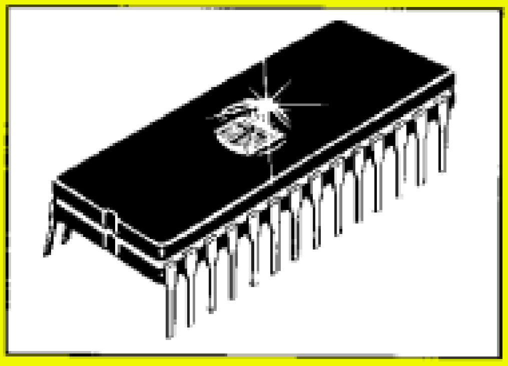
Recommended By
Lou Franklin

THE CB EPROM DATA BOOK

Advanced Frequency Modification Techniques



A Reference Work By

Martin T Pickering

THE CB EPROM DATA BOOK

This book is provided, by Lou Franklin and Martin Pickering, for FREE download in Portable Document Format from the following web sites:

http://www.cbcity.com

http://www.netcentral.co.uk/satcure

Other free gifts may be made available in the future so keep visiting the web sites!

CONTENTS

Written By: Martin T. Pickering - *“The Specialist”*

Third Edition 1997.

The information in this book is presented for educational purposes only and is not intended as an endorsement of any particular practice. The emphasis is upon using Eproms to convert existing CB transceivers to operate on the Amateur 10 meter band for the use of licensed radio amateurs. The publisher seeks neither to encourage nor endorse the illegal ownership or the operation of equipment for which the holder is not licensed.

INTRODUCTION

My interest in CB conversions began some years after Lou Franklin first published his "Screwdriver Expert's Guide" and "The CB PLL Data Book". Consequently I was able to read these and progressed quickly from having a passing interest in CB to actually running a repair business and publishing a quarterly newsletter for like-minded individuals. Lou's books have been a guiding influence in my hobby-turned-business and I am in no way putting them down when I suggest that they lack just one important chapter. I hope that this book will fill that gap and help you as much as Lou's books helped me.

This book assumes that you have read "The CB PLL Data Book" and have a basic grasp of the way in which a PLL frequency synthesiser works. The use of an Eprom will allow you to utilize the standard channel switch but to program the codes which will make the rig operate on whatever channels, in whatever order, that you wish (within the limitations of the rig itself). Most modern AM or FM CB radios use a dedicated PLL whose codes can not be altered. Sideband rigs, however, can almost always be converted for 10 meter operation because they still use the type of PLL which has a wide choice of codes. Even those with dedicated PLLs can often be altered if an Eprom is used in conjunction with a different PLL.

The book will be of especial interest to those who are looking for a low cost entry to the American "Novice" band.

This book is for all those CBers and Radio Amateurs out there who like to indulge in the practical aspects of the hobby.

My grateful thanks are due to my wife, Gill, for her patience; to my kids, for keeping out of the way; to Brian Spiller, who suggested the idea and checked the proofs; to my brother, Rob, for the use of his computer; to Chas for the use of his laser printer and to Lou and Sally Franklin for their continued support.

Dave Addy designed and drew all the PCB layouts and their circuits.

Cover design and “Gang Bang” drawing by:
Studio 3, Palace Vaults, Plymouth, PL1 2NA, England.

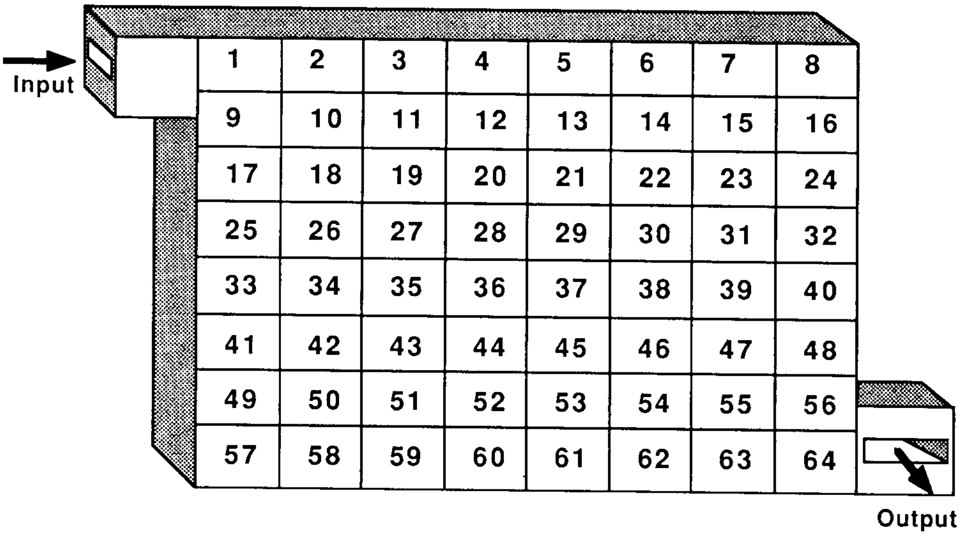
Other sketches by Martin – drawn large then reduced down to hide the defects!

Simple as A B C D E F

Mail Boxes

The Eprom is a basic, mechanical device which can be considered simply as a block of "mail boxes", each holding a message. Each "box" is numbered consecutively. At one end of the block of "boxes" is a slot and by posting the Address or LOCATION NUMBER of a particular "box" into this slot you will make its message contents appear from the output slot at the other end of the block.

You can read the contents of ANY "box" in this manner if you know its number.



Mail Boxes

Messages in Code

The actual message in each box consists of just eight numbers, each of which can be "1" or "0".

So a typical message might be 1 1 0 1   0 0 0 1

This you will recognise as precisely the same sort of code which a channel switch supplies to a PLL in order to select a channel.

The number of each box can also be represented as a code.

For instance:-
the first box may be numbered 0 0 0 0 0 0 0 0 0 0 0 1
and the second box could be 0 0 0 0 0 0 0 0 0 0 1 0

In practice, the box LOCATION NUMBER is applied to the Eprom input pins as high or low voltages representing "1" or "0".

The CONTENTS of the selected box appear on the eight Eprom output pins as high or low voltages.

For instance:-

	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
inputs												
box number-	0	0	0	0	0	0	1	0	1	0	1	0

might have the

	D7	D6	D5	D4	D3	D2	D1	D0	outputs
contents-	1	1	1	0	1	0	0	1	

As the LOCATION NUMBER of each box is applied to the input pins, so the CONTENTS of that box appear as voltages on the 8 output pins.

Select a Channel

Suppose we connect the Eprom between the channel selector switch and the PLL. Normally, the code from the switch will tell the PLL which channel to select but now we have altered things.

The code from the switch is seen by the Eprom to be a LOCATION NUMBER for a box. The Eprom puts the CONTENTS of this box on its output pins. The PLL sees a channel code which is DIFFERENT from the switch code and is "fooled" into producing the wrong channel!

All we have to do is to find out what code number comes from the switch for EACH channel and put into each box SELECTED BY THIS NUMBER the NEW code which we want the PLL to see. Putting the codes into the Eprom boxes is called PROGRAMMING and is a simple task which anyone can do, provided, of course, that he has an Eprom Programmer! A Programmer can be bought VERY cheaply and used in conjunction with a personal computer even if you have absolutely no knowledge of computer programming.

Bits in Line

Eproms work quite happily with binary codes consisting only of "1"s and "0"s, called "bits". We use the designations "1" and "0" simply for convenience to represent high voltage and low voltage respectively. The actual voltages are usually not critical - high will be some positive voltage equal to, or slightly less than, the supply voltage. Low voltage will usually be less than 1 volt. The position of each "bit" in line represents its actual value:-

	128	64	32	16	8	4	2	1	
the code	1	0	1	1	0	0	1	0	adds up to 178 in decimal.

Horrible Hex

That was a little tedious to work out but it is difficult to use decimal with Eproms. To make the numbers easier to deal with we can use the system called Hexadecimal. Before you shriek in horror at this word let me advise you that everything you ever read about how difficult it is, is nonesense. It makes life very easy indeed.

"Hexadecimal uses the base sixteen". That means that you count up to 16, carry one, then start again. In fact we use letters to represent numbers bigger than 9, as follows:-

0 1 2 3 4 5 6 7 8 9 A B C D E F

F represents ordinary decimal 15 and is as high as we go. To count higher we add another column and start again so ordinary 16 is represented by "10" (that’s “one-zero” HEX) and we count onwards in HEX ....

10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F

..and if you've followed so far you will understand that "1F" HEX represents a 16 in the first column plus a 15 in the second = 31.

Can you see that "20" HEX must be 32 ?

Now, we can convert binary to HEX very easily:-

```
8 4 2 1

0 0 0 0 = 0
0 0 0 1 = 1
0 0 1 0 = 2
0 0 1 1 = 3
0 1 0 0 = 4
0 1 0 1 = 5
0 1 1 0 = 6
0 1 1 1 = 7
1 0 0 0 = 8
1 0 0 1 = 9
1 0 1 0 = A
1 0 1 1 = B
1 1 0 0 = C
1 1 0 1 = D
1 1 1 0 = E
1 1 1 1 = F
```

You seldom need to consider decimal numbers at all. Usually you can go directly from binary to hex.

Get used to splitting the binary number into groups of four, with the Least Significant Bit (LSB) on the right hand side. If you split it up, then - it doesn't matter how big it is - you can always convert each group into hex and achieve the complete answer. Consider the following 11 Bit number:-

1024	512	256	128	64	32	16	8	4	2	1
0	1	0	0	0	1	1	1	0	1	1

can be written as

8	4	2	1	8	4	2	1	8	4	2	1
	0	1	0	0	0	1	1	1	0	1	1

= 2 + 2 +1 + 8 + 2 +1

= 2 3 B

= 23B HEX

Since this was an 11 Bit number it could not be divided exactly into groups of four but, as you see, it didn't matter. We ignore any bits missing at the left hand side, as if they were zero.

If we always split the binary codes into groups of four, the conversion remains easy:-

```
8 4 2 1   8 4 2 1   8 4 2 1

1 0 0 1   1 1 1 1   0 0 0 0   =   9 F 0

1 1 1 0   1 0 1 0   1 0 0 1   =   E A 9

  0 0 0   1 0 1 0   1 0 0 1   =     A 9
```

If you can count up to sixteen you can use Hexadecimal!

Conversion Tables

Now look at a table which shows part of the Eprom contents for a typical conversion. All numbers are in Hexadecimal. Each box actually contains an eight bit binary code which is shown as a 2 digit HEX number because otherwise the page isn't wide enough!

```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F

000   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 0F
001   10 11 FF 13 14 15 16 FF 18 19 1A 1B FF 1D 1E 1F
002   20 FF 22 23 24 25 FF 27 28 29 2A 2B 2C 2D 2E 2F
003   30 31 32 33 34 35 36 37 38 39 3A 3B FF FF FF FF
004
005 Out of this block of sixty four boxes, only forty are
006 selected by the channel switch whose codes range from 0F to
007 3B (HEX). The unused boxes in an Eprom are usually FULL, not
008 empty, and produce 1 1 1 1 1 1 1 1 on the output pins. This
009 table has been arranged  so that the contents of each box
00A are identical to the channel switch code which selects it.
00B For instance the switch code 2A selects the box whose
00C contents  are also 2A. This is because, in most conversions,
00D we want to KEEP the existing channels as well as to add new
00E ones. The new channels will be added in another block thus:-
00F
010   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3C
011   3D 3E FF 40 41 42 43 FF 45 46 47 48 FF 4A 4B 4C
012   4D FF 4F 50 51 52 FF 54 55 56 57 58 59 5A 5B 5C
013   5D 5E 5F 60 61 62 63 64 65 66 67 68 FF FF FF FF
```

These codes continue where the original ones ended so the box selected by switch code 1F is 3C. Note that the switch can only select 40 channels and will normally use six of the Eprom input pins to do this. An Eprom has more than six inputs, however, and any of these can be used to select another bank of 64 boxes.

Note, also, that five boxes are not used. These represent the "A" channels, "hops" or "skips" in FCC rigs. The Eprom can, of course, be programmed so that these channels are included, at the expense of losing five other channels in this bank, since the channel switch can select only forty in any one bank of 64.

The position of channel 23 is odd and the effect is as if the channel switch goes ---26-23-25-24--, so this arrangement in most FCC and European rigs must be taken into account when any new codes are calculated*. This comment applies especially when you are programming a straight sequence of channels without any "hops". The Eprom contents would then look like this:-

```
020   FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 3C
021   3D 3E FF 3F 40 41 42 FF 43 44 45 46 FF 47 48 49
022   4A FF 4B 4C 4D 4E FF 4F 50 51 53 54 52 55 56 57
023   58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 FF FF FF FF
```

The switch does not produce codes for the "hops" and can not access these Eprom locations. Also, channel 23 position is effectively moved. Nevertheless, with correct programming of the 40 Eprom LOCATIONS which ARE addressed by the switch, the result is a "straight" 40 channels selected by a channel switch which was never intended to give such a sequence! For the early UK channels and for the Amateur 10 meter band, this ability to produce a band of channels without gaps is especially useful.

You can understand, now, that an Eprom will allow you to use the standard codes, from a channel switch, to select Eprom LOCATIONS which contain any codes which your PLL will accept. The only limitations on frequency coverage would then be the actual "N" code set available in the PLL and the restricted bandwidth of the rig itself. Even the restriction of the "N" code set can be alleviated by changing the PLL for one with a larger set of codes.

An interesting point, in considering the use of an alternative PLL, is that you would be wise to choose one which will run on a higher supply voltage than the original because this will widen the voltage range of the PD output and automatically give an increased VCO frequency range. The MC145106P, for instance, can use a Vcc supply of up to 12 volts, whereas a uPD858, for example, can use only 5 volts, maximum.

* The hiistory of this odd channel arrangement is that, originally, only 22 channels were allocated. Subsequently, however, channel 23 was also authorized, 30kHz higher than 22. When, finally, the full 40 channels were permitted, channels 24 and 25 were allotted to the gap between 22 and 23 wiith channels 26 onwards falling in line above 23.

Note: In most counttries it is illegal to possess a transmitter which is capable of operating on CB frequencies as well as Amateur Radio frequencies.

PRACTICAL CONSIDERATIONS

Which Eprom?

The smallest Eprom currently available is the 2708. This is actually obsolete and has the disadvantage of requiring three different supply voltages: hence, it is never used for new designs. The next larger one is the 2516 or 2716 which, although small, nevertheless has room for 32 banks of 40 channels! It has 11 input pins of which only six would usually be connected to the channel switch. (Although some channel switches have more than six connections it is normally possible to get 40 unique codes from just six of them. Since we are going to use these switch codes merely to select the LOCATIONS in the Eprom which will contain the ACTUAL codes for the PLL, it hardly matters what the switch codes are, provided there are 40 which are all different.) The remaining 5 input pins, therefore, can be connected to a band selection switch to select each bank of 40 channel codes. Up to 32 permutations are possible - far more than we would ever need! This discussion assumes a 40 position switch, by the way, but the concept is equally valid for rigs with 18, 22 or 50 positions.

The next size of Eprom is the 2532 or 2732 which has 12 input pins, giving a possible 64 banks of channels from a 40 position channel switch. The 2764 Eprom has 13 input pins and the 27128 has 14. Finally, the 27256 has 15 inputs. Getting silly, isn't it. Who needs room for 512 x 40 = 20480 channels when the maximum coverage of the rig is unlikely to exceed 180 channels at 10 kHz spacing? Nevertheless, these big Eproms are becoming cheaper than the small ones and you might wish to use them, despite the waste of "space" inside them.

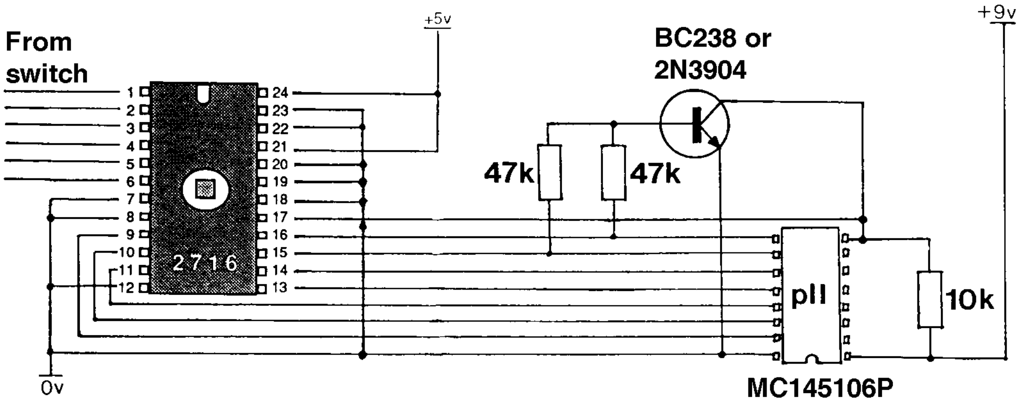
Outputs

No matter how many input pins an Eprom has, the number of output pins is always eight. This is because every "box" or location can hold only eight binary bits (or one "byte" as it is known). This fact puts one big limitation on the use of an Eprom. What do we do, for instance, if we want to connect the EIGHT Eprom outputs to a PLL with NINE input pins?

Transistor Inverter

If the codes require the 9th pin always to be a "1" or always a "0" then there is no problem. We connect that pin to the supply and ignore it. If, however, the state of the pin must change between "1" and "0" then we do have a problem. The answer is usually to connect the 8th pin to the base of a transistor whose collector is connected to the ninth pin. The 9th pin will then

always have the inverse voltage from that which is applied to the 8th pin. When the 8th pin is "1" the 9th pin will always be "0". (The Export version Cobra 148 GTL DX used this system with a 15.45 MHz downmix crystal to begin with. Eventually, however, it occurred to somebody that by using a 15.00 MHz crystal the range of codes could be shifted to make pin 9 permanently low and the transistor was no longer needed!)



