A NEW

MULTIPLE-ACCESS TELECOMMUNICATION

SYSTEM

Thesis submitted in partial requirement for the degree of Doctor of Philosophy
University of Warwick
by

Paul Robert Thomas MSc

September 2000

TABLE OF CONTENTS

CHAPTER 3 (continued)

CHAPTER 6 (continued)

CHAPTER 8 (continued)

LIST OF FIGURES

CHAPTER FIVE (continued)

CHAPTER SIX

CHAPTER SIX (continued)

CHAPTER SEVEN

CHAPTER SEVEN (continued)

CHAPTER SEVEN (continued)

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge the guidance and support given to me by my project supervisors, Dr Steven Chandler (University of Warwick, now Rural Radio Systems) and Professor David Hutchins (University of Warwick). I would also like to thank Professor David Whitehouse (University of Warwick) for his encouragement to get the work started and Professor Roger Green for his constructive comments.

I would also thank my colleagues at Cable and Wireless for putting up with me over the last five years, especially Professor David Ashton (Chief Executive - retired) for allocating me some time to get the work underway, Dr Habib Rashvand (ex-TT & E lecturer) for his helpful comments, Mark Abbott (ex-TT & E technician) for making the surface mount circuit boards and Richard Hallows (ex-Head TT & E) for his support.

Finally, my thanks to Alison for keeping our son Harry occupied, allowing me some time to finish writing up the Thesis.

DECLARATION

This thesis has been written by myself and is based on research undertaken at the University of Warwick between 1995 and 2000. The information recorded herein has not been previously published, except in [18] and [127-130]

All other sources of material are acknowledged and appropriately recorded.

Paul Thomas 09/ 2000

LIST OF SYMBOLS

Symbol	Meaning	Units
g	the phase shift and attenuation occuring between the transmitter and receiver	
$\bar{a}$	mean	
c	velocity of light (in vacuo)	metres per second
f	frequency	Hertz or cycles / second
I	electrical current	Amperes
j	square root of -1	
t	time	seconds
v	velocity	metres per second
V	electrical voltage	volts
W	power	watts
=	equals	
≡	identical	
<	less than	
>	greater than	
Σ	sum	
λ	wavelength	metres
η	correlation ratio	
$\therefore$	therefore	
$\forall$	all of	
$\in$	belongs to the set	
*	convolution	
σ	standard deviation	
[A]	matrix array	

Multiples and sub-multiples used:

m	milli	10^{-3}	K	kilo	10^{+3}
μ	micro	10^{-6}	M	mega	10^{+6}
n	nano	10^{-9}	G	giga	10^{+9}
p	pico	10^{-12}	T	tera	10^{+12}

SUMMARY

This thesis records the research and development work carried out on a new telecommunications multiple-access system called the 'Complex Valued Collaborative Coding Multiple-Access' (CV-CCMA) system. The CV-CCMA system enables more than one transmission to use a communication channel without significant bandwidth expansion. The work has been completed between 1995 and 2000 whilst researching for a (part-time) Doctor of Philosophy degree at the University of Warwick.

The new multiple-access system follows the general trend in communication systems that has been taking place over recent years. The reduction in the cost of processing power (million floating-point operation devices are now only a few pounds), has meant that the complex processing components of communication systems have moved from the analogue to the digital domain. Systems once regarded as too expensive to implement (except by the military) are now commonplace. The main digital processing element used for development of the system is a standard floating-point digital signal processor (DSP). A summary of the algorithm developed and software produced is included in this thesis. New techniques were developed to solve the multi-access resolution problems using the DSP and these are presented.

The fundamental theory underlying the CV-CCMA system is described in detail with simplified examples showing the processes involved. A practical two-transmitter, one-receiver implementation was designed and constructed in order to prove the viability of the system, and details of this work are included in the thesis. Computer programs were written to solve the various linear algebraic equations relating to the operation of the decoding algorithm. Simulation programs enabling parameter variation without circuit construction are also recorded in the thesis. Problems encountered with the new circuitry are discussed and the solutions detailed. This includes a new high-stability oscillator that was designed, built, and tested and which was required in order for the system to function properly. Also described is the high-speed conversion circuitry.

A new sequence of codewords has been developed, helping to reduce the stringent requirement on timing and synchronisation, and results are presented. New methods of synchronisation are discussed together with future development of the system. Projections as to possible uses of the system are also presented.

Finally, there is a short comparative discussion of the CV-CCMA system and the broadband code division multiple-access system, recently on trial in the United Kingdom. This is of specific interest to Cable and Wireless PLC, the main sponsor of the research.

1 INTRODUCTION

As we enter the second century of radio communication the demands on the radio frequency spectrum are becoming ever greater. The range extends from systems operating at very low frequency for world-wide submarine communications, to super high frequency telecommunications systems and direct broadcast satellite services for commercial use. With the advent of all these new services, the electromagnetic communications spectrum is nearing saturation. The usable section of the electromagnetic communications spectrum is bounded; therefore new techniques and systems are required to enable further expansion of services. A number of techniques have been developed which enable multiple-access. This requires the sharing of the communication bandwidth resource, giving increases in the efficiency of usage of the available spectrum. Although there are theoretical limits on the amount of 'information' that can be communicated over a given bandwidth, the new techniques aim to optimise the information transfer, and reduce interference between multiple users of the same bandwidth. At the present time there are many different contenders for this already large, and potentially very large, market. However, the circuitry required to enable most of these systems to operate is fairly complicated and requires a substantial amount of processing power in order to function. The cost of this processing power has decreased by orders of magnitude in recent years, and this has given additional impetus to the commercial exploitation of multiple - access systems.

This thesis records the research, design and development of one such system, called the 'Complex Valued Collaborative Coding Multiple-Access' system (CV-CCMA system). The system is demonstrated operating with a single floating-point digital signal processor (DSP), which, together with radio frequency

components designed for the global system mobile (GSM) system, produces a very cost-effective multiple- access telecommunication system.

The main objective of the research was not just to produce further simulations of the multiple-access system, varying either a transmission parameter or some component part of the coding or decoding process, but also to generate a hardware demonstrator that would identify any problems which might have been neglected in simulations. This path was followed because of problems experienced in communications research in other fields. A lot of simulation-based communication research is limited in use due to the many additional problems that become apparent only when the systems are implemented in hardware. This is particularly true in the telecommunications industry, which uses a lot of relatively old equipment. For example, Cable and Wireless PLC, the main sponsor of this research, is now over one and a quarter centuries old. This means that there are systems installed that are quite ancient, usually costing substantial sums to maintain, and a lot of the newer technologies need to be able to interface with, and often work alongside, these older systems. This is the area where many problems have arisen, for example in the implementation of newer statistical trunk systems with installed trunk transmission plant. Loading/capacity simulations have shown that the systems should work perfectly, but in practice the systems do not actually work, requiring further investment to get them working. Similar problems have been encountered with radio systems and so it was decided at an early stage to 'prove' the theory with some actual hardware, and to progress the research by implementation of any new techniques before advancing the whole system. This meant that the research tended to progress in bursts, but as the research was (is) being undertaken on a part-time basis the available

time could be utilised effectively. Practical demonstrations were found to be essential, to verify nothing had been forgotten about in the simulations.

1.1 History

It has been over a century since Alexander Graham Bell transferred 'voice-shaped currents' over copper wire, thus starting the telecommunications revolution. Early telephone systems were directly connected to an exchange and then calls were routed manually by switchboard operators. The operator's attention was gained by winding a handle attached to the telephone, connected to a magneto and generating a large voltage which rang a bell in the exchange. A similar system is still used today by dealer systems in the financial sector, called generator-generator signalling, and also for military field telephone systems. In an effort to stop illicit eavesdropping, an undertaker in the United States, named Almon B. Strowger, invented an automatic telephone exchange, which comprised electromagnetic relays linked to a dial on the telephone, automatically connecting subscribers together. These systems were in use until the 1970s in the United Kingdom. The next major advance came with the invention of the transistor by Brittain, Bardeen and Shockley at Bell Laboratories in the United States, designed for telephone switching and able to switch quite large currents very rapidly, making the slow and unreliable electromagnetic relays obsolete. Major advances have followed at an exponential rate and developments in other fields have been used in the telecommunications industry to gain competitive advantage. For example, field effect transistors, developed by NASA for the Apollo space program in the 1960s are now widely used in telecommunications systems. Virtually from the beginning of electronic communication, over a century ago, the demand for services has outstripped the available resources, mainly bandwidth, and techniques have been

continually developed to increase the capacity and reliability of telecommunications systems. Table 1.1 lists a few milestones with particular relevance to this thesis.

In the last twenty years the telecommunications industry has seen a number of techniques that enable transmission and reception by a number of users, all using the same transmission resource. Trunk telecommunications traffic is either carried over a fibre-optic cable (usually called a bearer) or over a radio system. The radio is usually a back-up to the fibre. However, the local loop is often still a twisted pair of copper cables, with all the associated problems of interference and limited bandwidth. There are a number of standard techniques to reduce these problems, enabling the transmission of data, at reasonable rates, over twisted pairs. Modern telecommunications networks are now completely digital, leading to simple interfacing with current high-speed digital circuitry and associated cheap processing power. The CV-CCMA system is a completely digital system for multiple access that will interface easily with current telecommunications systems.

The last few years have also seen unprecedented growth in data services, fuelled mainly by the explosive growth of low-cost Internet traffic, electronic commerce, etc. This trend shows no sign of abating; if anything the rate of increase is such that within a couple of years, voice traffic will be consigned to a small proportion of bandwidth carried on the back of data traffic. A new, low-cost, radio-based multiple-access system could be used for the implementation of these new services.

Today, in most of the developed world there is a substantial telecommunications infrastructure, which has cost thousands of millions of pounds to install and millions of pounds per annum to maintain. In contrast, the developing world has very little infrastructure, especially in the more rural areas. It will require a

Year	Event
1834	Carl F. Gauss and Ernst H. Weber build the electric telegraph.
1838	William F. Cooke and Sir Charles Wheatstone build the telegraph.
1844	Morse demonstates the Baltimore and Washington telegraph line.
1846	M. Faraday's 'Thoughts on ray vibrations' form basis for e-m field theory.
1858	The first transatlantic cable is laid, and fails after 26 days.
1864	James C. Maxwell predicts electromagnetic radiation.
1876	Alexander Graham Bell develops and patents the telephone.
1883	Thomas A. Edison discovers the flow of electrons in a vacuum, called the 'Edison effect' providing the foundation of the electron tube.
1885	Edward Branly invents the 'coherer' radio-wave detector.
1887	Heinrich Hertz verifies Maxwell's theory.
1894	Oliver Lodge demonstrates wireless communication over 150 yards.
1897	Guglielmo Marconi patents a complete wireless telegraph system.
1900	Guglielmo Marconi transmits the first transatlantic wireless signal.
1904	John A. Fleming invents the vacuum tube diode.
1918	Edwin H. Armstrong invents the superheterodyne receiver circuit.
1927	H. Black develops the negative-feedback amplifier at Bell Laboratories.
1933	Edwin H. Armstrong invents frequency modulation.
1935	Robert A. Watson-Watt develops the first practical radar.
1937	Alex Reeves conceives pulse code modulation (PCM).
1947	Walter H. Brittain, John Bardeen, and William Shockley devise the semiconductor transistor at Bell Laboratories.
1948	Claude E. Shannon publishes his work on information theory.
1950	Time-division multiplexing is applied to telephony.
1950s	Microwave telephone and communication links are developed.
1953	The first 'multiple-access' transatlantic cable (36 channels) is laid.
1960	Theodore H. Marman produces the first successful laser.
1963-66	Error-correction codes and adaptive equalization for high-speed error-free digital communications are developed.
1965	The first commercial comms. satellite, 'Early Bird', is placed into service.
1966	Kao and Hockham publish 'The principles of fiber optic communications'.
1971	Intel Corporation develops the first single-chip microprocessor, the 4004.
1976	Personal computers are developed. X25 Packet delivery standardised.
1979	64-Kbit random access memory launched.
1981	IBM PC is introduced. De-nationalisation of United Kingdom Post Office. Cable and Wireless partial privatisation. Falcon (Mercury) launched by Cable and Wireless as a competitor to British Telecom.
1984	First generation cellular telephones introduced to UK (Cellnet + Vodafone, BT - Securicor and Racal Electronics).
1988	Global system mobile launched in Europe.
1990s	Texas DSP chips increase processing power by orders of magnitude.
1990s	Explosive growth in Internet usage alters dynamics of telecom. networks. Mobile penetration is greater than fixed line in some European states.

Table 1.1 Important dates in electronic communications

major investment program to cable them up. For example, it cost Cable and Wireless PLC over one hundred million pounds to put a basic digital trunk network into Latvia, a small country with a fair amount of infrastructure already in place, and with most of the population concentrated in one city. This investment will take many years to repay. This is another area in which a cheap multi-access radio system, such as CV-CCMA, will find a ready market, specifically for high density / low-cost access, and low-density / low-cost access. This thesis will describe the possible implementation of the CV-CCMA system on existing low-cost rural access radio systems; this work is briefly covered in Chapter 8.

1.2 Techniques

It is standard practice in telecommunication networks for the electrical representation of speech to be first digitised by the process of pulse code modulation (PCM), a technique invented by Alex Reeves in 1937 [1] whilst working in Paris at the IT & T laboratory and patented by them in 1938. Some of the technology was used in the early radar systems to detect aircraft. After the war the process was further developed by the AT & T laboratories in the United States. However, the technique only became popular commercially with the advent of cheap semiconductor circuitry. The International Telecommunications Union has standardised the process of PCM with a series of recommendations, G701, G703, etc. [2], which have been adopted on a world-wide basis. In the G series of recommendations the process is defined briefly; the speech waveform is sampled at 8000 per second, and to comply with the sampling theorem the speech waveform is band-limited to 300–3400 Hz. Each sample is then converted to an 8-bit binary code. The code consists of a sign bit and 7 amplitude bits; the output is quantised to the nearest direct conversion value - there is no dither. The

conversion is not direct; it is non-linear or logarithmic. The audio input is assumed to be standard speech so more 'bits' are assigned to the lower-level signal amplitudes than to the higher ones. The process approximates the output / input transfer function to a logarithmic law and two world-wide standards have evolved, the 'μ' law in the United States and the 'A' law in the rest of the world (conversion circuitry is needed between the two standards). There are 8 bits output for each sample taken 8000 times per second, giving a digital output rate of 64 Kb/s, or 32 Kb/s with Adaptive PCM, an adaptive predictive technique defined in G732.

These digitised streams of audio information are then combined together with other digitised traffic, data or encoded video, by a process called time division multiplexing (TDM). Each digital input is buffered and assigned a period in the time domain called a time slot. The time division multiplexer then combines these inputs and outputs them, at a higher rate, together with synchronisation and signalling information. Figure 1.1 illustrates the concept of TDM and Figure 1.2 shows a standard primary rate 30 channel (2 Mb/s or S-bearer) TDM in operation, used throughout the world as the first level of multiplexing. The multiplexer's output aggregates are then combined with other multiplexer outputs, giving (currently) defined data rates up to about 10 Gb/s. The TDM process is now a mature technology, with international standards set by the International Telecommunications Union enabling the world-wide inter-connection of systems operating to these standards.

The main problems arise with synchronisation and management of the links. If synchronisation cannot be guaranteed then buffers must be used at the interfaces which causes delay problems, particularly with speech. The other problem with TDM is that the system does not allow for instant access; the transmitter must wait until its assigned time slot occurs before data is transmitted and this is wasteful

of transmission time, requiring buffers, and inefficient. Speech is time-critical, and so it becomes inefficient to assign a time slot, whether it is required or not. Data tends to be less time-critical, so queuing can be successfully applied. Recently, adaptive or statistical TDM systems have become popular, for example asynchronous transfer mode (ATM), especially with the larger telecommunications commercial operators. These systems work by dynamically assigning bandwidth on the trunk only when it is required, giving much higher efficiencies. Statistical systems are also more flexible. They can be used to implement quality of service parameters, required for today's multimedia systems. Most importantly, time-critical signals (speech and video) can be combined with non-time-critical signals (data) and passed over the same trunk network. The carrier's network can always be fully loaded, which means the carrier generates as much revenue as possible. As well as the benefits of flexibility the advantage to the user is mostly in cost reduction, rather than any fundamental improvement to quality of service.

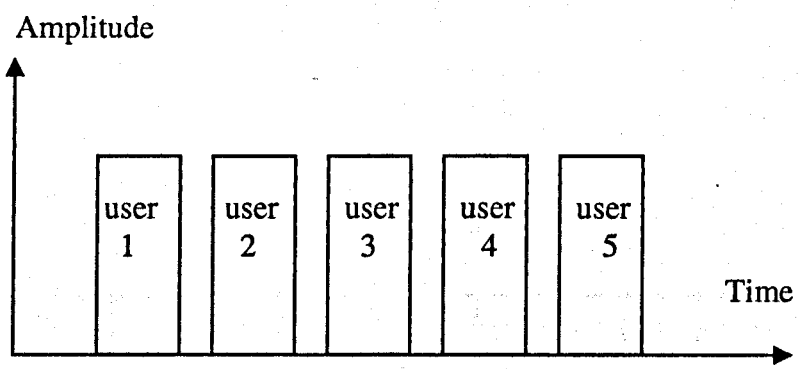


Figure 1.1 Time division multiplexing

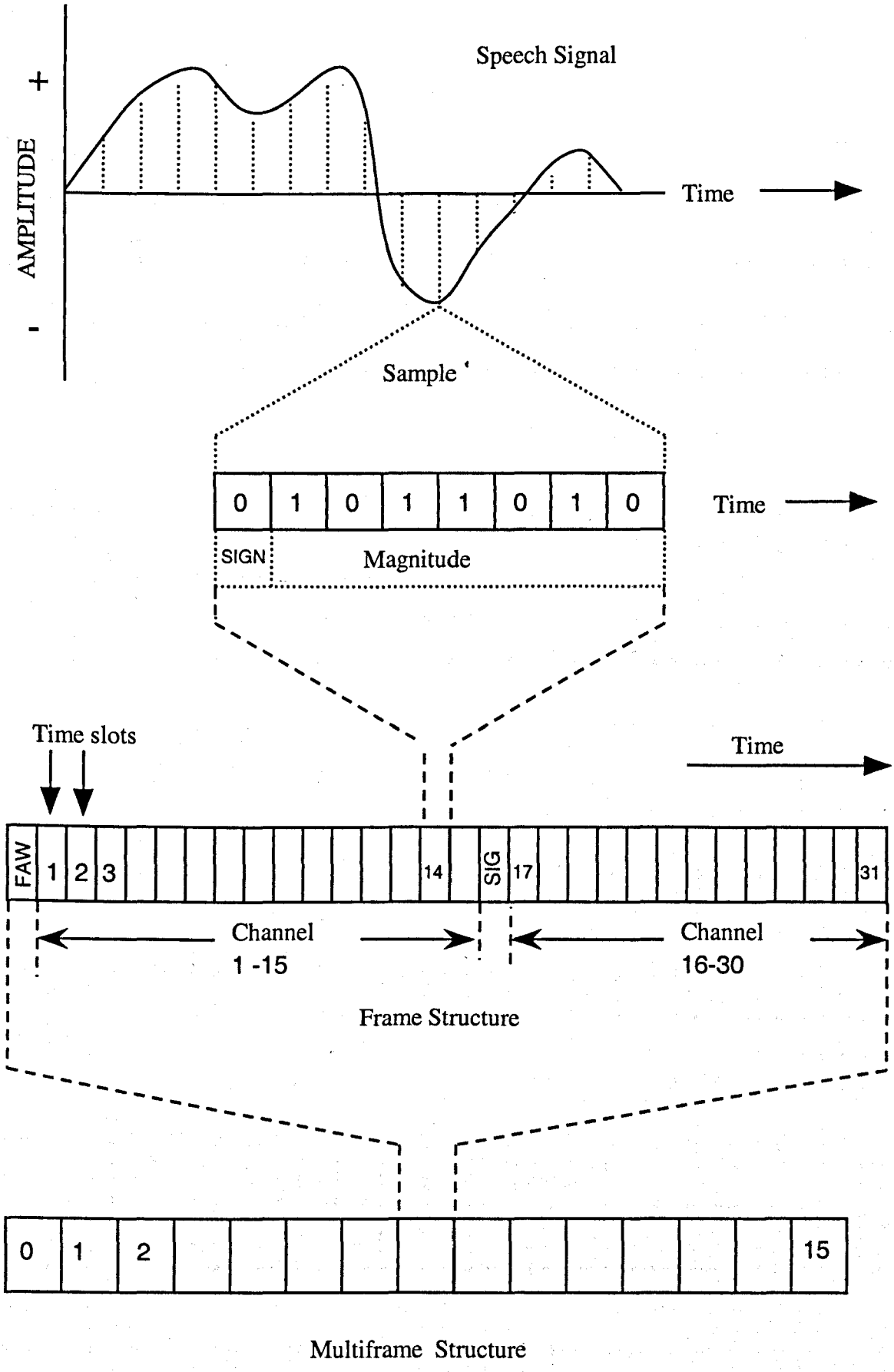


Figure 1.2 Primary rate time division multiplexing

The aggregate signals are then converted using frequency conversion techniques and combined in the frequency domain (FDM – frequency division multiplex or WDM – wavelength division multiplex); the translation may be up or down. The input signals are assigned to a band of frequencies and passed over the transmission link. In the case of the fibre-optic trunk cable transmission, these wavelengths are limited to the precisely defined boundaries of the low-loss window, and the line width of the laser transmitters and photodiode receivers. These techniques are generically called dense wavelength division multiplexing (DWDM), and current operating practice gives a maximum of twenty carriers each operating at data rates of 2.5 Gb/s (STM-16 rate with adaptation).

Current radio systems operate at maximum data rates of 140 Mb/s within various frequency bands up to 50 GHz. Figure 1.3 illustrates the concept of frequency division multiplexing and Figure 1.4 illustrates the AT & T standard FDM hierarchy, still in use throughout the world, although now declining in popularity.

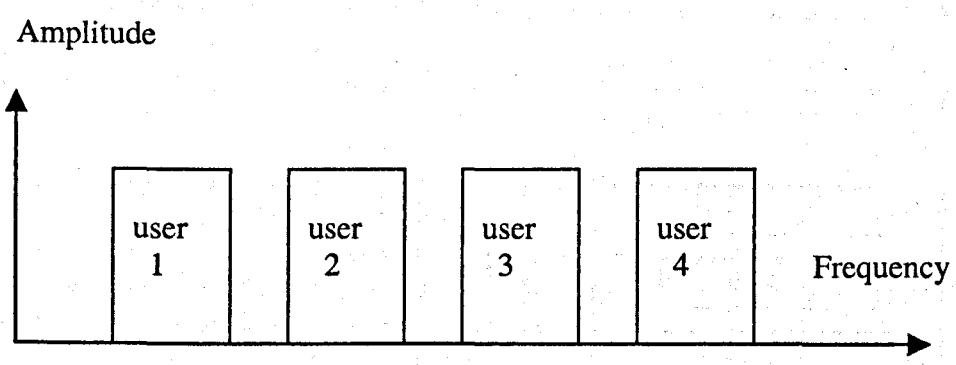


Figure 1.3 Frequency division multiplexing

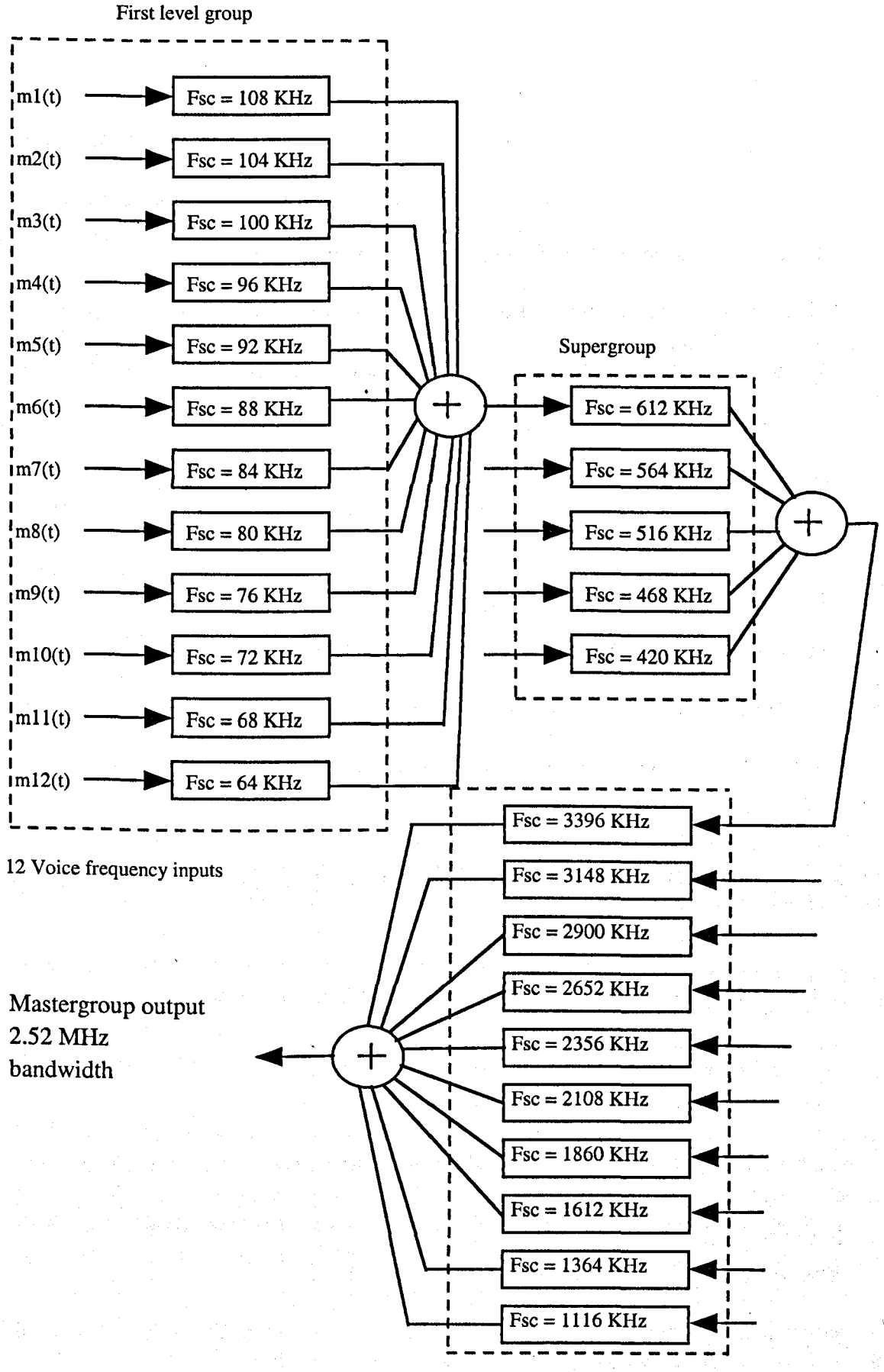


Figure 1.4 AT & T Frequency division multiplexing hierarchy

A third technique for multiple-access is called code division multiple-access (CDMA). There are two main types which mostly follow generic algorithms. The first type is called frequency hopping (FH) CDMA and is a system that modulates the signal onto carriers whose frequency is selected from a sequence or randomly. Slow FH is where the hopping rate is less than the information rate and fast FH is where multiple hops are used for each information bit. Bit interleaving is used as well as error-correcting codes to overcome some of the fading problems inherent to the system. These frequency-hopping systems are mainly used by the military and users requiring high levels of security.

The second type is called direct sequence (DS) and in this system the individual users are assigned sequences that are separated from one another by the use of a set of nearly orthogonal waveforms or sequences. These are often referred to as spreading codes because they cause the output bandwidth to spread. Due to the nearly orthogonal nature of the sequence when they are combined together, assuming they are properly synchronised and with no errors, the cross-correlation between different users will be nearly zero.

A number of different transmitters can then be overlaid using the same bandwidth providing they are using distinct sequences. The receiver multiplies the composite signal by the same orthogonal sequence as the required transmitter, producing an output which is integrated over a bit period to give the recovered data. Note that the multiplication of the signal by the orthogonal sequences produces an increase in bandwidth, and drop in efficiency: the signal is 'spread'. Figure 1.5 shows a simplified representation of code division multiple-access, and Figure 1.6 shows a block diagram of a typical spread spectrum system. The technique is derived from a military system used for secure communications [3]. Note that it is very difficult to

block a signal that has a wide bandwidth, without using large amounts of power. CDMA is more complex than the other techniques described for multiple-access and has a number of unique problems associated with it. Some of these problems are described in Chapter 3. The near–far problem, in which strong DS signals swamp weaker DS signals, requires quite complicated power-control circuitry to be implemented in DSCDMA systems. Also the coherence bandwidth must be less than the spread bandwidth, and this may be 10 MHz for small-room operation, and far in excess of this for satellite operation.

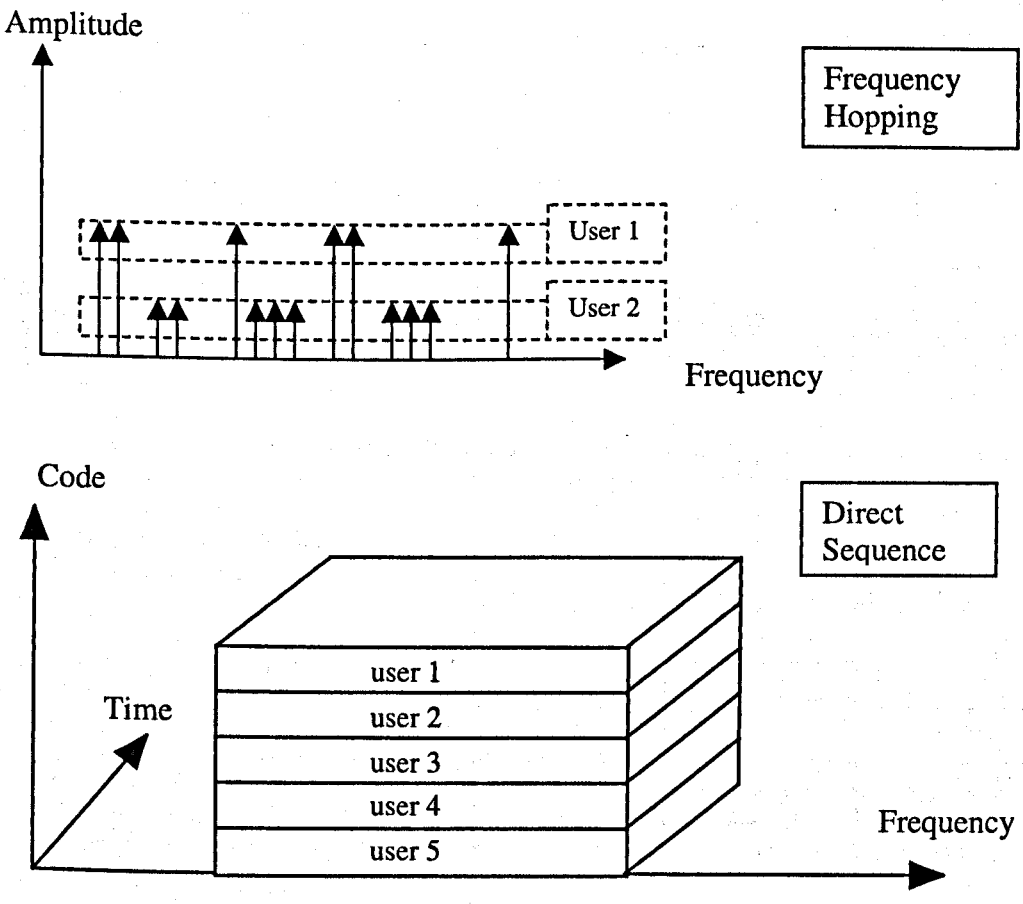
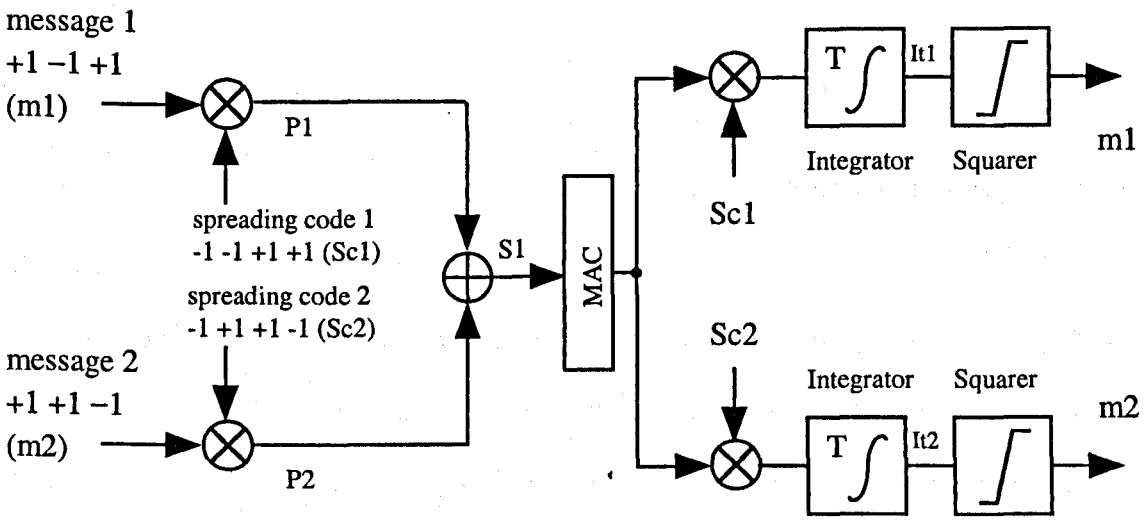


Figure 1.5 Code division multiplexing



Note : Sc1 and Sc2 are orthogonal, ie the cross-correlation is zero:

$$(-1)(-1)+(-1)(+1)+(+1)(+1)+(+1)(-1) = 0$$

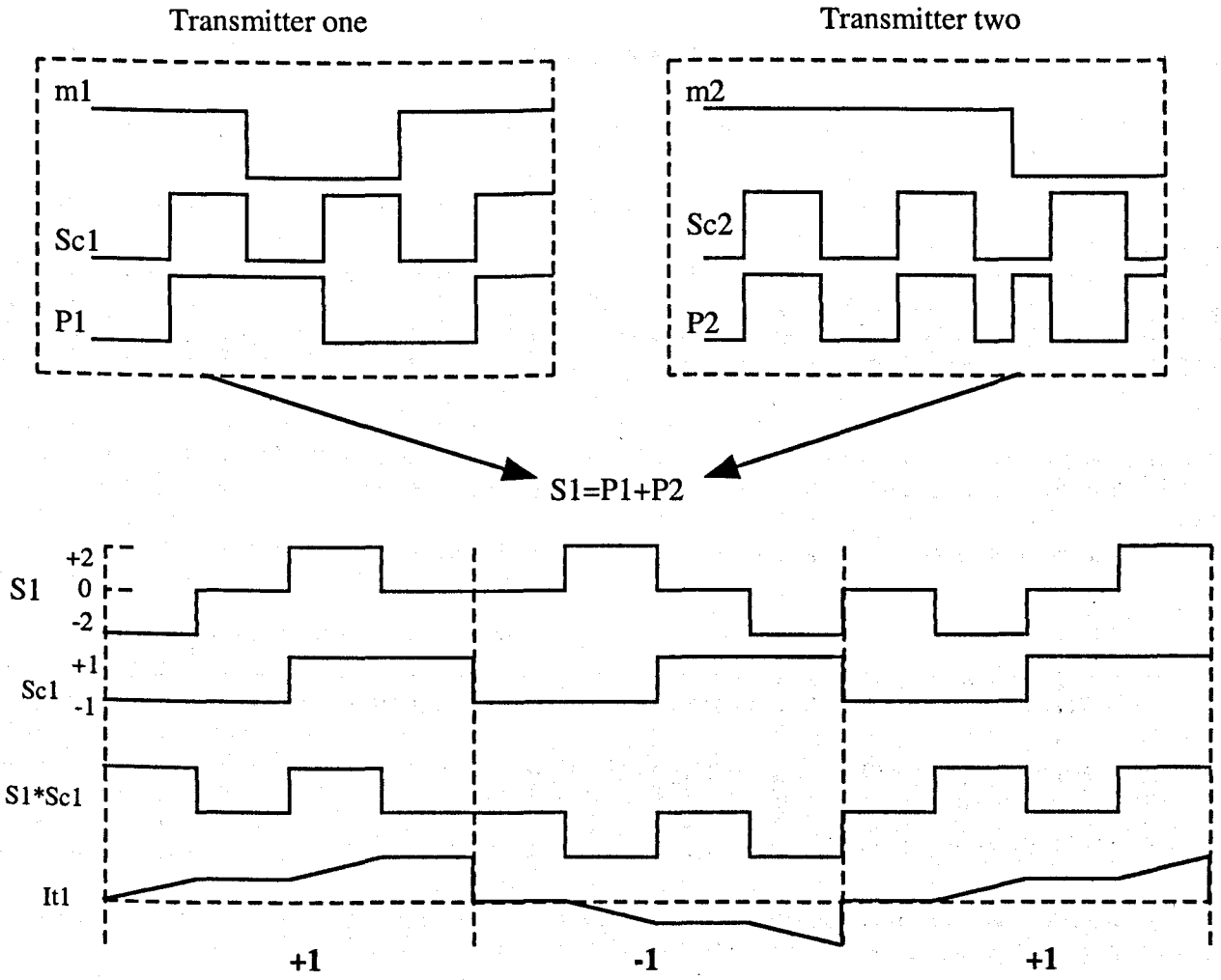


Figure 1.6 Code division multiple-access

A more recently developed multiple-access system, which is popular in the local area network (LAN), is the collision detection carrier sense multiple-access (CDCSMA) method. This is a variation of the time division multiplex (TDM) systems already described, generically called statistical TDM. In this system the access channel is sensed for activity and the transmitter output is buffered until a gap appears in bus activity when it can transmit its data. Weaknesses of this system are the lack of instant access, particularly when the network is busy, and the complexity of the sense circuitry. However, the system is cheap and is now very popular.

Another technique of multiple access is the space diversity multi-access (SDMA) system, mainly used with radio or optical systems. In these systems the network is subdivided into geographical areas, thus ensuring the transmitters and receivers operating on the same frequencies are always physically more than a certain distance apart. This distance is sufficient to stop interference between the systems because the attenuation on the signal drops it below the interfering threshold value. It is usual to employ different polarisation in the signals in an effort to reduce further any interference. This method is used in cellular radio systems with the same frequencies re-used in cells, but only when they are a minimum distance apart. Figure 1.7 illustrates the concept of SDMA. There are many problems with this simple system as radio propagation is very variable and the signals may propagate substantially further than intended, under certain atmospheric conditions. This causes problems with multiple cells both being accessed and the possibility of dropped calls. Another problem is that, at the higher frequencies, the waves propagate along overhead lines (power / rail feed, etc.), which act as waveguides and may lock into or interfere with a cell a considerable distance away from the transmitter. The network manager then has to decide which is the valid signal, and which is the ghost signal.

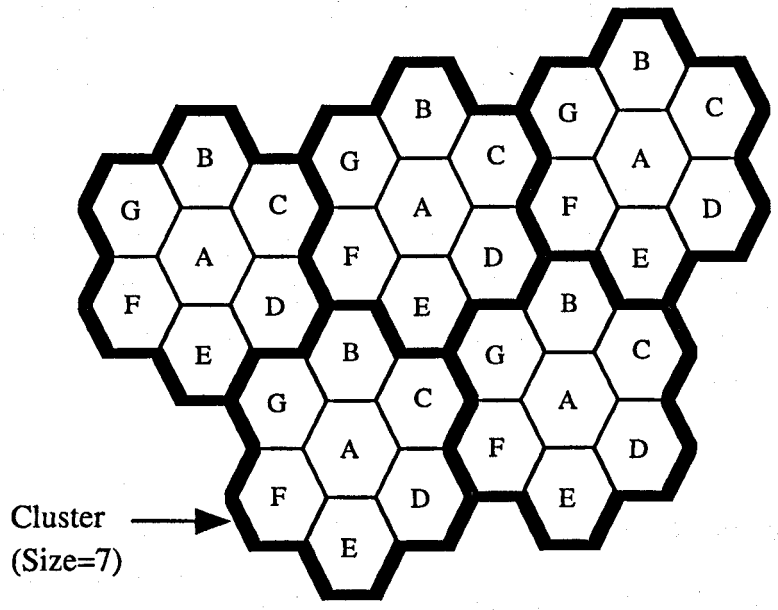


Figure 1.7 Space diversity multiple-access

Figure 1.7 illustrates a typical cellular radio network layout. The cells are drawn as hexagons and the letters A-G represent the group of frequencies that are used within a cell. Note that coverage area is only approximated by a hexagon. At the edge of coverage, where geographical features cause problems, directional aerials are used to prevent interference. The system is easily combined with one (or more) of the other multiple-access techniques already discussed. Note also that Figure 1.7 shows a frequency re-use factor of 7 and that there are always at least 2 cell diameters between cells that are using the same frequencies.

All of the systems described thus far combine the individual signals in a common medium, usually silica glass-fibre, copper co-axial cable or the radio spectrum ether. A substantial amount of work has been done over recent years

producing mathematical models of various different types of channel. The model used in this thesis is the adder channel with discrete input and output [4-12]. The other channel models used are detailed in Chapter 2, the research survey section of this thesis. A lot of the work in these papers is related to the theoretical maximum data rates that the multiple-access channels can accommodate. The individual transmitters and receivers are treated as independent units, which in the case of the implemented CV-CCMA system is not strictly true, as the components of the system need to be synchronised in order to function. For further discussion on the channel, including the effect of feedback and interference, see Chapter 2. Some work was done on a performance comparison between CV-CCMA and CDMA. This used a rather more complicated model of a mobile radio channel with different types of noise and fade / synchronisation loss (from [3]); see Chapter 3 for details.

To summarise, there are three other multiplex technologies, FDM, CDM and TDM. The fact that only a subset of a number of users wish to communicate at a time enables sharing of a channel between them; there is no need for each user to be allocated a permanent bandwidth. The larger the number of users that can share resources the larger the statistical gain possible. TDM often uses a multi-access protocol to partition time slots dynamically, e.g. CDCSMA.

CDMA uses near orthogonal sequences and so long as no more than a number proportional to the number of orthogonal channels are in use at any one time, error-free transmission is possible. This makes CDMA ideal for statistical multiplexing. FDM cannot easily be made to handle dynamically varying transmission rate requirements, with each user allocated its own frequency channel.

All of the multiple-access systems described are limited in 'information' transfer ability by the theorem of Shannon, which sets an upper limit on

all the systems. The relative advantages and disadvantages of the different systems lie in the ease of use, or otherwise, when used as part of a complete multiple-access system. Within the telecommunications field, the ability to interface with existing (legacy) systems is probably one of the most important considerations.

1.3 The CV-CCMA system

In 1994 work was started at Warwick investigating new multi-access techniques with specific application to the telecommunications industry, particularly satellite communications and digital mobile telephones. These are two very important areas of interest to the candidates employer, Cable and Wireless PLC. It was decided that the 'Complex Valued Collaborative Coding Multiple-Access' (CV-CCMA) system offered the most promise for development as a new multi-access radio technique. The work was developed from discussions with Dr Steve Chandler at Warwick, which indicated that ordinary collaborative coding techniques were impractical for use over a radio channel.

It was decided to develop the CV-CCMA system as a digital radio system using digital signal processing techniques with spatial diversity. This would enable simultaneous reception of a number of superimposed signals within the same bandwidth and without the use of spreading. The system can increase the capacity of telecommunication systems, particularly satellite and digital mobile telephones, providing significant performance improvements over other multiplexing and multiple-access techniques.

CV-CCMA is inspired by the technique of collaborative coding, which, though theoretically interesting, was not practicable for use on radio systems because of the variability of gain and phase of each of the contributing transmissions.

Collaborative coding is based on the superposition of two signals from different users, producing a combined signal which can be decoded to recover the original signal transmissions. The system is best explained with the aid of a simple example. Assume we have two transmitters and one receiver with the transmitters programmed to output codewords as below.

Transmitter 1:

'0' = 0 0

'1' = 1 1 ('Data input' = codeword output)

Transmitter 2:

'0' = 0 0

'1' = 0 1 ('Data input' = codeword output)

The codewords can be modulated by superimposing onto a radio frequency or optical frequency carrier as a varying amplitude level. Note that the combining channel is assumed to be a simple adder channel, that is each component bit is added to the corresponding bit from the other transmitter. The system is assumed to have ideal synchronisation, so both bit and frame timing are ideal. The only possible combinations are tabulated in Table 1.2 .

			User 1 data	User 1 data
			0	1
		Codewords 'generated'	0 0	1 1
User 2 data	0	0 0	**0 0**	**1 1**
User 2 data	1	0 1	**0 1**	**1 2**

Table 1.2 Collaborative coding - example 1

The bold figures indicate the allowed codeword combinations; thus, if we receive any one of the four combinations the original transmitted data can be recovered as it is a unique solution to the addition of the original collaborative codes. Also of note is the fact that, even with the simple two-user example described, the channel is loaded at an efficiency of 1.0 bit per baud, as each user runs at 0.5 bit/baud giving 1.0 when added. Table 1.3 shows how this can be extended to obtain better efficiencies.

			User 1 data	User 1 data
			0	1
		Codewords 'generated'	0 0	1 1
User 2 data	a	0 0	**0 0**	**1 1**
User 2 data	b	0 1	**0 1**	**1 2**
User 2 data	c	1 0	**1 0**	**2 1**

Table 1.3 Collaborative coding - example 2

Note that all the states (in bold) are still uniquely decodeable. The codewords shown give an efficiency of approximately 1.3 bits per baud. Work has been carried out on the efficiency of collaborative codes [13] setting out the definitions for capacity boundaries on these efficiency measures.

The system is quite simple and elegant, requiring accurate timing and an adder channel between the transmitters and receivers. However, in the case of radio transmission, there will not be an ideal adder channel between the transmitters and receivers, but an attenuation and a phase shift (called the $\boldsymbol{g}$ value). This $\boldsymbol{g}$ value may be variable, particularly if the systems are mobile. This means that in order to resolve the combined signals an accurate timing relationship between the transmitters and receivers is required. The receivers will be pre-programmed with all the transmitted codewords. Then, using the processing power of the digital signal processor, the original transmissions can be resolved. The receiver can resolve the phase shift, the attenuation and the transmitted codewords simultaneously. This now becomes a CV-CCMA system. The operation of the system is covered in detail in Chapter 5.

Following the logic of the argument, even if some of the combined codewords are theoretically identical, provided they are received over different paths involving different attenuations and phase shifts, then the original codewords can be reconstructed. This is because the signals are multiplied by the attenuation and phase shift, and they are different for each path. This is the principle of the first of the CV-CCMA systems to be designed and built at Warwick.

Previous research, (into radio multiple-access), has produced a number of proposed techniques designed to resolve the individual transmissions of a multiple-access system. In [14] the authors proposed enhancements to a Collaborative Coding

Multiple Access (CCMA) system. The receiver incorporated an estimation technique based on the gradient descent algorithm [15,16]. This, together with a predictor and corrector, was proposed as a receiver, with the capability of resolving multiple signals within the same bandwidth. The CV-CCMA system differs from this serial approach in that the $\boldsymbol{g}$ values are simultaneously estimated with the data values.

The basic algorithm of CV-CCMA is sufficiently computationally efficient to enable implementation of a system, with realistic data rates, using a standard floating-point digital signal processor. This is a major advance because the previously proposed advanced systems for co-channel interference suppression or simultaneous demodulation of superimposed signals have required large and expensive amounts of processing.

The CV-CCMA system does not separate the individual inputs to the MAC (multiple-access channel) in terms of frequency or time but assigns unique collaborative codes to each user. These codewords are unique to each transmitter and in the first practical implementation are all linearly independent [17]. The pre-calculated combined matrices, necessary for correct receiver operation, are stored in the receiver memory. The receive process then proceeds by calculating the minimum Euclidean distance between the received vector and the pre-calculated codeword combination. The recovered data is then output using a simple look-up table. The system is simpler in implementation than CDMA and without the need to 'spread' the output signal. The two systems are compared later in the thesis but there are advantages and disadvantages to each. CDMA is substantially more complicated in implementation but both systems require accurate timing in order to work.

The requirement for linearly independent codewords to be used at each transmitter restricts the range of available codewords so work was carried out

investigating this problem [18] at Warwick. It was shown that for unique decodability a less restrictive set of rules could be implemented. Details of this work are discussed and included later in this thesis. The simulation programs that were written, and the actual hardware constructed, were designed for ease of experimentation with codeword variations. Figure 1.8 illustrates the system concepts in a block diagram of a basic CV-CCMA system.

This introduction has described in simple terms some of the systems and techniques the CV-CCMA has been designed to improve upon, and work with, in the modern digital telecommunications industry. The system has known limitations, for instance the fact that perfect timing is assumed and the problems of multipath, which have yet to be addressed; the latter could be resolved but this would substantially increase the complexity of the system, as is the case with most radio systems. The issues raised in this introduction are discussed and expanded upon in later chapters. The thesis also records the work done to demonstrate the viability of the system. If the system were not viable there would be no point in solving the other issues. The thesis also develops some of the ideas generated during the research. For example, once the $\boldsymbol{g}$ values are determined they can be used to decode the multiple transmissions, as they are unique to each path. This removes the need for codes, timeslots or frequencies dedicated to each user.

The next chapter records a survey of recent research into the multiple-access and related fields.

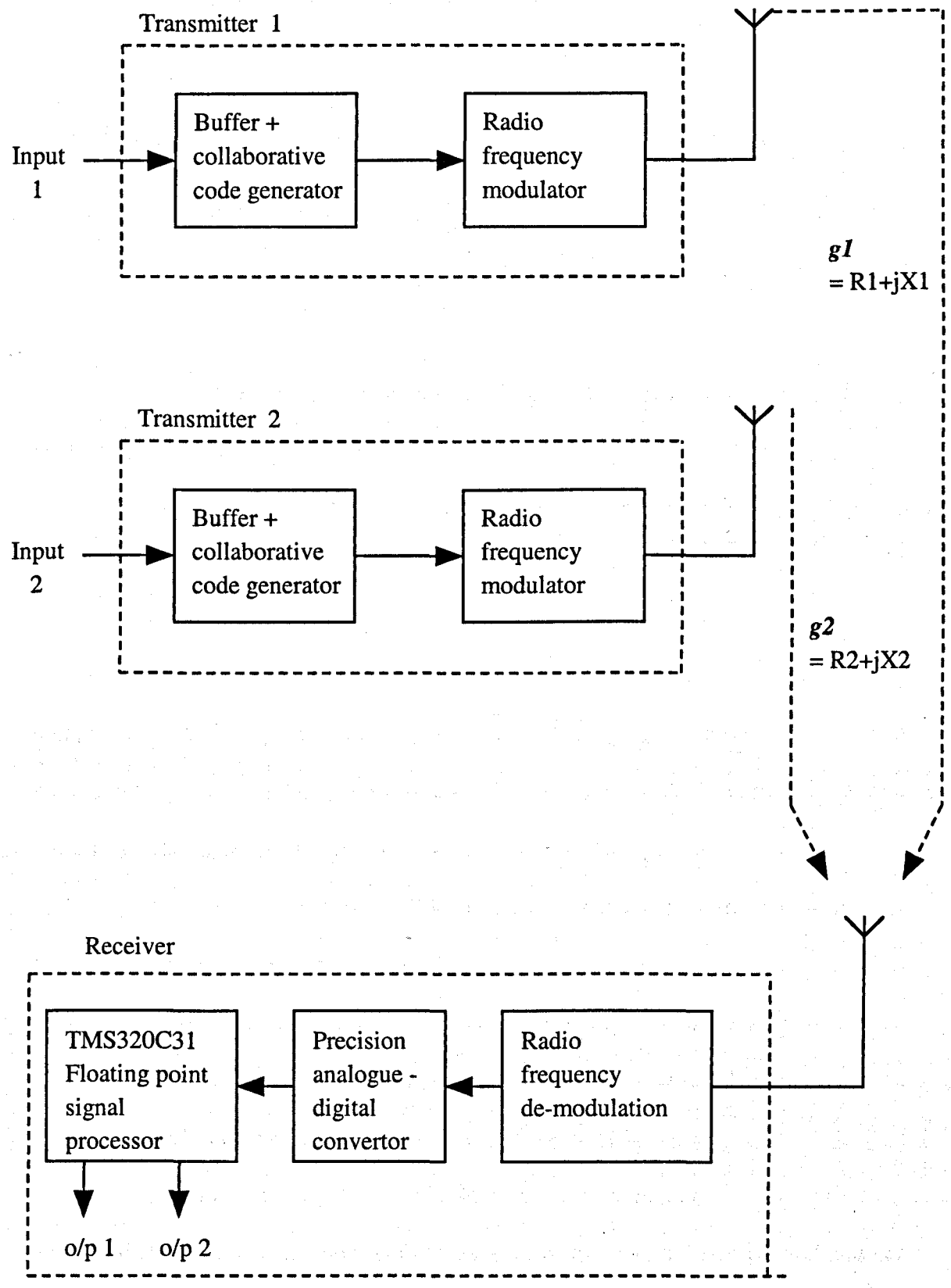


Figure 1.8 Complex valued collaborative coding multiple-access system

2 SURVEY OF RECENT RESEARCH

2.1 Introduction

The introductory chapter to this thesis gave a flavour of the different systems currently in use in the multiple-access telecommunications area. Whilst researching the thesis it became apparent that different people / companies and institutions interpret the term 'multiple-access' in different ways.

The more theoretical research encompasses the mathematical foundations of the multiple-access channel. This area is quite active and has produced a number of interesting papers, usually based on fundamental information theory papers, which have been adapted and advanced into the digital age. In the telecommunications industry the term multiple-access is common to a number of different systems: time division multiple-access - used in trunk multiplexing equipment; frequency division multiple-access - used mainly in satellite communications and high-capacity fibre-optic systems; code division multiple-access - used in mobile telephone systems; space division multiple-access – also used in mobile systems and carrier sense/collision detect systems which are mainly used in local area networks. There are fundamental restrictions to all the systems and each has advantages and disadvantages over the others. The more commercially orientated research mostly pushes the already established advantages and disadvantages of each system forward in the same direction.

In the military area, multiple-access communications are a 'hot' topic, but it is difficult to investigate this field, as a lot of the material is restricted and not available to the general public. However, useful research was done by investigating the military equipment manufacturers' data, particularly from the United States.

The initial research carried out revealed a vast amount of information in the multiple-access field. This has been filtered down to material specifically of interest and relevant to the work recorded in this thesis.

2.2 Information sources

The Institution of Electrical Engineers (IEE) [19] and the Institute of Electrical and Electronic Engineers (IEEE) [20] have proved to be a source of much material. The IEEE runs a communications society [21] and publishes *Communications Monthly*, containing interesting and up-to-date material. The IEE publishes *Electronic Letters* [22] and *Proceedings- Communications* [23] also containing much information. The most productive searching was done manually, searching the 'Inspec' documents of key abstracts in telecommunications research; these are produced monthly by the IEE and contain short abstracts of current research publications. The electronic search engines tended to miss information. One problem with both institutions has been a financial one. The cost of material is cheaper when requested from an academic institution than if your request originates from a commercial organisation.

The University of Bath database BIDS [24] was found to be very useful as were the Cable and Wireless [25] and British Telecom (BT) [26] databases of current research and state-of-the-art material references. The BT technical journals were also quite enlightening [27]. University libraries at Cambridge, Southampton, Sussex and Lancaster were found to contain useful research material, as well as Warwick.

A number of companies sponsor research into multiple-access techniques, and these companies publish some very useful material. The Qualcomm [28] web site contained detailed work on CDMA and equations developed for

network-loading calculations that were used in the later comparison work. The American company, Hewlett-Packard [29], produces a number of technical briefs and technical references, which can be ordered over the Internet, although many are not published in an electronic format. These were very useful in the radio frequency design stages, especially when problems were encountered with the modulators. The Dutch company, Philips [30], produces a wealth of literature relating to their semiconductors and microcontrollers, most of which is accessible electronically over the web. Philips sponsors a centre of excellence at Warwick and generously supplied a lot of equipment and reference material, particularly useful with the design and implementation of the synthesiser, control circuitry and frequency-conversion stages of the transmitters and receivers. The American company, Texas Instruments (TI), produces the digital signal processors that are used in the system. There are two main web sites and a data base, which can be accessed via [31], containing a lot of information that was very valuable when trying to get the various routines working. The company sponsors certain universities and Warwick was enrolled into the European scheme. This meant hardware and software could be bought at substantial reductions. Analogue Devices [32], Harris Electronics [33] and National Semiconductor [34] all had useful websites and integrated circuits were used from these manufacturers on the strength of the data placed at these sites.

2.3 Channel models

Most of the work recorded in this thesis treats the information-carrying channel as an adder channel with discrete input and output. This provides the closest model to the actual situation in the CV-CCMA demonstrator. In the demonstrator the channel has attenuation and phase shift; hence the relationships are complex. The other models

used to model the multiple-access channel are: logical OR [35-37]; logical XOR [38]; logical AND [39-40]; collision [41-43]; and switching [44].

2.4 Multiple-access information theory

A good starting point when researching information theory is the seminal paper written by Claude Shannon in 1948, 'A mathematical theory of communication' [45]. This paper derived the formula for the capacity of a channel (C bits/sec), with additive white Gaussian noise power (N watts), in terms of signal power (S watts) and bandwidth (W hertz):

$$C = W \log_2 (1 + S/N) \quad \text{bits/sec} \qquad \text{.....(1)}$$

This is for a single channel. So if we have multiple transmitters and receivers occupying the same channel, then if one occupies all the bandwidth and power the other must be zero. Therefore there must be definable bounds to equate these limits together. Shannon looked at this problem in 1957 at a symposium at Berkeley, in a paper entitled 'Two-way communication channels' [46] and defined these bounds for different codes.

Further work was done by Bergmans and Cover [47] in 1974, showing that several transmitters operating in an additive white Gaussian noise environment (N watts) can send at rates (R, bits/sec) strictly dominating time multiplex and frequency multiplex rates. This is done by the use of a superposition scheme that pools the time (T, secs), bandwidth (W, hertz) and power (P, watts) allocation of the transmitters. For frequency division multiplexing, where $\gamma = W_i/W$, $i = 1,2$, the fraction of the total

bandwidth allocated to each transmitter, and $\alpha_i = P_i/P$, $i = 1,2$, the fraction of the total power, then:

$$R_1 = \gamma_1 W \log_2\left(1+\frac{\alpha_1 P}{\gamma_1 W N_1}\right)$$
$$R_2 = \gamma_2 W \log_2\left(1+\frac{\alpha_2 P}{\gamma_2 W N_2}\right)$$

.....(2)

For time division multiplexing, where τ is the division in time, P_i is the power of transmission during communication to user i and $\lambda_i = \tau_i P_i/P$, then:

$$R_1 = \tau_1 W \log_2\left(1+\frac{\lambda_1 P}{\tau_1 W N_1}\right)$$
$$R_2 = \tau_2 W \log_2\left(1+\frac{\lambda_2 P}{\tau_2 W N_2}\right)$$

.....(3)

The two sets of equations, (2) and (3), illustrate the duality between time and frequency multiplex. The two can be considered equivalent if time variations in input power are allowed in time multiplex and spectral variations of power are allowed in frequency multiplex. Several other papers summarise the findings of researchers in the field. Wyner [48] summarises very concisely research up to the mid-seventies and van der Meulen [49] continues in a similar vein. The IEEE Information Theory Society published a selection of reprint papers in 1994 [50], which contains a lot of interesting material based around multiple-access communications. Bringing the research up to date is a paper by Verdu [51] summarising most of the work done over the last 50 years, including 440 references.

A number of papers appearing in the summaries just described are relevant to the CV-CCMA system of multiple-access. A broad spectrum of material is covered in the paper by Gamal and Cover [52] which sets out to characterise the

capacity region of multiple-access channels, broadcast, relay, etc., for all achievable rates. Following on is the paper by van der Meulen [53] which looks at capacity limits for multiple-access channels and amalgamates a lot of work. Especially interesting are the comments on asynchronous and quasi-synchronous multi-access channels.

Another paper which contained a lot of interesting information was written by W. Lee [54] in 1990 and concerns the channel capacity in a Rayleigh fading environment, the type of environment in which the CV-CCMA system would have to operate in a mobile telecommunications network. The channel capacity in a Rayleigh fading environment is derived and shown to be always lower than the Gaussian noise channel, a 33% reduction at 10 dBm average power but only a 11% reduction at 35 dBm average power. He also shows that a diversity scheme can bring the channel capacity up and that the average channel capacity with Rayleigh fading equals the channel capacity over a Gaussian noise channel when the bandwidth approaches infinity. Summarising:

$$\frac{\langle C \rangle}{B} = log_2 e.e^{-\frac{1}{\Gamma}}\left(-E + ln\,\Gamma + \frac{1}{\Gamma}\right) \quad \text{bit/sec/Hz} \qquad .....(4)$$

where $\langle C \rangle$ is the average channel capacity, B is the bandwidth, Γ is the power and E is the Euler constant. Note the channel capacity in a Rayleigh environment is an average value. Extrapolating these results to the multi-access channel was achieved by Ali [55] in 1991 who based his work on the previous work described and the work of Ni and Honary [56]. He produced two charts illustrating the capacity limits when operating in a flat Rayleigh fading environment with additive white Gaussian noise. These diagrams are reproduced as Figures 2.1 and 2.2, and represent binary phase shift keying and multi phase shift keying respectively.

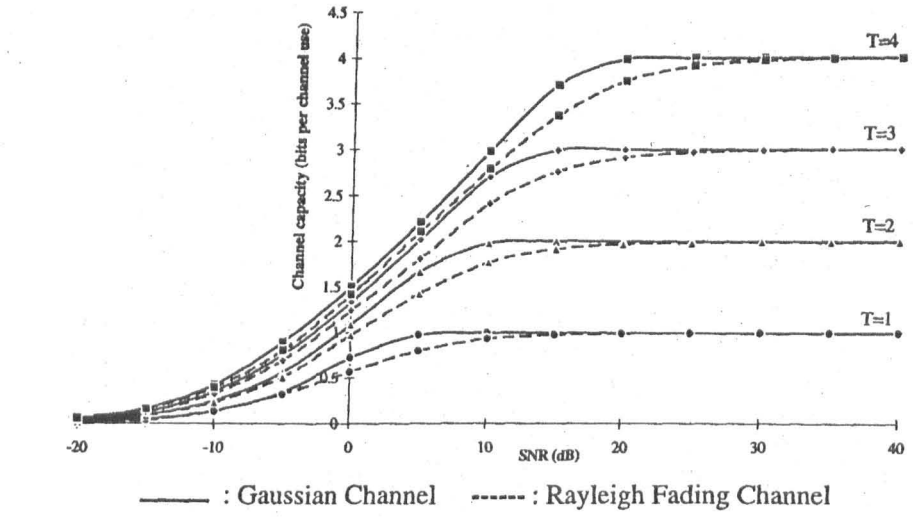


Figure 2.1 Channel capacity of T-user binary phase shift keying

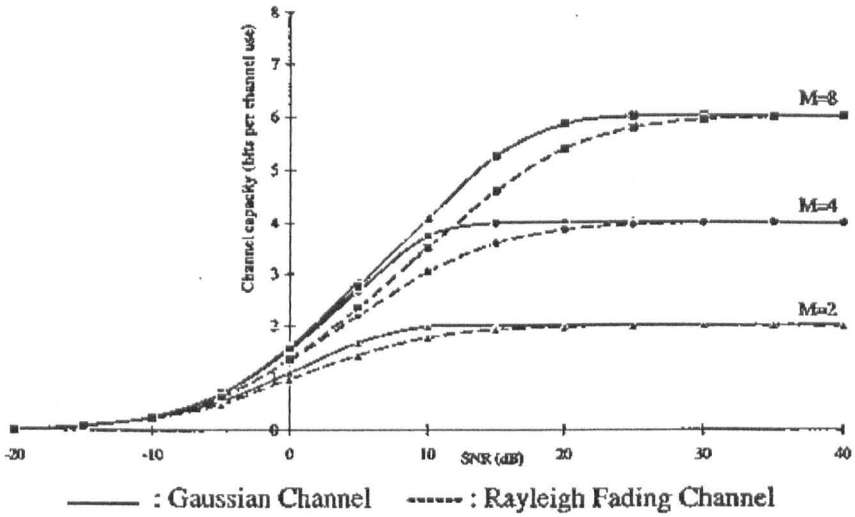


Figure 2.2 Channel capacity of 2-user binary multi-phase shift keying

2.5 Multiple-access coding

Research into the coding systems used in multi access communications also proved to be fruitful. A paper published by Ungerboeck [57] in 1982 describes a coding technique to improve error performance using channel coding with expanded sets of multilevel / phase signals. This increases the free Euclidean distance and hence the detection performance of the system. He shows that coding gains of up to 6 dB can be achieved with different multilevel and multiphase coding schemes; however, these gains are obtained with codes of 2^{10} states. To obtain 3-4 dB of gain only requires codes with four or eight states and, although the paper shows simulation results only, the implementation of these Viterbi-type schemes [58] would make an interesting experiment with the CV-CCMA system. However, any extra gain would be limited by the processing power of the receiver.

Another important piece of work with direct relevance to CV-CCMA was written by Farrell [59] in 1981 and included a concise explanation of the collaborative coding scheme, of which the complex valued system is a direct enhancement of the original system. Published by the same author, together with Brine, Clark and Tait [14], was another very useful and relevant paper. Two collaborative schemes are reported, with the suggestion that two users can operate on one channel with adequate performance. The paper also includes error-rate analysis of a proposed estimation and detection algorithm, showing performance improvements with diversity reception (i.e. 2 aerials). The CV-CCMA demonstrator, described in chapters 5 and 6 of this thesis, advances and develops some of the ideas presented in these papers. A prediction / detection algorithm was not implemented on the CV-CCMA. In the CV-CCMA system the g value was estimated at the same time as the data. This meant that the decode program, running in the signal processor, could

operate at a reasonable rate, and continue to operate whilst performing the other tasks that were required of it. See chapter 8 for further discussion.

