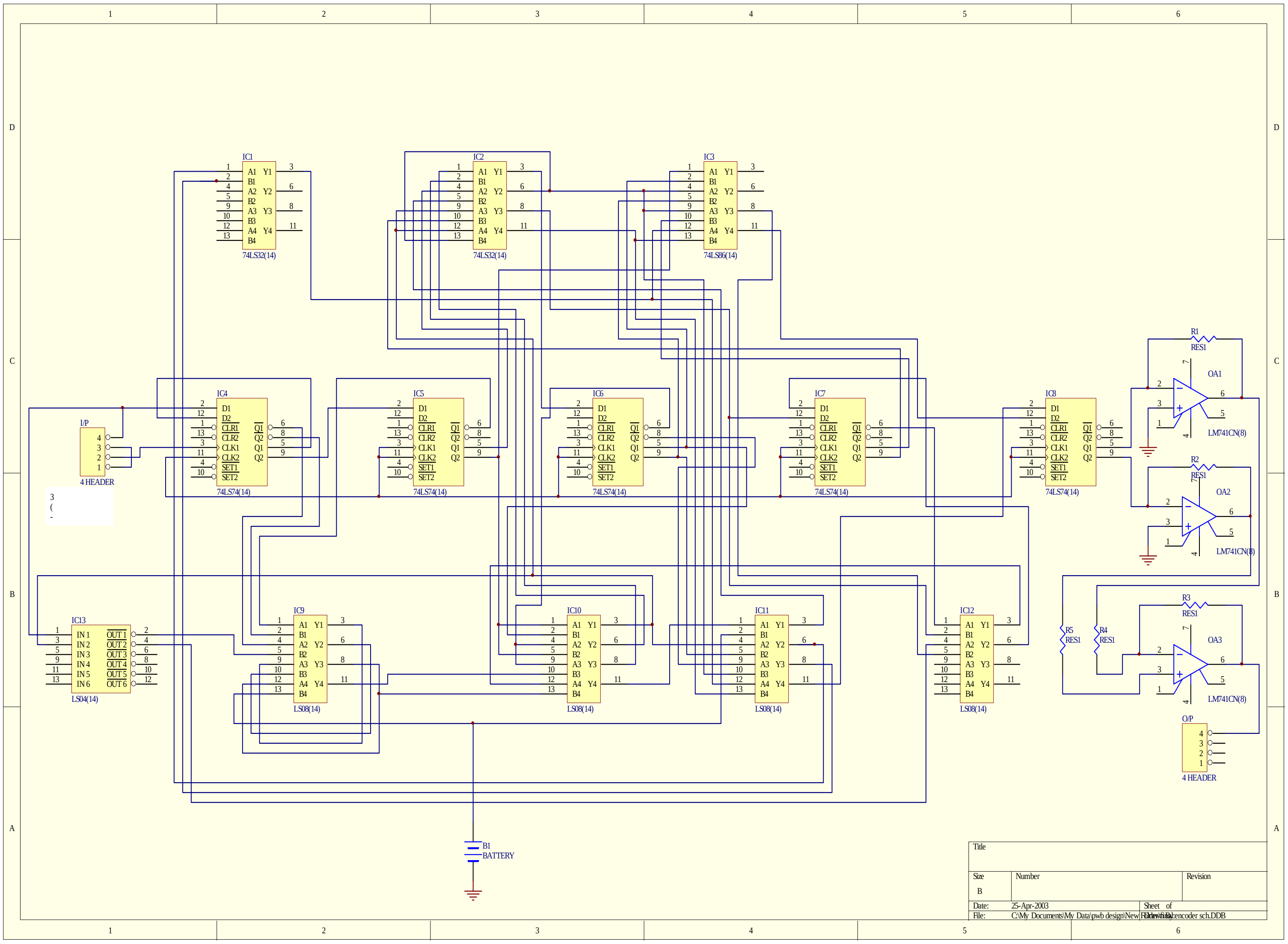


FIGURE 6.9 HDB3 encoder circuit diagram

b) PCB Layout

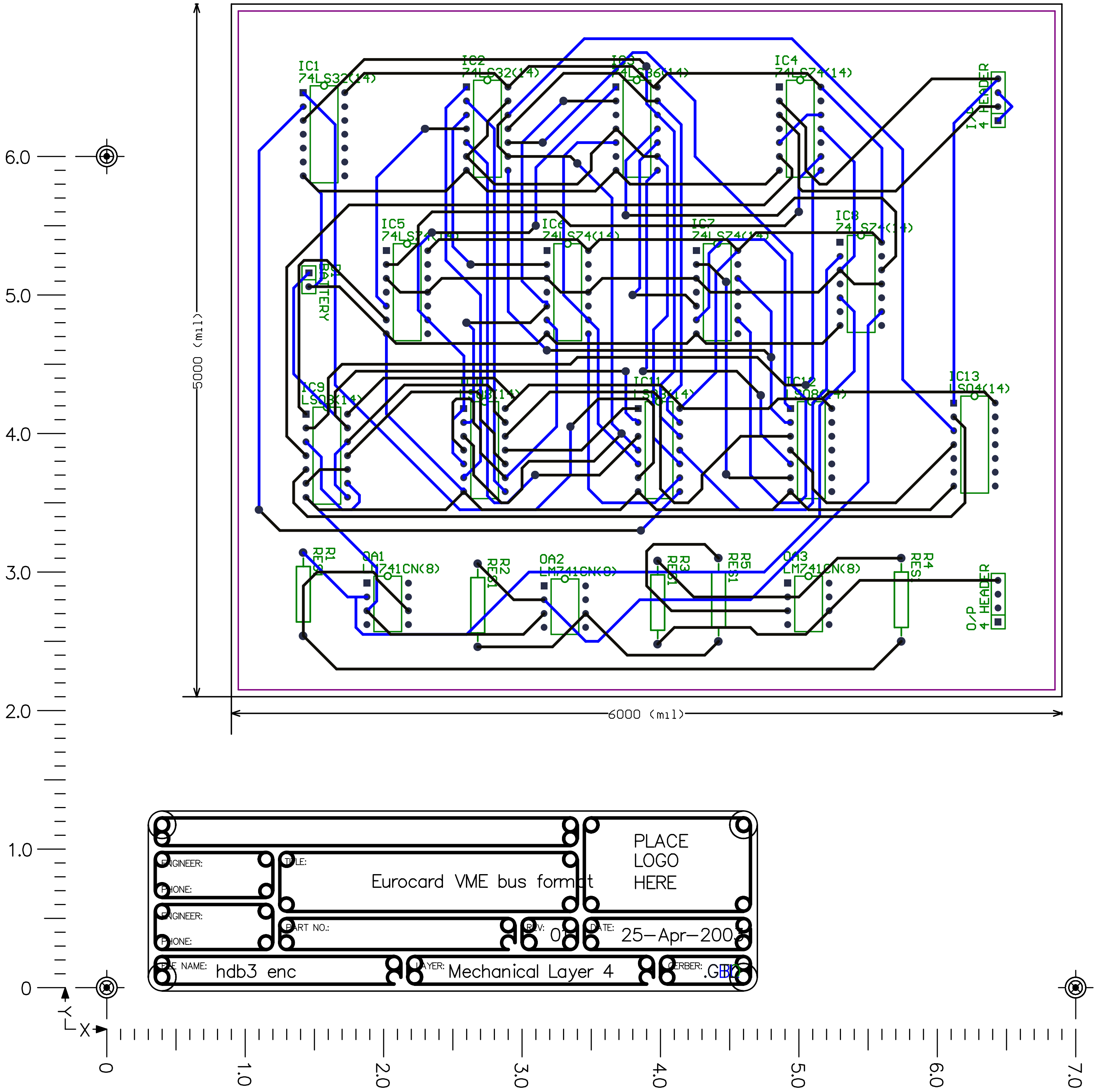

IC1
74LS32(14)
IC2
74LS32(14)
IC3
74LS86(14)
IC4
74LS74(14)
I/P
4 HEADER
IC5
74LS74(14)
IC6
74LS74(14)
IC7
74LS74(14)
IC8
74LS74(14)
BATTERY
IC9
LS08(14)
IC11
IC12
LS08(14)
IC13
LS04(14)
R1
RES1
OA1
LM741CN(8)
R2
RES1
OA2
LM741CN(8)
R3
RES1
R5
RES1
OA3
LM741CN(8)
R4
RES1
O/P
4 HEADER
5000 (mil)
6000 (mil)
6.0
5.0
4.0
3.0
2.0
1.0
0
Y
X
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
ENGINEER:
PHONE:
ENGINEER:
PHONE:
TITLE:
Eurocard VME bus format
PLACE
LOGO
HERE
PART NO.:
REV:
01
DATE:
25-Apr-2003
FILE NAME:
hdb3 enc
LAYER:
Mechanical Layer 4
GERBER:
.GBL