A Transistor Inverter

Range of Codes

It can be shown that this method can increase the possible number of codes by up to 128 within a particular range, but does not increase the overall number of codes possible with an Eprom, which will always be 256 due to the limitation of having only eight output pins. Sometimes the transistor base is connected to both the 7th and the 8th pin. In this way, the range of codes is shifted a little further and the 9th pin will be low if either the 7th pin or 8th pin or both are high. The following list of binary codes will help you to visualise this concept. Remember that a total of only 256 codes can be used, but the range can be moved up or down by use of the transistor inverter.

```
 1 0100 0000  1 0000 1010  0 1101 0100  0 1001 1110  0 0110 1000
•1 0011 1111  1 0000 1001  0 1101 0011  0 1001 1101  0 0110 0111
 1 0011 1110  1 0000 1000  0 1101 0010  0 1001 1100  0 0110 0110
 1 0011 1101  1 0000 0111  0 1101 0001  0 1001 1011  0 0110 0101
 1 0011 1100  1 0000 0110  0 1101 0000  0 1001 1010  0 0110 0100
```

```
1 0011 1011  1 0000 0101  0 1100 1111  0 1001 1001  0 0110 0011
1 0011 1010  1 0000 0100  0 1100 1110  0 1001 1000  0 0110 0010
1 0011 1001  1 0000 0011  0 1100 1101  0 1001 0111  0 0110 0001
1 0011 1000  1 0000 0010  0 1100 1100  0 1001 0110  0 0110 0000
1 0011 0111  1 0000 0001  0 1100 1011  0 1001 0101  0 0101 1111
1 0011 0110  1 0000 0000  0 1100 1010  0 1001 0100  0 0101 1110
1 0011 0101  0 1111 1111  0 1100 1001  0 1001 0011  0 0101 1101
1 0011 0100  0 1111 1110  0 1100 1000  0 1001 0010  0 0101 1100
1 0011 0011  0 1111 1101  0 1100 0111  0 1001 0001  0 0101 1011
1 0011 0010  0 1111 1100  0 1100 0110  0 1001 0000  0 0101 1010
1 0011 0001  0 1111 1011  0 1100 0101  0 1000 1111  0 0101 1001
1 0011 0000  0 1111 1010  0 1100 0100  0 1000 1110  0 0101 1000
1 0010 1111  0 1111 1001  0 1100 0011  0 1000 1101  0 0101 0111
1 0010 1110  0 1111 1000  0 1100 0010  0 1000 1100  0 0101 0110
1 0010 1101  0 1111 0111  0 1100 0001  0 1000 1011  0 0101 0101
1 0010 1100  0 1111 0110  0 1100 0000  0 1000 1010  0 0101 0100
1 0010 1011  0 1111 0101  0 1011 1111  0 1000 1001  0 0101 0011
1 0010 1010  0 1111 0100  0 1011 1110  0 1000 1000  0 0101 0010
1 0010 1001  0 1111 0011  0 1011 1101  0 1000 0111  0 0101 0001
1 0010 1000  0 1111 0010  0 1011 1100  0 1000 0110  0 0101 0000
1 0010 0111  0 1111 0001  0 1011 1011  0 1000 0101  0 0100 1111
1 0010 0110  0 1111 0000  0 1011 1010  0 1000 0100  0 0100 1110
1 0010 0101  0 1110 1111  0 1011 1001  0 1000 0011  0 0100 1101
1 0010 0100  0 1110 1110  0 1011 1000  0 1000 0010  0 0100 1100
1 0010 0011  0 1110 1101  0 1011 0111  0 1000 0001  0 0100 1011
1 0010 0010  0 1110 1100  0 1011 0110  0 1000 0000  0 0100 1010
1 0010 0001  0 1110 1011  0 1011 0101  0 0111 1111  0 0100 1001
1 0010 0000  0 1110 1010  0 1011 0100  0 0111 1110  0 0100 1000
1 0001 1111  0 1110 1001  0 1011 0011  0 0111 1101  0 0100 0111
1 0001 1110  0 1110 1000  0 1011 0010  0 0111 1100  0 0100 0110
1 0001 1101  0 1110 0111  0 1011 0001  0 0111 1011  0 0100 0101
1 0001 1100  0 1110 0110  0 1011 0000  0 0111 1010  0 0100 0100
1 0001 1011  0 1110 0101  0 1010 1111  0 0111 1001  0 0100 0011
1 0001 1010  0 1110 0100  0 1010 1110  0 0111 1000  0 0100 0010
1 0001 1001  0 1110 0011  0 1010 1101  0 0111 0111  0 0100 0001
1 0001 1000  0 1110 0010  0 1010 1100  0 0111 0110 •0 0100 0000
1 0001 0111  0 1110 0001  0 1010 1011  0 0111 0101  0 0011 1111
1 0001 0110  0 1110 0000  0 1010 1010  0 0111 0100  0 0011 1110
1 0001 0101  0 1101 1111  0 1010 1001  0 0111 0011  0 0011 1101
1 0001 0100  0 1101 1110  0 1010 1000  0 0111 0010  0 0011 1100
1 0001 0011  0 1101 1101  0 1010 0111  0 0111 0001  0 0011 1011
1 0001 0010  0 1101 1100  0 1010 0110  0 0111 0000  0 0011 1010
1 0001 0001  0 1101 1011  0 1010 0101  0 0110 1111  0 0011 1001
1 0001 0000  0 1101 1010  0 1010 0100  0 0110 1110  0 0011 1000
1 0000 1111  0 1101 1001  0 1010 0011  0 0110 1101  0 0011 0111
1 0000 1110  0 1101 1000  0 1010 0010  0 0110 1100  0 0011 0110
1 0000 1101  0 1101 0111  0 1010 0001  0 0110 1011  0 0011 0101
1 0000 1100  0 1101 0110  0 1010 0000  0 0110 1010  0 0011 0100
1 0000 1011  0 1101 0101  0 1001 1111  0 0110 1001  0 0011 0011
```

A range of 256 “N” codes

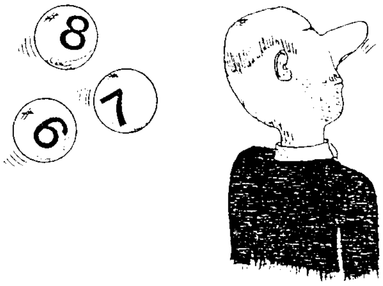
In the foregoing table each bit of the 9 bit code has a value according to its position from left to right, for instance:-

```
256 128  64  32  16  8   4   2   1
 1 . 0   1   0   0 . 0   0   0   0  =  140 HEX

or 256+64  =  320 decimal
```

The left-most bit is called the Most Significant Bit (MSB) and since an Eprom has only 8 output pins, as mentioned, the MSB must be generated external to the Eprom. In the table the range between "a" and "a" obeys a rule that, if either Eprom output in columns 2 and 3 (or both) is "1", then the MSB must be "0". The simple transistor inverter will perform this operation.

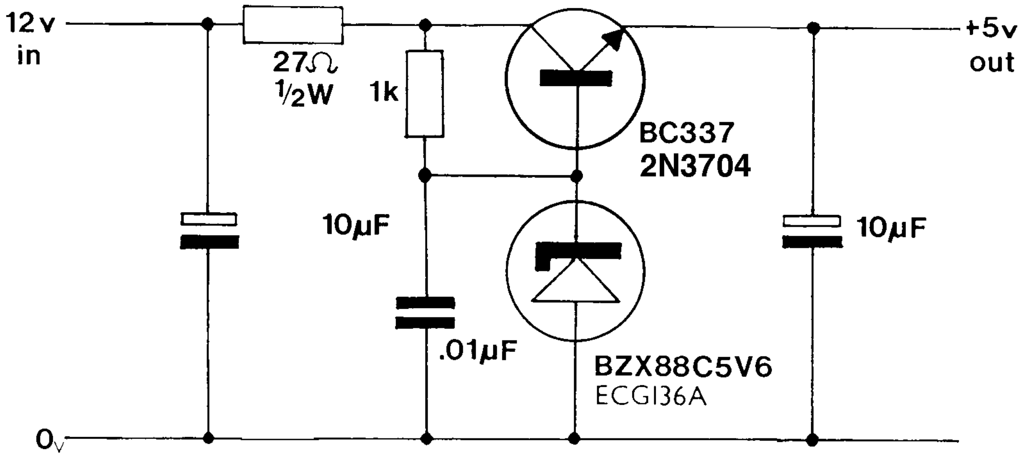
Sometimes you will need to override the operation of the transistor inverter for one band and it is quite straightforward to hold the transistor base connection permanently high, via a small value resistor, to keep the MSB low. Another method which has been used to "expand" the number of outputs from an Eprom is to connect a 74LS157 data selector IC to the Eprom but a full description of such a device is beyond the scope of this book. At present, eight outputs should be quite enough for you! By the way; in some instances you will not need to use all eight outputs, in which case simply ignore those which you don't want.



You will find it less confusing if you arrange the internal Eprom codes such that the unused output is always "0", although it is not actually necessary, since you will not be connecting the output to anything. It is worth mentioning, here, that unused Eprom outputs do not need to be connected to anything and most certainly should NEVER be connected directly to either ground or to a supply voltage. Outputs may be connected only to the input pins of other ICs or to resistors of at least 10k in value. However, Eprom inputs which are not used should always be connected to ground, otherwise the ADDRESS of the codes will be altered and the Eprom will have to be programmed accordingly.

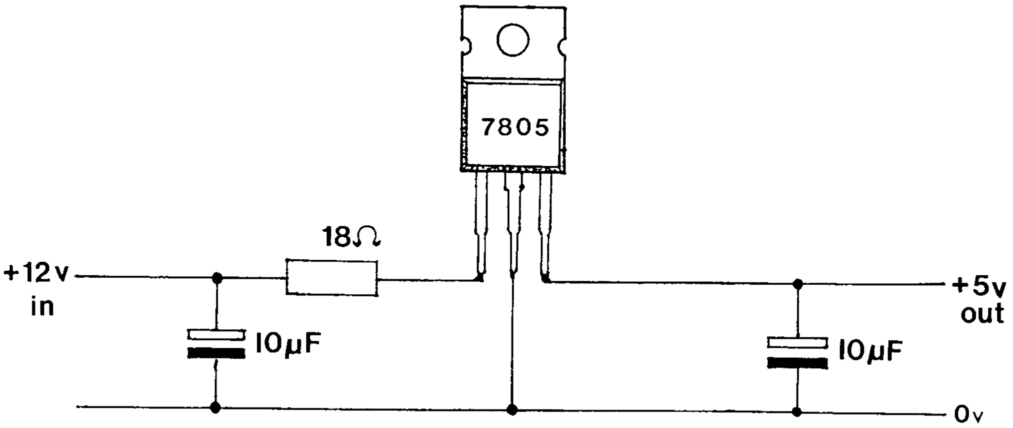
Operating Voltage

One point which is MOST important is the operating voltage of the Eprom. All those currently available require a supply voltage of 4.75 to 5.25 volts. A zener stabilized supply is essential and I would further recommend the use of a transistor series regulator. The transistor should be rated at 1 Watt minimum, especially for older Eproms whose operating current could exceed 100 milliamps.



A Series Regulator using a transistor

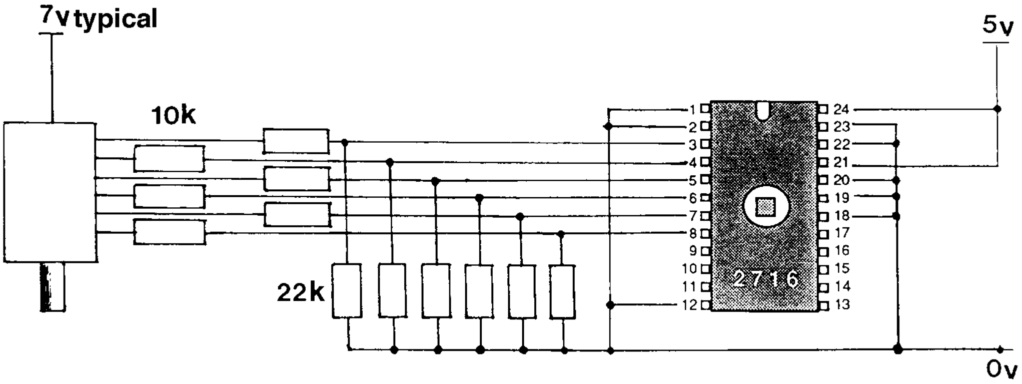
Alternatively, you could use a regulator device, such as a 7805. This is more expensive than a transistor alone but requires no zener diode or resistor. Regulators are available in various sizes and ratings from 78L05 (100mA) to 7805 (1A).



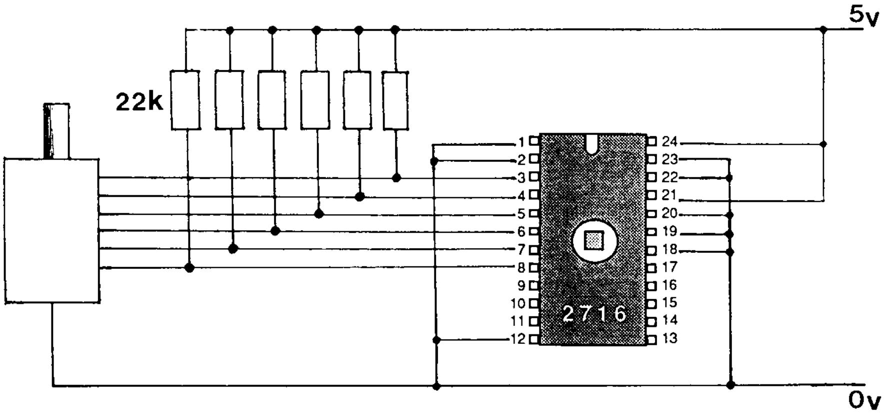
A series regulator using a regulator device

Input Voltage

The voltage applied to each input should not exceed the supply voltage or damage will eventually result. If the voltage from the channel switch is more than 5 volts then a couple of resistors are needed on each input to reduce the voltage seen by the Eprom. Some channel switches do not supply a positive voltage but ground the inputs, instead. Where this is true you must connect a resistor between each switch connection and the 5 volt supply to the Eprom. You could even introduce this method yourself where the channel switch voltage is high. Simply remove the supply wire from the switch common terminal(s) and ground the switch common. Now you need only half the number of resistors. The switch codes will all be inverted, of course but, since they are simply SELECTING Eprom locations this hardly matters, provided YOU know what the codes are!



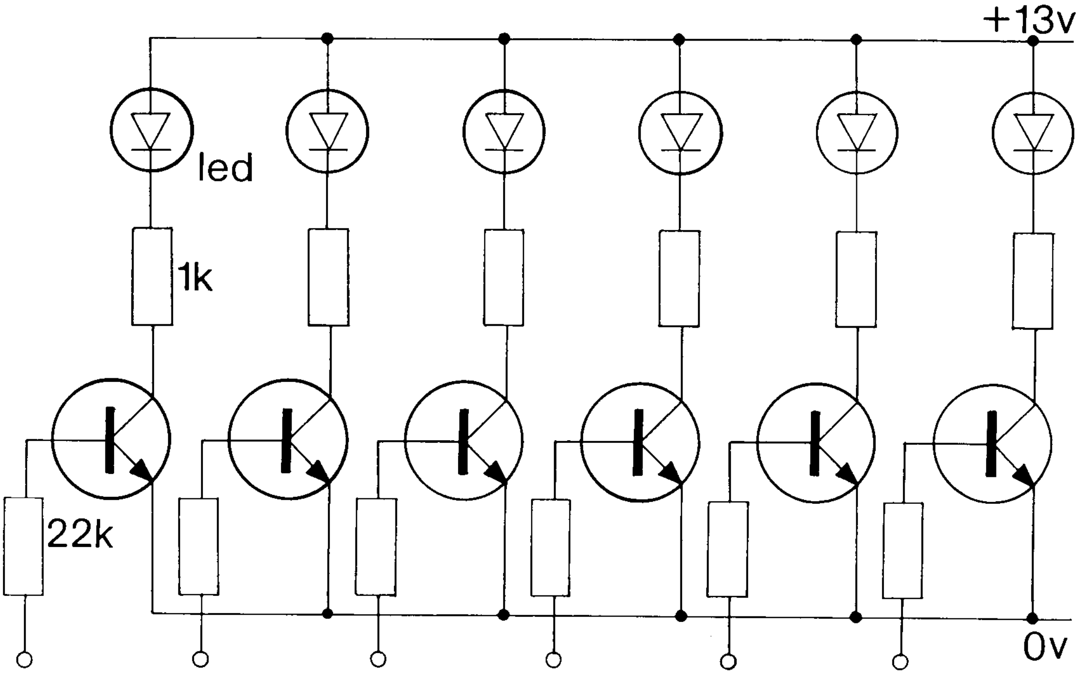
Resistor Input Divider Network For High Input Voltage



Resistor Pull-Up Network For Grounded Channel Switch

Switch Codes

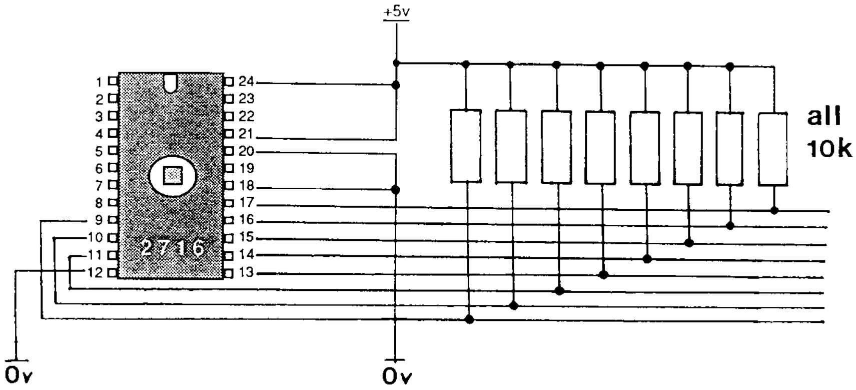
Before you begin a conversion you should ALWAYS use a voltmeter to determine the switch codes for every channel. Don't rely upon a book to tell you because you'll be awfully confused if the book is wrong or refers to a different model. Always check. If you start to design a lot of conversions it will help if you make up a row of LEDs with transistors on stripboard, which you can connect to the PLL input pins. Flicking the channel switch round causes different LEDs to light as the code changes and it is an easy matter to write them down. Keep records of everything you do because it's surprising how much is forgotten after a couple of months. The switch codes feeding the Eprom inputs specify the locations of the output codes. DO remember, however, that the complete location or "address" comprises the codes from the switch AND any voltage on the remaining Eprom inputs. These inputs can be used to switch from one bank of codes to another in order to change band. Unused inputs should be connected to 0 volts because, if left floating, they will pick up any stray RF which could be interpreted as a "1". I learnt this the hard way when a radio under test actually changed band on transmit. Great for duplex operation! You can create this effect deliberately if you need TX downshift for Repeater operation. Use the rig's own TX line to switch one of the Eprom inputs thereby swapping the code bank to one whose codes are programmed for 10 channels lower.



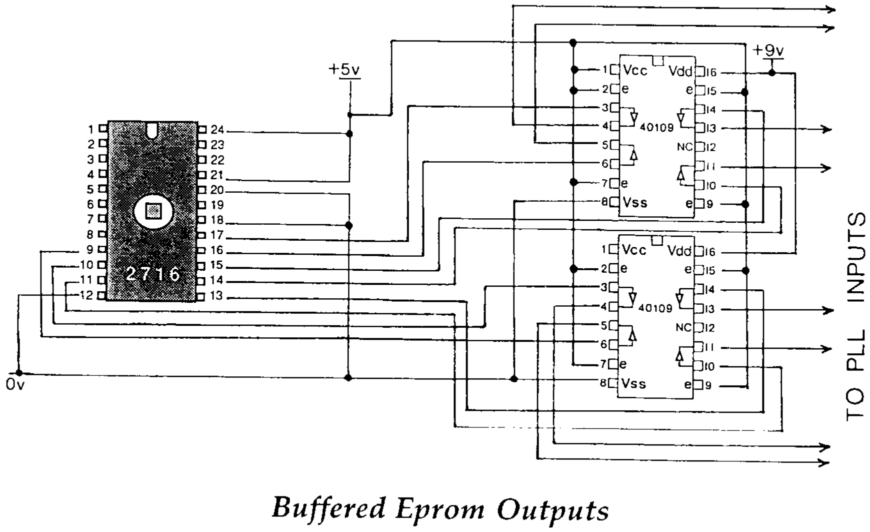
An LED Binary Code Display

Output Voltage

The output voltage from an Eprom can be a problem and will often be less than 3.5 volts when the Eprom outputs are connected to a PLL. With PLLs such as PLL02A there is seldom a problem since it usually runs on a supply of no more than 5.5 volts. Since 3.5 volts is more than half the supply voltage the PLL02A sees a "1". The situation with the MC145106P is different since this PLL usually runs with a supply of 9 volts and requires a 6 volt input to guarantee a "1". Although the Eprom manufacturers do not recommend it, the answer has been found, by experience, to be the addition of pull-up resistors from the Eprom output pins to the 9 volt supply. Resistors of about 10k in value should be used. Lower values can cause excess current to flow through the silicon substrate which will gradually kill the Eprom. This method has been in use in hundreds of units for several years with no problem. If you really want to play safe you could use a couple of 4050 non-inverting hex buffer ICs or 40109 level shifter ICs.



Eprom Output Pull-Up Resistors



Buffered Eprom Outputs

Practical Examples

The Cobra 148 on 10 meters

So much for the theory. Now suppose you want to carry out an actual conversion of a CB radio to the American 10 meter Novice band which covers 28.100 to 28.500 MHz. To make it a little more complicated let's choose the later model of the Export version Cobra 148 GTL DX which uses the PB-010 chassis. A single 15.000 MHz downmix crystal is employed and the three frequency bands are created by using a pair of MC14008 binary adder ICs between the channel switch and the MC145106P synthesizer.