References [60 - 63] give different insights into multi-access coding. The paper by Kasami and Lin [60] attempts to solve the synchronisation problem common to all these systems by applying an asynchronous collaborative code scheme to the channel. Following on from this work, Ni [64] introduced state-independent unique decodable collaborative multi-access codes. He ran some simulated systems and showed a fairly good service even in a noisy environment. A slightly different approach was adopted by Ali and Honary in 1991 [65] using a low-complexity soft-decision decoding technique for T-user collaborative coding multiple-access channels. Another paper of interest by Daett and Wolf [66] showed simple codes in a two-user multiple-access channel, apparently working with no synchronisation between codes.

Further relevant work was carried out by Ni [67] in 1992 on codes to be used in a CCMA. He considered the problem of varying numbers of active transmitters accessing the same channel, as would be the case in a mobile phone type network. He introduced a set of codes that would be always decodable, called state-independent uniquely decodable codes, and introduced a technique showing how they could be generated.

A set of codes was generated in a paper written by Wu and Chang [68] for the US military. It contains a new coding scheme with asymptotically good, uniquely decodable codes for the multiple-access adder channel, with and without idle users, and is developed from the Lindstrom combinatory detection algorithm. To complete the survey of current MAC theory, a paper by Steinberg [69] sets out the resolvability region of an independent input MAC, with equations relating the resolvability region to the capacity region of the multiple-access channel.

2.6 Time/frequency division multiple-access

TDMA techniques are a mature technology, and the Telecommunication Standards Sector (TSS) have standards in place governing the technical aspects of all of the TDMA systems used in modern telecommunications systems. The standard text by Feher [70] covers the more up-to-date TDMA techniques. The equipment manufacturers (Alcatel, Marconi, Newbridge etc.) provided information regarding research into fast synchronous digital hierarchy (SDH) multiplexers that are currently in development.

The FDMA radio systems used in telecommunications networks are also defined by the TSS. Feher [70] gives a good overview of the current state of the art, with explanations of the trade-off between performance, cost, modulation methods, etc. The Hewlett-Packard company [71] produces some excellent technical briefs covering the theoretical and practical aspects of FDMA radio systems. It would be relatively simple to incorporate the FDMA system into the CV-CCMA system. A separate channel carrying timing and control information could be used with CV-CCMA and could give performance enhancements.

Of more interest is the current research and development into the wavelength division multiplexing (WDM) systems used for very high capacity trunk links over fibre-optic cable. The *British Telecom Technology Journal* [27] contains a summary of current research at Martlesham Heath (British Telecom research site) and the University of Southampton [72] has a very active research group that publishes very interesting work. The IEEE *Communications Magazine* has recently produced a special issue dedicated to state-of-the-art WDM systems [73]. Included are technical details of current receivers and narrow-band lasers. Collaborative codes and complex valued collaborative systems could be used with certain fibres that have different, but

quite predictable, attenuation and phase change characteristics to give even greater capacity to the trunk systems currently being developed.

This area of research is important, especially to the trunk carriers of telecommunications traffic. The bandwidth requirements placed on the trunk network are doubling every few months, and the rate of increase of Internet traffic is colossal. Within a couple of years, speech will form only a small fraction of traffic, tagged on the back of Internet and data traffic.

2.7 CV-CCMA research

Research into the complex valued collaborative coding multiple-access system has been more restricted. The paper by Chandler [74] contains the distance calculation used to decode the incoming combined waveforms. Care is needed when reading it because there are a couple of misprints in the matrix calculations which can confuse. The matrix array at the bottom of page 223 has two elements crossed and a + / – missed. There is a similar calculation in Chapter 5, section 5.4, of this thesis. This is to illustrate the decoding of a two-transmitter one-receiver CV-CCMA system. A more recent paper [75] shows a method of code construction for binary block codes that can be resolved by the receiver using vector techniques.

When the work detailed in this thesis was started in 1995 it was originally planned that the research was to be a co-operative venture between the University of Sussex and the University of Warwick. However, problems due to the distances involved and the lack of funding have meant that the two institutions have progressed along different, though complementary routes.

The original plan was for hardware design and development to be done at Warwick and software to be done at Sussex. Problems arose due to different signal

processors being used. Once the hardware was assembled and working it was decided to write the software for the Texas Instruments processor at Warwick. This was done directly in assembler language, reproduced in Appendix 1.

2.8 Code division multiple-access

The research work on CDMA systems provided a wealth of information and many ideas as to possible solutions to the problems of the implementation of the CV-CCMA system. The two systems are both multiple-access, the main difference being that the CDMA technique of multi-access requires a spreading signal to be applied to the baseband data. This produces an output signal whose bandwidth is many times greater than required for normal communication systems. The Qualcomm web site [76] has a selection of technical documentation explaining the operation of the system, but no current research findings. In more detail is a paper by Scholtz [77] which describes an idealised spread spectrum system and shows the inherent robustness of communication performance of the system in the presence of interference. Spread spectrum systems have been used by the military for a number of years because of this immunity to jamming signals and the inherent security of a spread spectrum communication link. In the same journal, but published five years later, is a paper by Pickholtz, Schilling and Millstein [78] which is an in-depth tutorial on the workings of CDMA with some interesting comments on synchronisation. A paper by Verdu [79] gives a good comparison between different codes and different detector systems.

There are a number of texts covering the theory of CDMA; the engineering aspects of the system are covered in [80]. Two interesting papers by Viterbi [81,82], covering the coding used and the problems encountered with a practical implementation of a CDMA system, were found to be very useful. Recent

research at Bellcore, published by the IEEE [83], describes a number of important multi-user DS-CDMA detectors which have been proposed, showing that with additional information about the different users decoding can be done more efficiently. In these systems, the code and timing information of multiple users are jointly used to better detect each individual user, minimising the need for power control but greatly increasing the complexity of the receiver, which is already a fairly complicated piece of equipment. The sub-optimal detectors he describes have been developed over recent years, reducing the complexity but trading off performance. The linear multi-user detectors apply a linear transformation to the outputs of a matched filter bank to reduce the interference experienced by the other users. The detector applies the inverse of the correlation matrix to the matched filter bank outputs, thus decoupling the outputs. This type of technique could be tried with the CV-CCMA system, although the implementation would be complicated; the technique is in any case very similar to what is being done in CV-CCMA. The other type he reviews is the subtractive interference cancellation detector, which estimates and subtracts the interference. Again this could be useful with CV-CCMA but difficult to implement.

Another interesting paper written by Adachi, Sawashi and Suda [84] shows various techniques for increasing the data rate over a cellular mobile channel, up to 2 Mb/s and above. This is an area of great interest as current GSM / CDMA systems are pushed to transfer data at 9.6 Kb/s and much more capacity is required for the future. There is currently (1999) a stand-off between the authorities in Europe and the USA as to the promotion of the next generation of 'wideband' cellular telephones, both sides wishing to protect their own manufacturers with claims and counter-claims of patent infringements. The system that is adopted will effectively become a world

standard, so the potential market is colossal. A new system, or variation that could be easily implemented, could upset both sides in the dispute and this is what may happen if the stalemate continues. Adachi, et al introduce an intercell pilot channel, which would be very useful with the CV-CCMA system for control and synchronisation. In the CDMA system it allows fast cell search, fast power control, coherent spreading code tracking, coherent Rake receiver, orthogonal multispreading factor forward link and variable rate transmission with blind rate detection. It allows code division multiple-access to support interference cancellation and adaptive aerial techniques to significantly increase link capacity and coverage. They present computer simulation results showing the advantages of the proposed new system.

Finally, a code division multiplex system using Manchester coding and shown with a simplified radio MAC is written up by Habudi, Hasequi and Makura [85]. They show how good error performance can be obtained with a '3M plex' system, a proprietary multi-level orthogonal code system.

2.9 High-speed radio digital signal processing

Digital signal processing research is mainly restricted to the larger manufacturers of the devices themselves, who endeavour to push the performance of their products to the limit and drive the cost per million instructions per second (MIP) down. Texas Instruments [31] publishes very useful literature, which can be found at the Internet file transfer protocol (FTP) site, and includes interface circuits and illustrations of calculations of performance limits. There is also a bulletin board containing software that is of great value when trying out systems, useful test routines and initialisation routines. The company sponsors worldwide research and a number of papers contained very useful information. For an overview of the Texas Instruments floating

point system [86] is quite detailed; for interfacing [87] contained the most information. A most useful paper showing the floating point format in operation and how to convert between the different standard formats of floating point operation, with code included, is [88]. There are a number of standard self-teach texts available [89,90,91] which give excellent examples of standard floating point operational telecommunications systems. *British Telecom Technology Journal* produced a special issue on digital signal processing in telecommunications [92], which included a number of interesting papers. One describes a low-complexity digital transceiver using low-cost components, and another gives a good overview of digital signal processing in telecommunications.

The filters within the CV-CCMA were implemented using a fixed-point TMS320C50; useful information was also found at the Texas Instruments FTP web site. There are a lot of theoretical texts and papers that cover the design of filters. A design text [93] provided charts and code illustrating standard implementations. An article in *IEEE Communications Magazine* [94] on digital signal processors in cellular radio applications gives standard architectures and cost per MIP comparisons. They include an interesting section on future trends in which they speculate about the problems (already apparent) of very high-density chips having difficulty interfacing with the external world. The solution is to incorporate everything on one piece of silicon, and the new CDMA systems have signal processing systems built into them as part of the ASIC (application specific integrated circuit). They also review some of the new digital signal processing integrated circuits that are specifically designed for the wireless market. The integrated circuits would form an interesting implementation project for the CV-CCMA system.

2.10 Modern digital radio design

This is the most difficult area to research, as most companies (understandably) do not publish their development and research into radio systems. Most British universities do not appear to be active in the field. The textbook literature contains standard and stable techniques for radio frequency generation and reception. However, there are two American magazines [95] and [96] that cover the technical aspects of radio research design and implementation, and publish some interesting research. The CV-CCMA system requires stable and accurate frequency sources in order to work, necessary to synchronise the component parts of the system together. CDMA uses a military technique adopted by the global positioning system (GPS) using satellites, however there is a licence to pay and CV-CCMA needs to be able to stand alone. An article published in 1998 [97] gives an extremely good summary of the various techniques used to generate very stable control oscillators. The author gives comparisons of stability with different crystals and ovens, different circuits and different costs. This was very useful and led to a redesign of the TcXo in the CV-CCMA system. Another article in the same journal in 1995 [98] gives a detailed explanation of the measurement techniques used in digital radio systems, and illustrates with spectra and constellation diagrams the effects of various signal impairments.

Liu [99] investigates the effects of in-phase / quadrature (I-Q) imbalance in quadrature phase shift keying (QPSK) receivers. This was useful because it calculates some bit error rate (BER) relationships in terms of amplitude and phase imbalance.

The demonstrator of the CV-CCMA system described in the thesis uses a type of Delta speech modulation, and a paper by Dunn and Sandler [100]

shows alternative techniques that could be used to improve the resolution and hence the 'quality' of the speech with different multi-bit quantisers. It shows that with ideal multi-bit quantisers, fixed modulators exhibit superior characteristics to adaptive systems, which is interesting and has importance for the next generation of CV-CCMA. Another interesting paper by Armstrong and Strickland [101] shows a novel method of symbol synchronisation. Signal values between the sample points are calculated by interpolation and then processed to find a suitable strobe point; this could be useful on initial set-up of the CV-CCMA system. Some time was spent investigating different synchronisation systems, the CV-CCMA would operate most effectively with reverse synchronisation, ie equalisation applied at the transmitter. As the information for the different links needs to be transferred from the receiver back to the transmitter, the situation becomes complicated. Some possible solutions were investigated and are discussed in chapter 8. There are a number of coding papers which were interesting but did not really advance the solution of the CV-CCMA problems.

Finally, Hewlett-Packard [71] have produced a series of digital radio frequency technique refresher modules for design engineers, which were extremely useful.

2.11 Summary

This chapter contains previously published research material that has direct relevance to the rest of the thesis. The research into current design practice proved beneficial and prevented unnecessary design work being undertaken. Modern filter design and digital signal processor techniques were areas that were intensively studied, producing tangible benefits to the overall work.

However, it was found that the most useful background reading was material and papers published by other researchers in the field. These led to investigations into different codewords, synchronisation techniques and speech digitisation, amongst others.

The next chapter is a short comparative discussion of the merits of CDMA and CV-CCMA.

3 CDMA/CV-CCMA COMPARATIVE DISCUSSION

3.1 Introduction

One of the reasons for Cable and Wireless PLC sponsoring the research, which is in part recorded in this thesis, was to increase the general company knowledge-base of modern multiple-access techniques, particularly radio-based systems. This was desired because a number of competitors were becoming very active in local radio access systems, with other mobile phone operators offering web-browsers etc. on mobile phones. It is thus of interest to discuss the comparative merits of CDMA and CV-CCMA in the context of the present work.

CDMA is a multiple-access technique that is being promoted by certain parties as a solution for all the ills of the current generation of mobile telecommunication systems, that is, a fix for the lack of capacity and a technique whereby significant data rates (Mb/s) can be transferred over a mobile telephone. It is under debate at the present time which of the variations of the system will be implemented as the next worldwide standard for the third generation (3G) of mobile telephones. There are two camps, the American and the European, with governments backing their own manufacturers and interest groups. The most likely outcome is that the manufacturer with the more substantial financial backing will buy out the others and this is likely to be a Swedish company. In the United Kingdom a trial has recently finished of a number of different CDMA systems and because of this it was decided to investigate the proposed systems and compare them with the CV-CCMA system. It is difficult to do a direct technical comparison, particularly as some of the solutions to the problems inherent in using CDMA are kept strictly confidential for commercial reasons. This section attempts to outline some of the strengths and weaknesses of both systems.

3.2 CDMA summary

CDMA is a spread-spectrum technique that grew out of military anti-jamming systems developed in the second world war. A introduction to CDMA operating principles is included in Chapter 1. The system has only been used commercially in the last ten years; prior to this it was the preserve of the military. CDMA has an inherent weakness when used as a multiple-access technique. Assume that there is a 20 dB difference between the signal-to-noise ratios of two units that are trying to access a single receiver (like the CV-CCMA system). To accommodate this difference, the system would need to operate a spreading bandwidth at least 40 dB greater than the data rate, i.e. massive. If a lesser bandwidth system is used the lower signal will not be able to access the local node (usually called DN – distribution node). The problem is solved in commercial multiple-access systems by controlling the power of the transmitters so that the signal strengths at the receiver are of the same order of magnitude.

If we assume perfect power control, with M users, and energy per bit of E_b, with noise power density N_o , where S is the average modulating signal power and T is the time duration of each bit, and bit rate is R , then :

$$E_b = S\,T \qquad \text{.....(5)}$$

and $$E_b = S / R \qquad \text{.....(6)}$$

therefore :

$$E_b / N_o = S / RN_o \qquad \text{.....(7)}$$

Now substitute the noise power density N_o for total noise power N divided by bandwidth W, $N_o = N / W$:

$$E_b / N_o = (S / N)(W / R) \qquad \text{.....(8)}$$

Because perfect power control is assumed it can be stated that at the receiver the received powers from all users are equal. So with M users:

$$E_b / N_o = (1 / M - 1)(W / R) \qquad \text{.....(9)}$$

Solving for M–1 gives the equation relating the number of users to the energy per bit per noise power density (E_b / N_o) and the processing gain (W / R) :

$$M - 1 = (W / R) / (E_b / N_o) \qquad \text{.....(10)}$$

Note that other types of interference are ignored, for example thermal noise, which would degrade the capacity of the system.

Using a 9.2 Kb/s data rate gives a value for M of about 15 dB, or about 32 users in the same bandwidth compared to just two users on the old AMPS system. This gives an idea of the capacity increase available with CDMA, but remember this assumes perfect power control. At present with the CV-CCMA system we can cope with a maximum variation between input signal powers of 40 dB without the complexity of the power control circuitry required in the CDMA system. This illustrates the first possible advantage of the newer system, showing

the reduction in complexity over the first system and hence cost savings and reliability benefits.

3.3 Standards

The North American CDMA standard IS95 [102], which has become one of the more popular world standards specifies that each user's baseband data is input at 9.2 Kb/s . This is called rate set 1, and the final spread signal is set to 1.2288 Mb/s giving an output bandwidth of approximately 1.25 MHz. This frequency was chosen because it is one eighth of the original advanced mobile phone system (AMPS) allocation, so telcommunication operators can migrate to CDMA in sections. IS95 specifies the transmission as a forward channel, from the mobile to the base station, and a reverse channel from the base station to the mobile. Tables 3.1, 3.2, 3.3 and 3.4 summarise the specification; for the complete details please refer to [102]. The third-generation CDMA system standards are in the process of being defined; refer to the relevant ITU-T committee for details.

Channel	Sync	Paging		Traffic			
Data rate (b/s)	1200	4800	9200	1200	2400	4800	9200
Code repetition	2	2	1	8	4	2	1
Modulation symbol rate (s/s)	4800	19,200	19,200	19,200	19,200	19,200	19,200
PN chips/mod symbol	652	64	64	64	64	64	64
PN chips/bit	1024	252	128	1024	512	252	128

Table 3.1 Forward link channel parameters, rate set 1

The IS95 forward link currently supports a 9200 b/s rate family in the three data-bearing channel types, traffic, sync. and paging. In all cases the forward error correction code rate is 1/2 and the pseudo-random sequence rate is 1.2288 MHz.

Channel	Traffic			
Data rate (b/s)	1800	3200	7200	14400
Code repetition	8	4	2	1
Mod. symbol rate (s/s)	19,200	19,200	19,200	19,200
PNchips/modulation symbol	64	64	64	64
PN chips/bit	282.27	341.33	170.27	85.33

Table 3.2 Forward link channel parameters, rate set 2

The second channel rate family has a maximum rate of 14400 b/s . Rate set 2 uses a forward error correction code rate of ¾; this is created by puncturing the code used in rate set 1.

Channel	Access	Traffic			
Data rate (b/s)	4800	1200	2400	4800	9200
Code rate	1/3	1/3	1/3	1/3	1/3
Symbol rate before repetition (s/s)	14,400	3,200	7,200	14,400	28,800
Symbol repetition	2	8	4	2	1
Symbol rate after repetition (s/s)	28,800	28,800	28,800	28,800	28,800
Transmit duty cycle	1	1 / 8	1 / 4	1 / 2	1
Code symbols/ modulation symbol	6	6	6	6	6
PN chips / modulation symbol	256	256	256	256	256
PN chips transmitted /bit	252	128	128	128	128

Table 3.3 Reverse link channel parameters, rate set 1

The IS95 reverse channel currently supports a 9600 b/s rate family in the access channel and traffic channel. The forward error correction code rate is 1/3, the code symbol rate is always 28,800 symbols per second and there are six code symbols per modulation symbol; the pseudo-random sequence rate is the same as before, 1.2288 MHz. The modulation is 24-ary orthogonal using the same Walsh functions as the forward channel. The secondary traffic channel also operates at 14400 b/s and uses a rate 1/2 code in place of the rate set 1 code.

Channel	Traffic			
Data rate (b/s)	1800	3200	7200	14400
Code rate	1 / 2	1 / 2	1 / 2	1 / 2
Symbol rate before repetition(s/s)	3,200	7,200	14,400	28,800
Symbol repetition	8	4	2	1
Symbol rate after repetition	28,800	28,800	28,800	28,800
Transmit duty cycle	1 / 8	1 / 4	1 / 2	1
Code symbol/modulation symbol	6	6	6	6
PN chips/modulation symbol	256	256	256	256
PN chips transmitted/bit	256/3	256/3	256/3	256/3

Table 3.4 Reverse link channel parameters, rate set 2

3.4 Comparisons

Although both systems are multiple-access radio technologies, it is difficult to do a technically valid direct comparison between the two because the two systems are fundamentally different in operation. Both systems have advantages and disadvantages when compared to one another and to the other multiple access technologies. The main operational characteristics of both systems are highly interrelated, especially the geographical coverage of the system, the quality of service and the capacity of the system. All these parameters are dependent upon one another, that is you cannot improve one parameter without degrading another.

3.4.1 Capacity

The capacity of the CDMA system is limited by a number of factors: the receiver demodulator and sensitivity, power control and interference. If we assume the transmitter stations have perfect power control, then the received signals will all have approximately the same power on arrival, but obviously this will not happen in reality because of the fluctuations present on a mobile signal. If we assume an available bandwidth of 1.25 MHz (IS95), and taking E_b/N_o to be between 3 and 9 dB, we have a maximum capacity of 24 users. The AMPS systems, which the CDMA is designed to replace, would have only 42 channels (1.25 MHz / 30 KHz). However, this is not a fair comparison as the adjacent cells in the AMPS network need to be taken into account because the channels cannot be re-used in adjacent cells.

Taking into account a frequency re-use factor, F and interference I_o:

F = Total interference power / own cell interference

$$N = \frac{(W/R)}{\left(E_b/(N_o + I_o)\right) * F} \qquad \text{.....(11)}$$

Assuming even cell loading gives CDMA an advantage of 21/1.6 over the AMPS system, with sectored cells, or 7/1.6 without.

In the CV-CCMA system, if we assume a maximum of four users per 30 KHz channel of the 1.25 MHz bandwidth (this is the maximum that has been proved to work in the laboratory, although in theory this can be much improved upon), then we have 160 (40 x 4) users simultaneously communicating in the same

bandwidth at 9.6 Kb/s. This shows another advantage of the CV-CCMA system: a large possible capacity increase over CDMA.

3.4.2 Power control

One of the weaknesses of CDMA is an overloading effect which means that transmitters need to have quite complicated two-way power control. A crude version is built into cellular systems anyway but CDMA requires very accurate control because the other users will appear to a station as noise, so the level of that noise must not desensitise the receiver. This is sometimes known as the near-far problem. If there are two transmitting stations communicating with one base station or receiver and one of the transmitters is much nearer to the receiver than the other one, then with no power control the receiver would receive a much stronger signal from the near station than from the far one. If the required signal-to-noise ratio is 1:10, and the difference between the distances between the transmitters and receiver is of an order of ten, then the distant station is on the limit of the required signal-to-noise ratio. This is because of the 'noise' from the other transmitter. If another station fires up in the cell, the third station will not gain access and will probably drop the others out as well. To prevent turn-on problems of this type, a series of access probes are defined to set up communication into a CDMA network; a low-power probe is initially sent and then the power is gradually ramped up until access is made.

Standard mobile phone cellular networks operate with the reverse of this system; they transmit a burst of maximum power to log-on to the network, and then ramp down the RF power output, mainly to conserve battery power. This power control and access technology is an additional complexity of the CDMA

system. The CV-CCMA system will be less prone to power desensitising. The test set-up had hardwired connections with attenuator/phase shifters interconnecting the transmitters and receivers which showed the system to be resistant to desensitising. Provided there is sufficient input signal, nominally greater than –100 dBm into the Philips front-end integrated circuit, which has a noise figure of 2-3 dB, the CV-CCMA will decode the incoming waveforms. On the initial test set-up a larger signal was required because the UHF amplifiers were disconnected. If an interfering signal is approximately 1 MHz removed from the wanted carrier, it needs to be at 0 dBm or greater before the mixer outputs become affected. However, closer interference caused quite drastic problems and would need to be resolved in a production unit by better filter and layout design. The CDMA system is inherently more conservative with transmitter radio frequency output, as each transmitter is constantly adjusted for minimum output power in order to achieve optimum performance. The carrier is also lowered to a pilot level whenever gaps in the speech are presented to the transmitter. CV-CCMA has none of these facilities (at present) but could have power control implemented relatively easily as the Philips driver integrated circuits are inter integrated circuit (I^2C) devices [103] and therefore the gain could be controlled via the I^2C bus.

3.4.3 Timing

The CV-CCMA system is as prone to timing problems as the CDMA system, but with any multi-access technique it is evident that in order to decode the contributing signals the start and stop points are required; the ideal way to do this is with very accurate timing. The optimum multiple-access technique would incorporate an asynchronous timing scheme; however, this has proved to be an

elusive goal. Both systems therefore require expensive timing circuitry. The initial demonstrator incorporated hard-wired, central clock synchronisation in which all the component parts of the demonstrator were locked to a master timing source, (type 1 sync). The system was then developed to recover timing from the data streams (type 2 sync). However, the implementation of the synchronisation recovery from a separate radio frequency channel, (referred to as the type 3 system), and multiple sampling software together with pulse shaping and RF filters helped to reduce the sensitivity of the CV-CCMA to timing drift and other synchronisation anomalies.

3.4.4 Noise

A comparison of the two systems in terms of noise performance is difficult, mainly because of the many different types of noise affecting the systems in different ways. In a mobile radio environment noise is often spiky; if these spikes, which often have sweep bandwidths of tens of MHz, fall across the CV-CCMA channels then data will be lost. The CDMA system is more resilient due to the spreading of the waveform and data will be preserved. If the noise is a more broadband noise, for instance other CDMA network stations, secure communication systems, etc., then CV-CCMA should prove the more resilient.

3.4.5 Multipath propagation

System capacity is affected by propagation phenomena. The fading of signals in a moving vehicle is caused by the motion of the vehicle through interference patterns. This is a big problem with the older cellular technologies and will cause problems with CV-CCMA as the g value will be subject to some large variations.

CDMA is more robust, and multipath components can be resolved if their delays are separated by at least the de-correlation time of the spreading. They will not interfere because each component correlates to a different delay. If the delay is less than the de-correlation time then CDMA falls over; this is due to flat fading. Most modern systems incorporate fairly substantial forward error correction and interleaving to reduce these effects, but at the cost of yet more complexity.

3.4.6 Cost

Comparing the two systems on a cost per bit basis gives the CV-CCMA system a major advantage due to its greater spectral efficiency, and the (much) lower cost of the electronic components needed to construct the system, a cost that would further reduce if the system was ever put into mass production. In the United States, and recently in the United Kingdom, the governments have raised quite significant revenues by auctioning the radio spectrum. The CV-CCMA would therefore make an ideal candidate for trialling in one of these bands, for example, a cheap phone plus internet access service. The low cost of the system means that it could be used in non-professional applications, such as amateur radio. The system could be used as an access coding requirement for input to a radio repeater, or on one of the packet data channels as this would increase the spectrum-usage efficiency and prevent some of the overcrowding problems currently plaguing the restricted amateur bands.

The next chapter looks at the theoretical capacity limits for multiple-access systems.

QCP™ 1960 CDMA
Digital PCS Phone

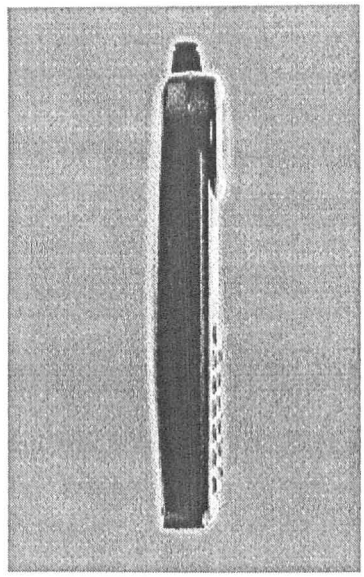


Figure 3.1 Typical CDMA handsets (courtesy Qualcomm)

4 MULTIPLE - ACCESS CAPACITY LIMITS

A brief introduction to, and explanation of, the basic operating principles of the complex valued collaborative coding multiple-access system has been given in Chapter 1. This chapter contains a more detailed investigation of the background theory relating to the capacity limits of multiple-access channels. This theory is then expanded in the next chapter, with specific reference to CV-CCMA.

4.1 Single-channel theoretical capacity limits

Most telecommunication systems are designed so that the minimum amount of bandwidth is used in order for the system to operate correctly. However, there is a theoretical minimum bandwidth requirement for any communication channel that is defined in the theorem of Nyquist [104]. It states that R_s symbols per second can be received without inter-symbol interference (ISI) in a $R_s/2$ Hz minimum bandwidth channel. If we take into account filtering or pulse shaping:

minimum (Nyquist) bandwidth = W

$$W = 0.5(1 + r)\,R_s \qquad \text{.....(12)}$$

where r is the filter roll-off factor.

When we then double sideband (DSB) modulate a radio frequency carrier by amplitude shift keying (ASK) or phase shift keying (PSK) the output, we require twice the bandwidth of the original baseband signals, giving:

$$W_{PSK} = W_{ASK} = (1 + r)R_s \qquad \text{.....(13)}$$

Most modern, especially lower frequency, communication systems suppress one of the sidebands, and also possibly the carrier, in order to give better spectral efficiency.

These systems are called single sideband (SSB) systems. Current telecommunications microwave radios operate up to 256 quadrature amplitude modulation points in the output constellation (256 QAM).

If we have a multi-symbol coding system, as CV-CCMA effectively is at the receiver, with the coder assigning to each of M symbols a k bit representation giving $M = 2^k$, then as k increases the efficiency in bits/s/Hz also increases. However, efficiency is traded against E_b / N_o where E_b = energy per bit and N_o = noise power spectral density, as more power is required at the transmitter and/or a better noise figure (NF) at the receiver. NF is the ratio of the signal to noise at the input to the signal to noise at the output. Bit error probability curves [105] are reproduced in Figure 4.1 to show this trade-off, showing different modulation schemes versus best-case bit error rates. The theorem produced by Claude Shannon [45] in the late 1940s showed channel capacity to be a function of noise in the channel, receive power and bandwidth. Although interference and synchronisation are not included, the theorem provides a good basis to bound maximum channel capacity. The theorem is stated as:

$$\text{channel capacity} = C = W \log_2(1 + S/N) \quad \text{bit/sec} \qquad \text{.....(14)}$$

where W = bandwidth (Hz)

S = average signal power (Watts)

N = average noise power (Watts)

Thus it is theoretically possible to transmit data at a rate R over the channel with an arbitrarily small error rate if R is less than C, by using suitable coding schemes. Using (14), Shannon plotted the achievable limits of practical systems, as

reproduced in Figure 4.2. Note the noise received is proportional to the bandwidth available for the channel as explained below:

At the receive detector, $N = N_o W$, and substituting back into (14) gives:

$$C / W = \log_2 (1 + S/N_o W) \qquad \text{.....(15)}$$

At the limit of capacity, where $R = C$ (i.e. transmission bit rate = capacity):

$$\frac{S}{N_o C} = \frac{E_b}{N_o}$$

$$\frac{C}{W} = \log_2\left[1 + \frac{E_b}{N_o}\left(\frac{C}{W}\right)\right] \qquad \text{.....(16)}$$

where E_b = bit energy and N_o = noise power spectral density.

There is a limit of E_b/N_o below which there can be no error-free communications. This can be calculated from (16) using the mathematical identity:

$$\lim_{x \to 0} (1 + x)^{1/x} = e \qquad \text{.....(17)}$$

$$\text{let } x \quad = E_b/N_o \, (C/W) \qquad \text{.....(18)}$$

where x is a variable and e is the base of natural logarithms.

$$\frac{C}{W} = x \log_2 (1 + x)^{1/x}$$

$$1 = \frac{E_b}{N_o} \log_2 (1 + x)^{1/x}$$

At the limit C/W goes to zero;

$$\frac{E_b}{N_o} = \frac{1}{\log_2 e} = 0.693 = -1.59\text{dB} \qquad \text{.....(19)}$$

This value is called the Shannon limit. It is not possible to reach the Shannon limit because as $1/x$ increases, the bandwidth and implementation complexity increase to infinity. For a bit error rate of 1 in 100,000, using direct binary phase shift keying (as used in the first practical implementation of the CV-CCMA system) an E_b/N_o of 9.6 dB per channel is required. Using coding schemes implemented over each channel of the CV-CCMA this could be reduced.

Shannon's work predicted improvements of over 11 dB ($-1.59 \rightarrow +9.6$ dB) by using coding techniques. However, modern commercial systems improve E_b/N_o by approximately 7 dB. For detailed work on these limits refer to [106].

The Shannon theorem assumes an additive Gaussian noise enviroment. However, digital radio communication in the mobile enviroment, a major potential application of CV-CCMA, has channel characteristics often approximating to multiplicative Rayleigh varying statistics.

W. Lee [54] explained that the channel capacity in a Rayleigh fading enviroment has to be calculated in an average sense due to the fading, and that the channel capacity can be increased to Gaussian channel capacities if we introduce diversity with M (the number of branches) greater than four. The paper shows the reduction in capacity of the channel to be approximately 30% and proves the average channel capacity over Rayleigh fading equals the channel capacity over Gaussian noise when the bandwidth approaches infinity. The calculations for channel capacity in the mobile enviroment take into account the Rayleigh effects, and as a first approximation will reduce the capacity for each link by one third.

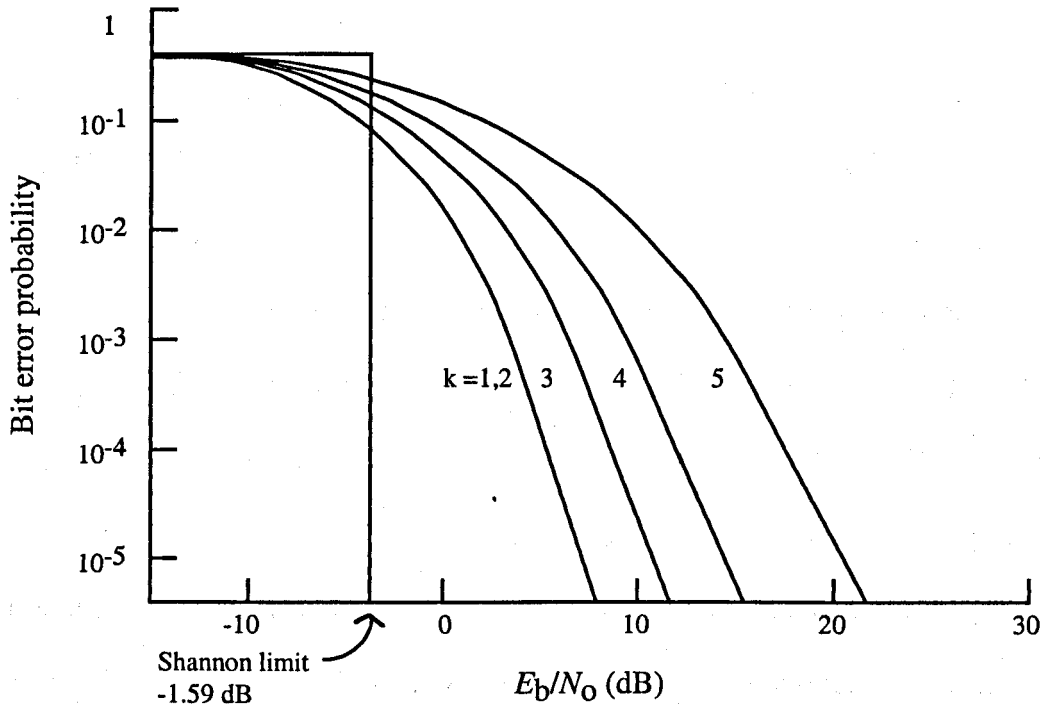


Figure 4.1 Bit error probability for multi-phase signalling (coherent $M = 2^k$)

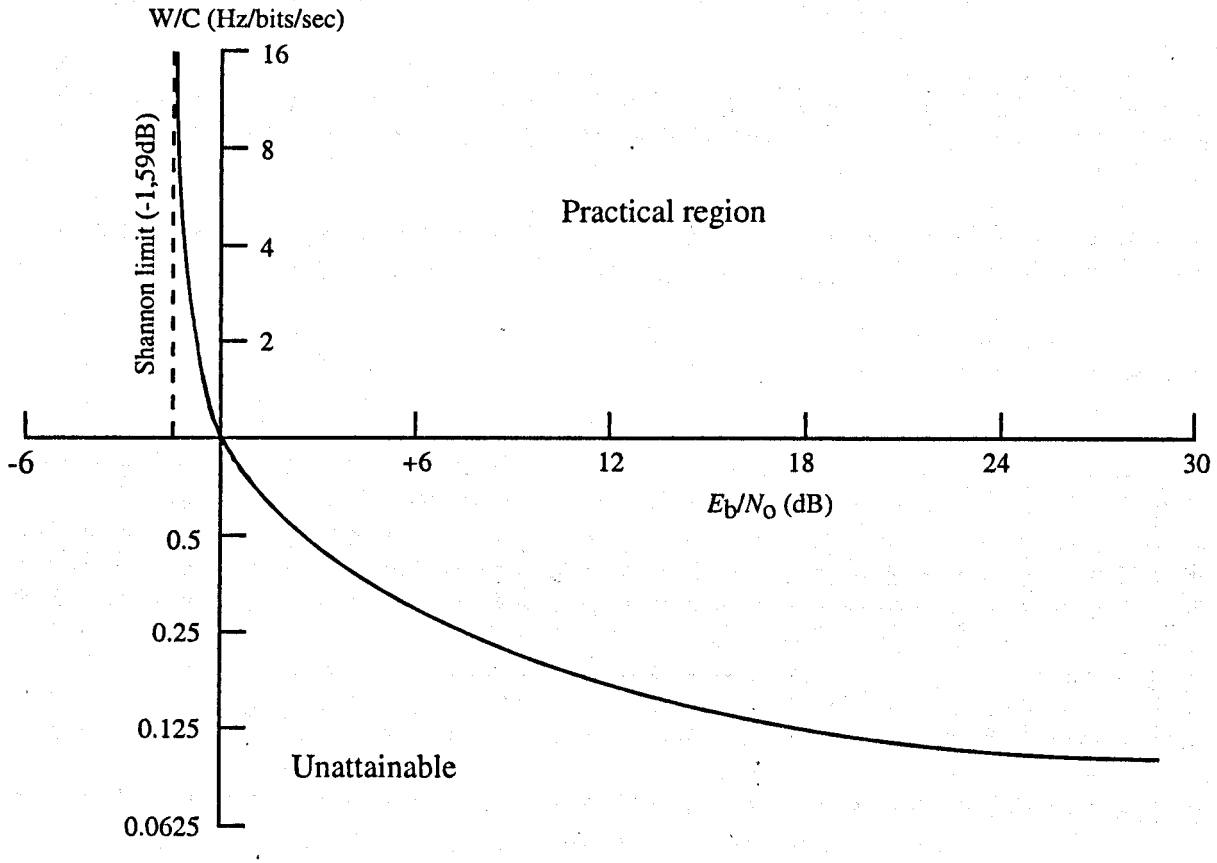


Figure 4.2 Normalised channel bandwidth versus E_b/N_o

4.2 Multiple-channel theoretical capacity limits

The CV-CCMA is a multiple-access system and each of the contributing transmissions needs to be added together (as the channel is treated as an adder channel). Shannon produced a paper in 1961 [46] on two-way communication channels which looks at decode probabilities with different codes and different channel types. Work was also carried out by El-gamal and Cover [52], Slepian and Wolf [107] and Ahlswede [108] in the 1970s and 1980s. The following is derived from these together with the summary of Shannon theory by Wyner [109].

Assume that we have two transmitters and one receiver, and Gaussian characteristics in the communication multiple-access channel. The output of the channel Y is the sum of X_1, X_2 and G, where X_1 and X_2 are the inputs and G is the noise random variable; then all achievable rates must lie in the shaded area indicated in Figure 4.3. The references contain more rigorous mathematical derivations.

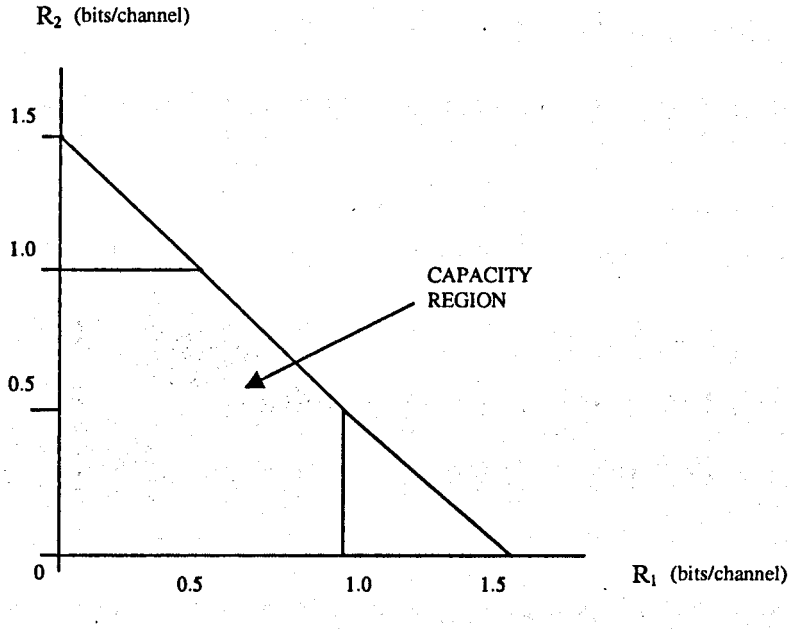


Figure 4.3 Capacity region of 2-user multiple-access channel

If we combine the achievable rate region found by Bergmans and Cover [47] and the capacity region found by Wyner [109] with the proof by Keilers [110] and the summary of van der Meulen [49], the capacity region of the discrete time multiple-access channel is given by:

$$C(\text{AWGN},k_{21},N,P_1,P_2,I)=\left\{(R_1,R_2):0\le R_1\le\frac{1}{2}\log_n\left(1+\frac{P_1}{N}\right)\right.$$
$$\left\{0\le R_2\le\frac{1}{2}\log_n\left(1+\frac{P_2}{N}\right)\right.$$
$$\left\{R_1+R_2\le\frac{1}{2}\log_n\left(1+\frac{P_1+P_2}{N}\right)\right. \qquad \text{.....(20)}$$

The channel has additive white Gaussian noise, K_{21} defines a discrete memoryless MAC with two input users and controlled powers P_1 and P_2 and noise variance N.

Work by Chandler et al studied channel capacity limits for multiple-access channels with slow Rayleigh fading [111]. They used the technique described by Ungerboeck [112] with a Monte-Carlo technique to calculate the capacity for a multi-user CV-CCMA system in a slow Rayleigh fading enviroment (the MAC) with additive white Gaussian noise.

Summarising, assuming there are T transmitters, each with M levels or phases, then there are $N=M^T$ inputs to the MAC and at the output of the MAC:

$$Z=\{z_0,\ldots,z_k,\ldots z_{N-1}\} \qquad \text{.....(21)}$$

where Z_k is the kth composite signal representing the kth combination of T user transmissions given by:

$$Z_k = \sum_{i=0}^{T} s_i^{\,k}(b_i g_i) \qquad \text{.....(22)}$$

where s_i^k is the ith user transmitted complex symbol and b_i is the ith user constant. They extended the expression from [112] for channel capacity C:

$$C = \max_{\Pr(0)\ldots\Pr(N-1)} \sum_{k=0}^{N-1} \Pr(k).\int_{-\infty}^{+\infty} P(y \mid z_k)\log_2\left\{\frac{P(y \mid z_k)}{\sum_{u=0}^{N-1}\Pr(u)P(y \mid z_u)}\right\}dy \qquad \text{.....(23)}$$

where $\Pr(k)$ is the probability associated with z_k, and treated as equiprobable, and $P(y|z_k)$ is the conditional probability for the received composite symbol y, $y = Z_k+w$, and w is the complex valued noise sample with zero mean and variance σ_w^2. The capacity equation is then simplified by assuming the values of g_i are drawn from a zero mean uncorrelated bivariate Gaussian distribution, giving a Rayleigh amplitude distribution and a uniform phase distribution:

$$C = \log_2(n) - \frac{1}{N}\sum_{k=0}^{N-1} E\left\{\log_2 \sum_{u=0}^{N-1}\exp\left[\frac{-|z_k + w - z_u|^2 - |w|^2}{2\sigma_w^{\,2}}\right]\right\} \qquad \text{.....(24)}$$

This equation is used with a Monte Carlo analysis to find the capacity limits for the T user MAC, and they show that the maximum channel capacity for a given value of T and M is achieved in a slow Rayleigh fading channel at a signal-to-noise ratio approximately 10 dB higher than in a Gaussian channel.

The results do give a good first approximation for capacity limits, but the real life situation is somewhat more complicated. The probablities Pr(k) are assumed to be equiprobable, in order to simplify the equations, which they may or may not be. More significant is the fact that the complex channel loss $\boldsymbol{g}$ values are taken as constant; this is fine in the laboratory set-up but this will not be the case in the mobile-radio type enviroment. Due to the fact that CV-CCMA identifies the $\boldsymbol{g}$ values as a part of the demodulation process, this must reduce the net capacity under time-varying conditions. It therefore seems logical and reasonable that the reduction in capacity would be related to the $\boldsymbol{g}$ value estimation, more precisely the entropy of the $\boldsymbol{g}$ value estimation. If we can increase entropy by other means, for example a multiple aerial array which would give additional information about the incoming signals, then a benefit in the capacity of the system could be expected. This would be especially beneficial if we have multiple receivers which all require a $\boldsymbol{g}$ value estimation in order to decode the incoming signals. The aerial system would then be connected to the various receivers and information about the signals could be resolved by a high-speed scan and used by all of the receivers. Another dedicated processor would be required but the implementation need not be too complicated. The system could incorporate other useful features, possibly automatic transmitter power output adjust and/or input sensitivity variation. This is another interesting offshoot of the CV-CCMA research which will hopefully be pursued in the future.

In the laboratory set-up the processing/speed limitations are caused by the time taken by the digital signal processor to input, multiply and resolve the incoming data. The first implementation uses a 50 MHz DSP and, taking instruction time as 40ns, the minimum number of calculations are approximately (11 x 3) + 19 + (16 x 4) = 116 , giving a calculation time of 4640ns , and a maximum symbol clock of

21.55 KHz. This calculation is approximate and assumes the parallel commands of the digital signal processor are fully used. The multi-sample software, used with the type 3 synthesiser to reduce timing criticality, has a reduced maximum data rate of 1.5 Kb/s per channel. See the software section, Chapter 6, section 4, for comment and further discussion about this. Obviously a more powerful processor would increase these capacities. Texas Instruments have recently launched a processor with an order of magnitude more processing power than the devices used in the current CV-CCMA and these will be tried when funds allow. It was found that the much-vaunted parallel command sets of the digital signal processors were so restrictive in operation that it was often found that faster code could be written by ignoring them and going for restrictions to the iterative loops in the receiver.

In the single transmitter/single receiver channel, feedback does not increase the capacity of the channel. However, with multiple-access the situation becomes somewhat more complicated. Gaarder and Wolf [113] showed that the capacity of an additive white Gaussian noise multiple-access channel increases with noiseless feedback. This work was enhanced by Thomas [114] who proved that the maximum increase in capacity for the noiseless multiple-access channel with feedback is approximately doubled. Apart from the filter implementations and some experimentation with a coarse analogue automatic gain control, no feedback was applied to the CV-CCMA systems developed at Warwick.

The next chapter looks in detail at the theory underlying the CV-CCMA system.

5 THEORY OF OPERATION

5.1 Introduction

This section explains in detail the theory behind the operation of the new CV-CCMA system. The CV-CCMA is designed for multiple-access operation and the simplest implementation is therefore two transmitters and one receiver, operating over the same bandwidth. This is how the initial design was envisaged and so the explanation of the theory of operation will be expanded upon assuming this configuration, unless stated otherwise.

Each of the transmitters has programmed into its coder memory a number of unique collaborative codes. The input signal, either a test pattern or digitised speech, is buffered and then fed to the coder. In the initial implementation of the system these codes were required to be linearly independent [115]. This means that there are no multiplying constants (apart from zero) which would set the vector sum of any number of transmissions to zero, preventing reception at certain combinations of complex loss (g) values. That is:

If g_1, g_2, g_3, etc. are complex loss values, treated as constants,

and V_1, V_2, V_3, etc. are vectors, of the same type,

then for linear independence:

$$g_1V_1 + g_2V_2 + g_3V_3 + \ldots g_nV_n = 0 \qquad \ldots\ldots\ldots(25)$$

only when all constants are zero.

The codewords were also required to be partitioned between users. This enables any combination of signals transmitted by the users to be uniquely decoded and allocated

to the particular user. However, with the codewords all linearly independent, the capacity of the system is restricted more than is necessary, as will be described later. Decoding is based on the fact that each combination of transmitted codewords describes a unique hyperplane as g is varied. Positioning on it will be determined by the complex loss coefficient (the g value), that is the phase change and attenuation occurring between the transmitter(s) and the receiver(s). Note this is assumed to be constant over the period of the codeword and unique to each transmission path.

The transmitter outputs may be considered as connected together via attenuators and phase shift networks when launched into free space via the aerials and hence transferred to the receiver. At the input of the receiver a waveform is received that is made up of the sum of all the transmitted codewords, weighted by complex transmission loss coefficients (the g values). In the first stage of the receiver the signal is amplified and is then presented to the next stage as the received vector. The distances of the received signal vector from each of the hyperplanes is calculated and the one with the minimum distance is chosen as the most likely transmitted codeword pair. The position of the projection onto the plane gives and its distance a measure of the noise. The mathematics relating to the process will be covered later in this section. Note that in order for the system to function it is necessary to have synchronisation between the transmitters to provide simultaneous bit transitions at the receiver to which it can be synchronised. This is so that the receiver will 'capture' the combined signal at the correct instant and hence be able to process and decode the individual contributing codewords. Sub-section 5.2 looks at this problem in more detail.

Stored in the memory of the receiver's digital signal processor are pre-calculated matrix values associated with all combinations of the transmitted codes used within the system. The transmitted codeword matrix array is multiplied by its

generalised inverse and then the result is subtracted from the standard identity matrix (I). The size of I is determined by the size of the codeword array. If vector array S_k is the transmitted codeword array, and P_k is the array corresponding to the pseudoinverse (generalised inverse) of the transmitted codewords, the calculation required is $I - S_k P_k$. This equation will be explained in more detail later in the section. Then, as a subroutine running on the receiver signal processor these stored arrays are multiplied by the received vector, and the Euclidean norm is calculated for each of them and stored. The minimum value is selected and this corresponds with the minimum distance to any plane defined by a valid combination of transmitted codewords. It is then necessary to use a simple look-up table to recover the original codewords and output the recovered data to the speech decoder or the data sink. It is also possible to recover the g values, assuming the codewords have been correctly decoded, by simply multiplying the pseudoinverse by the received vector. The g value could be stored in a temporary buffer and used as an aid to decode the next received codeword. The codewords used in the demonstrator used a common first symbol. This was to simplify the operation of the decoder. The g value is assumed not to change between codewords and if the first symbol is known then less processing is required. The complete receive process will be explained in more detail in the receiver processing section.

In the laboratory, an initial training sequence is used with a fixed, known value of loss, and a fixed phase shift between the transmitter and the receiver. This makes a useful check on the operation of the decoding process, it is beneficial to compare calculated g with actual g. In the laboratory set-up, attenuation and phase shift are easily applied. Apart from training, the sequence of events just described needs to occur for every codeword combination received.

In order to reduce sensitivity to timing impairments, a multiple-sample system was implemented. Three samples were taken, based at an even spacing and centered around the recovered synchronisation points. The last sample of the group of three was used as the first sample of the next group, providing overlap. This reduced the throughput of the system but made it more robust in operation. The operation of this later system is discussed in sub-section 5.2 and the results obtained with it are reproduced in Chapter 7. The Texas Instruments dual floating point digital signal processor (development system), together with the Burr-Brown dual 18-bit analogue-to-digital converter, was found to have sufficient dynamic range and processing power to achieve this.

In the initial implementation(s) of the system, the transmitter codeword generators restrict the codewords to a subset of the total number of codewords available, enabling unique decodability. Only one codeword can be shared, as otherwise it would be impossible to know who had transmitted which codeword, or so it was initially thought. Also, there must be some certainty of two transmitters transmitting, as without this it would be difficult to know whether both had sent it or one had stopped (although this would give a $g = 0$ value which could be correctly interpreted by the DSP). However, if we assume that the g value will remain relatively constant from one codeword to the next, and providing that the g values for the different transmitters are not too close together, then data from successive pairs of codewords can be separated into distinct streams.

This means that all users could use the same codeset, and identification data could be embedded within each transmitter's output stream. This effectively removes the need for pre-allocation of timeslots, frequency or codes, as required by other multiplex systems, and gives a major advantage to CV-CCMA. The only

weakness is that of decoding the signals when the g values are similar, as, for example, with mobile CV-CCMA users located in the same car in a traffic jam or in the same carriage of a train. However, the g value is two-dimensional so the probability is small, and the two users would need to be within a fraction of a wavelength of one another, or an exact integer number of wavelengths apart.

5.2 System synchronisation

It is an essential requirement of the CV-CCMA system that the component parts of the system be correctly synchronised. The receiver needs to know the correct synchronisation points in the incoming stream of data in order to decode and process the information. Three types of synchronisation were used with the CV-CCMA demonstrator, type 1 synchronisation, type 2 synchronisation, and type 3 synchronisation. Type 1 comprised one master oscillator, with all system components slaved to it. Type 2 recovers timing from the incoming data streams and type 3 has a separate 10MHz timing channel, with a single master oscillator.

The transmitters, two or more using a particular channel, are operated with the same output frequency. In the demonstrator they are effectively synchronised together due to the high-stability oscillators which are used as reference frequency sources in the transmitters and receivers. These are identical devices having very little drift - in the order of fractions of a part per million. Even so, if there is a relative frequency difference of one part per million with the system operating at one gigahertz, then the frequencies would be one kilohertz awry. The codewords must be short compared to one millisecond in order for constant relative phase to hold for the duration of the codeword. Enhanced synchronisation techniques were investigated and better performance would be achieved by synchronisation to a recovered timing

channel. This synchronisation technique was implemented as a separate 10 MHz channel and used to phase-lock the PLL's to the master timing source (referred to as a 'type 3' synthesiser). With the system operating in the mobile environment the Doppler effect (relative frequency change due to the movement of the stations) would provide an ultimate limit to this synchronisation technique. If we have a mobile transmitter (A) operating at an output frequency (f) of 1 GHz, communicating with a fixed base station (B), and the duration of the transmission is t seconds, then the number of waves emitted is ft and they cover a distance AB to the base station. If the wavelength is λ:

$$AB = ft\lambda \qquad \text{.....(26)}$$

If the mobile is moving with velocity v m/s towards the base station, then it will have moved to A′ in t seconds and the waves emitted will have a shorter wavelength, λ':

$$A'B = ft\lambda' \qquad \text{.....(27)}$$

Distance $AB = A'B + vt$

$$ft\lambda = ft\lambda' + vt$$

$$\lambda = \lambda' + v/f \qquad \text{.....(28)}$$

The ratio of the wavelengths

$$\lambda/\lambda' = 1 + v/f.\lambda'$$

Now $v = f.\lambda = c$ (c in free space) and the ratio of the frequencies;

$$f'/f = 1 + v/f\lambda'$$

$$(f'-f)/f = v/f\lambda'$$

$$(f'-f) = vf/c \quad = \text{Doppler frequency.} \qquad \text{.....(29)}$$

If we take the frequency as 1 GHz and velocity as 70 mph, approximately 31 m/s, then the Doppler frequency works out at 103 Hz or 0.1 ppm, within the tolerance limits using the coding scheme described later in this thesis. However if the transmitter was airborne, or on a satellite, then the relative speeds would be orders of magnitude greater and the Doppler effect would limit the operation of the system. Possible improvements would be a tracking of g values and implementing some form of compensation circuitry at the transmitters.

The other possible problem area when using a synchronisation recovery system would be multipath propagation. This is the arrival at the receiver of signals that have travelled there by different paths. The multipath problem is noticeable on broadcast television and is seen as the ghost effect on television screens; commercial digital radio also suffers quite badly from multipath effects. Multipath could cause problems with the CV-CCMA system in a number of areas. The synchronisation could be compromised by the recovery circuit locking onto a signal that has arrived via a different route. This is quite likely to occur in the type 3 synchronisation system as the frequencies used for synchronisation and data are quite different (10 MHz as opposed to 850 MHz). This could be reduced by careful design and/or using very high stability circuits with long sample times as the timing source. In view of the problems described, the first laboratory circuit was initially hard-wired together, a technique often called central synchronisation, in which all of the phase-locked loops were hard-wired to one master frequency source. This would not be viable in the field. Note that even with the type 3 synthesisers and the multi-sample software with filters applied, the oscillators must still be plesiochronous within tight constraints and also preferably locked to a recovered sync channel.

To try to reduce the sensitivity of the system to synchronisation impairments the characteristics of the filters used within the system were deliberately altered. This was in order to increase the window size of valid data at the receiver. With fixed g values, some improvement was obtained; some results are presented in Chapter 7. However, the increase in processing required to implement the filters reduces the transfer capacity of the system. The trade-off between different performance parameters is a common problem in modern communication systems. The optimum selection of codewords/synchronisation/power, etc., would depend on the overall system requirements. Also, further work was undertaken on the receive algorithm in order to reduce the sensitivity of the system to timing impairments. A multi-sample process was implemented. Although limited to 3 samples per symbol timing pulse, due to processor limitations, it did provide an improved system performance in the presence of applied jitter. The algorithm operated by outputting the majority decision (of the 3 samples), as to the correctly decoded data. If no agreement could be made, the result with the minimum distance to a known, pre-programmed value is output as the valid data. The software section of Chapter 6 contains flow charts, and explanations of the operation of the various systems developed. Another solution would have been to use the distance to projection (error) signal and chose the best, thus requiring only 2 samples.

5.3 Receiver processing

This section looks in more detail at the actual decoding process taking place in the receiver of the CV-CCMA, or more specifically in the signal processor within the receiver, where most of the 'intelligence' of the system resides. In a T-user

collaborative code we can represent the allowable combinations of codewords resulting from the T-user transmissions by:

$$\mathbf{A}_k = (a_{k1}, a_{k2}, \ldots a_{kn}) \text{ where } k = 1, 2 \ldots L, \qquad \ldots\ldots\ldots(30)$$

Here, L is the maximum number of allowable codeword combinations and a_{kj} is the j^{th} symbol of the k^{th} allowed codeword combination. Therefore, for each $\mathbf{A}_k$ there is a set of T symbols s_{ij} which defines a hyperplane described by g_i, (g_i is the complex loss coefficient from the i^{th} transmitter).

The distance squared between the received vector r and the k^{th} codeword combination can be written as;

$$\|d_k\|^2 = \sum_{j=1}^{n}\left(r_j - \sum_{i=1}^{T} s_{ij} g_i\right)^2 \qquad \ldots\ldots\ldots(31)$$

where s_{ij} is the j^{th} symbol of the i^{th} codeword. The estimated values of g_i can be found from minimising $\|d_k\|^2$ with respect to all g_i.

Using partial derivatives and equating to zero

$$\frac{\partial \|d_k\|^2}{\partial g_i}\Big|_{gi=Gi} = \sum_{j=1}^{n} 2 s_{ij}\left(-r_j + \sum_{i=1}^{T} s_{ij} G_i\right) = 0 \qquad \ldots\ldots\ldots(32)$$

This gives T equations which can be solved for T values of G_i and can be substituted back to find the distance squared to the k^{th} combination of codewords. This process is repeated for all values of k, and the combination of codewords with the minimum squared distance is chosen as the correct transmitted codewords.

In matrix form, where:

$\boldsymbol{I}$ = Identity matrix

$\boldsymbol{S_k}$ = The transmitted codeword combination matrix

$\boldsymbol{P_k}$ = Pseudoinverse of $\boldsymbol{S_k}$ (General Inverse in *Mathcad*)

$\boldsymbol{r}$ = The received vector

then $\boldsymbol{G} = \boldsymbol{P_k r}$ .....(33)

and (from (31)):

$$\|d_k\|^2 = \|r - S_k G\|^2 = \|r - S_k P_k r\|^2 = \|(I - S_k P_k) r\|^2 \quad \text{.....(34)}$$

Therefore the square of the distance can be calculated by multiplication of $\boldsymbol{r}$ by a fixed matrix. The value of k is estimated as that which minimises the expression. For a more rigorous treatment, including worked examples, refer to [74]. The first practical implementation of the CV-CCMA system used two transmitters and one receiver, and was constructed with the QPSK (quadrature phase shift keying) modulators restricted to two states, i.e. binary phase shift keying (BPSK). The initial simulations of the system are worked through to illustrate the theoretical operating principles. Note that all quantities in these expressions and simulations are, in general complex.

5.4 System simulations

To prove the viability of the system and to give confidence before hardware design commenced, the system operation, or more precisely the receiver decode system, was simulated using a proprietary mathematical program (*Mathcad*). With the modulation stage set to BPSK, a logic one is coded as a phase of 180 degrees and a logic zero coded as a phase of 0 degrees. Figure 5.1 recaps BPSK and QPSK.

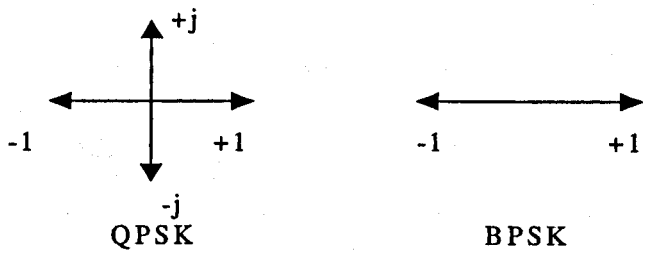


Figure 5.1 QPSK / BPSK phasor plots

N.B. The vectors rotate anti-clockwise by convention.

Table 5.1 summarises one set of codewords selected and the corresponding modulated output for the system, using length three codewords with BPSK modulation. Note that the codewords used are unique and they are partitioned between the two transmitters, enabling the identity of the transmitter to be determined as soon as the codeword is recovered at the receiver. This was a deliberate design feature of the first systems because fault finding / signal tracing is much simplified with unique and therefore easily identifiable transmission sources. In the initial implementation, the first phase state is the same in all codewords, and although this is inefficient, it aids the decode process because the decoder always has a fixed starting point.

Station ID	Data Input	Codeword output	Modulated output
User 1	0	0 0 0	+1 +1 +1
User 1	1	0 0 1	+1 +1 −1
User 2	0	0 1 0	+1 −1 +1
User 2	1	0 1 1	+1 −1 −1

Table 5.1 Codewords and modulated output for system 1

The combinational matrices are then calculated, using computer programs *Mathcad* and *Matlab*. Programs were written in both languages to solve the equations; see Appendix 2 for the programs and Chapter 7 for the programs and results. With binary inputs to both transmitters (0 or 1), four possible combinations of codewords are possible, ***S*1-*S*4,** and pseudo-inverses for these combinations are calculated, ***P*1-*P*4**:

$$S1=\begin{bmatrix}+1+1\\-1+1\\+1+1\end{bmatrix} \qquad S1.\text{Pseudoinverse} = P1=\begin{bmatrix}+0.25-0.5+0.25\\+0.25-0.5+0.25\end{bmatrix}$$

$$S2=\begin{bmatrix}+1+1\\-1+1\\+1-1\end{bmatrix} \qquad \text{S2.Pseudoinverse} = \text{P2}=\begin{bmatrix}+0.5-0.25+0.25\\+0.5+0.25-0.25\end{bmatrix}$$

$$S3=\begin{bmatrix}+1+1\\-1+1\\-1+1\end{bmatrix} \qquad S3.\text{Pseudoinverse} = P3=\begin{bmatrix}+0.5-0.25-0.25\\+0.5+0.25+0.25\end{bmatrix}$$

$$S4=\begin{bmatrix}+1+1\\-1+1\\-1-1\end{bmatrix} \qquad S4.\text{Pseudoinverse} = P4=\begin{bmatrix}+0.25-0.5-0.25\\+0.25+0.5-0.25\end{bmatrix}$$

Next, the combinational matrices are calculated.

$$C1=(I-S_kP_k)_{00}=\begin{bmatrix}100\\010\\001\end{bmatrix}-\begin{bmatrix}+1+1\\-1+1\\+1+1\end{bmatrix}\begin{bmatrix}+0.25-0.5+0.25\\+0.25+0.5+0.25\end{bmatrix}=\begin{bmatrix}0.5&0&-0.5\\0&0&0\\-0.5&0&0.5\end{bmatrix}$$

$$C2=(I-S_kP_k)_{01}=\begin{bmatrix}100\\010\\001\end{bmatrix}-\begin{bmatrix}+1+1\\-1+1\\+1-1\end{bmatrix}\begin{bmatrix}+0.5-0.25+0.25\\+0.5+0.25-0.25\end{bmatrix}=\begin{bmatrix}0&0&0\\0&0.5&0.5\\0&0.5&0.5\end{bmatrix}$$

$$C3=(I-S_kP_k)_{10}=\begin{bmatrix}100\\010\\001\end{bmatrix}-\begin{bmatrix}+1+1\\-1+1\\-1+1\end{bmatrix}\begin{bmatrix}+0.5-0.25-0.25\\+0.5+0.25+0.25\end{bmatrix}=\begin{bmatrix}0&0&0\\0&0.5&-0.5\\0&-0.5&0.5\end{bmatrix}$$

$$C4=(I-S_kP_k)_{11}=\begin{bmatrix}100\\010\\001\end{bmatrix}-\begin{bmatrix}+1+1\\-1+1\\-1-1\end{bmatrix}\begin{bmatrix}+0.25-0.5-0.25\\+0.25+0.5-0.25\end{bmatrix}=\begin{bmatrix}0.5&0&0.5\\0&0&0\\0.5&0&0.5\end{bmatrix}$$

These results are stored in the DSP (manually input into an assembler array). To illustrate system operation with an example, assume complex loss component 1 has attenuation but no phase change, $g_1 = 0.5 + 0j$; and complex loss 2 has both attenuation and phase change, $g_2 = 0.9 + 0.2j$. The received vector $\boldsymbol{R1}$ will be:

$$\mathrm{R1} = 0.5\begin{bmatrix}1\\1\\1\end{bmatrix} + (0.9+0.2j)\begin{bmatrix}+1\\-1\\+1\end{bmatrix}$$

This vector is multiplied by the pre-calculated combinational matrices, the Euclidean norm of the result is calculated and the minimum value assumed to equate to the valid combination of original codewords. Program (*Paul3.mcd*) in Appendix 2 performs these operations and can be used to try out all combinations (can be used for test purposes). Next the Euclidean distances ($\boldsymbol{D1}$-$\boldsymbol{D4}$) are calculated by multiplying the received vector by the $\boldsymbol{I}$-$\boldsymbol{S_kP_k}$ matrix and hence the minimum distance can be found.