FIGURE 6.10 HDB3 encoder PCB layout

6.5.1.2 HDB3 Decoder

a) Circuit Diagram

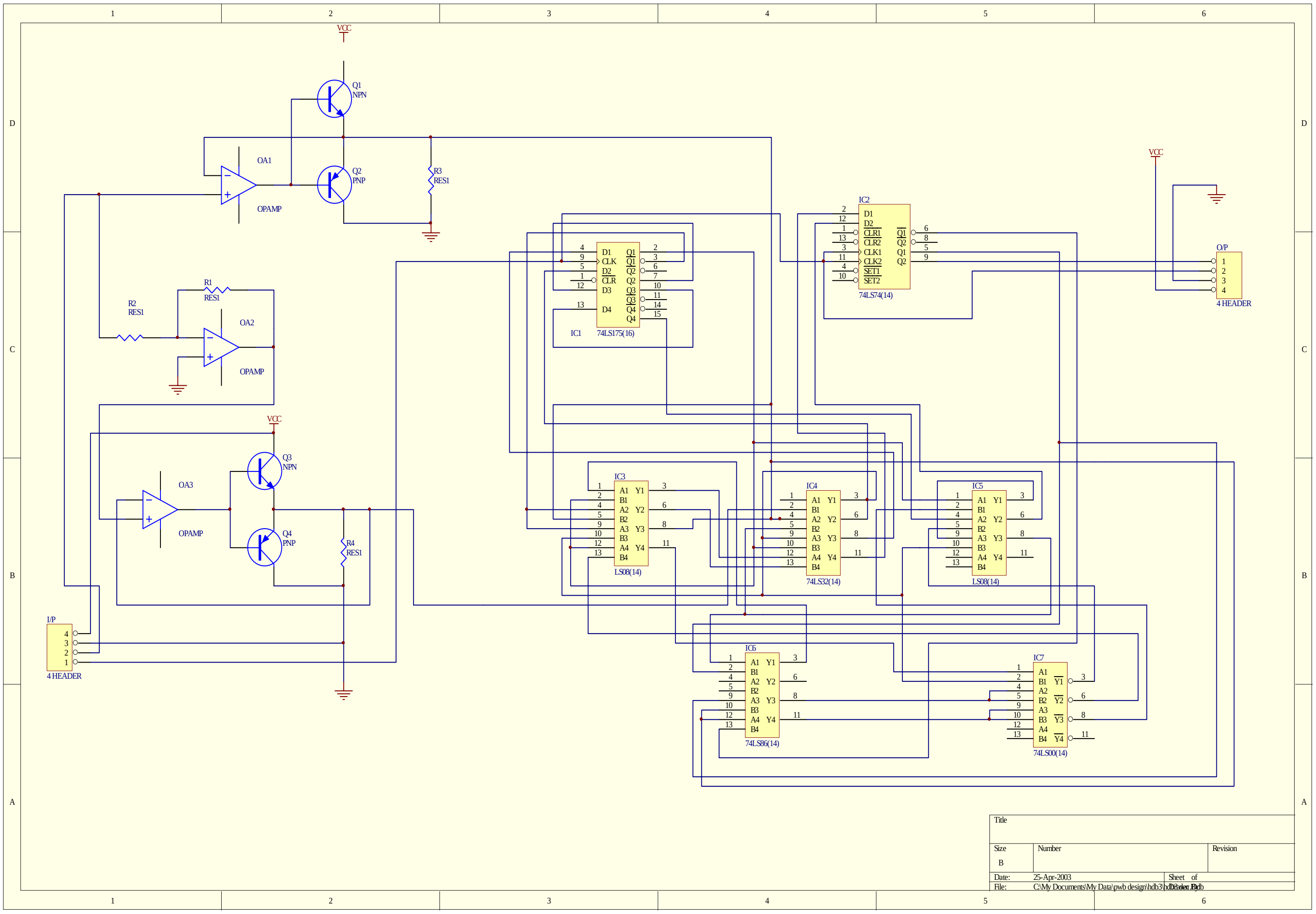


FIGURE 6.11 HDB3 decoder circuit diagram

b) PCB Layout

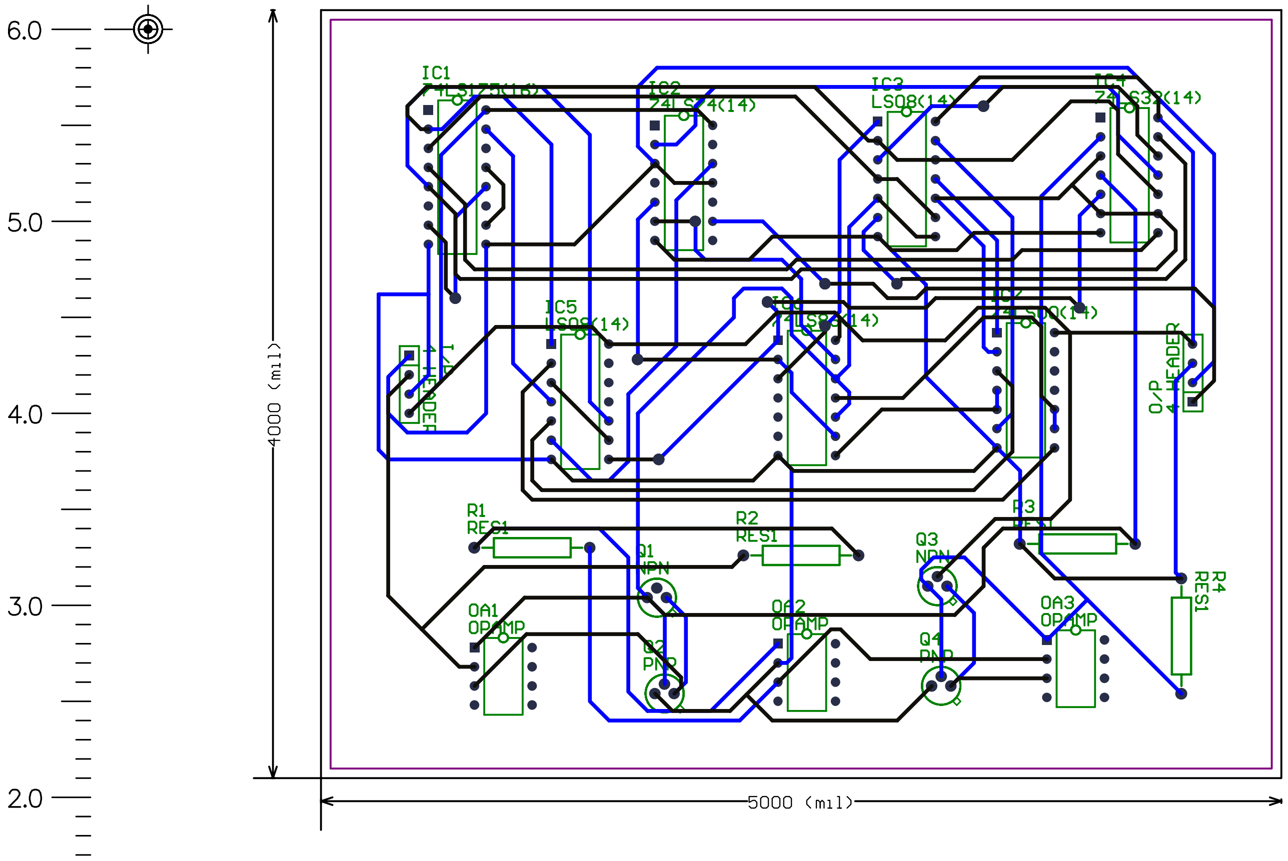


FIGURE 6.12 HDB3 decoder PCB Layout

6.5.1.3 2B1Q Encoder

a) Circuit Diagram

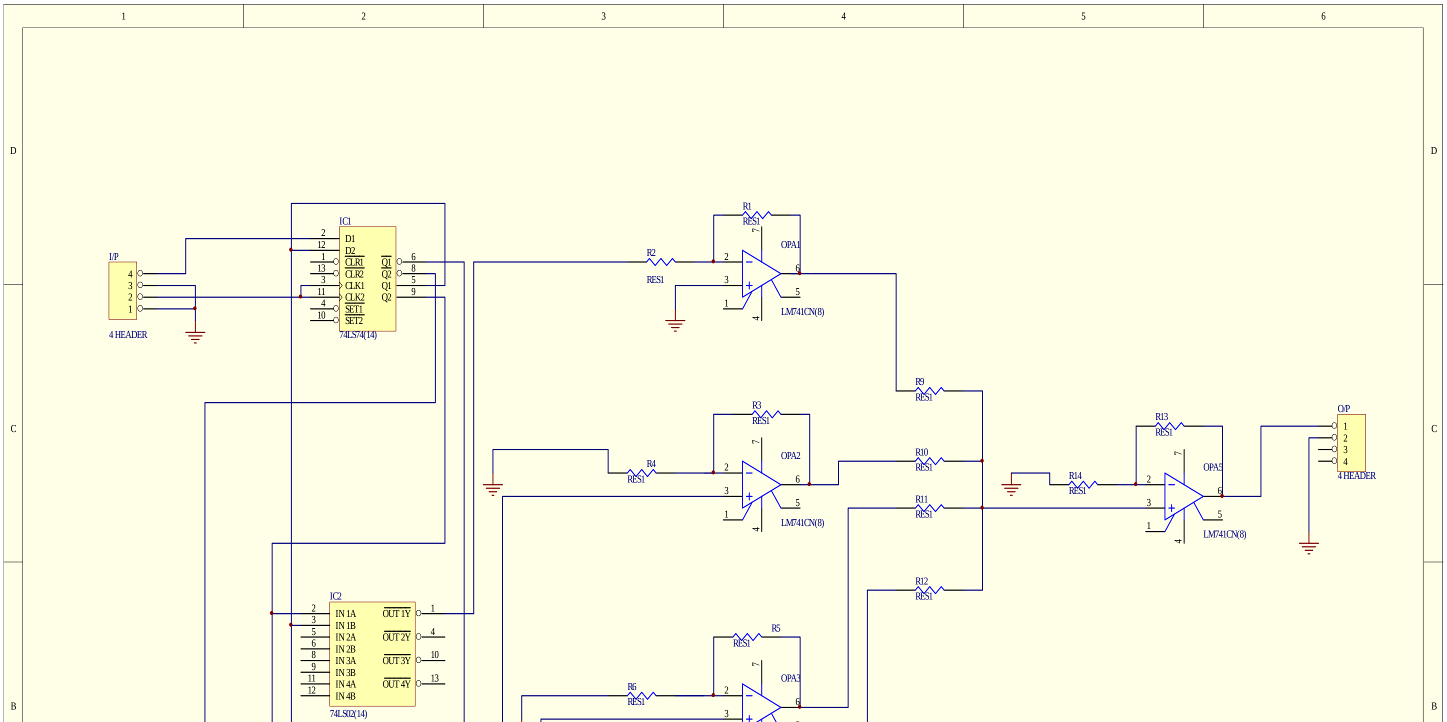


FIGURE 6.13 2B1Q encoder circuit diagram

b) PCB Layout

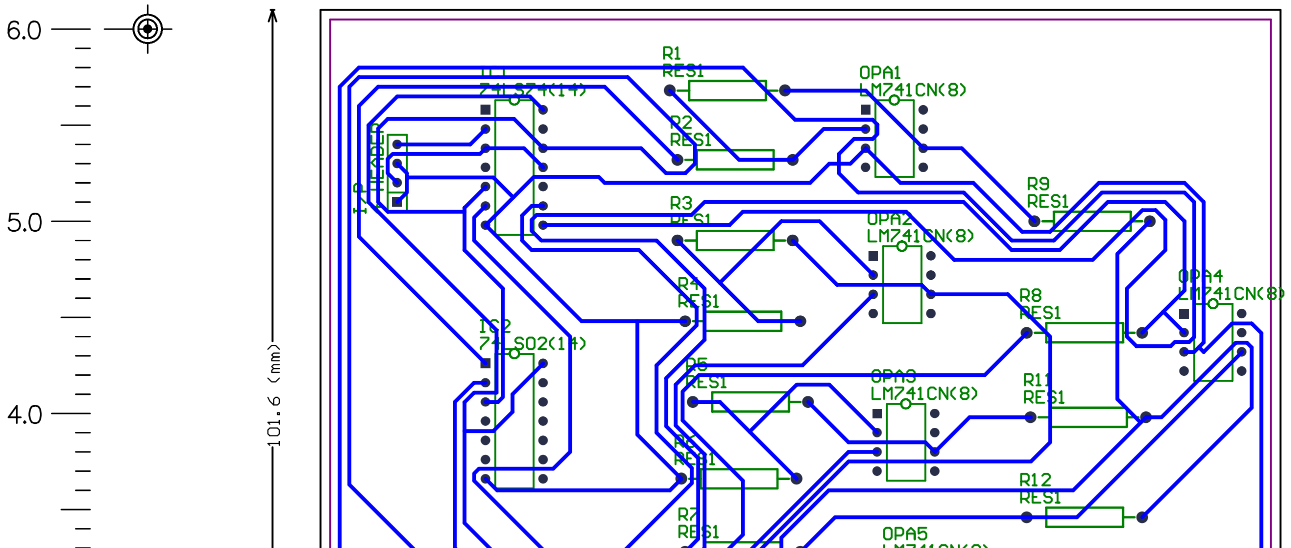


FIGURE 6.14 2B1Q encoder PCB Layout

6.5.1.4 2B1Q Decoder

a) Circuit Diagram

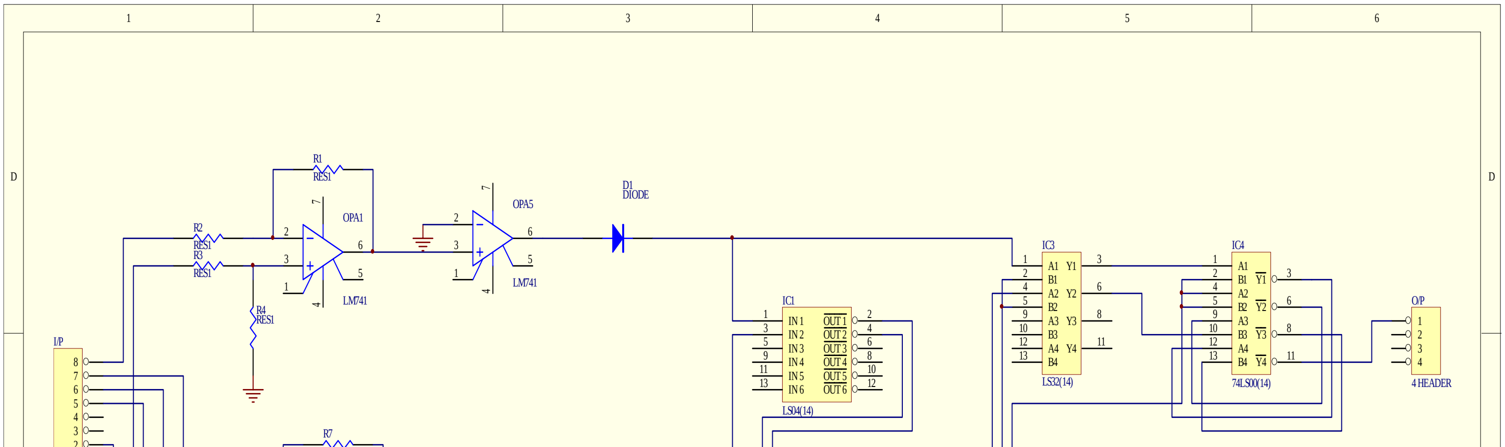


FIGURE 6.15 2B1Q Decoder circuit diagram

b) PCB Layout

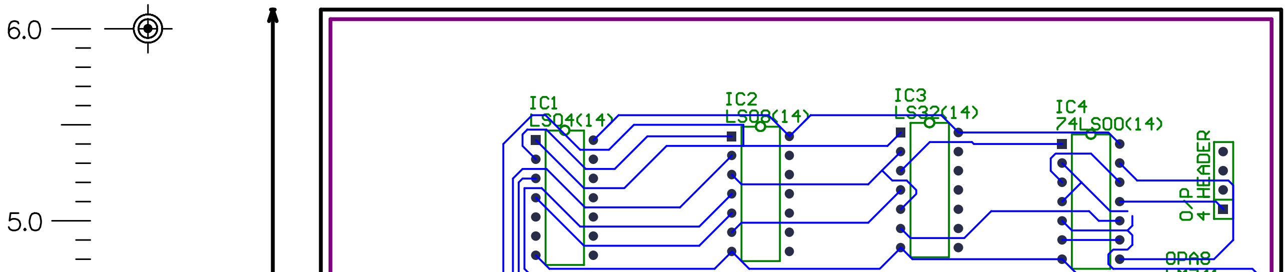


FIGURE 6.16 2B1Q decoder PCB layout

CHAPTER 6

Printed Circuit Boards (PCB's)

6.1 Introduction

There are two generally main methods for the small-scale production of Printed Circuit Boards (PCB's) [13]. The first, preferred by most people, is by laying down special etch resistant transfers onto clean copper board and then etching the board in a bath of ferrous chloride solution. The second is to produce the artwork (foils) for the PCB layout using a PC software application, and then to transfer the track pattern to the copper board using a technique similar to developing and printing a photograph.

Both methods are quite straightforward, but the latter method, which is more expensive but quicker, produces better, results and allows more dense population of the PCB. The summary of the steps involved in making the PCB by second method is described in the following sections.

6.2 Method

There are six main steps for making a PCB [13], which are shown in the graphic below.

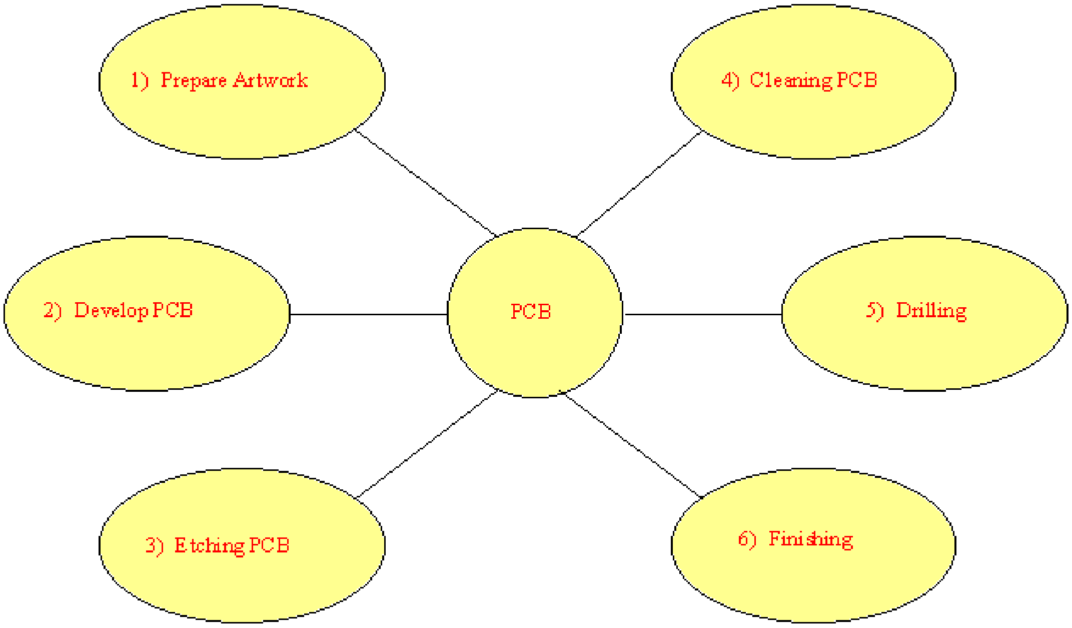


FIGURE 6.1 Steps for making a PCB [13]

6.2.1 Preparing the Artwork

6.2.1.1 Using PCB Layout Software

There are a large number of suppliers of PCB layout applications, which run on a PC like Protel, Tango, Eagle etc. Protel SE 99 has considerably a large library depending on the functions and complexity (i.e.: number of layers, pads and size of library), so it is the best available option for designing the lay out of the design.