A full description of the operation can be found in the Uniden "Export" Service Manual, which is published by CB City International. Briefly, the channel switch produces codes from 0F to 3B HEX (that is 15 to 59 decimal) which are fed to the binary adders. For the Low band, three diodes are switched to add the value 67 to each switch code making a series which now runs from 82 to 126. For Mid band the value 112 is added to give a series from 127 to 171 and, similarly, the High band codes run from 172 to 216.

High band covers 23.415 to 27.815 MHz and the obvious question is "why can't we add still higher values to achieve the codes for the Novice Band?". The answer is that you can, but with one drawback, which is that you will still have the "A" channel "hops" and the odd arrangement of channel 23 which is a damn nuisance when the other party says "QSY ten kiloHerz" and you promptly lose him! Notice, by the way, that the codes given above span 45 channels, not 40, because of the "hops".

It is very easy to work out the code required for 28.105. Since each increment of 1 in the code will move the rig up another 10 kHz just subtract 2810.5 from 2785.5 = 25 (working in units of 10kHz). So if we add 25 to the code for 27.855 MHz, which is 216, we get 241 which must produce a frequency of 28.105.

Note that all this was done without considering loop mixing frequencies, VCO frequencies or anything complicated beyond simple addition! Now we know that CH1 in the Novice Band requires a PLL code of 241. The switch code is still 15 for channel 1 so we must add 226 by switching in diodes.

We will still be left with the "hops" but let's look at this in a little more detail before moving on to the Eprom method. 226 in Binary is

256	128	64	32	16	8	4	2	1
0	1	1	1	0	0	0	1	0

So we need to switch in four diodes to add these bits to the appropriate inputs of the binary adders. The end result will be codes of 241 to 241+45 which is 286. In binary this represents

```
Ch40- 1   0   0   0   1   1   1   1   0  =  286 = 1 1 E (HEX)
Ch 1- 0   1   1   1   1   0   0   0   1  =  241 = 0 F 1 (HEX)
```

A further complication can be seen here because the Most Significant Bit changes from "0" to "1" somewhere in the range. This means that an inverter transistor must be used as described earlier. Helpfully, the PB-010 chassis already has a position for a transistor, which is TR28.

"QSY 10 kiloHertz"

Eprom Method

We have already done the hardest part, which was to work out the "N" codes. For the Novice band these are 241 to 280 giving 40 channels from 28.105 to 28.495. Only 6 Eprom inputs are connected to the channel switch to select 40 locations. The remaining Eprom inputs are not required and may be grounded to 0 volts. The Eprom can hold only 8 BITs so the 9th BIT must be generated from the 8th by using the transistor inverter circuit. The HEX code, therefore, omits the MSB and has just two digits, as you can see in the following table.

	code in Eprom			**Eprom location**	
Ch	Binary "N"Code	HEX	Dec	Switch code	HEX
40	1 0 0 0 1 1 0 0 0	18	280	1 1 1 0 1 1	3B
39	1 0 0 0 1 0 1 1 1	17	279	1 1 1 0 1 0	3A
38	1 0 0 0 1 0 1 1 0	16	278	1 1 1 0 0 1	39
37	1 0 0 0 1 0 1 0 1	15	277	1 1 1 0 0 0	38
36	1 0 0 0 1 0 1 0 0	14	276	1 1 0 1 1 1	37
35	1 0 0 0 1 0 0 1 1	13	275	1 1 0 1 1 0	36
34	1 0 0 0 1 0 0 1 0	12	274	1 1 0 1 0 1	35
33	1 0 0 0 1 0 0 0 1	11	273	1 1 0 1 0 0	34
32	1 0 0 0 1 0 0 0 0	10	272	1 1 0 0 1 1	33
31	1 0 0 0 0 1 1 1 1	0F	271	1 1 0 0 1 0	32
30	1 0 0 0 0 1 1 1 0	0E	270	1 1 0 0 0 1	31
29	1 0 0 0 0 1 1 0 1	0D	269	1 1 0 0 0 0	30
28	1 0 0 0 0 1 1 0 0	0C	268	1 0 1 1 1 1	2F
27	1 0 0 0 0 1 0 1 1	0B	267	1 0 1 1 1 0	2E
26	1 0 0 0 0 1 0 1 0	0A	266	1 0 1 1 0 1	2D
25	1 0 0 0 0 1 0 0 1	09	265	1 0 1 0 1 0	2A
24	1 0 0 0 0 1 0 0 0	08	264	1 0 1 1 0 0	2C
23	1 0 0 0 0 0 1 1 1	07	263	1 0 1 0 1 1	2B
22	1 0 0 0 0 0 1 1 0	06	262	1 0 1 0 0 1	29
21	1 0 0 0 0 0 1 0 1	05	261	1 0 1 0 0 0	28
20	1 0 0 0 0 0 1 0 0	04	260	1 0 0 1 1 1	27
19	1 0 0 0 0 0 0 1 1	03	259	1 0 0 1 0 1	25
18	1 0 0 0 0 0 0 1 0	02	258	1 0 0 1 0 0	24
17	1 0 0 0 0 0 0 0 1	01	257	1 0 0 0 1 1	23
16	1 0 0 0 0 0 0 0 0	00	256	1 0 0 0 1 0	22
15	0 1 1 1 1 1 1 1 1	FF	255	1 0 0 0 0 0	20
14	0 1 1 1 1 1 1 1 0	FE	254	0 1 1 1 1 1	1F
13	0 1 1 1 1 1 1 0 1	FD	253	0 1 1 1 1 0	1E
12	0 1 1 1 1 1 1 0 0	FC	252	0 1 1 1 0 1	1D
11	0 1 1 1 1 1 0 1 1	FB	251	0 1 1 0 1 1	1B
10	0 1 1 1 1 1 0 1 0	FA	250	0 1 1 0 1 0	1A
9	0 1 1 1 1 1 0 0 1	F9	249	0 1 1 0 0 1	19
8	0 1 1 1 1 1 0 0 0	F8	248	0 1 1 0 0 0	18
7	0 1 1 1 1 0 1 1 1	F7	247	0 1 0 1 1 0	16
6	0 1 1 1 1 0 1 1 0	F6	246	0 1 0 1 0 1	15
5	0 1 1 1 1 0 1 0 1	F5	245	0 1 0 1 0 0	14
4	0 1 1 1 1 0 1 0 0	F4	244	0 1 0 0 1 1	13
3	0 1 1 1 1 0 0 1 1	F3	243	0 1 0 0 0 1	11
2	0 1 1 1 1 0 0 1 0	F2	242	0 1 0 0 0 0	10
1	0 1 1 1 1 0 0 0 1	F1	241	0 0 1 1 1 1	0F

Novice Band codes for the Cobra 148 GTL DX

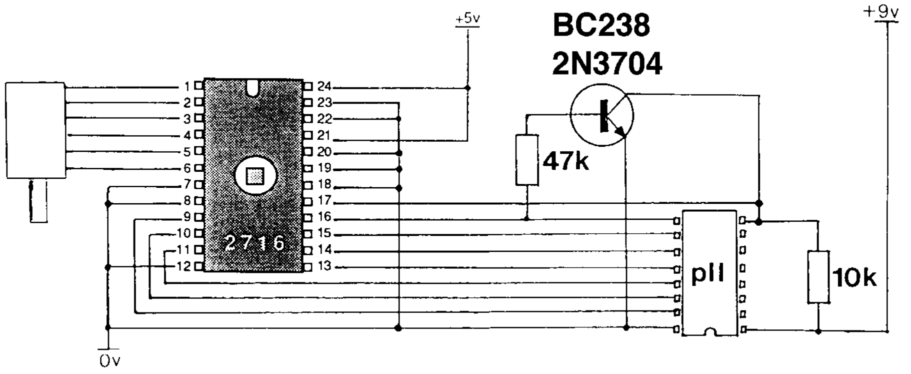
Note: the conversions will serve dozens of makes and models which use identical chassis and PLL circuits. Refer to the ‘CB PLL DATA BOOK’ or ‘THE “SCREWDRIVER EXPERT’S” GUIDE’ for listings of comparable rigs which can be converted.

Eprom Codes

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	F1
001	F2	F3	**	F4	F5	F6	F7	**	F8	F9	FA	FB	**	FC	FD	FE
002	FF	**	00	01	02	03	**	04	05	06	09	07	08	0A	0B	0C
003	0D	0E	0F	10	11	12	13	14	15	16	17	18	**	**	**	**
004	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**

This picture of the codes inside the Eprom should clarify the situation. We are working only in HEX now because that is what Eprom programming units use, so you MUST get used to it. The **location** is shown by the left hand column plus the top line. These numbers are SELECTED by the code from the channel switch itself. We can remove or disconnect the binary adder ICs which are no longer required. The content of each location is the code for that particular channel. Notice, once again, that some locations are not used by the switch. These have been marked "**" for clarity but would normally be FF. You might have noticed that, for this conversion, the code for Channel 15 also happens to be FF. This is pure coincidence but leads us to the next point.

Channel 16 starts at 00 again. Its "N" code ought to be 100 (HEX) or 100000000 (binary) but, since the Eprom can hold only an eight BIT code, we rely upon the inverting transistor to supply the "1" to the PLL. Whenever the eighth BIT from the Eprom is "1", therefore, the ninth BIT of the PLL is "0" from the transistor. The sketch shows how this is achieved.



Transistor Inverter Application

Channel Switch Codes

The codes from the channel switch don't matter, as mentioned previously, since they serve only to SELECT the LOCATION in Eprom for the actual "N" code we need. Consequently, we can often make the hardware connections simpler, although the encoding of the Eprom might become more complex.

The Export Cobra is a good example because the channel switch wires, as they come off the switch PCB, are in the order P5 P2 P4 P1 P3 P0 starting at the side near to the chassis. If we connect wires directly from the switch to the Eprom, therefore, they will cross over each other because the Eprom inputs are in the order P5 P4 P3 P2 P1 P0. The result is that we can't easily use ribbon cable to make the connection, or can we? The problem, of course, could be solved by using crossover links on the board upon which we mount the Eprom but why add to the hardware?

A better solution is to use the switch connections in precisely the order in which they emerge from the switch PCB. The only effect is to alter the LOCATIONs where we need to put the "N" codes in the Eprom. The Cobra codes start at 3B for channel 40. Taking the switch codes in the existing order changes the locations, as shown in the next table.

The effect, of mixing up the binary codes like this, is to change the LOCATIONS from the HEX number in the left hand column (which the switch normally supplies to the PLL inputs) to the HEX number in the right hand column. It is a little tedious to work out all 40 but, once done, does allow you to use ribbon cable instead of individual wires and no links are needed on the Eprom board.

Once again I must emphasise that we have simply rearranged the LOCATIONS for the "N" codes in the Eprom. With six connections there can only be 64 different locations and the channel switch selects 40 of these. Which 40, and in what order is unimportant to the actual operation of the unit. By mixing up the wires we only alter the selection of which 40 locations out of 64 the switch will select. Regardless of the order of the connections the result will always be 40 unique location codes out of a possible 64.

The arrangement of the 8 BITs of the "N" code in each location can also be swapped if it simplifies the wiring of the Eprom board. The method is, exactly as above, to write each "N" code in Binary; swap the BITs then convert into HEX to give the code which must be programmed into Eprom.

Ch	Old Loc.	5	4	3	2	1	0	5	2	4	1	3	0	New location
40	3B	1	1	1	0	1	1	1	0	1	1	1	1	2F
39	3A	1	1	1	0	1	0	1	0	1	1	1	0	2E
38	39	1	1	1	0	0	1	1	0	1	0	1	1	2B
37	38	1	1	1	0	0	0	1	0	1	0	1	0	2A
36	37	1	1	0	1	1	1	1	1	1	1	0	1	3D
35	36	1	1	0	1	1	0	1	1	1	1	0	0	3C
34	35	1	1	0	1	0	1	1	1	1	0	0	1	39
33	34	1	1	0	1	0	0	1	1	1	0	0	0	38
32	33	1	1	0	0	1	1	1	0	1	1	0	1	2D
31	32	1	1	0	0	1	0	1	0	1	1	0	0	2C
30	31	1	1	0	0	0	1	1	0	1	0	0	1	29
29	30	1	1	0	0	0	0	1	0	1	0	0	0	28
28	2F	1	0	1	1	1	1	1	1	0	1	1	1	37
27	2E	1	0	1	1	1	0	1	1	0	1	1	0	36
26	2D	1	0	1	1	0	1	1	1	0	0	1	1	33
25	2B	1	0	1	0	1	1	1	0	0	1	1	1	27
24	2A	1	0	1	0	1	0	1	0	0	1	1	0	26
23	2C	1	0	1	1	0	0	1	1	0	0	1	0	32
22	29	1	0	1	0	0	1	1	0	0	0	1	1	23
21	28	1	0	1	0	0	0	1	0	0	0	1	0	22
20	27	1	0	0	1	1	1	1	1	0	1	0	1	35
19	25	1	0	0	1	0	1	1	1	0	0	0	1	31
18	24	1	0	0	1	0	0	1	1	0	0	0	0	30
17	23	1	0	0	0	1	1	1	0	0	1	0	1	25
16	22	1	0	0	0	1	0	1	0	0	1	0	0	24
15	20	1	0	0	0	0	0	1	0	0	0	0	0	20
14	1F	0	1	1	1	1	1	0	1	1	1	1	1	1F
13	1E	0	1	1	1	1	0	0	1	1	1	1	0	1E
12	1D	0	1	1	1	0	1	0	1	1	0	1	1	1B
11	1B	0	1	1	0	1	1	0	0	1	1	1	1	0F
10	1A	0	1	1	0	1	0	0	0	1	1	1	0	0E
09	19	0	1	1	0	0	1	0	0	1	0	1	1	0B
08	18	0	1	1	0	0	0	0	0	1	0	1	0	0A
07	16	0	1	0	1	1	0	0	1	1	1	0	0	1C
06	15	0	1	0	1	0	1	0	1	1	0	0	1	19
05	14	0	1	0	1	0	0	0	1	1	0	0	0	08
04	13	0	1	0	0	1	1	0	0	1	1	0	1	0D
03	11	0	1	0	0	0	1	0	0	1	0	0	1	09
02	10	0	1	0	0	0	0	0	0	1	0	0	0	18
01	0F	0	0	1	1	1	1	0	1	0	1	1	1	17

Swapping the binary address codes to simplify the wiring

The Eprom used on the PCB which is shown on page 29 uses the “old” locations because it was designed with wire links. This table serves merely to show how binary codes *might* be swapped to avoid the use of links.

The MB8719 used in Uniden Chassis

This PLL is unusual in having only seven programming pins but an internal inverter which makes it work as if it had eight. The seventh pin controls the MSB which is always "1" (~64) or "0" (.128). Unfortunately, this system means that the range of codes is limited, going from 64 minimum to 128+63 = 191 maximum: a range of 127 channels or just three banks of 40, in practice.

The early export Cobra 148GTL DX uses three crystals with this PLL. The channel switch is the standard Uniden switch with codes from 15 to 59 but, because the MSB pin on the MB8719 is wired high, you must add another 64. With HI band crystal selected (15.93MHz) the channel 40 code is 59+64=123 and produces a frequency of 27.855MHz (which is not legal, by the way!). The highest "N" code is 191 and you can add 191-123=68 channels at 10kHz to give a maximum frequency of 28.535MHz with this crystal. The "Novice" band of 28.10 to 28.50 is, therefore, only just possible. If you need higher frequencies in the 10 meter band you must change the crystal.

The top channel of 28.495MHz requires an "N" code of 187 at channel 40 and channel 1, 28.105MHz will have a code of 148.

	128	64	32	16	8	4	2	1		
The codes range from	1	0	1	1	1	0	1	1	=	BB HEX (ch40)
to--	1	0	0	1	0	1	0	0	=	94 HEX (ch 1)

Remember that the eighth bit is generated automatically by the inverter built into the MB8719 itself. You need, therefore, only seven Eprom outputs connected to the PLL and the eighth bit can be "0" since it is not connected. In that case, BB HEX becomes 3B HEX and 94 HEX becomes 14 HEX. Since the MSB is always "0" you can connect it to ground (cut the copper trace) and use only six Eprom outputs. Getting easier, isn't it!

As in the MC145106P version, the switch codes 15 to 59 (with hops) select the locations where the new "N" codes are to be found in the Eprom. The channel switch in the Cobra is always wired to ground and external 47k pullup resistors are used to provide a positive voltage of 9v. These should be removed from the main Cobra board if Eprom inputs are to be connected directly and pull-up resistors connected to +5 volts used instead. This is accomrnodated on the Cobra Eprom board described later.

ch	<...Eprom Contents...>			Eprom location
40	1 0 1 1	1 0 1 1	3B	3B
39	1 0 1 1	1 0 1 0	3A	3A
38	1 0 1 1	1 0 0 1	39	39
37	1 0 1 1	1 0 0 0	38	38
36	1 0 1 1	0 1 1 1	37	37
35	1 0 1 1	0 1 1 0	36	36
34	1 0 1 1	0 1 0 1	35	35
33	1 0 1 1	0 1 0 0	34	34
32	1 0 1 1	0 0 1 1	33	33
31	1 0 1 1	0 0 1 0	32	32
30	1 0 1 1	0 0 0 1	31	31
29	1 0 1 1	0 0 0 0	30	30
28	1 0 1 0	1 1 1 1	2F	2F
27	1 0 1 0	1 1 1 0	2E	2E
26	1 0 1 0	1 1 0 1	2D	2D
25	1 0 1 0	1 1 0 0	2C	2A
24	1 0 1 0	1 0 1 1	2B	2C
23	1 0 1 0	1 0 1 0	2A	2B
22	1 0 1 0	1 0 0 1	29	29
21	1 0 1 0	1 0 0 0	28	28
20	1 0 1 0	0 1 1 1	27	27
19	1 0 1 0	0 1 1 0	26	25
18	1 0 1 0	0 1 0 1	25	24
17	1 0 1 0	0 1 0 0	24	23
16	1 0 1 0	0 0 1 1	23	22
15	1 0 1 0	0 0 1 0	22	20
14	1 0 1 0	0 0 0 1	21	1F
13	1 0 1 0	0 0 0 0	20	1E
12	1 0 0 1	1 1 1 1	1F	1D
11	1 0 0 1	1 1 1 0	1E	1B
10	1 0 0 1	1 1 0 1	1D	1A
9	1 0 0 1	1 1 0 0	1C	19
8	1 0 0 1	1 0 1 1	1B	18
7	1 0 0 1	1 0 1 0	1A	16
6	1 0 0 1	1 0 0 1	19	15
5	1 0 0 1	1 0 0 0	18	14
4	1 0 0 1	0 1 1 1	17	13
3	1 0 0 1	0 1 1 0	16	11
2	1 0 0 1	0 1 0 1	15	10
1	1 0 0 1	0 1 0 0	14	OF

You will notice that the "N" codes required are very similar to the switch codes. In fact you can simply cut the trace to the MB8719 pin 10 and ground-it and be very nearly there using just the channel switch outputs. This is the simplest 10 meter mod!