Distance 1 =$\boldsymbol{D1} = \boldsymbol{C1}*\boldsymbol{r}$

$$\boldsymbol{D1} = \begin{bmatrix}0.5 & 0 & -0.5\\ 0 & 0 & 0\\ -0.5 & 0 & 0.5\end{bmatrix}\begin{bmatrix}+1.4+0.2\mathrm{j}\\ -0.4-0.2\mathrm{j}\\ +1.4+0.2\mathrm{j}\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$|\boldsymbol{D1}| = 0$$

Distance 2 =$\boldsymbol{D2} = \boldsymbol{C2}*\boldsymbol{r}$

$$\boldsymbol{D2} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0.5 & 0.5\\ 0 & 0.5 & 0.5\end{bmatrix}\begin{bmatrix}+1.4+0.2\mathrm{j}\\ -0.4-0.2\mathrm{j}\\ +1.4+0.2\mathrm{j}\end{bmatrix} = \begin{bmatrix}0\\0.5\\0.5\end{bmatrix}$$

$$|\boldsymbol{D2}| = 0.707$$

$$\boldsymbol{D3} = \begin{bmatrix}0 & 0 & 0\\ 0 & 0.5 & -0.5\\ 0 & -0.5 & 0.5\end{bmatrix}\begin{bmatrix}+1.4+0.2\mathrm{j}\\ -0.4-0.2\mathrm{j}\\ +1.4+0.2\mathrm{j}\end{bmatrix}$$

$$\boldsymbol{D3} = \begin{bmatrix}0\\ -0.9-0.2\mathrm{j}\\ 0.9+0.2\mathrm{j}\end{bmatrix}$$

$$|\boldsymbol{D3}| = 1.304$$

$$\boldsymbol{D4} = \begin{bmatrix}0.5 & 0 & -0.5\\ 0 & 0 & 0\\ -0.5 & 0 & 0.5\end{bmatrix}\begin{bmatrix}+1.4+0.2\mathrm{j}\\ -0.4-0.2\mathrm{j}\\ +1.4+0.2\mathrm{j}\end{bmatrix}$$

$$\boldsymbol{D4} = \begin{bmatrix}1.4+0.2\mathrm{j}\\ 0\\ 1.4+0.2\mathrm{j}\end{bmatrix}$$

$$|\boldsymbol{D4}| = 2$$

$|D1|$ is the minimum distance; therefore the receiver selects codewords (111) and (1-11) as the correct transmitted codewords and outputs (0,0) as the decoded data from the two transmitters. The g values can also be output if required for further processing.

$$g = \begin{bmatrix} 0.25 & -0.5 & 0.25 \\ 0.25 & 0.5 & 0.25 \end{bmatrix} \begin{bmatrix} 1.4+0.2j \\ -0.4-0.2j \\ 1.4+0.2j \end{bmatrix}$$

$$g = \begin{bmatrix} 0.9+0.2j \\ 0.5 \end{bmatrix}$$

The results show that the system decode process is a viable technique for the reception of at least two CV-CCMA transmitters operating in the same bandwidth. Chapter 7 contains details of the actual results obtained with the laboratory implementation of the system, and the receiver decode simulation just described matches the actual results quite closely. The DSP code written for the first test system is an assembler language version of the *Mathcad* program and so is not as easy to interpret, but the code is sectioned to follow the same procedure; refer to Appendix 3 to view the code.

5.5 Codewords

In the original conception of CV-CCMA, codewords were required to be linearly independent, partitioned into unique sets, one for each of the transmitters, but operating an algorithm where any number of combined user unique codewords could (theoretically) be decoded. Although this gave poor capacity/bandwidth characteristics, it made the practicalities of actually getting it to work easier. This system was developed and is currently at a breadboard stage. With the system up and

running it was decided to produce another system that was more readily reconfigured and upgraded. The first task was to investigate the possibility of reducing the restrictions on the generation of codewords, that is the requirement for linear independence, which substantially increases the coding rate, particularly with short codes and a restricted number of users. In all systems, signals from both transmitters are summed by the channel, having been multiplied by complex gain coefficients g_i to give the combined signals:

$$R_n = \sum_{i=1}^{T} s_{in} g_i$$
$$for$$
$$1 \le n \le N$$
.....(35)

where N is the length of the codeword and S_{in} is the nth symbol of the codeword sent by the ith of a set of T transmitters. The locus described by the vector R_n forms a subspace of T complex dimensions as the g_i are varied. Since g_i may be zero these hyperplanes must go through the origin. The reception process involves the calculation of distances from these hyperplanes. To be able to do this it is necessary that $T < N$, as if $T = N$ the subspace includes the whole vector space, and the distances are all zero. This can be demonstrated easily using program *paul2.mcd* in Appendix 3.

The greater the difference between T and N the less the likelihood of error. However, to uniquely decode the received signal the hyperplanes only need to be distinct. If two combinations of codewords give identical hyperplanes, then both of the vectors defining the first combination would have to be a linear combination of the second pair of vectors. Therefore, it is logical that for unique decodability the codewords do not lie on any hyperplane defined by the combination of any pair of

other codes. This provides much less restriction on the choice of codewords than full linear independence, as required in the initial implementation. In fact, if we could remove the restrictions on the symbol sequence altogether and insert identity codes into the streams we would also remove the need for frame timing, although accurate bit timing would always be required. Detection could be performed on an arbitrary sequence of QPSK symbols with decoding using overlapping codewords by considering three-symbol CV-CCMA codewords. This will then define the set of codewords to be searched in the next step; effectively each three-symbol codeword would be selected from sixteen or four of the sixty-four possible codewords for which the first symbol or pair of symbols were the same as the last ones. This would give us a processing advantage in performing the CV-CCMA decoding.

As an example to illustrate the relaxation on codeword restriction, consider the first implementation equipment, with two transmitters and only one receiver ($T = 2$), and with the modulation restricted to BPSK and codewords of length $N = 3$ symbols, which are selected from (+1 –1). This gives us $2^3 = 8$ possible codewords, as illustrated in figure 5.2.

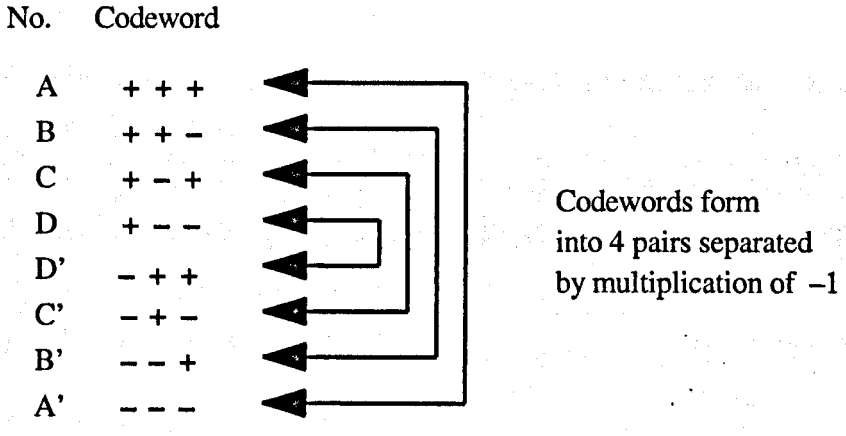


Figure 5.2 Codeword generation

These codewords form a set of vectors pointing to each of the vertices of a cube based around the origin. We eliminate those that are negatives of one another as these would be the same sign if $\boldsymbol{g}$ were reversed, leaving four (1,1,1), (1,1,–1), (1,–1,1) (1,–1,–1) vectors, none of which can be formed from any other two. This shows that with two users and the codewords described, the combined codewords, (A+B), (A+C), (B+D), (C+D) produce unique combinations which can be resolved even with identical $\boldsymbol{g}$; (A+D) and (B+C) give identical combinations, which could still be resolved with different $\boldsymbol{g}$. Figure 5.3 shows all possible combinations and indicates that the restriction on choice of codeword is less than full linear independence, as was previously stated as required for CV-CCMA.

Note, however, any codeword can be formed from the other three. We now advance the argument with a second example, using the later equipment and removing the restriction on the modulators, thus giving full QPSK with symbols selected from (+1,–1,+j,–j). Again we have two transmitters, one receiver and codewords of length $N = 3$. In this case we have $4^3 = 64$ possible codewords but we can only select codewords that are not scaled versions of one another, leaving $4^{3-1} =$ 16 possible codewords. If there are M symbols in the alphabet there are M^{N-1} codewords that are not scaled versions of others.

One system implemented for making the selection of codewords is to make the same choice of first symbol for all the codewords, as shown in the previous example (combinations grouped into sets). This also gives the advantage of simplifying the decode algorithm, as previously explained.

Codeword Set 1	Codeword Set 2
A = 1 1 1	A' = –1 –1 –1
B = 1 1 –1	B' = –1 –1 1
C = 1 –1 1	C' = –1 1 –1
D = 1 –1 –1	D' = –1 1 1

Possible combinations (ignoring zero solutions)

A B	2	2	0		A' B'	-2	-2	0	
A C	2	0	2		A' C'	-2	0	-2	
A D	2	0	0	i	A' D'	-2	0	0	ii
B C	2	0	0	i	B' C'	-2	0	0	ii
B D	2	0	-2		B' D'	-2	0	2	
C D	2	-2	0		C' D'	-2	2	0	
A B'	0	0	2	iv					
A C'	0	2	0	iii					
A D'	0	2	2						
B A'	0	0	-2	vi					
B C'	0	2	-2						
B D'	0	2	0	iii					
C A'	0	-2	0	v					
C B'	0	-2	2						
C D'	0	0	2	iv					
D A'	0	-2	-2						
D B'	0	-2	0	v					
D C'	0	0	-2	vi					

Note : Combinations indicated by roman numerals
The original codewords cannot be made from any combination of any other two

Figure 5.3 BPSK possible codeword combinations

It may still be possible that:

$$s_k = Cs_l + Ds_m \qquad \text{.....(36)}$$

For this to happen:

$$C\frac{S_{lr}}{S_{kr}} + D\frac{S_{mr}}{S_{kr}} = 1 \qquad \text{.....(37)}$$

for $r = 1, 2, 3$ such that $C \neq 0$ and $D \neq 0$

If $\alpha_r = S_{lr} / S_{kr} \in \{1, j, -1, -j\}$ and $\beta_r = S_{mr} / S_{mr} \in \{1, j, -1, -j\}$ this can be rewritten as:

$$\begin{aligned} C + D &= 1 \\ C\alpha_2 + D\beta_2 &= 1 \\ C\alpha_3 + D\beta_3 &= 1 \end{aligned}$$

Rewriting:

$$\begin{bmatrix} C \\ D \end{bmatrix} = \frac{\begin{bmatrix} \beta_3 - \alpha_3 \\ -\beta_2 + \alpha_2 \end{bmatrix}}{\alpha_2\beta_3 - \alpha_3\beta_2} \qquad \text{.....(38)}$$

Substituting back gives us the condition for a set of three codewords to be unacceptable:

$$\alpha_2 - \alpha_3 + \beta_2 + \beta_3 = \alpha_2\beta_3 - \alpha_3\beta_2 \qquad \text{.....(39)}$$

Thus each of the products on the right-hand side must be $\in \{1,j,-1,-j\}$ so the whole of the right-hand side must be $\in \{2,-2,1+j,-1-j,1-j,j-1,0,-1-j,2j,-2j\}$, and the left-hand side must be $\in \{4,2,0,-2,-4,3+j,3-j,2+2j,2-2j,1+3j,1-3j,-1+3j,-1-3j,4j,-4j,2j,-2j\}$. The only common elements are $\{2,-2,0,2j,-2j\}$. This implies $\alpha_2 = \alpha_3$ or $\beta_2 = \beta_3$ corresponding to $C = 0$ or $D = 0$ or both are zero and thus all are trivial. This proves the set of codewords consisting of three QPSK symbols with the first symbol fixed is an allowed codeset for CV-CCMA.

The two examples described show that the previously-stated restriction on codewords of linear independence is not required in order for CV-CCMA to work. The combined codewords are shown to be able to be individually decoded. Figure 5.4 illustrates in a block diagram form the CV-CCMA system with two transmitters and one receiver. The processes that have been described can be pictured fitting into the component blocks of the diagram.

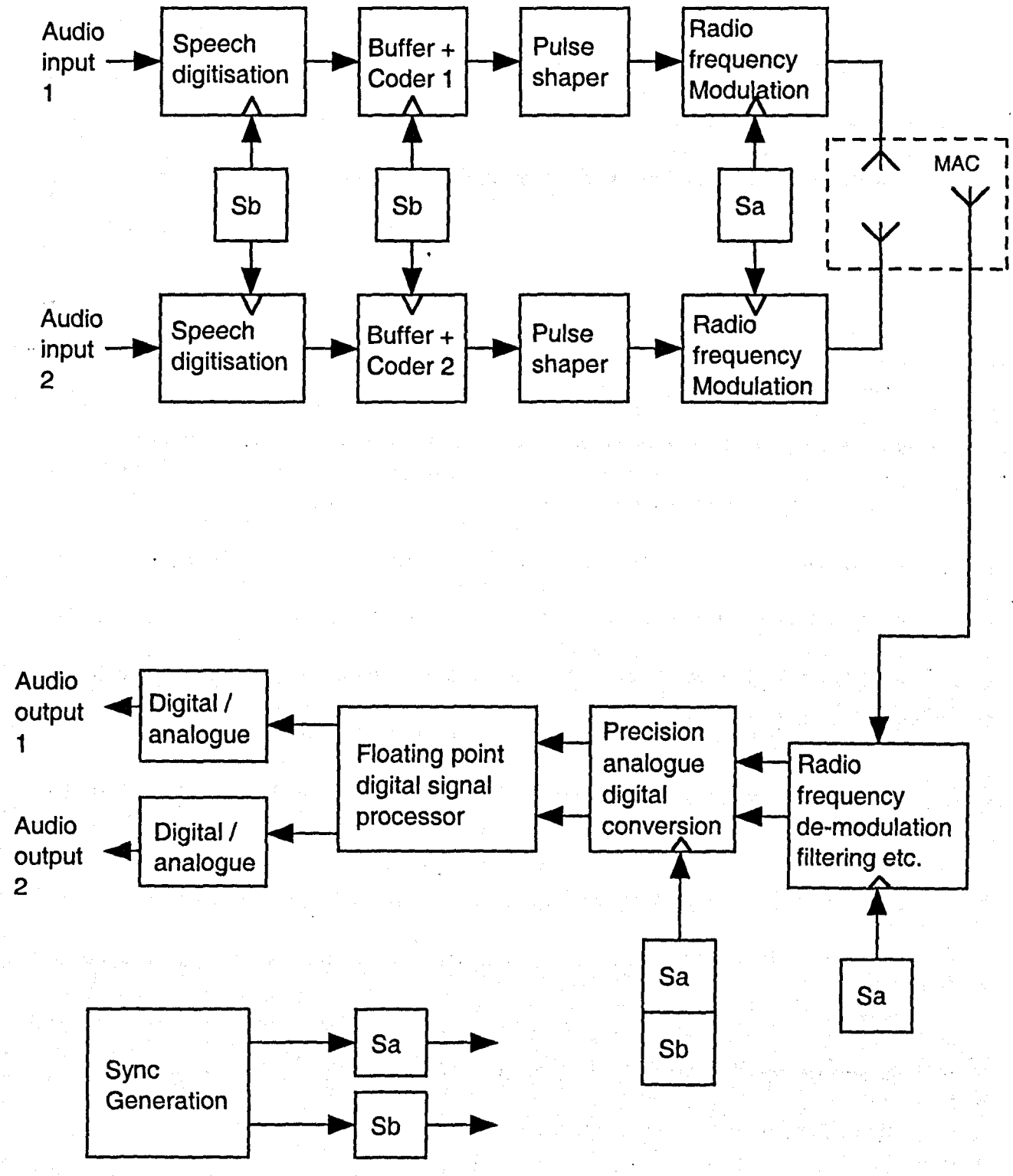


Figure 5.4 Block diagram of CV-CCMA demonstrator

5.6 Convolutional coding

Convolutional coding is a channel-coding technique that is mainly used where high rates of error correction are required, for example with satellites / mobiles etc., due to the slow degradation in these environments. It is essentially discrete convolution, that is the superposition of appropriately weighted and delayed unit response sequences.

If y = output, x = input and h = unit response:

$$y = \sum_{k=0}^{i} x_k h_{i-k} \qquad \text{.....(40)}$$

At the transmitter, bits are added into the output stream and the output becomes restricted to follow a certain pattern of changes. At the receiver, if there is a deviation from this sequence of changes (i.e. errors are received) then, in the maximum likelihood demodulator, the receiver can back-track and restore the original sequence of changes, usually called the 'maximum likelihood' path. The CV-CCMA system could use a 'version' of convolutional coding, or more particularly the reception process, to help cope with error propagation problems by storing the received signals and comparing the sequence with a known good sequence, and / or using the sequence to predict where the next codeword will be.

Convolutional encoders are simple to implement. The data to be encoded is loaded into a shift register, and the number of stages in the shift register is called K, the constraint length of the code. The shift register is tapped at various points depending on the code polynomials used, and these sets of taps are chosen according to the coding rate required. This is usually expressed as a fraction; for example R = ½ means one bit is shifted into the register per encoder cycle and two is the number of code polynomials generated. The sets of tapped data from the shift registers are then modulo two added, that is they are fed through an exclusive OR gate and the output of the gate forms one of the encoder outputs. Now at the receiver /

decoder the data is compared to locally generated 'hypotheses'. Then, because of the additional information encoded we can generate a metric which gives an estimate of the closeness of fit of the received data to the calculated metric. This may be just a single value or may consist of a number of computed metrics, with the final decoded data being selected from the nearest solution to a valid sequence. The decoder investigates all the possible user data sequences and picks the closest match; certain sequences cannot belong to the 'maximum likelihood' path and are therefore discarded. The complexity of the decoder increases exponentially with the constraint length K so there is obviously a trade-off between better noise performance with the larger length codes and processing power. NASA implemented a $R = ½$, $K = 7$ encoder on the *Voyager* probes, a typical compromise.

A variation of the technique is being used in an experimental radio fast packet (asynchronous transfer mode) system at Cable and Wireless PLC. The whole packet of information is buffered and the most likely path selected before passing the data on. Systems that run very high error rates use convolution coding together with different types of block coding to cope with large burst errors. The simple hardware implementation is also a major advantage.

Some time was spent considering the possibility of adapting convolutional coding into the CV-CCMA system. This would be done by adding circuitry to generate the correct sequences of codewords at the encoders. The receiver processor would have the correct sequences programmed and sufficient memory to operate correctly. The system could also be altered to operate in a 'cell' mode: with the cell size kept small the decoder could operate on the entire cell before a good/bad/best effort decision is made. If a CV-CCMA network were to be constructed then this more secure channel could be overlaid onto the system and used as a

network manager channel. However, a substantial increase in processing would be required but this is another very interesting area and hopefully will be pursued in the future.

Implementation over a CV-CCMA link would be beneficial because it would make the system more robust under the influence of channel impairments. However, apart from simulation data (see Chapter 6), it was not possible to test the convolutional coding implementation due to processor limitations. The Texas DSP integrated circuit was running at near capacity with the standard decode functions already implemented.

Chapter 6 details the actual implementation of the system at the University of Warwick and Cable and Wireless PLC.

6 IMPLEMENTATION

6.1 General implementation

Working on the implementation of the CV-CCMA system has occupied a sizeable amount of time. A number of novel circuit techniques have been developed, particularly regarding the codeword generation/recovery and synchronisation of the CV-CCMA system. The implementation of the digital signal processor and its associated interfacing circuitry followed a more conventional approach. In an effort to keep costs down, wherever possible integrated circuits were used that were intended for the GSM (global system mobile) market; no custom or semi-custom integrated circuits were used and standard two-layer printed circuit boards were used in all of the designs.

The design of the circuits involved using computer-aided design techniques. *Spice* (simulation program integrated circuit emphasis) was used as a general purpose nodal circuit simulation tool. *Compact* was used to simulate the radio frequency circuits, in particular the high-frequency components and the modulators. The programs and results are included in Appendix 4. The board designs (which had to be surface-mounted due to the high frequencies involved) were generated using *Protel* software, an advanced electronic circuit design package, and produced using lithographic equipment at Cable and Wireless PLC. *Protel* was also used to simulate the combinational logic used in the transmitters. It was decided at an early stage to try to produce 'functional units' that could be used as component parts in the receivers, the transmitters and the control circuitry throughout the entire CV-CCMA system. It was envisaged that this approach would keep costs down, and this has proved to be the case.

6.1.2 System control

A number of the integrated circuits that were investigated for possible use in the system had high-speed control circuitry built into them, to enable the control of circuit operation by a microprocessor or other controlling system. This is particularly useful with dynamic circuits, such as frequency synthesisers that require rapid changes in output frequency, or to shut down amplifiers in order to conserve battery power. It was decided to design each of the 'functional units' so as to be controllable via a standard bus. The interconnecting bus chosen was a Philips development bus called the I^2C bus or inter-integrated circuit bus. The I^2C bus was introduced approximately twenty years ago and is a high-speed serial bus designed for implementation in silicon and aimed at reducing the chip area used by conventional high-speed parallel bus architectures. The bus is specified to bi-directional data rates of 400 Kb/s but can function well up to ~0.5 Mb/s. The higher speeds were not required in the CV-CCMA system, but would be required if the bus was to be used to control the other component parts of a radio 'node'. The bus has only three connections: clock, data and ground. Each device on the bus has its own unique address and collision detection and arbitration is done in software. Each device connected to the bus is software-addressable and there is a master-slave relationship between devices connected to the bus at all times. A master is a device which initiates a data transfer on the bus and generates the clock signals to correctly time the data transfer. The device addressed is the slave. There are unique conditions that signify the start (clock = high, data = high to low) and stop (clock = high, data = low to high).

All data on the bus is in a serial format with 8-bit bytes. The address is transmitted first by the master, the slave responds with an acknowledge and then data transfer can take place. There is no limit to data length so a timer was incorporated in

the programming software to prevent lock-up problems. There is no limit as to minimum operating speed so a program was written to output serial data from the printer port of a portable personal computer, and an interface designed to buffer the signals, level shift and prevent contention on the bus. The program, circuit diagram and operating notes are included in Appendix 5 and a reduced circuit is reproduced in Figure 6.1. The interface and software have been used in a number of test functions, and specifically used in the CV-CCMA to program the memory chips containing the test patterns that are used at initial set-up.

At this stage Philips Electronics offered reduced-cost integrated circuits, so it was decided to use these components in the radio frequency circuitry as well as the synthesiser.

The integrated circuits were all interconnected using the I^2C bus and some memory chips were also controlled via the bus, so this meant frequency and codeword shifting could all be done from a portable computer. The computer interface worked well and was used in the bench-test implementations. However, to make the system more portable a new control circuit was designed and made. The new circuit was stand-alone and used a PCF8584 chip specifically for the control of the I^2C bus. This proved to be very difficult to program as the in-built memory could only be written to a few times before it locked up. The required programming pulses were of a very short duration and quite high voltage.

Thus, the circuit was re-designed around a more general purpose controller chip, the 87C750, which proved to be easier to program, was cheaper and more reliable. The circuit diagram (simplified) is reproduced in Figure 6.2 and the full circuit and control software is included in Appendix 6.

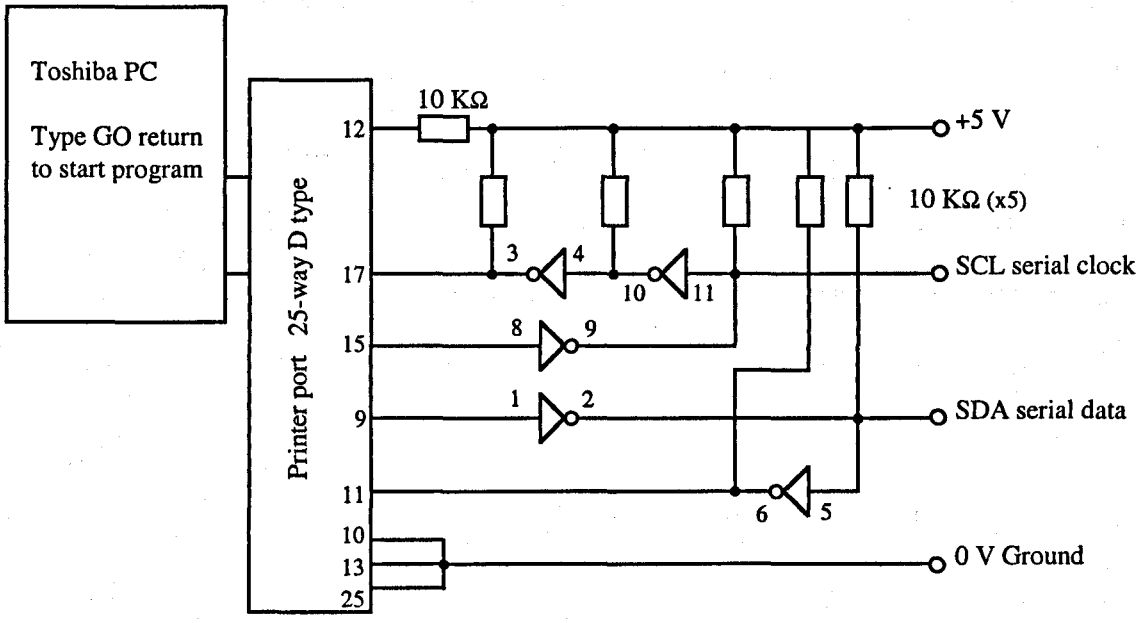


Figure 6.1 I²C bus PC interface
(Pin out for 74LS05)

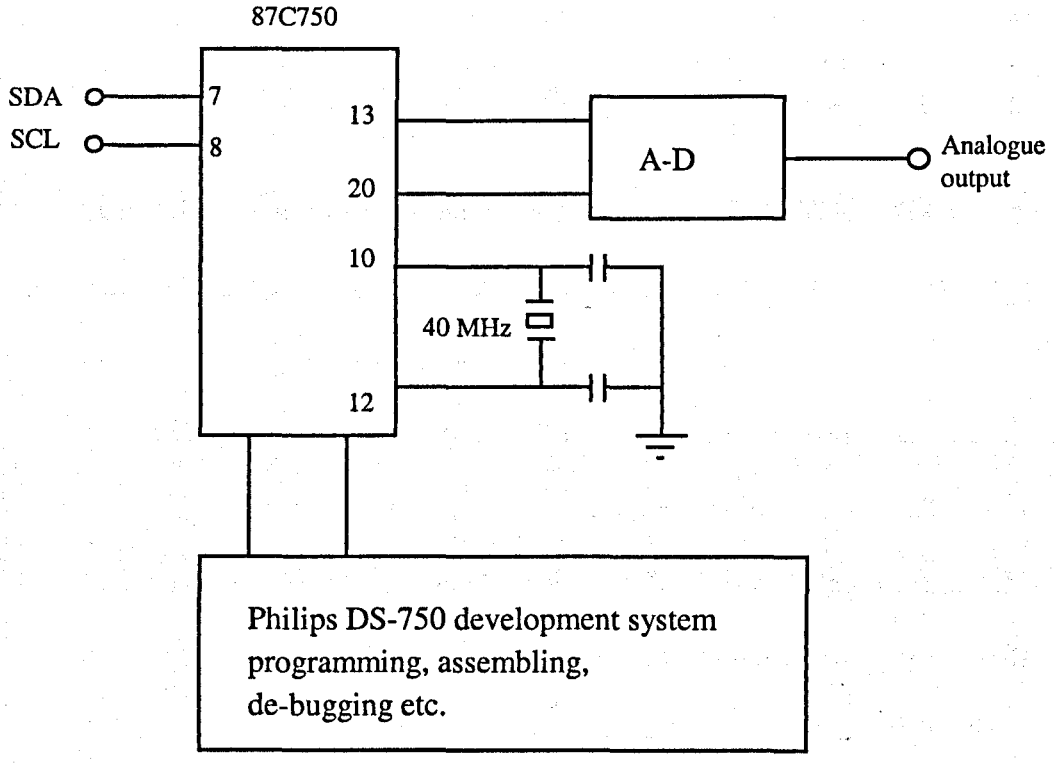


Figure 6.2 I²C bus microcontroller interface

The only additional complexity and expense was the requirement to use the Philips DS750 development system to program the controller. This prevented contentions arising between the bus driver integrated circuits. Using the storage oscilloscope to view the data lines showed a number of handshake sequences which were not detailed in the literature. These sequences were required by the controller chip to make sure it was operating in the correct mode. All of the sequences were implemented with the DS750 I^2C driver software.

6.1.3 Synthesisers

The first stage in the design involved researching the current state-of-the-art synthesiser technology. It appeared that the small level of drift required for the system to operate could be achieved by independent synthesisers, as long as they were never turned off and were regularily recalibrated (every few months). This was deemed to be impractical in the field and so it would be necessary to recover timing and/or lock to a central source, for example the GPS (global positioning system) satellites. The synthesisers are an important component of the CV-CCMA system; they supply the control and timing signals for the transmitters/receivers and, once locked together, cannot be allowed to drift more than a bit period per symbol message. Theoretically only a very small amount of drift can be tolerated between transmitters and receivers. Hence one of the most expensive components in the CV-CCMA is the TCXO (temperature controlled crystal oscillator). This is used as a reference for all of the frequency-generating components within the system. The drift coefficients of the TCXOs are restricted to fractions of a part per million - annually. A number of synchronisation techniques were investigated, using clock-recovery schemes from pilot channels, etc. However, for speed of implementation the initial (X) models were

constructed using a hardwired central clock system, that is all the frequency-generating circuits were locked to a single source via a coaxial cable. Once this system was working the other techniques were investigated. The initial synchronisation system, 'type 1', is shown in Figure 6.3.

The first enhancement was a clock recovery scheme based on the 'ethernet' system, called a 'type 2' system. This involved setting the CV-CCMA system up with a single timing master source, at a designated master node, with the rest of the nodes recovering timing from it, and being designated slave nodes. The initial timing sequence is then used to lock the synthesisers together. The master station will then continue to transmit bursts of timing synchronisation sequences at regular intervals. In a lightly loaded network this would be controlled by a timer, and, in a heavily loaded network during gaps in the transmission of data. The data flow across the network would be in a 'packet' format, ideally of length 53 bytes. This would enable the use of standard high-speed ATM integrated circuits, thus reducing costs.

Some time was spent working on the implementation of the type 2 timing system. All the nodes had to be converted to transceivers, with the transmit/receive function and the synthesiser master inputs controlled via the I^2C bus. This proved difficult to achieve with the simple controller, really requiring a fully compatible microprocessor to run the system as a background operation. Also, the 'packets' of information and/or 'packets' of timing synchronisation needed to be identified with unique headers within the 'packet'. This proved difficult to implement. The SRTS (serial residual time stamp) system, as used in ATM, could not be translated to CV-CCMA due to the high CDV (cell delay variation) figures. A more reliable detection process is required. In view of time constraints it was decided to

progress with the type 3 system, as the implementation is simpler. The type 2 system is sound in theory, but proved difficult to implement with the limited facilities available. The type 2 system is shown in Figure 6.4.

The third enhancement to the synchronisation system involved setting up a dedicated synchronisation channel, running alongside the data channel. This system is called the 'type 3' system. The synthesisers used by the CV-CCMA are PLL (phase-locked loop) types that are referenced to a 10 MHz TCXO. In the type 3 system, one node is designated a master, and at this node the 10 MHz signal is buffered, amplified and transmitted. All other nodes are then designated slave nodes and are fitted with a radio receiver to recover the 10 MHz signal. This recovered 10 MHz signal is then used to phase-lock the slave nodes to the master node. The receivers used are hybrid devices, modified to operate at 10 MHz and originally used for the remote control of garage doors etc. The system was implemented into the demonstrator and worked satisfactorily. The type 3 synchronisation system is shown in Figure 6.5.

However, further development of the system is required. The frequency of the timing channel needs to be moved closer to the data channel frequency. This would mean that the two signals would suffer similar impairments (loss/phase shift) between the transmitter and receiver. Also, to ensure the system can be used legally, the frequency needs to be moved into a licenced band. The synchronisation circuitry could be re-designed to fit within a 'timing' module. This module could be shielded from external interference with a small Faraday cage. Some equalisation could be applied at the transmitters, see Chapter 8 for discussion of this. Results obtained with the type 3 system are reproduced in Chapter 6.

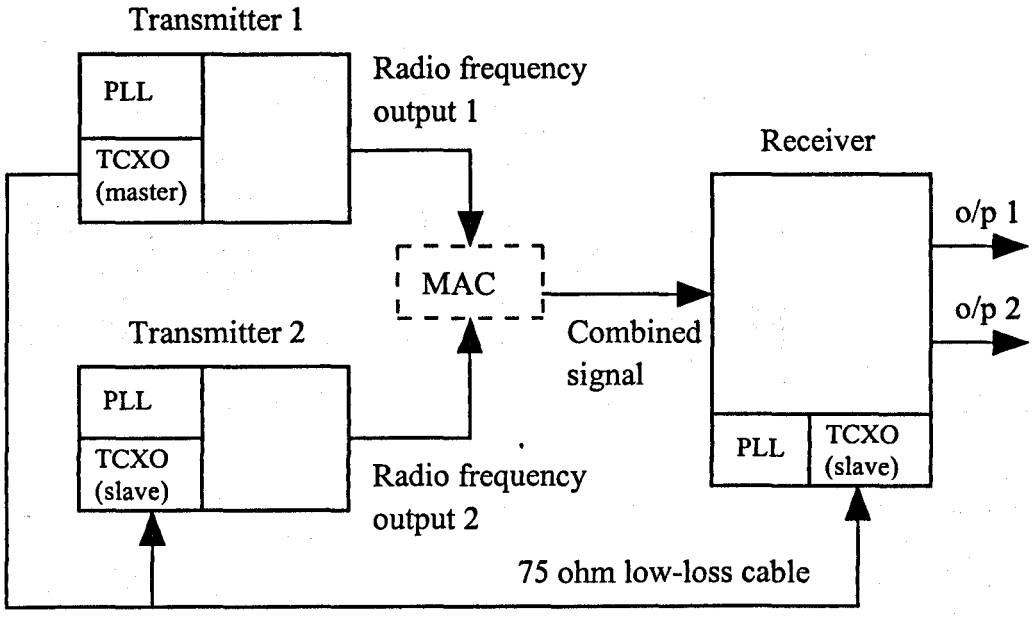
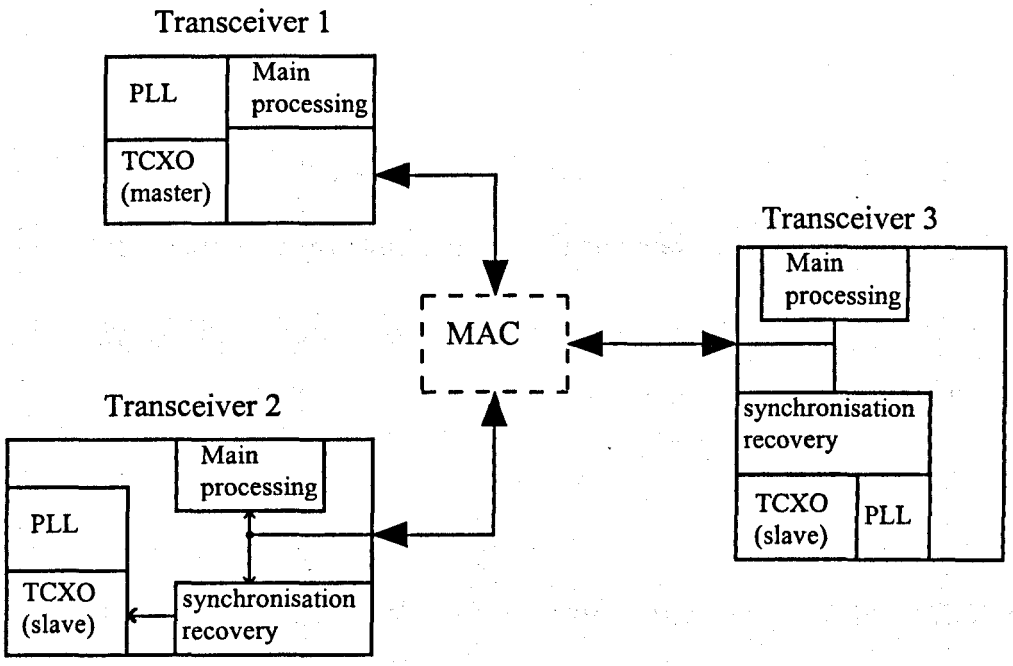


Figure 6.3 Type 1 synthesiser timing



Notes:

Data is formed into packets, fixed length (53 bytes) comprising header (5 bytes) for synchronisation / addressing and data (48 bytes) Timing packets are generated by master node. Decoding of sync/data by standard header error checksum in byte 5 of the header.

Figure 6.4 Type 2 synthesiser timing

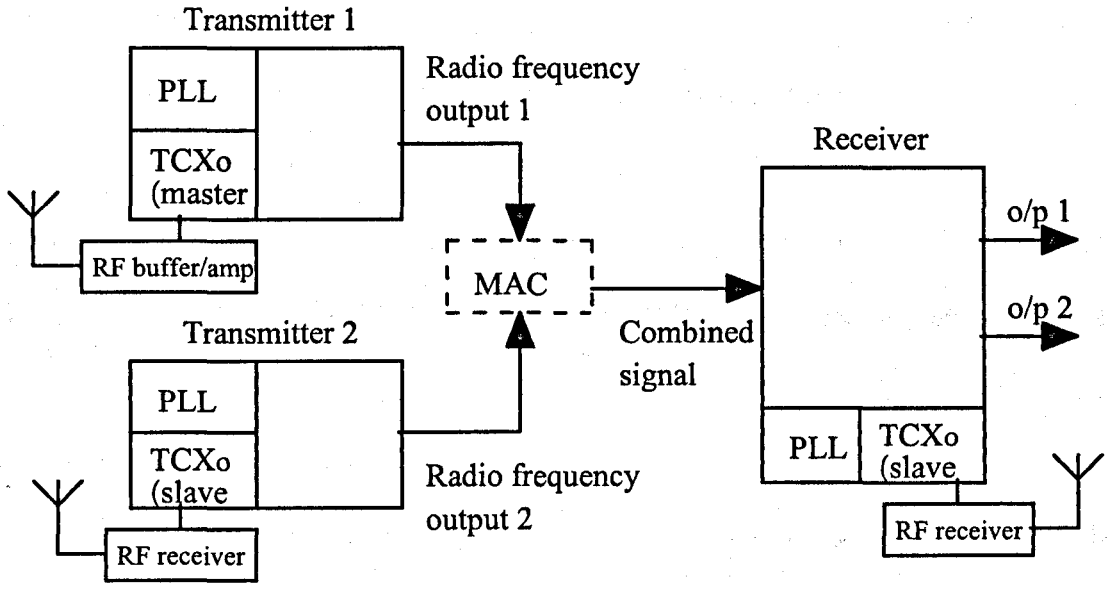


Figure 6.5 Type 3 synthesiser timing

It was decided to construct a phase-locked loop (PLL) type synthesiser as this would give the best option for good frequency stability due to the outputs being referenced to a standard very high-stability TCXO. This would also mean that the output frequency could be quickly and easily altered, simply by changing the division ratio in the feedback loop. With the advent of high-speed, cheap logic circuitry and modular radio frequency components the phase-locked loop has become the standard circuit used for frequency generation. The basic PLL consists of three components connected in a feedback configuration, a voltage-controlled oscillator (VCO), a phase detector and a low-pass filter. Figure 6.6 shows the block diagram of the basic PLL.

The VCO produces a periodic output waveform, not necessarily sinusoidal, the frequency of which may be varied about some free-running frequency,

Fo , according to the value of the applied voltage *Va* . The phase detector produces an output *Vb* that is a function of the phase difference between the input signal *Vin* and the VCO output signal *Vo* The filtered signal *Va* , which will be a slowly varying DC waveform due to the low-pass filter, is the control signal that is used to control the output frequency of the VCO. The output of the VCO will effectively be the same as the input signal, and if the input signal varies, the VCO will track the changes; in this state the PLL is locked to the input waveform.

In order to vary the output frequency of the VCO, without varying the input or reference frequency, it is necessary to alter the frequency of the feedback signal, and/or the frequency of the reference input to the phase comparator. As long as the phase comparison is still valid, that is the two waveforms are harmonically related in some way, the phase lock will be preserved and the output frequency will be stable. This technique is used in the CV-CCMA synthesisers; both the feedback signal and the reference signal are fed to programmable dividers which can be set to match the signals together. Figure 6.7 illustrates the CV-CCMA synthesiser operation in block-diagram format. The Philips phase detector used in the circuit not only outputs pulses of current to charge the capacitor in the low-pass filter up, but also sinks current, thus enabling the voltage on the input to the VCO to be rapidly altered.

A number of integrated circuits were obtained and tested against one another in terms of ease of construction, spectral purity, drift and ease of programming. A Philips UMA1014 synthesiser chip came out ahead and was built into a breadboard with the other components. The integrated circuit operates using the Philips I^2C (inter-integrated circuit) patented system; this restricted the choice of the other components used in the design. [116]. The integrated circuit is manufactured in bipolar technology but has a power-down mode for saving power in an idle state.

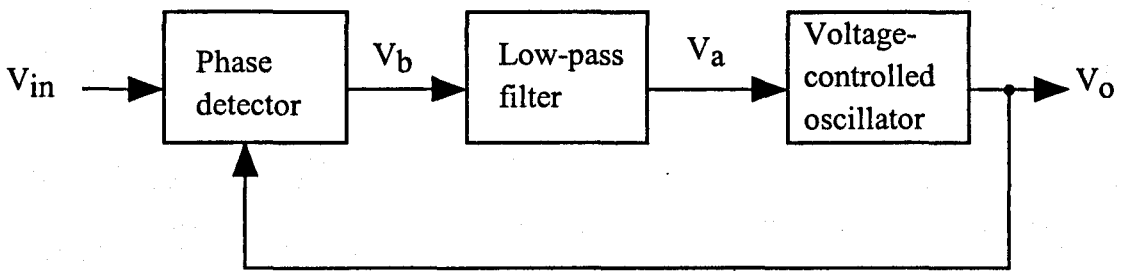


Figure 6.6 Basic phase locked loop

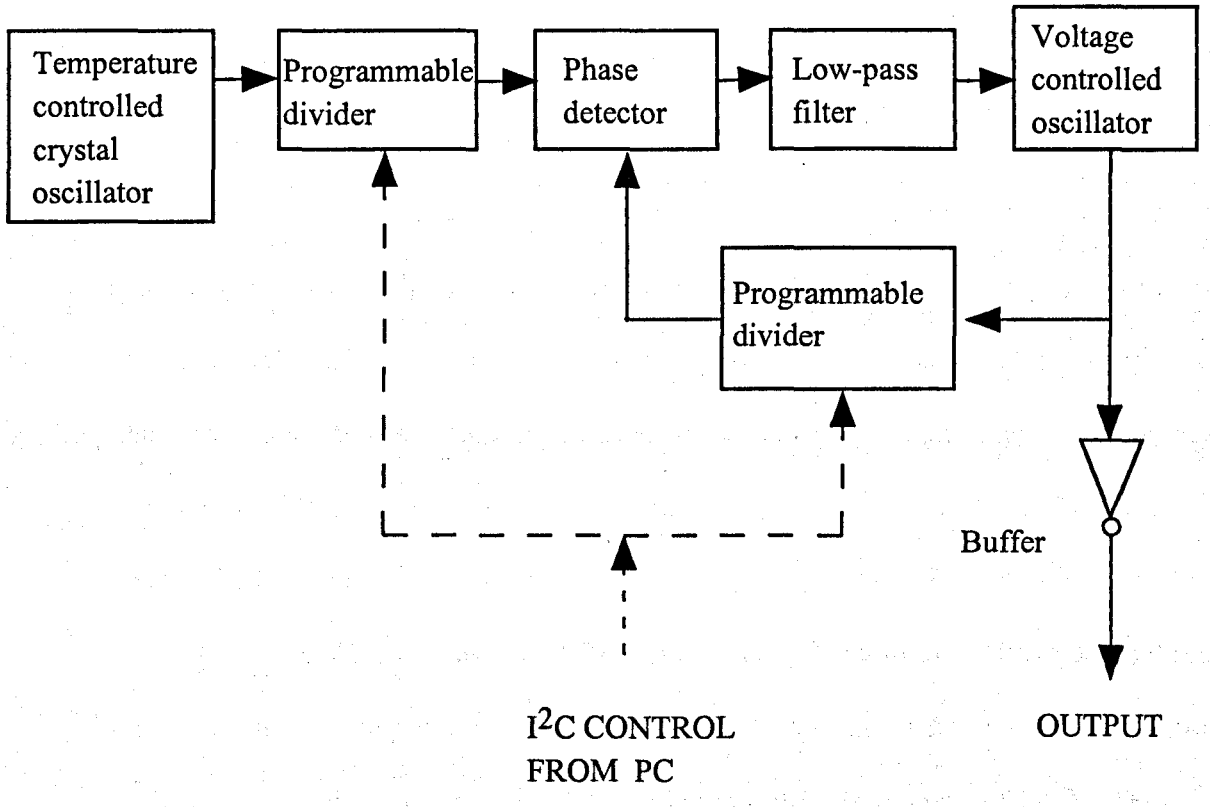


Fig. 6.7 Synthesiser block diagram (simplified)

The circuit is designed to operate with a channel spacing of 5 to 100 KHz and radio frequency input of 50 to 1100 MHz. The circuit diagram reproduced later in this chapter illustrates in block-diagram form the internal working of the chip. Of note is the dead-zone eliminator, which was a problem with earlier PLL type synthesisers, caused when the synthesiser was in a locked state, with effectively no, or constant, output from the comparator to the filter. This level may have drifted a small amount, causing phase noise and some output drift. The Philips integrated circuit eliminates the problem by having switched charge pump rates and a cyclical output when locked.

6.1.4 Synthesiser implementation

The first synthesiser that was designed and built (synthesiser 1) follows standard practice and uses a voltage-controlled oscillator (VCO) to generate the radio frequency (RF) output. The RF carrier is centred on 90 MHz and some output is fed back to the dividers before being phase-compared with the reference oscillator (operating at 10 MHz). The result of this comparison is a series of pulses, a series of pulse 'sinks', or a tri-state high impedance. These are fed to a charge pump, the output of which is filtered and amplified before being fed back to drive the voltage-controlled oscillator.

A VCO has an output frequency proportional to the voltage on its drive inputs, and the charge pump circuitry effectively charges up or discharges a capacitor in a low-pass filter circuit, that is, it controls the output frequency of the VCO. This gives a feedback circuit with a stable output phase locked to the frequency reference. The feedback circuitry was designed with a switch-timing constant of 15 ms and the VCO chosen was a mini circuits POS100 device, chosen for availability and 4.5 MHz/V sweep over the range of frequencies of interest, i.e. initially 45 - 100 MHz.

The second synthesiser, which was used to generate the higher frequencies (up to 1.1 GHz) necessary for the down-conversion stages, used a POS 1025 VCO and was locked to the first synthesiser by using the same TCXO.

A high-stability buffer amplifier was chosen for the loop filter/amplifier. A summary of the calculations for the components is given below. A complete set of derivations and calculations is included in Appendix 7.

Loop bandwidth		=	3 / switching time	.....(41)
		=	200 Hz	
Loop time constant	T1	=	$1 / \omega_n^2 \times T2$	.....(42)
	T2	=	$1 / \cos\varphi - \tan\varphi / \omega_n$	.....(43)

Summarising Appendix 7 :

$$C3 + C1 = K\sqrt{\frac{1+(\omega_v xT1)^2}{1+(\omega_v xT2)^2}} = 8.59x10^{-7}$$

$$C1 = \frac{T2x(C3+C1)}{T1} = 1.47x10^{-7}$$

$$C3 = (C3+C1) - C1 = 7.123x10^{-7}$$

$$R2 = T1/C3 = 2695.49$$

.....(44)

The design was based on an active second-order filter/amplifier and was simulated with *Spice* before being constructed. Some 'tweaking' of the filter values was required due to the loading effect of the VCO.

The next calculation required was for the divide ratios. There are two sets of dividers in the Philips system, the reference dividers (also called the swallow

dividers), which divide the reference input so that the frequency is compatible with the divided main sample frequency. Both are fed in as 4 x 8 bit data streams via the I^2C bus; there is no default with the Philips system. If the divide ratios are not fed in correctly the synthesiser will not lock; therefore, this is required to be the first activity on the I^2C bus. The integrated circuit provides an output indicating an out-of-lock condition which is wired up to a LED, this is fed back to the PC and used to indicate that results may not be valid due to oscillator drift. The initial fill routine is reproduced in Table 6.1.

Data Hex.	Function
00	Null string indicates start
00	of valid data
C4	Write to main synth
18	Null
06	Swallow divide ratio
87	Main divide ratio
00	Fill register C
00	Fill register D

Table 6.1 Synthesiser I^2C programming data

It was found to be necessary to output a number of complete null strings before the UMA1014 would accept programming. This is because when the I^2C bus is not active the chip switches to a power-save mode, as it assumes the batteries in the mobile

phone are going flat, and therefore need to be 'woken up'. In a production unit background scans would be output every 0.25 s, preventing this problem. However, the synthesiser unit consumes an order of magnitude less power than the signal processor. Therefore, as low power consumption is not an essential requirement an easier solution would be to implement the circuit using an earlier integrated circuit without the power controls. Chapter 7 contains the output strings necessary for the synthesisers to lock at the correct frequency for the system to operate; also included are the lock-up times for the two synthesisers.

Quite a few problems were encountered with the synthesiser, many of which were traced to the Mini Circuits VCO not having a linear characteristic over the range of output frequencies being used, causing some instability in the control loop. This showed up as slow lock and jitter on initial set-up and after frequency changes. The solution would be a better quality VCO and in a production unit this would have to be done. However due to cost restraints the POS unit was kept in service with the loop filter adjusted manually until fast lock was achieved. This was fine in the bench units as they were not required to change frequency, just to lock up quickly when commanded to do so via the I^2C bus.

Synthesiser 1 produces an output of 0 dBm into 50 Ω and this is buffered by a wideband amplifier (OMA type) before it is connected to other circuitry. The feedback signal is taken prior to the buffer stage, before additional phase shifts that could be problematic. With the loop filter parameters as previously described the synthesiser produces a nice 'clean' output. A copy of the spectrum analyser trace is reproduced in Figure 6.8. The basic circuit diagram of the synthesiser is reproduced in Figure 6.9. The I^2C circuitry is shown in Figures 6.10 and 6.11. The surface-mount PCB is shown in Figure 6.12.

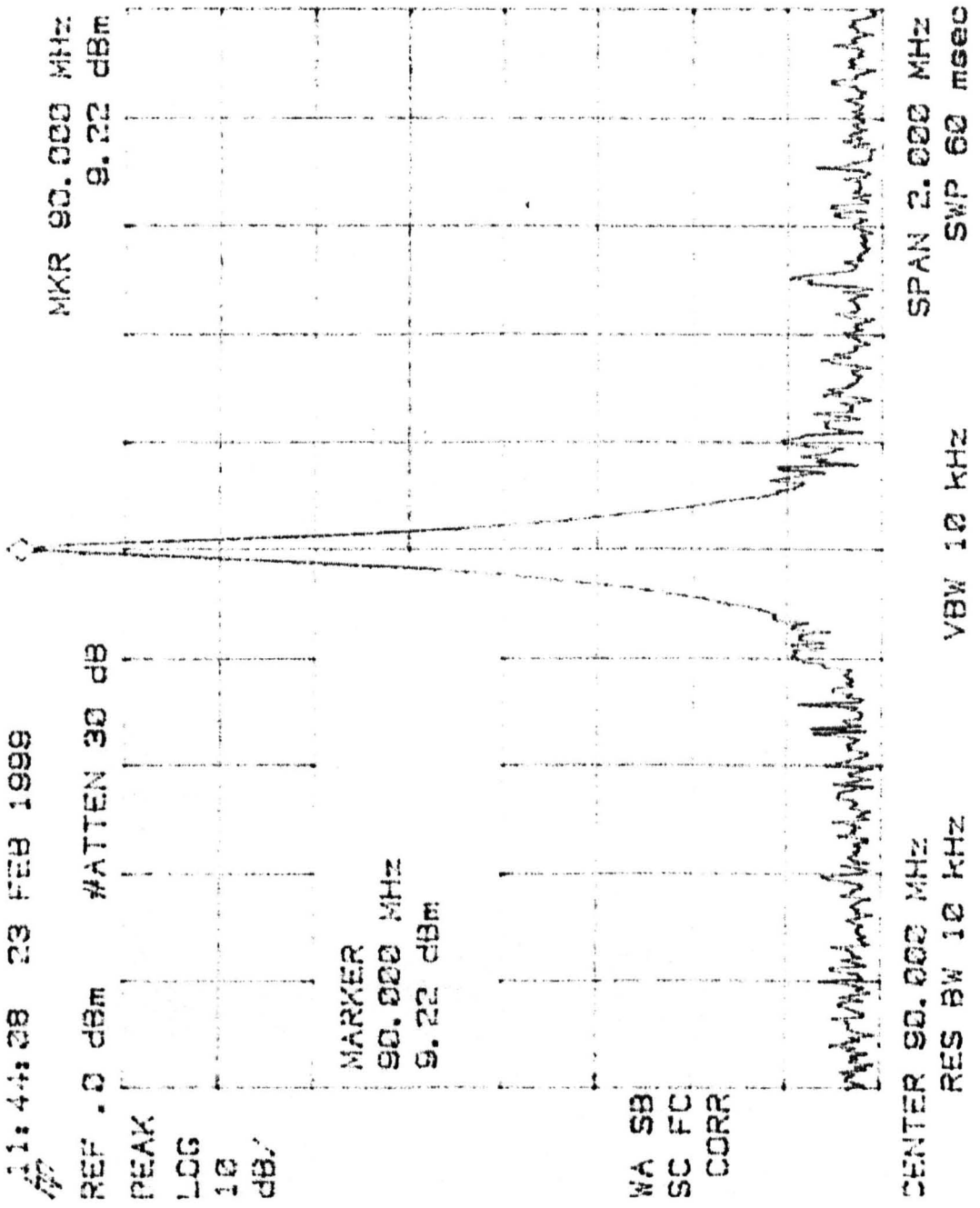


Figure 6.8 Synthesiser output spectrum

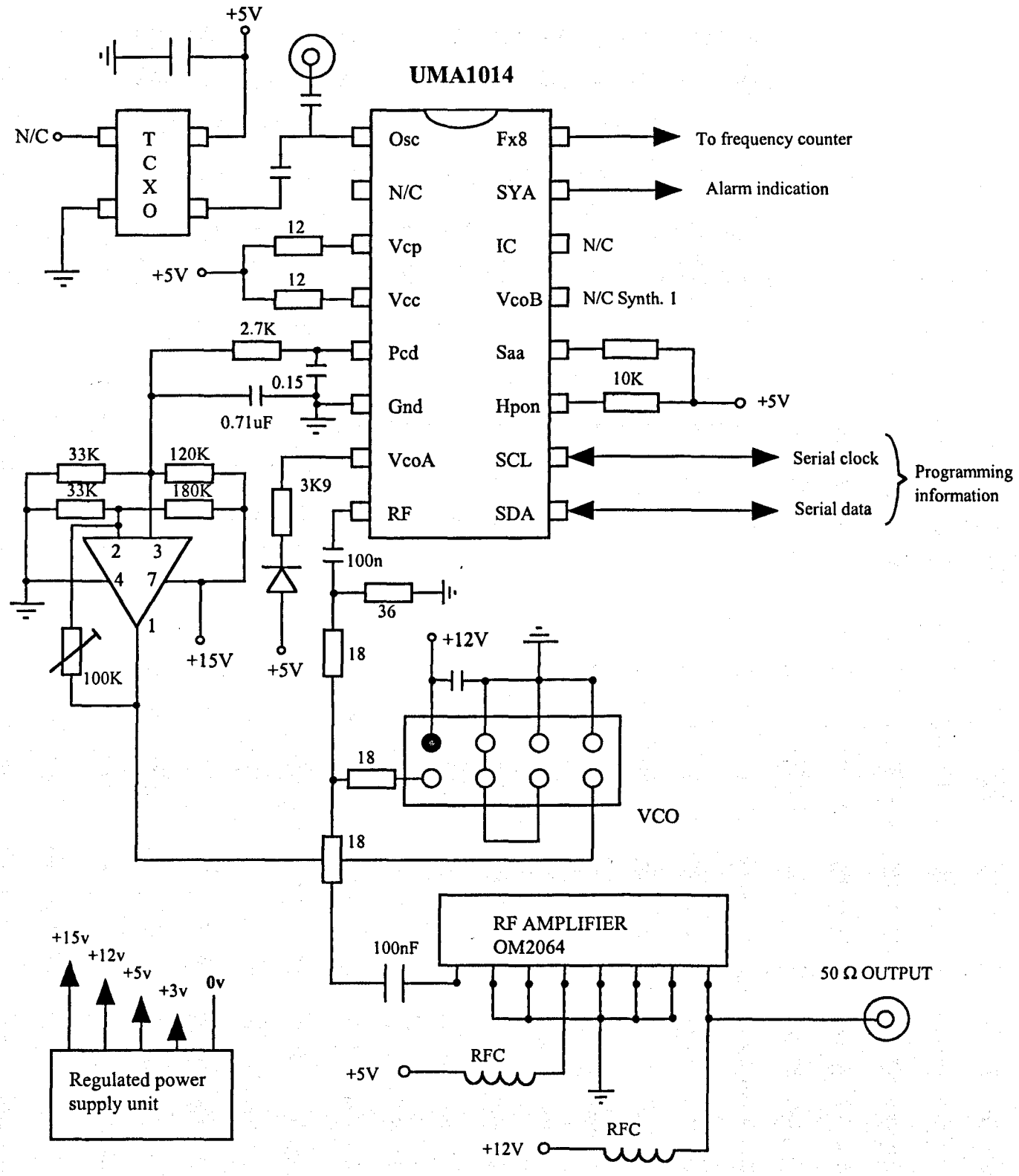


Figure 6.9 CV-CCMA synthesiser circuit diagram (reduced)

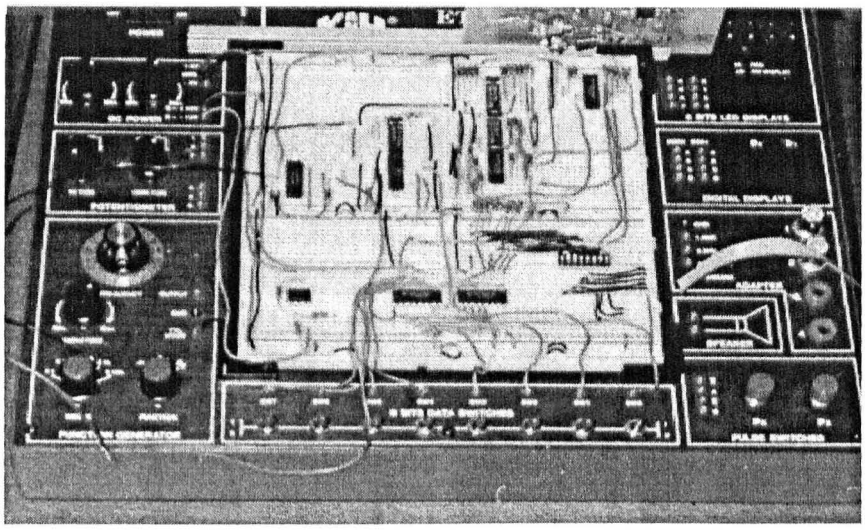


Figure 6.10 Inter–integrated circuit bus loading circuitry

Output bytes are set by the switches on the bottom of the board and then output on the I^2C by pressing the adjacent push button.

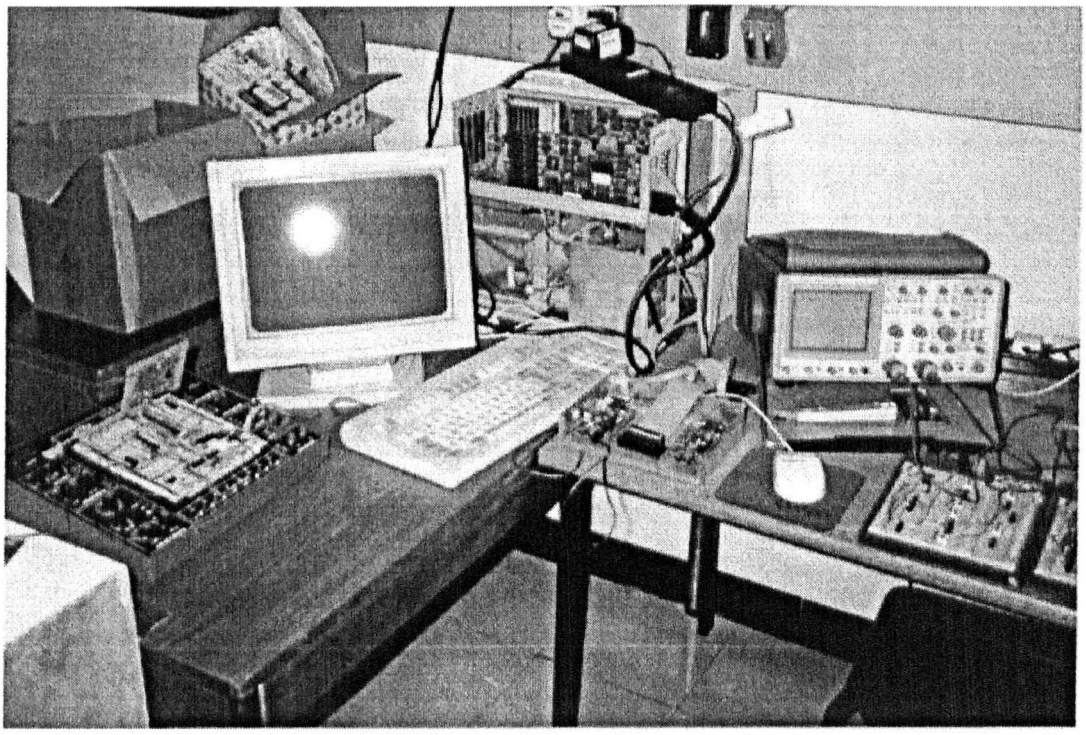
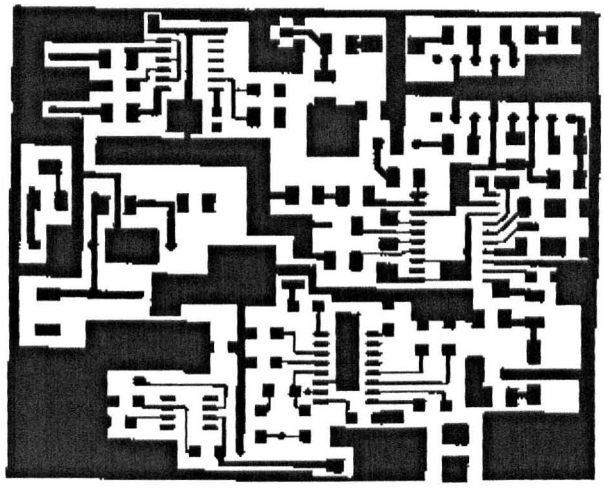


Figure 6.11 PC386 I^2C Bus loading circuitry +TMS320C50 filters

The PC is used to output I^2C data, to control the TMS320C50 DSPs and to program the EPROMs that contain the test patterns.

PCB TOP VIEW



PCB BOTTOM VIEW

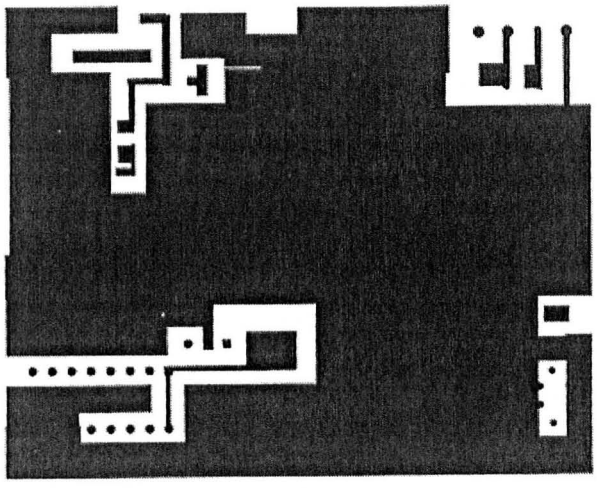


Figure 6.12 Synthesiser surface-mount printed circuit board

6.2 CV-CCMA transmitter implementation

The transmitters were designed and built in three stages. The first stage involved a literature search and a comparison of the various integrated circuits available for speech digitisation and modulation/transmission at the frequencies required by the system. The decision was reduced to a cost basis and because Maxim were offering free devices, and Philips reduced-price devices, these manufacturers became the two main suppliers. The second stage was to design and test the operation of the circuit using one of the simulation packages available, *Spice* (general analogue), *Protel* (digital), *Compact* (radio frequency) and *Z-match* (radio frequency). The third stage involved the construction of a breadboard or printed circuit board and then testing the operation of the circuit. It was decided to go for circuitry that was electrically and functionally robust; hence speech digitisation and recovery were implemented by a system of Delta modulation/de-modulation. A functional block diagram of the first transmitter is reproduced in Figure 6.13.