To make an artwork open the layout application and using the library of packages provided, select all the component packages to be used in the layout (like 74LS08, TIP 31, resistors etc). These packages are then placed in their rough positions on the board area and their pins connected together as required. The screen shot below shows a Layout Software Tool in use.

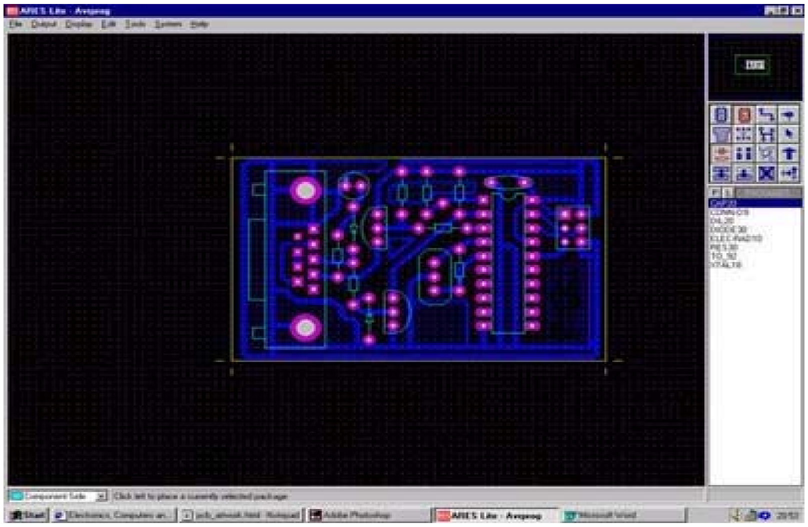


FIGURE 6.2 Layout software screen display

Making an artwork is a time consuming process, and one has to be very careful to connect the pins together correctly because there is no checking mechanism. Alternatively, the circuit can be entered in an accompanying schematic capture application and the PCB layout can be laid out automatically using the auto-router.

When the artwork is finished, the layers (usually top and bottom) are printed onto either acetate film or good quality tracing paper.

6.2.1.2 Using transfers

This is a slow method, which we adopted for our project design. We initially made the schematics in EWB 5.0a and the netlist files obtained were imported to layout software. Inorder to get required layouts the netlist files are loaded but we faced problems in respect of footnotes, so we switched over to first option.

6.2.2 Developing the Artwork

6.2.2.1 Pre-Sensitised boards

These are relatively expensive, but results can be excellent and quick. The boards are supplied with black plastic covering the surfaces to protect the Ultra-violet (UV) sensitive surfaces and this covering is removed immediately prior to using.

For the double-sided board, before removing the plastic, four pilot holes can be drilled to assist lining up the layers.

The bottom foil is not reversed when printing so that the printed sides of the artwork results close as possible to the copper surface. This will result in sharper and better resolution for thin tracks, because the UV light has less opportunity to 'spread' within the thickness of the plastic film or tracing paper used for the foil.