Rigs which use an 11 MHz crystal can not be modified so easily as the range of MB8719 divide ratios is not sufficiently large. It is necessary to change the crystal. A value of 11.755MHz produces 28.105MHz when the "N" code is 64 and leaves the possibility of producing another two higher bands with the same crystal.

ch	<....Eprom Contents....>			Eprom location
40	0 1 1 0	0 1 1 1	67	3B
39	0 1 1 0	0 1 1 0	66	3A
38	0 1 1 0	0 1 0 1	65	39
37	0 1 1 0	0 1 0 0	64	38
36	0 1 1 0	0 0 1 1	63	37
35	0 1 1 0	0 0 1 0	62	36
34	0 1 1 0	0 0 0 1	61	35
33	0 1 1 0	0 0 0 0	60	34
32	0 1 0 1	1 1 1 1	5F	33
31	0 1 0 1	1 1 1 0	5E	32
30	0 1 0 1	1 1 0 1	5D	31
29	0 1 0 1	1 1 0 0	5C	30
28	0 1 0 1	1 0 1 1	5B	2F
27	0 1 0 1	1 0 1 0	5A	2E
26	0 1 0 1	1 0 0 1	59	2D
25	0 1 0 1	1 0 0 0	58	2A
24	0 1 0 1	0 1 1 1	57	2C
23	0 1 0 1	0 1 1 0	56	2B
22	0 1 0 1	0 1 0 1	55	29
21	0 1 0 1	0 1 0 0	54	28
20	0 1 0 1	0 0 1 1	53	27
19	0 1 0 1	0 0 1 0	52	25
18	0 1 0 1	0 0 0 1	51	24
17	0 1 0 1	0 0 0 0	50	23
16	0 1 0 0	1 1 1 1	4F	22
15	0 1 0 0	1 1 1 0	4E	20
14	0 1 0 0	1 1 0 1	4D	1F
13	0 1 0 0	1 1 0 0	9C	1E
12	0 1 0 0	1 0 1 1	4B	1D
11	0 1 0 0	1 0 1 0	4A	1B
10	0 1 0 0	1 0 0 1	49	1A
9	0 1 0 0	1 0 0 0	48	19
8	0 1 0 0	0 1 1 1	47	18
7	0 1 0 0	0 1 1 0	46	16
6	0 1 0 0	0 1 0 1	45	15
5	0 1 0 0	0 1 0 0	44	14
4	0 1 0 0	0 0 1 1	43	13
3	0 1 0 0	0 0 1 0	42	11
2	0 1 0 0	0 0 0 1	41	10
1	0 1 0 0	0 0 0 0	40	OF

If, however, a conversion board can be fitted which uses the MC145106P IC then the range of "N" codes is greatly extended and permits the standard crystal of either 11.1125 or 11.3258 MHz to be retained without the need to buy a specially made crystal. The conversion board uses an Eprom to translate the switch codes into "N" codes for the new PLL IC and the full range of frequencies from 28.105 to 29.695 MHz can be achieved, together with 100 kHz repeater downshift on transmit. A complete conversion board is available but the following information is given to allow you to design your own, should you so wish.

For the 11.1125 MHz crystal the codes required for the "Novice" band start at HEX 101 for 28.105 MHz (ch. 1) and go up in steps of 1 to HEX 128 for 28.495 MHz (ch. 40). The "N" codes for the rest of the 10 meter band rise from HEX 129 to HEX lA0 for 29.695 MHz. Notice that a transistor inverter is needed for the MSB.

For the 11.3258 MHz crystal the codes required for the "Novice" band start at HEX C1 for 28.105 MHz (ch. 1) and go up in steps of 1 to HEX E8 for 28.495 MHz (ch. 40). The "N" codes for the rest of the 10 meter band rise from HEX E9 to HEX 160 for 29.695 MHz.

The six outputs from the channel switch produce HEX OF (ch. 1) to HEX 3B (ch. 40) and provide the location codes for the Eprom inputs, together with connections "T", "X" and "Y" on the PCB.

The circuit diagram shows the interconnections and, since they have been optimized for simplicity of the PCB layout, the outputs from the Eprom are not connected to the inputs of the MC145106P in straight order of numbers. The actual Eprom contents must, therefore, have the individual BITs rearranged as explained on a previous page so that the PLL sees the correct codes.

Note that link LK1 must NOT be fitted if an 11.3258 MHz crystal is used. Also, because of basic differences between the MC145106P and the MB8719, which this conversion replaces, it is necessary to remove the 1.5k resistor which is soldered near to the MB8719 position in the CB radio.

The MB8719 is removed and the 16 pin ribbon connector soldered in its place but NOT in the two holes used by MB8719 pins 9 and 10.

All the necessary information is given here, together with the complete Eprom codes, but this is a most ADVANCED conversion so if you do not fully understand the way in which the conversion works you should not attempt the conversion.

LOC. EPROM CONTENTS FOR 11.1125 CRYSTAL

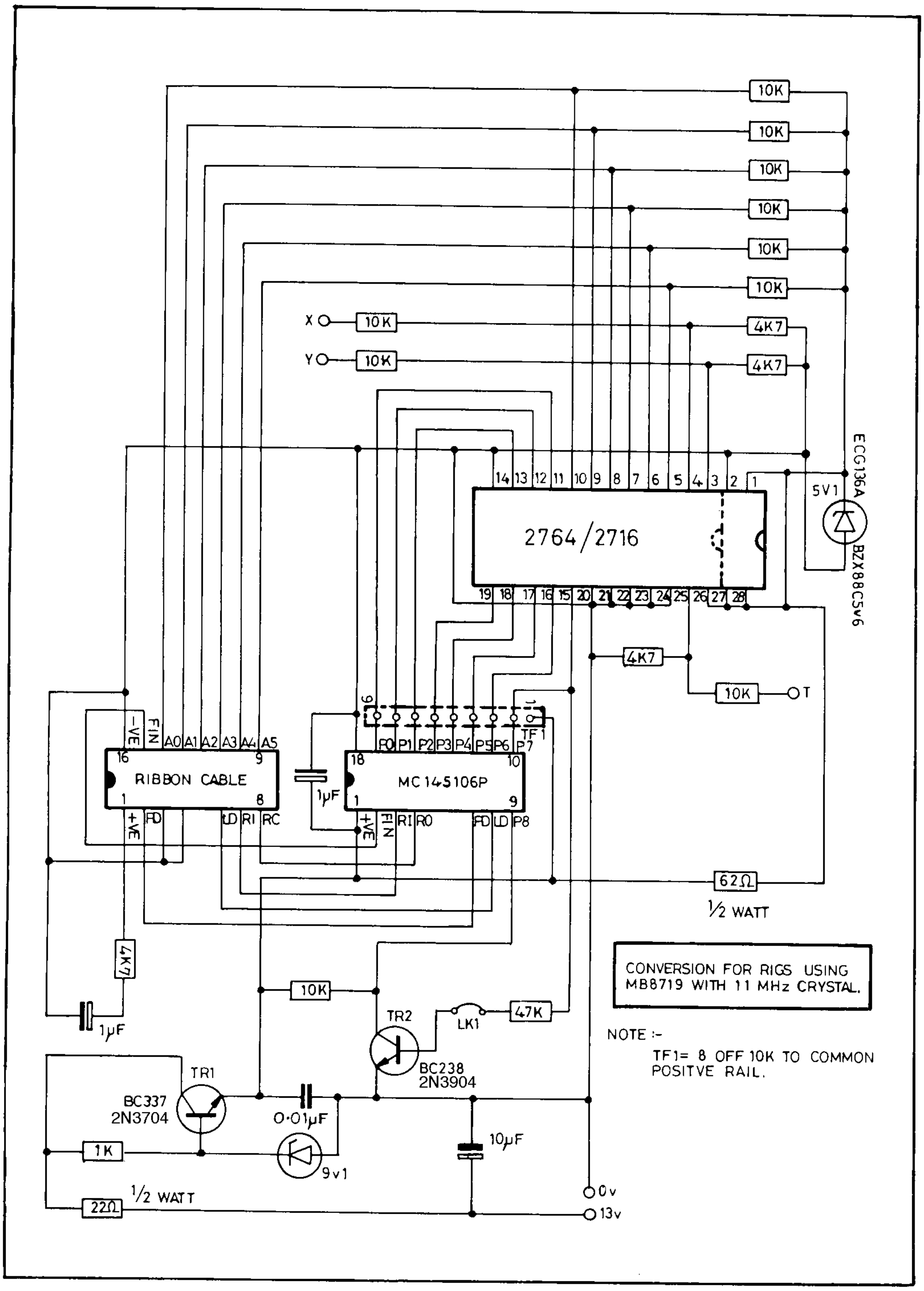
```
0000 .. .. .. .. .. .. .. ..
0008 .. .. .. .. .. .. .. 01
0010 02 03 .. 04 05 06 07 ..
0018 80 81 82 83 .. 89 85 86
0020 87 .. 40 41 42 43 .. 44            28.105-28.495MHz
0028 45 46 C0 C1 47 C2 C3 C4
0030 C5 C6 C7 20 21 22 23 24
0038 25 26 27 A0 .. .. .. ..
0040 .. .. .. .. .. .. .. ..
0048 .. .. .. .. .. .. .. A1
0050 A2 A3 .. A4 A5 A6 A7 ..            X = 1
0058 60 61 62 63 .. 64 65 66
0060 67 .. E0 E1 E2 E3 .. E4            28.505 - 28.895 MHz
0068 E5 E6 10 11 E7 12 13 14
0070 15 16 17 90 91 92 93 94
0078 95 96 97 50 .. .. .. ..
0080 .. .. .. .. .. .. .. ..
0088 .. .. .. .. .. .. .. 51
0090 52 53 .. 54 55 56 57 ..            Y = 1
0098 D0 D1 D2 D3 .. D4 D5 D6
00A0 D7 .. 30 31 32 33 .. 34            28.905 - 29.295 MHz
00A8 35 36 B0 B1 37 B2 B3 B4
00B0 B5 B6 B7 70 71 72 73 74
00B8 75 76 77 F0 .. .. .. ..
OOCO .. .. .. .. .. .. .. ..
00C8 .. .. .. .. .. .. .. ..
00D0 .. .. .. .. .. .. .. ..
00D8 .. .. .. .. .. .. .. ..
00E0 .. .. .. .. .. .. .. ..
00E8 .. .. .. .. .. .. .. ..
00F0 .. .. .. .. .. .. .. ..
00F8 .. .. .. .. .. .. .. ..
0100 .. .. .. .. .. .. .. ..
0108 .. .. .. .. .. .. .. F1
0110 F2 F3 .. F4 F5 F6 F7 ..            T = 1
0118 08 09 OA 0B .. 0C OD 0E
0120 OF .. 88 89 8A 8B .. 8C            29.305-29.695MHz
0128 8D 8E 48 49 8F 4A 4B 4C
0130 4D 4E 4F C8 C9 CA CB CC
0138 CD CE CF 28 .. .. .. ..
0140 .. .. .. .. .. .. .. ..
0148 .. .. .. .. .. .. .. ..
```

.. = Don't care

```
LOC. EPROM CONTENTS FOR 11.3258 CRYSTAL

0000 .. .. .. .. .. .. .. ..
0008 .. .. .. .. .. .. .. 19
0010 1A 1B .. 1C 1D 1E 1F ..
0018 98 99 9A 9B .. 9C 9D 9E
0020 9F .. 58 59 5A 5B .. 5C            28.105 - 28.495 MHz
0028 5D 5E D8 D9 5F DA DB DC
0030 DD DE DF 38 39 3A 3B 3C
0038 3D 3E 3F B8 .. .. .. ..~
0040 .. .. .. .. .. .. .. ..
0048 .. .. .. .. .. .. .. ..
0050 BA BB .. BC BD BE BF ..            X =
0058 78 79 7A 7B .. 7C 7D 7E
0060 7F .. F8 F9 FA FB .. FC            28.505 - 28.895 MHz
0068 FD FE 00 01 .. 02 03 04
0D70 05 06 07 80 81 82 83 84
0078 85 86 87 40 .. .. .. ..
0080 .. .. .. .. .. .. .. ..
0088 .. .. .. .. .. .. .. 41
0090 42 43 .. 44 45 46 47 ..            Y =
0098 C0 C1 C2 C3 .. C4 C5 C6
00A0 C7 .. 20 21 22 23 .. 24            28.905 - 29.295 MHz
00A8 25 26 A0 A1 27 A2 A3 A4
00B0 A5 A6 A7 60 61 62 63 64
00B8 65 66 67 E0 .. .. .. ..
00C0 .. .. .. .. .. .. .. ..
00C8 .. .. .. .. .. .. .. ..
00D0 .. .. .. .. .. .. .. ..
00D8 .. .. .. .. .. .. .. ..
00E0 .. .. .. .. .. .. .. ..
00E8 .. .. .. .. .. .. .. ..
00F0 .. .. .. .. .. .. .. ..
00F8 .. .. .. .. .. .. .. ..
0100 .. .. .. .. .. .. .. ..
0108 .. .. .. .. .. .. .. E1
0110 E2 E3 .. E4 E5 E6 E7 ..            T = 1
0118 10 11 12 13 .. 14 15 16
0120 17 .. 90 91 92 93 .. 94            29.305 - 29.695 MHz
0128 95 96 50 51 97 52 53 54
0130 55 56 57 D0 D1 D2 D3 D4
0138 D5 D6 D7 30 .. .. .. ..
0140 .. .. .. .. .. .. .. ..
0148 .. .. .. .. .. .. .. ..

 .. = Don't care
```



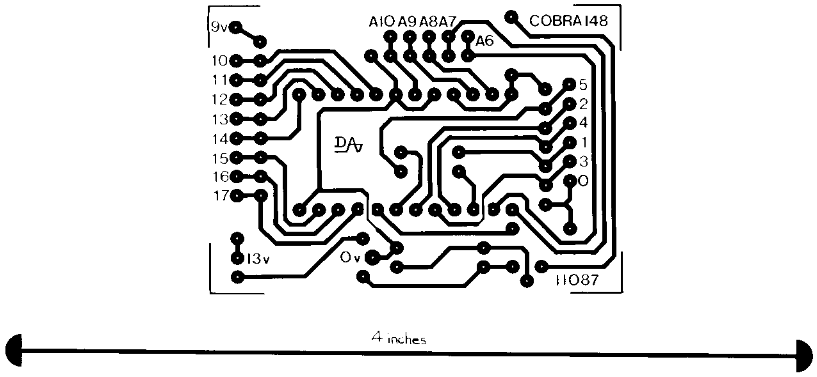
CONVERSION FOR RIGS USING MB8719 WITH 11 MHz CRYSTAL.

NOTE :-

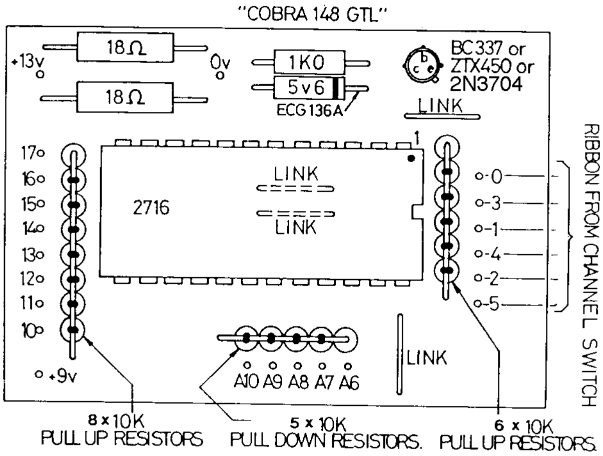
TF1= 8 OFF 10K TO COMMON POSITVE RAIL.

Another Eprom Board

The previous design was specifically for certain rigs but this Eprom board for the Cobra may be used with most versions. Six pull-up resistors are connected to the channel switch inputs O to 5, which are arranged exactly in the right order for the ribbon cable from the channel switch itself. Eprom inputs A6 to A10 are available to select several more bands, if required. However, no more than 5 volts must be applied to these inputs unless series resistors (e.g. 10k for +9v) are connected. The eight pull-up resistors on the Eprom outputs are required for the MC145106P but not for the MB8719 which has internal pull-ups. If you fit them, connect the 9 volt input to pin 18 of the MC145106P. Note that this board uses the STANDARD Eprom locations OF to 3B.



P.C. Foil Pattern



Component Layout

Example

The TC9109 and MB8733 PLL ICs are both similar in that they use eight of the connections between the channel switch and the LED channel number display as inputs to determine the "N" code. In effect, there is a built-in "Prom" which translates the LED code into the "N" code and the PLL can not, therefore, be fooled. They produce the 40 FCC approved channels, but no others. Unlike most PLLs, these produce the transmit frequency by doubling the VCO output, instead of by mixing. There is, however, a British PLL which, although now obsolescent, serves to illustrate another method of using an Eprom if the PLL is changed. This British PLL, the LC7137, also doubles the VCO output to produce the transmit frequency, but its input codes are Binary Coded Decimal (BCD) with the exception of channel 40 which is 000000 instead of 1000000.

Since these PLLs operate in basically the same manner we can substitute, for one of the FCC ICs, a board containing an LC7137 plus an Eprom to convert the eight LED display code inputs to six BCD inputs, instead. The LED codes then become the LOCATION numbers which tell the Eprom where to find the BCD codes for the LC7137. This example is of interest, firstly because the LED codes seem to be in no logical order but, nevertheless, select 40 unique locations for the BCD codes. Secondly, because the British channels produced are in ascending order, without any "hops", therefore giving a better basis to convert to the 10 meter band, perhaps by mixing methods or by digital subtraction (which has nothing to do with getting your finger out!).

Note that the input for pin 17 is high for every channel except channel 40. It seems a singular waste of a wire to use 8 connections instead of 7, just for the sake of one zero. Close inspection of the codes reveals that we can eliminate this first connection if we use the zero on channel 40 to make the pin 15 voltage go high. This is fairly simple to achieve with a single transistor switch and the result for channel 40 is still a unique code of 1 1 0 0 1 0 0 or 64 instead of 44 HEX. This code does not occur for any other channel, so can be used to identify the Eprom location of channel 40. The first input can be eliminated and all the other location codes will have to be modified accordingly. The hardware advantage of this change is questionable but the example shows, again, that the actual switch codes are not important, provided that each can be used to generate a unique location code. (When BCD is translated into HEX, the numbers A to F do not appear since only the binary codes for 1 to 9 are used.)

TC9109 input codes.

	17	16	15	14	13	12	11	10	- 9109 pins
CH	2f	2a	2c	1d	1e	1g	1a	1f	- LED Segment
40	0	1	0	0	0	1	0	0	
39	1	0	0	1	1	0	0	0	
38	1	0	0	0	0	0	0	0	
37	1	0	0	1	1	1	0	1	
36	1	0	0	0	0	0	1	0	
35	1	0	0	0	1	0	0	0	
34	1	0	0	1	1	0	1	0	
33	1	0	0	0	1	0	0	1	
32	1	0	0	0	0	0	0	1	
31	1	0	0	1	1	1	1	1	
30	1	0	0	0	0	1	0	0	
29	1	0	1	1	1	0	0	0	
28	1	0	1	0	0	0	0	0	
27	1	0	1	1	1	1	0	1	
26	1	0	1	0	0	0	1	0	
25	1	0	1	0	1	0	0	0	
24	1	0	1	1	1	0	1	0	
23	1	0	1	0	1	0	0	1	
22	1	0	1	0	0	0	0	1	
21	1	0	1	1	1	1	1	1	
20	1	0	1	0	0	1	0	0	
19	1	1	0	1	1	0	0	0	
18	1	1	0	0	0	0	0	0	
17	1	1	0	1	1	1	0	1	
16	1	1	0	0	0	0	1	0	
15	1	1	0	0	1	0	0	0	
14	1	1	0	1	1	0	1	0	
13	1	1	0	0	1	0	0	1	
12	1	1	0	0	0	0	0	1	
11	1	1	0	1	1	1	1	1	
10	1	1	0	0	0	1	0	0	
9	1	1	1	1	1	0	0	0	
8	1	1	1	0	0	0	0	0	
7	1	1	1	1	1	1	0	1	
6	1	1	1	0	0	0	1	0	
5	1	1	1	0	1	0	0	0	
4	1	1	1	1	1	0	1	0	
3	1	1	1	0	1	0	0	1	
2	1	1	1	0	0	0	0	1	
1	1	1	1	1	1	1	1	1	

LC7137 input codes.