6.2.1 Input stages

An electrical representation of the audio speech signal was input to the transmitter from a standard microphone via a band-limiting filter. This restricted the frequencies of the incoming audio in preparation for the next stage and ensures the Nyquist criteria are met. The filter is a standard pulse code modulation (PCM) telecommunications filter with a 300–3400 Hz bandwidth.

The amplitude variations present in the speech are now controlled by using a VOGAD (voice-operated gain adjustable device). The VOGAD is a high-gain differential amplifier circuit that gives a fairly constant analogue output in response to

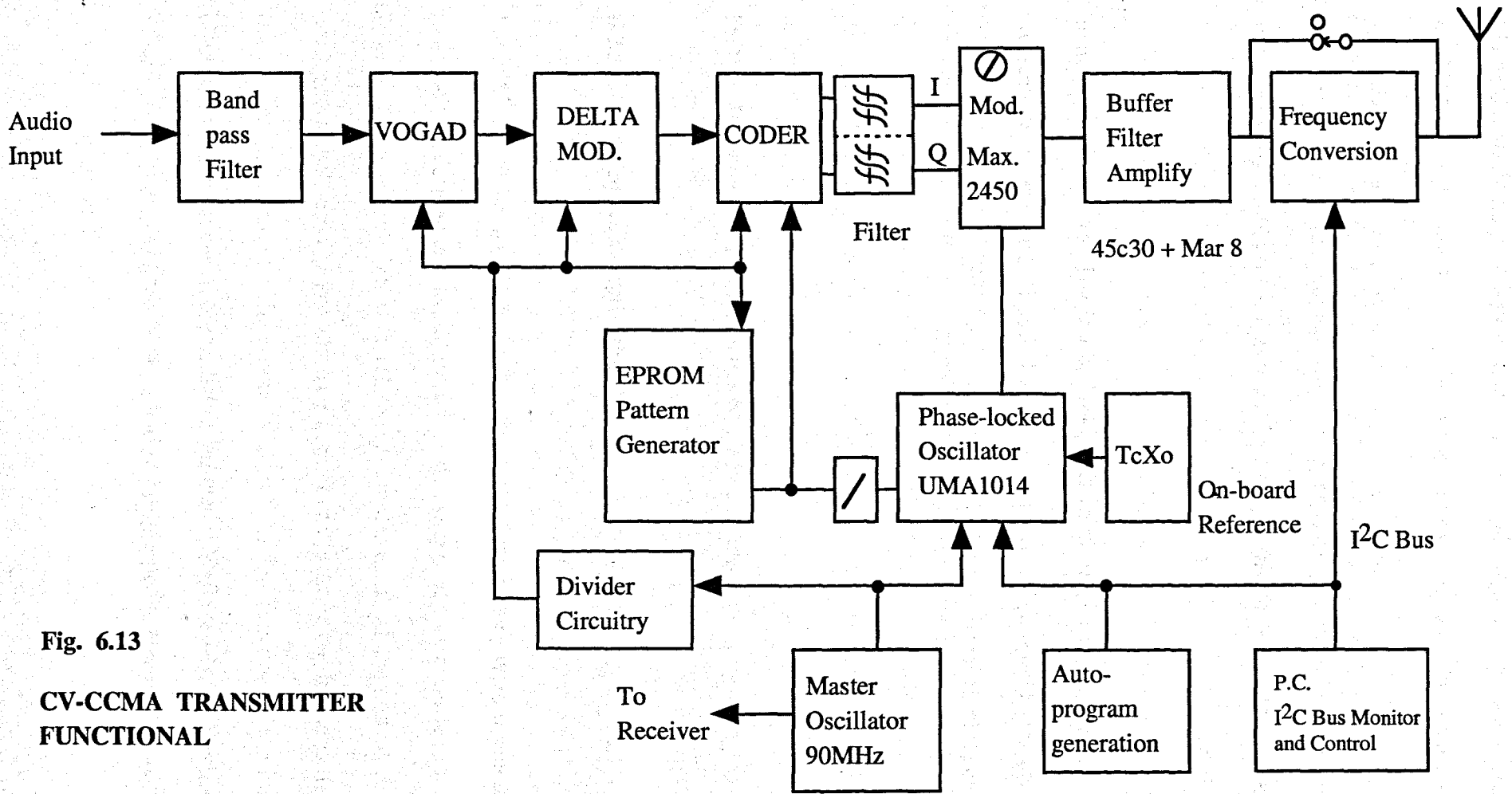


Fig. 6.13

CV-CCMA TRANSMITTER FUNCTIONAL

a large dynamic range of inputs, that is the typical range of inputs may have a 50 dB amplitude range – too large for the modulator to digitise. The attack time of the VOGAD is set to a small value (<1 ms) to ensure there is a limited delay introduced into the processed audio signal. Any delay, especially a variable delay, would cause problems with the rest of the system. The output of the VOGAD is now fed to the digitisation stages. Figure 6.14 illustrates the audio stages of the first demonstrator transmitter. The transmitters were constructed with a buffer stage in parallel with the analogue input stages. The buffer is a FIFO (first-in first-out) type and enables the connection of external digital TTL level signals. This would be useful for the connection of external test equipment used for generating particular line codes and for the testing of different signalling systems. There is also a pattern-generator circuit built into the transmitter breadboards. This provides for a test pattern, which is used to ensure that the transmitters and receivers are synchronised; it does not connect to the buffers. The simplified circuit is reproduced in Figure 6.15.

6.2.2 Speech digitisation

The speech digitisation process was designed around a standard Delta modulator. The lower bit rate schemes (code excited linear predictive, etc.) were deemed to require too much processing and would introduce delays which would cause problems in the system. Delta modulation is a relatively simple, efficient method of digitising an analogue signal, usually speech. The name derives from the mathematical delta or derivative, indicating that the output is a 'differential' function of the input, that is changes in the input signal are coded rather than specific levels. To increase dynamic range, modern systems adopt techniques of varying the digitisation depending on the type of signal present.

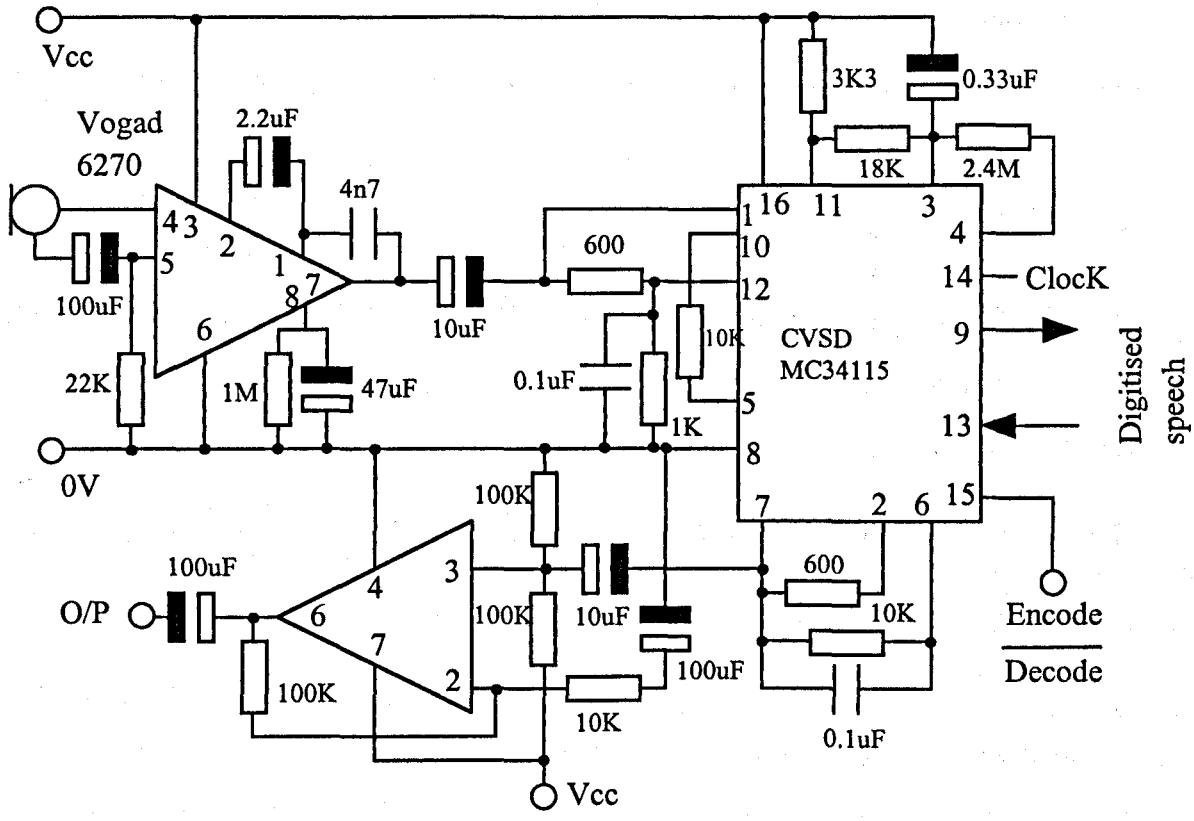


Figure 6.14 CV-CCMA Demonstrator transmit audio stages

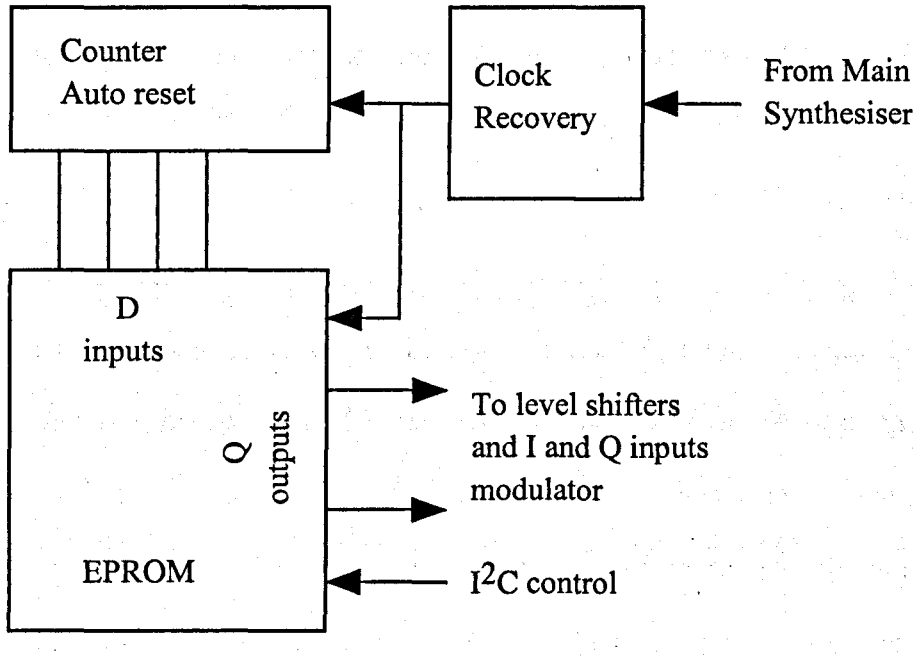


Figure 6.15 CV-CCMA Demonstrator pattern generator circuit

In the CV-CCMA it was decided to opt for a CVSD (continuously variable slope delta-modulation) system. This gave the best compromise between cost and speech quality. Relatively good speech quality can be obtained at an output data rate of 16 Kb/s and intelligible speech can be obtained at data rates as low as 4 Kb/s. A Delta modulator is a sampled data system employing a negative feedback loop. A comparator senses whether or not the instantaneous level of the analogue voice is greater or less than the feedback signal. If the voice input is greater than the feedback signal the output is a one, if it is less the output is a zero. The comparator output is clocked by a flip-flop to produce a continuous non-return to zero (NRZ) data stream. The feedback signal is set up so the integrator ramps up and down to produce an approximation of the speech waveform, which, when filtered at the demodulator, reproduces the original speech. Figure 6.16 illustrates the basic Delta modulator.

The standard Delta modulator cannot track large amplitude high-frequency signals with its fixed integrator; neither will it resolve small amplitude changes which are less than the height of the integrator ramp during one clock period. The CV-CCMA requires better quality than this so the CVSD technique was implemented. The simple Delta modulator has a second integrator built in with additional logic and an analogue multiplier to become a CVSD. The circuit of the CVSD is reproduced in Figure 6.17. Under small input signals the second integrator, known as the syllabic filter, has no input, and circuit function is identical to the basic delta modulator, except the multiplier is biased to output quite small ramp amplitudes giving good resolution for the smaller signals. A larger signal input is characterised by consecutive strings of '1's or '0's in the data as the integrator attempts to track the input. The logic input to the syllabic filter actuates whenever three or more consecutive '0's or '1's are present in the data. When this happens, the syllabic filter

output starts to build up, increasing the multiplier gain and passing larger amplitude ramps to the comparator, enabling the system to track the larger signal. Up to a limit, the more consecutive '1's or '0's generated, the larger the ramp amplitude. Since the larger signals increase the negative feedback of the modulator and the forward gain of the demodulator, companding takes place. A syllabic filter of between four and about ten milliseconds was found to give the optimum reproduction of speech (even with an industrial English Midland accent, which is accentuated by the distinctive 'nasal' sound of Delta modulation).

Standard telecommunication type digitisation (e.g. ADPCM) will be interfaced at a later date, when time and funds allow. In a production device the low-bit rate processing could be implemented as a sub-routine performed by the signal processor, with the necessary rate adaptation transmitted as unique codes. Good quality speech can now be communicated at very low digital bit rates, but the systems involve prediction techniques with the addition of variable delay; this would cause problems with synchronisation of the CV-CCMA system and so Delta modulation was used in both the prototype units. The first breadboard versions of the Delta modulator used a discrete implementation, which created problems due to the filters drifting. A Harris military integrated circuit was then used. This worked very well and had security coding options built in which could be implemented in the future. The manufacturers of the integrated circuit have been bought out and have now stopped production. The modulation/demodulation circuitry could be implemented in a semi-custom device, together with the rest of the transmitter coding logic; this upgrade is presently underway.

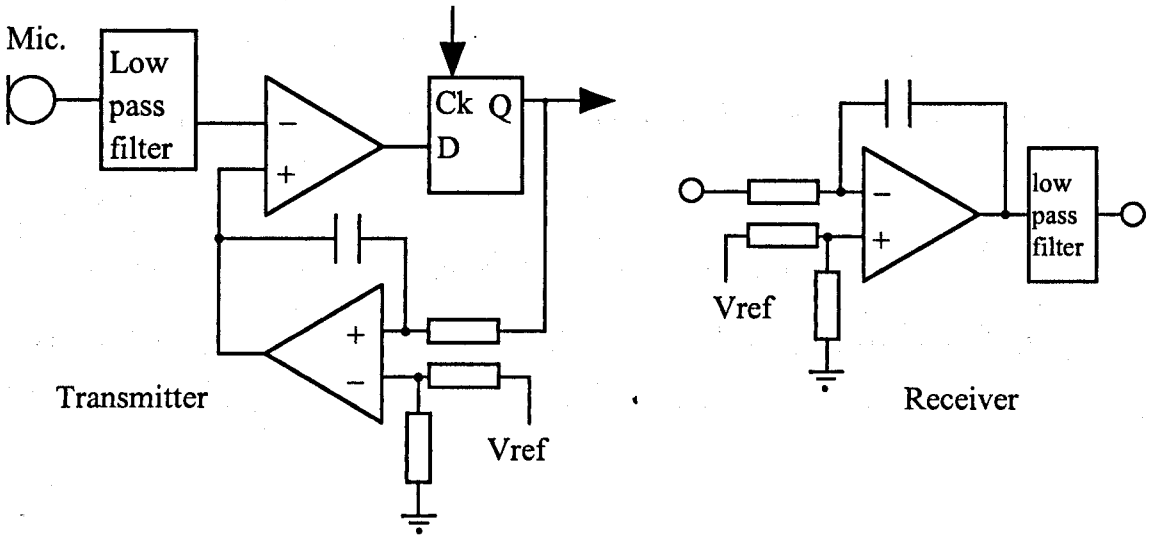


Figure 6.16 Basic Delta modulator / demodulator

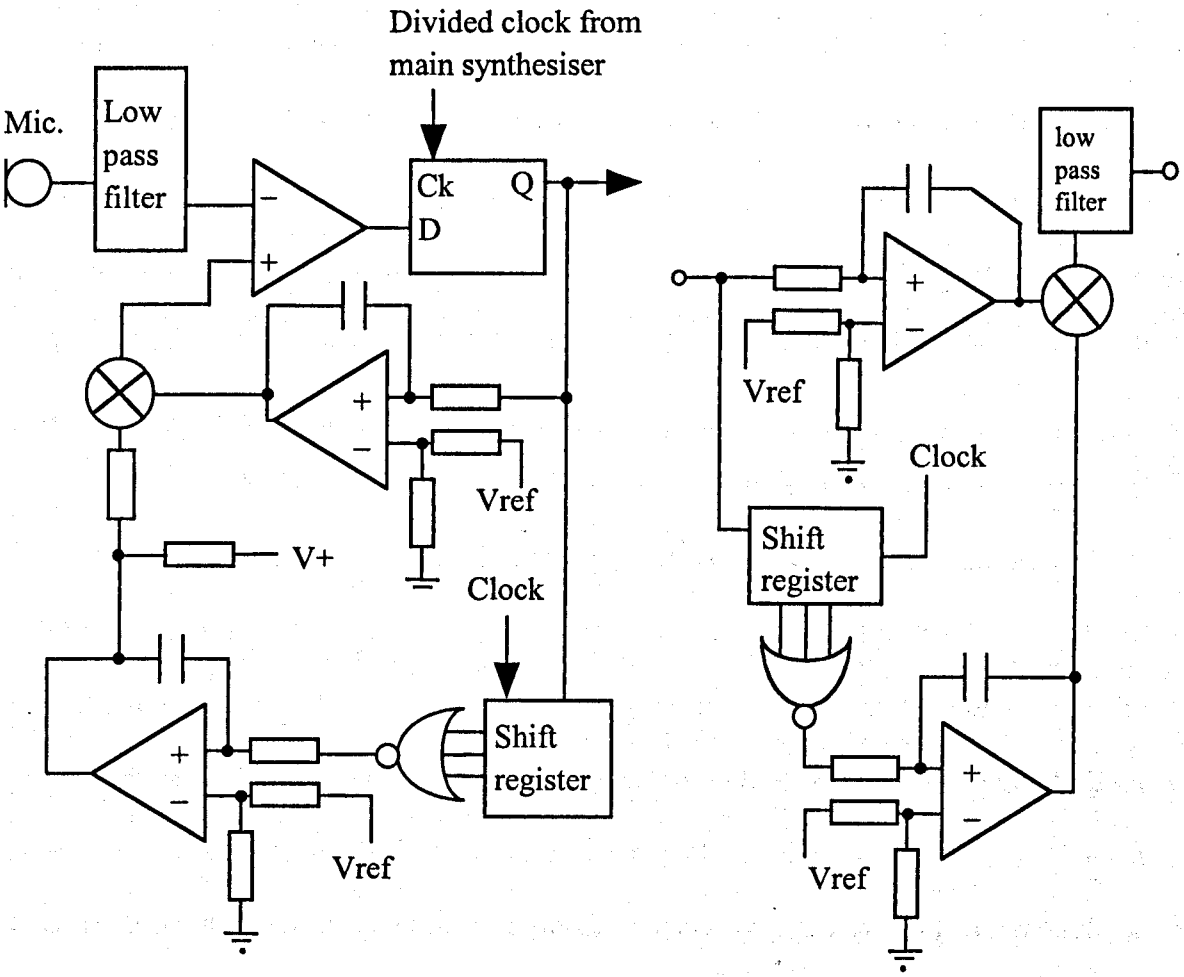


Figure 6.17 Continuously variable slope Delta modulator / demodulator

6.2.3 Implementation

The second part of the design involved actually developing the circuit with simulation using *Spice* to check the viability of the designs. Appreciable effort was put into working out how the systems could be synchronised together and a number of unique circuit schemes were developed. The *Spice* exercise (see Appendix 4) indicated that the audio amplifiers would give sufficient dynamic range for the modulators to work correctly.

The VOGAD output is fed to the input of the Delta modulator, which samples at a clock frequency derived from the central clock. This is implemented with standard divider integrated circuits which are fed from the feedback signal at the synthesiser. The Delta modulator is a CVSD (continuously variable slope delta-modulator) type; this gives an adequate speech quality at low bit rates. This is achieved by adjusting the slope of the feedback integrator when strings of continuous '1's or '0's are being fed back from the output. This gives a closer approximation for speech with rapid amplitude changes and is reproduced with better clarity after re-integration.

The main problem with Delta modulators is due to the integrators built into the circuit. If a very low signal, that is no audio or only quiet background noise, is fed into the modulator the integrator will ramp up (akin to charging a capacitor) and then discharge. This causes a large popping at the receiver audio output and the receiver can lock up when in this state. To get around this problem in the CV-CCMA there is a level detector built in; if an input signal falls below a certain level for a certain time, a 'quieting' pattern is output. This causes the receiver's Delta modulator to mute its audio output and prevent the 'popping'. The output mute function is also connected to the 'out of lock' indication on the synthesiser, preventing noise on start-

up and when out of lock. The quieting pattern is very similar to the pattern transmitted at start-up to synchronise the transmitters and receivers, this is actually an advantage as the speaker is muted when the system is initially trying to synchronise itself. Refer to the circuit diagram of the audio stages, which were built onto a separate board, reproduced in Figure 6.14.

The Delta modulator output is fed into a buffer and then into the codeword generator. The codeword generator consists of standard combinational logic circuitry built onto a large breadboard with the codewords selected by dual in-line (DIL) switches. The QPSK implementation has eight parallel-to-serial converters, the outputs of which are logically OR'd together and then fed to the buffer/level shifter and then to the In phase and Quaternary inputs of the modulator. The parallel-to-serial converters are arranged in pairs and each pair is selected by its associated two bits of input, i.e. 00, 01, 10 or 11. The required output sequence is programmed into the buffer by selecting the correct DIL switch and pushing the load button. The programmed output sequence will continue to be output as long as power is on and the converter pair is selected. To change the output the board needs to be powered down and the switches reset; this will lose synchronisation and the circuitry needs improving to stop this problem. The bit and symbol timing pulses are generated by interconnected flip-flops that are slaved to a Harris timing generator integrated circuit, which is itself slaved to the divide circuits fed from the main synthesiser. A simplified codeword-generation circuit is reproduced in Figure 6.18 and the full circuit is reproduced in Appendix 8. The codeword generator also outputs a 'word-valid' pulse which is used to frame each codeword. The timing for the word-valid generator is also derived from the dividers that feed the modulator.

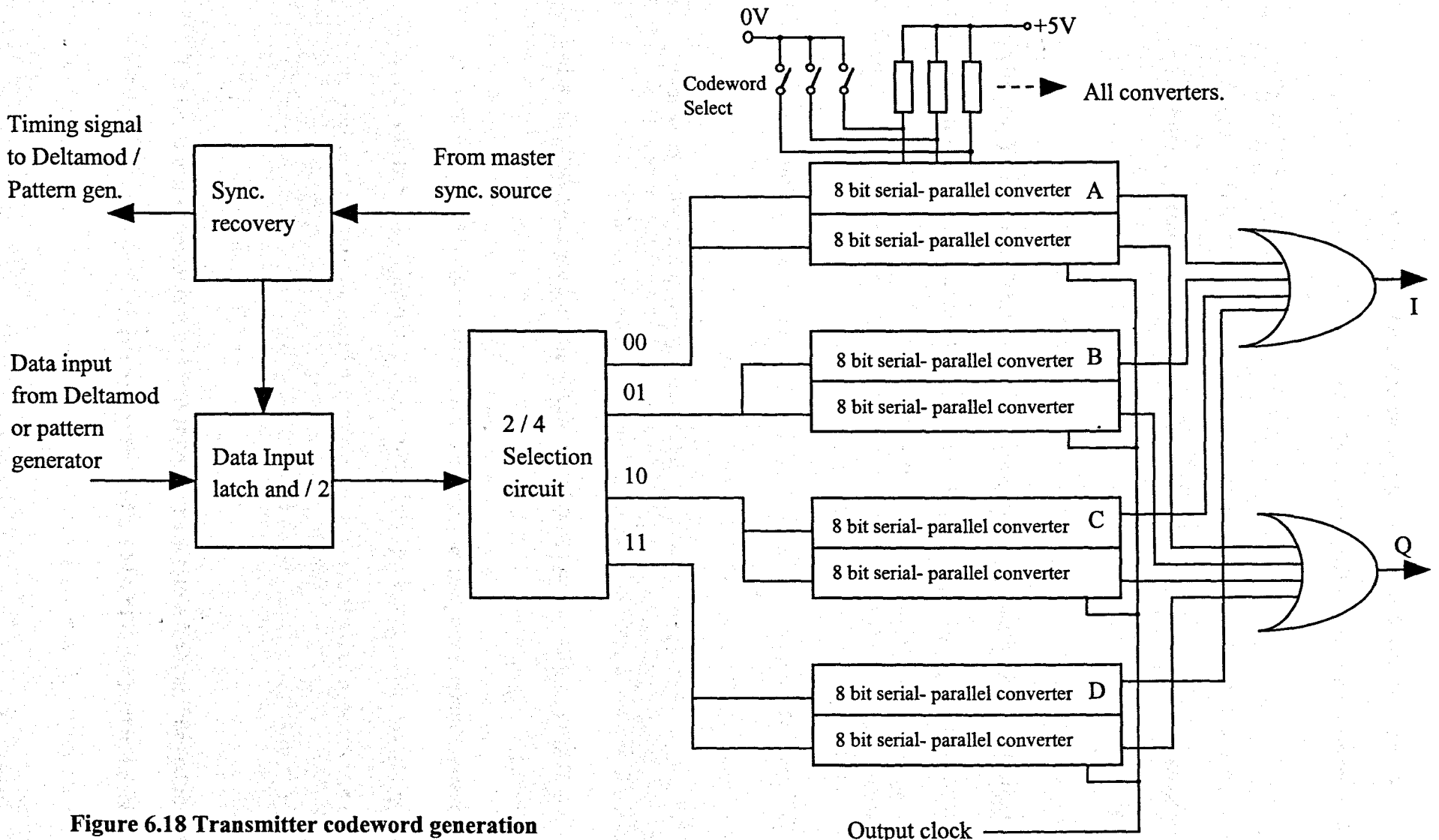


Figure 6.18 Transmitter codeword generation

N.B. For complete circuit diagram refer to Appendix 8

Alongside the circuitry used to digitise and code the speech waveform, on a separate breadboard, is an erasable programmable read-only memory (EPROM) code generator which is used to generate the initial synchronisation sequence of codewords. This can also be used to generate test patterns, by operating a simple push-button selector switch on the board, and used to check the operation of an already-synchronised system. The EPROM circuitry consists of a Harris timing generator integrated circuit, slaved to the main synthesiser via the divider integrated circuits. The output of the Harris integrated circuit is squared up with a flip-flop and fed into a binary counter set for auto reset. The parallel outputs of the counter are fed to the least significant address lines of the EPROM; thus the various memory locations of the EPROM are cycled through in sequence. The outputs of the EPROM are level-shifted and fed to the I and Q inputs of the modulator. The EPROM is programmed, using proprietary software and interface, via the large computer visible in the laboratory photographs, Figure 6.11.

The codeword-generation circuitry is an ideal candidate for inclusion in a standard logic gate array, which would make an interesting project and would reduce the size of the circuitry and improve the reliability of the system. The large breadboards, with many connections between the integrated circuits, are the main cause of the CV-CCMA's current reliability problems. The pattern-generation circuitry could also be included in a gate array, also simplifying implementation.

6.2.4 Transmit filters

The generated codewords are output from the buffer, to a level shifter and then to the I (in-phase) and Q (quaternary phase) inputs of the MAXIM modulator. Level shifting is required because the 5 V combinational logic code generator is required to interface

with the 3 V modulator, which can only tolerate a maximum voltage of 1.35 V on its inputs. The pulse-shaping filters (matched) would be connected at this point; however, in the initial test set-up they were not required and were hard-wire bypassed. All of the filters were connected in the later implementations. One reason for this increase in complexity was to try and reduce the criticality of the synchronisation of the system. The results obtained are described later in Chapter 7. The filters would need to be implemented in a commercial radio, in order to reduce the bandwidth of the modulated carrier. If the pulses were fed in directly the rapid changes between states would increase substantially the bandwidth requirement of the system; there is a good derivation of this in [117]. An approximation to the root raised cosine shape gave satisfactory results. Note that the first type 1 synchronisation transmit spectrum plots in Chapter 7 are taken without the transmit filters being connected in circuit; this gives the spectra the 'wideband' plot shown.

Some time was spent implementing different filters using a pair of Texas TMS320C50 digital signal processor development boards. This is a small project in its own right and in order to reduce the construction time it was decided to use standard code to implement the filters, obtained from the Texas Instruments ftp website, a summary is copied in Appendix 9. It was found that the Texas code produced the most reliable results. Various other sites were investigated, and a number of the code-generation programs written in the standard DSP textbooks were also tried, but produced some very cumbersome code that did not work very well.

A lot of useful code was developed using the Texas development boards in order to implement different filters. Figure 6.19 illustrates the filter implementation using two Texas TMS320C50 fixed-point digital signal processor development boards as implemented on the test set-up. Figure 6.20 illustrates the

typical pulse-shaping curves used in modern digital telecommunications and summarises their properties. It was originally planned to use differential QPSK modulation together with square root raised cosine pulse shaping at the transmitter' with a matching square root raised cosine filter at the input of the receiver; thus when combined together the overall response is of a raised cosine shape. The phase change values could be Gray-coded (or similar) to ensure the minimum change of phase between states, thus reducing the bandwidth of the transmitted waveform.

The baseband filters have linear phase and a nominal square root raised cosine frequency response of the form:

$$|H(f)| = \begin{cases} 1 \\ 0 \\ \sqrt{0.5(1-\sin(\pi(2f-1)/2a))} \end{cases}$$

$$f \le (1-\alpha)/2T$$

$$1-\alpha/2T \le f \le 1+\alpha/2T$$

$$f \ge 1+\alpha/2T$$

.....(45)

where T is the symbol period and α is the roll off factor, 0.35 in Tx1 and 0.5 in Tx2.

A 20 tap FIR (finite impulse response) filter was implemented using the Texas TMS320C50 DSK development board. Tap coefficients are stored in the program memory, the same coefficients being used for the in-phase and the quadrature channels. The transmit filter was implemented using one of the circular buffers of the TMS320C50 with an interpolation factor of four, reducing the number of multiplications required by a factor of four. The five I delays are immediately followed by the five Q delays in the internal dual-access RAM.

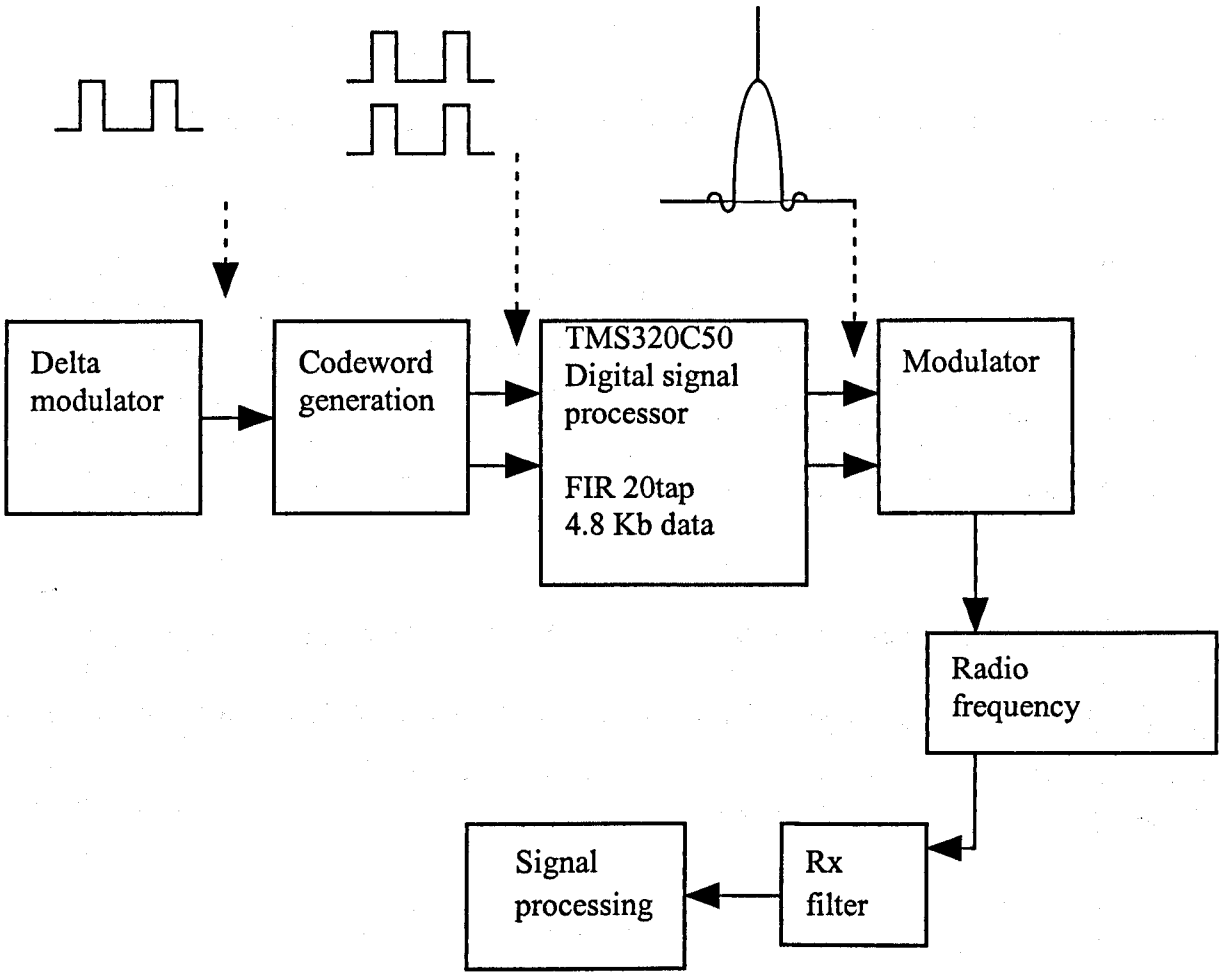


Figure 6.19 Pulse shape filtering

N.B. Timing / frequency conversion and RF filtering not shown for clarity

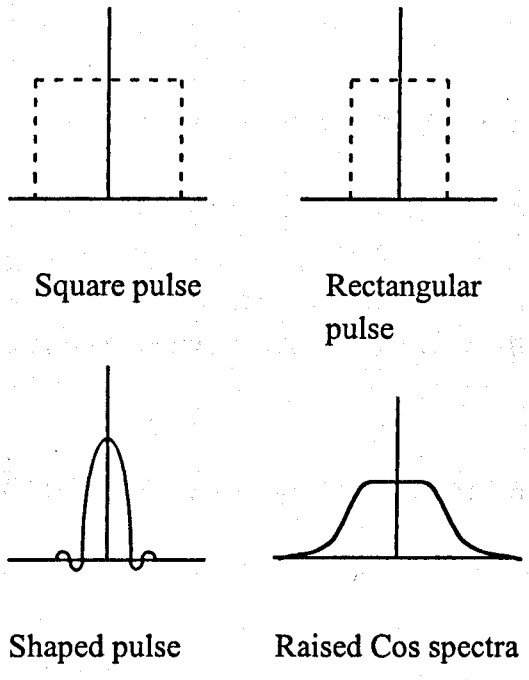


Figure 6.20 Typical pulse shapes / spectra

The MADS and MADD (multiply and accumulate with dynamic address) instructions are used to simplify the process. In the receiver the matching baseband filter is implemented using similar code but without the interpolation. Code is reproduced in Appendix 9. The filters worked well but the additional delays introduced caused problems. The system now comprised six digital signal processors, which were all critical to the operation of the system and so needed to be carefully implemented. To reduce the complexity a simple bypass was implemented to get the system working. This comprised a hardwire switch to lock out the operation of the filters. Further work is required on the filters; due to time constraints this had to be assigned a lower priority and will be worked on when time allows.

6.2.5 Radio frequency modulation

The conditioned data baseband signal, at 7.2 Kb/s in the first test system, is now fed into the radio frequency stages of the transmitter. Initially, this was the MAXIM standard QPSK modulator. Within the MAXIM modulator, circuitry is designed to minimise spurious feedback; the internal oscillator is set to twice the IF (intermediate frequency) or output frequency. The modulator gives sideband rejection of 38 dB, which is very good and adequate for the CV-CCMA system. A fairly large input signal (+2 dBm at 90 MHz) was required to override the internal free-running oscillator and lock the system to the synthesiser. This may cause problems in a production unit and another modulator requiring less drive, and hence less leakage, less mixing products and therefore blocked channels, would be preferable. However, as the MAXIM units were free they were screened properly and made to work. The MAXIM integrated circuit contains a prescale /divider and this only gives an output when the input levels are within specification to run the circuit. This output is taken as

an indication of the correct operation of the modulator, and is fed back to the frequency counter as a check. The modulated signal is fed out as a differential pair, at a low level. A summary of the modulator specification is included in Appendix 10. The rest of the transmit circuitry is conventional in design. The output of the modulator is filtered by a standard 45 MHz centre frequency, 300 KHz bandwidth crystal-type filter in order to remove any unwanted product frequencies. It is then amplified by a Philips NE5219 and a Mini-Circuits MAR-8, giving an output power of +10 dBm into a 50 Ω load. This is sufficient power to free-transmit approximately half a mile, with external aerials. Circuits of the modulator and amplifier are shown in Figure 6.21. The frequency up-converters are disconnected in the test set-up. Complete program listings are included in the Appendices.

6.2.6 Discussion

One of the main problems with the CV-CCMA transmitters was a tendency to 'take-off', producing unwanted radio frequency outputs and causing the amplifiers to overheat and shut down. This seemed prevalent when operating with a common code set, and was caused by certain mixer/modulator product frequencies being passed to the amplifiers and hence causing some instability. Tweaking the filters reduced the effects.

Initially, to check the operation of the system, it was decided to operate with known g values between the transmitters and receiver. A number of possible techniques were studied and the best solution was thought to be the transmission of a training sequence at the start of transmission. In the breadboard implementation the transmitters and receivers are locked together, effectively hardwired, with attenuators connected between the transmitters and the receivers. On initial start - up these

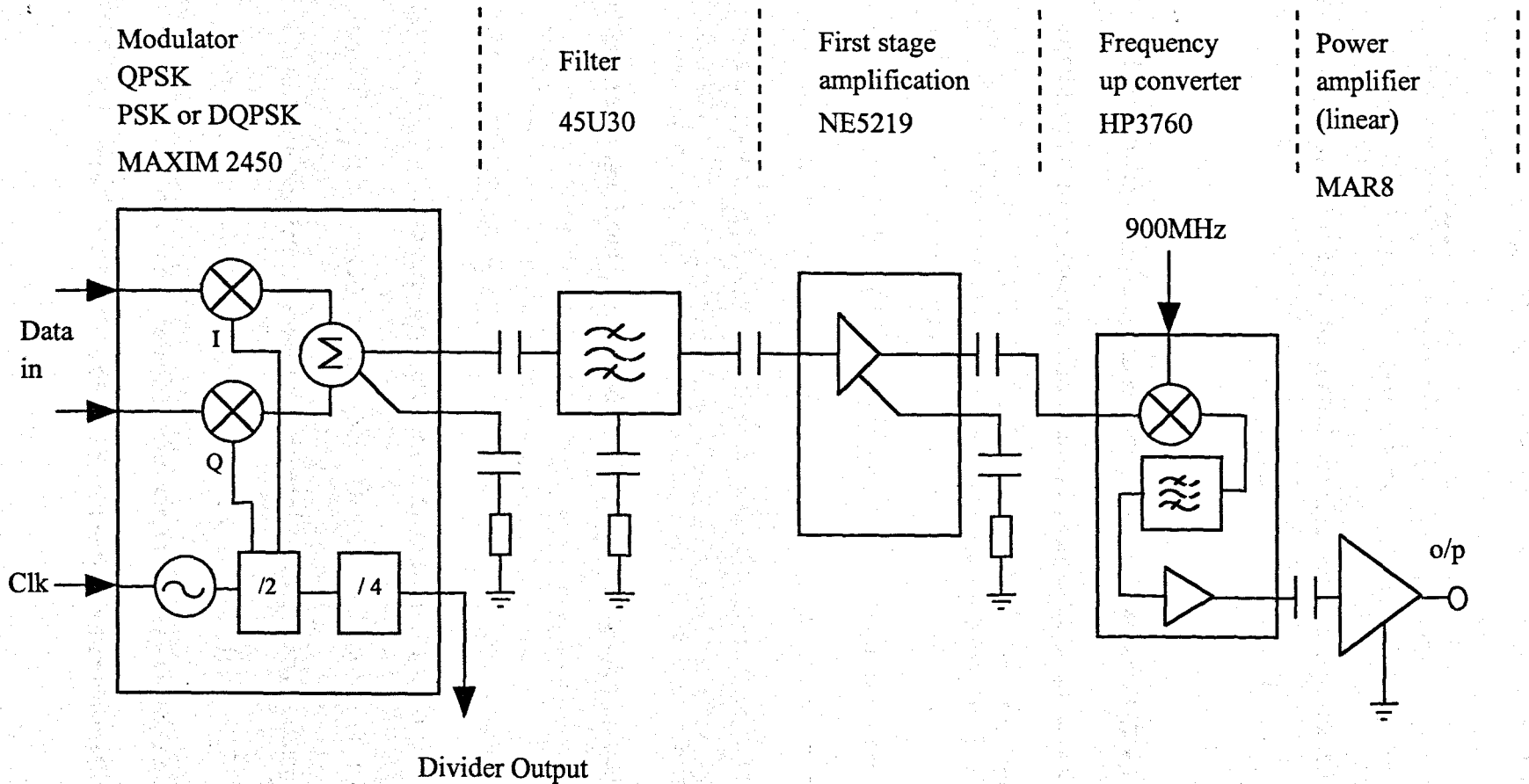


Figure 6.21 Transmitter radio frequency stages

attenuators are set to zero attenuation and zero phase shift, and then one at zero and the other at 0.5+j0. The system requires a test pattern to be transmitted before information transfer can take place. This is achieved by programming a test pattern into an EPROM . This memory circuit is also controlled by the I^2C bus so it can be switched in and out by the controlling computer as required. On receipt of this test transmission the receiver locks to the test pattern and, when valid decoded data appears at the receiver, the transmitter can then switch to encoded speech or data. This is done manually in the bench set-ups but it has been found that a very short sequence is all that is required to synchronise the two systems. Therefore a simple timer would do for synchronisation at power-up. In a production model the situation is somewhat different and in a mobile radio situation the g value would be constantly varying. This requires a fast lock up (if a signal is present) or, if no signal is present, slows down the effective scan until a signal is captured. To increase dynamic range the g value could be fed from an AGC circuit (automatic gain control) in the receiver, derived from the signal processor and some additional hardware. This section of the CV-CCMA still requires some development, and possibly some research into other methods (CDMA adopts a two-way power adjusting method to prevent similar problems, but increases the circuit complexity greatly).

The circuitry designed to be used in the CV-CCMA system is only at an 'X' (experimental) model state. Development work is required to sort out outstanding problems; some of these were known before but others have come to light during the research. See Chapter 8 for more details. Figure 6.22 shows the codeword generator boards and contains the circuitry for clock recovery and system synchronisation, used to time the generated output patterns with the rest of the system.

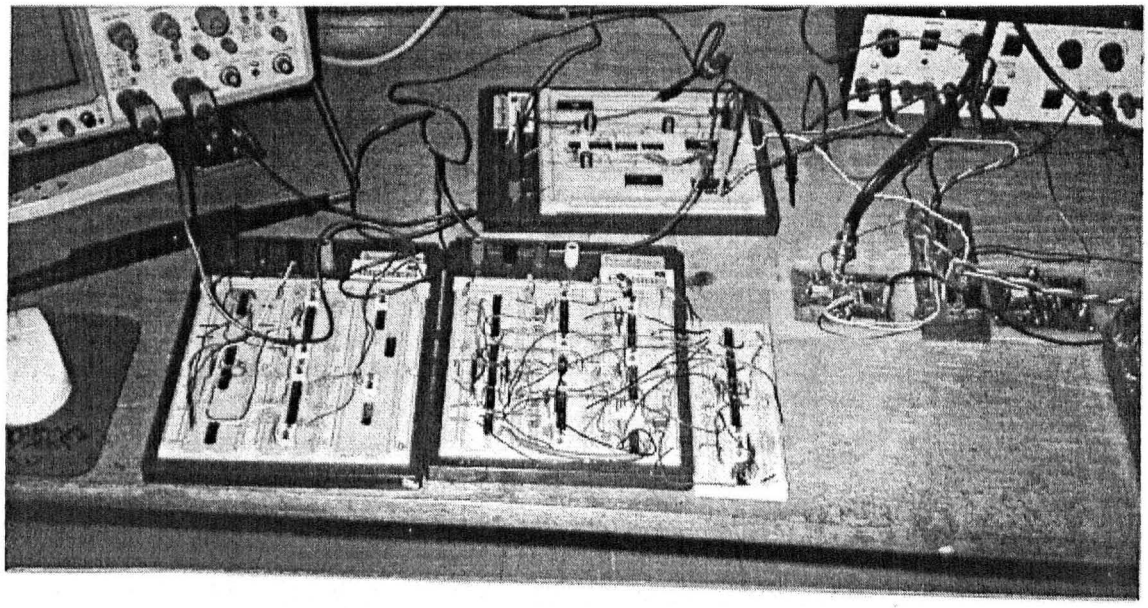


**Figure 6.22 Codeword-generator boards (foreground)
Test pattern generator boards (background)**

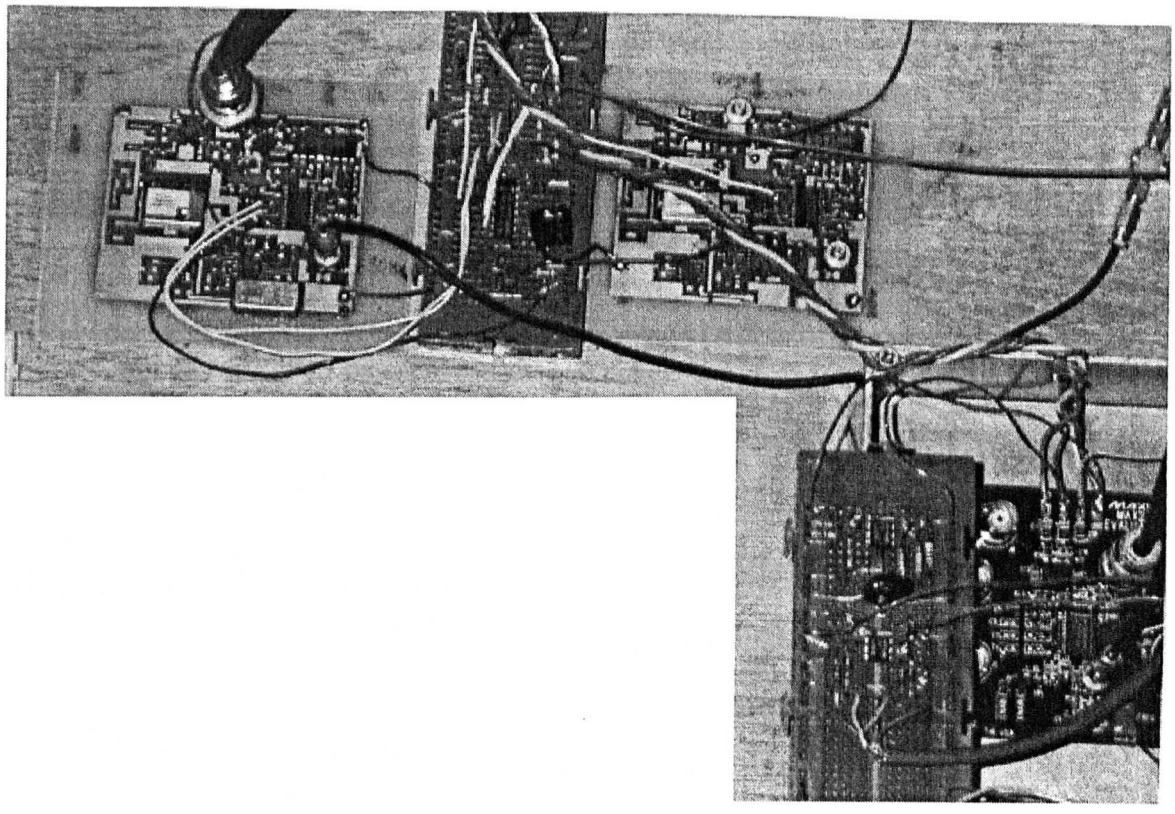


Figure 6.23 Two transmitters and data load circuitry

The board on the left of the photograph also contains one Delta modulator and demodulator, together with an output buffer. Codeword selection is by the DIL switches visible on the breadboards. There are eight sets of switches, one for each I and Q input for each of the four phase states of the QPSK modulation format. The test-pattern generator board also contains clock-recovery circuitry and the two EPROMs used to generate the patterns are clearly visible.

The photograph illustrates that the associated circuitry necessary for the system to run, in the prototype system is quite complicated, so the amount of general wiring on the breadboards is fairly substantial. As previously mentioned, this part of the circuit is ideally suited to implementation in a semi-custom integrated circuit and this is planned for the future.

Figure 6.23 is a photograph of the two transmitters set up in the laboratory at Warwick, where some of the interconnecting wiring is removed for clarity. The blue breadboard between the transmitters contains buffer circuitry and clock dividers and the blue breadboard in the foreground is the receiver I/Q buffer. The Maxim development board is visible in the diecast metal box. The box was found to be necessary in order to keep interference levels to a minimum.

6.3 Receiver implementation

The design of the receiver followed conventional digital radio design procedures. In order to generate the various frequencies that were required the same synthesiser was used in the receiver as in the transmitter. Also, as with the transmitter, the output frequency was controlled from a Toshiba computer, via the I^2C bus. This necessitated the implementation of additional combinational circuitry, as the address of the divider/synthesiser integrated circuit is only one bit address selectable. The oscillator select pin on the integrated circuit was used, giving the option of two synthesisers on the same bus. However, there are a minimum of three synthesisers required, and different address lines were used to control the different synthesisers. Hence the additional wiring was implemented from the start.

The receiver follows the superheterodyne arrangement, which is a circuit technique developed in the 1930s to improve the performance of radio receivers [118], especially the early radar receivers. In a superheterodyne receiver the incoming signal is mixed with a locally generated signal called a local oscillator (LO).The output of the mixer contains sum and difference frequencies as well as the original frequencies. If the local oscillator is made variable the receiver can tune to different input frequencies whilst the same frequency (called the intermediate frequency - IF) is output from the mixer. This gives the advantage of being able to amplify, detect and decode at lower frequencies with fixed filters that can have much better characteristics than variable filters, at much lower cost. The end result is a receiver that is sensitive and selective. The receiver was designed from the outset to be a single conversion type, where only one intermediate frequency is employed for the bulk of the amplification. An IF of 45 MHz was chosen because of the restricted range of frequencies available with the

Maxim integrated circuits used. In a production unit the performance of the receiver would be improved by changing to a multi-conversion type superheterodyne. This would produce an improved image rejection and self-blocking performance. However, this requires board and circuit redesign and so was not attempted on the initial test set-up. The AGC circuitry built into the first-stage amplifiers is disabled but this would need to be implemented on a production unit. On the test model the RF amplifier was simply switched in or out of circuit depending upon whether it was required or not. Refer to [118] for a good analysis of the trade-offs involved in receiver design.

6.3.1 Front-end circuitry

In the CV-CCMA system the incoming signal is first amplified by an amplifier that is designed for optimum noise performance, rather than maximum gain. Any noise generated at this stage will be amplified by the later stages of the receiver and degrade the performance of the whole receiver by a proportionally large amount. This first stage is called the low-noise block (LNB), and was designed in stripline using computer program *Compact* to check the design. The amplifier is a single-stage low- noise HEMT (high electron mobility transistor) MOSFET (metal oxide semiconductor field effect transistor) device. The circuit was constructed using standard-quality circuit board and worked satisfactorily when tested, giving a power gain of approximately 15 dB at 850 MHz with a noise factor of 1.5 dB. The noise factor is the ratio of the signal to noise at the input to the signal to noise at the output, and thus gives a good indication of the noise that has been added by the stage. A copy of the circuit, the simulation results and the actual results are included in Appendix 12 and Figure 6.24 illustrates the stripline layout of the amplifier. A production unit would require the low-noise block to be constructed on a more professional substrate. This is needed for better consistency of

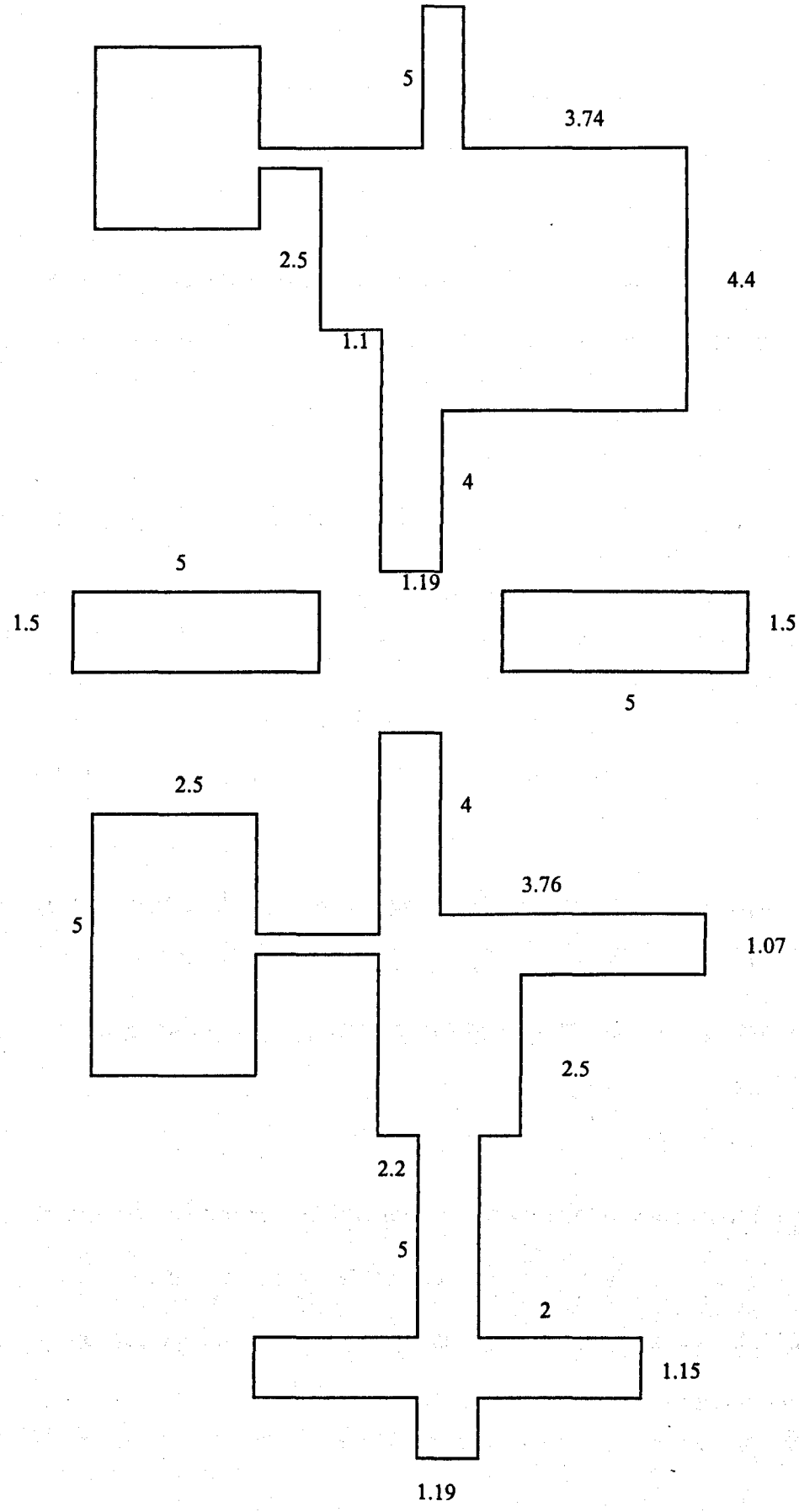


Figure 6.24 Low-noise block stripline layout (not to scale all dimensions in mm)

permittivity and improved dimensional stability; an alumina/beryllia based material would probably be best suited. This was not implemented on the test set-up due to cost restraints. The low-noise block amplifier was constructed on a separate board and housed within the diecast metal box containing the rest of the receiver components; this was done to reduce interference and noise problems. A separate low-noise block amplifier would not be required on a production unit because the more up-to-date mobile phone integrated circuit sets now incorporate most of the radio frequency stages within the main integrated circuit.

6.3.2 Second stage and mixer

The front-end second stage incorporated another radio frequency amplifier and used a Philips GSM integrated circuit (NE600) as its first-stage amplifier and frequency down converter. The low-noise amplifier selected had a 2 dB noise figure at 900 MHz with 16 dB of gain. Although these figures are not state of the art the integrated circuit has excellent stability with automatic compensation for temperature drift. These parameters are more critical for the correct operation of the CV-CCMA system, rather than absolute values of gain and intercept point. The integrated circuit also has automatic overload capability: the on-board amplifier is bypassed and the input is fed straight to the mixer, which switches to mode 2, giving a +26 dBm third intercept. All the ports are designed to operate into 50 Ω, which makes interfacing easier. The mixing frequencies were generated by another Philips synthesiser integrated circuit (UMA1014) which was programmed up via the I^2C bus with data fill generated by a computer. (Refer back to Section 6.1 for a more detailed description of the design of the synthesiser). The generated intermediate frequency signal (45 MHz) was then fed via a filter and Philips standard digital IF strip into the demodulator section of a MAXIM 2450 integrated

circuit. Crystal filters were used, the 45U30 having a 3 dB bandwidth of 30 KHz. This worked fine with minimum shift type modulation schemes, and a broader bandwidth unit (300 KHz) was used in the QPSK implementation. Better quality filters would be needed in a production unit. The digital intermediate frequency amplifier used was a NE5209, which has gain to 1.5 GHz with an 850 MHz bandwidth. The gain is controlled via a single input between 0 and 1 V, giving over 60 dB of control range. The automatic gain control implemented in the IF stages of the receiver is very basic in operation. The automatic gain control voltage, required for a production unit, could be generated by a sub-routine of the signal processor. This ensures the input values to the analogue-to-digital converter are always within specification and the $\boldsymbol{g}$ values can be simply calculated. The circuitry is fairly standard so development of existing circuits rather than new circuit design is all that is required. In the breadboards that were constructed for the demonstrator CV-CCMA the front-end stages are built onto the surface-mount circuit boards. However, in the first test set-up the circuitry is bypassed, so the receiver input is connected directly to the 45 MHz generated by the transmitters. The IF is fed directly into the demodulator which contains a single-ended to differential converter. Gilbert cell multipliers are used to mix the IF signal with the quadrature local oscillator signal (90 MHz input, then divide by two in order to prevent spurious feedback). Baseband in-phase and quadrature signals are then output and amplified by the baseband amplifiers (21 dB). The I and Q outputs are further amplified and fed to the input of the analogue-to-digital converter. A Burr-Brown precision device was chosen, again mainly on cost considerations rather than performance criteria. The later implementations of the test set-up incorporated additional filtering. This comprised two Texas Instruments fixed-point DSPs connected in the I and Q lines, after the Maxim integrated circuit. They are connected to act as root-raised cosine filters, matching the

transmit filter. The design was based on a MODEM (modulator demodulator) design published by Texas Instruments [119]. The system uses 4 interrupts in a 20 tap FIR (finite impulse response) filter. To increase the sampling rate 3 zero pads are inserted in between each of the 5 channel data stream symbols. This means that 15 of the multiplications in each convolution are by zero, so that only 5 coefficients are required to be multiplied. Consult [119] for a detailed explanation of operation and assembler code used. Functional block diagrams of the receiver and a reduced circuit diagram are shown in Figures 6.25 and 6.26.

6.3.3 Analogue-to-digital conversion

The CV-CCMA system requires a fast and accurate conversion of the recovered baseband signal to a digital format. Some effort was put into investigating the many different types of analogue-to-digital (A-D) converters currently available in order to choose the best device. The decision was taken to go for a Burr-Brown [120] DSP102 device, again mainly on cost parameters. The DSP102 is a two-channel A-D and is configured to sample its two inputs and concatenate the two 16-bit outputs as a 32-bit serial word, which is read into a buffer on the serial port of the signal processor. This is a major advantage because with the low-cost Texas Instruments devices only one serial port is built into the DSP. This means a single port can be directly interfaced with a dual A-D converter without resorting to store and shift buffering.

The DSP102 consists of two independent 18-bit sampling analogue-to-digital converters, which can sample up to 200 KHz. The sample clock is derived from the master synthesiser, but the signal processor generates the output clock. The buffer is only 32 bits so it is essential that the processor can 'read' the samples faster than the receiver can generate them. The DSP102 is a fast successive approximation device with

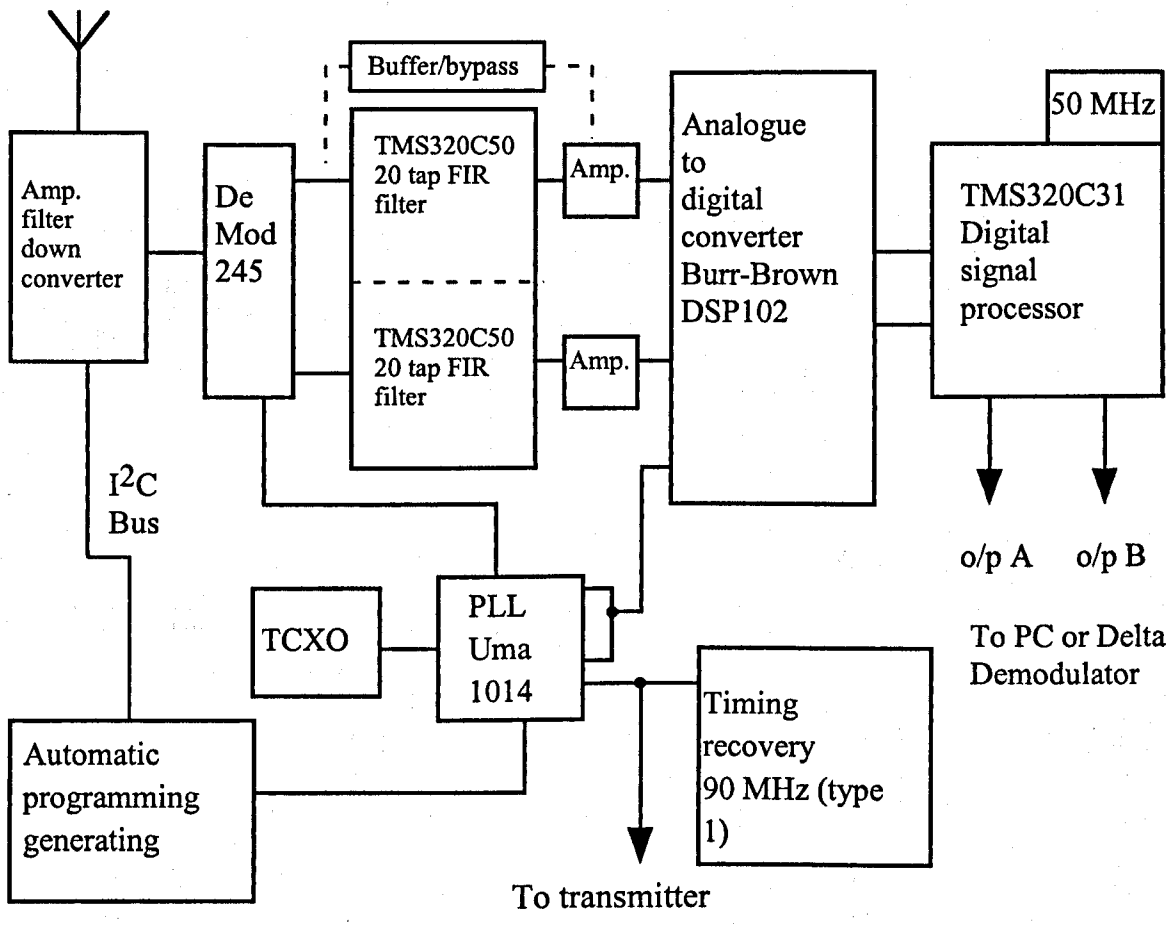


Figure 6.25 CV-CCMA Receiver functional

Notes:

- Front end is standard GSM mobile phone chip set
- Auto-Program is standard PC with interface to drive the I^2C bus
- De-modulator is Maxim device designed for mobile low power
- Buffer-amps are a matched pair op-amps
- Burr-Brown precision A-D has concatenated outputs (2 x 16 bit) and interfaces with the Texas signal processor via the serial port.
- Data is clocked out of the A-D a sample in delay at a rate determined by the signal processor.There is limited buffering so it is important to keep synchronisation once obtained.