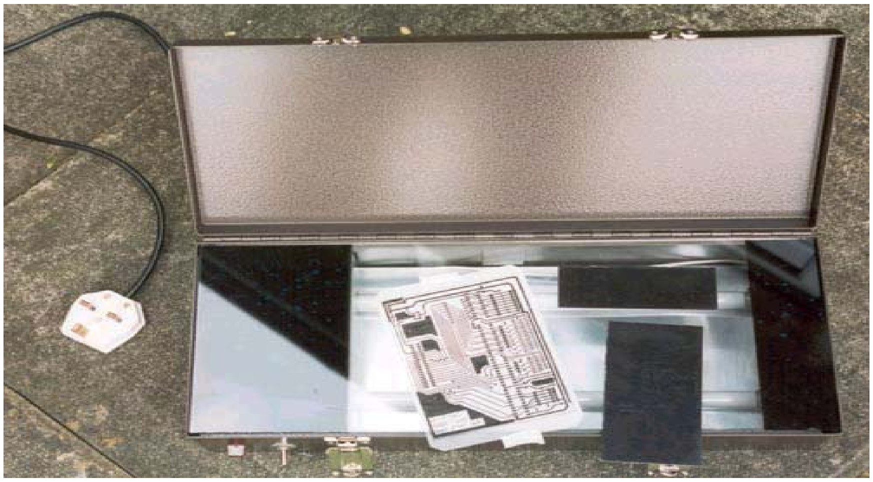


FIGURE 6.3 Light box being used for exposing [13]

The foils are affixed to the board with small pieces of adhesive tape. At this stage the artwork and PCB should be cut larger than the finished board. The board is then placed in the UV exposure box for an appropriate amount of time to allow the PCB pattern to be transferred to the board.

Each side of the board is usually exposed separately when using non-professional equipment. The photo' shows light box with the Development Board Foil ready to be used.

After exposure, the foils are carefully removed and the board is placed in a solution of developer for a couple of minutes. The tracks and pads will magically appear, similar to developing a photograph. Caustic Soda can be used with the pre-sensitized boards and this is available from most hardware stores for cleaning drains etc. It should be used in a well-ventilated area.

As soon as the developing is complete, the board is washed under cold running water with care taken to avoid damaging the etch resist on the board surfaces, which will be very soft at this stage. Etching should now be undertaken as soon as possible.

6.2.2.2 Coated Boards

A cheaper method is to use plain copper board and to apply an UV sensitive coating to it. A coating is applied from an aerosol spray under low light conditions.

This is a very hit and miss process, where good results are hard to obtain. If this method is used, it is important for the same manufacturer developer to be used throughout the process if the process is to work successfully.

6.2.3 Etching

This is the hazardous part and great care should be taken with the Ferric Chloride while preparing, using and disposing it off. This chemical (and to a lesser extent) the caustic soda developer solution, should be used in a well-ventilated area.

Before etching begins, the artwork on the PCB should be inspected for damaged tracks and hairline cracks, which should be corrected using an etch resist pen or similar. If this is necessary, the board should first be dried off, as soon as possible after developing.

FIGURE 6.4 Artwork dipped in Etching solution [13]

Practically it is found that etching is best completed with the chemical heated to a little above room temperature, using a hot water-bath. Etching should then take little more than 15 to 20 minutes with constant agitation of the board. Leaving the etching bath floating in the hot water-bath makes agitation easy but care is taken to avoid spilling of the chemical.

When the PCB looks ready, it should be carefully removed from the chemical, using plastic gloves and thoroughly rinsed in a cold water bath.

After inspection, if it is finished then it should be returned to the caustic soda solution, to soften the resist, which can then be removed with a soft abrasive (e.g. fine wet and dry paper). However, it is preferred to remove the resist at the end, after all other stages have been completed.

6.2.4 Cleaning the PCB

Cleaning the PCB, is perhaps easiest to do at this stage, as the etch resist is soft, but it is preferred to complete the drilling and cutting of the board to size, first. Otherwise, a further, final session of cleaning will be needed later.

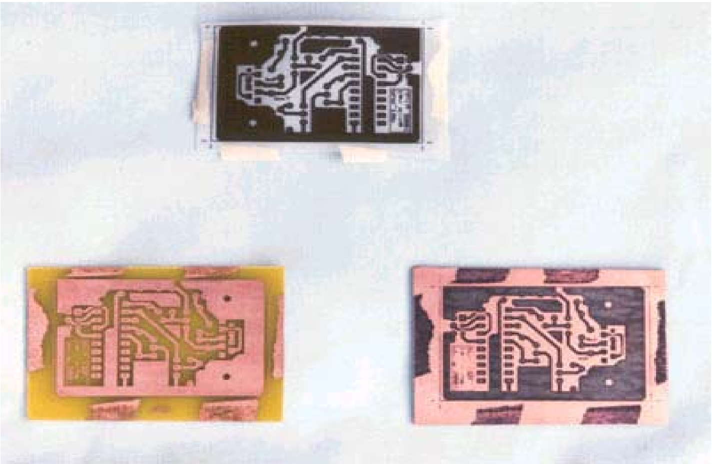


FIGURE 6.5 Etched PCB layout [13]

Transfers and etch resist is fairly easily removed with a medium density, waterproof, abrasive paper, that can be used under running water. Only light pressure is needed, to avoid damaging the thinner copper tracks. This can be followed by use of a very fine paper to give a better finish. The picture, above, shows from top clockwise, the some artworks. Finally, the etched layout is ready for drilling and finishing.

6.2.5 Drilling the PCB

Most PCBs these days contain a few IC's as a minimum, and this can quickly multiply the number of holes that need to be drilled. It is important, especially with dual sided boards, that the holes are drilled with the drill 'upright' so that the holes are lined up in the middle of the pads on both sides. This is easy if you have a small bench drill, which will fit into a pillar stand.

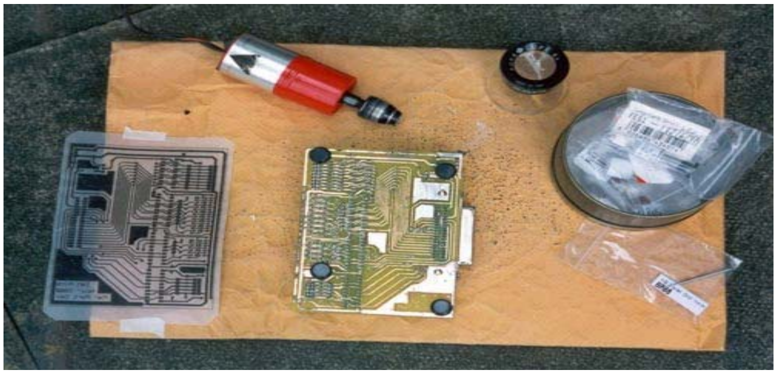


FIGURE 6.6 Drill process [13]

A soft material should be placed under the board for the drill to pass into, such as a spare piece of cork or an old 'jiffy' bag! Whatever method is used, it is important not to allow any sideways movement of the PCB (or the drill) to avoid breakage. The drills used should be having a larger shank, as these will not blunt quickly [13].

It is best to use a range of drill sizes - 0.8mm for IC pads and most other components, 1.0mm for thicker component leads (diodes and regulators) and 1.2mm for some larger components [13]. The normal practice of drilling a pilot hole and then the final size later. Therefore, drill each hole only once, with the correct sized drill.

6.2.6 Finishing the PCB

After the etching has been done, the holes have been drilled and the last task before soldering the components, is to finish the PCB so that it looks as professional as possible.

First, the oversize board can be cut to size, using a hacksaw, to allow for filing/smoothing of the rough-cut PCB edge. Take care not to rub fingers and hands against the rough PCB edges, as the glass fibres are so fine, they can enter the body. Similarly, do not breathe in dust generated when drilling, cutting or filing the board.

The board should now be cleaned as described in the earlier page, but if this has already been completed, then a light rub over with a fine, waterproof, abrasive paper should be carried out.

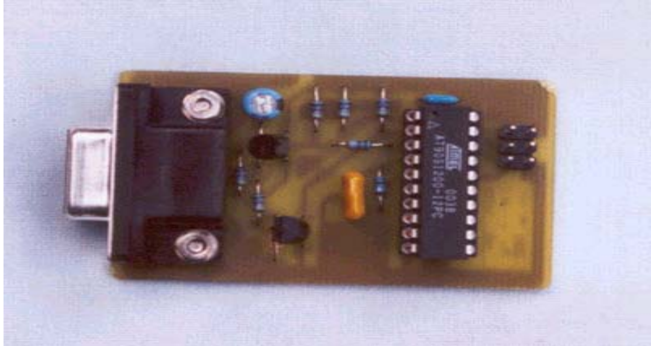


FIGURE 6.7 PCB in Final Shape [13]

The board, with shiny copper tracks, is now ready for assembly and soldering. After this has been completed, and basic functional testing carried out (to spot the mistakes), the bottom surface should be coated with a protective lacquer, to prevent oxidization of the tracks, over time. This should be done as soon as possible after component assembly.

In order to avoid damaging the tracks avoid using too much heat on thinner tracks. Finally, inspect the board for solder bridges between tracks and pads.

The flux is messy and this is best removed using cellulose thinners, in a well-ventilated area, followed by a wash with soapy water. A protective lacquer is not needed with tinned boards, but will enhance appearance, if applied to the finished board after components have been assembled and soldered.

The picture here shows a tinned PCB ready for cutting and finishing.

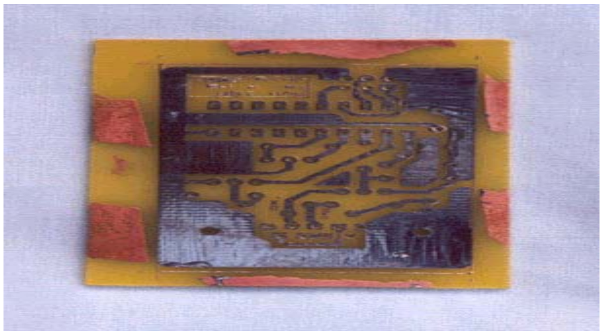


FIGURE 6.8 A Tinned PCB Layou [13]t

6.3 Potential Health Hazards

5. Inhalation of concentrated vapors or mist may cause irritation of the respiratory tract.

6. Ingestion may cause severe liver and/or kidney damage, and may be fatal.

7. Contact with liquid, mist, or vapor can cause immediate irritation or corrosive burns to all human tissue. Severity is generally determined by the concentration of the solution and duration of exposure.

8. Contact with eyes may cause irritation, tearing and eye tissue discoloration, and may result in permanent visual loss unless removed quickly by thorough irrigation with water

6.4 Safety Precautions

5. Handle under a fume hood whenever possible.

6. Work with the fume hood splash guard as low as possible. This improves the effectiveness of the fume hood ventilation as well as providing protection from splashed solution.