	6	5	4	3	2	1	-7137 pins
location	BCD						HEX
44	0	0	0	0	0	0	00
98	1	1	1	0	0	1	39
80	1	1	1	0	0	0	38
9D	1	1	0	1	1	1	37
82	1	1	0	1	1	0	36
88	1	1	0	1	0	1	35
9A	1	1	0	1	0	0	34
89	1	1	0	0	1	1	33
81	1	1	0	0	1	0	32
9F	1	1	0	0	0	1	31
84	1	1	0	0	0	0	30
B8	1	0	1	0	0	1	29
A0	1	0	1	0	0	0	28
BD	1	0	0	1	1	1	27
A2	1	0	0	1	1	0	26
A8	1	0	0	1	0	1	25
DA	1	0	0	1	0	0	24
A9	1	0	0	0	1	1	23
A1	1	0	0	0	1	0	22
BF	1	0	0	0	0	1	21
A4	1	0	0	0	0	0	20
D8	0	1	1	0	0	1	19
C0	0	1	1	0	0	0	18
DD	0	1	0	1	1	1	17
C2	0	1	0	1	1	0	16
C8	0	1	0	1	0	1	15
DA	0	1	0	1	0	0	14
C9	0	1	0	0	1	1	13
C1	0	1	0	0	1	0	12
DF	0	1	0	0	0	1	11
C4	0	1	0	0	0	0	10
F8	0	0	1	0	0	1	09
E0	0	0	1	0	0	0	08
FD	0	0	0	1	1	1	07
E2	0	0	0	1	1	0	06
E8	0	0	0	1	0	1	05
FA	0	0	0	1	0	0	04
E9	0	0	0	0	1	1	03
E1	0	0	0	0	1	0	02
FF	0	0	0	0	0	1	01

µPD858 – Example

There is a range of AM/SSB rigs which includes the Cobra 138/139 XLR, Realistic TTRC457/458/449 and "President Adams" that use the uPD858 PLL and can readily be converted to the American "Novice" Band. This particular PLL uses BCD coding and the standard "N" code values are 91 to 135. Although ten pins exist on this PLL, only eight are fed by the channel switch. The codes on pins 22 and 19 do not change and are permanently grounded. We need use only seven (*) of the eight, however, to get 40 unique location codes. The resultant codes are thus:-

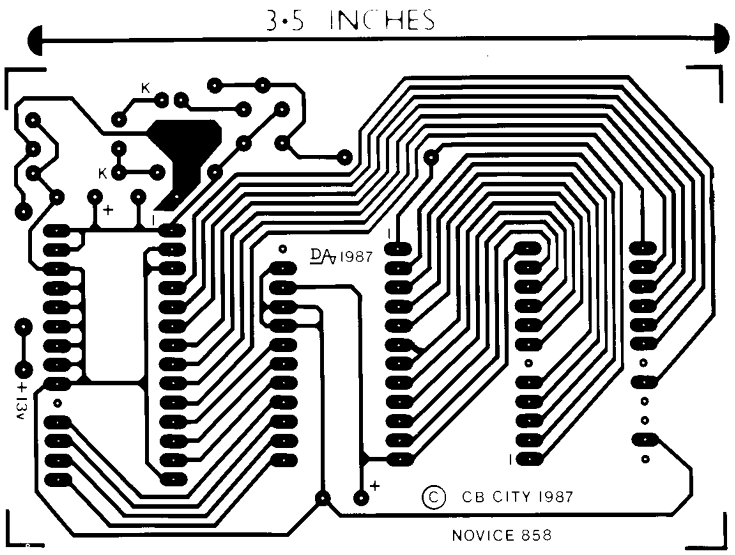
pins	22	-	-	-	-	-	-	-	-	13	"N"	*switch
			*		*	*	*	*	*	*		
40	0	1	0	0	1	1	0	1	0	1	135	35
39	0	1	0	0	1	1	0	1	0	0	134	34
38	0	1	0	0	1	1	0	0	1	1	133	33
37	0	1	0	0	1	1	0	0	1	0	132	32
36	0	1	0	0	1	1	0	0	0	1	131	31
35	0	1	0	0	1	1	0	0	0	0	130	30
34	0	1	0	0	1	0	1	0	0	1	129	29
33	0	1	0	0	1	0	1	0	0	0	128	28
32	0	1	0	0	1	0	0	1	1	1	127	27
31	0	1	0	0	1	0	0	1	1	0	126	26
31	0	1	0	0	1	0	0	1	1	0	126	26
30	0	1	0	0	1	0	0	1	0	1	125	25
29	0	1	0	0	1	0	0	1	0	0	124	24
28	0	1	0	0	1	0	0	0	1	1	123	23
27	0	1	0	0	1	0	0	0	1	0	122	22
26	0	1	0	0	1	0	0	0	0	1	121	21
25	0	1	0	0	0	1	1	0	0	0	118	18
24	0	1	0	0	1	0	0	0	0	0	120	20
23	0	1	0	0	0	1	1	0	0	1	119	19
22	0	1	0	0	0	1	0	1	1	1	117	17
21	0	1	0	0	0	1	0	1	1	0	116	16
......	...	...	...	...	...	...	...	...	...	...	....	......
......	...	...	...	...	...	...	...	...	...	...	....	......
09	0	1	0	0	0	0	0	0	0	1	101	01
08	0	1	0	0	0	0	0	0	0	0	100	00
07	0	0	1	0	0	1	1	0	0	0	98	58
06	0	0	1	0	0	1	0	1	1	1	97	57
05	0	0	1	0	0	1	0	1	1	0	96	56
04	0	0	1	0	0	1	0	1	0	1	95	55
03	0	0	1	0	0	1	0	0	1	1	93	53
02	0	0	1	0	0	1	0	0	1	0	92	52
01	0	0	1	0	0	1	0	0	0	1	91	51

The "N" codes to shift from 29.965 MHz up to 28.105 MHz are easily worked out as in the following table. The Eprom codes need consist only of the first seven BITs (+) since the three Most Significant Bits do not need to change and can be wired permanently high or low, as indicated. The eighth output from the Eprom may be left unconnected since it is not required.

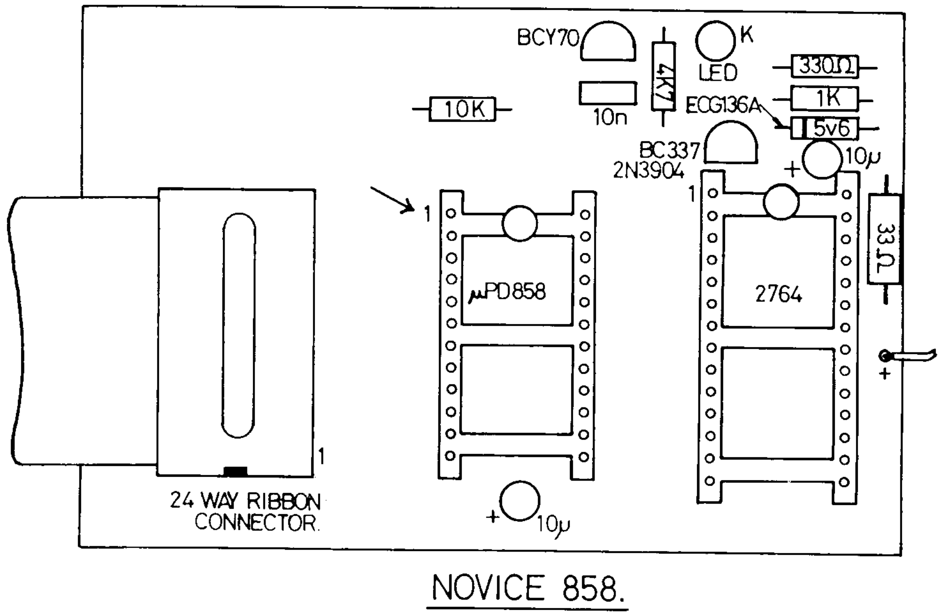
pins-	22	-	-	-	-	-	-	-	-	13	"N"	+Eprom code	*location
ch.				+	+	+	+	+	+	+			
40	1	0	0	1	0	0	0	1	0	0	244	44	35
39	1	0	0	1	0	0	0	0	1	1	243	43	34
38	1	0	0	1	0	0	0	0	1	0	242	42	33
37	1	0	0	1	0	0	0	0	0	1	241	41	32
36	1	0	0	1	0	0	0	0	0	0	240	40	31
35	1	0	0	0	1	1	1	0	0	1	239	39	30
34	1	0	0	0	1	1	1	0	0	0	238	38	29
33	1	0	0	0	1	1	0	1	1	1	237	37	28
32	1	0	0	0	1	1	0	1	1	0	236	36	27
31	1	0	0	0	1	1	0	1	0	1	235	35	26
30	1	0	0	0	1	1	0	1	0	0	234	34	25
29	1	0	0	0	1	1	0	0	1	1	233	33	24
28	1	0	0	0	1	1	0	0	1	0	232	32	23
27	1	0	0	0	1	1	0	0	0	1	231	31	22
26	1	0	0	0	1	1	0	0	0	0	230	30	19
25	1	0	0	0	1	0	1	0	0	1	229	29	21
24	1	0	0	0	1	0	1	0	0	0	228	28	20
23	1	0	0	0	1	0	0	1	1	1	227	27	18
22	1	0	0	0	1	0	0	1	1	0	226	26	17
21	1	0	0	0	1	0	0	1	0	1	225	25	16
20	1	0	0	0	1	0	0	1	0	0	224	24	15
19	1	0	0	0	1	0	0	0	1	1	223	23	13
18	1	0	0	0	1	0	0	0	1	0	222	22	12
17	1	0	0	0	1	0	0	0	0	1	221	21	11
16	1	0	0	0	1	0	0	0	0	0	220	20	10
15	1	0	0	0	0	1	1	0	0	1	219	19	08
14	1	0	0	0	0	1	1	0	0	0	218	18	07
13	1	0	0	0	0	1	0	1	1	1	217	17	06
12	1	0	0	0	0	1	0	1	1	0	216	16	05
11	1	0	0	0	0	1	0	1	0	1	215	15	03
10	1	0	0	0	0	1	0	1	0	0	214	14	02
09	1	0	0	0	0	1	0	0	1	1	213	13	01
08	1	0	0	0	0	1	0	0	1	0	212	12	00
07	1	0	0	0	0	1	0	0	0	1	211	11	58
06	1	0	0	0	0	1	0	0	0	0	210	10	57
05	1	0	0	0	0	0	1	0	0	1	209	09	56
04	1	0	0	0	0	0	1	0	0	0	208	08	54
03	1	0	0	0	0	0	0	1	1	1	207	07	53
02	1	0	0	0	0	0	0	1	1	0	206	06	52
01	1	0	0	0	0	0	0	1	0	1	205	05	51

Table of channel switch codes

+ These binary codes are rearranged to match the order in which the Eprom outputs are connected to the µPD858 on the PCB.

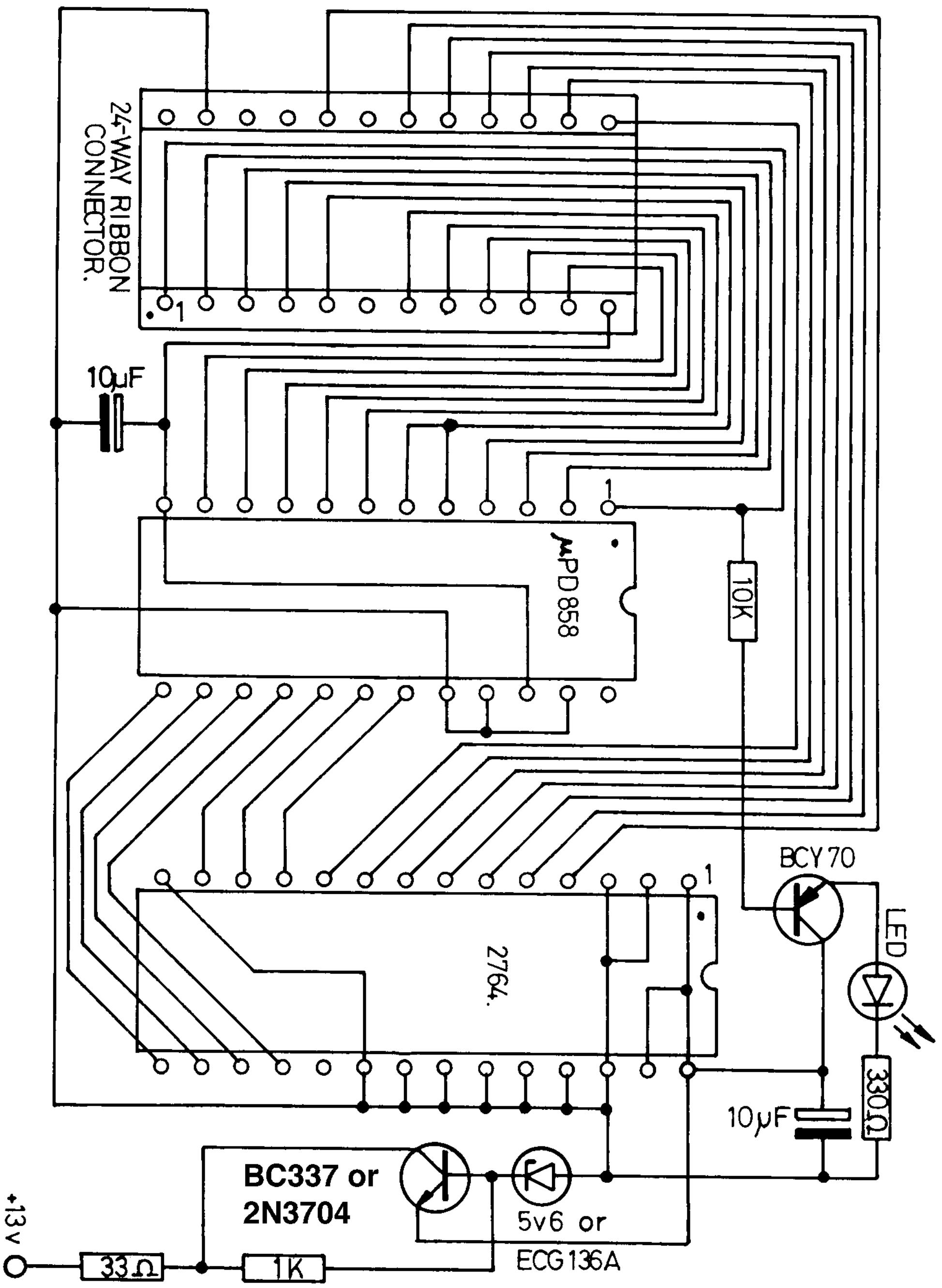


P.C. Foil Pattern



Component Layout

Novice 858 10m. conversion



µPD2824 – Example

The uPD2824 is a popular PLL in America but is internally programmed for only forty channels. The internal "N" codes are 91 to 135 with the usual "hops". Notice that these are exactly the same "N" codes which were used in the uPD858 rigs but they are accessed, here, by a channel switch which provides a BCD output of 00 to 39. This is one of the "impossible" chips and must be replaced if we are to use the rig on the 10 meter band. The obvious choice is the MC145106P because, not only does it give us the required "N" codes, but also the essential 5.12 MHz output pin, without which the rig will not work. Unfortunately it does not have the internal amplifier for PD filtering, consequently we must add a VMOS Field Effect Transistor to complete the conversion.

The MC145106P can be used to provide the standard FCC channels and the table shows the codes in their entirety so that you can compare them with the new codes required for the "Novice" band. The rig MUST NOT be made capable of working on both bands, otherwise you are likely to be in breach of the law in most countries.

A provisional PCB layout is given but it must be pointed out that this has not been tested.

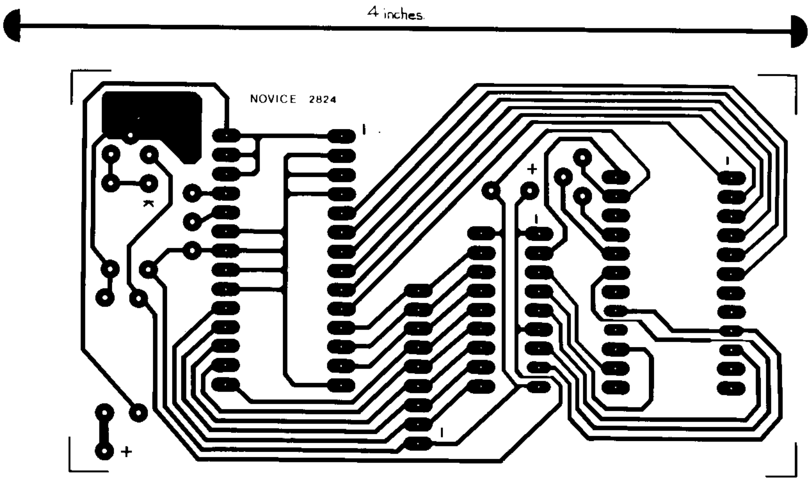
You can see from the column marked "Eprom Location" that the Address code is taken directly from the channel switch which, as mentioned, gives Binary Coded Decimal from 00 to 39. Only six Eprom inputs are required and, for the "Novice" band, only six outputs are required, although we have used all eight in the actual layout. The three Most Significant Bits remain the same for all forty channels and will change only if you need to add more channels in the 10 meter band. If that is the case, you will probably need to alter our layout to accommodate a transistor inverter and allow access to at least one other Eprom input pin in order to switch between bands.

Notice how much simpler and cheaper the Eprom method is when compared with the alternative of changing THREE crystals for every band which you want to add! Even then, without an Eprom board, you would still be left with the FCC "hops".