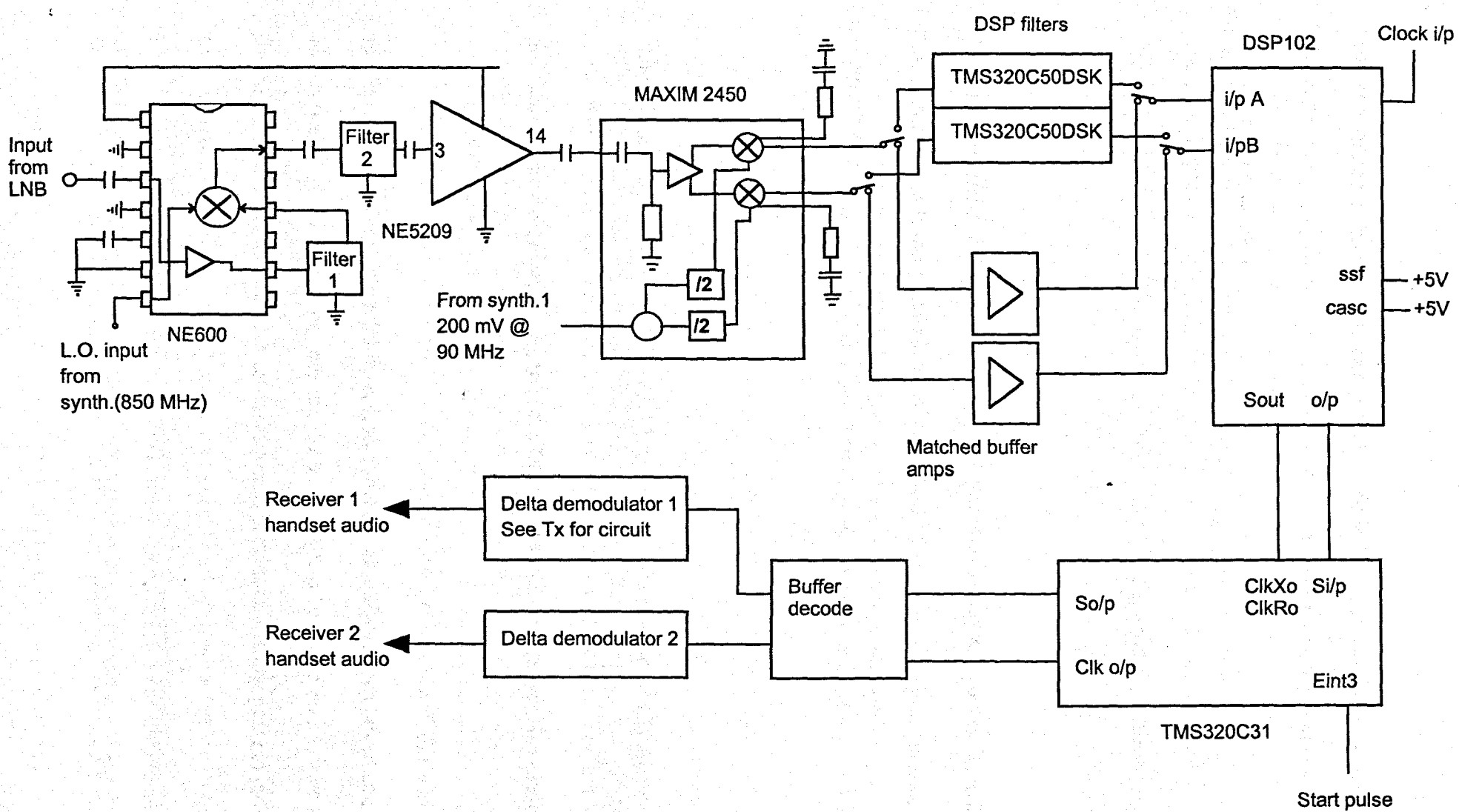


Figure 6.26 Receiver circuit (reduced)

all the required reference voltages generated internally. The sampling is done via a switched capacitor array inside the integrated circuit. It was found necessary to buffer the signals on the two inputs because, although the specified input impedance was quoted as approximately 1 KΩ, in practice it was found to be less than half this. This caused problems with loading of the IF chain. The buffers were also set up to level-shift the signals, as the full input range of the A-D is –2.75 V to +2.75 V. In the BPSK receiver the range is restricted to the positive section only; the QPSK system uses the full range of input voltages in order to obtain the better resolution required. It was important to match the performance of the two buffers closely, and it was decided to select the Burr-Brown devices as the two amplifiers are built onto the same silicon and hence have identical temperature and drift characteristics.

Figure 6.27 illustrates the range of inputs, voltages and the digital outputs generated by the analogue-to-digital converter. Note that the minimum step size shown is not applicable to the CV-CCMA as the two least significant bits are treated as noise, with no valid data on them. The A-D chosen has an 18-bit resolution but 16 bits are chosen for ease of interfacing with the DSP. Figure 6.28 is a simplified diagram of the analogue-to-digital board. The analogue-to-digital converter, buffers and associated circuitry were constructed on a separate multi-layer printed circuit board; this was to try to keep noise problems to a minimum. It was found that the lower two bits of converted output were just noise, so concatenating the output to two 16-bit streams lost no useful information. The initial codeword set-up sequence provides a framing pulse to provide a start reference point for the signal processor to enable differentiation between symbols. In the initial implementation this start pulse is hard-wired via some combinational logic between the transmitter and the receiver. In the later bench set-up the start pulse is derived from the training sequence, but only when using fully independent and unique

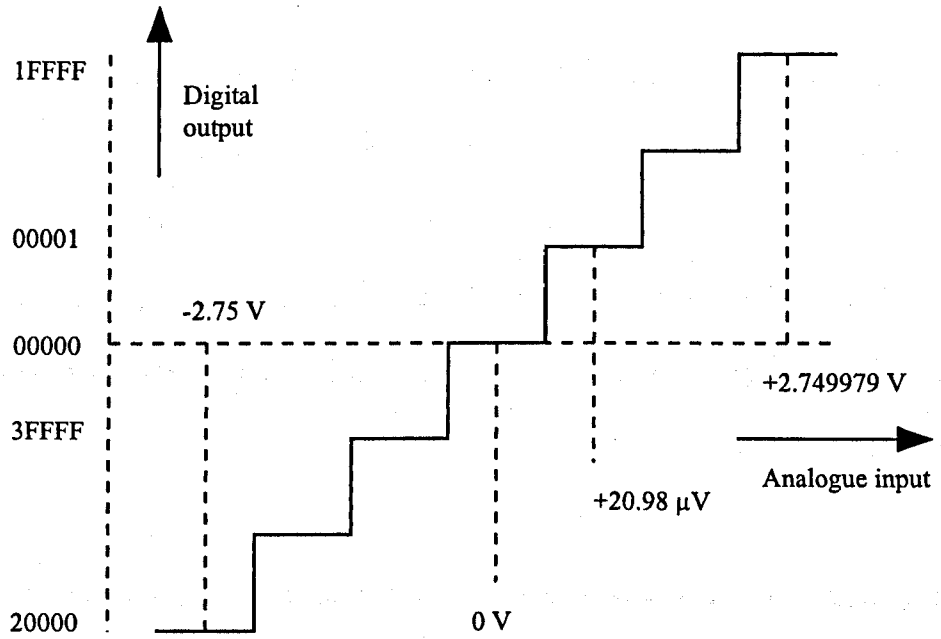


Figure 6.27 Burr-Brown DSP102 analogue input / digital output

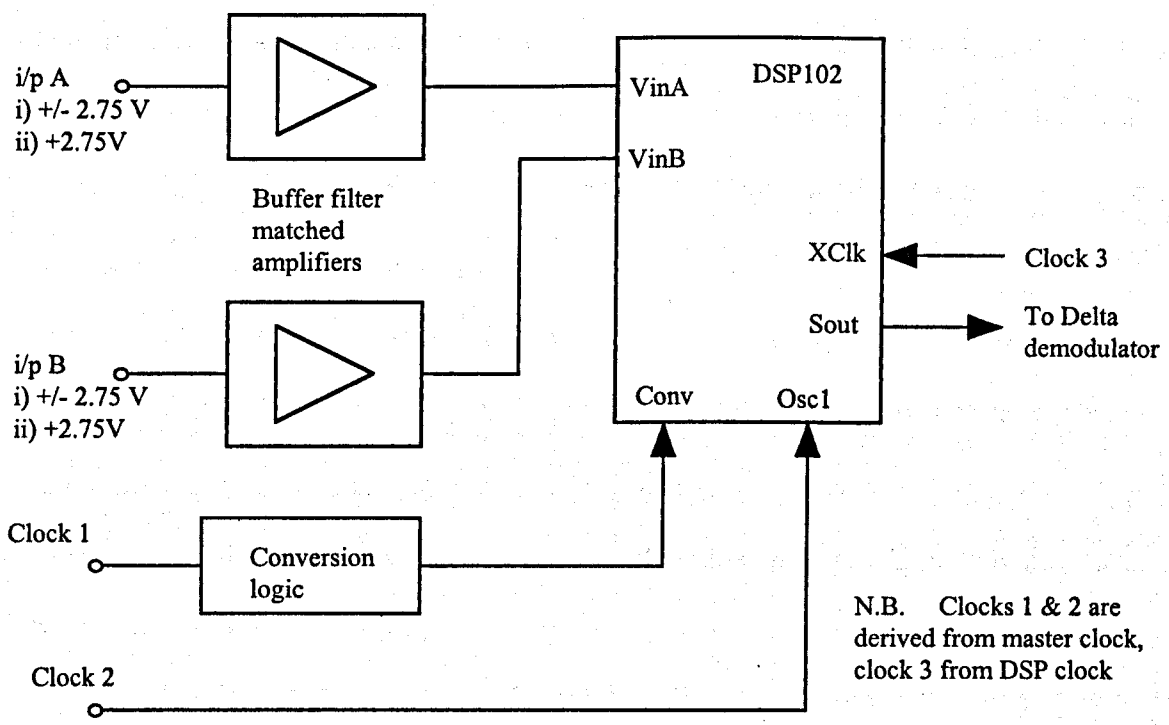


Figure 6.28 Analogue-digital daughter board

code sets at the transmitters. The data is now temporarily stored in the output buffer of the analogue-to-digital converter ready to be output to the digital signal processor to be decoded. Note that the output of the analogue-to-digital converter is controlled by the digital signal processor and the input of the analogue-to-digital converter is controlled, that is timed, by a derived system clock. It is therefore essential that the digital signal processor can operate on the data being fed to it faster than the analogue-to-digital converter can produce it. A feedback technique was trialled, which slowed down the sampling rate of the analogue-to-digital converter when the digital signal processor was overloaded. This circuit created more problems than it cured and was dropped. The timing problems were fixed by careful programming in assembler language, the code generated by the 'C' compiler proving to be very cumbersome and too slow.

6.3.4 Signal processor interface

The interface with the signal processor was designed and built a number of times, and was eventually reduced in size to a single circuit board. The TMS320C30 digital signal processor has two serial input ports and on the breadboard models the I and Q inputs were fed directly into serial port 1 and serial port 2 respectively. The later implementations used the low-cost TMS320C31 which has only one serial port interface and necessitated the two outputs from the analogue-to-digital converter being concatenated. This was no problem with the Burr-Brown device but required major additional circuitry to be designed for use with the other analogue-digital devices tested (H-P, National and Analogue devices).

An external interrupt line is used to provide the 'start' information, indicating the first burst of valid data will immediately follow. This is generated by combinational logic in the hard-wired type 1 system, and by receipt of the initial

synchronisation sequence in system types 2 and 3. Then the data is clocked into the signal processor from the analogue-to-digital converter buffer under the control of the signal processor using a clock derived from the DSP 50 MHz clock. This is the only clock in the system that is not synchronised to the master clock. The framing information is derived by the signal processor from the first valid clock pulse, so it is vital that the first pulse is correctly interpreted; if errors run continuously the system will reset itself and attempt to synchronise again. The later versions of the system attempted to use Viterbi-type coding and identifiers in the data streams and hence the frame timing was not required. This area of research is still ongoing and results will not be presented in this thesis.

With no audio input to the transmitter the Delta modulator outputs a quieting pattern; this is in order to switch the audio off at the receiver and prevent 'popping' as the integrators ramp to rail. This caused a few problems as the repetitive pattern was sometimes interpreted as start synchronisation so the next few bytes of data were lost, producing annoying gaps in the speech. The problem did not arise if the system (type 1) was transmitting data as there are no gaps in the traffic. This is another development requiring further work before the system could be viable as a production unit. There are also recently announced analogue-to-digital converter integrated circuits aimed at the digital audio market from Texas Instruments that could be used with the signal processor and enable the 'glue' logic to be reduced.

The various set-up routines for the registers, clocks, timers, etc., are reproduced in Appendix 13. Figure 6.29 illustrates the first of the receivers (that worked), as set up in the laboratory at the University of Warwick. The computer display shows the recovered data and the disassembled code. The spectrum analyser shows the spectrum of the combined signal produced by the two transmitters with codesets 1 and 2

fed into them from the EPROM, that is linear-independent and unique 3-bit codewords.

Figure 6.30 illustrates the two Texas Instruments TMS320C50 fixed-point digital signal processor development boards, which were used as filters in the receiver, connected into the in-phase and the quadrature feeds from the intermediate frequency board. They are not connected into the system 1 set-up (as shown in Figure 6.29). The computer in the background is used to download code to the digital signal processors and to program the EPROMs with the test-pattern data used for synchronisation and testing.

The receiver boards, viewed left to right, are:

i) Timing recovery board (buffer removed)

ii) Maxim demodulator board

iii) Burr-Brown precision analogue-to-digital converter board

iv) Texas Instruments DSP board.

The next sub-section explains the operation and implementation of the software developed for the CV-CCMA system.

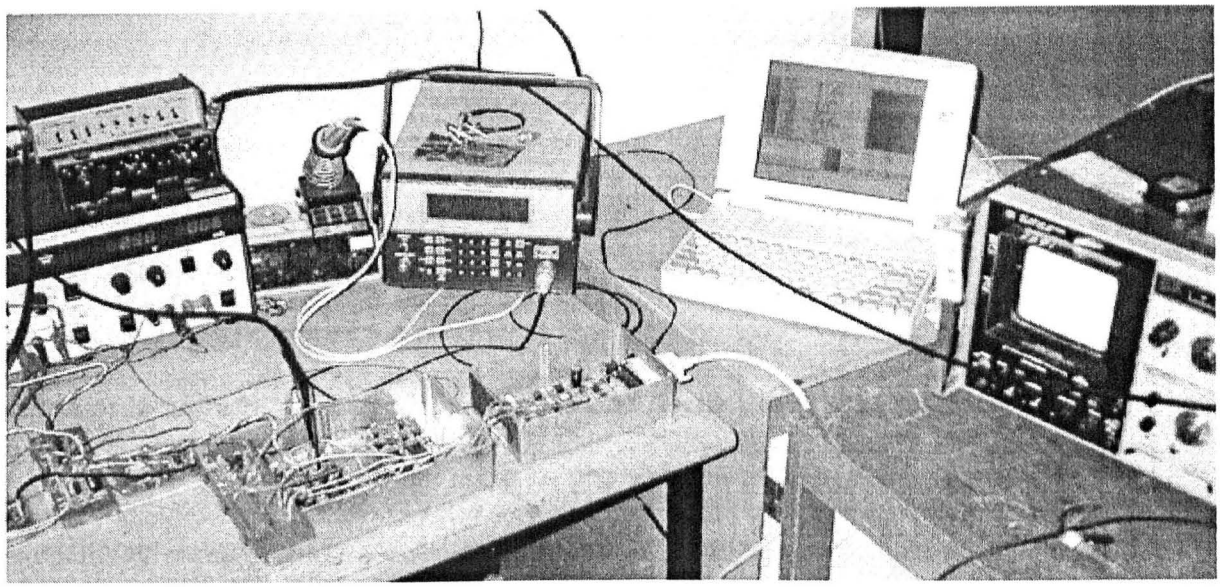


Figure 6.29 Receiver 1 in the laboratory at Warwick

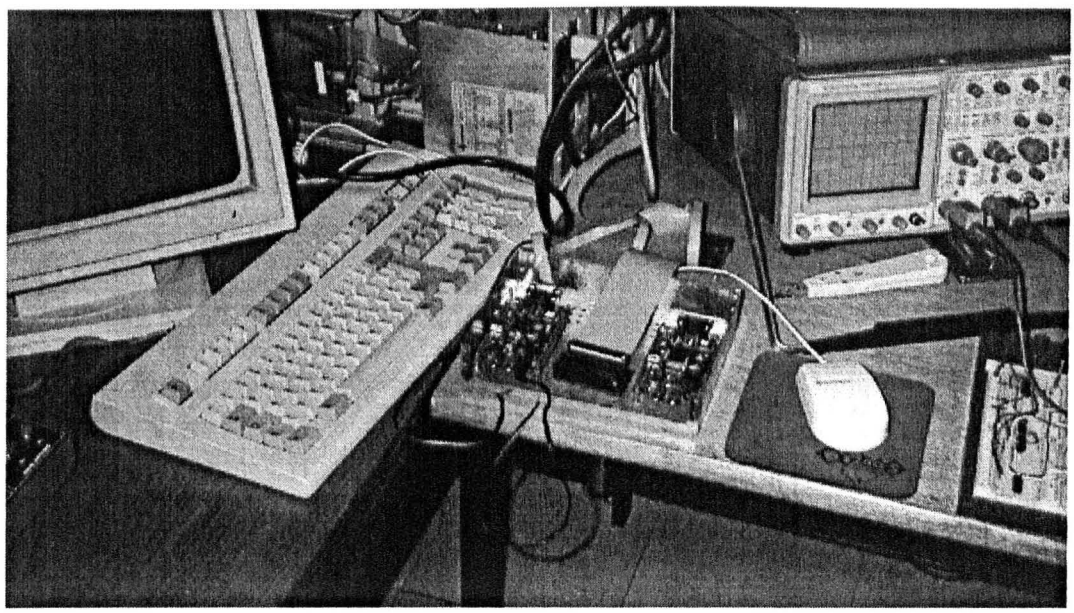


Figure 6.30 Receiver filter implementation

6.4 Software implementation

6.4.1 Introduction

A number of digital signal processors were investigated for possible use with the CV-CCMA system. The choice was restricted to relatively few devices due to the unique nature of the system and the processing required. Firstly, the signal processor needed to be a floating-point device because of the very large differences in the signal strengths of received signals, especially in a mobile radio implementation. The Texas Instruments digital signal processor was fully floating-point compatible; although the floating-point system used by Texas is unique, the numbers are easily translated to fixed-point format and IEEE floating-point format. This made interfacing with other systems easier. The second consideration was processing power. The DSP needed to perform a minimum number of floating-point operations in order for the system to function. The processing power requirements are dependent on the software that is running, and a more detailed analysis is worked through later in this chapter. In the first instance, processing power of the order of five million floating-point operations per second gives an approximate minimum requirement. This takes into account an overhead for system management. The third consideration was cost, and on this score the Texas Instruments devices won easily. The Analogue Devices and the Hewlett Packard processors were shortlisted, but the Texas Instruments floating-point signal processor was chosen because it was also the easiest to interface with external circuitry. Also, development software was issued free by Texas Instruments and a number of development boards were available in the telecommunications laboratory at the University of Warwick. Cable and Wireless PLC also supplied a Texas Instruments development system. This also helped to keep the costs down.

6.4.2 Software operations

A number of operations needed to be carried out prior to running the main receiver decode software. The first operation that needed to be performed was the calculation of the $\boldsymbol{I\text{-}S.K}$ matrices (where $\boldsymbol{I}$ = identity matrix, $\boldsymbol{S}$ = codewords and $\boldsymbol{P}$ = pseudo-inverse) *Mathcad* and *Matlab* programs were written to do this as it is very laborious to calculate pseudo-inverses by hand. A pseudo-inverse is the inverse of a non-square matrix. The complete programs are reproduced in Appendix 2. A flowchart illustrating the process is reproduced in Figure 6.31. Shortened versions of the *Mathcad* programs, with results obtained for standard linear independent codewords (test pattern 1) with QPSK modulation and BPSK modulation (test pattern 2), are reproduced later in this section.

When the pre-calculation program was run, two sets of output data were produced: a standard denary number array which was used in the simulation program and any rough calculation in case of errors; and a number array consisting of the results, the input combination giving rise to the set of results. Also, if required, the numbers could be output in Texas Instruments floating-point number format, ready to be input into the data array of the digital signal processor. These calculated results were loaded manually (via an assembly language array) into the signal processor memory. In a production device these would all be loaded into an EPROM and codeword selection achieved via a simple switch connected to the address lines. This would then access the pre-calculated matrices. These could be calculated on-line using a more powerful signal processor. Once loaded the signal processor was initialised, all the registers to be used were flushed and interrupts correctly primed and set-up. The various output strings required are reproduced in detail in Appendix 14 and a summary of the set-up routine is reproduced in flowchart form in Figure 6.32.

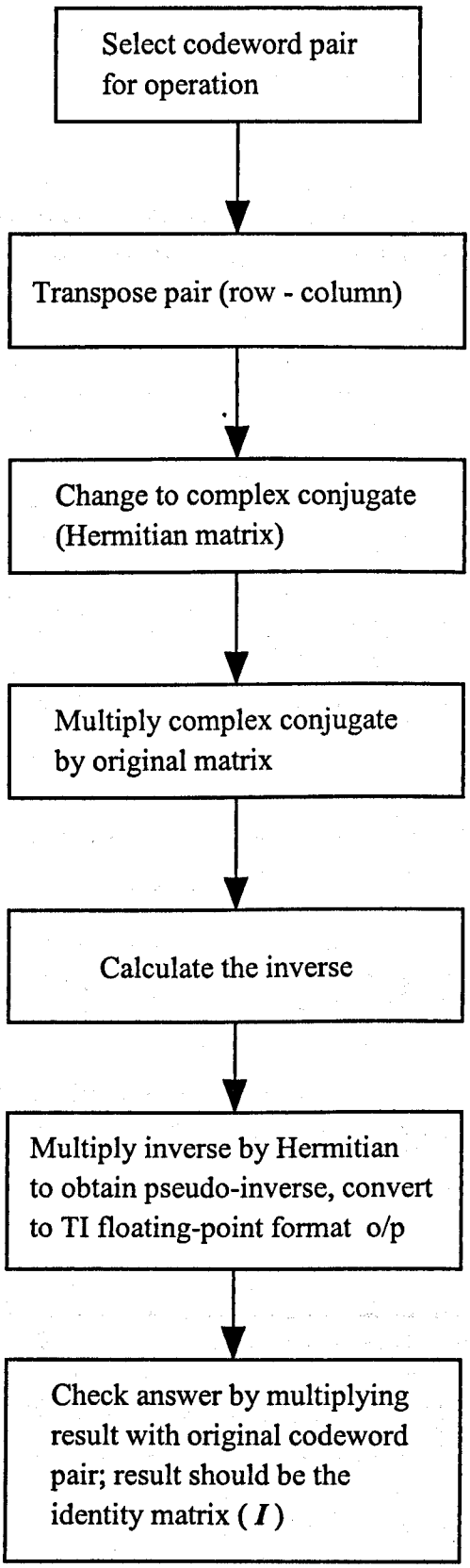


Figure 6.31 Pseudo-inverse calculation flowchart

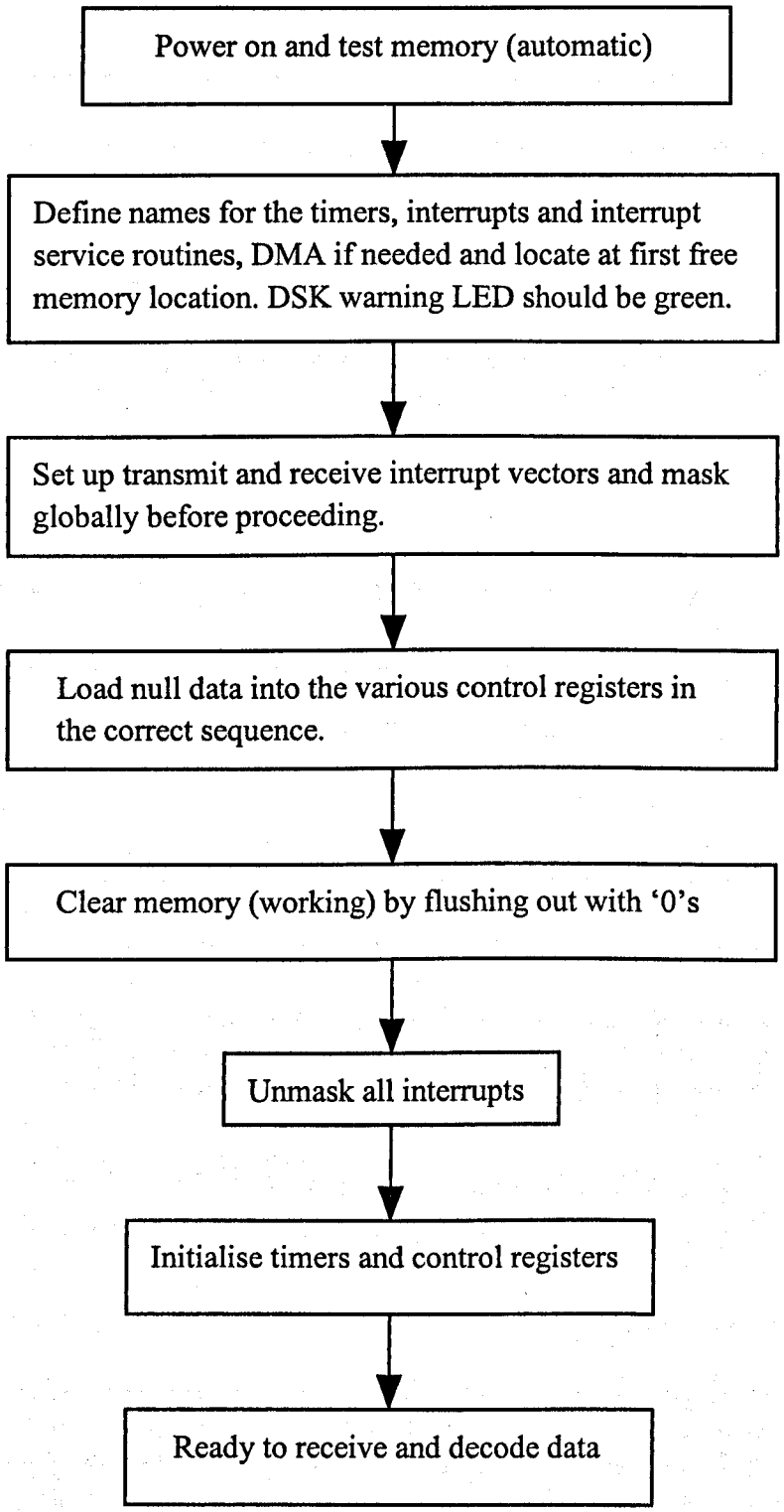


Figure 6.32 Digital signal processor initialisation flowchart

Also, at this stage, global control and receive/transmit timing were initialised. The stand-alone signal processor (TMS320C30) implementation required much more rigorous attention to detail in the programming, particularly in the set-up stages. The various registers all had to be initialised in the correct sequence. The development system, using the Texas TMS320C31, was far simpler to initialise due to the inclusion of a basic 'kernel' of code, ensuring that the processor was always correctly initialised. This was for operation with an on-board speech interface, and always performed at power-up and was easily selected from the personal computer during operation.

To ensure correct system timing, both hardware and software interrupts were used to correctly time the data into and out of the system. The bench demonstrator system (types 1-3) included a set-up stage in which the two transmitter outputs are connected together and then fed to the receiver input. The attenuators and one phase shifter were set to zero phase shift with a fixed attenuation only (e.g. 1 + j0, j0 = no phase shift). The other attenuator has fixed attenuation together with a fixed phase shift (e.g. 0.5 + j0.5). These values were then programmed into the DSP. The received values in this state were stored as the reference value to be used later when calculating $\boldsymbol{g}$ values. Some time was spent trying to develop a system which would enable the transmitters and receivers to synchronise on start-up. The problem is that if the combined states are uniquely definable they can be decoded, so long as we send a start code, as in asynchronous data systems, to enable correct resolution. However, if we do not know the gain and phase shift occurring between the transmitters and the receivers we cannot say that they are both transmitting and if not thus producing a false start point and incorrect resolution. This was the purpose of the synchronisation sequence run at the start of the test sessions, which also gave confidence in the

workings of all the component parts of the system before switching to the decode algorithm.

Most of the theoretical work on MACs has assumed perfect synchronisation; however, this was without giving any ideas how to achieve it. Cover *et al* [52] did some interesting work and envisaged a 'quasi-synchronous' channel sometimes called 'mild synchronisation' in which the decoders are kept in synchronisation with the encoders, but the encoders are allowed to slip out of synchronisation with each other by a restricted amount. They illustrated the effect on the capacity of the MAC; not suprisingly it dropped.

In the practical implementation of the system it was found that with an initial training sequence of patterns 101 – 010 – 101, etc., with known ***g*** values, the symbol synchronisation could be easily established. In the type 1 system the bit synchronisation is assumed to be perfect because the transmitters and receivers are all hardwired together. In the type 3 radio system a background timing channel is used, similar to the CDMA synchronisation channels (a CDMA forward link actually consists of four channel types, pilot, paging, synchronisation and traffic). This increases bandwidth usage, but all stations use the same synchronisation channel. The areas of timing and synchronisation are important to the operation of any multi-access system and further research and development is required in this area. The concept of a separate timing channel, with other functionality, has already been discussed in the theory section and this would appear to be the best way to progress.

The final stages of the set-up procedure, for the type 3 system, comprised initialising and running the DSP filters (they are bypassed in the type 1 system). These are pulse-shaping filters that were placed in the I and Q data streams. The filters were implemented using Texas Instruments TMS320C50DSK

development systems, and using industry standard code. The performance of the fixed-point digital signal processor is sufficient for this simple filter function. It is planned for the filter function to be incorporated into the main digital signal processor function as development progresses with the new digital signal processor.

When the set-up stage was completed the receiver idled until the first interrupt was received. An EINT (external interrupt) was used to time the initial bit and then RINT (receive interrupt) to time the symbols that were required to be loaded into the digital signal processor. The RINT is a simple load counter, so whatever information was received, as a codeword, was loaded directly, and was then processed by the digital signal processor. The Burr-Brown analogue-to-digital converter had an output to indicate that the input was out of conversion range. This was used as an input to an interrupt line on the signal processor and then used to interrupt the decode and jump to the next set of inputs. This section of the programming caused a lot of problems; the DSP locked up and caused the receiver to crash. The problem was tracked down to the addresses that were used by the DSP to communicate with the PC. The code was nulled out in the prototypes. This could be reinstated quite easily, when de-bugged, and fitted into a production unit.

The sampled receive vector was then loaded into memory, as a floating-point number. Later versions of the software, designed to operate with the type 3 system, used a multiple-sample decode process. This was developed to reduce the systems timing sensitivity and involved 3 evenly-spaced samples being taken and resolved. The decoded output was taken as the agreement between 2 of the 3 samples, or if no agreement, then the output becomes the sample with the minimum distance to a valid combination.

There was some initial mathematical manipulation required because of the two numbers being concatenated and this was performed with a series of shift commands. The two floating-point numbers were then stored in the digital signal processor's precision registers. The multiplication of the pre-calculated matrices with the received vector was then performed as a series of nested loops. In the BPSK system this produced four floating-point numbers that were stored in the four precision registers, R4, R5, R6 and R7. The minimum value of Euclidean norm was then selected by a series of interconnected sub-routines. Included at the end of each sub-routine is a simple load and output which effectively transmits the recovered data back out of the serial data port. This was then clocked into a buffer and fed to the Delta demodulator or PC for decoding. The position of the bits in the buffer determines the destination address, for example in the two-transmitter set-up this is simply receiver output 1 or output 2. The program then jumped back to await the next interrupt, indicating the arrival of the next set of codewords. With a 40 ns instruction time the maximum symbol clock frequency worked out to approximately 21 KHz. This does not directly translate to data rate due to the synchronisation/coding procedure but is still equivalent to the capacity of current GSM data capability and was adequate for the test set-up. Due to the additional processing required, the multi-sample system has a reduced data rate, results were obtained with rates up to 1.2 Kb/s, these are reproduced in Chapter 7.

Digital signal processor systems are designed to perform floating-point multiplications very rapidly. A selection of commands are also available to perform mutiplication simultaneously with load and shift commands. A typical floating-point multiplication and floating-point addition only takes a single cycle, and these were widely used in the programming. The Euclidean norm calculation initially involved an

iterative process; this proved too long for the system to operate with 50 MHz processing. This was reduced to 50 cycles in order to reduce the processing time, and then further reduced to a simple self-multiplication, to remove any negative values. This did not cause any obvious loss of accuracy and meant that the TMS320C31 development board could be used in the receiver. The latest DSP devices have orders of magnitude more processing power and it is planned to rebuild the system using one of these devices. This will require re-programming, however the Texas digital signal processors are generic, this means a lot of the earlier work can easily be 'ported' onto the later devices.

The complete receiver decode program listings, which were written in Texas Instruments assembler code, are included in Appendix 1. The 'C' programs that were written were found to compile to twice the size of the assembler programs. The performance, especially the sort routines, was found to be inferior. This was a weakness of the compiler; later versions of the compiler are reported to have much improved performance.

Figure 6.33 illustrates, in a flowchart format, the operation of the overall receiver (type 1) system. Figure 6.34 illustrates the multi-sample (type 3) system, with the additional selection processes shown at the bottom of the diagram. Figure 6.35 illustrates the input store and multiplication routines, the multiplication routines were found to have the fastest resolution when calculated as nested loops, as shown in the diagram. Figure 6.36 illustrates the BPSK Euclidean norm minimum calculation, and Figure 6.37 the QPSK Euclidean norm minimum calculation, all are drawn in flowchart form.

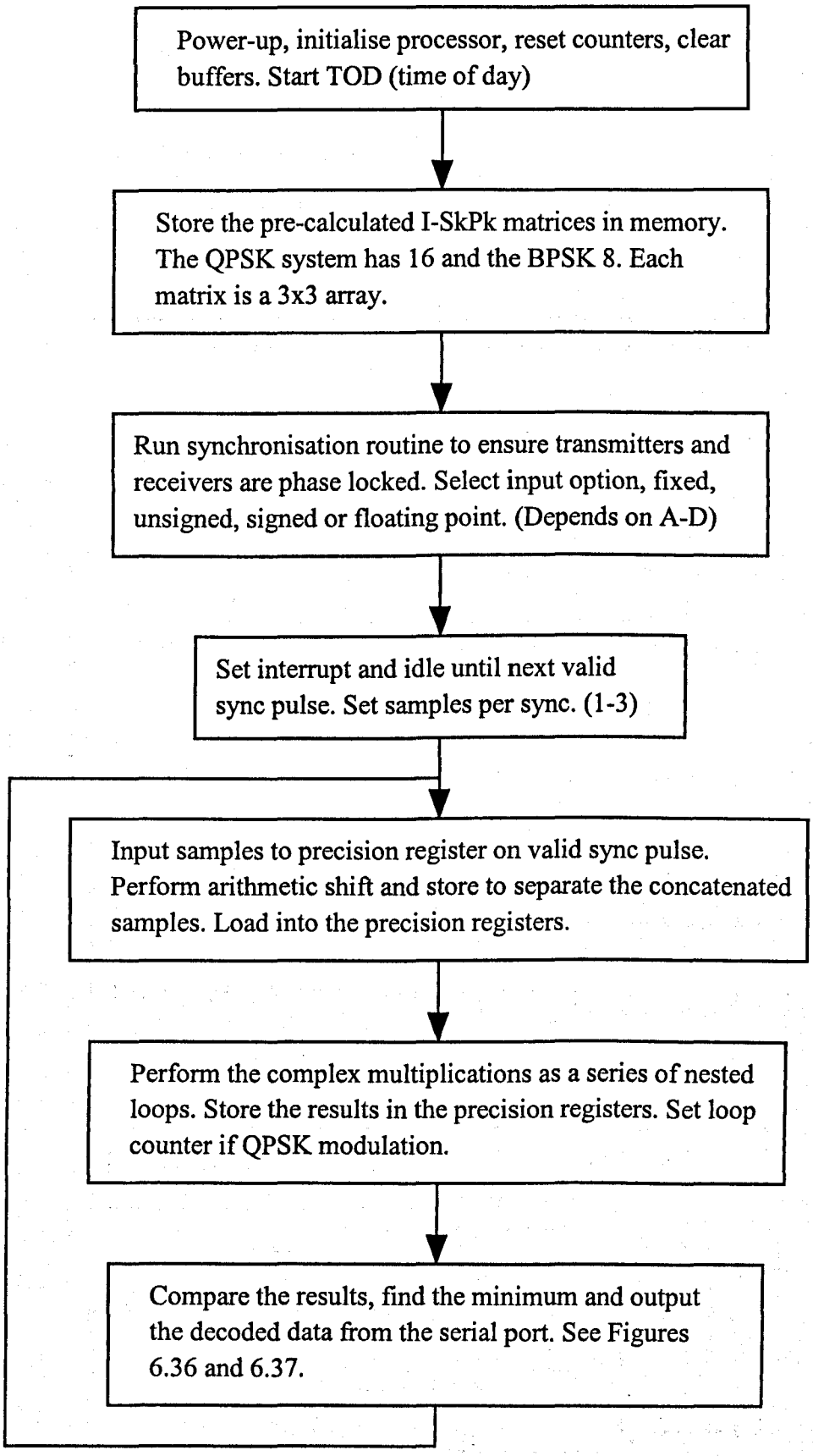


Figure 6.33 Digital signal processor receiver flowchart (type 1)

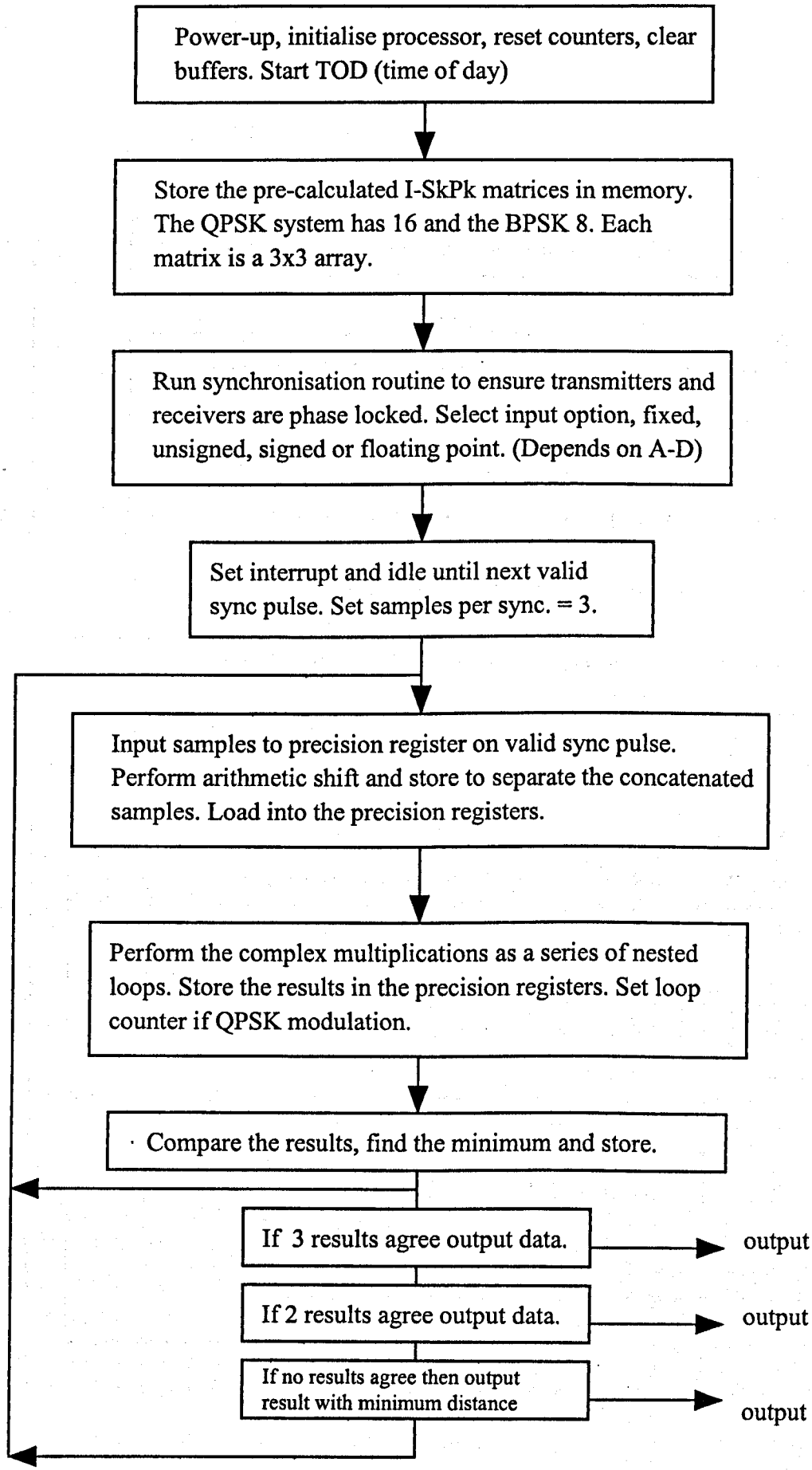


Figure 6.34 Digital signal processor receiver flowchart (type 3)

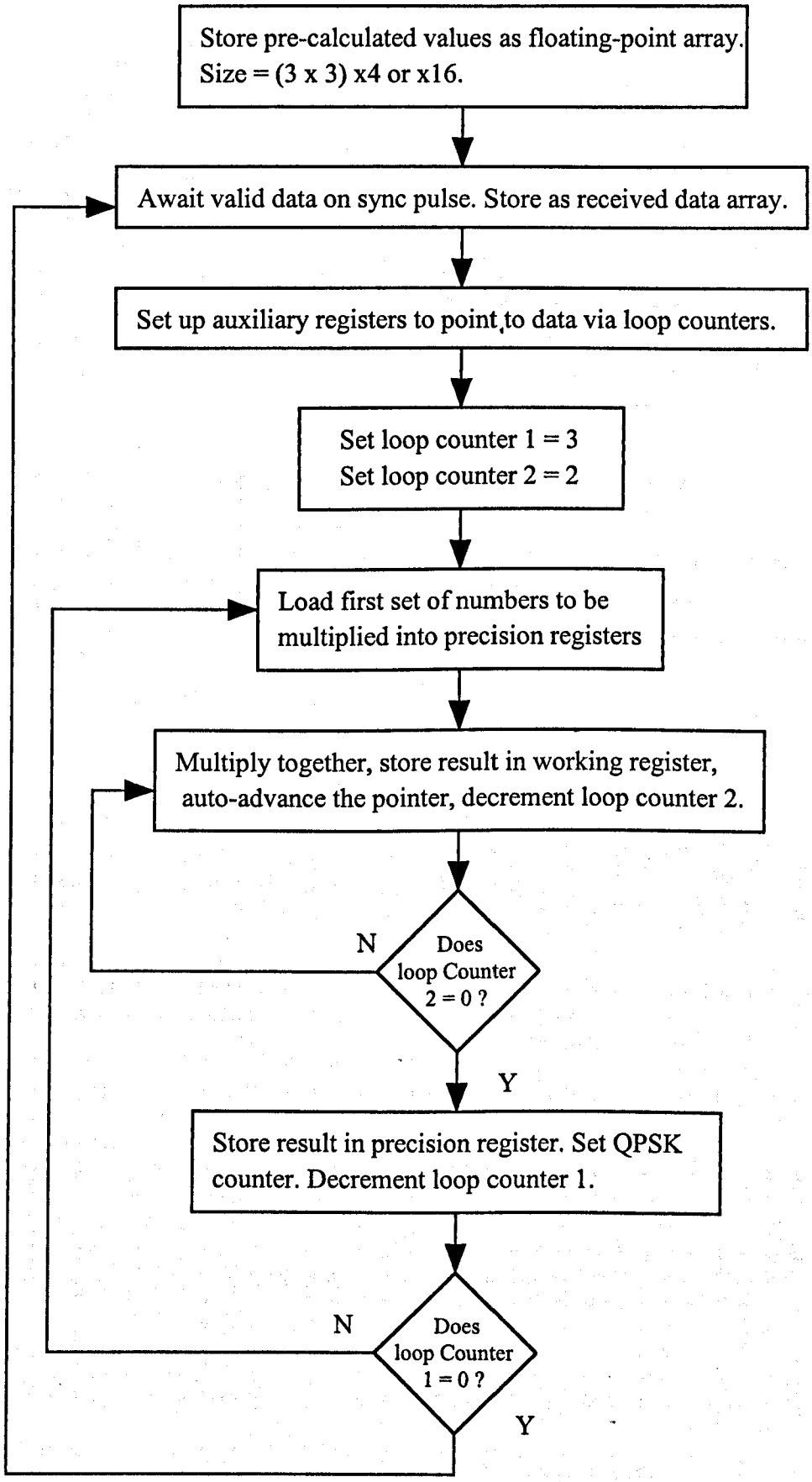
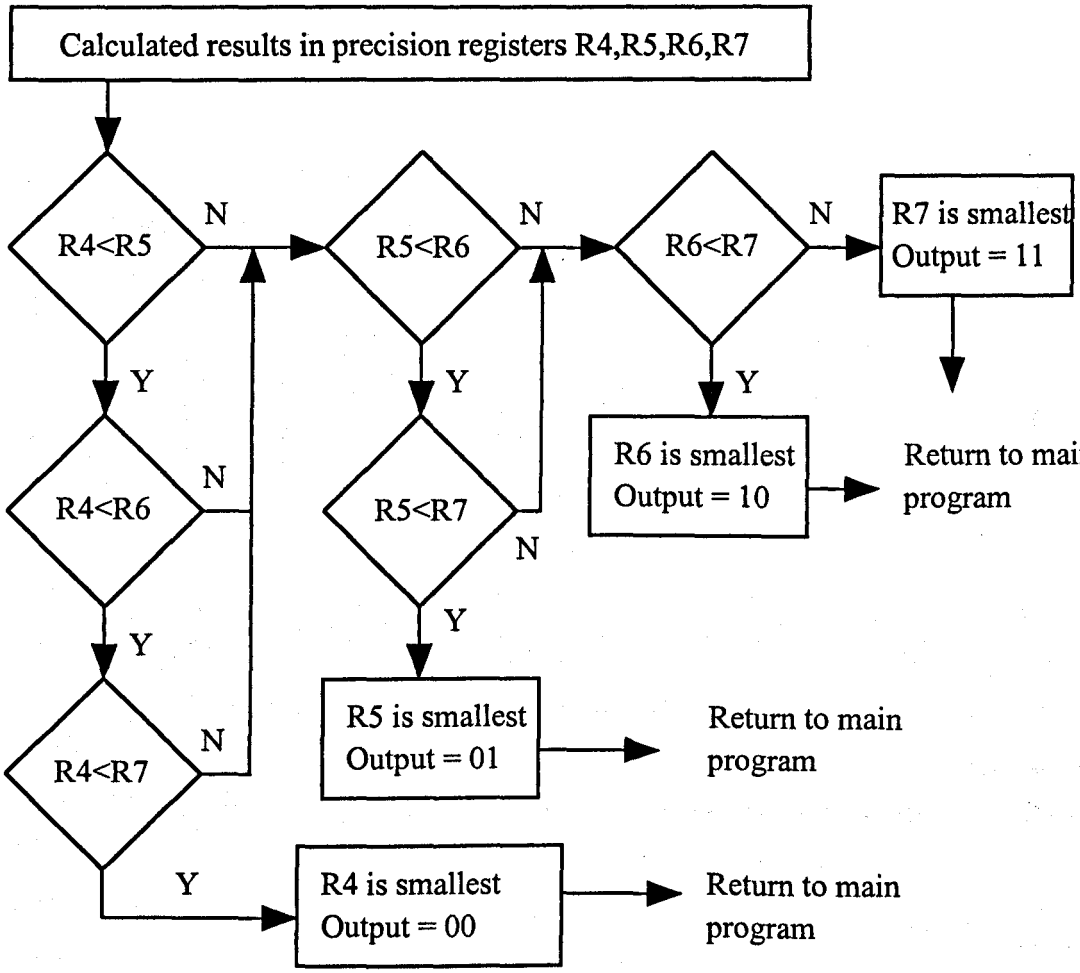


Figure 6.35 Multiplication routines flowchart



Assembler Code

```
LDI     0000h,R0        Output when R4 contains smallest value
LDI     0001h,R1        Output when R5 contains smallest value
LDI     0100h,R2        Output when R6 contains smallest value
LDI     1000h,R3        Output when R7 contains smallest value
CMPF    R5,R4           Compare floating point
CALLLT  NEG1            Call subroutine Neg 1 if less than
CALL    NEG2            Call subroutine Neg 2
B       SLOOP           Branch if floating overflow or underflow
NEG1    CMPF  ,R6,R4
CALLGT  NEG2
CMPF    R7,R4
CALLGT  NEG2
STI     R0,@DATAOP      (R0) to serial port
B       SLOOP
NEG2    CMPF,R6,R5
CALLGT  NEG3
STI     R1,@DATAOP      (R1) to serial port
B       SLOOP
NEG3    CMPF,R7,R6
CALLGT  NEG4
STI     R2,@DATAOP      (R2) to serial port
B       SLOOP
NEG4    STI,R3,@DATAOP  (R3) to serial port
B       SLOOP           Return to main program
```

Figure 6.36 Sort minimum routine (BPSK)

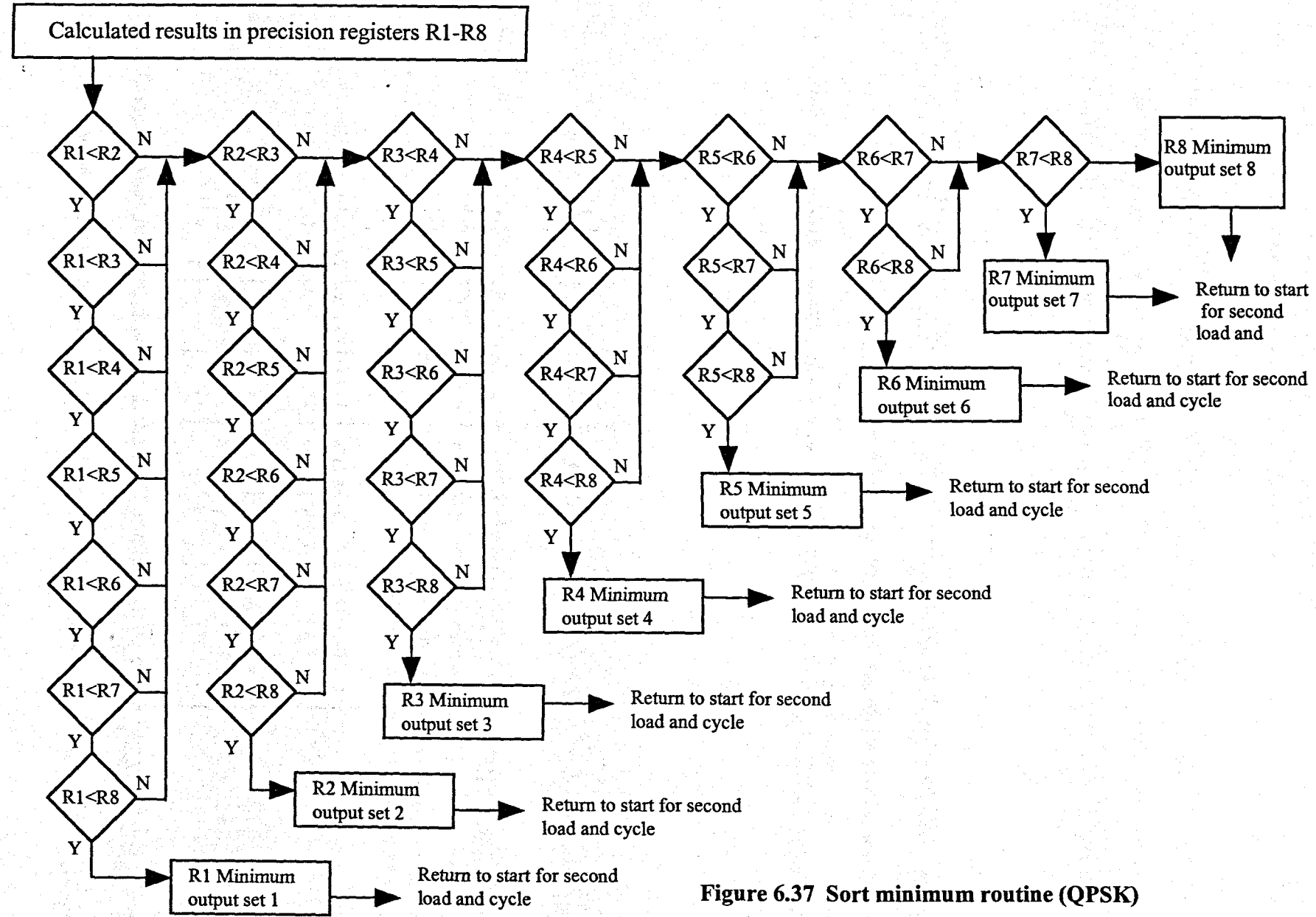


Figure 6.37 Sort minimum routine (QPSK)

6.4.3 Software emulation of receiver (QPSK) combinational matrix calculations

To illustrate the operation of the decoder software, the following is a shortened *Mathcad* program, which solves for the generalised inverse and checks the result. It works well with linearly independent sets of vectors. A more general solution and program is included in Appendix 2 which will work with all codewords.

The Moore-Penrose generalised inverse or pseudo-inverse of a non-square matrix can be found by multiplying the Hermitian result of the matrix with itself, then finding the inverse and multiplying again with the Hermitian matrix. A Hermitian matrix is simply the complex conjugate transpose of the original matrix. Note that ':=' is a *Mathcad* symbol which forces the program to solve the equation. Defining the codeword pairs, $P1 - P16$:

$$P1 := \begin{bmatrix} -j & -j \\ -j & 1 \\ 1 & 1 \end{bmatrix} \quad P2 := \begin{bmatrix} -j & -j \\ -j & 1 \\ 1 & -1 \end{bmatrix} \quad P3 := \begin{bmatrix} -j & -j \\ -j & 1 \\ 1 & j \end{bmatrix} \quad P4 := \begin{bmatrix} -j & -j \\ -j & 1 \\ 1 & -j \end{bmatrix}$$

$$P5 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -1 & 1 \end{bmatrix} \quad P6 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -1 & -1 \end{bmatrix} \quad P7 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -1 & 1 \end{bmatrix} \quad P8 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -1 & -j \end{bmatrix}$$

$$P9 := \begin{bmatrix} -j & -j \\ -j & 1 \\ j & 1 \end{bmatrix} \quad P10 := \begin{bmatrix} -j & -j \\ -j & 1 \\ j & -1 \end{bmatrix} \quad P11 := \begin{bmatrix} -j & -j \\ -j & 1 \\ j & j \end{bmatrix} \quad P12 := \begin{bmatrix} -j & -j \\ -j & 1 \\ j & -j \end{bmatrix}$$

$$P13 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -j & 1 \end{bmatrix} \quad P14 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -j & -1 \end{bmatrix} \quad P15 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -j & j \end{bmatrix} \quad P16 := \begin{bmatrix} -j & -j \\ -j & 1 \\ -j & -j \end{bmatrix}$$

Codewords were selected to match the type1 implementation. The first symbol is the same for all transmitters, to aid recovery and to match the work done on codeword

selection, recorded in Chapter 5 of this thesis. Selecting a codeword combination for illustration: $= P1$

$$= \begin{bmatrix} -j & -j \\ -j & 1 \\ 1 & 1 \end{bmatrix}$$

Calculate PT, the transpose of $P1$: (press ctrl +1)

$$PT := P1^{T}$$

$$PT = \begin{bmatrix} -j & -j & 1 \\ -j & 1 & 1 \end{bmatrix}$$

Calculate R, the complex conjugate: (press ")

$$R := \begin{bmatrix} j & j & 1 \\ j & 1 & 1 \end{bmatrix}$$

Calculate Z, the first product ;

$$Z := (R\ P1)$$

$$Z = \begin{bmatrix} 3 & 2+j \\ 2-j & 3 \end{bmatrix}$$

Then calculate Y, the Inverse:

$$Y := Z^{-1}$$

$$Y = \begin{bmatrix} 0.75 & -0.5-0.25 \\ -0.5+0.25j & 0.75 \end{bmatrix}$$

Calculate $W1$, the Pseudo-inverse:

$$W1 = Y.R$$

$$W1 = \begin{bmatrix} 0.25+0.25j & -0.5+0.5j & 0.25-0.25j \\ -0.25+0.25j & 0.5-0.5j & 0.25+0.25j \end{bmatrix}$$

Finally, check result: $C = W1\ .P1$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This proves that $W1$ is the general inverse of $P1$ because the product is the I matrix. This program is run for the 16 sets of codeword combinations possible with the QPSK system. The program now calculates the $I\text{-}S_kP_k$ matrices ready to be stored in the DSP. Define I, identity matrix:

$$I := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Define F as calculation of $I\text{-}S_kP_k$ matrices:

$$F := I - (P1.W1)$$

Taking the first combination :

$$F1 = \begin{bmatrix} 0.5 & 0 & 0.5j \\ 0 & 0 & 0 \\ -0.5j & 0 & 0.5 \end{bmatrix}$$

The rest of the matrix array is calculated in a similar manner and loaded into the memory of the DSP. Table 6.2 is a summary of the results obtained.

6.4.4 QPSK decode example

As an example to test the results, and prove the viability of the system, two transmitters and one receiver are simulated with g_1 and g_2 as below; *Mathcad* is used to check the mathematical integrity of the system.

$$g_1 := 0.2 + 0.9j$$

$$g_2 := 0.8 + 0.6j$$

Assuming combination codeword set 15 is transmitted, that is transmitter 1 transmits (–j 1 j) and transmitter 2 transmits (–j –j –j) :

Calculate the Received vector = Rv

I^* / $-S^*_P$	-j 1 1	-j 1 -1	-j 1 +j	-j 1 -j
-j -j 1	$\begin{bmatrix} 0.5 & 0 & 0.5j \\ 0 & 0 & 0 \\ -0.5j & 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.25 & -0.25+0.25j & -0.25 \\ -0.25+0.25j & 0.5 & 0.25+0.25j \\ -0.25 & 0.25-0.25j & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5j \\ 0 & 0.5j & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.25 & -0.25+0.25j & -0.25 \\ -0.25+0.25j & 0.5 & 0.25+0.25j \\ -0.25 & 0.25-0.25j & 0.25 \end{bmatrix}$
-j -j -1	$\begin{bmatrix} 0.25 & -0.25+0.25j & 0.5j \\ -0.25+0.25j & 0.5 & -0.25+0.25j \\ 0.25 & -0.25+0.25j & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0 & -0.5j \\ 0 & 0 & 0 \\ 0.5j & 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 1 & -0.5+0.5j & 0.5-0.5j \\ -0.5+0.5j & -0.5j & 0.5j \\ -0.5+0.5j & -0.5j & 0.5j \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & -0.5j \\ 0 & 0.5j & 0.5 \end{bmatrix}$
-j -j +j	$\begin{bmatrix} 0.5 & 0 & 0.5j \\ 0 & 0 & 0 \\ -0.5j & 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.25 & -0.25+0.25j & 0.25 \\ -0.25-0.25j & 0.5 & 0.25-0.25j \\ -0.25 & 0.25+0.25j & 0.25 \end{bmatrix}$
-j -j -j	$\begin{bmatrix} 0.5 & 0 & 0.5j \\ 0 & 0 & 0 \\ -0.5j & 0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.5 & -0.25+0.25j & -0.25-0.25j \\ -0.25-0.25j & 0.25 & -0.25j \\ -0.25+0.25j & -0.25j & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.25 & -0.25+0.25j & -0.25j \\ -0.25-0.25j & 0.5 & -0.25+0.25j \\ +0.25j & -0.25-0.25j & 0.25 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$

Table 6.2 Codeword Test Set 1 - summary of I-S.P matrix calculations

$$R_V := g1.(-j \quad 1 \quad j) + g2.(-j \quad -j \quad -j)$$

$$R_V := (1.5 - j \quad 0.8 + 0.1j \quad -0.3 - 0.6j)$$

Now calculate the transpose of the receive vector:

$$R_V{}^T = \begin{bmatrix} 1.5 - j \\ 0.8 + 0.1j \\ -0.3 - 0.6j \end{bmatrix}$$

Now multiply the receive vector by the $I\text{-}S_kP_k$ matrices from Table 6.2.

$$R1 := F1.R_V{}^T$$

$$R1 = \begin{bmatrix} 1.05 - 0.65j \\ 0 \\ -0.65 - 1.05j \end{bmatrix}$$

$$|R1| = 1.746$$

$$R5 := F5.R_V{}^T$$

$$R5 = \begin{bmatrix} 0.075 - 0.225j \\ -0.3 + 0.15j \\ -0.075 - 0.225j \end{bmatrix}$$

$$|R5| = 0.474$$

$$R2 := F2.R_V{}^T$$

$$R2 = \begin{bmatrix} 0.225 + 0.075j \\ -0.15 - 0.3j \\ -0.225 - 0.075j \end{bmatrix}$$

$$|R2| = 0.474$$

$$R6 := F6.R_V{}^T$$

$$R6 = \begin{bmatrix} 0.45 - 0.35j \\ 0 \\ 0.35 + 0.45j \end{bmatrix}$$

$$|R6| = 0.806$$

$$R3 := F3.R_V{}^T$$

$$R3 = \begin{bmatrix} 0 \\ 0.7 - 0.1j \\ -0.1 - 0.7j \end{bmatrix}$$

$$|R3| = 1$$

$$R7 := F1.R_V{}^T$$

$$R7 = \begin{bmatrix} 0.6 - 0.8j \\ 0.1 + 0.7j \\ 0.1 + 0.7j \end{bmatrix}$$

$$|R7| = 1.414$$

$$R4 := F4.R_V{}^T$$

$$R4 = \begin{bmatrix} 0.75 - 0.25j \\ -0.5 - 0.25j \\ -0.5 - 0.25j \end{bmatrix}$$

$$|R4| = 1.118$$

$$R8 := F8.R_V{}^T$$

$$R8 = \begin{bmatrix} 0 \\ 0.1 + 0.2j \\ -0.2 + 0.1j \end{bmatrix}$$

$$|R8| = 0.316$$

$$R9 := F9.R_V{}^T$$

$$R9 = \begin{bmatrix} 0 \\ -1.2 + 1.1j \\ 1.2 - 1.1j \end{bmatrix}$$

$$|R9| = 2.302$$

$$R13 := F13.R_V{}^T$$

$$R13 = \begin{bmatrix} 0 \\ 0.55 + 0.35\,j \\ -0.55 - 0.35\,j \end{bmatrix}$$

$$|R13| = 0.922$$

$$R10 := F10.R_V{}^T$$

$$R10 = \begin{bmatrix} 0 \\ 0.25 - 0.25\,j \\ 0.25 - 0.25\,j \end{bmatrix}$$

$$|R10| = 0.5$$

$$R14 := F14.R_V{}^T$$

$$R14 = \begin{bmatrix} 0.45 - 0.35\,j \\ 0 \\ 0.35 + 0.45\,j \end{bmatrix}$$

$$|R14| = 0.806$$

$$R11 := F11.R_V{}^T$$

$$R11 = \begin{bmatrix} -0.6 - 0.8\,j \\ 0 \\ -0.6 - 0.8\,j \end{bmatrix}$$

$$|R11| = 1.414$$

$$R15 := F15.R_V{}^T$$

$$R15 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$|R15| = 0$$

$$R12 := F12.R_V{}^T$$

$$R12 = \begin{bmatrix} 0.3 - 0.15\,j \\ -0.45 - 0.15\,j \\ -0.15 - 0.3\,j \end{bmatrix}$$

$$|R12| = 0.671$$

$$R16 := F16.R_V{}^T$$

$$R16 = \begin{bmatrix} 0.9 - 0.2\,j \\ 0 \\ -0.9 + 0.2\,j \end{bmatrix}$$

$$|R16| = 1.304$$

The matrices shown above illustrate how the minimum value is selected as the valid result corresponding to the original combination of codewords. The example shows the correct decoding of the transmitted codewords, i.e. transmitted set 15. Note that the example gives a flavour of the amount of processing required.

6.4.5 BPSK combinational matrix calculations

The calculations for the BPSK combinational $\boldsymbol{I}$-$\boldsymbol{S}_k\boldsymbol{P}_k$ matrices are much simplified due to *Mathcad* having the ability to calculate a real-valued pseudo-inverse (general inverse) directly. The following is a shortened *Mathcad* program to solve for the generalised inverse and check the result.

Using BPSK codeword pairs ($\boldsymbol{P}1$-$\boldsymbol{P}4$):

$$\boldsymbol{P}1 := \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \qquad \boldsymbol{G}1 := \text{geninv}(\boldsymbol{P}1) \qquad \boldsymbol{G}1 = \begin{bmatrix} 0.25 & -0.5 & 0.25 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}$$

$$\boldsymbol{P}2 := \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \boldsymbol{G}2 := \text{geninv}(\boldsymbol{P}2) \qquad \boldsymbol{G}2 = \begin{bmatrix} 0.5 & -0.25 & 0.25 \\ 0.5 & 0.25 & -0.25 \end{bmatrix}$$

$$\boldsymbol{P}3 := \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad \boldsymbol{G}3 := \text{geninv}(\boldsymbol{P}3) \qquad \boldsymbol{G}3 = \begin{bmatrix} 0.5 & -0.25 & -0.25 \\ 0.5 & 0.25 & 0.25 \end{bmatrix}$$

$$\boldsymbol{P}4 := \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \qquad \boldsymbol{G}4 := \text{geninv}(\boldsymbol{P}4) \qquad \boldsymbol{G}4 = \begin{bmatrix} 0.25 & -0.5 & -0.25 \\ 0.25 & 0.5 & -0.25 \end{bmatrix}$$

Now define the identity matrix $\boldsymbol{I}$:

$$\boldsymbol{I} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$F1 := I - (P1 \cdot G1)$

$$F1 = \begin{bmatrix} 0.5 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$$

$F2 := I - (P2 \cdot G2)$

$$F2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

$F3 := I - (P3 \cdot G3)$

$$F3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$$

$F4 := I - (P4 \cdot G4)$

$$F4 = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

The results are summarised in Table 6.3.