7. Wear rubber gloves and eye protection whenever handling solution.

8. Ferric Chloride will stain anything that it contacts. You may want to wear an apron or lab coat to protect clothing.

6.5 Project Modules PCB Designs

All the PCB layouts were prepared in Protel environment using the appropriate footprints positioning on the required board area.

6.5.1 HDB3 Modules

6.5.1.1 HDB3 Encoder

a) Circuit Diagram

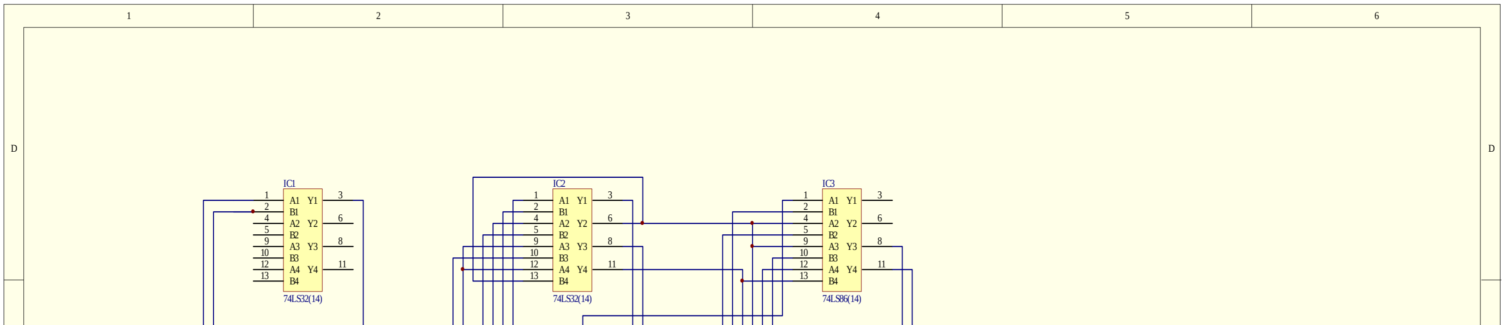


FIGURE 6.9 HDB3 encoder circuit diagram

b) PCB Layout

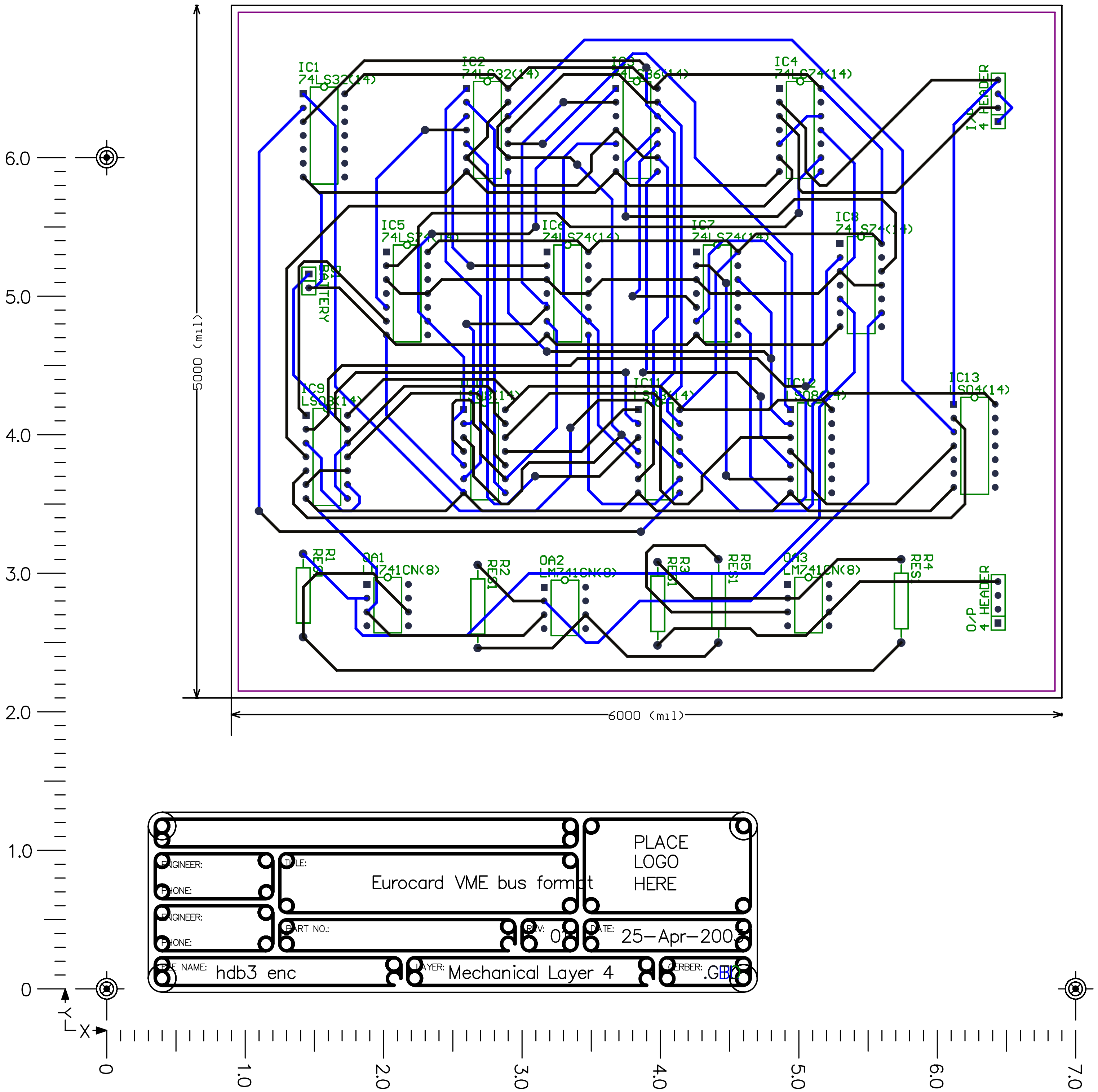

IC1
74LS32(14)
IC2
74LS32(14)
IC3
74LS86(14)
IC4
74LS74(14)
I/P
4 HEADER
IC5
74LS74(14)
IC6
74LS74(14)
IC7
74LS74(14)
IC8
74LS74(14)
B1
BATTERY
IC9
LS08(14)
LS08(14)
IC11
LS08(14)
IC12
LS08(14)
IC13
LS04(14)
R1
RES1
OA1
LM741CN(8)
R2
RES1
OA2
LM741CN(8)
R3
RES1
R5
RES1
OA3
LM741CN(8)
R4
RES1
O/P
4 HEADER
5000 (mil)
6000 (mil)
6.0
5.0
4.0
3.0
2.0
1.0
0
Y
X
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
ENGINEER:
PHONE:
ENGINEER:
PHONE:
TITLE:
Eurocard VME bus format
PLACE
LOGO
HERE
PART NO.:
REV:
01
DATE:
25-Apr-2003
FILE NAME:
hdb3 enc
LAYER:
Mechanical Layer 4
GERBER:
.GBD