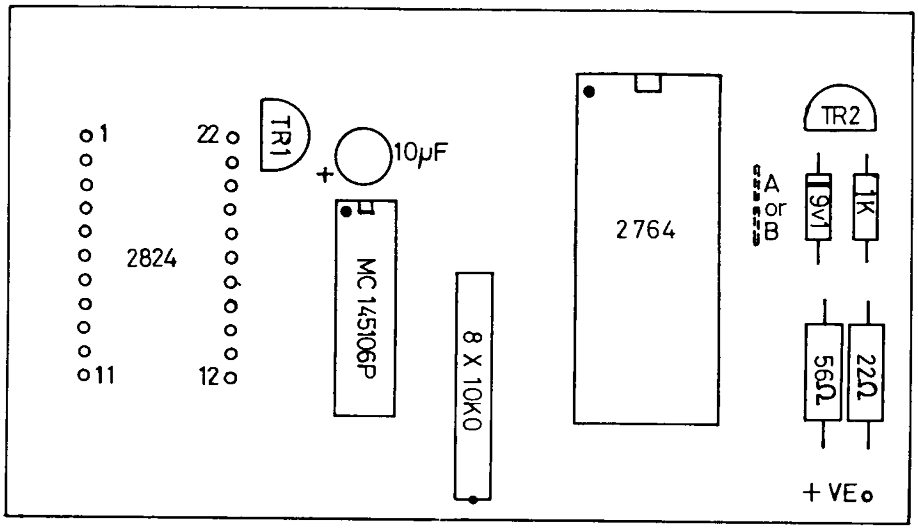
Cobra 146 GTL uPD2824 "Novice" using MC145106P

FCC channels			Novice Channels				
					(location)		
dec.	HEX	binary "N" code	dec.	HEX	binary "N".	switch	ch.
135	87	0 1000 0111	244	F4	0 1111 0100	00	40
134	86	0 1000 0110	243	F3	0 1111 0011	39	
133	85	0 1000 0101	242	F2	0 1111 0010	38	
132	84	0 1000 0100	241	F1	0 1111 0001	37	
131	83	0 1000 0011	240	F0	0 1111 0000	36	
130	82	0 1000 0010	239	EF	0 1110 1111	35	
129	81	0 1000 0001	238	EE	0 1110 1110	34	
128	80	0 1000 0000	237	ED	0 1110 1101	33	
127	7F	0 0111 1111	236	EC	0 1110 1100	32	
126	7E	0 0111 1110	235	EB	0 1110 1011	31	
125	7D	0 0111 1101	234	EA	0 1110 1010	30	
124	7C	0 0111 1100	233	E9	0 1110 1001	29	
123	7B	0 0111 1011	232	E8	0 1110 1000	28	
122	7A	0 0111 1010	231	E7	0 1110 0111	27	
121	79	0 0111 1001	230	E6	0 1110 0110	26	
118	76	0 0111 0110	229	E5	0 1110 0101	25	
117	78	0 0111 1000	228	E4	0 1110 0100	24	
119	77	0 0111 0111	227	E3	0 1110 0011	23	
117	75	0 0111 0101	226	E2	0 1110 0010	22	
116	74	0 0111 0100	225	E1	0 1110 0001	21	
115	73	0 0111 0011	224	E0	0 1110 0000	20	
113	71	0 0111 0001	223	DF	0 1101 1111	19	
112	70	0 0111 0000	222	DE	0 1101 1110	18	
111	6F	0 0110 1111	221	DD	0 1101 1101	17	
110	6E	0 0110 1110	220	DC	0 1101 1100	16	
108	6C	0 0110 1100	219	DB	0 1101 1011	15	
107	6B	0 0110 1011	218	DA	0 1101 1010	14	
106	6A	0 0110 1010	217	D9	0 1101 1001	13	
105	69	0 0110 1001	216	D8	0 1101 1000	12	
103	67	0 0110 0111	215	D7	0 1101 0111	11	
102	66	0 0110 0110	214	D6	0 1101 0110	10	
101	65	0 0110 0101	213	D5	0 1101 0101	09	
100	64	0 0110 0100	212	D4	0 1101 0100	08	
98	62	0 0110 0010	211	D3	0 1101 0011	07	
97	61	0 0110 0001	210	D2	0 1101 0010	06	
96	60	0 0110 0000	209	D1	0 1101 0001	05	
95	5F	0 0101 1111	208	D0	0 1101 0000	04	
93	5D	0 0101 1101	207	CF	0 1100 1111	03	
92	5C	0 0101 1100	206	CE	0 1100 1110	02	
91	5B	0 0101 1011	205	CD	0 1100 1101	01	1

Table of channel switch codes



P.C. Foil Pattern

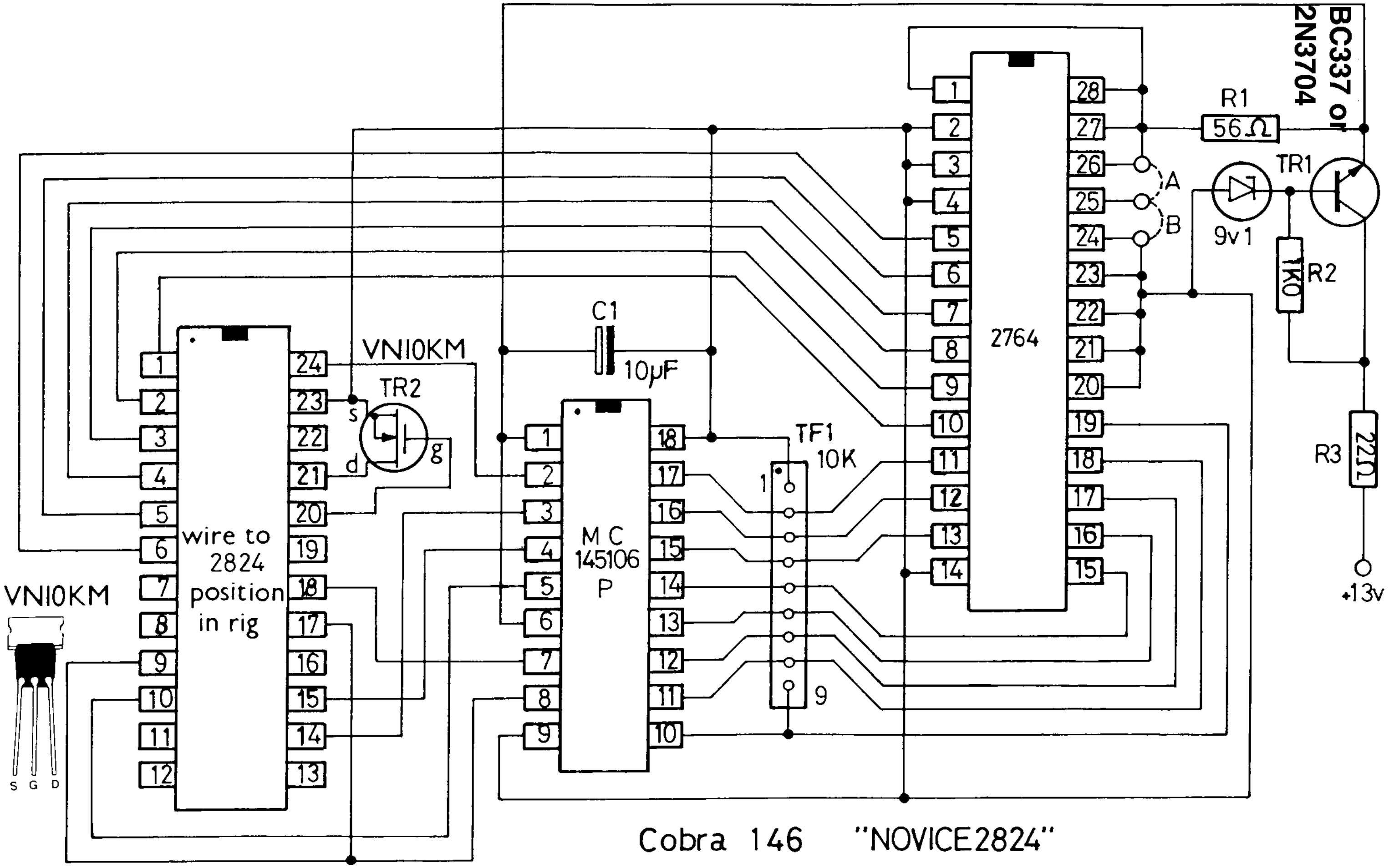


Novice 2824 eprom board.

NOTE :-

HOLES 1 to 22 SHOULD CONNECT TO MAIN BOARD VIA RIBBON CABLES. DIRECT TO µPD 2824 POSITION.

Component Layout



Cobra 146 "NOVICE2824"

PLL02A Example

The PLL02A chip is used extensively for export models and in 40 channel rigs in the USA. It is a versatile PLL with nine programming pins, however the export rigs do not take advantage of this and normally switch in different crystals to change the frequency band.

The crystal fitted for the FCC band is normally 10.0525 MHz followed by a doubler circuit, although a few export models use a crystal of twice this frequency at 20.105 MHz without the doubler. The coding for the Eprom is unaffected by which of these crystals is fitted and the PLL02A has just enough range of coding to allow conversion to the "Novice" band without the need to change the crystal. A very cheap board can be made with only an Eprom, voltage regulator and a few resistors.

The normal "N" codes are 255 to 211, giving 26.965 to 27.405 MHz and the Novice band would therefore require codes of 71 to 32. You can see from the table that the two MSBs are "1" for the FCC band but "0" for the "Novice" band. Pins 7 and 8 on the PLL02A must, therefore, be grounded. Pin 7 is already grounded but, since the 5.5 volt track runs through 8 and 9, you will have to do a bit of cutting to achieve the alteration! The Eprom output pins must be connected to pins 15 - 10 on the PLL02A. The tracks between these pins and the channel switch will have to be cut and six Eprom inputs connected to this switch.

In all of the conversions mentioned so far you will have to retune quite a lot of coils in the rig. This procedure is not for the "novice" and should be left to a qualified technician. Suitable 10 meter conversion boards with fitting instructions are available from CB City, in the USA, or S.J. Tonks in Britain. There should be plenty of information in this book to allow you to make your own, however, should you so wish!

The general purpose Eprom board, described later, is suitable for this, and other, conversions.

The location codes in the table are worked out from the channel switch codes in the American rig. There are six inputs: see if you can understand how they were derived. As a clue, channel 40 "N" code is normally 211 in decimal. That's 011010011 in binary, but the three MSBs are not needed, leaving 01 0011 or 13 in HEX.

Ham International "Novice" conversion

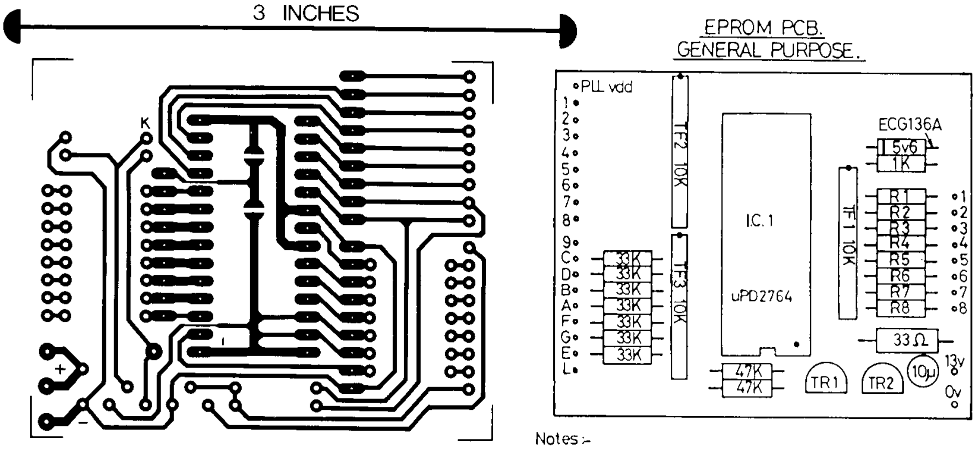
"N" code		PLL02A pin numbers									Eprom contents	location
ch	dec.	7	8	9	10	11	12	13	14	15	HEX	HEX
40	32	0	0	0	1	0	0	0	0	0	20	13
39	33	0	0	0	1	0	0	0	0	1	21	14
38	34	0	0	0	1	0	0	0	1	0	22	15
37	35	0	0	0	1	0	0	0	1	1	23	16
36	36	0	0	0	1	0	0	1	0	0	24	17
35	37	0	0	0	1	0	0	1	0	1	25	18
34	38	0	0	0	1	0	0	1	1	0	26	19
33	39	0	0	0	1	0	0	1	1	1	27	1A
32	40	0	0	0	1	0	1	0	0	0	28	1B
31	41	0	0	0	1	0	1	0	0	1	29	1C
30	42	0	0	0	1	0	1	0	1	0	2A	1D
29	43	0	0	0	1	0	1	0	1	1	2B	1E
28	44	0	0	0	1	0	1	1	0	0	2C	1F
27	45	0	0	0	1	0	1	1	0	1	2D	20
26	46	0	0	0	1	0	1	1	1	0	2E	21
25	47	0	0	0	1	0	1	1	1	1	2F	24
24	48	0	0	0	1	1	0	0	0	0	30	22
23	49	0	0	0	1	1	0	0	0	1	31	23
22	50	0	0	0	1	1	0	0	1	0	32	25
21	51	0	0	0	1	1	0	0	1	1	33	26
20	52	0	0	0	1	1	0	1	0	0	34	27
19	53	0	0	0	1	1	0	1	0	1	35	29
18	54	0	0	0	1	1	0	1	1	0	36	2A
17	55	0	0	0	1	1	0	1	1	1	37	2B
16	56	0	0	0	1	1	1	0	0	0	38	2C
15	57	0	0	0	1	1	1	0	0	1	39	2E
14	58	0	0	0	1	1	1	0	1	0	3A	2F
13	59	0	0	0	1	1	1	0	1	1	3B	30
12	60	0	0	0	1	1	1	1	0	0	3C	31
11	61	0	0	0	1	1	1	1	0	1	3D	33
10	62	0	0	0	1	1	1	1	1	0	3E	34
09	63	0	0	0	1	1	1	1	1	1	3F	35
08	64	0	0	1	0	0	0	0	0	0	40	36
07	65	0	0	1	0	0	0	0	0	1	41	38
06	66	0	0	1	0	0	0	0	1	0	42	39
05	67	0	0	1	0	0	0	0	1	1	43	3A
04	68	0	0	1	0	0	0	1	0	0	44	3B
03	69	0	0	1	0	0	0	1	0	1	45	3D
02	70	0	0	1	0	0	0	1	1	0	46	3E
01	71	0	0	1	0	0	0	1	1	1	47	3F

Table of channel switch codes

A General Purpose Eprom Board

We have designed a simple printed circuit board which you could use for experimentation, or even build and sell to your friends. It allows you to fit a 2516, 2716, 2532, 2764, 27128 or 27256 Eprom and the layout makes provision for all the resistors you are likely to need. The design includes a lot of options which you may not require and it is left to you to decide which components you want to fit for your particular conversion, based on the information given in this book.

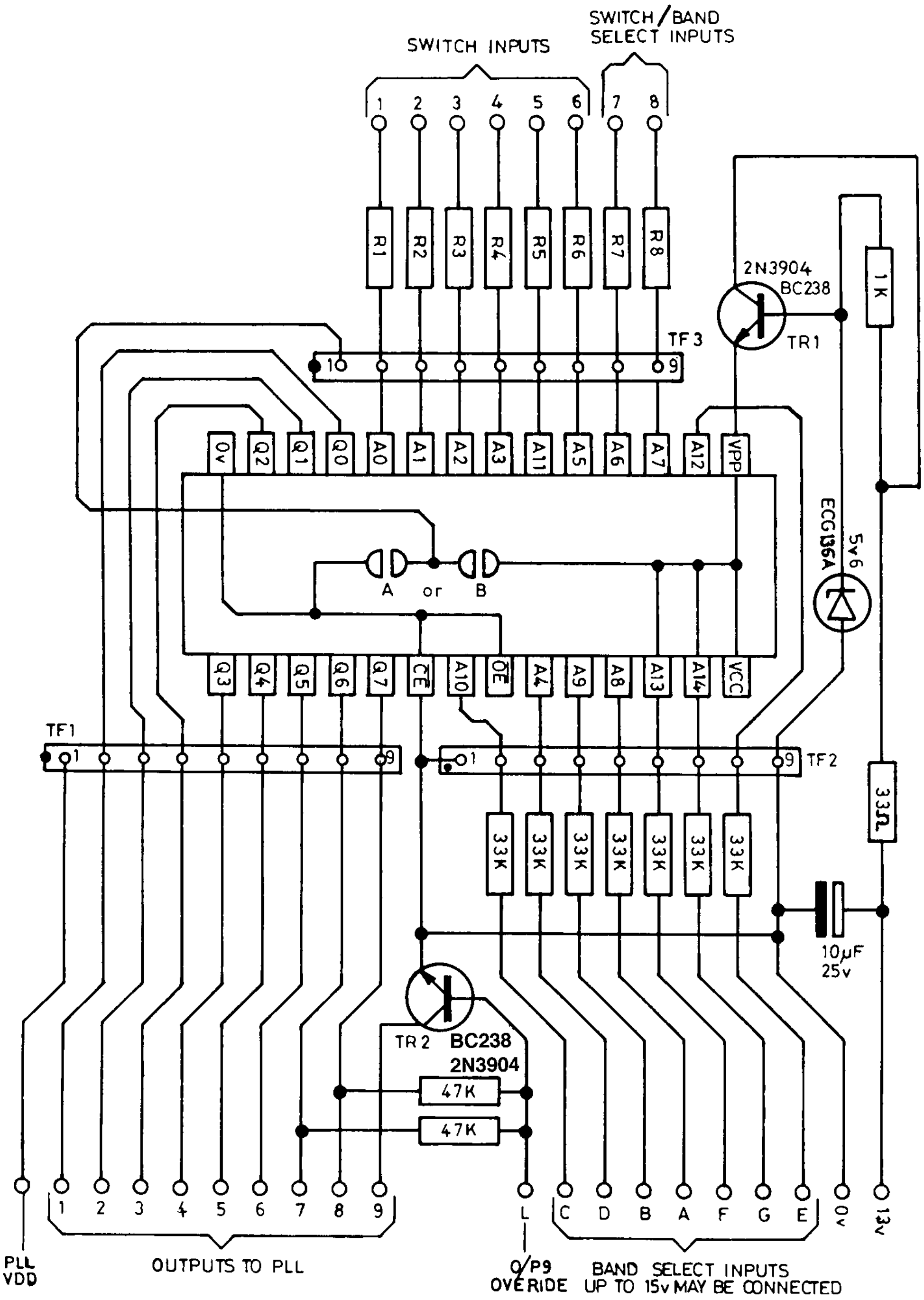
Note that, in order to accommodate such a wide range of Eproms, some input pins are connected permanently to +5v. The input locations must, therefore, take this into account. Alternatively, to suit the particular Eprom fitted, you could cut the offending tracks and connect the inputs to ground, instead.



Notes:-

1) VALUE FOR RESISTORS R1 – R8 TO BE SELECTED SUCH THAT 5v MAX IS OBTAINED ON A0 – A7.
2) PINS 1 – 9 SHOWN ABOVE LEFT ARE OUTPUTS TO THE PLL.
3) PINS MARKED A – G ARE BAND SELECT INPUTS.
4) PIN MARKED 'L' IS OUTPUT 9 OVERIDE.
5) PINS 1 6 ON THE RIGHT ARE SWITCH INPUTS.
6) PINS 7 & 8 ON THE RIGHT ARE SWITCH OR BAND SELECT INPUTS.

General purpose Eprom board



General purpose eprom board.

APPENDIX 1

Erasing

Eproms, by definition, are erasable under Ultra Violet light. Since many second hand Eproms are available, cheaply, it is worth considering the use of such devices to save money. Before you can utilize them you must erase the existing contents, effectively resetting all bytes to FF.

The "window" in each device is not glass but quartz which allows UV light to pass through much more readily than glass does. The UV light must be within a specific frequency band to be effective. Ordinary sunlight will erase an Eprom but it would probably take weeks to effect complete erasure. A "black light" tube, as used for display purposes and in discos is no use since the UV which it produces is much too low in frequency. The type of light which does work is produced by a white tube which is called an "actinic" tube because of its ability to kill germs by radiation. You can be sure, therefore, that your Eproms are clinically clean, after erasure!

Cleaning Time

The time required to erase an Eprom depends upon its distance from the light source, the intensity of the light, the contents of the Eprom and the physical characteristics of the Eprom itself. As a very rough guide, an "average" Eprom will require at least 15 minutes but not more than an hour to erase when placed one inch from, and facing, an 8 Watt tube. It is possible to make an eraser, yourself, but due regard must be paid to all aspects of Electrical safety, the high temperature of the tube and the danger to living tissue caused by UV light.

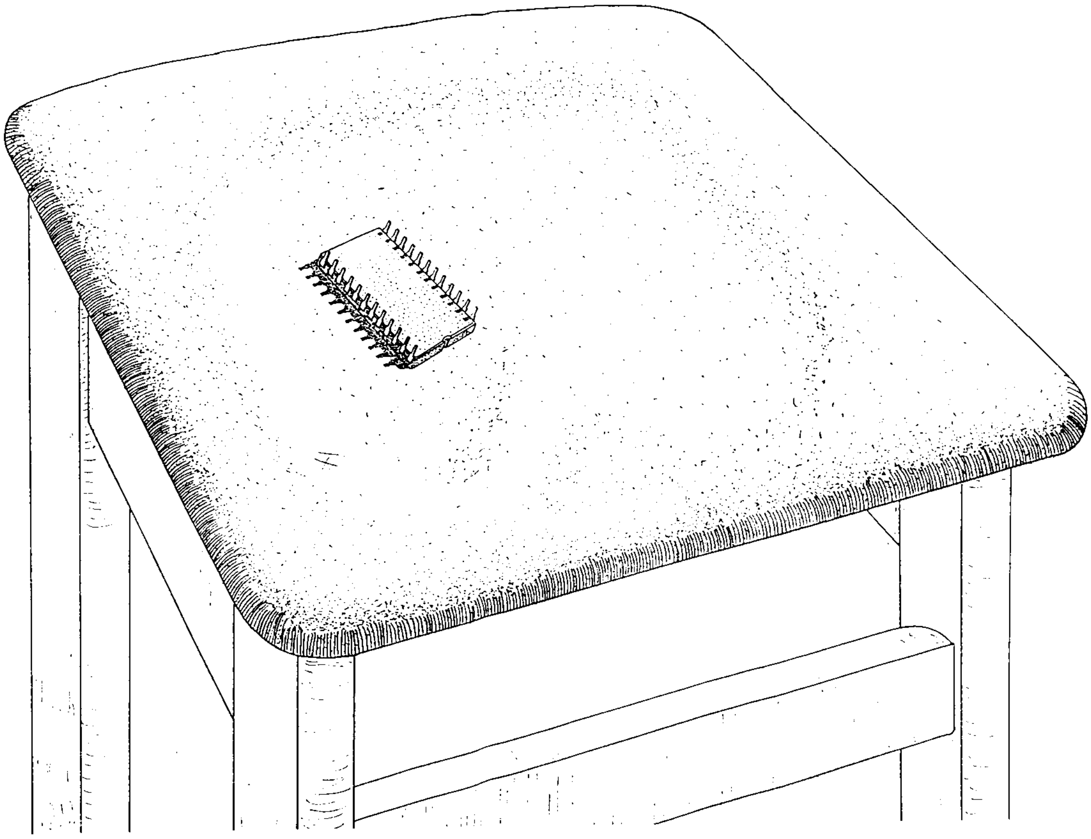
WARNING. UV LIGHT CAN DAMAGE SKIN AND CAUSE BLINDNESS. The unit must be ventilated but should not allow UV light to escape. Some excellent commercial units are available, moderately priced, with safety interlocks and with or without a timer. One unit even "interrogates" the Eprom to ensure it is erased, exposes it for a further calculated period to make sure, then turns off. This type of unit is supposed to avoid over-exposure, thereby increasing the long term reliability of the Eprom and increasing the number of erasures which are possible. There is a limit to the number of times that an Eprom may be erased before damage occurs. This limit varies widely from one Eprom to the next. Some do not survive even the first erasure, whilst others can be wiped and re-used numerous times.