$I-S_k.P_k$	111	11-1
1-11	$F1 = \begin{bmatrix} 0.5 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix}$	$F2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0.5 & 0.5 \end{bmatrix}$
1-1-1	$F3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & -0.5 \\ 0 & -0.5 & 0.5 \end{bmatrix}$	$F4 = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}$

Table 6.3 Test set 1 summary

In order to check the results and operation of the system :–

g_1 and g_2 are the complex loss values in a 2 transmitter 1 receiver set-up.

$$g_1 := 0.9 + 0.2\,j$$

$$g_1 := 0.6 + 0.9\,j$$

Calculate received vector R_v :

$$R_v := g_1.(1\ \ 1\ \ -1) + g_2\,(1\ \ -1\ \ 1)$$

$$R_v = (1.5 + 1.1j \quad 0.3 - 0.7j \quad -0.3 + 0.7j)$$

Calculate the transpose $R_V{}^T$:

$$R_V{}^T = \begin{bmatrix} 1.5 + 1.1j \\ 0.3 - 0.7j \\ -0.3 \quad 0.7j \end{bmatrix}$$

Solving the vectors:

$$R1 := F1 \cdot R_V{}^T \qquad R1 = \begin{bmatrix} 0.9 + 0.2j \\ 0 \\ -0.9 - 0.2j \end{bmatrix} \qquad |R1| = 1.304$$

$$R2 := F2 \cdot R_V{}^T \qquad R2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad |R2| = 0$$

$$R3 := F3 \cdot R_V{}^T \qquad R3 = \begin{bmatrix} 0 \\ 0.3 - 0.7j \\ -0.3 + 0.7j \end{bmatrix} \qquad |R3| = 1.077$$

$$R4 := F4 \cdot R_V{}^T \qquad R4 = \begin{bmatrix} 0.6 + 0.9j \\ 0 \\ 0.6 + 0.9j \end{bmatrix} \qquad |R4| = 1.53$$

This illustrates that the two transmitted codewords would be correctly decoded when using BPSK modulation. Additional programs were written to illustrate the effect of varying g on the receiver decode capability and to illustrate the problems that arise when g varies during the length of a codeword; see Appendix 3.

6.4.6 Summary

The digital signal processor chosen to carry out the bulk of the receive processing was easily capable of decoding the combined codewords from the two transmitters in the CV-CCMA demonstrator. The software written for the decoder's digital signal processor, written in Texas assembler language, was developed to a functional state and could be ported easily onto other, more powerful, Texas Instrument's digital signal processors.

The *Mathcad* and *Matlab* programs that were written to test the viability of the decode algorithm were found to be useful, because they gave confidence in the operation of the CV-CCMA system. This was especially true during the initial period of the research, and gave confidence to the hardware development stages that followed. The *Mathcad* programs that were written to check system operation with transmission parameter variation also produced some useful results. The *Mathcad* programs that were written to calculate the matrix arrays were extremely useful, and saved a lot of time.

Most of the software is written in a modular format. This will enable the functionality of the component parts of the CV-CCMA system to be easily transported to other applications. For example, a new optical asynchronous transfer mode system, that is being trialled by Cable & Wireless PLC, could usefully adapt the multiply and select minimum routine to its receiver system.

Any future development of the decode software should be continued using Texas Instruments development tools; this would prevent the time-consuming problems (particularly with addressing) that slowed CV-CCMA development.

7 RESULTS

7.1 Introduction

This chapter contains results obtained with the system that was constructed during the latter period of work undertaken at the University of Warwick. The results mostly take the form of recorded data and waveforms, in both the time domain and frequency domain, separated into sections corresponding to the component blocks shown in the system diagram, Figure 7.1.

Using BPSK modulation, the system, although restricted in data rate, functions quite satisfactorily. The full bandwidth QPSK system worked on the bench with hardwire synchronisation, and with some fettling it was also possible to obtain reliable results over an open link. With the filters implemented, the parameters were too tight and some information was lost. Also, the gain and phase loss parameters are more difficult to interpret with a full QPSK modulation scheme using the integrated circuits selected. Therefore, when transmitters and receivers are hardwired together the g values needed to be very accurately converted before being fed into the signal processor. When analysing some of these factors it was found that a range of values can be open to interpretation. Different hardware decoding schemes would be a help here and further work needs to be done on this. Due to the superior reliability of the BPSK system most of the results quoted are for the BPSK system. A standard Hewlett Packard oscilloscope was used for generating the results and was fitted with a direct HPIB (IEEE488) output. This was used to input the data directly to the word processor, producing good-quality reproduction. The spectrum analyser, however, had no output, so the screens were printed, and then scanned. The results obtained with the two transmitter, one receiver system are compared with simulation results from other researchers, and the simulations are shown to be valid.

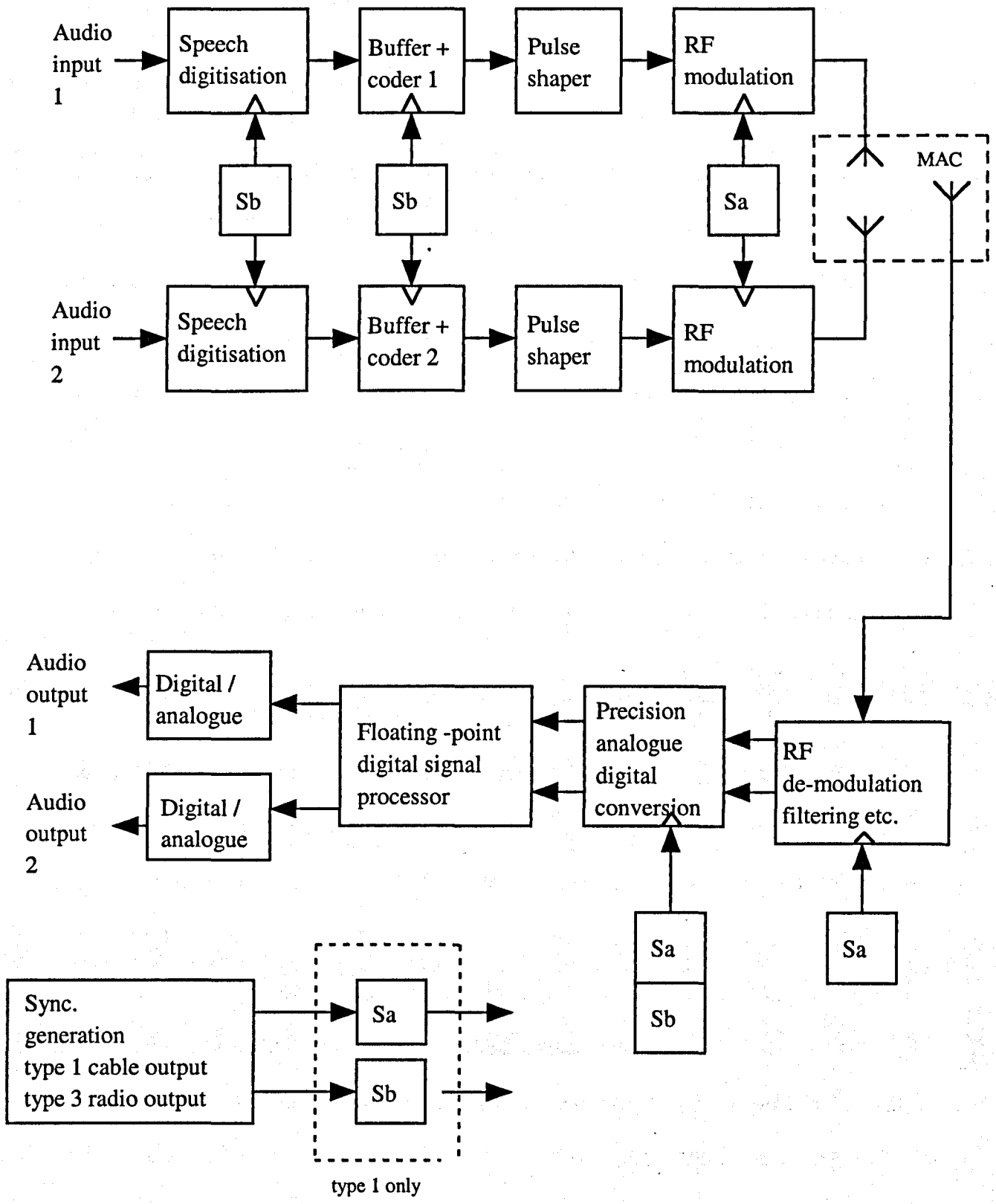


Figure 7.1 CV-CCMA Block Diagram

7.2 The synthesisers

The synthesisers were found to perform well, producing spectrally pure outputs at the designed output powers. The output spectrum of synthesiser 1 is shown in Figures 7.2, 7.3 and 7.4, which illustrate the lower frequency, standard frequency and upper frequency respectively. The nice clean spectrum with low level sidebands may be observed. Table 7.1 shows the standard input programming format for the synthesisers. The associated lock codes and lock times are shown in Tables 7.2, 7.3 and 7.4. The output spectrum of synthesiser 2 is shown in diagrams 7.5, 7.6 and 7.7, with associated lock codes in Tables 7.5, 7.6 and 7.7. Note that the output spectrum is reduced in amplitude (approximately 30 dB down on synthesiser 1) and has a broader spectrum and an inferior sideband performance. This was mainly due to the voltage-controlled oscillator used, which had been damaged by excessive input voltage (on the control input). Also, the printed circuit board layout is much more critical at these higher frequencies.

The delay in lock-up on synthesiser 2, when operating at 850 MHz, is due to non-linearities in the VCO output, causing interactions with the loop filter. This shows itself as a false lock. The lock codes, etc., transmitted from the PC should produce an output frequency of 850 MHz. However, the synthesiser locks up at about 840 MHz. It was resolved on the second models by forcing the VCO to output the correct frequency, by overloading the control input and then transmitting the lock codes. The solution is a new voltage-controlled oscillator.

At the higher frequencies components were 'taking off', or self-oscillating at different frequencies. This situation was corrected by inputting the correct I^2C codes whilst simultaneously setting the control voltage so that the output frequency was correct.

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit1	Bit0
A	00	PD	0	IPCD	X	RD3	RD2	RD1	RD0
B	01	1	0	1	PHI	VCOa	VCOb	MD17	MD16
C	10	MD15	MD14	MD13	MD12	MD11	MD10	MD9	M8
D	11	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0

Table 7.1 Synthesiser programming format

Required programming data:

MD0-MD17 = Main dividers division ratio load
RD3-RD0 = Reference dividers division ratio load
PD = Power down
PCD = Current pump, 0.5 mA or 1 mA
VCO A / B = Enables two synthesisers with same address on same bus to . be uniquely identified.

Note that device address = 14, sub-address 00 with auto-increment, four data bytes.

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	0	1	0	0
C	10	0	0	0	1	0	1	0	0
D	11	0	0	0	0	0	0	0	0

Table 7.2 I^2C data transmitted (50 MHz output)

Programming data summary;

10 MHz reference / 1024 = 9765.625 Rd codes = 1100

50 MHz output / 5120 = 9765.625 Md codes = 000001010000000000

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	0	1	0	0
C	10	0	0	1	0	0	1	0	0
D	11	0	0	0	0	0	0	0	0

Table 7.3 I^2C data transmitted (90 MHz output)

Programming data summary:

10 MHz reference / 1024 = 9765.625 Rd codes = 1100

90 MHz output / 9216 = 9765.625 Md codes = 000010010000000000

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	0	1	0	0
C	10	0	0	1	0	1	0	0	0
D	11	0	0	0	0	0	0	0	0

Table 7.4 I^2C data transmitted (100 MHz output)

Programming data summary~:

10 MHz reference / 1024 = 9765.625 Rd codes = 1100

100 MHz output / 10240 = 9765.625 Md codes = 000001010000000000

The 100 MHz spectrum, illustrated in Figure 7.3, shows a lower output level (-3.3 dBm) and worse sideband (not measured) performance; this was due to the voltage-controlled oscillator operating on the edge of its specified frequency range.

The Toshiba PC with I^2C interface generated all synthesiser programming codes. The device is first addressed, then sub-addressed and finally the data bytes are transmitted. Pin 15 on the synthesiser is an out-of-lock alarm and is held high by an external 27 KΩ resistor. The pin is latched low in the event of an out-of-lock condition and is used in the circuit to trigger an alarm. The internal registers of the synthesiser can be read back into the PC via the I^2C bus and checked for correct programming. On synthesiser 2 there was a delay in the VCO changing frequency so an out-of-lock signal was latched in before the VCO had settled at the new frequency. A reload of the I^2C data cured this; a faster/more linear VCO would improve any production unit.

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	1	0	0	1
C	10	0	0	0	0	0	1	1	1
D	11	0	0	1	1	0	0	1	1

Table 7.5 Synthesiser 2 I^2C data transmitted
658 MHz output, no pre-scaler

Programming data summary:

10 MHz reference / 1024 = 9765.625 Rd codes = 1100
658 MHz output /67379.2 = 9765.625 Md codes = 010000011100110011

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	1	0	0	1
C	10	0	1	0	1	0	1	0	0
D	11	0	0	0	0	0	0	0	0

Table 7.6 Synthesiser 2 I^2C data transmitted
(850 MHz output, no pre-scaler)

Programming data summary:

10 MHz reference / 1024 = 9765.625 Rd codes = 1100
850 MHz output / 87040 = 9765.625

Md codes = 010101010000000000

Register	Pointer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	00	0	0	1	X	1	1	0	0
B	01	1	0	1	0	1	0	0	1
C	10	1	0	1	1	1	0	0	1
D	11	0	0	1	0	1	1	1	0

Table 7.7 Synthesiser 2 I^2C data transmitted
(1.12095 GHz output, no pre-scaler)

Programming data summary:

10 MHz reference / 1024 = 9765.625 Rd codes = 1100

1.10295 GHz output /112,942.08 = 9765.625

Md codes = 011011100100101110

Synthesiser	Frequency (MHz)	Start-up lock time (warm TCXO) (mS)	Lock time from receipt of correct I^2C code (mS)
1	50	120	13
1	90	120	13
1	100	120	13
2	658	200	19
2	850	200	19
2	11029.5	see notes	see notes

Table 7.8 Lock times for synthesisers

All lock times were measured with the Hewlett-Packard storage oscilloscope and analyser connected together on the HPIB / IEEE488 bus. The higher frequency output would not lock directly and required a manual input to force the VCO to approximate the output frequency required, and then a re-load of lock codes. The signals were captured on the pins of the synthesiser and stored on the oscilloscope. The stored information could then be printed directly on the HPIB printer, or stored on the PC as a Gerber file.

The Hewlett-Packard spectrum analyser also had a HPIB output. This was used to connect to a special high resolution printer, which was specially designed for use with the spectrum analyser. These files were too large too be input directly, so the spectra were printed and then scanned. This meant that some of the fine resolution was lost.

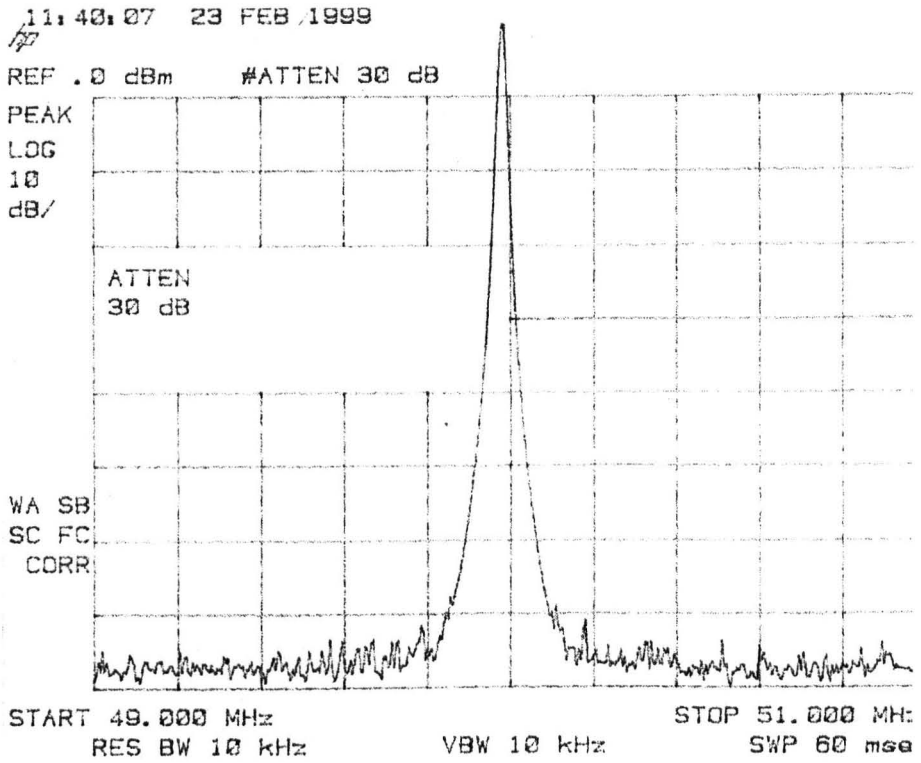


Figure 7.2 Synthesiser 1 spectrum, minimum output frequency

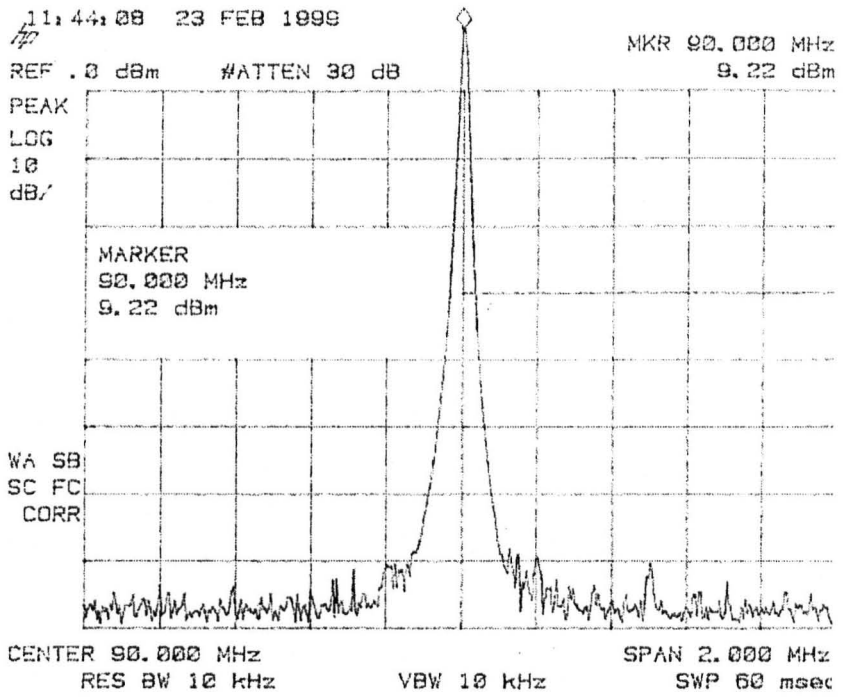


Figure 7.3 Synthesiser 1 spectrum, operating output frequency

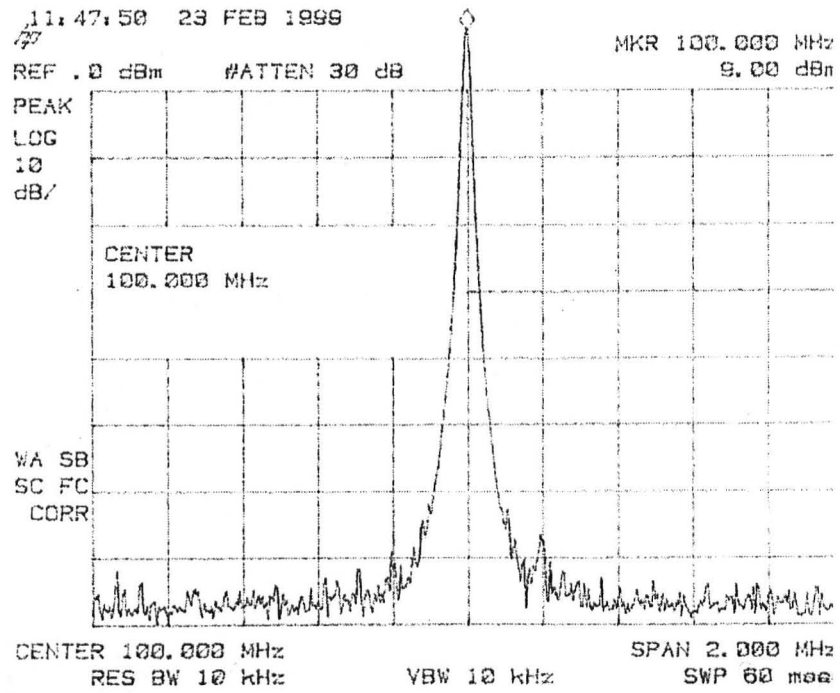


Figure 7.4 Synthesiser 1 spectrum, maximum output frequency

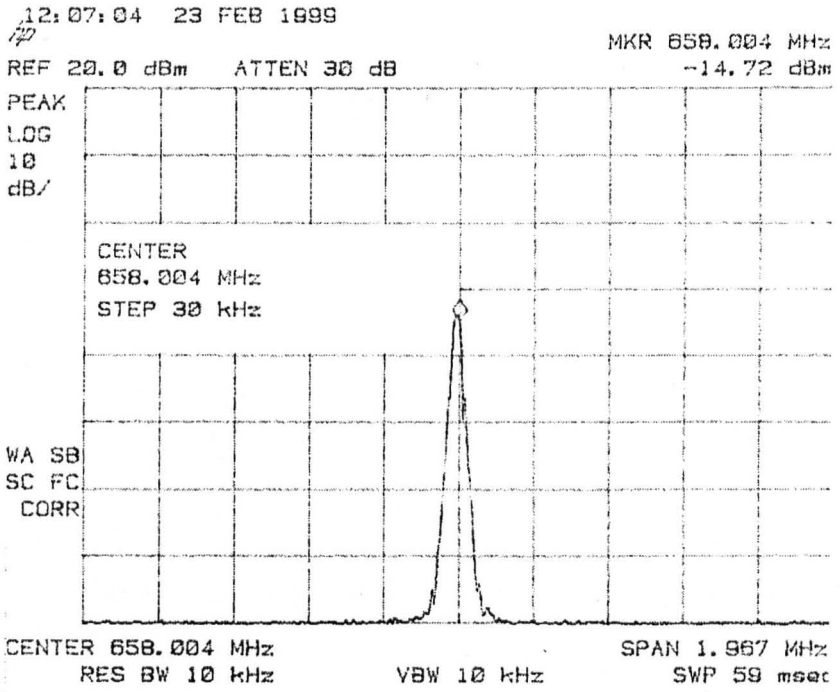


Figure 7.5 Synthesiser 2 spectrum, minimum output frequency

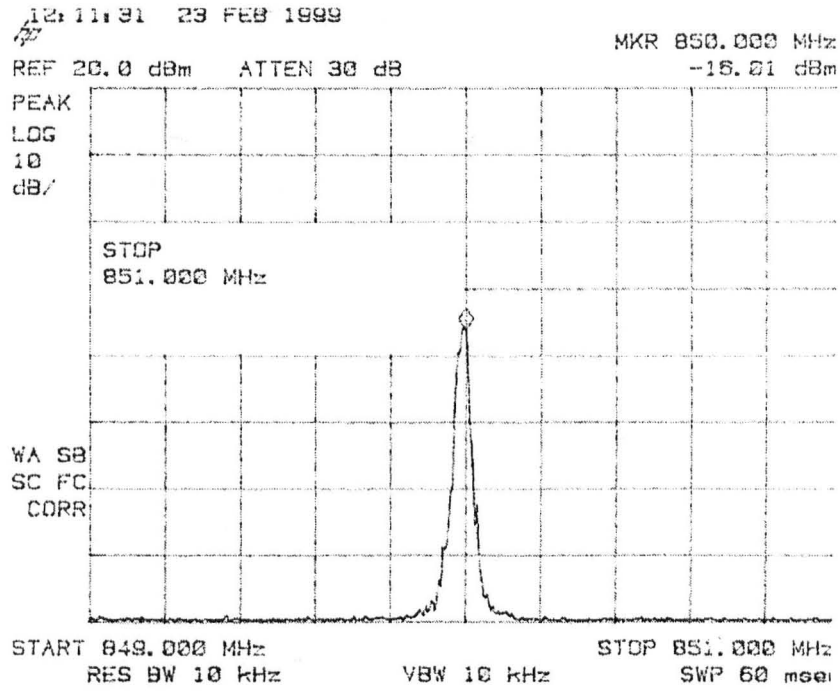


Figure 7.6 Synthesiser 2 spectrum, operating output frequency

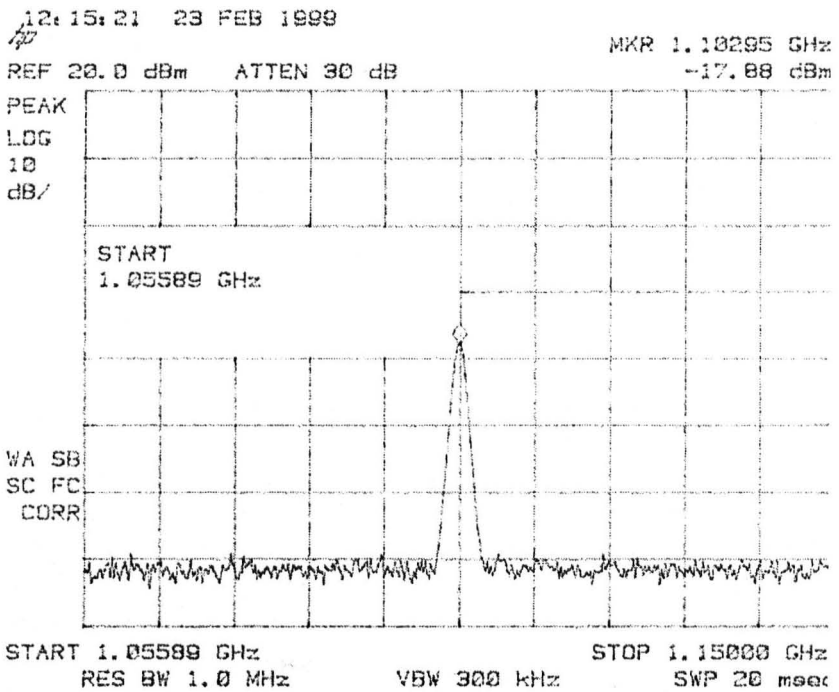


Figure 7.7 Synthesiser 2 spectrum, maximum output frequency

7.3 The transmitters

Firstly, the transmitters were set up individually and connected via the attenuator/ phasor array to the spectrum analyser. Figure 7.8 illustrates the equipment layout and Figures 7.9 and 7.10 illustrate the radio frequency output of transmitters 1 and 2 respectively. The data input was generated by the EPROM board, which produced a continuous test pattern; this is shown in Figures 7.11 and Figure 7.12.

The circuitry was then connected together with both transmitter outputs connected via the attenuators/phase-shifters to the input of the receiver. The frequency-translation stages were disconnected (using a bypass) so the output frequencies were identical (45 MHz). Test patterns were input to both transmitters and the resulting combined spectrum is shown in Figure 7.13. The EPROM test pattern generators were then switched over to the Delta modulators and the coders were switched to 'linear independent test set 1' output, i.e. unique codes for each transmitter. A test tone was input to the Delta modulators at 1.5 KHz. Figure 7.14 shows the audio input, the Delta modulator output and the coder output. The attenuators were adjusted such that each transmitter contributed a similar signal amplitude and the combined output spectrum is reproduced in Figure 7.15.

The coders were then switched over to output 'test set 2', i.e. a partially shared codeset, and Figure 7.16 shows the audio input, the Delta modulator output and the combined spectrum output for the system. The slight increase in bandwidth with the non-linear coders switched in may be noted. The bandwidth is large anyway due to the filters used in the type 1 system. Also, the larger 'ripples' on the slopes of the spectrum peak, may be observed, caused by the modulation applied, and the differences in output amplitude between the transmitters.

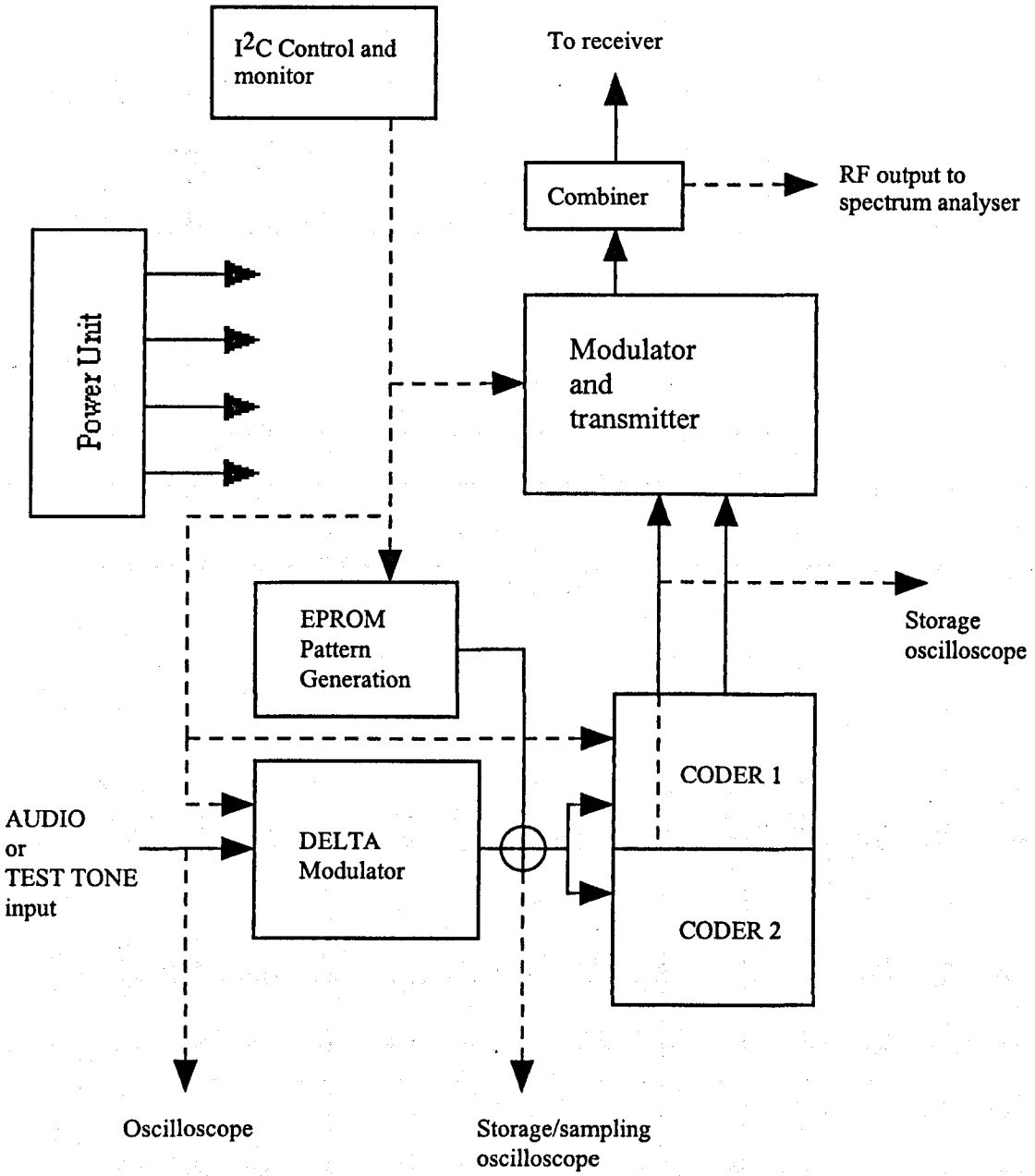


Figure 7.8 **Test equipment layout (transmitter)**
Blocks indicate PCB or breadboard
Duplicated for second transmitter

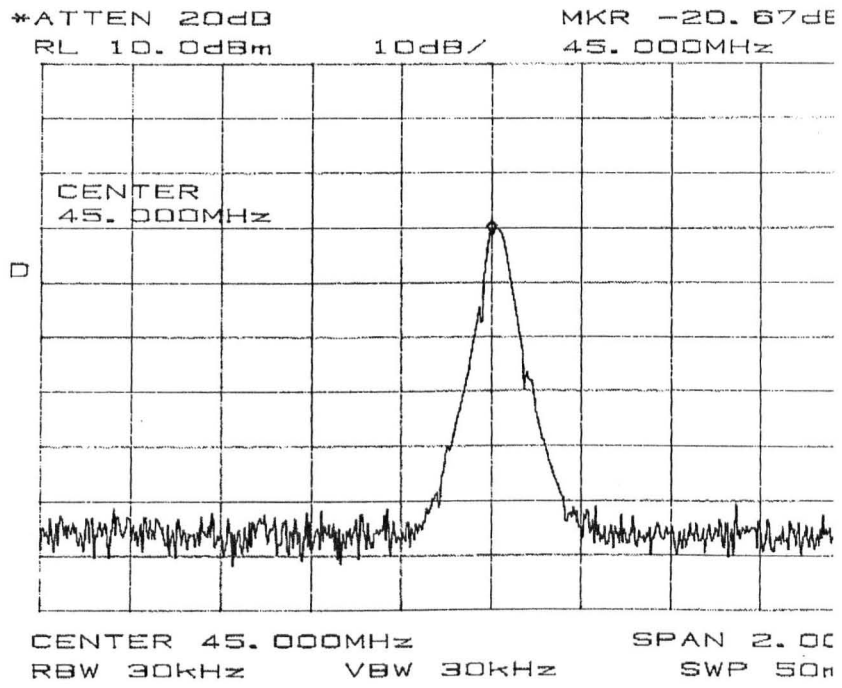


Figure 7.9 Transmitter 1 output spectrum (test pattern 1 input)

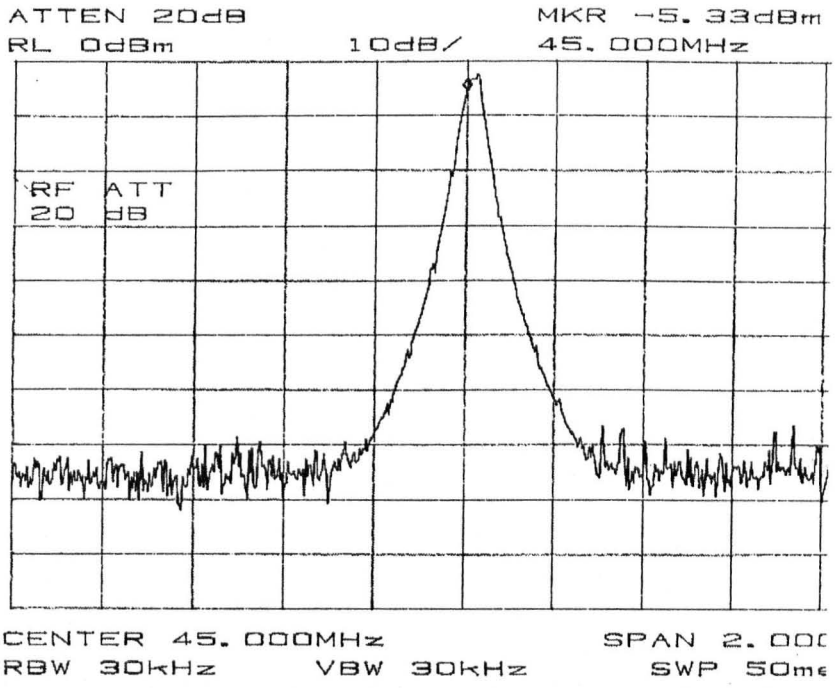


Figure 7.10 Transmitter 2 output spectrum (test pattern 1 input)

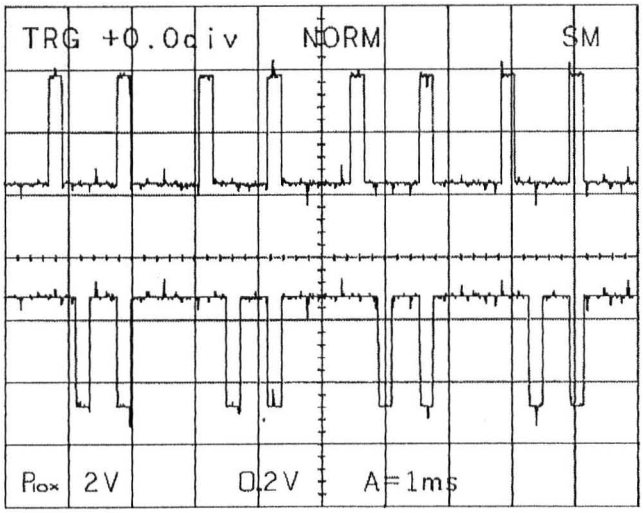


Figure 7.11 Transmitter 1: input test pattern

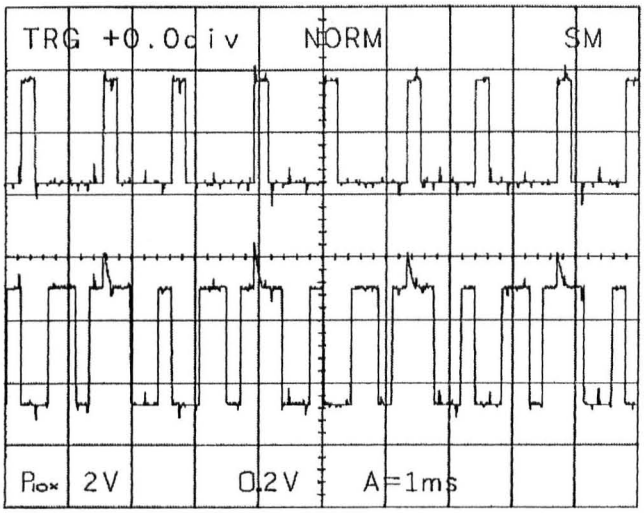


Figure 7.12 Transmitter 2: input test pattern

N.B. Figures 7.11 and 7.12 were captured with a Hewlett Packard storage oscilloscope. The top trace represents the in-phase input to the modulator and the lower trace the quadrature-phase input to the same modulator. The spikes are problems with decoupling and were removed later with extra decoupling capacitors and new earth rail.

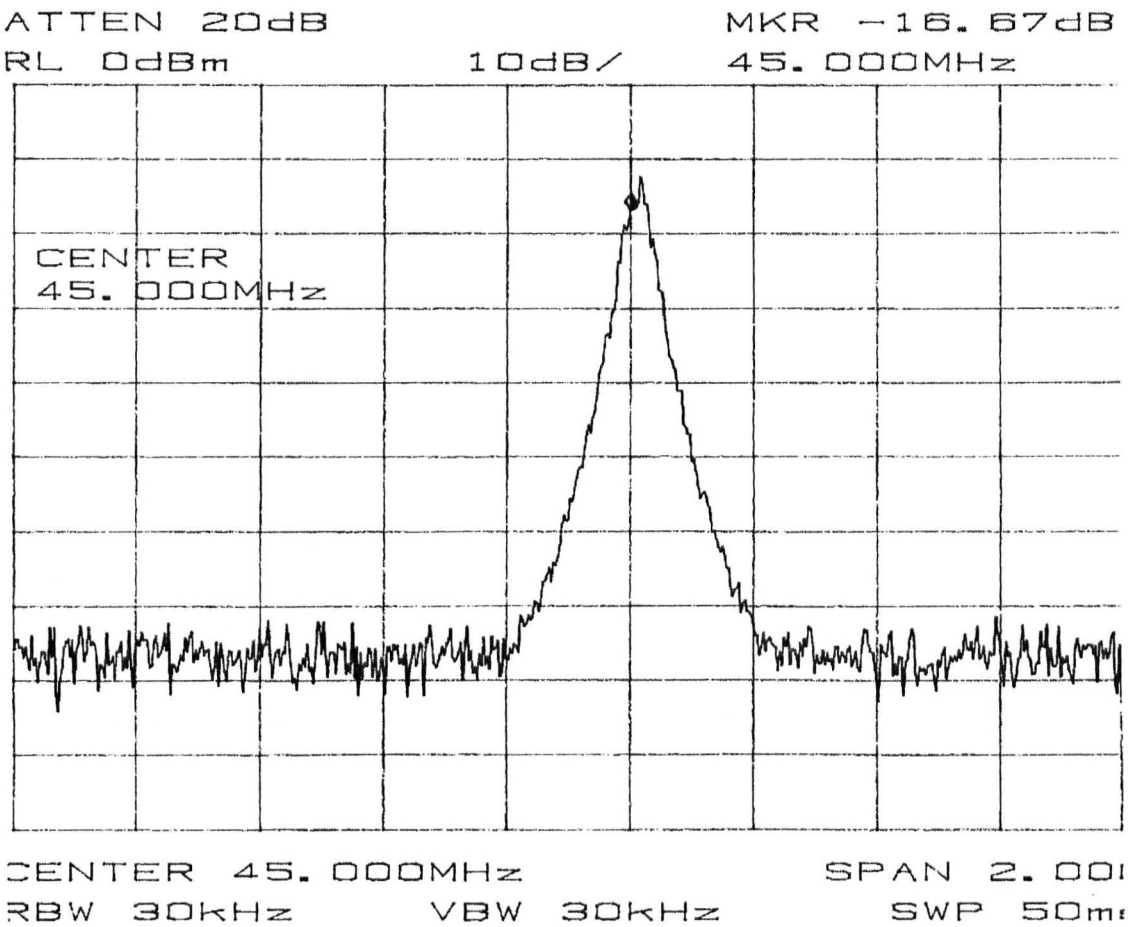


Figure 7.13 Combined output spectrum (test pattern inputs)

Figure 7.13 shows the combined output spectrum, with the two transmitters connected together and the radio frequency output amplitude levels adjusted via the attenuators to give similar output levels. The code patterns input are illustrated in the previous figures. The increase in modulation causes the curious effect whereby the crest of the peak 'wobbles' at a frequency related to a beat frequency between the two transmitter output frequencies.

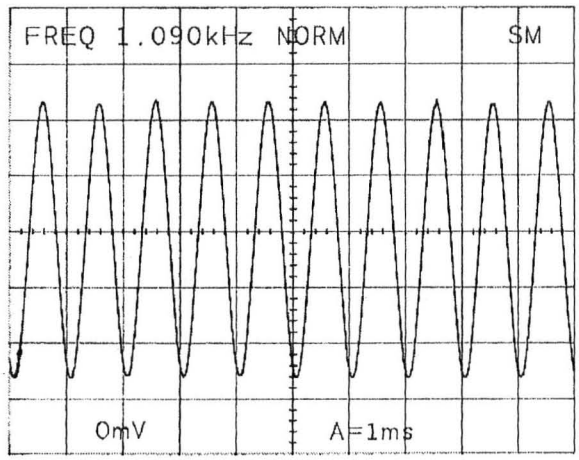


Figure 7.14a Audio test tone input

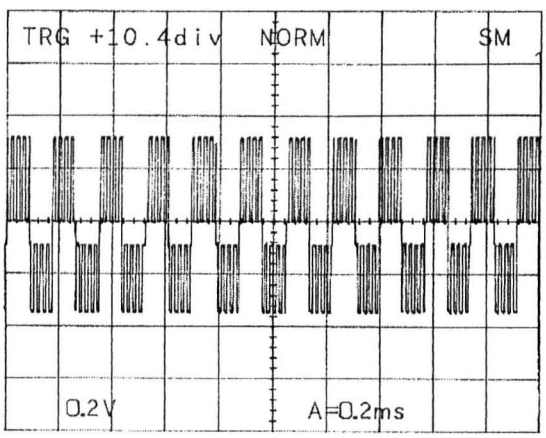


Figure 7.14b Deltamodulator output (overdriven for clarity)

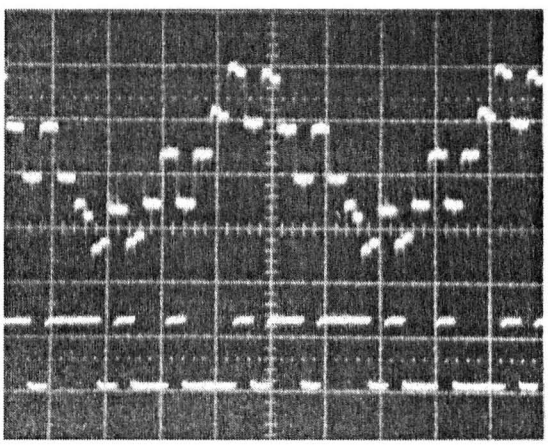


Figure 7.14c Recovered Deltamod (top) Transmitted Deltamod (lower)

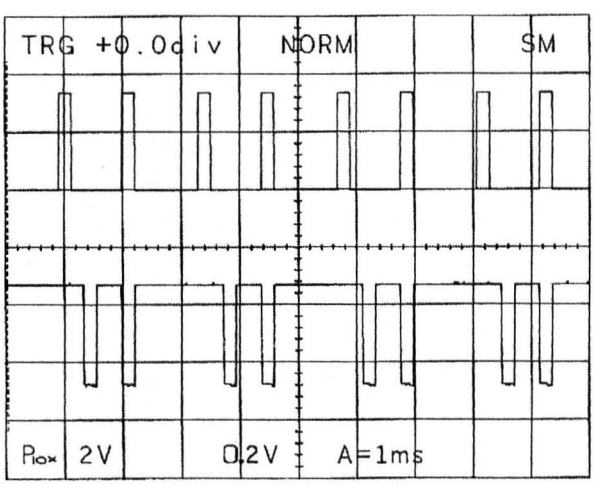


Figure 7.14d Coder output

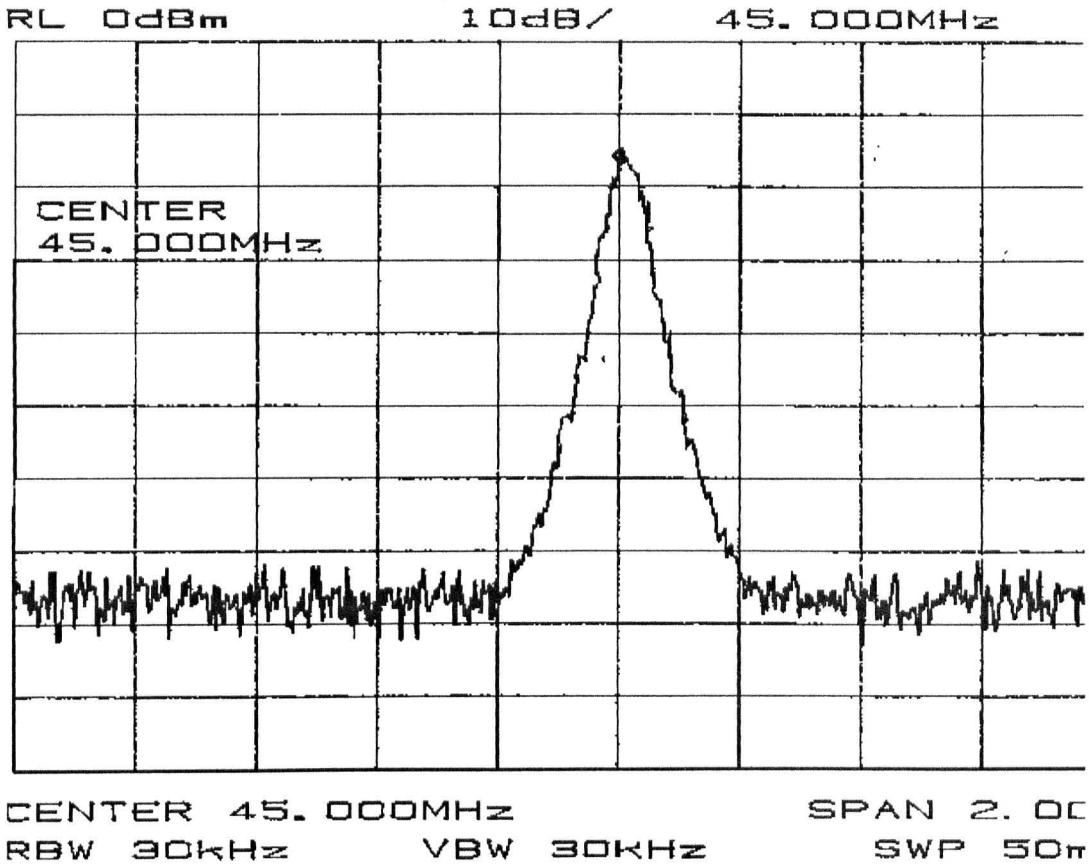


Figure 7.15 Combined output spectrum with recorded audio inputs and code-word set 1 (unique set for each transmitter)

Figure 7.15 shows the spectrum of the combined signal produced by transmitters 1 and 2 connected together through the attenuator array. The signal appears to spread more than with the constant test tone inputs, and the modulation peaks are even less pronounced than before due to the variability of the modulating signal.

Note that all results described so far are taken with the type 1 synchronisation system (central synchronisation). In order to make the system operational, as quickly as possible, the various system filters were not connected. The later type 3 system incorporated the filters (pulse shaping and intermodulation product reduction) and these results are presented in section 7.5.

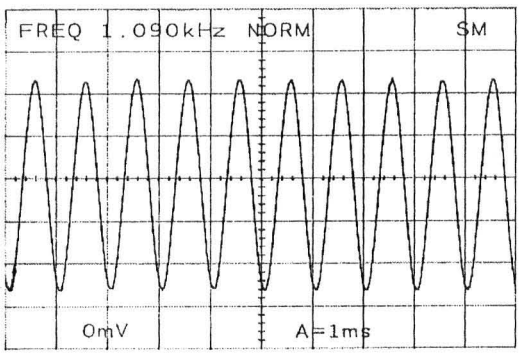


Figure 7.16a Audio input test tone

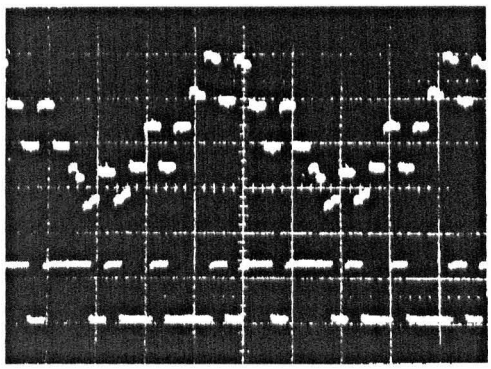


Figure 7.16b Deltamodulator output

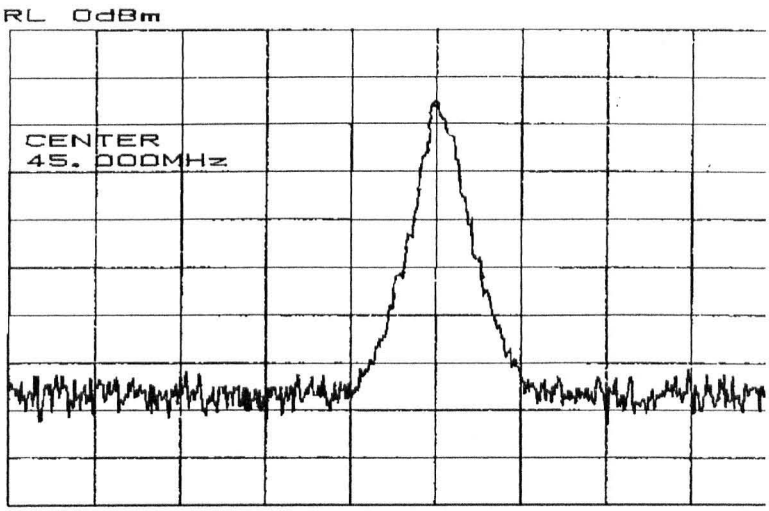


Figure 7.16c Combined spectrum (input test set 2)

The results obtained with the coders set to generate test set 2 codes are similar to the results obtained with the coders set to generate test set 1 codes. The spectrum was found to fluctuate, hence, when the spectrum is captured and stored this additional information is lost. The 'wobble' observed at the crest of the spectral peak is caused by the two frequencies being very nearly identical, and producing the beat frequency of the 'wobble'. This beat, and other effects, were related to the codes used in the transmitters and shows up as a variation in the spectrum slope, near to the peak. This effect was more pronounced when using the second set of codewords.

7.4 The receiver

Initially, the complete system was set up by connecting the individual transmitters to the receiver via an attenuator/phase shift array. A test sequence was input to the coder board from the EPROM board and the modulated and amplified transmitter output was monitored on the spectrum analyser (refer to section 7.3 for diagrams of the spectra and data signals of both transmitters and the combined output). This signal is then used as the input signal for the receiver and is fed directly to the demodulator, which was also synchronised to the main synthesiser. The I^2C bus is still required to be connected to the frequency conversion circuits, so that the internal bypass can be initiated by outputting the correct data string to the frequency conversion circuits. Refer to Appendix 15 for details.

The de-modulated outputs from the Maxim integrated circuit (in-phase and quadrature) were monitored on a standard oscilloscope which was triggered from the divided main synthesiser output. Once set up, the demodulated signals were fed into a storage oscilloscope and the output printed as a record of the signals present. The pulse-shaping filters were removed from the initial type 1 set-up. The outputs were then buffered and fed to the parallel inputs of the Burr-Brown precision analogue-to-digital converter. In order to prevent a two's complement complication the inputs were restricted to the positive input range only, i.e. a maximum of +2.65 V. The timing of the analogue-to-digital conversion is critical to the operation of the CV-CCMA system. The convert pulse is derived from the main synthesiser via divide chain 2, refer to Appendix 16 for a full set of timing diagrams. The converted digital output is fed to the serial port of the TMS320C31 signal processor. A timing pulse from the EPROM pattern-generation board was used to start the signal processor program by triggering an external interrupt line (EINT3). Once running, the program

generates the timing signals necessary to recover the samples from the analogue-to-digital converter. In the initial test set-up it was found that a buffer was required between the two transmitter outputs and the receiver. The outputs/inputs were connected together via the attenuator/phase shifter and the combined signal was input directly into the Maxim modulator/demodulator integrated circuit. The frequency-translation stages were disconnected in order to simplify any diagnostics required and the 45 MHz signal was monitored on the spectrum analyser and the sampling oscilloscope. The sampling scope produced the inverted figures (more black than white). The first implementation involved restricting the inputs to the modulator so that the system operated in a form of BPSK, that is each bit output is a single phase state.

Figure 7.17 shows the receiver equipment layout. A copy of the output screen of the PC connected to the TMS320C31 is shown in Figure 7.18. The top screen is a scan of the PC screen during program load and the lower screen is a scan of the PC screen with the program running, this shows the individual recovered test pattern data. Figures 7.19 and 7.20 show the recovered data for the combined test pattern data (i.e. two transmitters and one receiver recovering the two superimposed signals), for BPSK and QPSK respectively.

Figures 7.21 and 7.22 are scans of photographs of the equipment rooms at the two sites, University of Warwick and Cable and Wireless PLC. Some of the difficulties encountered whilst making the system work, particularily with full QPSK modulation, are discussed in Chapter 8.

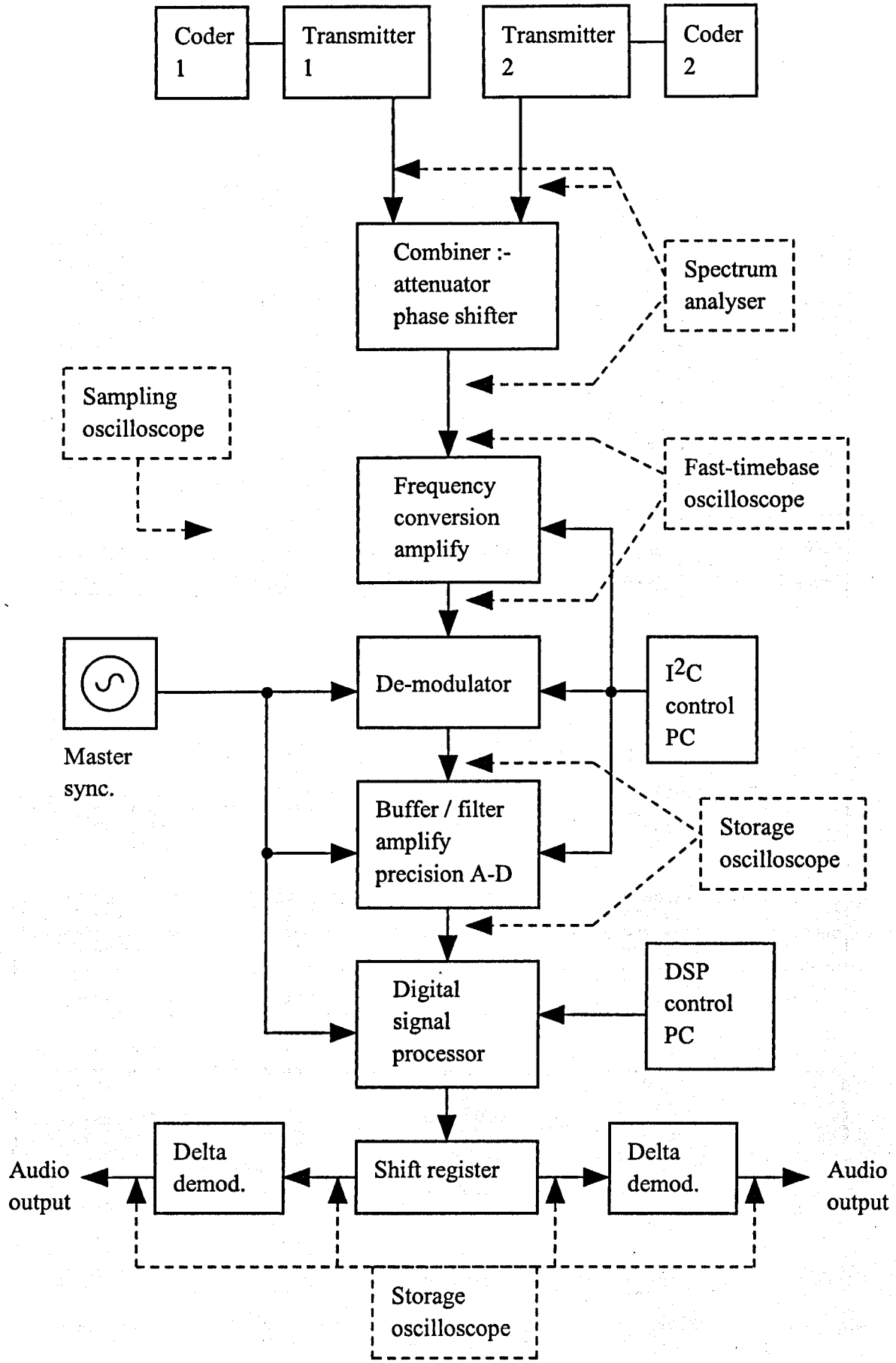
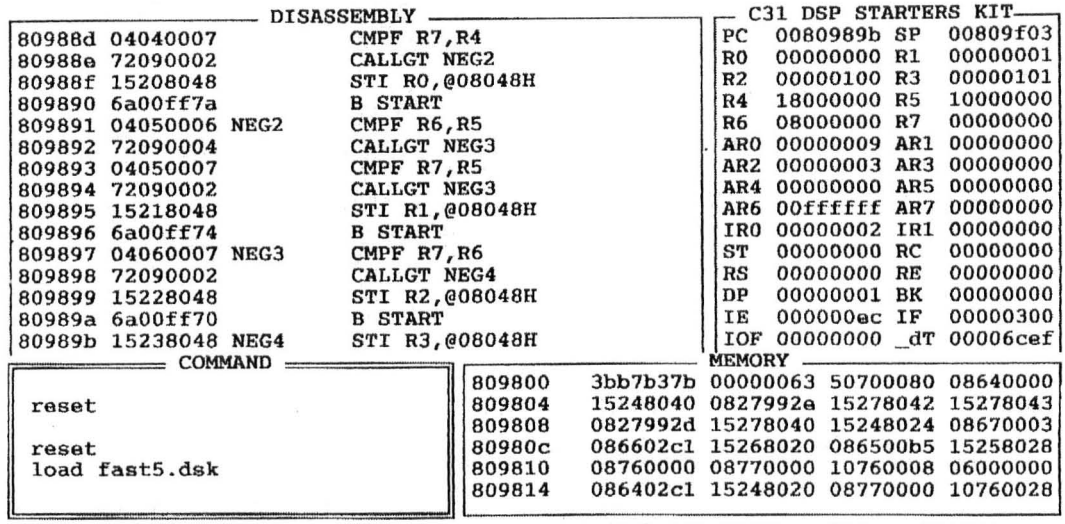


Figure 7.17 Receiver test equipment layout (type 1)



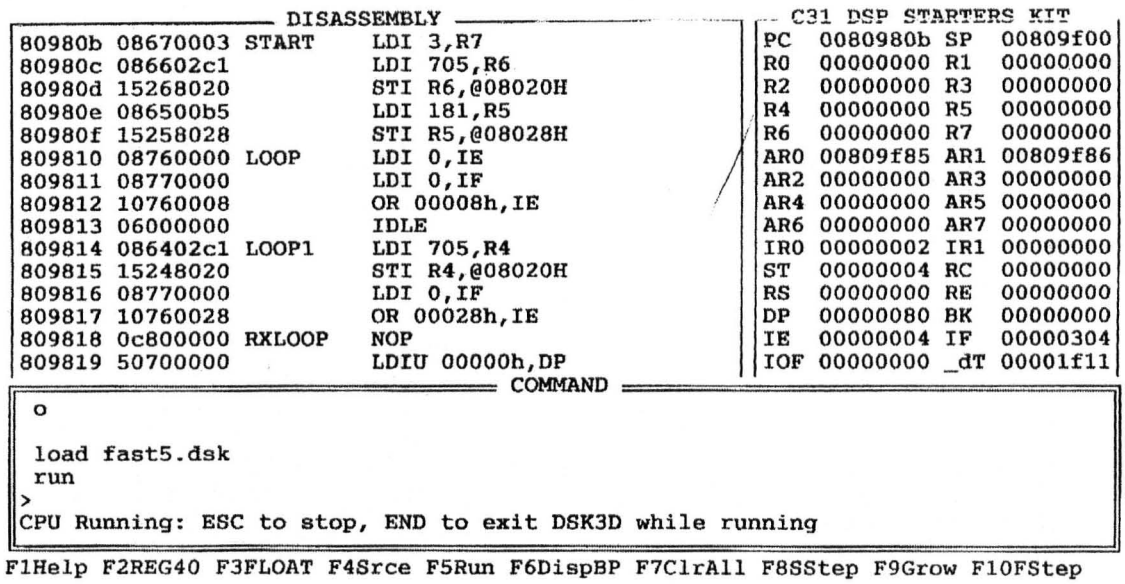


Figure 7.18 PC screen showing disassembly of fast5.asm

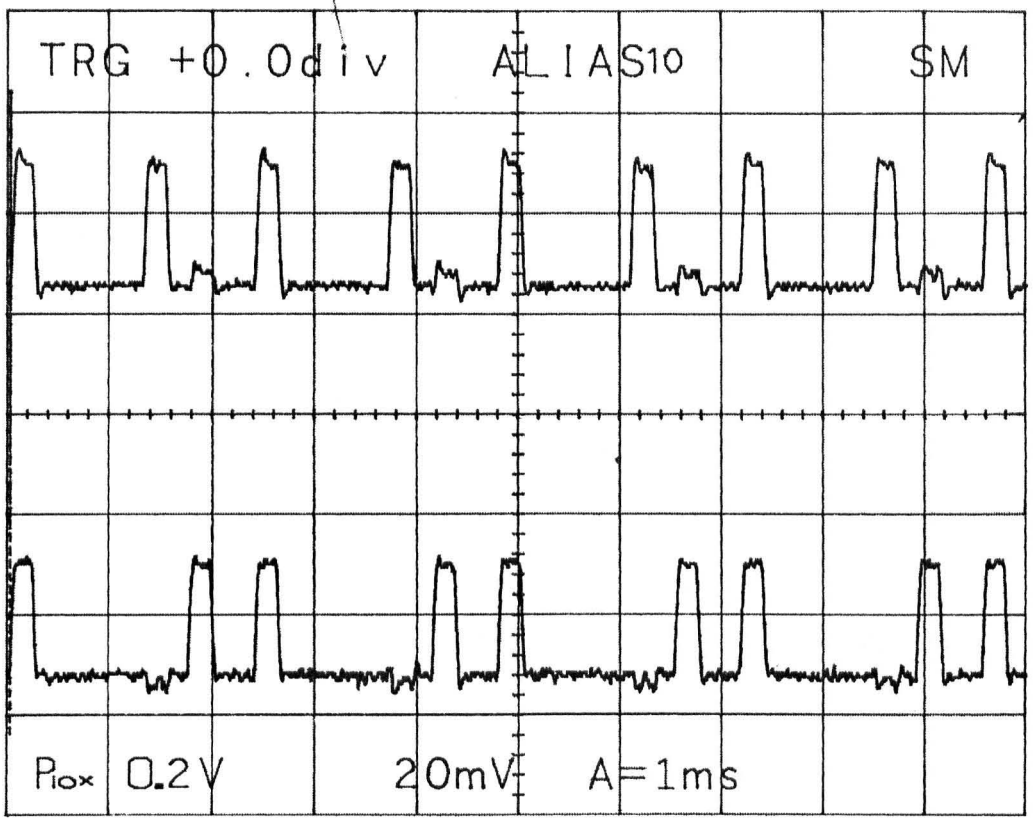


Figure 7.19 Recovered test pattern data (BPSK)