FIGURE 6.10 HDB3 encoder PCB layout

6.5.1.2 HDB3 Decoder

c) Circuit Diagram

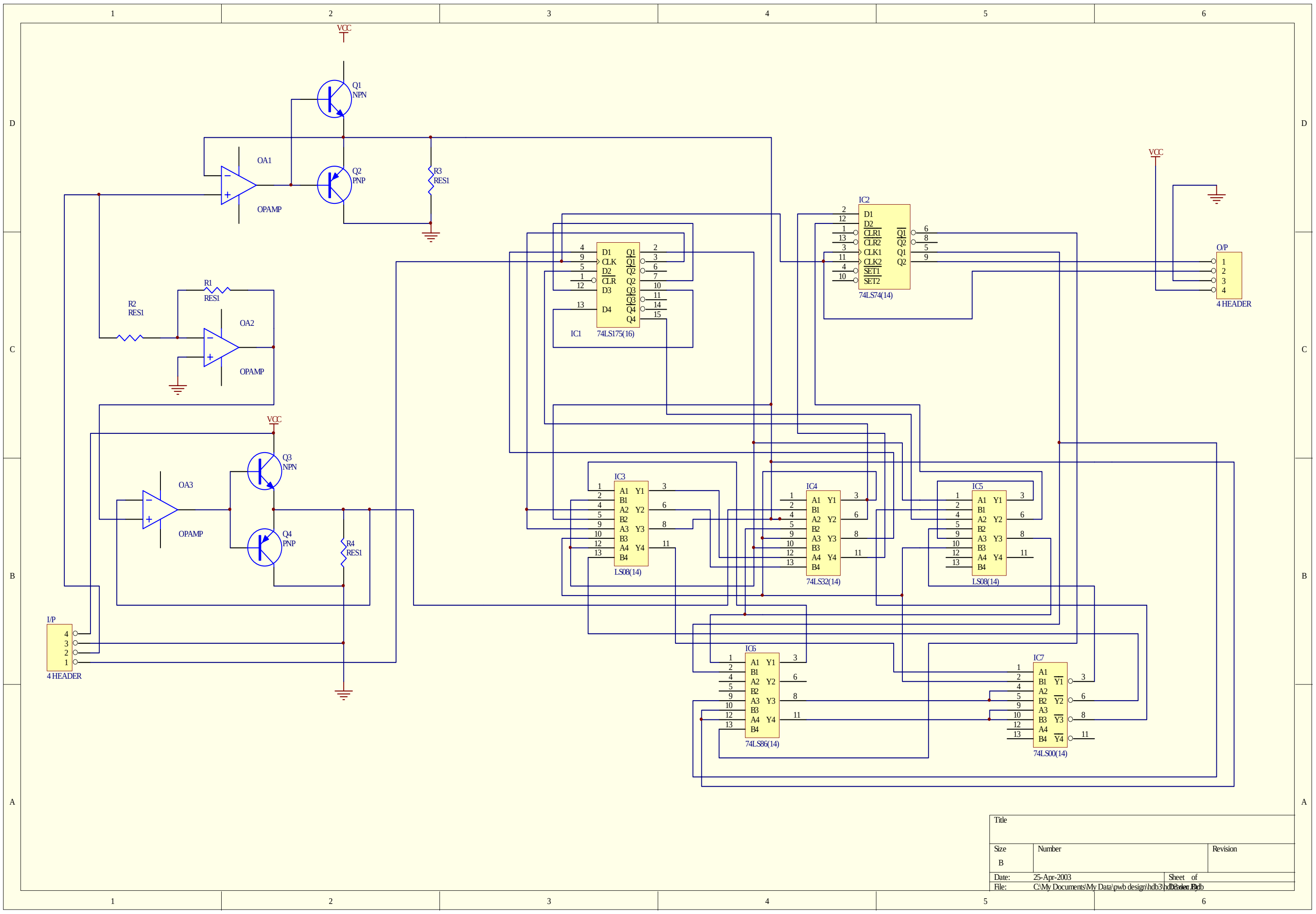


FIGURE 6.11 HDB3 decoder circuit diagram

d) PCB Layout

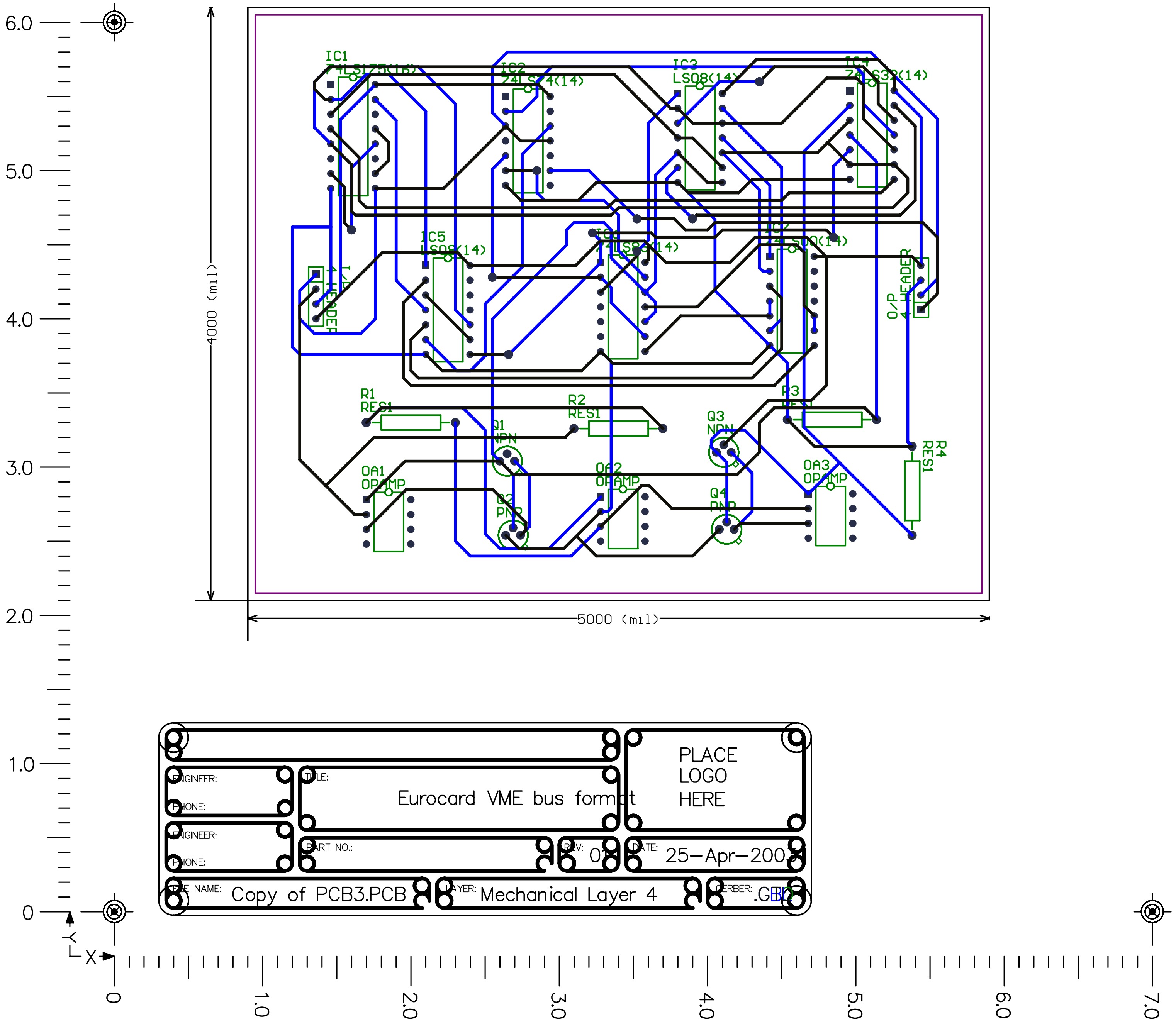

IC1
74LS175(16)
IC2
74LS74(14)
IC3
LS08(14)
IC4
74LS32(14)
IC5
LS08(14)
IC6
74LS86(14)
IC7
74LS00(14)
I/P
4 HEADER
O/P
4 HEADER
R1
RES1
R2
RES1
R3
RES1
R4
RES1
Q1
NPN
Q2
PNP
Q3
NPN
Q4
PNP
OA1
OPAMP
OA2
OPAMP
OA3
OPAMP
4000 (mil)
5000 (mil)
6.0
5.0
4.0
3.0
2.0
1.0
0
Y
X
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
ENGINEER:
PHONE:
ENGINEER:
PHONE:
TITLE:
Eurocard VME bus format
PLACE
LOGO
HERE
PART NO.:
REV:
01
DATE:
25-Apr-2003
FILE NAME:
Copy of PCB3.PCB
LAYER:
Mechanical Layer 4
GERBER:
.GBO