Non-E-proms

There are also some plastic encapsulated "Eproms" which can't be erased. These are known as One Time Programmable (OTP) Eproms and can be somewhat cheaper than the standard type, since you don't pay for a ceramic housing with a quartz window. Obviously, it's no use buying these second hand! The programming time for these can be seconds, rather than minutes. They are ideal for high volume production, but of little use to the experimenter.

Unreadable Devices

There is another type of Eprom which results from incorrect programming voltage or, especially, from being plugged into its socket the wrong way round. This device is known as the **T**otally **U**n**R**eadable **D**evice, or TURD, and should be placed pins uppermost upon your chair to remind you to be more careful next time.



"...should be placed, pins uppermost, on a chair to remind you..."

Programming

Many designs for Eprom programming units have been published. Some are very good, some mediocre. All work well, within their limitations. All the units, commercial or home built, which connect to a PC are very versatile because you have the full programming power of the PC to assist. Usually you can type the HEX codes directly into the computer memory, then "dump" them into the Eprom. It is also possible to save the relevant section of memory to disc or tape. In addition you may be able to write simple routines for converting decimal to binary or to swap the bits in each binary code, as described previously, to simplify the hardware design of the Eprom board. Most computers can convert readily between decimal and HEX, thus eliminating even this simple task.

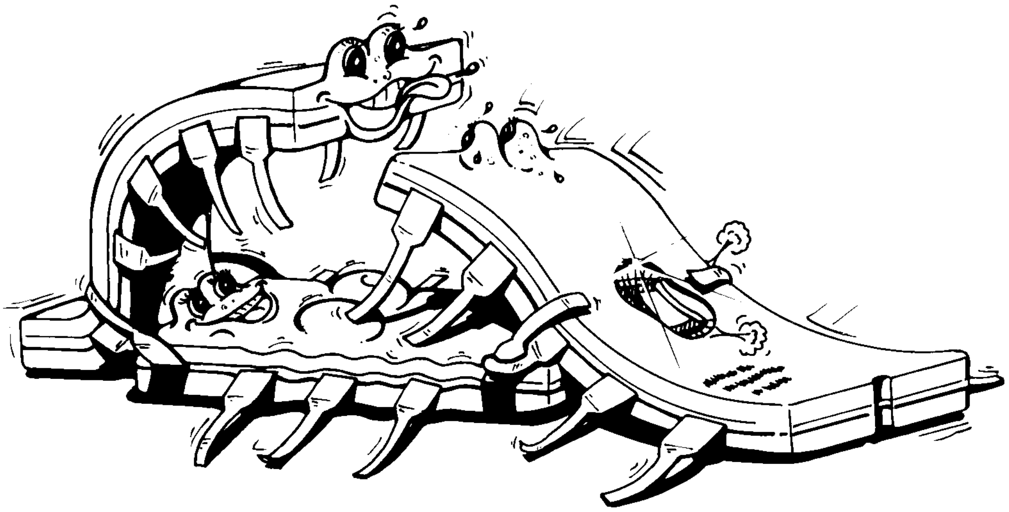
If you intend to buy one then get the best programming unit you can afford. Most of the units sold for connection to a Personal Computer (PC) are quite intelligent and use a friendly Menu system to guide you through the necessary choices. They all comprise a printed circuit assembly, usually in a box, which you must connect to the computer, and a software package on disc or Eprom. The box will contain a socket - usually only one - which permits easy insertion of an Eprom by releasing a locking tab or lever. It may also contain a number of option switches, although the better models use internal switching which is controlled by the software. Usually, the box will need a supply of mains electricity.

Options

You will be required to choose the options, which normally include setting a switch or pressing a computer key, to choose the part number of the Eprom (2716 etc.) and the programming voltage. Three programming voltages are used: 26 volts 21 volts and 12.5 volts, the latter sometimes (but not always) being designated by an "A" suffix to the Eprom part number, for instance the 27128A. An immediate exception to this "rule" is the 2716A which programs at 21 volts. Failure to choose the correct voltage can destroy the Eprom or even damage the programming unit, consequently you MUST consult the manufacturer's literature. In the very cheap units you may not have the option of 12.5 volts and the choice of Eproms will be restricted to those of certain sizes from specific manufacturers, whilst in the more expensive units you are likely to be asked, by the software program, which manufacturer's device you have so that the programming pulses can be set accordingly. They might also perform the programming in a much shorter time.

Gang Bangers

If you are going into Eprom copying in a big way there are commercial units (known as "gang bangers" in the trade) which will program several Eproms at a time. The simpler type for connection to a PC will seldom program more than one device at a time, but is far more versatile for development purposes.

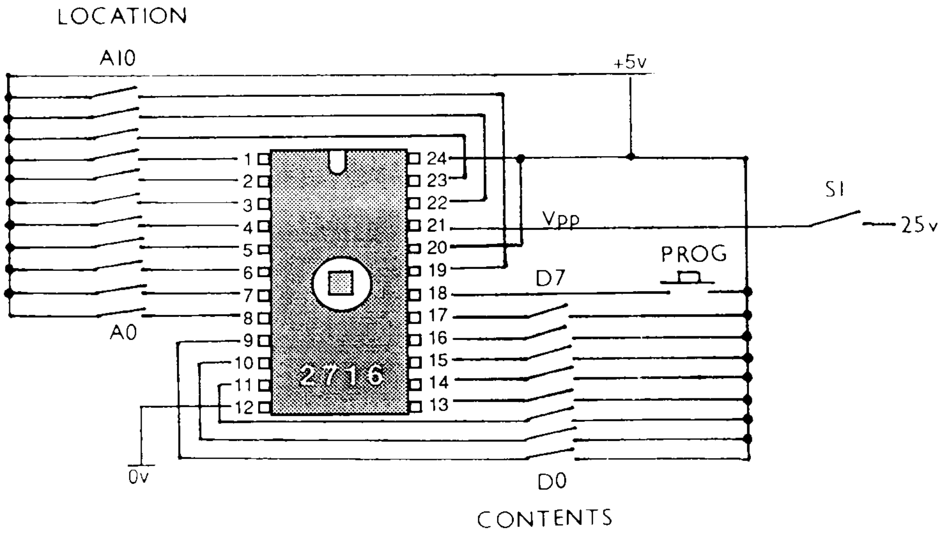


gang bangers

Manual Programming

It is actually possible to enter data into an Eprom without using an automatic programming unit at all. The method is tedious and varies according to the type of Eprom you are using. Basically, you must connect a row of switches to the input address pins to select either 0v or +5v. Similarly, the output Data pins must be connected to switches. Some Eproms require a Vcc supply of +5v and others require +6v during programming. In addition, a supply of +26v, +21v or +12.5v is required on the Vpp pin. Set the address switches to select the location; set the Data switches to choose the contents; apply a programming pulse to the Pgm pin for the appropriate duration (approximately 50 milliseconds for a 2716, less for other devices). The polarity and duration of the pulse are very important and must be chosen from the manufacturer's data. A simple pulse timing circuit would be required, in addition to the stabilised voltage supplies. Of course, since you are using switches, the locations and codes must be in Binary notation. It is easy to damage Eproms when programming in this way and the manufacturer's literature should be consulted. The supply voltage Vcc must be switched simultaneously with the voltage on pin Vpp, or else Vcc must already be switched on at the time when Vpp is switched either on or off. If this point is ignored you will certainly destroy the Eprom.

In the VERY crude circuit shown here you must first set the binary location code on switches A0 to A10. Then set the binary data on switches D0 to D7. Close S1; poke the "PROG" button for an instant (can you guess 50mS ?); open S1. It *might* work!



manual programming

For a typical 2716, the following conditions must exist during programming mode: Vcc must be at +5v; Vpp must be at +26v; Program (PGM) at 0v; the required contents present on the output pins and the **location** code on the input pins. Pulsing PGM to +5v for 50 milliseconds burns the data present on D0 to D7 into the location addressed by A0 to A10.

Because there is such a wide variation of programming parameters between even Eproms bearing identical numbers but from different manufacturers, this book will not attempt to delve further into methods of programming. It already forms the subject of several books from each manufacturer, which should be consulted according to the specific type that you wish to program. If you buy a comprehensive programming unit and follow its instructions carefully, you should have little difficulty in programming most Eproms without problem.

WARNING Incorrect programming methods can cause an Eprom to become EXTREMELY HOT. The publishers accept no responsibility for damage or injury caused by careless or uneducated use of information printed in this book. If in doubt seek expert advice!

Eproms

The 2716 and 2516 have 24 pins and are generally interchangeable, although slight differences in programming parameters exist between the various manufacturers' devices.

The 2732 and 2532 also have 24 pins and have the same capacity as each other but BEWARE: they are NOT pin for pin compatible, as the sketches at the back of this book show!

The 2764 and 27128 have 28 pins and can be interchanged, but bear in mind that the 27128 has twice the capacity of the 2764.

The 27256, with twice the capacity of the 27128, is also compatible, having only one pin function different. It is generally very fast to program, compared with the smaller capacity devices!

Double again, the 27512 is probably the largest capacity device you will want to deal with, although there are even bigger Eproms available and their prices are falling all the time.

CMOS devices are also available. These are usually denoted by a letter C in the number, for example 27C128. The current consumption is lower than that of standard devices but they are somewhat more susceptible to damage from static electricity and also more expensive.

Eproms usually have number suffixes such as 2764-25. The -25 refers to the speed at which the Eprom can be read. For CB applications we are not the slightest bit interested in the speed, since it takes a relatively long time to change channel, compared with the speed at which a microprocessor works! Consequently, you can buy the cheapest, slowest Eproms you can find, which will often have the suffix -30 or -45 or nothing. Remember, too, that we mentioned earlier that a letter suffix might refer to the programming voltage. Check this and ensure that the Eprom is compatible with your programming unit.

All Charged Up

Whenever you are handling ICs you should take precautions to avoid the generation of static electricity. Always touch the surface on which an IC rests before picking it up and do the same before putting it down. This will ensure that your body is at the same electrical potential as the surface and thereby ensure that you do not discharge current through the IC. You can generate a large voltage simply by moving around in clothing made of synthetic material. Removing a nylon shirt or pullover is a good method of creating static electricity and if you touch a grounded surface after doing so you can expect to receive a nasty shock. Imagine, therefore, what this does to an IC which contains tiny conductors only a few atoms wide!

Decimal to Binary

The following simple computer program in BASIC will convert any decimal number less than 4096 into Binary.

```
 10 DIM W(12)
 20 INPUT"DECIMAL ? "N
 30 IF N>4095 THEN GOTO 20
 40 IF N<0 THEN GOTO 20
 50 J=N DIV 2048
 60 Z=N MOD 2048
  70 K=Z DIV 1024
 80 Z=Z MOD 1024
 90 L=Z DIV 512
 100 Z=Z MODl512
 110 M=Z DIV\256
 120 Z=Z MOD 256
 130 A=Z DIV 128
 140 Z=Z MOD 128
 150 B=Z DIV 64
 160 Z=Z MOD 64
 170 C=Z DIV 32
 180 Z=Z MOD 32
 190 D=Z DIV 16
200 Z=Z MOD 16
210 E=Z DIV 8
220 Z=Z MOD 8
230 F=Z DIV 4
240 Z=Z MOD 4
250 G=Z DIV 2
260 Z=Z MOD 2
270 H=Z
280 W(0)=J
290 W(1)=K
300 W(2)=L
310 W(3)=M
320 W(4)=A
330 W(5)=B
340 W(6)=C
350 W(7)=D
360 W(8)=E
370 W(9)=F
380 W(10)=G
390 W(ll)=H
400PRINTW(0);W(1);W(2);W(3);
W(4);W(5);W(6);W(7);W(8);
W(9);W(10);W(ll)
410 GOTO 20
```

Note: Some dialects of BASIC do not provide the MOD and DIV operators used in the above program. The alternative method available in less comprehensive dialects is far from elegant, but works just the same.

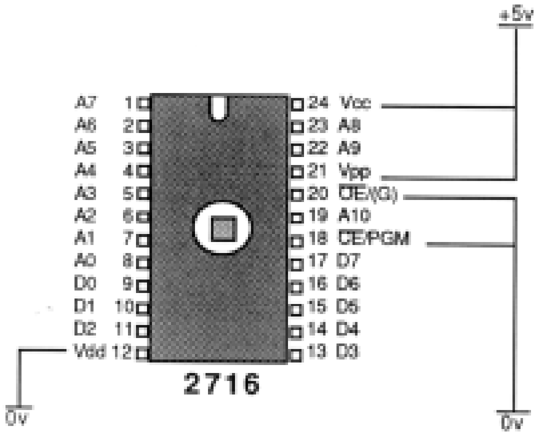
`For A=B MOD C ...` use `... A=((B/C)-(INT(B/C)))*C`

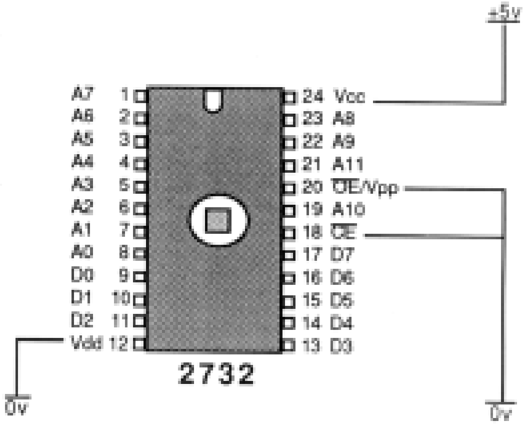
`For A=B DIV C ...` use `... A=INT(B/C)`

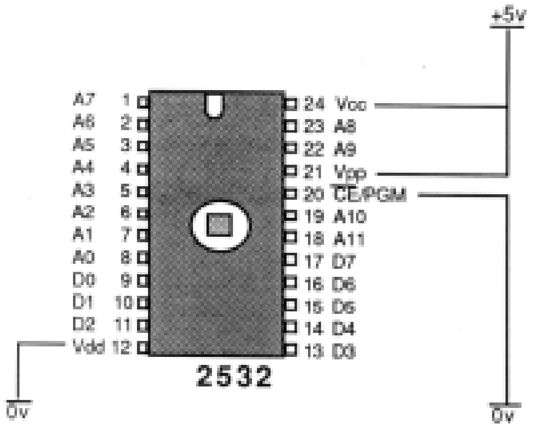
The pin-out sketches, on the following pages, indicate the connections necessary for normal read operation, NOT for programming. Every care has been taken to ensure that the information is accurate, but no responsibility can be accepted for problems arising from errors or omissions in this book. It is recommended that you consult the appropriate literature from the supplier of each particular device.

APPENDIX 2

Pin Out Sketches



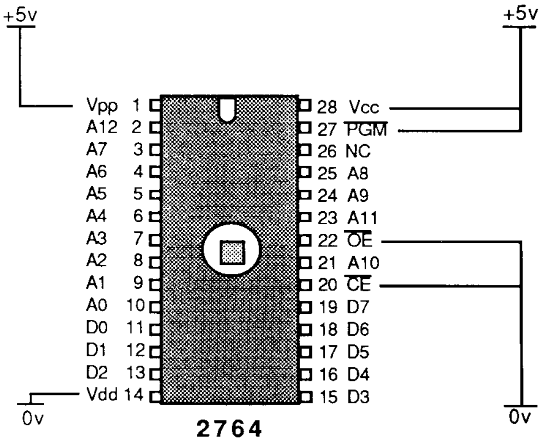




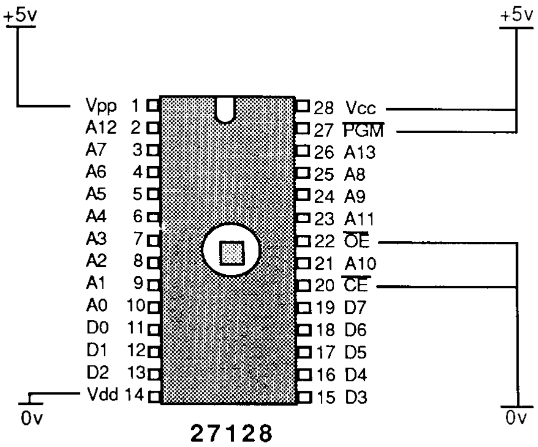
Note: A bar above the pin designation indicates that the input is active when 'low'. See glossary of terms for an explanation of pin functions.

Warning!
Devices labelled TMS2716 require three power supply connections and have a different pin-out from other 2716 and TMS2516 Eproms. They can damage your programmer!

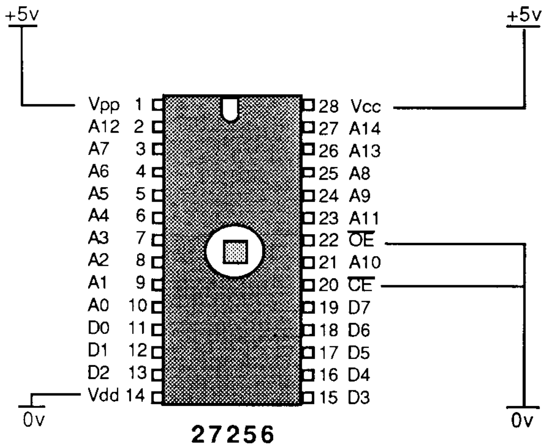
…more pinouts...



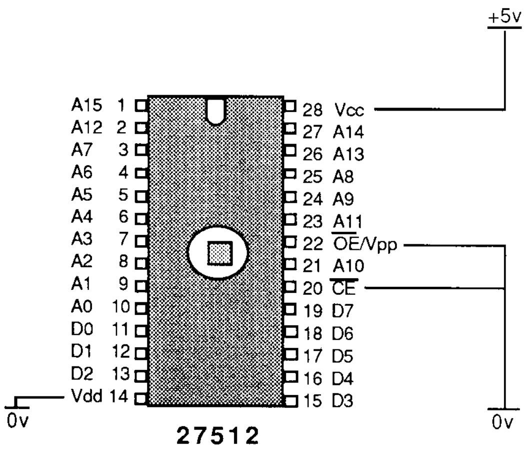
2764



27128



27256



27512

APPENDIX 3 (not included in the original publication)

Convert to 10 metres the Chip Way

You may wonder why an ordinary mortal who is not a licensed Amateur should have wished to design a ten metre conversion which he can't legally use. The simple answer was, of course, money and the design was a natural development from much work on other successful conversions for C.B. Radios. Unfortunately, the same apathy which prevents my learning morse also prevented the successful marketing of the unit, so I decided to offer the design to the world in return for a pittance.