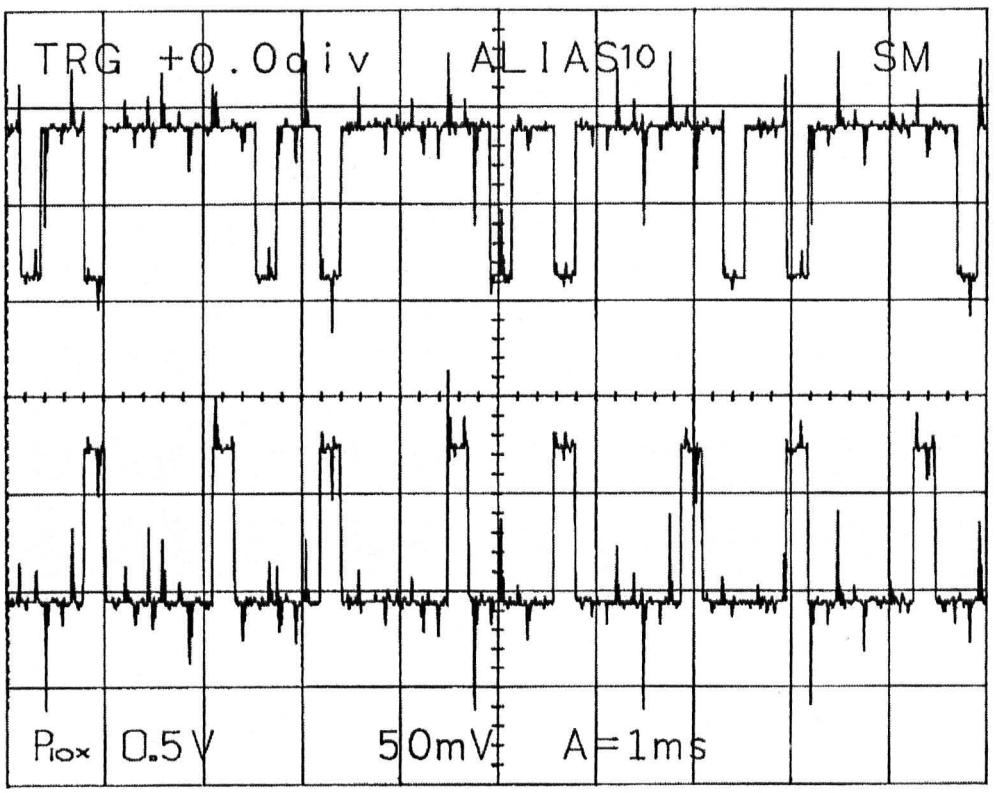
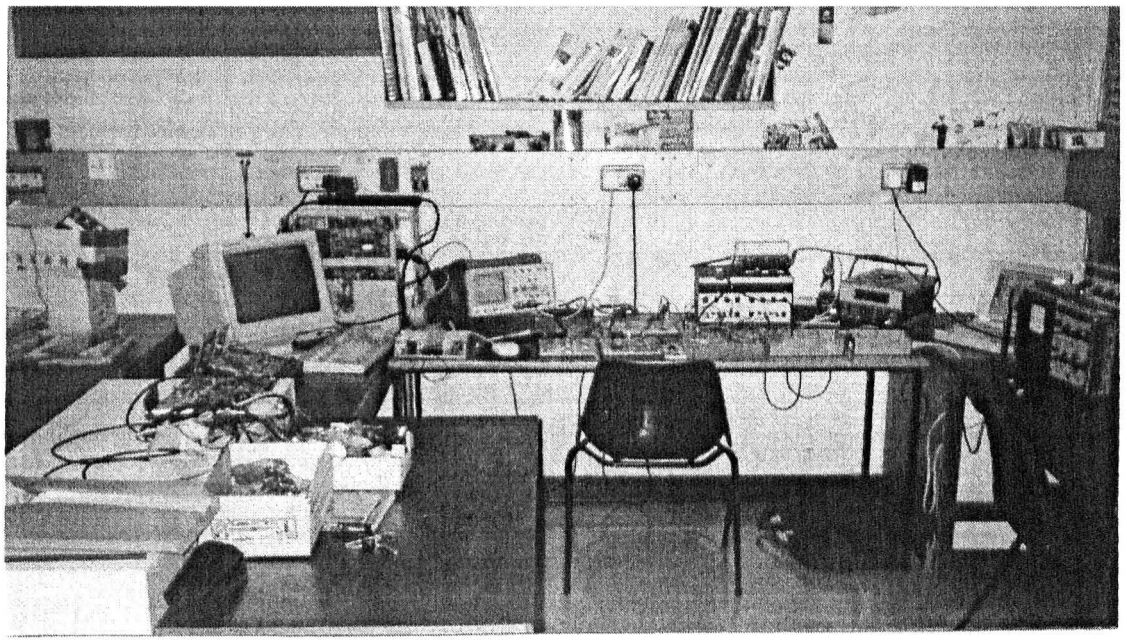


Figure 7.20 Recovered test pattern data (QPSK)



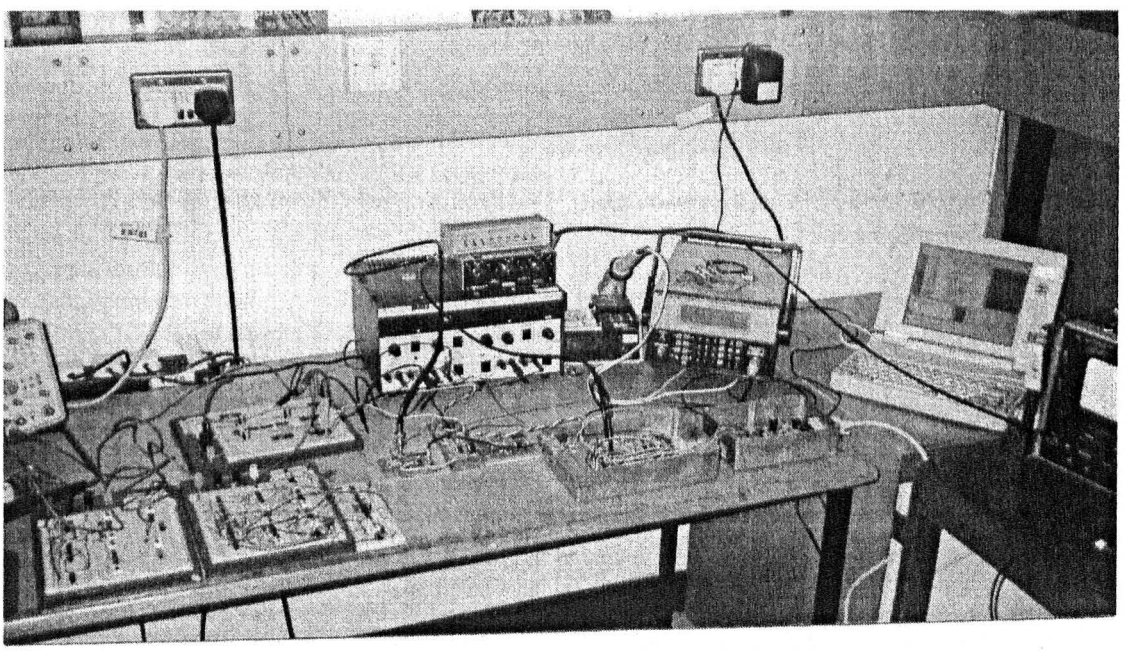


Figure 7.21 Equipment set-up (University of Warwick)

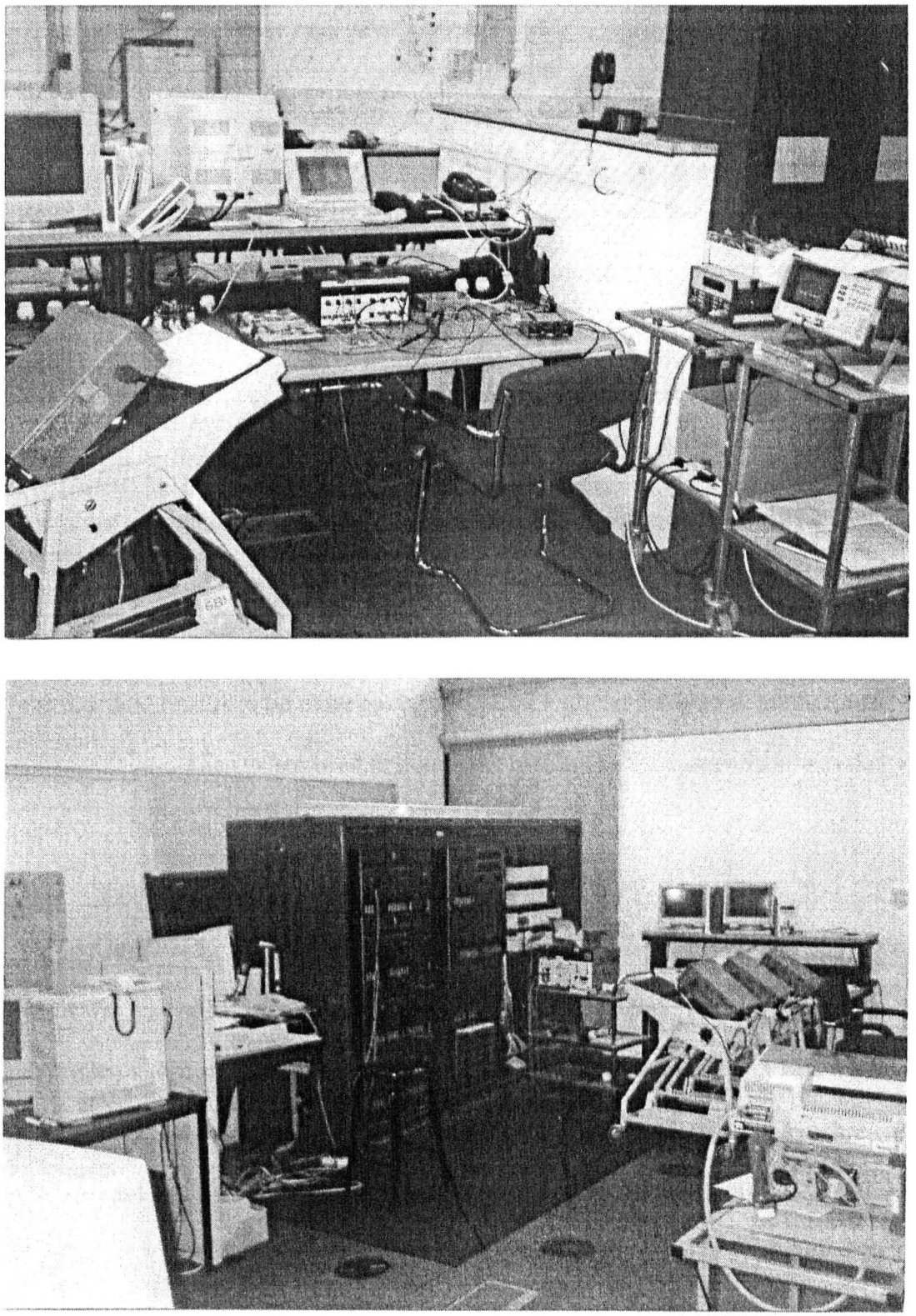


Figure 7.22 Equipment set-up (Cable and Wireless PLC, Coventry)

7.5 Bit error rate performance

In order to reduce the problems caused by electrical noise, (generated by the switching equipment at Cable and Wireless PLC), the CV-CCMA system was re-assembled at the University of Warwick. After allowing a few days for the system to stabilise, the bit error rate performance was determined.

This was to compare practical results with the simulations produced by Ali, Chandler and Soysa in [74]. They plot information bit error-rate against signal energy per bit for various average fading powers. The simulations involve 2 users and therefore can be directly compared with the actual results obtained with 2 transmitters and 1 receiver. The simulation results are reproduced in Figure 7.23, which is an electronic scan of the original paper. The results are redrawn in Figure 7.24, to enable the practical results to be more easily compared with the theoretical predictions.

The system used to obtain the results for the comparison was the type 3 system. This was developed as a more practicable CV-CCMA system and involved synchronisation recovery via a separate 10 MHz channel. The system also contained the pulse-shaping filters and the transmitter band-pass filters. The multi-sample software was also implemented on the type 3 system. This restricted the data throughput to a maximum data rate of 1.2 Kb/s per channel, in the two transmitter, one receiver set-up. However, it gives the system a better tolerance to synchronisation impairments. Sub-section 6.6 contains some results which verify this.

The simulations comprise 6 sets of results, 3 sets of results with space diversity (7-12) and 3 sets without (1-6). The CV-CCMA demonstrator did not include any provision for diversity, so a comparison between the sets of results could not be done. Intuitively, with the additional 'information' provided by space diversity, the BER performance should improve, (over no diversity), as shown in Fig.2 of [74].

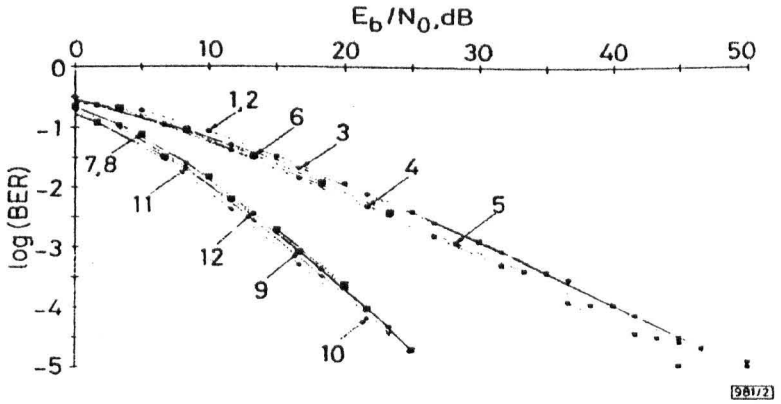


Figure 7.23 Simulation results from [74]

Plots 1 and 2 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 1
Plots 3 and 4 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 0.1
Plots 5 and 6 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 0.00001

Plots 7 and 8 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 1
Plots 9 and 10 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 0.1
Plots 11 and 12 show users 1 and 2, average fading powers: user 1 = 1, user 2 = 0.00001

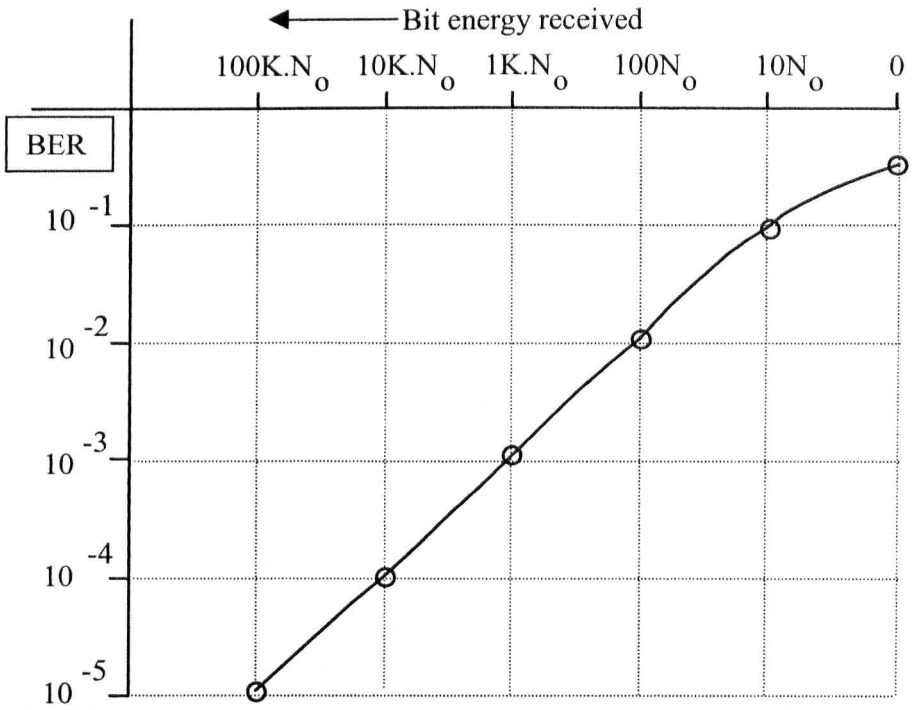


Figure 7.24 Simulation (1) from 7.23 redrawn for comparison purposes

N.B. Figures 7.24, 7.26, 7.27 and 7.28 are plotted with *Macdraw Pro-plot* with least squares best-fit algorithm.

The equipment test configuration comprised two transmitters, one of which was designated the master node (transmitter 1), and a single receiver. System synchronisation was set-up so that transmitter 2, and the receiver, were slaved to the master transmitter's synthesiser. This configuration effectively locked the three TCXOs together via the radio link. Trend data testers (x4) were used and connected to give two simplex test links. Trend data testers were selected for the testing because they are simple to set-up, have a direct read out of average BER, and have timers built-in which can be used for automatic testing. The testers connected to the transmitters were set-up as below:-

Rate :	1200	DTE
Code used :	511P/Hex8	FDX
Sync.:	SYN1	NoP
Bit errors :	NI	

Selection of the 'DTE' (Data terminal equipment) setting means transmit data is sourced from pin 2 on the 25-way D-type connector, 'Rate' is transmit data rate, 1.2 Kb/s. 'Code used = (511P)' is a standard test pattern and 'FDX' means full-duplex operation. The 'sync.' selection of 'SYN1' means synchronisation was derived from pins 15 and 24 on the 25-way D-type and 'NoP' means no parity testing. Finally, 'Bit errors = NI' means no injection of errors. The tester at the other end of the link was set-up in the same way, the only difference is that synchronisation was set to 'Recovered'. The transmit testers were timed from a feed derived from the synthesiser, the receive testers needed to be set to recovered timing as the DSP feeding the buffers was not slaved to the master synchronisation source. The testers incorporated one test pattern length of data in a buffer, so some tolerance was built-in. The voltage levels on the output of the tester are RS232 (recommended standard 232,

[121]), these were level-shifted, for correct matching, using a simple resistor network. The level was lowered to match the input of the CV-CCMA and increased to match the tester input at the CV-CCMA output.

The tests were performed on the links (link 1 is tester 1 transmit to tester 2 receive and link 2 is tester 3 transmit to tester 4 receive) by varying the signal strengths of the signals at the input to the combiner. Assuming that the noise component remains constant for the duration of the testing, this will provide an effective variation of signal/noise ratio, which can be used to compare with the simulation results from [74]. The system was configured with hardwire interconnection, via the combiner array, of the two transmitter outputs and the receiver input. This was due to the variability of the noise background at the University. The room used is adjacent to the lift motor room, and electrical interference was experienced when the motor was operating. The screening effect of the hardwire interconnection reduced these noise effects. Also, for each signal level, the results were taken over a minimum time period of one hour, to average out these effects.

The phase component of the g value was not altered during the testing. The levels were measured using a true RMS radio frequency voltmeter and a calibrated active probe connected to the Hewlett-Packard spectrum analyzer. At each different input level, tests were performed for 1-hour duration to obtain an average BER value from the data testers. Three tests were performed:

i) Link 1 fixed, link 2 varied

ii) Link 2 fixed, link 1 varied

iii) Link 1 varied, link 2 varied

All radio frequency levels were measured using the active probe with direct read out from the Hewlett-Packard spectrum analyzer. The RF voltmeter was used as back up to give confidence in the level readings. The Trend data testers that were used give an average BER over the time period of the test, these are the values reproduced in Tables 7.9, 7.10 and 7.11.

Figure 7.25 illustrates the test equipment set-up used to obtain the BER test results. The data is plotted in Figures 7.26, 7.27 and 7.28, these figures can be used to compare with the simulation results, which are shown earlier in Figure 7.23 and Figure 7.24.

Test no.	RF level at attenuator 1 output (dBm)	RF level at attenuator 2 output (dBm)	Average BER (tester 1 – 2)	Average BER (tester 3 – 4)
i	–20.0	–10.0	$<10^{-9}$	$<10^{-9}$
i	–20.0	–20.0	$<10^{-9}$	$<10^{-9}$
i	–20.0	–30.0	$<10^{-9}$	$\leqq 10^{-8}$
i	–20.0	–40.0	$<10^{-9}$	$\leqq 10^{-7}$
i	–20.0	–50.0	$<10^{-9}$	$\leqq 10^{-6}$
i	–20.0	–60.0	$<10^{-9}$	$\leqq 10^{-6}$
i	–20.0	–70.0	$<10^{-9}$	$\leqq 10^{-5}$
i	–20.0	–80.0	$<10^{-9}$	$>10^{-3}$
i	–20.0	–90.0	$<10^{-9}$	$>10^{-3}$
i	–20.0	–100.0	$<10^{-9}$	$>10^{-3}$

Table 7.9 BER results, link 1 fixed with link 2 varied

Test no.	RF level at attenuator 1 output (dBm)	RF level at attenuator 2 output (dBm)	Average BER (tester 1 – 2)	Average BER (tester 3 – 4)
ii	–10.0	–20.0	low power	$<10^{-9}$
ii	–20.0	–20.0	$<10^{-9}$	$<10^{-9}$
ii	–30.0	–20.0	$<10^{-9}$	$<10^{-9}$
ii	–40.0	–20.0	$\leqq 10^{-8}$	$<10^{-9}$
ii	–50.0	–20.0	$\leqq 10^{-7}$	$<10^{-9}$
ii	–60.0	–20.0	$\leqq 10^{-5}$	$<10^{-9}$
ii	–70.0	–20.0	$\leqq 10^{-4}$	$<10^{-9}$
ii	–80.0	–20.0	$>10^{-3}$	$<10^{-9}$
ii	–90.0	–20.0	$>10^{-3}$	$<10^{-9}$
ii	–100.0	–20.0	$>10^{-3}$	$<10^{-9}$

Table 7.10 BER results, link 2 fixed with link 1 varied

Test no.	RF level at attenuator 1 output (dBm)	RF level at attenuator 2 output (dBm)	Average BER (tester 1 – 2)	Average BER (tester 3 – 4)
iii	–20.0	–20.0	$<10^{-9}$	$<10^{-9}$
iii	–30.0	–30.0	$<10^{-9}$	$<10^{-9}$
iii	–40.0	–40.0	$<10^{-9}$	$<10^{-8}$
iii	–50.0	–50.0	$\leqq 10^{-8}$	$\leqq 10^{-8}$
iii	–60.0	–60.0	$\leqq 10^{-6}$	$\leqq 10^{-6}$
iii	–70.0	–70.0	$\leqq 10^{-6}$	$\leqq 10^{-5}$
iii	–80.0	–80.0	$>10^{-3}$	$>10^{-3}$
iii	–90.0	–90.0	$>10^{-3}$	$>10^{-3}$
iii	–100.0	–100.0	$>10^{-3}$	$>10^{-3}$
iii	–110.0	–110.0	$>10^{-3}$	$>10^{-3}$

Table 7.11 BER results, link 1 varied and link 2 varied

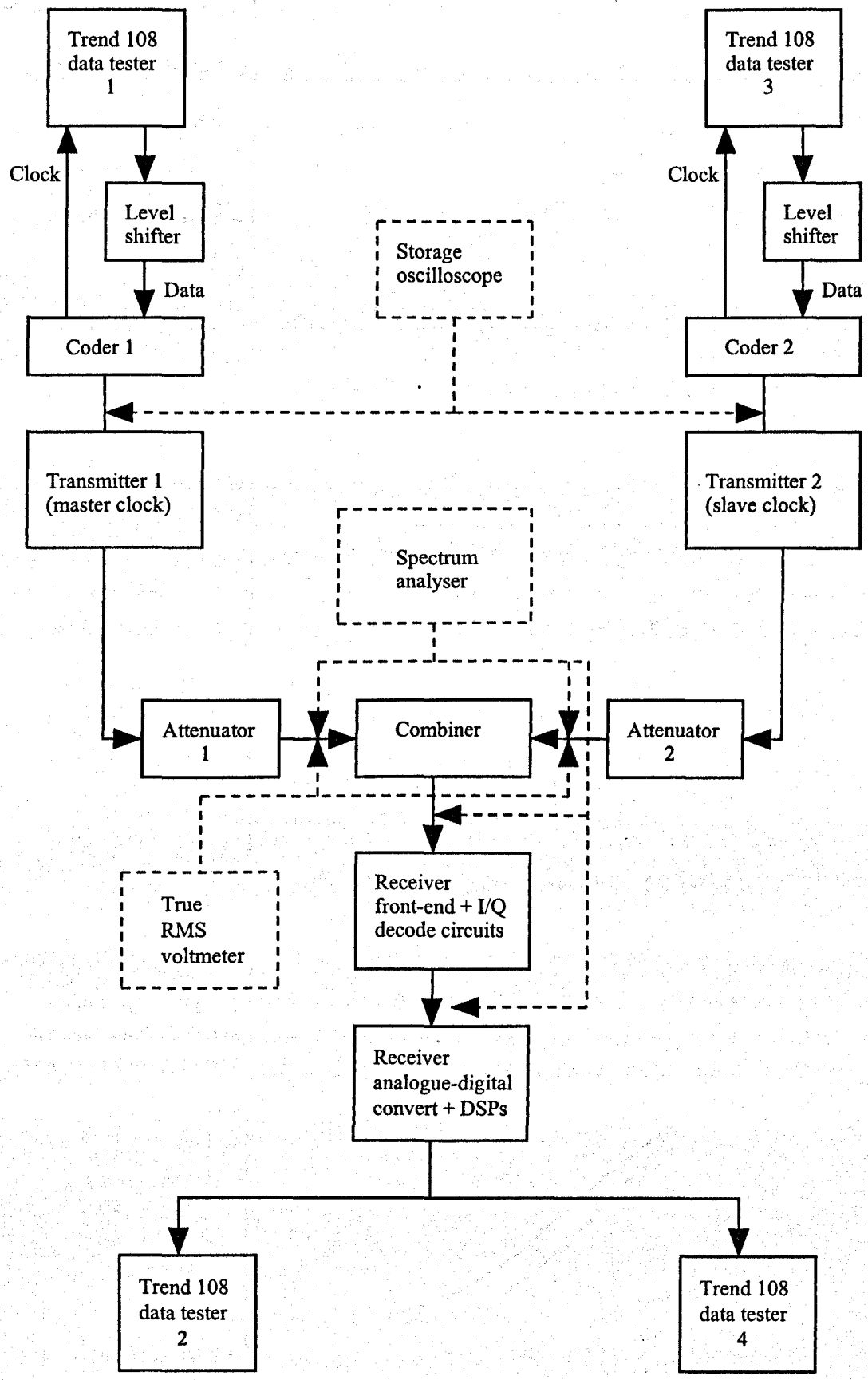


Figure 7.25 Test equipment layout (BER testing)

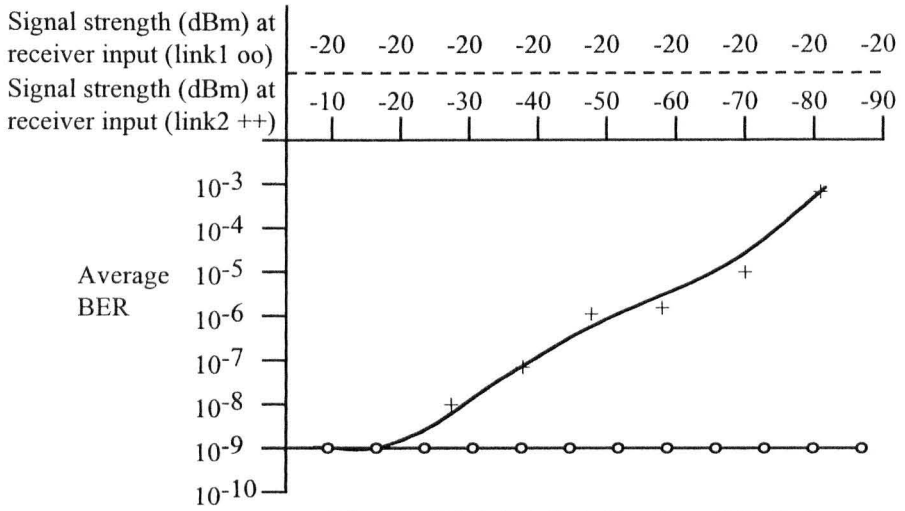


Figure 7.26 Link 1 fixed and link 2 varied

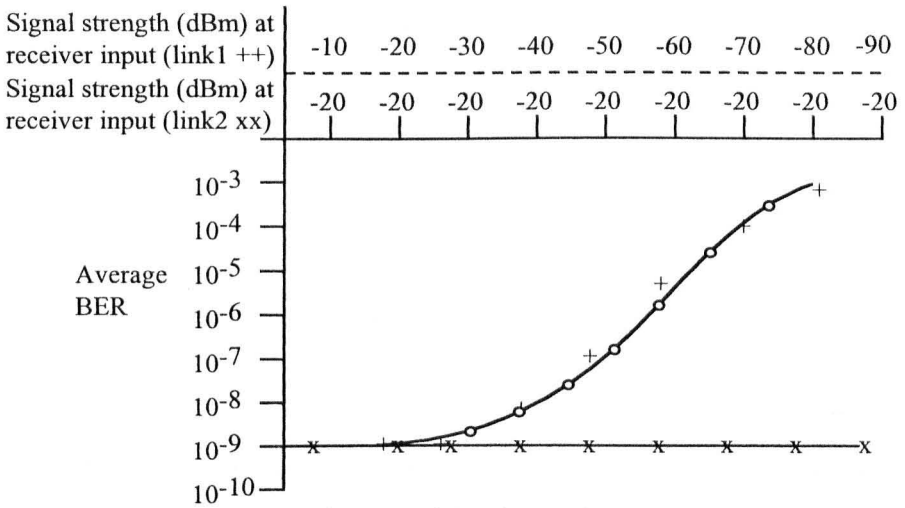


Figure 7.27 Link 2 fixed and link 1 varied

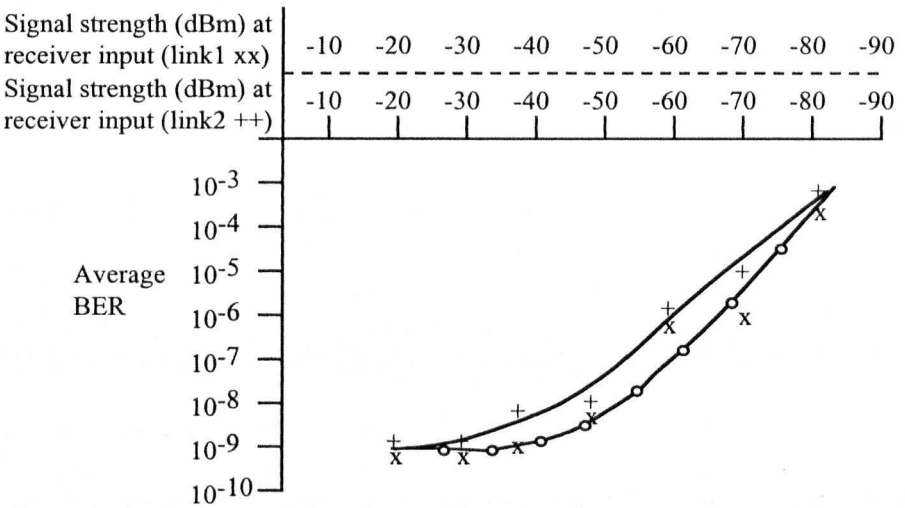


Figure 7.28 Link 1 varied and link 2 varied

N.B. Link 1 o—o—o—o , Link 2 ——— , plotted as least square fit.

When interpreting Figures 6.26 – 6.27 it is important to remember that the Trend data testers report an average BER. Any BER level worse than 10^{-3}, or 1 error bit in 1000, is treated as a failing link and reported as such. Any BER better than 10^{-9} is treated as a satisfactory link. These BER values set the limits on the tests performed and restrict the range of values for the comparison. Between these limits the testers report the average BER over the period of the test, reporting BER to the nearest single bit error in 1000 bits received.

The results clearly show that the 2 links are operating independently, and varying signal strengths on the first channel link (within the limits shown) does not affect the operation of the second channel. This confirms the simulation results. However, if the level was increased above –10dBm then the demodulator would saturate and the links would both fail.

The simulation results from [74] plot E_b/N_o against log BER. E_b is the energy per bit and N_o is noise power density. By varying the received signal strength we are effectively varying the S/N ratio, S is signal power and N is noise power, assuming N remains constant for the duration of the test. The two equations are related as follows, where W is the bandwidth and R is the bit rate ;

$$E_b/N_o = (S/N)(W/R)$$

Comparing the two curves shows that the practical results produce a less smooth curve than the simulation results, this is to be expected due to the variability of the practical results. However, the general shape of the curves are similar and show that the simulation results vary in the same manner as the actual results.

7.6 Synchronisation

In order to test any improvement (of the type 3 synchronisation system) over earlier synchronisation systems, an experiment was set-up to test timing jitter tolerance. A simple delay line was built into the synchronisation feeds at the slave nodes (transmitter 2 and the receiver). The delay line comprised a series of high speed logic gates that could be switched in/out of circuit. The maximum delay that could be applied was 2 unit interval (UI), this equates to 2 clock pulses.

The delay line was controlled by a signal generator, this meant a mobile fading enviroment could be simulated, because the delay could be cycled to simulate reception in a mobile vehicle. The test equipment and system layout is shown in Figure 6.29. Figure 6.30 is a scan of the spectrum of the received 10 MHz signal, used for synchronisation in the type 3 system. Figure 6.31 shows the specrum of the recovered combined data signals. The tests were performed with the Trend data testers supplying test patterns over the two simplex channels. The radio frequency levels were adjusted to similar levels and the data testers at the receiver were monitored whilst the variable delay was applied.

The results were interesting. With delay less than 1 UI applied to the receiver synchronisation feed the receiver function was not impaired. Jitter frequencies below 1 KHz had no effect. However as soon as the delay passed 1 UI the receiver failed. The immediate failure at this precise point indicated a hardware limit, which was tracked to the sampling points at the Maxim decoder. This sets the limit on timing performance of the current receiver. The results show that the type 3 system is far more robust, from a synchronisation point of view, than the earlier systems. The type 1 hardwire synchronisation system, without the filters or multiple sample software, could not tolerate any jitter.

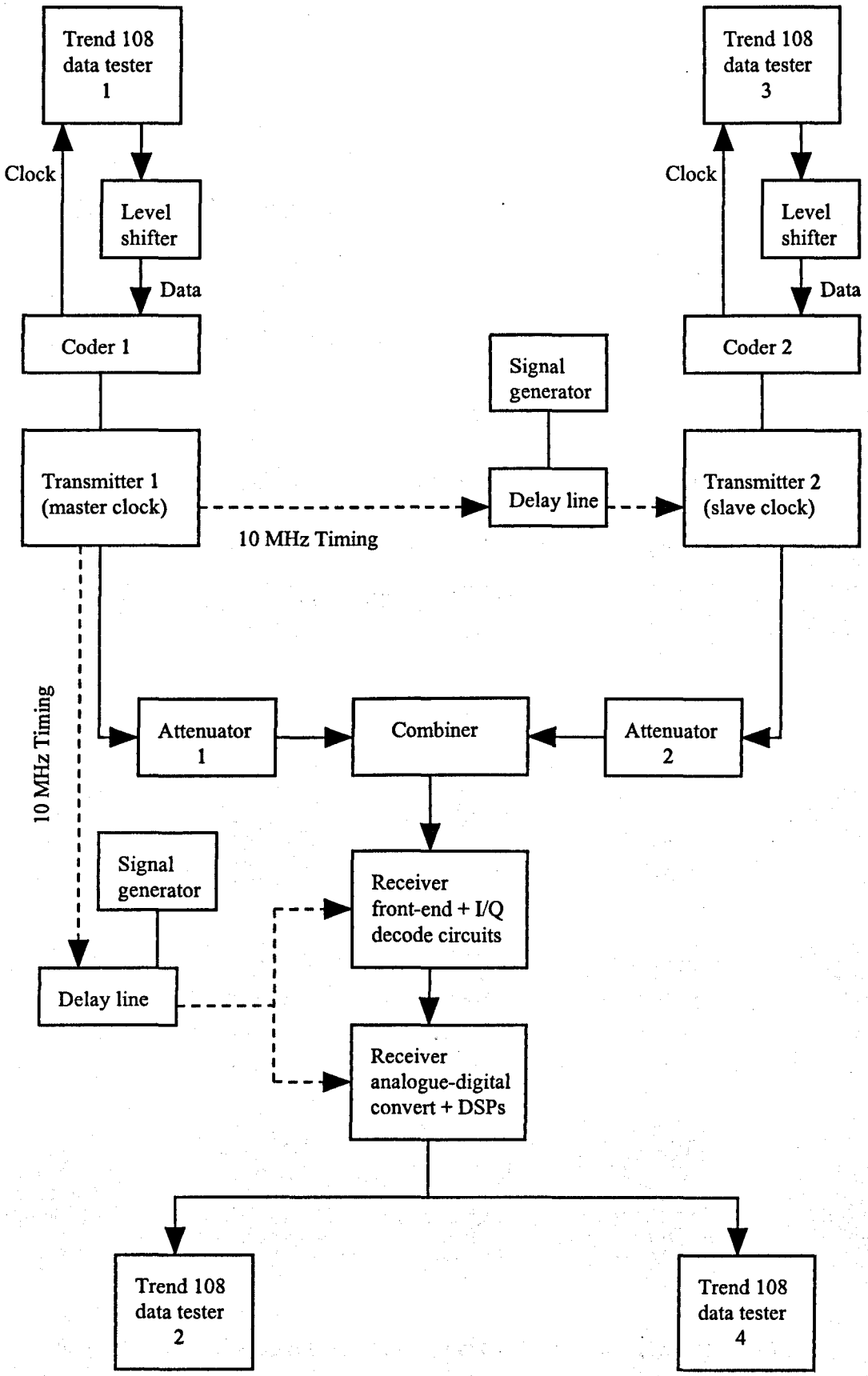


Figure 7.29 Test equipment layout (synchronisation testing)

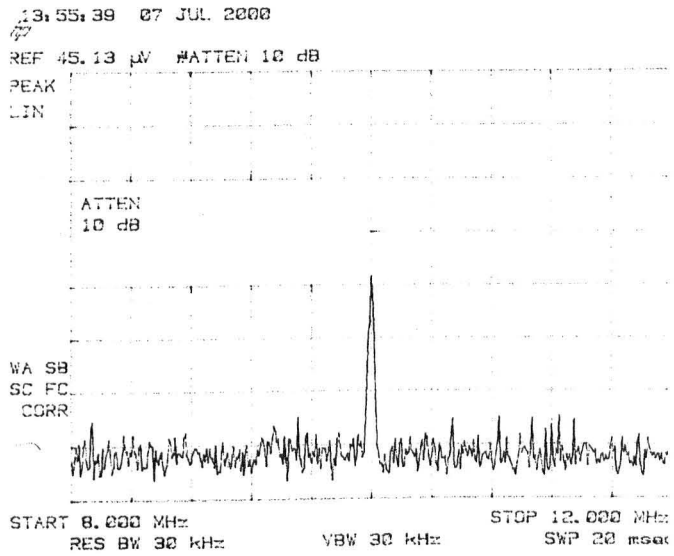


Figure 7.30 Spectrum of recovered 10MHz signal (type 3 sync)

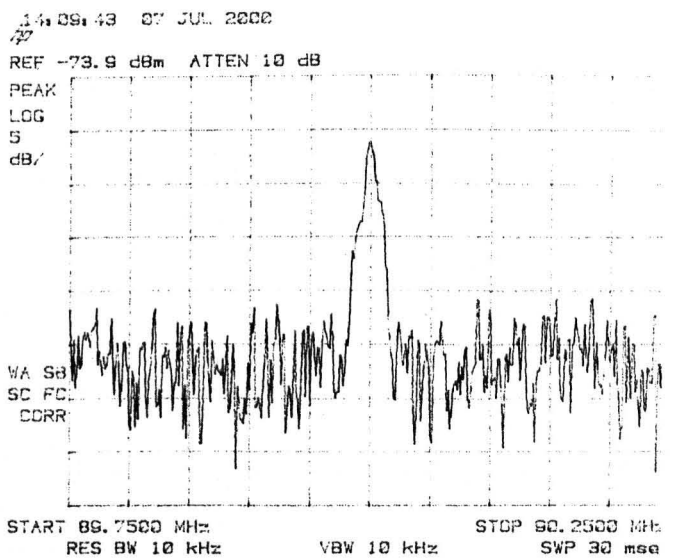


Figure 7.31 Spectrum of recovered data signal (type 3 sync)

7.7 Summary

The results presented in this section show that the CV-CCMA system is a practical and viable multi-access system that can be implemented on current telecommunication systems with a relatively cheap floating-point digital signal processor and commercial integrated circuits. The algorithm used to decode the data is compendious because the software is written in Texas Instruments assembler language. Further development will require a professional development system, as already discussed in the implementation section. The software is reproduced in the appendices. The software will be rewritten in a higher-level language when time allows. This will enable the system to be modularised and allow further research.

The results show up some weaknesses in the hardware developed for the initial implementation of the system. The stable frequency sources showed more drift than was expected and the voltage-controlled oscillators were not linear over the specified frequency ranges, better quality components will improve this. Problems with the noise and unreliability of the circuitry could be resolved by better implementation using a gate array or similar.

The BER tests showed the system to be viable as a possible commercial system and with further development could be adapted for a number of service providers. Some of the simulation work done by other authors has been shown to be technically sound. With some circuit design and construction, the diversity simulations could also be checked.

The more fundamental problems that the results showed up are discussed later, together with possible solutions and further work. The next chapter concludes the thesis, with discussion and suggestions for future work.

8 CONCLUSIONS, DISCUSSION AND FUTURE WORK

8.1 Introduction

The CV-CCMA system has been shown to be a viable multi-access technique, theoretically and practically. The circuitry developed for the implementation of the system has been refined to an experimental (X) model state. Further development work is required before the system can be implemented over a commercial telecommunication network. However the techniques developed at the University of Warwick provide a solid basis for further work. This concluding chapter is subdivided in order to enable anyone requiring information, on specific areas, to be able to find the relevant section quickly. The enhanced CV-CCMA system, with the reduced restrictions on codeword generation (developed at Warwick) and the synchronisation recovery technique with multi-sample software, (also developed at Warwick), has been proven to be a viable multi-access technique.

More theoretically, it would be useful to explore the capacity limits of the new system, particularly with regard to the relationship between the entropy of the g value and the capacity of the associated channels. It would also be interesting to quantify the possible improvements to the system by adding in diversity. This would involve a redesign of the receiver with two aerial systems which would recover the signal and then feed into the main signal processor. Some circuit construction would be required.

The further development of ideas, using the g value to identify which data belongs to which transmitter, leads to possibly the most useful aspect of CV-CCMA. This is the fact that, so long as the g value is unique, there is no requirement to separate users with codes, frequencies or timeslots. This could be the most significant advance of the CV-CCMA system, more significant than the improvement

in bandwidth efficiency (as we are getting close to the Shannon limit). Some experiments were performed, using the CV-CCMA demonstrator, which encouraged this line of research. The receiver requires to be redesigned, with some diversity built in (i.e. multiple aerials) and the ***g*** decode algorithm needs to produce a more accurate result. This is one of the most interesting offshoots to the research.

Further development work is needed on the clock recovery circuitry and on enhancing the multiple-sample software and filter implementations, to further reduce timing sensitivity. This would make the CV-CCMA system a more commercially viable project.

8.2 Coding

It should be possible to implement convolutional-type coding over this system without too much difficulty. Convolutional codes are used where high rates of error correction are required, as in a satellite link stream. However, it would have meant an increase in the processing required and additional buffer circuitry. The software algorithm written to decode the CV-CCMA codewords is relatively simple and quite elegant; it is easy to alter parameters and fault find. This would have been lost to an extent with additional coding and feedback circuitry.

A number of fully developed systems are described in the literature [122], and manufacturers' data. With a more powerful processor this could have been implemented on the Warwick test-bed. This work will be carried out in the future when time and funds allow. This technique will prove to be useful to Cable and Wireless PLC, mainly over heavily loaded satellite links.

Further related work is also planned, when the new signal processor is running (because more memory/processing is required), to investigate the possibility

of applying a Viterbi-type recovery system within the CV-CCMA system. This would make the system more robust in higher noise/low signal strength situations and could be combined with the work already undertaken, and recorded in this thesis, on the codeword restrictions. With a set of correct sequences to be tracked and used for prediction, the recovery of codewords could be made much more robust. Intuitively, this should bring many benefits; however, time limitations restricted these investigations. Further research along this route would be interesting and most of the system could be simulated with *Mathcad*, so the outlay required for equipment and other costs need not be too great.

8.3 Optical systems

Another area of possible further research would be implementation of a more basic collaborative system over a high-capacity optical link, as used by all major telecommunications companies. Current state-of-the-art systems run at data rates in terabits per second, with very low error rates. These systems use a process of wavelength-division multiplexing to pass very large amounts of data over special optical fibres. Very high quality lasers, with narrow beamwidth and high linearity, are used to prevent any crosstalk. Identifying codes are superimposed on the data streams to prevent self-lock problems, and make sure that only the intended receiver responds to the transmit laser. This optical technology is advancing very rapidly and coherent type systems are now being tested around the world.

The collaborative code system could possibly be implemented to reduce further the problems of interference and crosstalk. In addition, a fibre-optic cable could be doped to produce a defined phase shift at a specific wavelength, which could then be used with a CV-CCMA type receiver to recover the original transmitted

data. This work is again very interesting and would be of benefit to an operator running tests on high-capacity wavelength division multiplexing systems.

Currently, research into optical radio systems is producing some useful results [123]. The systems are best deployed in office environments and use an infra-red wavelength multi-access communications link to interconnect a number of computers; a packet-type protocol is usually implemented. The CV-CCMA system could be implemented over these links, reducing interference problems, and enabling the light-emitting diodes to operate on identical wavelengths, which would help to reduce costs.

8.4 Mobile radio

The mobile radio (operator) situation is in a constant state of flux, with revenues driven down to very low levels by regulatory action and by the strength of competition, particularly in the United Kingdom. GSM (global system mobile) has become the *de-facto* worldwide standard for current mobile radio networks and it defines frequencies, modulation techniques and even 'add-on's like voice mail and short messaging services. The main advantage of the system is the tightly defined standards enabling inter-network working on a worldwide basis. However, this stranglehold on the operation and defining technical standards has meant the GSM system has been slow to adapt to changes in the telecommunications environment, in particular the requirement for data transmission at ever-increasing rates and at ever-decreasing costs. GSM struggles to transfer data at 9.6 Kb/s, which is not adequate for most requirements. One possible method of increasing the capacity of the GSM link would be to implement a CV-CCMA type data channel, in addition to the speech

channel, within the GSM telephone and radio base station. Some preliminary work has been done on investigating this.

The problem with GSM mobile telephones is that they are quite complicated devices. Most of the circuitry is fully customised and it is impossible to physically connect to the required interface points, in order to experiment with the system. The earlier Philips devices have been found to be the easiest to work on as the CV-CCMA system can be interfaced using the I^2C (inter-integrated circuit) bus, which is also used on the synthesiser and memory select chips of the CV-CCMA. The control code could be rewritten to select the CV-CCMA circuitry when switched to data mode. To experiment with this implementation an Ericsson radio base station was obtained from One2One, the ex Cable and Wireless PLC mobile phone (GSM) operator in the United Kingdom. It is presently being stripped down. Work will continue into practical installation when time becomes available and this would make another interesting industrially based project.

There are now a number of multi-standard phones on the market, switching between quite different systems. Future work could involve collaboration with a mobile phone manufacturer to try out the system over a test network, possibly implementing CV-CCMA as an add-on to an existing GSM phone and using it for internet/data access.

When testing the CV-CCMA system an amount of forced jitter was applied to the transmitted signal. Although crudely applied it showed that the system is not tolerant to jitter of greater than approximately 0.06 Unit Interval (UI), of any frequency. The later type receiver, with the synchronisation recovery circuit built in, was far more robust than the earlier system, tolerating nearly 1 UI. This is also due to the multiple sample system implemented in the later signal processor. This requires

some further work, because in a mobile situation the signal will be constantly jittered and would not work without much higher tolerance. This also assumes the jitter is not fast enough to affect the synchronisation over a symbol period, another area where further work is required.

Some of the problems could be solved by adding memory into the channel. Gaarder and Wolf [113] give a theoretical start in this direction and a lot of development work has been undertaken on the new spread spectrum systems to cure similar problems.

The code division multiple-access (CDMA) system was investigated as a 'competitor' to the CV-CCMA system in the mobile telephone market. The CDMA system is derived from a secure military system and continues to have the associated advantages and disadvantages of the original system. The proposed worldwide broadband CDMA third generation (3G) standard was investigated and used as a basis for a comparative discussion; this work is recorded in Chapter 3. Quite a momentum has built up in the industry behind 3G mobile systems, so CV-CCMA would be more likely to be accepted in a more specific or unique application. Work could be undertaken looking into one of these.

8.5 Low-cost radio telecommunications

Another current project being considered is an investigation into implementing the CV-CCMA system over a distributed-architecture radio network. Dr Chandler and Rural Radio Systems have produced a system that is designed for local/rural access in developing countries [124]. The system has now been developed and tested at a number of locations and proved reliable and robust. The Rural Radio system comprises a set of nodes that are deployed throughout the area requiring service. The

system is unique in that some of the switching and routeing is done at each node throughout the network, rather than at dedicated switching nodes, which are one of the main expenses in a modern telecommunications network. The system is designed as a low-capacity/low-cost telecommunications network for developing countries with little infrastructure. The nodes are independent and may be quite isolated so power may be solar or some other local source. CV-CCMA could be implemented at the interface with the nodes and at the interconnection of the nodes. This would further reduce the complexity of the planning of the network as the frequencies would not require to be planned as in a cellular network. Effectively they could all operate on the same frequency within each local coverage area, and over the wider network just assigned different sets of codes.

The system is designed as a low-capacity system so each node would only switch and communicate with single-figure numbers of other stations. This work was planned for implementation in the near future; however Dr Chandler has now left Warwick and this would have to be carried out with the agreement of his new company, Rural Radio Limited. The Rural Radio system also uses Texas Instruments signal processors so it would not require much redesign to transfer the system across; however there are other issues which need to be resolved. This is another interesting, virtually stand-alone project that would produce benefits for both parties.

8.6 Military applications

The CV-CCMA system has many applications and during the research and development a number of new applications have been thought of. The original intention was for the system to be used in telecommunication networks, particularly the mobile phone and satellite branches of the industry. It is possible that with further

research and development the CV-CCMA could be used as part of a secure communication system. At present, in this application, a weakness of the system is that with multi-access operating over a narrow band of frequencies it is easy for jamming or monitoring stations to capture the information and re-transmit modified information, or simply block all stations on that channel. This makes the system virtually unusable by the military and security services. With a redesign of the oscillators it should be possible to switch the system to a frequency-hopping mode in which the collaborative codes are transmitted as binary amplitude or phase-shift modulation at a series of pseudo-random frequencies which are pre-programmed into the transmitters and the receivers.

To acquire synchronisation, the receiver would simply slow down its scanning rate of the spot frequencies until a sequence of valid codes was received, at which time it would speed up to recover the data. The system would offer a high degree of security because the scanning rate could be relatively fast – up to about twenty thousand hops per second. The synthesiser need not actually acquire synchronisation as in the current system but would be in a state of constant self-adjustment: with each scan it would adjust its feedback loop and if a stronger signal was received on the next scan then it would adjust a bit further; if the signal was weaker then it would adjust back the other way, that is it would be like a self-servo system. For added security the data could be encrypted and the hop frequencies could be transmitted over the link and changed every few scans.

Prototype synthesisers were simulated, but as the work did not fall into the telecommunications research area and really comprises a whole new project, and as available time was limited, the research was capped. However, some work was

done and this offshoot is interesting and could be progressed forward by somebody else in the future.

8.7 Speech digitisation

The prototype circuits used a speech-digitisation technique known as Delta modulation. Although this system works efficiently, the quality of the speech is poor; at 9.6 Kb/s it sounded very 'metallic'. When the modulator and demodulator were connected back-to-back the quality of the speech did not really improve until the output bit rate was increased to 16 Kb/s. To improve upon this requires further work and the testing and implementation of other speech-digitisation techniques. There are different versions of linear predictive type digitisers that offer reasonable quality at low bit rates, and these could be implemented using the DSP or another daughter DSP, dedicated to the audio processing of the system. The Delta modulators work and are reliable, though further research could be done towards improving the circuitry, perhaps a different system of variable slope integration. The paper by Dunn and Sandler [100] shows a possible line of progression.

In order for a CV-CCMA system to interconnect with established telecommunications networks, it will be necessary to implement one of the standard speech digitisation processes, such as pulse code modulation (PCM) as specified in ITU standard G711 or adaptive PCM or to implement an interface to convert between the standard systems. The lower bit rate schemes, such as linear predictive coding, could be implemented but are processor-intensive, probably requiring a separate processor and hence pushing the costs up. This work is more 'development' than 'research' so was left out of the budgeted time allocation for the University of Warwick.

Speech digitisation is now a relatively stable subject and modern techniques give good quality at low bit rates. Some investigation could be undertaken into which would work best with the CV-CCMA system. Using CV-CCMA to transfer the new low bit rate video streams would also provide an interesting experiment.

8.8 Filters

There is a small amount of work remaining to be done on the filters. On the transmit side, root raised cosine-type filters were implemented in order to reduce the bandwidth of the modulated output signal. If no filters were included the transitions on the I and Q inputs would be sharp and hence the bandwidth occupied would be large, as can be seen on the spectrum diagrams in Chapter 7. The filters were designed with an alpha factor (steepness of cut-off) of 0.5. Problems were encountered because the filters were not matched: a slightly different delay was introduced on the I line to the Q line. The problem was solved by constructing a pair of finite impulse response (FIR) digital filters with TMS320C50 signal processors, and although the filters worked well (see Appendix 9 for taps and circuit), the additional complexity and interfacing caused some problems. Ideally the filter function would be incorporated in the main signal processor. Filter design is a topic in its own right and the design of a set of filters to optimise the performance of the CV-CCMA system would make another interesting project/further work.

8.9 Control

An area requiring further work is in the generation of an automatic gain control signal, adding complexity to the generation of an accurate g value. At present there

are two, 16-bit conversions giving a range of 0 to 65536 or approximately 50 dB. This is reduced because the last two bits are lost in the noise generated and the Burr-Brown integrated circuit gives a negative-to-positive shift, which has to be compensated for and reduces the accuracy of the conversion, so the range is approximately 40 dB in total. In a mobile situation, the amplitude range of the incoming signals will be greater than this and so the automatic gain control (AGC) cannot be directly fed from the signal processor. Additional circuitry is required to control the gain of the amplifiers and report to the signal processor the strength of the incoming signal. Also, any phase variation in the signal will also need to be accurately reported.

It is also possible that the incoming signal will be very large and so will require some attenuation adding in; this fact will need to be reported to the signal processor and used in the calculation of $\boldsymbol{g}$. This is to try to prevent a problem that is similar to the near-far problem of CDMA systems, that is when a strong local signal swamps a more distant weaker signal, by desensitising the receiver and possibly overloading the front end.

Some work was done on investigating the effect of timing variations, i.e. enough timing jitter superimposed on the signal to cause a 'slip' of a bit, over a symbol period. The papers by Poltyrev [125] and Hui / Humblet [126] derived some results for capacity regions of MACs with asynchronous stations. Further study is required of these asynchronous systems, but from a practical point of view it is very difficult to envisage how non-synchronised stations could be resolved at anything like a reasonable data rate. However, this is another very interesting area that deserves further work.

8.10 General

A major enhancement to the system would be the implementation of receiver decoding using one of the new generation of powerful digital signal processors. This is underway at present and work is progressing well. Some reworking of the multi-sample assembler code is required, which is stalling progress at the moment. The new processor gives an order of magnitude increase in processing power over the old processor (1.5 Gflops vs 40 Mflops) and it is hoped to implement most of the system control with the new processor. There are a number of new, low-cost, very high accuracy analogue-to-digital converters that have been launched recently, aimed at the high-fidelity audio market, they would find a ready use in the CV-CCMA system. Work is planned to implement these in the near future.

The current software requires a major amount of work in order to make it more user-friendly, and is available from *Kane Computing* [128]. With the new processor it is planned to write code in a higher level language and use a compiler to produce implementable code. This did not work with the TMS320C31 as the code produced was too time-consuming for the system to work. The new processor will reduce these requirements and the system can be menu-driven from a PC, which can run the I^2C program as a background task. The multi-sample software, which was developed to make the system less prone to timing problems, needs some further development work. On a more practical note, the circuitry would benefit from redesign and implementation in semi-custom silicon; the transmitters and receivers could then be combined and physically located in vehicles for some genuine field-testing. This is currently beyond budget, but, in the future, if time and funds allow it will be done. The work would be expensive, requiring either a grant or some industrial sponsorship, but would be most rewarding.

Another area requiring further research and development is in the area of timing and synchronisation. A number of possible improvements were investigated. The one which showed most promise was the separate synchronisation channel, similar to the synchronisation channel used in CDMA. As in CDMA it could be used for power level adjustment and background data traffic as well. A standard radio link was constructed at 10MHz and used to lock the transmitters and receivers together. Some problems were encountered with receiver desensitising; ideally the synchronisation channel frequency needs to be completely removed from the frequencies used in the decoders and away from any frequencies that would give rise to intermodulation problems. The type 3 synchronisation system was implemented by transmitting the PLL reference frequency (10 MHz) at one master node, and recovering the signal at the slave nodes. This was the simplest and most cost effective implementation. However, the frequency needs to moved to a licensed band to prevent legal problems. The hybrid receivers would need to be modified and improved upon. However, the system works well and is fine for laboratory use.

Given sufficient time and resources, the recovered packet system (type 2) would be another way of advancing the CV-CCMA as a commercially viable product. However, this will require substantial work.

The original system free-runs the oscillators in the receiver and transmitter, assuming they will not drift out of synchronisation (1 bit) for the duration of the communication. This meant using quite expensive temperature-controlled oscillators, which would still be used, but phase-locked to the recovered synchronisation channel. Another solution would be to time from the GPS satellites, as in some CDMA systems; this requires more research. With the recovered timing solution (type 2) specific coding was used to maintain synchronisation, the

synchronisation was initialised by the training sequence and then maintained by the receipt of packets of data, and timing packets, from the master source. This did not cause a problem with the Delta modulators, as they would output the 'quiet' pattern with no input. With a data message transferred over the link it was essential to keep the output sequencing, in order to recover a valid clock. These problems could be solved with further development.

Symbol timing, and start of message / end of message timing have caused practical problems getting the system to work. The initial idea of running a training sequence, although proven to work, takes up too much time and needs further work to develop better solutions. Start and stop codeword sequences need developing with the necessary detection circuitry.

The idea of applying equalisation at the transmitter, to reduce multipath problems (and other impairments) could be further investigated. The latter CV-CCMA systems (developed at Warwick), would adapt most easily to this type of enhancement. The information required to perform the equalisation could be transferred over a modified timing channel, or over a specific control channel, or control 'packets'. This would be another interesting development to the CV-CCMA system and is worthy of some more research and development.

8.11 Conclusions

One of the main conclusions of the research has been to show the CV-CCMA system to be a practicable multi-access radio system. The simulations that were done at the start, both in *Mathcad* and *Matlab*, suggested that the system was a realisable possibility. The design and construction of the new system, even with restricted numbers of transmitters and sets of codewords used, proved the system to be viable.

The theory behind the new less-restrictive codeword sets, which was derived and developed at Warwick alongside the coder design, also proved to be valid. It has been shown that a relatively cheap digital signal processor has sufficient processing power to run the CV-CCMA system. The circuitry designed for use with the system, especially the high stability-oscillators, will be put to good use as test-beds for future developments. The clock recovery circuits, developed as part of the second receiver circuit to reduce timing sensitivity, could be used for timing recovery from an independent timing channel. The first CV-CCMA system was prone to timing related problems, these were reduced by the implementation of the multi-sample software, the clock recovery system and the filters.

The research has increased Cable and Wireless PLC's, knowledge base of multi-access systems and updated electronic circuit design and development techniques. The digital signal processor programming skills acquired will also be put to good use in the future.

During the research it has been found that the information theory aspects of multiple-access theory have been very thoroughly investigated by a number of researchers, especially in recent years. Details such as channel capacity limits, error performance in multiple-access channels, etc., have been subject to rigorous mathematical treatment, often with results 'proven' by simulation. Now that the CV-CCMA system is working it would be interesting to try and explore some of these limits, particularly the relationship of g to channel capacity, discussed earlier in this chapter.

At the present time (1999-2000), Cable and Wireless PLC is undergoing a major reorganisation. The global network is being reconfigured towards an internet protocol (IP) based architecture. Large chunks of the company have been

sold and the remaining units are changing beyond recognition; even in the relatively short period of this research, the company has changed drastically. These changes mean that the original reasons for starting the research are no longer valid; however, an IP-based system will still require customer access points, especially at radio frequencies, and the research undertaken over the last four years will be updated to work with these new systems as and when they start to be implemented in the network.

REFERENCES

[1] G.Bennett, *Pulse code modulation and Digital Transmission,* Marconi Instruments Publications, ISBN 0950494127, pp8-13, March 1972

[2] *ITU-T Recommendations 1998* Published by ITU, Place des Nations, CH-1211, Geneva 20 , Switzerland (CD-ROM) or download from http://www.itu.int

[3] S.Yang ,*C.D.M.A. Engineering* , Artech House Publications, ISBN 0 89006-991-3 pp75-102, June 1998

[4] T.Kasami, S.Lin, S.Yamamura, 'Further results on coding for a multiple access channel', *Conference proceedings Hungarian colloq. on Information Theory* , pp 369-391, August 1975

[5] E.Weldon, 'Coding for a multiple access channel', *Information and control* ,Vol. 36, pp 256-274, March 1978

[6] M.Deatt, J.Wolf, 'Some very simple codes for the non-synchronised two-user multi-access adder channel with binary inputs', *IEEE Transactions on Information Theory* , Vol. IT-24, No.5, pp 635-636, 1978

[7] S.Chang, E.Weldon, 'Coding for T-user multiple access channels' *IEEE Transactions on Information Theory* ,Vol. IT-25, No. 6, pp 684-691, November 1979

[8] T.J. Ferguson, 'Generalised T-user codes for multiple-access channels' *IEEE Transactions on Information Theory,* Vol. IT-28, No. 5, pp 775-778 Sept., 1982

[9] S.C.Chang, 'Further results on coding for T-user multiple-access channels' *IEEE transactions on information theory* ,Vol. IT-30, No. 2, pp 411-415, Mar. 1984

[10] P.van der Coeberg, H.C.van der Tilborg, 'A family of good uniquely decodable code pairs for the two access binary channel', *IEEE Transactions on Information Theory* Vol. IT-31, No. 1, pp 3-9 ,Jan. 1985

[11] G.H. Khachatrian, 'New construction of linear delta decodable codes for two user adder channel' *Control and Information theory*, Vol. 13, pp 275-279, 1984

[12] J.Wilson, 'Error correcting codes for a T-user binary adder channel', *IEEE Transactions on Information Theory* ,Vol. IT-34, No. 4, pp 880-890, July.1988

[13] A.Cohen, J.Heller, A.Viterbi, 'A new coding technique for asynchronous multiple-access communication' *IEEE Transactions on Communications* ,Vol. COM19, No. 5 pp 849-855, Oct 1971

[14] P.G.Farrell, A.Brine, A.P.Clark, D.J.Tait, 'Limits of radio communication – collaborative transmission over cellular radio channels', presented at, *NATO advanced study institute*, 'Performance limits in communication theory and practical' Il Ciosco, Italy, July 1986

[15] A.P. Clarke, F. McVerry, 'Channel estimation for a HF radio link', *IEE Proc. Pt. F*, Vol.128, pp33-42, Feb 1981

[16] F.R. Magee, J.G. Proakis, 'Adaptive maximum likelihood sequence estimation for digital signalling in the presence of intersymbol interference', IEE Trans. Vol. IT-19, pp 120-124, Jan 1973

[17] R.B.J.T. Allenby, *'Linear Algebra'* Modular mathematics series, published by Edward Arnold ,ISBN 0340 610441, pp127-141 ,1998

[18] S.Chandler, P.Thomas, 'Permissible codewords for CV-CCMA system', *IEE Electronic Letters* , Vol 33, No 22, pp 1863, October 1997,

[19] *Institution of Electrical Engineers*, Savoy Place, London

[20] *Institute of Electrical and Electronic Engineers*, 345 East 47th Street, New York NY10017-2394, U.S.A.

[21] Communications Society, *Institute of Electrical and Electronic Engineers*, 345 East 47th Street, New York NY10017-2394, U.S.A., www.comsoc.org

[22] www.iee.org.uk/publish/journals/profjml/eleclett.html

[23] www.iee.org.uk/publish/journals/profjml/ieeproc.html

[24] BIDS, *Bath Institute Database* http://www.bids.org

[25] *Cable and Wireless PLC* Library Reference, 320 Westwood Heath Road, Coventry, West Midlands CV4 8GP. also, *Cable and Wireless Communications PLC* Reference, Millars Close, Bracknell, Berks.