FIGURE 6.12 HDB3 decoder PCB Layout

6.5.2 2B1Q Modules

6.5.2.1 2B1Q Encoder

c) Circuit Diagram

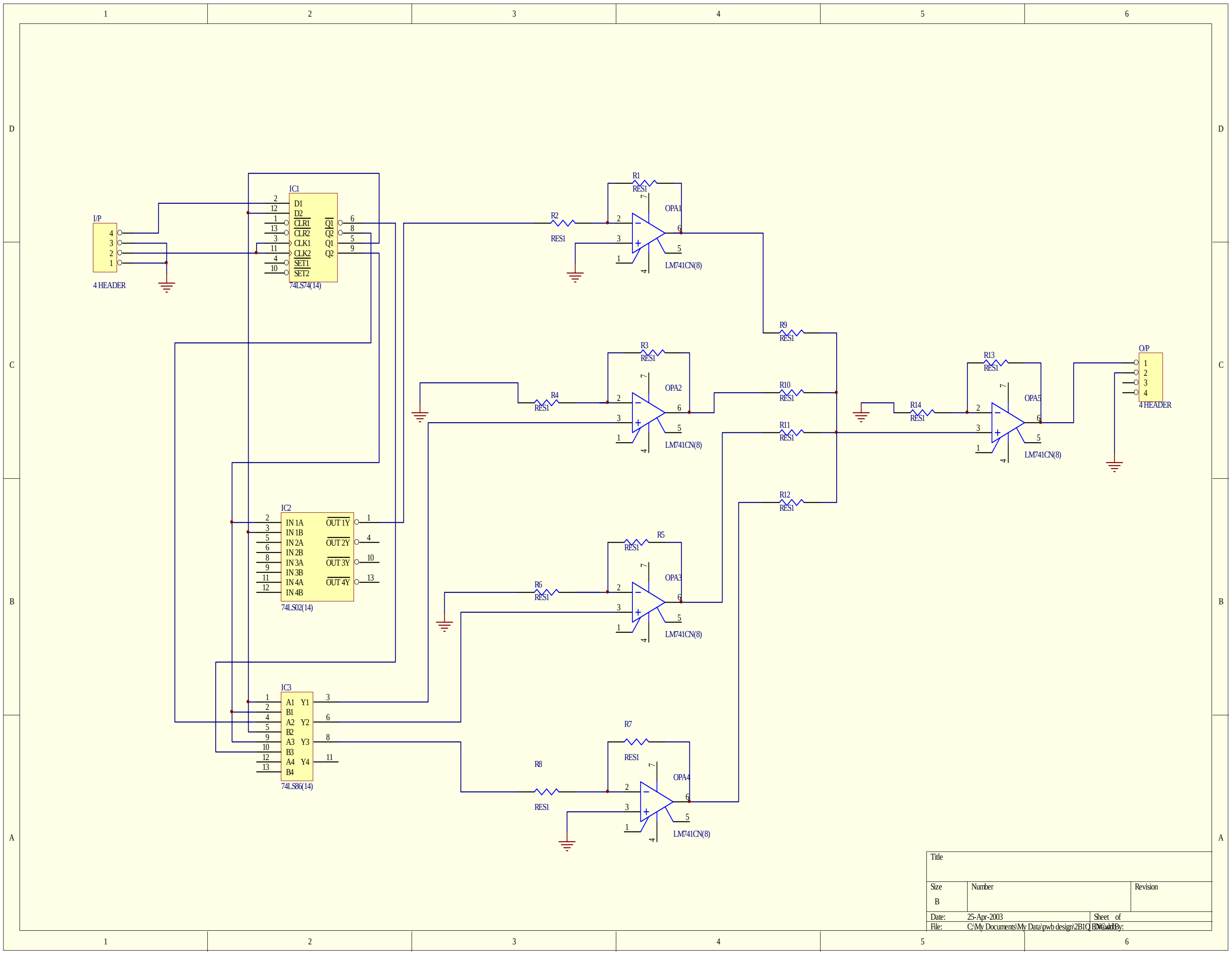


FIGURE 6.13 2B1Q encoder circuit diagram

6.5.1.2 HDB3 Decoder

e) Circuit Diagram

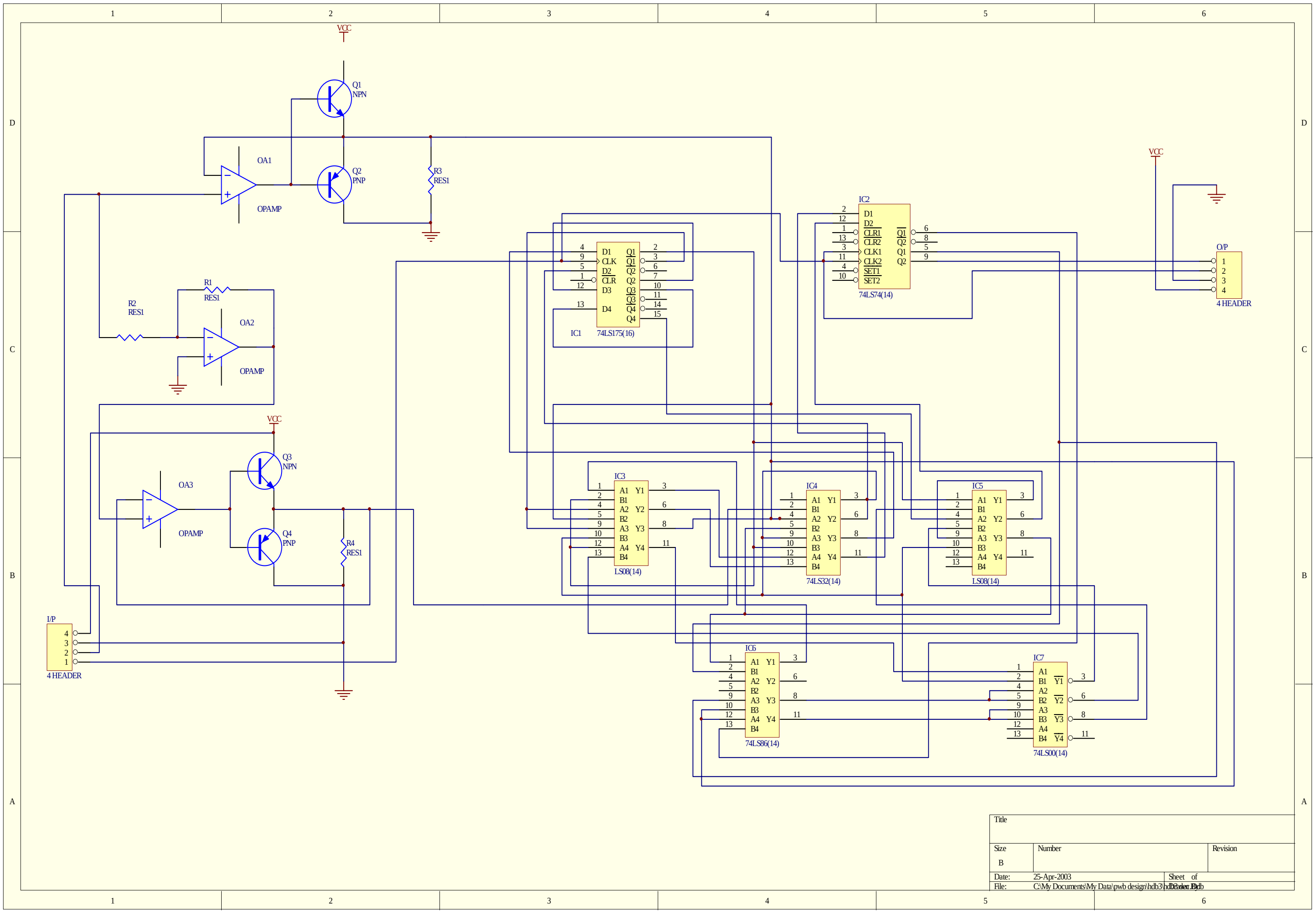


FIGURE 6.11 HDB3 decoder circuit diagram

f) PCB Layout

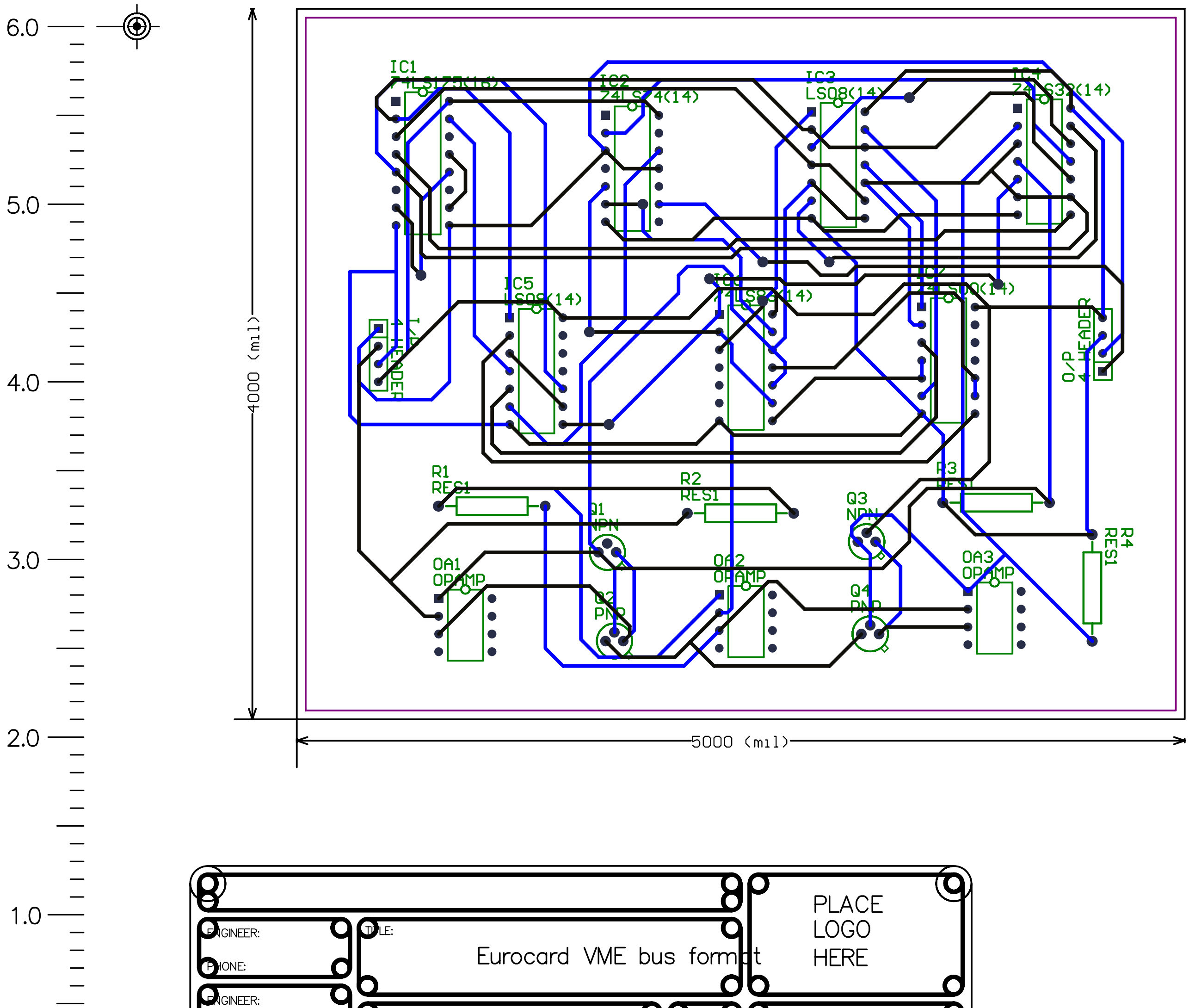


FIGURE 6.12 HDB3 decoder PCB Layout

6.5.2 2B1Q Modules

6.5.2.1 2B1Q Encoder

d) Circuit Diagram

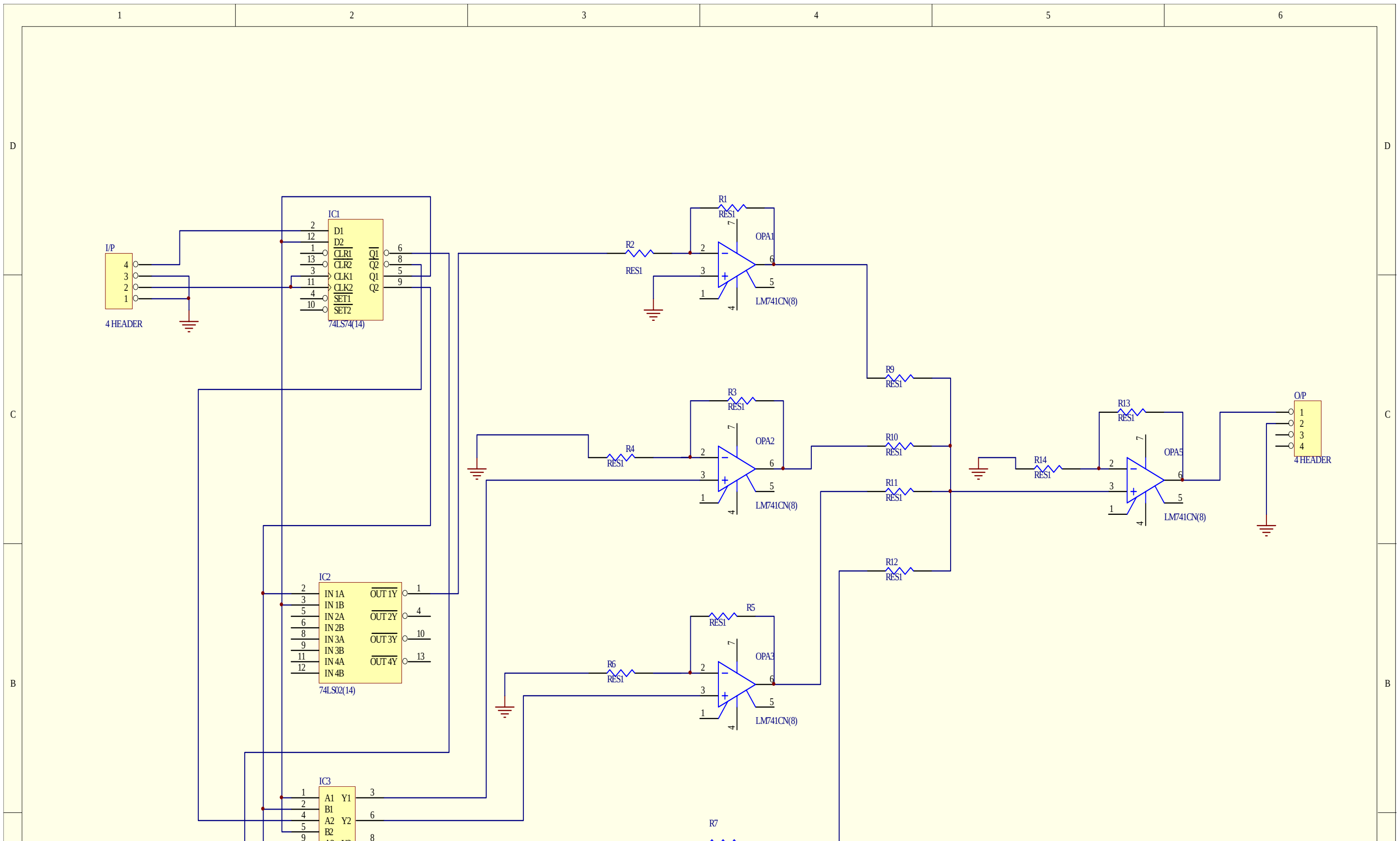


FIGURE 6.13 2B1Q encoder circuit diagram

e) PCB Layout

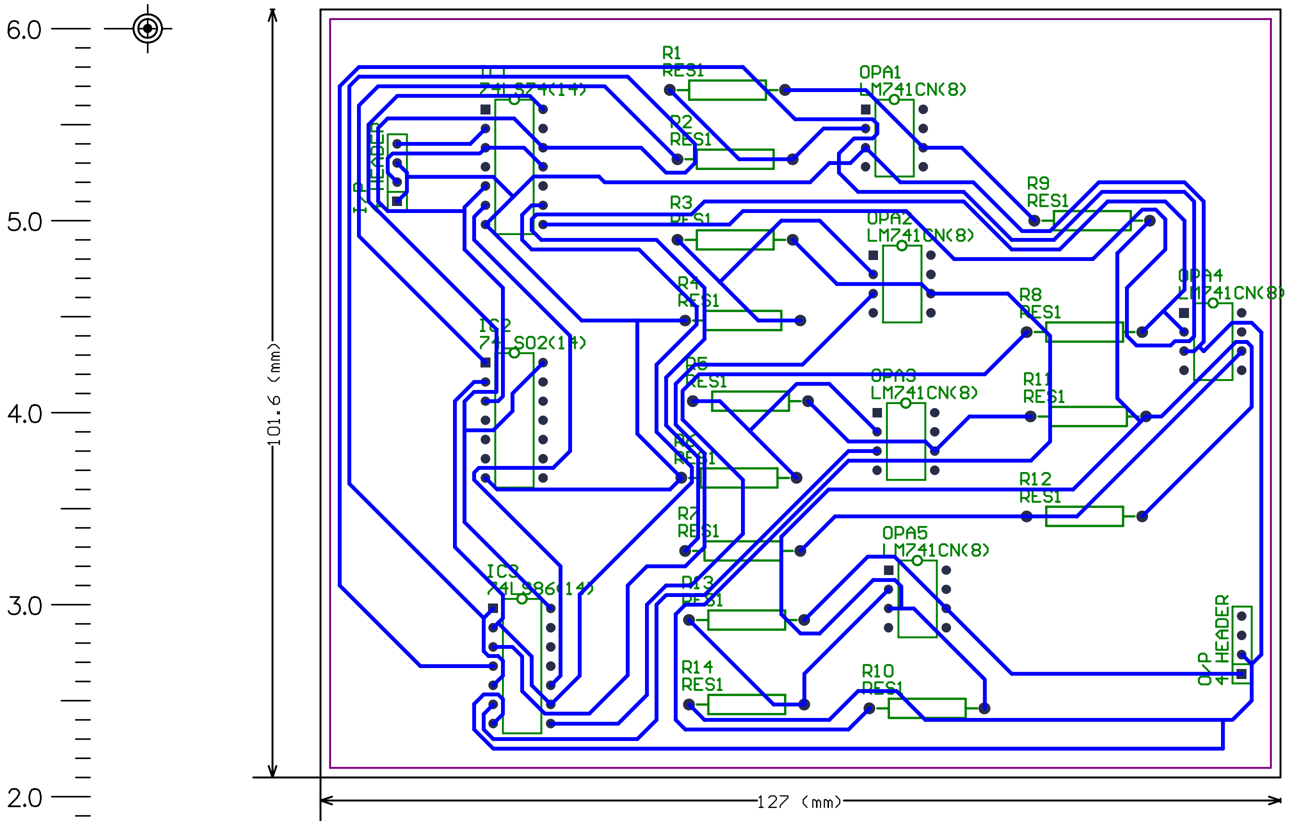


FIGURE 6.14 2B1Q encoder PCB Layout

6.5.2.2 2B1Q Decoder

c) Circuit Diagram

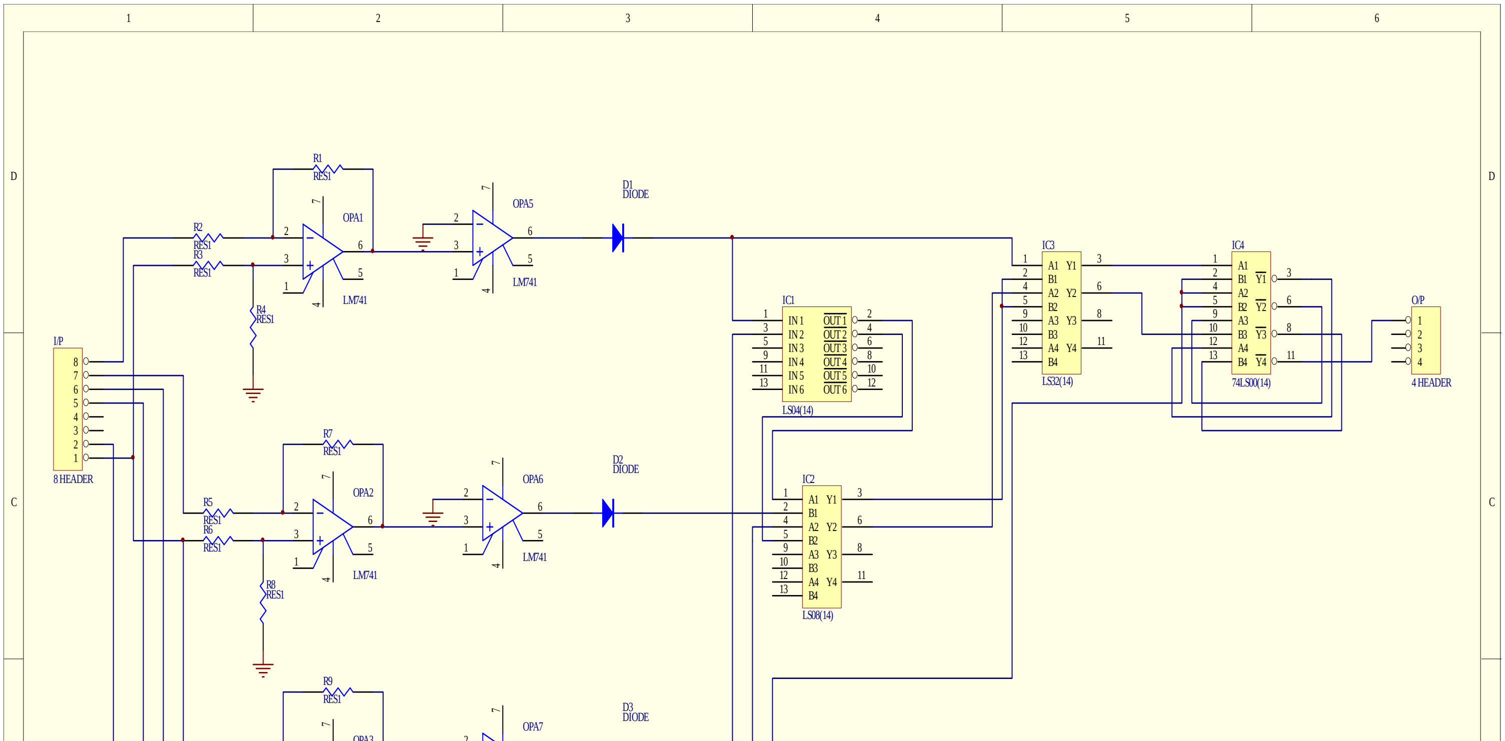
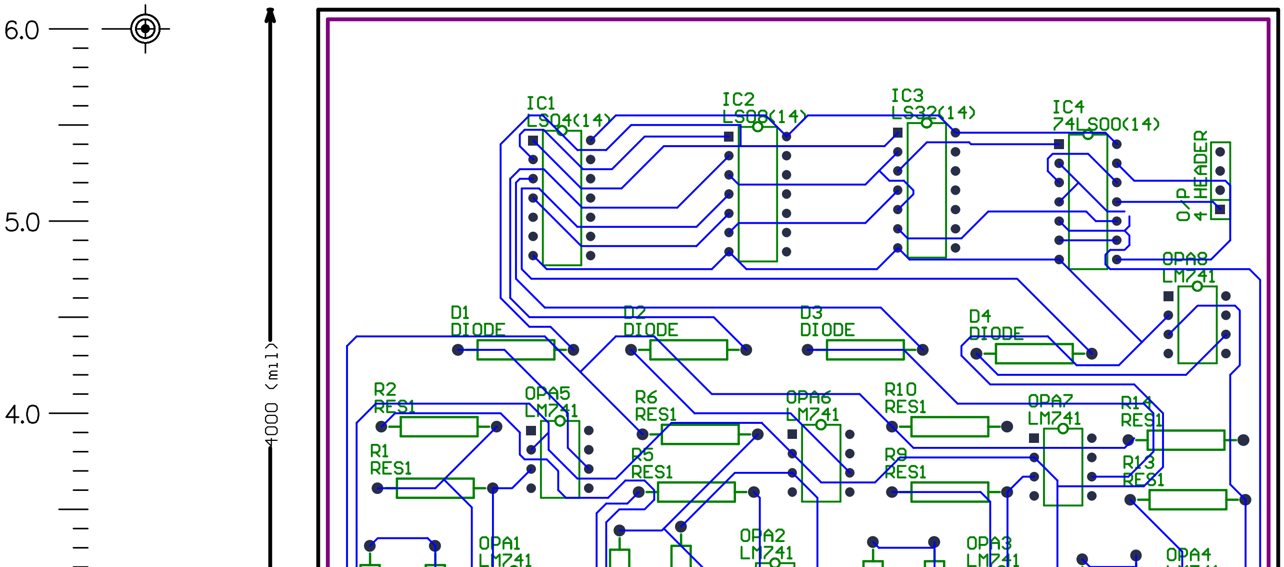


FIGURE 6.15 2B1Q Decoder circuit diagram

d) PCB Layout



References

1. Low Cost, Low Power Devices For HDSL Applications by *Ed Spencer* www.google.com

2. Omniplexer Voice/Data Multiplexer by *Bayly Comm, Inc* www.bayly.com

3. ISDN Primary Rate (E1/T1) Interface by *IBM Corp*. www.ibm.com

4. Why DSL Still Matters by *TeleChoice Telecomm Industry* www.telechoice.com

5. HDSL Basics by *TTC* www.ttc.com

6. 2.048 Mbps Technology Basics And Testing Fundamentals by *TTC* www.ttc.com

7. 2B1Q Tutorial by *Zarlink Semiconductors* www.zarlink.com

8. TP 3410 ISDN Basic Access by *NSC* www.nsc.com

9. EWB Tutorial by *Prentice Hall, Inc.* www.prenhall.com

10. Multisim 2001 Getting Started and Tutorial by *Interactive Image Technologies* www.electronicworkbench.com

11. Protel Tutorial by *Protel* www.protel.com

12. Transformerless Power Supply by *Ronj* www.mitedu.freeserve.co.uk

13. How to make PCBs www.kingim.btinternet.co.uk