The circuit to be described was developed for use with the most common type of C.B. radio which uses the ubiquitous LC7137 or LC7132 frequency synthesiser. When the C.B. radio is receiving, the V.C.O. runs at the input frequency minus 10.695 MHz. On transmit, the V.C.O. runs at half the output frequency. For example, channel 40 normally produces a frequency of 27.99125 MHz and to raise this to, say, 29.70 we need to add 1.70875 MHz to the V.C.O. during receive and, during transmit, 0.853 MHz which is half that amount. One beauty of the method to be described stems from the realisation that the 10.24 MHz crystal oscillator output can be divided to produce almost exactly these frequencies. Mixing the frequency with the V.C.O. output then filtering out the required sum frequency, however, is guaranteed to create unwanted spurii at the antenna.

Brainwave

An idea came from a circuit published by Hugh Allison in "The Short Wave Magazine" (Ref. 2) where the 10.24 MHz division frequency is subtracted from the V.C.O. output to the P.L.L. synthesiser only. The actual V.C.O. frequency still goes directly to the transmit and receive circuitry. No mixing of analogue signals is necessary and transmitted interference is thus minimised. The P.L.L. is fooled into thinking that the V.C.O. frequency is too low and increases it accordingly!

How It Works

Initial tests were performed on the circuit shown in Fig. 2. IC1 is a divider chip which is configured to produce a divide-by-12 output on pin 12 and a divide-by-6 output on the connection of pins 8 and 14. The 10.24 MHz reference frequency enters IC1 at pin 1. Dividing 10.24 MHz by 12 and by 6 gives 0.853 and 1.706 MHz respectively. These values are fractionally too low for our needs but we can adjust the 10.24 MHz reference crystal frequency quite easily to compensate.

IC2 acts as a simple switch controlled by the TX/RX line which goes low on transmit. During transmit, the 0.853 MHz square wave is passed to IC3a and during receive the 1.706 MHz square wave is gated through, instead.

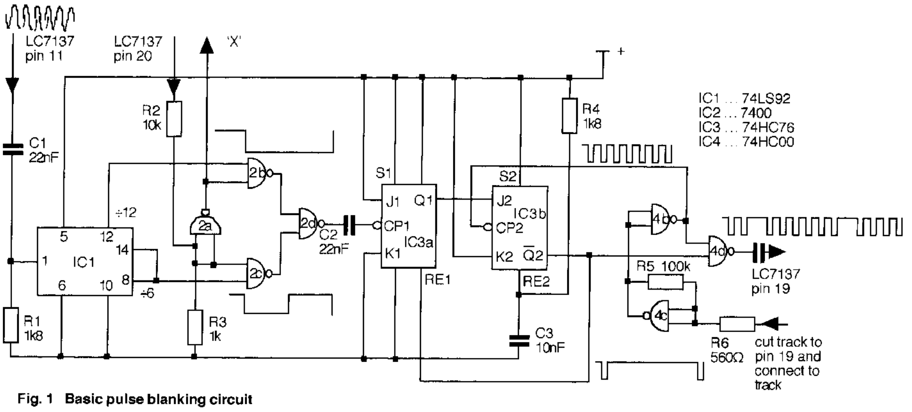


Fig. 1 Basic pulse blanking circuit

Now, the original idea was to use a simple transistor mixer to combine the V.C.O. output with the output from IC2. The difference frequency would then be filtered out by an L-C 'tank' circuit to be passed to the LC7137. This idea failed to work in practice and was abandoned in favour of the digital approach (Ref. 1) pioneered by Bill Sparks G8FBX and Colin Horrabin G3SBI.

Instead of mixing and filtering, we simply blank out a certain number of Hz from the V.C.O. output during each second. The LC7137 does not really care about the waveform shape since it incorporates digital circuitry itself.

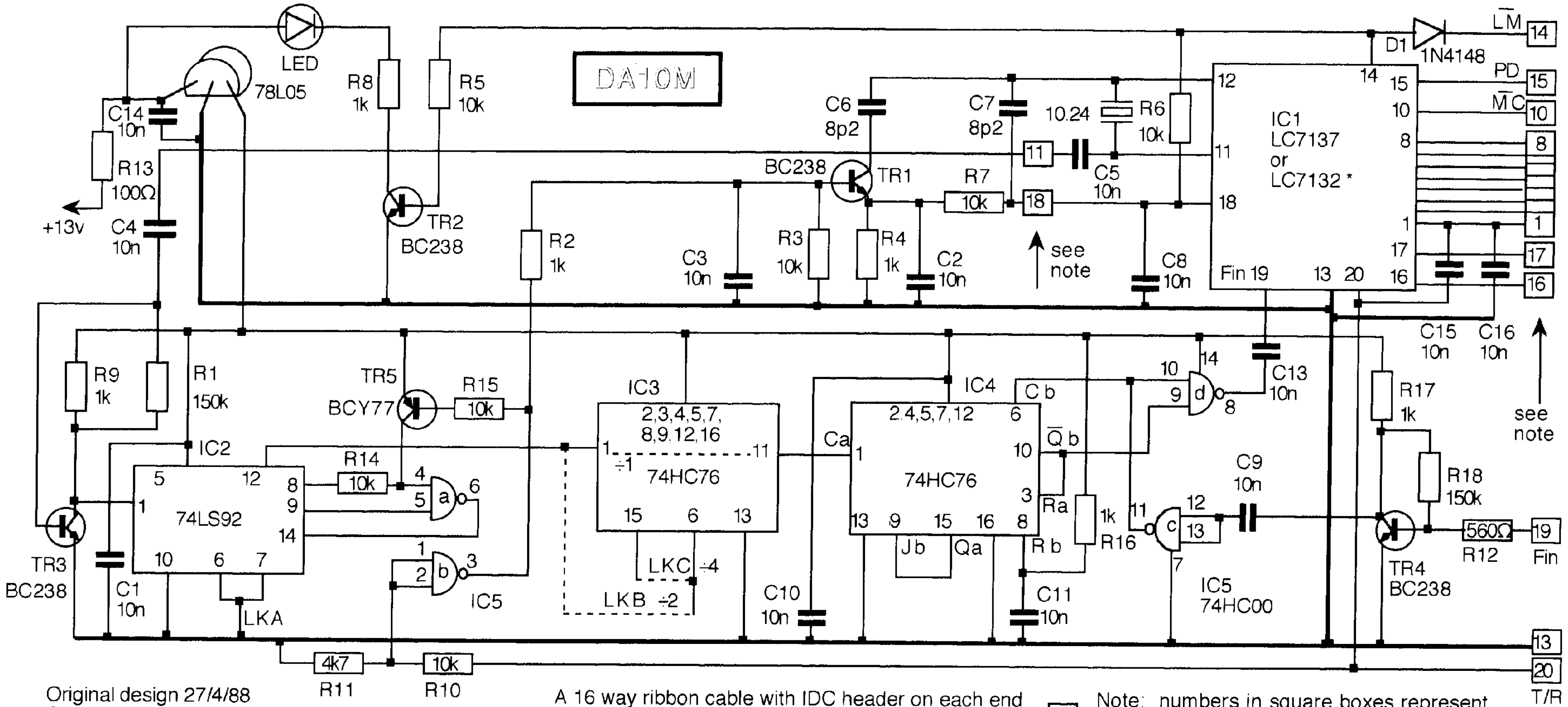
The occasional missing pulse, therefore, causes no problems. Since the count is less than that produced by the V.C.O. the LC7137 increases the V.C.O. frequency until it receives the correct average rate of pulses. The object, then, is to remove one V.C.O. pulse for every square wave which is produced by the 10.24 MHz divider circuit.

You might be tempted to think that simply to differentiate the square wave to form a spike which drives a monostable would be a satisfactory method of generating a blanking pulse for each leading edge of the square wave. Well, it would be, were it not for the fact that the V.C.O. and the 10.24 MHz crystal oscillator are not synchronised.

Also, the length of the blanking pulse needs to be varied between transmit and receive because the V.C.O. frequency changes. Finally, the length of the blanking pulse would also have to be adjusted according to the channel frequency which also alters the V.C.O. frequency. The problem is solved by the use of two "JK Flip-Flops" in IC3. IC3a is programmed with J high and K low so that each time the clock input CP1 sees the falling edge of a square wave, output Q1 goes high. This action sets J2 high. (The output of IC3b has been reset low by the combination of R4/C3 at switch-on and is unaffected at this time by the state of J2).

Now, thanks to the efforts of Messrs Sparks and Horrabin who developed a way to use a 74COO as an amplifier, we have a nice, sharp V.C.O. pulse waveform appearing from IC4b and feeding the clock input CP2. (This explanation is not difficult, just 'wordy'; follow the circuit in Fig. 1.)

When CP2 sees the falling edge of a pulse, the contents of input K2 are transferred to /Q2 so this output goes low. IC4d is now disabled so the next V.C.O. pulse from IC4b can not pass. IC3a input RE1 goes low and resets Q1 low so input K2 goes high. When the next clock pulse falling edge appears from IC4b it transfers the contents of K2 to the output which goes high, thus enabling IC4d which can then pass more clock pulses until the next square wave falling edge hits Cp1.



Original design 27/4/88
© 1991 M T Pickering

A 16 way ribbon cable with IDC header on each end is used to connect DA10M to main PCB.

□ Note: numbers in square boxes represent pin numbers of LC7137/7132 in main PCB

Only rigs using LC7137 can be modified to operate on 29.31 to 29.70 MHz. LC7132 bands start 0.64 MHz lower and have 'gaps'.

Fig. 2 Complete circuit of 10 metre conversion

*Use I.C. as fitted in rig.

By this method we blank out one V.C.O. pulse after the input square wave goes low, reset the square wave at the end of the blanked V.C.O. pulse, then do nothing until the square wave input goes low again. Since the circuit entirely eliminates the need for any timing elements, but relies totally on pulse falling edges, its operation is independent of the frequencies involved and noods no adjustment.

Practical Considerations

The frequencies can be as high as 19 MHz and the rise times are very short, consequently pay careful attention to the need to decouple every I.C. with a lnF ceramic capacitor. Connections must be as short as possible and the risk of interference to the receiver section of the transceiver can not be stressed too highly. However, this circuit will work on pre-drilled copper stripboard — I've done it — but better results can be obtained from a printed circuit board with an earth plane on the component side.

The use of a low-power Schottky device for IC3 is important and IC4 must be a 74HCOO. A 7400 is definitely not fast enough. The slightly more expensive 74ACOO works well but, since the pulse rise times are actually shorter, more interference is generated than with the 74HCOO and this noise affects performance during receive.

Point 'X' goes high on transmit and may be used to control a simple transistor or diode circuit to add capacitance to the crystal oscillator whose frequency must be lowered slightly during transmit. If this is not done, the transmit frequency will be some 2 kHz higher than the receive frequency. A stabilised 5 volt supply is required. I used a 78L05 regulator fed by a 1OO Ohm resistor from the 13 volt supply, to dissipate most of the power.

Complete Circuit

The final circuit is shown in Fig. 2 and includes a few more components to improve the repeatability and stability of the basic design. TR3 creates a buffer between IC2 and the 10.24 MHz crystal to eliminate the effects of loading. TR1 is used to load the crystal with an extra 8.2pF to reduce the frequency during transmit. Since 8.2pF capacitors proved hard to obtain, I used two pairs of 15pF capacitors with each pair connected in series.

TR2 lights an L.E.D. when pin 14 of IC1 goes high to indicate that the V.C.O. is in lock: an unnecessary frill, you might think, but it makes the re-tuning of the V.C.O. coil so much easier. To reduce the chip count, TR5 was introduced and the gating circuit was modified accordingly. TR4 was introduced to buffer the V.C.O. which otherwise is loaded by IC5. Some radios tolerated the loading and some did not so this addition ensures correct operation in any radio.

To simplify fitting the unit to a C.B. radio, a 16 wire ribbon cable with 16 pin header at each end was used. C15 and C16 were added to remove interference which the ribbon capacitance caused. Although the interference might have been eliminated by reverting to individual wires it was considered that the simplified fitting was important and, in any case, each constructor would connect individual wires in different ways using different lengths which might cause interference that would be difficult to eradicate.

The unit may also be fitted to a FCC specification radio which employs an LC7132 frequency synthesiser. This I.C. will produce frequencies which are 0.64 MHz lower, on average, than the LC7137 since it is designed to create the old American 'FCC' band or, as we now know it, the 'FCC' band. The channel spacings with the LC7132 are irregular and, for historical reasons, have 'gaps', rather than going in straightforward 10 kHz steps.

Normally IC2 would be linked directly to IC4 but, if IC3 is added, there is an option to lower the frequency band by the use of an extra divider stage. I leave it to you to calculate the actual frequencies which would be achieved. Note that it might be necessary to adjust the values of C6 and C7 if this option is used.

Construction

The entire unit can be built on pre-drilled copper strip board but you will never eliminate, entirely, the interference on receive which is caused by the fast risetime pulses in the logic circuit. The track layout shown can be made on single- or double-sided copper laminate and good performance with this layout is guaranteed. The parts are readily available and should be soldered to the board in the usual way. If a board with a copper ground plane is used, you will have to use a drill bit to remove the copper from around all holes except those for leads which are connected to 0 volts. As far as is practicable, these leads should be soldered on both sides of the board.

For best decoupling performance the board should use 10nF surface mount capacitors on the underside. These should be fitted last.

Alternatively, you might like to modify the design to suit a different radio such as the UnidenTM or TandyTM rigs which employ a TC9119 synthesiser chip. In this case the 10.24 MHz divider is simplified since the V.C.O. frequency is not doubled for transmission but an additive mixing method is employed for both receive and transmit. Consequently, only the 1.706 MHz division is required. Pinout of the TC9119 and information about other synthesiser I.C.s is available in The PLL Data Book (Ref. 3).

Fitting and setting up

Fitting the P.C.B. to a C.B. radio is quite straightforward. Use solder wick to de-solder the LC7137 from the main board and fit this I.C. into the conversion board. Remove the 10.24 MHz crystal from the radio and solder this to the conversion board. Connect the crystal output from C5 back into the main board at the hole vacated by pin 11 of the LC7137. The connection to pin 10 is usually not needed since its function is to ground whatever is connected to it if a 'miscode' is presented to inputs on pins 1 to 8. If a 'Delta Tune' facility is present, the associated capacitors may need to be removed or they will load the crystal unacceptably.

The 16 pin header plug should be soldered to the pads vacated by the LC7137. Make sure that pin 1 goes to pin 1. By the way, the reason for using a 16 way ribbon is that nobody manufactures a 20 pin header with 0.3" pitch.

Connect the 13 volt supply from the on/off switch to the board. Now, all that remains to be done is to remove the wax from the V.C.O. coil with the aid of a tooth pick (not a soldering iron!) and unscrew the ferrite core until the L.E.D. on the conversion board lights for both transmit and receive. In some receivers you will need to screw the core nearly all the way out. Tune the output coils for best performance. Do make sure that interference is within the accepted limits. Peak the receiver coils for best reception then wax all adjusters to prevent movement.

References:

1) "27 - 29 MHz Conversions". Bill Sparks G8FBH and Colin Horrabin G3SBI. "Radio and Electronics World" November 1984 and March 1985.

2) "How to Convert 'Unconvertible' CB rigs to Work on the Ten-Metre Band". Hugh Allison. "The Short Wave Magazine" March 1984. 3) "The CB PLL Data Book". Lou Franklin. ISBN 0-943132-05-3. 4) "The CB Eprom Data Book". Martin Pickering. ISBN 0-943132-16-9 5) "The Midnight Express" CB Radio repair manual. Martin Pickering.

Shopping List

Resistors

5% 0.4W carbon film

100Ω R13

560Ω R12

1kΩ..............R2, R4, R8, R9, R16, R17

4k7Ω R11

10kΩ R3, R5, R6, R7, R10, R14, R15

150kΩ.............R1, R18

Capacitors

Ceramic miniature plate

8p2F C6, C7

(or use 15pF in series)

multilayer ceramic 1206 size Surface Mount

10nF C1,C2,C3,C4,C5,C8,C9,C10,C11,C13,C14,C15,C16

(C12 was originally used to decouple IC3 but was deleted as unnecessary).

Semiconductors

BC238.............TR1, TR2, TR3, TR4 NPN general purpose silicon

BCY77.............TR5 PNP general purpose silicon

5 volt regulator..78L05

1 N4148...........D1

74LS92............IC2

74HC76 ....IC3 (optional)

74HC76 ....IC4

74HC00 ....IC5

LED 3mm dia.

Miscellaneous

P.C. Board l.C. Socket 20 pin/0.3" for IC1 16 way ribbon cable with header each end. Overall length 60mm.

Glossary of Terms

• "A" channel
A frequency between legally allocated CB channels which may not be used for CB transmissions.

• Actinic
Radiating light which can cause a chemical change.

• Address
The number which will select a specific location in Eprom.

• BCD *"Binary Coded Decimal"*
A binary code in which the usual notations for the HEX numbers "A" to "F" are not used.

• Bit *"Binary digit"*
Represents low voltage or "off' with a zero and high voltage or "on" with a one.

• Byte
A group of eight "Bits" which may be written as a combination of eight ones or zeros or by two Hex digits.

• CE *"Chip Enable"*
The Eprom is on standby until this pin is activated.

• Cleaning … see Erasing.

• Decimal
Normal counting up to ten before a "carry".

• Digital Subtraction
A method of artificially reducing a frequency count by selectively removing some of the pulses.

• Downmix
The creation of a new (lower) frequency by mixing one frequency with another and filtering.

• Eprom *"Erasable, Programmable, Read-Only Memory"*
An IC which can store information in the form of electrical charges.

• Erasing
Changing every single Bit inside an Eprom to "1". That is the same as changing every Byte to 11111111 or FF HEX.

• FET *"Field Effect Transistor"*
A transistor which changes its internal resistance with the voltage applied to its "gate" connection.

• Gang Banger
A programming unit which can copy the contents of one Eprom into several others at a time.

• Ground
In this context, refers to a zero voltage connection.

• HEX
Hexadecimal counting method which allows counting in sixteens, instead of tens, before a carry.

• Hop … see "A" channel.

• Inverter
A device which produces an output voltage which is the opposite to the input voltage, high or low.

• LSB *"Least Significant Bit"*
The Binary Digit which represents the lowest value: by convention, the right-most Bit.

• MSB *"Most Significant Bit"*
The Binary Digit which represents the highest value: by convention, the left-most Bit.

• "N" code
The binary code, on the input pins of a PLL, which determines the internal divide ratio (and VCO frequency).

• OE *"Output Enable"*
All Eprom outputs will be floating until this pin is activated.

• NC *"No Connection"*

- **OTP** ***"One Time Programmable"***
A type of (E)prom which can not be erased.

- **P.C.** ***"Personal Computer"***

- **PD** ***"Phase Detector"***
This PLL pin provides a pulsed output whose average voltage, after filtering and smoothing, controls the frequency of the VCO.

- **PGM** ***"ProGraM"***
A suitable voltage pulse on this Eprom pin will enter into an Eprom the code on its output data pins, according to the location code on its input or address pins.

- **PLL** ***"Phase Locked Loop"***
In this context, refers to an Integrated Circuit which controls the frequency of an oscillator according to the "Nt' code on its inputs.

- **Quartz**
A mineral which is transparent to light and, more importantly, to ultra-violet light.

- **Regulator**
A device or circuit which produces a constant voltage output even if the input voltage varies.

- **Skip …** see "A" channel.

- **Software**
Refers to a microprocessor or computer program, rather than the circuit components themselves.

- **Switch codes**
The binary outputs of a channel switch.

- **UV**
Ultra Violet light which is invisible to the human eye and can be harmful.

- **Vcc**
The IC pin which requires a positive supply voltage.

- **Vdd**
The IC pin which is connected to ground.

- **VCO** "Voltage Controlled Oscillator"

- **Vpp**
The IC pin which must be connected to a specified voltage in order to allow programrning of the device.

- **Zener**
A type of diode which produces a specific voltage when current flows through it in the reverse direction, that is, with the anode more negative than the cathode.