[26] www.labs.bt.com/library

[27] *British Telecommunications Technical Journal*, BT Research Laboratories, Martlesham Heath, Ipswich, Suffolk, IP5 7RE

[28] *Qualcomm Inc.* , http://www.qualcomm.com

[29] *Hewlett-Packard Inc.* , http://www.hewlett-packard.com (technology/docs)

[30] *Philips Electronics NV.* , http://www.semiconductors.philips.com

[31] *Texas Instruments,* Dallas, Texas, USA, http://www.ti.com

[32] *Analog Devices,* One Technology Way, PO Box 9106, Norwood, MA, USA http://www.analog.com

[33] *Harris Electronic* (Defence Systems Division) http://www.harris.com

[34] *National Semiconductor*, PO Box 58090 Santa Clara, California, USA http:// www.national.com

[35] R.C. Sommer, 'High efficiency multiple-access communications through a signal processing repeater', *IEEE Transactions on Communications* , Vol COM-16, pp 222 – 232, April 1968

[36] A.J. Viterbi 'A processing satellite transponder for multiple-access by low-rate mobile users', *Digital Satellite Communications*, 4th international conference, Montreal, pp 166- 174, 1978

[37] L.Gyorfi, I.Kerekes 'A block code for noiseless asynchronous multiple-access OR channel', IEEE Transactions on Information Theory, Vol. IT-27, No. 6, pp 788-791, November 1981

[38] J.Wolf, 'Constructive codes for multi-user communication channels' Part 2 *,Information theory, new trends and open problems*, CISM courses and lectures No.219, Springer-Verlag, June 1975

[39] Edward C. van der Meulen, 'A survey of multi-way channels in information theory:1961 - 1976', *IEEE Transactions on Information Theory*, Vol. IT-23, No.1 pp1 – 37 , January 1977

[40] J.Schalkwijk, 'The binary multiplying channel, a coding scheme that operates beyond Shannons inner bound region', *IEEE Transactions on Information Theory* ,Vol. IT-28, No. 1, pp107-110, January 1982

[41] R.Gallager, 'A perspective on multi-access channels', *IEEE Transactions on Information Theory*, Vol. IT-31, No. 2, pp 124-142, March 1985

[42] J.Massey P.Mathys, 'The collision channel without feedback', *IEEE Transactions on Information Theory* ,Vol. IT-31, No.2, pp192-204, 1985

[43] J.Massey, 'Channel models for random access systems', *NATO advanced study institute*, 'Performance limits in communication theory and practice' Castelvecchio Pascoli, Italy, July 1986

[44] P.Vanroose, 'Code construction for the noiseless binary switching multiple-access channel' *IEEE Transactions on Information Theory* ,Vol. IT-34, No.5, pp 1100-1106 September 1988

[45] C.E. Shannon, 'A mathematical theory of communication', *Bell System Technology Journal* ,vol. 27, pp 379-423, July and October 1948

[46] C.E. Shannon, 'Two-way communication channels', *Procedings 4th Berkeley Math Symposium,* University California Press, pp 611-644, 1961

[47] P.Bergmans and T.Cover, 'Co-operative broadcasting', *IEEE Transactions on Information Theory* , vol. IT-20, no.3, pp 317-324, 1974

[48] A.Wyner, 'Recent results in the Shannon theory', *IEEE Transactions on Information Theory*, vol. IT-20, no.1, pp2-10, Jan.1974

[49] Edward C.van der Meulen, 'A survey of multi-way channels in information theory 1961-1976', *IEEE Transactions on Information Theory*, vol. IT-23, no.1, pp1-37, Jan 1977

[50] Multiple Access Communications, foundations for emerging technologies, *IEEE Press,* edited by Norman Abrason, ISBN 0879422920, Feb. 1994

[51] S.Verdu, 'Recent progress in multi-user detection' from *'Advances in communication and signal processing'* , Heidelberg Press, 1989

[52] A. El Gamal, T.Cover, 'Multiple user information theory', *Proceedings of the IEEE* ,vol. 68, no.12, pp1466-1483, December 1980

[53] Edward C.van der Meulen, 'Capacity Limits for Multiple Access Channels',in *'NATO advanced study institute'* on 'Performance limits in communication theory and practice', Ciocco, Italy, 1986

[54] W.Lee, 'Estimate of channel capacity in a Rayleigh fading enviroment', *Proceedings of the IEEE* ,vol. 79, no.12, December 1990

[55] F.Ali, B.Honary, 'Low complexity soft decision decoder for multi access', *IEE Electronic letters*, vol. 27, no. 13, pp1167-1169, June 1991

[56] J.Ni, B.Honary, 'State independent codes for multi-access', *IEE Proceedings 1, Communications, Speech and Vision*, Vol.140, No.3, pp185-188, June 1993

[57] G.Ungerboeck, 'Channel coding with multilevel/phase signals', *IEEE Transactions on Information Theory*, vol. IT-28, no.1, Jan1982

[58] A.J.Viterbi, J.K.Omura, *'Principles of digital communication and coding'*, section 2, published by McGraw-Hill, ISBN 0-07-066626-1, pp411-423, Jan. 1979

[59] P.G.Farrell, 'Survey of channel coding for multi-user system' *in Skwirzynski J. 'New concept in multi user communication'*, Sitjhoff and Nordhoff, pp113-159, 1981

[60] T.Kasami, S.Lin, 'Coding for a multiple-access channel', *IEEE Transactions on Information Theory,* Vol. IT-22, No.2, pp129-137, March1976

[61] J.Wolf, 'Coding techniques for multiple-access communication channel' in Skwirzynski *'New concept in multi-user communication'* , Sitjhoff and Nordhoff, pp 83-103, 1981

[62] S.Chang, E.Weldon, 'Coding for T-user multiple-access channel', *IEEE Transactions on Information Theory*, vol. IT-25, No.6, pp 684-691, Nov.1979

[63] E.Weldon Jr., 'Coding for multiple-access channel' *Information and Control* , vol. 36 pp 256-274, March 1978

[64] J.Ni, 'Codeword synchronisation of CCMA Scheme', *IEE Proceedings, Communications, Speech and Vision* , Vol.140, No.6, pp425-428, Dec.1993

[65] F.Ali , B.Honary, 'Low complexity soft decision decoding technique for T-user collaborative coding multiple-access channels', *IEE Electronic Letters* ,Vol. 27 No.13, pp1167-1169, June 1991

[66] M.Daett, J.Wolf, 'Some very simple codes for the non-synchronised two user multiple-access channel with binary inputs' *IEEE Transactions on Information Theory,* Vol. IT-24, No.5, pp635-636, September1978

[67] J.Ni, B.Honary, 'On state independent unique decodable CCMA codes' *IEE Proceedings, Communications, Speech and Vision* , vol. 140, no. 3, pp 185-188, June 1993

[68] Y.Wu, S.Chang, 'Coding scheme for multiple-access adder channel with and without idle users' *IEE Electronic Letters* ,vol. 31, no.7, pp523-524

[69] Y.Steinburg, ' Resolvability theory for the multiple-access channel' *IEEE Transactions on Information Theory* ,vol. 44, no. 2, pp 472-487, March1998

[70] K.Feher, '*Advanced digital communications systems and signal processing techniques*' , Prentice-Hall Inc., ISBN 0130111988, 1989

[71] http://www.tm.agilent.com/Masterservlet

[72] http://www.orc.soton.ac.uk

[73] *IEEE Communications Magazine*, Vol. 36, No.12, December 1998

[74] F.Ali, S.Chandler, S.Soysa, 'Complex-valued receiver for multiuser collaborative coding schemes', *IEE Electronic Letters* , Vol. 31, No.5, pp341-342, March 1995

[75] G.Khachatrian, S.Martorissian, 'Code construction for T-user noiseless adder channel', *IEEE Transactions on Information Theory* , vol. 44, no.5, pp1953-1957, September 1998

[76] http://www.qualcomm.com

[77] R.A.Scholtz, 'The Spread Spectrum Concept', *IEEE Transactions* , vol. 25, no.8 pp748-755, August 1977

[78] R.Pickholtz, D.Schilling, L.Milstein, 'Theory of spreadspectrum communications-a tutorial', *IEEE Transactions on Communications* , vol. com 30, no.5, pp 855-884 May 1982

[79] S.Verdu, 'Recent progress in multiuser detection', from '*Advances in communication and signal processing*', Heidelberg Press, Jan. 1989

[80] S.Yang, '*CDMA RF System Engineering*', Artech House Publications ISBN 0-89006-991-3, pp149-180, 1998

[81] A.Viterbi, 'Spread spectrum communications – myths and reality', *IEEE Communications Magazine* , vol.17, no.3, pp 11-18, May 1979

[82] A.Viterbi, 'Wireless digital communication, a view based on three lessons learned' *IEEE Communications Magazine* , vol. 29, no.9, pp33-36, Sept 1991

[83] S.Moshari, 'Multi-user detection for DSCDMA communications', *IEEE Communications Magazine* ,vol.34, no. 10, pp116-123, October 1996

[84] F.Adachi, M.Sawahashi, H.Suda, 'Wideband DS-CDMA for next generation mobile communication systems', *IEEE Communications Magazine* ,vol.36, no. 9, pp56-69, September 1998

[85] H.Habudi, T.Hasequi, Y. Makura, 'A code division multiplex using Manchester coded orthogonal sequences', *Electronic Communications* , Japan, vol. 75, no. 7, pp 58-61

[86] Lin, Franz and Senior, 'TMS320 signal processor' *IEEE Transactions on Information Theory* , vol. 75, No.9, pp538-552, 1987

[87] J.Bradley, '*TMS320C30 hardware applications*', TI press, 1991

[88] Randy Nestle, '*IEEE floating point or Texas Instruments floating point*', TI press, 1993, (also electronic format from website)

[89] Rulph Chassaing, '*Digital signal processing with C and the TMS320C30*', published by John Wiley, ISBN 0-471-57777-4, 1993

[90] Okhay Alkin, ' *Digital Signal Processing a laboratory approach using PCDSP*' published by Prentice Hall, ISBN 0-13-328139-6, 1994

[91] H.Sorenson, J.Chen, '*A digital signal processing laboratory using the TMS320C30*' published by Prentice Hall, ISBN 0-13-741828-0, 1997

[92] *BT Technology Journal* ,Vol. 10, No.1, January 1992

[93] A.Williams, F.Taylor, '*Electronic filter design handbook*' , published by McGraw Hill, ISBN 0-07-070434-1, 1989

[94] Z.Kostic, S.Seelaraman, 'DSP in cellular radio', *IEEE Communications Magazine* vol.35, no.12, pp22-35, December 1997

[95] *Microwave and RF Magazine* , http://www.wirelessdesignmag.com

[96] *Microwave Journal*, http://www.mwjournal.com

[97] Gene Heftman, 'Changing times challenge the crystal industry', in *Microwave and RF Magazine* , pp 31-38, November 1998

[98] Bob Buxton, 'Measurement methods and analysis of digital radio signals', in *Microwave and RF Magazine* , pp32-47, July 1995

[99] Liu, 'Impact of I and Q inbalance on QPSK' *IEEE Transactions on Communications'*,vol.44, no.3, pp984-989, August 1998

[100] C.Dunn, M.Sandler, ' Fixed and adaptive sigma delta modulators with multi-bit quantisers' in *Applied Signal Processing(UK)* , vol.3, no. 4, pp212-222, 1996

[101] J.Armstrong, D.Strickland, 'Symbol synchronication using signal samples and interpolation', *IEEE Transactions on Communications* , vol. 41, no. 2 pp318-321

[102] TIA/EIA IS95, '*Mobile base station compatibility standard for dual mode wideband spread spectrum cellular system'*, published by Telecommunications Industry Association/ Electronic Industry Association, 1994

[103] http://www.semiconductor.philips.com/I2C

[104] H.Nyquist, 'Certain topics of telegraph transmission theory', *Transactions of the American Institute of Electrical Engineers*, vol.47, pp617-644, April 1928

[105] B.Sklar, '*Digital Communications*', Publisher Prentice Hall International I.S.B.N. 0-13-212713X, pp155-182, 1988

[106] J.G.Proakis, '*Digital Communications*' Publisher McGraw Hill International ISBN 0-07-113814-5, pp127-148, 1st edition 1983, 2nd edition 1989

[107] D.Slepian and J.Wolf, 'A coding theorem for multiple access channels with correlated sources', *Bell system technology* , vol. 52, pp 1036-1076, Sept.1976

[108] R.Ahlswede, 'Multi-way communication channels' in proceedings 2nd *International Symposium Information theory.* , pp 23-52, Armenia, 1984

[109] A.Wyner, 'Recent results in the Shannon theory' *IEEE Transactions on Information Theory,* vol. IT-20, No.1, pp2-10, Jan 1984

[110] C.Keilers, '*The capacity of the spectral Gaussian multiple access channel*' PhD Thesis, Department of Electrical Engineering , Stanford University, Stanford, California, April 1976

[111] F.Ali, S.Soysa, S.Chandler, 'Channel capacity limits for multiple-access channels with slow rayleigh fading', *IEE Electronic Letters,* Vol. 31, No.5, pp 1435 – 1437 August 1996

[112] G.Ungerboeck, 'Channel coding with multilevel/phase signals', *IEEE Transactions on Information Theory*, vol. IT28, pp55-67, 1982

[113] N.Gaarder and J.Wolf, 'The capacity region of a multiple access discrete memoryless channel can increase with feedback' *IEEE Transactions on Information Theory,* Vol. IT-21, No.1, pp100-102, January 1975

[114] J.A. Thomas, 'Feedback can at most double Gaussian multiple access channel capacity', *IEEE Transactions on Information Theory*, Vol. IT-33, No.5, pp711-716, September 1987

[115] D.Strang, '*Linear Mathematics*', section 2.5.1, pp127-141, Publisher McGraw Hill, ISBN 0-34061044-1, Mar. 1982

[116] Philips Semiconductor Division, '*I^2C Bus Technical Specifications and Implementation*' , available at ftp website or datasheet published by Philips, 1990

[117] Hewlett Packard *'Digital modulation in communications systems'*, Application note 1298, pp20-32, available from HP European Marketing Centre, PO Box 999, 1180A2 Amsteveen, The Netherlands

[118] Forrest Barker, *'Communications electronics, systems devices and circuits'* published by Prentice-Hall, ISBN 0-13-153982-5, pp112-128, 1987

[119] http://www.ti.com/dev/sigproc/filter/menu.html

[120] *Burr-Brown* Corporation, 6730 South Tucson Boulevard, Tucson, Arizona, AZ 85706 (Farnell at Leeds stock all data books for B-B)

[121] Electronic Industry Association, '*Recommended Standard 232*', available from http://www.eia.com or ITU Geneva as equivalent V24/V28 standards

[122] Oktay Alkin, '*Digital signal processing a laboratory approach using PC-DSP*', pub. Prentice-Hall, pp137-142, ISBN 0-13-328139-6, 1994

[123] *IEEE Communications Magazine* , Special Edition, Optical Radio, Feb.1998

[124] Rural Radio Limited, Hill Street, Coventry, England

[125] G.Polytyrev, 'Coding in an asynchronous multiple-access channel' *Problemy Peradachi Informatsii,* Vol. 19, No.3, July-Sept. 1983

[126] J.Hui, P.Humblet, 'The capacity region of a totally asynchronous multiple-access channel', *IEEE Transactions on Information Theory* , Vol. IT-31, 1985

****

[127] P.Thomas, 'Radio multi-access system aimed at the telecommunications market' *Cable and Wireless World* , Vol.12, No.51, pp9-10, August 1998

[128] P.Thomas, '*Multi-access decode software*' , Applications CD-ROM, Texas Instruments, Kane Computing, March 1999

[129] P.Thomas,S.Chandler, 'The implementation of a multiple-access radio system' in proceedings of *The Second Texas Instruments DSP Conference*, pp 130-133, Paris, Sept. 1998

[130] P.Thomas, 'The implementation of a multiple-access radio system', Cable and Wireless internal report, July 2000

GLOSSARY OF ACRONYMS

A/D	Analog to digital conversion
ABM	Asynchronous balanced mode
ACELP	Algebraic code(-book) excited linear predictive
ADM	Asynchronous disconnected mode
ADPCM	Adaptive differential pulse code modulation
AF	Audio frequency
AFC	Automatic frequency control
AGC	Automatic gain control
AM	Amplitude modulation
AMPS	Advanced mobile telephone system
AMR	Adaptive multirate (speech codec)
ANSI	American National Standards Institute
ARQ	Automatic repeat request
ASIC	Application-specific integrated circuit
ASK	Amplitude shift keying
AT&T	American Telegraph and Telephone
ATM	Asynchronous Transfer Mode
BABT	British Approvals Board for Telecommunications (UK)
BB	Baseband
BER	Bit error ratio
BiCMOS	Bipolar complementary metal-oxide semiconductor
BID	Bath (University) information database
BPSK	Binary phase shift keying
BS	Base station
BSC	Base station controller
BT	Bandwidth (B) multiplied by bit period (T)
BT	British Telecom
BTS	Base transceiver station
BZT	Bundesamt fur Zulassung in der Telekommunikation
C/I	Carrier-to-interference ratio
CAW	Cable and Wireless

CB	Cell broadcast
CC	Convolutional code
CCMA	Collaborative coding multiple access
CCIR	International Radio Consultative Committee
CCITT	International telegraph and telephone Consultative Committee
CDMA	Code division multiple access
CELP	Code(-book) excited linear predictive (speech codec)
CEPT	European Conference of Posts and Telecommunications
CMOS	Complementary metal oxide semi-conductor
CODEC	Coder-decoder
CPU	Central processing unit
CRC	Cyclic redundancy code/check
CS	Coding scheme
CSPDN	Circuit switched public data network
CT	(1) Call transfer, call completion (2) cordless telephone
CTIA	Cellular Telephone Industry Association (USA)
CV-CCMA	Complex valued collaborative coding multiple access
CWC	Cable and Wireless Communications
D/A	Digital-to-analogue conversion
DCA	Dynamic channel allocation
DECT	Digital Enhanced Cordless Telecommunications
DFE	Decision feedback equaliser
DGPT	Directorate General of Post and Telecommunications (France)
DIL	Dual in line
DISC	Disconnect
DLC	Data link control
DM	Disconnect mode
DQPSK	Differential quadrature phase shift keying
DS-CDMA	Direct sequence code division multiple access
DSB	Double sideband
DSP	Digital signal processor
DSSS	Direct sequence spread spectrum
DTC	Digital traffic channel
DTE	Data terminal equipment

DTI	Department of trade and Industry, United Kingdom
DTMF	Dual-tone multifrequency
DTX	Discontinuous transmission
DWDM	Dense wavelength division multiplex
EPROM	Erasable programmable read-only memory
EEPROM	Electrical erasable and programmable read-only memory
EFR	Enhanced full-rate speech codec
EIA	Electronics Industry Association (USA)
EINT	External interrupt (to the DSP)
EMC	Electromagnetic compatibility
ETSI	European Telecommunications Standards Institute
EVRC	Enhanced variable rate (speech) codec
FCC	Federal Communications Commission (USA)
FCCH	Frequency correction channel
FCS	Frame check sequence
FDM	Frequency division multiplex
FDMA	Frequency division multiple access
FDX	Full duplex
FEC	Forward error correction
FIFO	First in first out (electronic buffer)
FIR	Finite impulse response (type of digital filter)
FM	Frequency modulation
FR	Full-rate speech codec, standard speech codes used in GSM
FTP	File transfer protocol
FVC	Forward voice channel
GaAs	Gallium Arsenide, used in high speed integrated circuits
GFSK	Gaussian-filtered frequency shift keying
GPRS	General packet radio service
GPS	Global positioning system
GSM	Global system for mobile communications
HEMT	High electron mobility transistor
HPIB	Hewlett Packard interface bus (high speed parallel)
IC	Integrated circuit
IEE	Institute of Electrical Engineers (UK)

IEEE	Institute of Electrical and Electronic Engineers (USA)
IF	Intermediate frequency
IM	Intermodulation
IN	Intelligent networks
I/O	Input/output
IP	Internet protocol
IP3	Intercept point of the third order
ISDN	Integrated services digital network
IIC	Inter Integrated Circuit (Philips 3-wire bus)
ISI	Inter-symbol interference
ITU	International Telecommunication Union
IWF	Interworking function
JTC	Joint Technical Committee (TIA body)
LAN	Local-area network
LAPM	Link access procedure for modems, protocol for data exchange
LEC	Local exchange centre, local switch in a PSTN or ISDN
LED	Light emitting diode
LEO	Low-earth orbit
LFSR	Linear feedback shift register
LNA	Low-noise amplifier
LNB	Low noise block (amp. and freq. conversion)
LO	Local oscillator
LP	Linear prediction/predictive (speech coding)
LPC	Linear predictive (speech) coding
MAC	Medium access control
MAD	Multiply accumulate with dynamic address (TI assembler)
MIP	Million instructions per second
MLSE	Maximum likelihood sequence estimation
MMI	Man/machine interface
MMIC	Monolithic microwave integrated circuits
MODEM	Modulator demodulator
MOS	(1) metal-oxide semiconductor; (2) mean opinion score
MOSFET	Metal-oxide semiconductor field effect transistor
MoU	Memorandum of Understanding

MSC	Mobile services switching centre
MSK	Minimum shift keying
MUX	Multiplexer
NASA	North America Space Agency
NB	Narrowband
NF	Noise figure (signal to noise i/p to signal to noise o/p)
NRZ	Non-return zero (logic signal)
NT	Nontransparent services or Northern Telecom
NTP	Nominal transmitted power
NTT	Nippon Telephone and Telegraph company
NULL	Null information
OA	Originating address
OQPSK	Offset quadrature phase shift keying
OSI	Open systems interconnections
PA	Power amplifier
PACS	Personal Access Communications System
PAD	Packet assembler/dissembler
PAMR	Public access mobile radio
PBX	Private branch exchange
PC	Personal computer
PCB	Printed circuit board
PCM	Pulse code modulation
PCN	Personal comms. network, also referred to as DCS 1800
PCS	Personal comms. system, referred to as PCS 1900
PDS	Packet data on signalling channels
PID	Protocol identifier
PIN	Personal identification number
PLC	Public liability company (UK)
PLL	Phase-locked loop
PLMN	Public land mobile network, cellular network
PMR	(1) Private mobile radio; (2) professional mobile radio
POTS	Plain old telephony service
ppm	Parts per million
PRBS	Pseudorandom bit sequence

PSK	Phase shift keying
PSPDN	Packet-switched public data network
PSTN	Public-switched telephone network, standard wireline network
PTM	Point-to-multipoint
PTP	Point-to-point
PTT	Push-to-talk
PWT	Private wireless telecommunications
QAM	Quadrature amplitude modulation
QCELP	Qualcomm codebook excited linear predictive
QPSK	Quadrature phase shift keying
RA	Rate adaptation
RACE	Research and Development of Advanced Communications Technologies and Services
RAM	Random access memory
RAND	Random value
RDS	Radio display system
REJ	Reject, layer 2 frame/information
RF	Radio frequency
RINT	Receive interrupt (to the DSP)
RISC	Reduced instruction set computer
RLL	Radio local loop
RLP	Radio link protocol, link protocol to reduce the error rate
RNR	Receive not ready, layer 2 frame/information
ROM	Read-only memory
RPE-LTP	Regular pulse-excited-long-term prediction
RR	Receive ready, layer 2 frame/information
RS	Recommended standard (EIA standards)
Rx	Receiver
SABM	Set asynchronous balanced mode
SAPI	Service access point identifier
SAW	Surface acoustic wave (ceramic-based filter)
SCH	Synchronisation channel
SDMA	Space diversity multiple access
SEQAM	Spectrally efficient quadature amplitude modulation (IS-661)

SIM	Subscriber identity module
SMS	Short message service
SNR	Signal-to-noise ratio
SREJ	Selective reject, layer 2 frame/information
SRTS	Serial residual time stamp (used to sync real time ATM)
SSB	Single sideband
SS No.7	Signaling System Number 7
TA	Type approval or terminal adapter
TCXO	Temperature controlled crystal oscillator
TDM	Time division duplex
TDMA	Time division multiple access
TE	Terminal equipment
TETRA	Terrestrial trunked radio (ex trans-European trunked radio)
TI	Texas Instruments
TIA	Telecommunications Industry Association (USA)
TSS	Telecommunications standardisation sector (part of ITU)
TTL	Transistor transistor logic
Tx	Transmitter
UA	Unumbered acknowledge, layer 2 frame/information
UCI	Universal computer interface
UHF	Ultra high frequency
UI	(1) Unnumbered information (2) user interface (3) Unit interval
UMTS	Universal Mobile Telecommunication System
VA	Viterbi algorithm
VAD	Voice activity detection
VCO	Voltage-controlled oscillator
VGA	Variable gain amplifier
VOGAD	Voice Operated Gain Adjusting Device
VPN	Virtual private network
VSELP	Vector sum excited linear predictive (IS-136)
WACS	Wireless Access Communications System
WCDMA	Wideband CDMA
WIN	Wireless intelligent network (usually refers to IS-41/IS-136)
WLAN	Wireless local-area network

```
;                FAST5.ASM                                  26 Feb 1998
;                                                           P.Thomas
;
; CV-CCMA main decode program.
; This prog initialises the serial port of the c31dsk,;receives the ;vector on (3 values BPSK) a
RINT and;multiplies by the stored ISP ;matrix, finds; the minimum Euclidean distance (no
root=call ;fast,root=call ;euclid<x50>); and either stores the results decodes them and stores at
the transmit buffer.; The clocking into the DSP is derived from the 50MHz ;system clock; and
the data is clocked out by tcklo; NB START FRAMES AT SAME TIME, sync 00 11 00 11 00
to ;start; Serial port global control register initialisation ;first; One sample buffer only, ie 32
bits (I=16 Q=16)


      .start "text",809802h
      .start "servect1",809FC2h                ;location of INT3 int. vect
      .start "servect2",809FC6h                ;location of RINT int. vect
      .start "data" ,809900h
      .sect  "data" ;
A11   .FLOAT 0.5,0,-0.5,0,0,0,-0.5,0,0.5       ;ISP matrix, pre-calc Mathcad
A12   .FLOAT 0,0,0,0,0.5,0.5,0,0.5,0.5         ;BPSK figures
A21   .FLOAT 0,0,0,0,0.5,-0.5,0,-0.5,0.5
A22   .FLOAT 0.5,0,0.5,0,0,0,0.5,0,0.5
;     test   B1    .FLOAT 1,2,3
;     test   B2    .FLOAT 0.5,2,2
;     test   B3    .FLOAT 0.2,1,2              ; Test Data
A1ADDR .WORD  A11
A2ADDR .WORD  A12
A3ADDR .WORD  A21
A4ADDR .WORD  A22
;B1ADDR .WORD  B1
;B2ADDR .WORD  B2
;B3ADDR .WORD  B3
OUT1   .WORD  809B01h
OUT2   .WORD  809B04h
OUT3   .WORD  809B07h
OUT4   .WORD  809B0Ah
RVADDR .WORD  809FD0h
SGCR0  .SET   808040h                          ; serial port 0 global control
SPCX0  .SET   808042h                          ; FSX/DX/CLKX control reg.
SPCR0  .SET   808043h                          ; FSR/DR/CLKR control reg.
TGCR0  .SET   808020h                          ; timer 0 global control reg.
TCNT0  .SET   808024h                          ; timer 0 counter register
TPR0   .SET   808028h                          ; timer 0 period register
SINIT0 .WORD  0C3C0C44h                        ; 32 bit transfers, Rx ints on
SINIT1 .WORD  0111h
DATAIP .SET   80804Ch
DATAOP .SET   808048h
     .sect  "text"
     LDP    @SGCR0
     LDI    0h,R4
     STI    R4,@SGCR0
     LDI    @SINIT1,R7
     STI    R7,@SPCX0
     STI    R7,@SPCR0
     LDI    @SINIT0,R7
     STI    R7,@SGCR0
     STI    R4,@TCNT0
     .entry START
```

```
START
SLOOP
            LDI   3,R7
            LDI   2C1h,R6
            STI   R6,@TGCR0
            LDI   0B5h,R5
            STI   R5,@TPR0
LOOP
            LDI   00,IE
            LDI   00,IF                 ; clear interupts and flags
            OR    08h,IE                ; enable ext int 3
            IDLE                        ; data valid on EINT3,active low
LOOP1
            LDI   2C1h,R4
            STI   R4, @TGCR0            ; set tckl0 going
            LDI   00, IF                ; clear flags
            OR    28h,IE                ; enable RINT interupts,turn off EINT3
RXLOOP
            NOP
            LDP   809AFE,DP
            LDI   @DATAIP,AR0           ;load receive  ;data into ar0
            LDI   @DATAIP,AR1           ; ar1
            LSH   010h,AR0              ;shift left 16  ;discard upper 16
            LSH   0FF0h,AR0             ;shift back right 16
            LDI   AR0,R0                ;store in precision reg 0
            LSH   0FF0h,AR1             ;shift right 16
            LDI   AR1,R1
            FLOAT R0,R0                 ;convert to floating point
            FLOAT R1,R1
            ADDF3 R0,R1,R3              ;BPSK code +1=00, -1=11
            LDI   0FFh,R0               ;BPSK code 1/2 full scale convert
            FLOAT R0,R0                 ;BPSK code
            CMPF  R0,R3
            BZ    LOOPNEG
            LDI   DP,AR5
            STF   R3,*AR5
            ADDI  1,DP
            SUBI  1,R7
            BNZ   LOOP
; This section performs the multiplications,the received vectors are at;809AFE,809AFF,809B00,the result at OUT1(3)-4,decode at 808048 ;Remember data output is two 16 bit words
; First multiplication
            LDP   809900h,DP
            LDI   @A1ADDR,AR0
            LDI   @809AFEh,AR1
            LDI   @OUT1,AR2
            LDI   3,R3
LOOPI
            LDF   0,R0
            LDI   2,AR6
LOOPJ
            MPYF3 *AR0++,*AR1++,R1
            ADDF  R1,R0
            DB    AR6,LOOPJ
            STF   R0,*AR2
            ADDI  1,AR2
            LDI   @809AFEh,AR1
            SUBI  1,R3
            BNZ   LOOPI
```

```
; the result of the first multiplications are now at 809b01 809b02 809b03
; now calculate the euclidean norm
                LDF   @809B01h,R0
                LDF   @809B02h,R1
                LDF   @809B03h,R2
                CALL  FAST
                LDF   R0,R4
; Second multiplication
;               LDP   809904h,DP
                LDI   @A2ADDR,AR0
                LDI   @809AFEh,AR1
                LDI   @OUT2,AR2
                LDI   3,R3
LOOPK
                LDF   0,R0
                LDI   2,AR6
LOOPL
                MPYF3 *AR0++,*AR1++,R1
                ADDF  R1,R0
                DB    AR6,LOOPL
                STF   R0,*AR2
                ADDI  1,AR2
                LDI   @809AFEh,AR1
                SUBI  1,R3
                BNZ   LOOPK
; the result of the second multiplication is now at 809b04 809b05 809b06
; now calculate the euclidean norm
                LDF   @809B04h,R0
                LDF   @809B05h,R1
                LDF   @809B06h,R2
                CALL  FAST
                LDF   R0,R5
; Third multiplication
;               LDP   809907h,DP
                LDI   @A3ADDR,AR0
                LDI   @809AFEh,AR1
                LDI   @OUT3,AR2
                LDI   3,R3
LOOPM
                LDF   0,R0
                LDI   2,AR6
LOOPN
                MPYF3 *AR0++,*AR1++,R1
                ADDF  R1,R0
                DB    AR6,LOOPN
                STF   R0,*AR2
                ADDI  1,AR2
                LDI   @809AFEh,AR1
                SUBI  1,R3
                BNZ   LOOPM
; the result of the third multiplication is now at 809b04 809b05 809b06
; now calculate the euclidean norm
                LDF   @809B04h,R0
                LDF   @809B05h,R1
                LDF   @809B06h,R2
                CALL  FAST
                LDF   R0,R6
; Final multiplication
;               LDP   80990ah,DP
                LDI   @A4ADDR,AR0
```

```
                LDI   @809AFEh,AR1
                LDI   @OUT4,AR2
                LDI   3,R3
LOOPO
                LDF   0,R0
                LDI   2,AR6
LOOPP
                MPYF3  *AR0++,*AR1++,R1
                ADDF   R1,R0
                DB     AR6,LOOPP
                STF    R0,*AR2
                ADDI   1,AR2
                LDI    @809AFEh,AR1
                SUBI   1,R3
                BNZ    LOOPO
;result of the final multiplication is now at 809b0a 809b0b 809b0c
; now calculate the euclidean norm
                LDF    @809B0Ah,R0
                LDF    @809B0Bh,R1
                LDF    @809B0Ch,R2
                CALL   FAST
                LDF    R0,R7
;  norms are in r4,r5,r6,r7
                LDI    13h,R4
                FLOAT  R4,R4
                LDI    12h,R5
                FLOAT  R5,R5
                LDI    11h,R6
                FLOAT  R6,R6
                LDI    10h,R7
                FLOAT  R7,R7
;  now find the min and op the decoded data
                LDI    0000h,R0
                LDI    0001h,R1
                LDI    0100h,R2
                LDI    0101h,R3
                CMPF   R5,R4
                CALLLT NEG1
                CALL   NEG2
                CALL   NEG3
                CALL   NEG4
                B      SLOOP
;Return to start and wait next EINT3, external interupt 3. Now for the subroutines
NEG1   CMPF    R6,R4
                CALLGT NEG2
                CMPF   R7,R4
                CALLGT NEG2
                STI    R0,@DATAOP
                B      SLOOP
NEG2   CMPF    R6,R5
                CALLGT NEG3
                CMPF   R7,R5
                CALLGT NEG3
                STI    R1,@DATAOP
                B      SLOOP
NEG3   CMPF    R7,R6
                CALLGT NEG4
                STI    R2,@DATAOP
                B      SLOOP
NEG4   STI     R3,@DATAOP
```

```
                B     SLOOP
; Always return to start after outputting the decoded data Tx1 is in 16MSB and Tx2 is in
16LSB
EUCLID          MPYF   R0,R0
                MPYF   R1,R1
                MPYF   R2,R2                    ; square em up
                ADDF   R2,R1
                ADDF   R1,R0                    ; squared sum is now in r0
                LDF    R0,R3                    ; now find the square root
                PUSHF  R0
                POP    R1
                ASH    -24,R1
                ADDI   1,R1
                ASH    -1,R1
                NEGI   R1,R1
                ASH    24,R1
                PUSH   R1
                POPF   R1
                MPYF   0.5,R0
                MPYF3  R1,R2,R3
                MPYF   R0,R2
                SUBRF  1.5,R2
                MPYF   R2,R1
                RND    R1,R1
                MPYF3  R1,R2,R3
                MPYF   R0,R2
                SUBRF  1.5,R2
                MPYF   R2,R1
                RND    R1,R1
                MPYF3  R1,R2,R3
                MPYF   R0,R2
                SUBRF  1.5,R2
                MPYF   R2,R1
                RND    R1,R1
                MPYF3  R1,R2,R3
                MPYF   R0,R2
                SUBRF  1.5,R2
                MPYF   R2,R1
                RND    R1,R1
                MPYF3  R1,R2,R3
                MPYF   R0,R2
                SUBRF  1.5,R2
                MPYF   R2,R1
                RND    R1,R0
    MPYF   R3,R0
    FAST    MPYF   R0,R0
    MPYF   R1,R1
    MPYF   R2,R2                                ; square em up
    ADDF   R2,R1
    ADDF   R1,R0                                ; squared sum is now in r0
    RETS
LOOPNEG NEGF   R3,R3
    RETS
.sect  "servect1"                               ; Ext int 3 service vector
    B     LOOP1
    .sect  "servect2"                           ; RINT service vector
    B     RXLOOP
```

Appendix 1b

Disk Kernal Communications Program

```
                .start "vectors", 0x809FC1
                .start "kernel", vectors-0xAB ; Use size report from DSK3A
                .start "sstack",0x809F00      ; output to pack to end of RAM
                .entry START
                .sect   "sstack"
        stack:  .word   stack-1
MMRBASE         .word   0x00808000
PRD0            .word   0x0000A000
PRD1            .word   0x0000A060
TSTART          .word   0x000003C3
START           ldp     @START          ; set up stack and other params
                ldi     @stack, SP      ;
                ldi     @MMRBASE,AR0    ;
                ldi     3 , R0          ; HALT timers
                sti     R0, *+AR0 (0x20)        ;
                sti     R0, *+AR0 (0x30)        ;
                sti     R0, *+AR0 (0x24)        ; Init count registers
                sti     R0, *+AR0 (0x34)        ;
                ldi     @PRD0, R0               ; Init periods
                sti     R0, *=AR0 (0x38)        ;
                ldi     @PRD1, R0               ;
                sti     R0, *AR0 (0x38)         ;
                ldi     @TSTART, R0             ; Start timers
                sti     R0, *+AR0 (0x30)        ;
                b       spin0                   ;
S0_xdata        .set    0x808048                ; SP 0 Data transmit
S0_rdata.       .set    0x80804C                ; SP 0 Data recieve
                .word   0x00320C31              ;
                .word   0x00320C31
XSTEP           or      0x40, IF                ; set XINT1
XFUNF           or      0xC4, IE                ; set EXINT1
                sti     IE,@_freerun
                ldi     @CPUCTXT,AR0
                ldi     AR0, AR1                ;
                addi    1,AR1                   ;
                ldi     @S0_rdata,R0            ; Clear under/over conditions before exit
                ldi     0, R0                   ; 0 ensures low bits durning Sp recovery
                sti     R0,@S0_xdata            ; XSR resends – should all be zero
                ldf     *AR0++(IR0), R0         ;       load floats (exponents)
ldf     *AR1++(IR0), R1         +       ldi     *AR0++(IR0), R0         ;load longs
ldf     *AR0++(IR0), R2         +       ldi     *AR1++(IR0), R1
ldf     *AR1++(IR0), R3         +       ldi     *AR0++(IR0), R2
ldf     *AR0++(IR0), R4         +       ldi     *AR1++(IR0), R3
ldf     *AR1++(IR0), R5         +       ldi     *AR0++(IR0), R4
ldf     *AR0++(IR0), R6         +       ldi     *AR1++(IR0), R5
ldf     *AR1++(IR0), R7         +       ldi     *AR0++(IR0), R6
ldi     *AR1++(IR0), R7         ;
ldi     @_AR0, AR0              : load ARx ldi   @IR0, IR0      ;
ldi     @_AR1, AR1              : ldi   @IR1, IR1               ;
ldi     @_AR2, AR2              : or    @_IF, IF                ; CPU interrupt flags
ldi     @_AR3, AR3              : ldi   @_IOF, IOF              ; 10 flags
ldi     @_AR4, AR4              : ldi   @_RS, RS                ; Repeat start
ldi     @_AR5, AR5              : ldi   @_RE, RE                ; Repeat end
ldi     @_AR6, AR6              : ldi   @RC, RC                 ; Repeat counter
ldi     @_AR7, AR7              : ldi   @_BK, BK                ; Block size
                ldi     @_SP, SP                ; get user SP
                ldi     @_PC, R5
```

```
                andn    0x4, IF                 ; Clear/Poll INT2 before SSTEP or RUNF
                tstb    4, IF
                bnz     $-3
                ldiu    @_ST, ST                ; restore status
                or      @_IE, IE
                BUD     R5
                or      2000h, ST               ; turn on INT's
                ldiu    @_R5, R5
                ldiu    @_DP, DP                ; restore DP
        XHALT   pop     AR1                     ; restore original registers before save
                pop     AR0
                pop     IR1
                opo     R-
                pop     DP
                pop     ST
SSTEP           push    DP                      ; temp storage of user DP
                ldp     @_ST                    ; DP for kernal
                sti     ST, @_ST                ; store ST
                sti     IR0, @_IR0
                pop     IR0                     ; save user SP
                sti     IR0, @_DP
                pop     IR0                     ; save user PC
                sti     IR0, @_PC
                sti     SP, @_SP                ; save user SP
                sti     BK, @_BK                ; Block size
                sti     IE, @_IE                ; Interal int enable
                sti     IF, @_IF                ; CPU interrupt flags
                sti     IOF, @_IOF              ;10 flags
                sti     RS, @_RS                ; Repeat start
                sti     RE, @_RE                ; Repeat end
                sti     RC, @_RC                ; Repeat counter
                sti     IR0, @_IR0              ; Keep everything <- IR0 saved previously
                sti     IR1, @_IR1
                sti     AR0, @_AR0
                sti     AR1, @_AR1
                ldi     @CPUCTXT,AR0
                ldi     AR0, AR1
                addi    1, AR1
                ldi     2, IR0                  ;
stf     R0, *AR0++ (IR0)        Store floats +  sti     R0, *AR0++ (IR0)        ; Store longs
stf     R1, *AR1++ (IR0)        +               sti     R1, *AR1++ (IR0)
stf     R2, *AR0++ (IR0)        +               sti     R2, *AR0++ (IR0)
stf     R3, *AR1++ (IR0)        +               sti     R3, *AR1++ (IR0)
stf     R4, *AR0++ (IR0)        +               sti     R4, *AR0++ (IR0)
stf     R5, *AR1++ (IR0)        +               sti     R5, *AR1++ (IR0)
stf     R6, *AR0++ (IR0)        +               sti     R6, *AR0++ (IR0)
stf     R7, *AR1++ (IR0)        +               sti     R7, *AR1++ (IR0)
sti     AR2, @_AR2              ; AR0 & AR1 Already saved
sti     AR3, @_AR3              ; sti   AR4, @_AR4              ;
sti     AR5, @_AR5              ; sti   AR6, @_AR6              ;
sti     AR7, @_AR7              ; ldi   0, R0   ; Freerun = 0 indicates HALT (spin0)
                sti     R0, @_FREERUN           ;
TRAP_AK         call    W_HOST                  ; Send ACKNOWLEDGE (zero) to host
                b       spin0                   ; <- Branch is removed (spin0 in inline)
        spin0   or      4, IE                   ; Enable DSK31 HPI interrupt
                and     4, IE                   ; Shut down all interrupts except host
                b       spin0                   ;
                S0xdata .word 0
        GIE     .set 0x2000
context
```

```
_F0     .word   0       ; R0    +       _R0     .word   0       ; F0
_F1     .word   0       ; R1    +       _R1     .word   0       ; F1
_F2     .word   0       ; R2    +       _R2     .word   0       ; F2
_F3     .word   0       ; R3    +       _R3     .word   0       ; F3
_F4     .word   0       ; R4    +       _R4     .word   0       ; F4
_F5     .word   0       ; R5    +       _R5     .word   0       ; F5
_F6     .word   0       ; R6    +       _R6     .word   0       ; F6
_F7     .word   0       ; R7    +       _R7     .word   0       ; F7

_AR0    .word   0       ; AR0   +       _DP     .word   0       ; Data page
_AR1    .word   0       ; AR1   +       _IR0    .word   0       ; Index register 0
_AR2    .word   0       ; AR2   +       _IR1    .word   0       ; Index register 1
_AR3    .word   0       ; AR3   +       _BK     .word   0       ; Block size
_AR4    .word   0       ; AR4   +       _SP     .word   stack-1 ; Stack pointer
_AR5    .word   0       ; AR5   +       _ST     .word   0       ; Status
_AR6    .word   0       ; AR6   +       _IE     .word   0       ; Internal int enable
_AR7    .word   0       ; AR7   +       _IF     .word   0       ; CPU interrupt flags
_IOF    .word   0       ; I/O flags
_RS     .word   0       ; Repeat start
_RE     .word   0       ; Repeat end
_RC     .word   0       ; Repeat counter
_PC     .word   0       ; program counter
_FREERUN.word0          ; 1 = DSK is freerunning, 0 = DSK is HALT'ed
CPUCTXT         .word   context
INTx    ;       maxspeed
                push    ST                      ; Push ISR variables
                push    DP
                push    R0                      ; NOTE: A Halt command pops these
                push    IR1                     ;       values followed by a full save
                push    AR0
                push    AR1
                ldp     @JUMP                   ; Get address of command from JUMP table
                ldi     @S0_xdata, R0           ; Put a zero in the DXR making startup
                sti     R0, @S0xdata
                ldi     0, R0
                sti     R0, @SO_xdata
                tstb    4, IF                   ; Get here by driving INT2 low
                bz      SR2                     ; Make sure INT2 is active
                call    R_HOST                  ; R0==command
                ldi     R0, AR1
                addi    @JUMP, AR1
                ldi     *AR1, AR1
                b       AR1                     ; execute command
COMN            call    R_HOST
                ldi     R0, AR1                 ; data packet length
                call    R_HOST
                ldi     R0, IAR0                ; source address
                call    R_HOST
                ldi     R0, IR1                 ; source index
                subi    1, AR1                  ;
                rets                            ;
SR2             ldi     07F00h, AR0             ; Dummy non-HPI read releases READY
                ldi     *AR0, AR0               ;
                pop     AR1                     ; restore ISR variables
                pop     AR0
                pop     IR1
                pop     R0
                pop     DP
        ;       andn    0x4, IF
        ;       or      4, IE
```

```
                pop     ST
                reti                            ; return to original code
                .sect   "vectors"
INT0            b       $                       ; 0x809FC1      0x001
INT1            b       $                       ; 0x809FC2      0x002
INT2            b       INTx                    ; 0x809FC3      0x004   <- HPI
INT3            b       $                       ; 0x809FC4      0x008
XINT0           b       $                       ; 0x809FC5      0x010
RINT0           b       $                       ; 0x809FC6      0x020
XINT1           b       SSTEP                   ; 0x809FC7      0x040 <- SSTEP
RINT1           b       SSTEP; TRAPFIX          ; 0x809FC8      0x080   <- ETRAP
0x74000008
TINT0           b       $                       ; 0x809FC9      0x100
TINT1           b       $                       ; 0x809FCA      0x200
DINT            b       $                       ; 0x809FCB      0x400
W_HOST          push    AR0     ; HPI address  R_HOST          push    AR0     ; HPI address
push    AR1                     ; loop counter          push    AR1             ; loop
counter
push    ST                      ; Keep flags            push    ST
push    DP                      ;                       push    R1              ; temp
register
ldp     WSCOUNT                 ;                       ldi     0xF000, AR0     ;
ldi     0xF000, AR0             ; HPI                   ldi     3, AR1
ldi     @WSCOUNT, AR1           ;                       lsh     _8, R0
sti     R0, *AR0++(16) ; Store lsbs to HPI              ldi     *AR0++, R1
lsh     @WSHIFT, R0 ; shift to next lsbs                lsh     24, R1
db      AR1, WH                 ; loop until done       or      R1, R0
pop     DP                                              db      AR1, RH pop     R1
b       COMNHST                 ;
COMNHST         pop     ST                      ; Next 4 opcodes common to
                                                 W_HOST/R_HOST
                pop     AR1
                pop     AR0
                rets
XWRIT           call    COMN            ;
XW1             call    R_HOST
                sti     R0, *AR0++(IR1)
                db      AR1, XW1
                b       SR2
XREAD           call    COMN
XR1             ldi     *AR0++(IR1), R0
                call    W_HOST
                db      AR1, XR1
                b       SR2
;               .start  "JMPTBL", 0x809FF4
;               .sect   "JMPTBL"
JUMP            .word   JUMP                    ; 0x809FF4      Jump table base address
                .word   XWRIT ;1        ; 0x809FF5      for DSK3 routines
                .word   XREAD ; 2       ; 0x809FF6
                .word   XCTXT ; 3       ; 0x809FF7
                .word   XRUNF ; 4       ; 0x809FF8
                .word   XSTEP           ; 5     ; 0x809FF9
                .word   XHALT ; 6       ; 0x809FFA
                .word   W_HOST          ; 7     ; 0x809FFB
                .word   R_HOST          ; 8     ; 0x809FFC
                .word   spin0           ; 9     ; 0x809FFD
WSCOUNT         .word   7                       ; 0x809FFE
WSHIFT          .word   -4                      ; 0x809FFF buswidth parameters
```

MISSING PAGES ARE UNAVAILABLE

Some s ı a copy of
the resı

2/3 MISSING

i) Inpu

netlist
*
*
>IC1
>IC2 out:14

B1	+out:11	-out:0	v=10		
B2	+out:2	-out:0	v=10		
Vs	+out:15	-out:17	v=100m	Ex=Sine	Fr=1k
R1	p1:0	p2:1	v=22k		
R2	p1:0	p2:5	v=1M		
R3	p1:7	p2:8	v=600		
R4	p1:0	p2:8	v=1k		
R5	p1:0	p2:12	v=100k		
R6	p1:12	p2:11	v=100k		
R7	p1:10	p2:13	v=10k		
R8	p1:14	p2:13	v=100k		
C1	p1:1	p2:17	v=100u		
C2	p1:3	p2:4	v=2.2u		
C3	p1:5	p2:0	v=47u		
C4	p1:6	p2:7	v=10u		
C5	p1:8	p2:0	v=.1u		
C6	p1:12	p2:9	v=10u		
C7	p1:10	p2:0	v=100u		
C8	p1:14	p2:16	v=100u		
C9	p1:4	p2:6	v=4.7n		

* Probe nodes on 8 and 16
* Input between 15 and 17
* All references to node 0 at 0v
* Use frequency analysis on pull down menu and then plot Bode

The Bode plot shows the gain falling away nicely with the amplifier unconditionally stable at the conditions presented. The C.V.S.D. model did not give any sensible outputs, the amplifier would always ramp to rail and stay there and so is not included.

ii) Level shifting stage (drive to modulators)

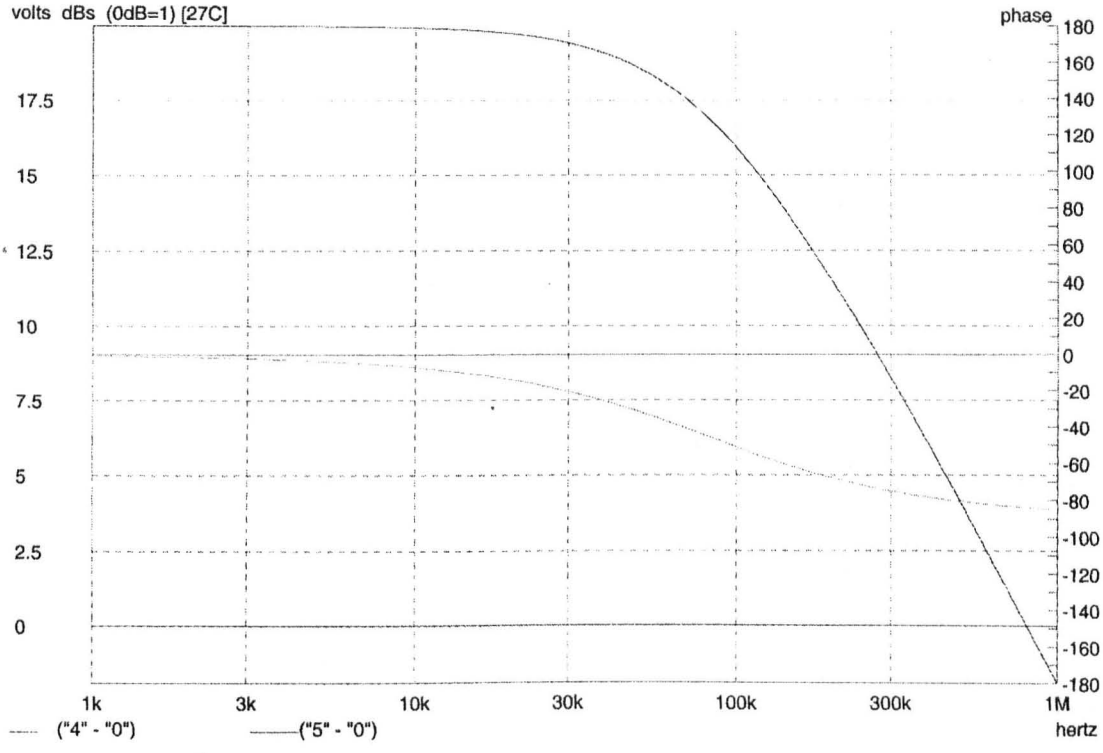
netlist

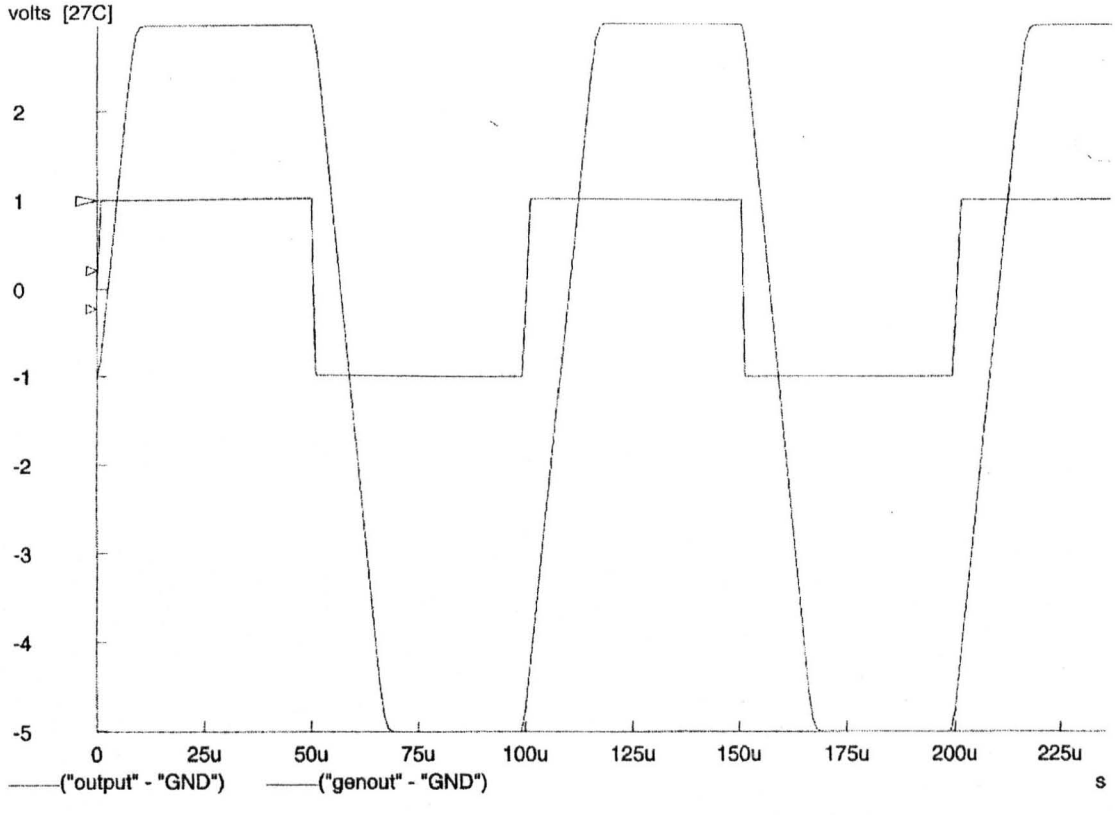
```
*       Paul4b.cmp
*
>IC2    uA741       +in:5       +v:4          -in:6          -v:0          out:7
B1      +out:4      -out:0      v=12
B2      +out:       -out:0      v=10
Vs      +out:1      -out:0      v=5    Ex=Square    Fr=4k
R1      p1:1        p2:2        v=47k
R2      p1:2        p2:0        v=47k
R3      p1:4        p2:3        v=150
R4      p1:3        p2:2        v=10k
R5      p1:3        p2:0        v=150
R6      p1:13       p2:5        v=10k
R7      p1:6        p2:0        v=47k
R8      p1:6        p2:7        v=27k
C1      p1:5        p2:0        v=3.3n
C2      p1:3        p2:7        v=3.3n


*       probe node on node 7
*       input on node 1
*       reference and ground node 0
```

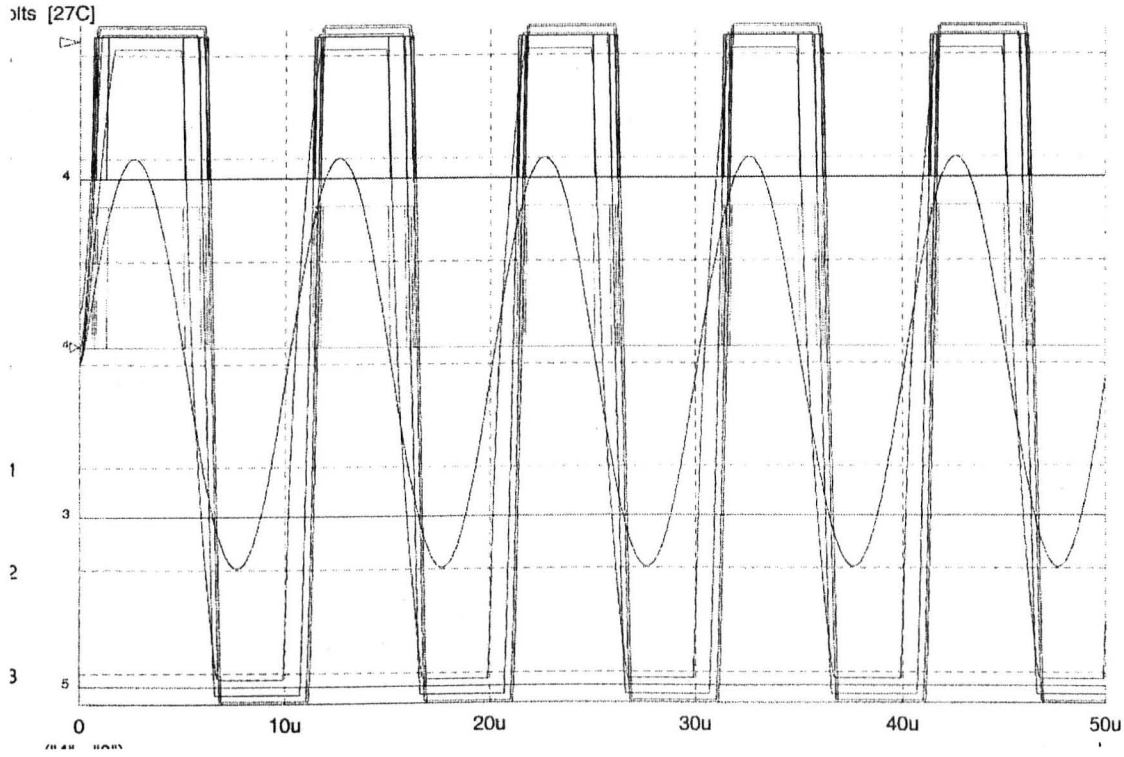
The diagram shows the input being overdriven, but still working. The Burr Brown analogue to digital converter was supplied with simulation data but again did not model well in the circuit configuration. The netlist is reproduced below and the timing / output diagram is the third plate.

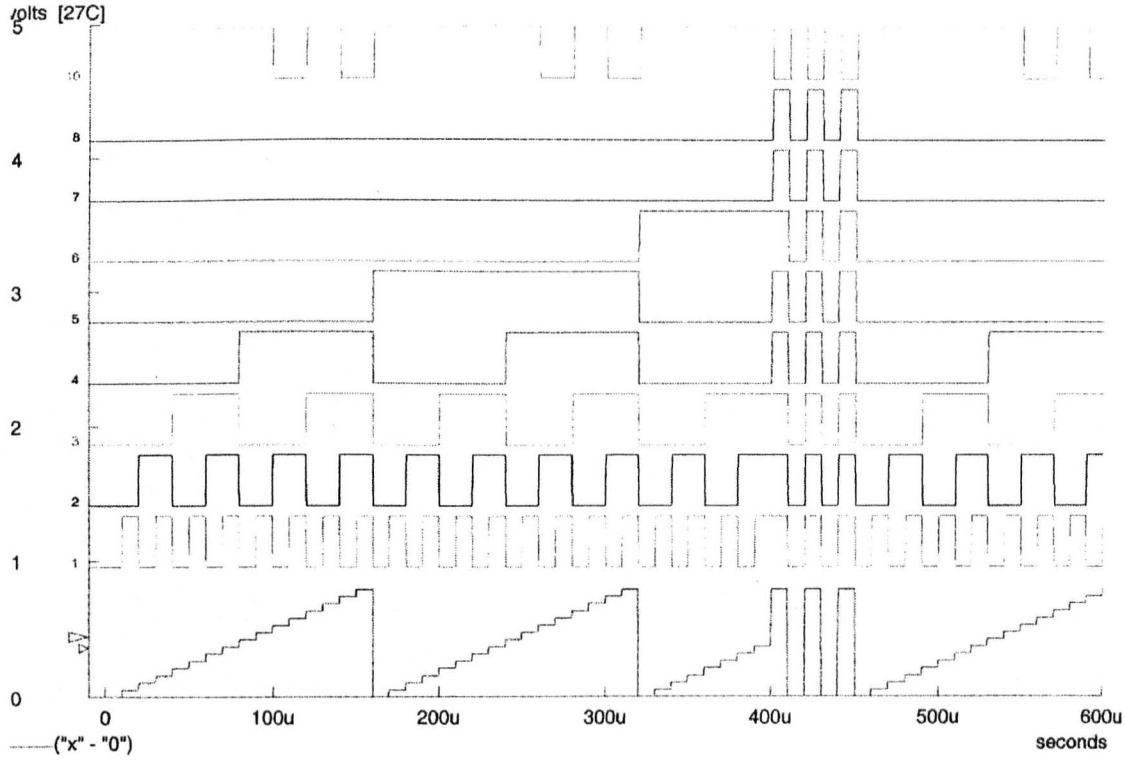
iii) Analogue to digital converter netlist

```
*       Paul4c.cmp
*
>IC1    101define   p1:1   p2:2   p5:3   p6:4   p7:50  p8:0
>IC1    101define   p9:5   p10:6  p11:6  p12:5  p13:7  p14:7  p15:10
>IC1    101define   p16:7  p18:0  p20:8  p21:9  p22:0  p23:0
>IC1    101define   p24:0  p25:0  p26:11 p27:12 p28:0
B1      +out:5      -out:0       v=5
B2      +out:4      -out:0       v=-5
Vs      +out:2      -out:0       v=2.5          Ex=Step          Fr=100
R1      p1:4        p2:3         v=10
R2      p1:9        p2:0         v=1k
R3      p1:8        p2:0         v=1k
C1      p1:1        p2:0         v=10u
C2      p1:4        p2:0         v=10u
C3      p1:5        p2:0         v=10u
C4      p1:12       p2:0         v=0.1u
C5      p1:11       p2:0         v=10u
```


volts dBs (0dB=1) [27C]
phase
17.5
15
12.5
10
7.5
5
2.5
0
180
160
140
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
1k
3k
10k
30k
100k
300k
1M
("4" - "0")
("5" - "0")
hertz

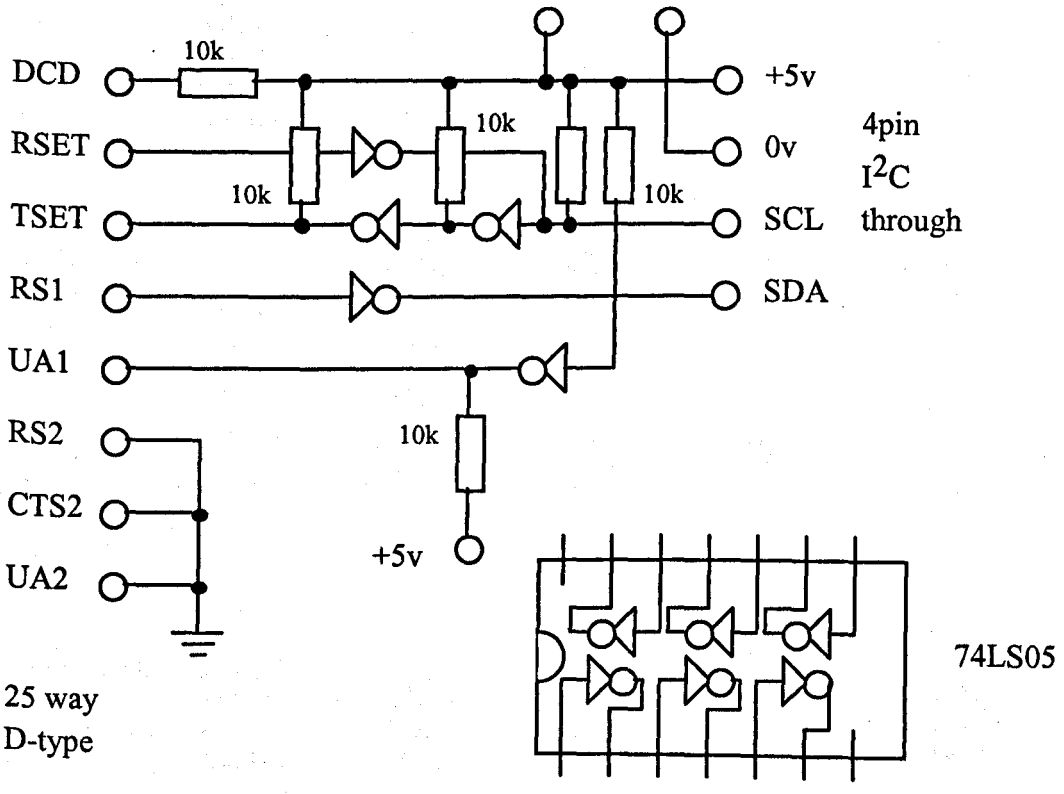

volts [27C]
2
1
0
-1
-2
-3
-4
-5
0
25u
50u
75u
100u
125u
150u
175u
200u
225u
("output" - "GND")
("genout" - "GND")
s


olts [27C]
4
1
3
2
3
5
0
10u
20u
30u
40u
50u


olts [27C]
5
4
3
2
1
0
0
100u
200u
300u
400u
500u
600u
seconds
("x" - "0")

Appendix 5

I^2C Interface for P.C. Printer port



Program to drive I^2C via the parallel port.

Program to drive the I2C bus via the printer port

```
;   Kbd clock on interrupt INT0 P1.5
;   Kbd data on pin          P0.2
IICADD    .equ 088h      ; our i2c slave address
MAXBYTES  .equ 1         ; max bytes to recv or trans
rcvdat    .equ 04h       ; I2C received data buffer
xmtdat    .equ 06h       ; I2C transmitter buffer
STACK     .equ 08h
flags     .equ 020h      ; byte used as flags
noack     .equ (flags-20h)  ; I2C flags.0, ...1, ...2, etc
recvd     .equ (flags-20h)+1 ;
sent_flag .equ (flags-20h)+2 ;
i2c_busy  .equ (flags-20h)+3 ;
Cntrl     .equ (flags-20h)+8 ; Control key flag
Shift     .equ (flags-20h)+9 ; Shift  key flag
bitcnt    .equ flags+2
bytecnt   .equ flags+3
adrrcvd   .equ flags+4
rwflag    .equ (adrrcvd-20h)*8  ; adrrcvd.0
```

```
tick      .equ 025h       ; count 10mS ticks to give 1sec tick
i2ctime   .equ 027h       ; i2c timeout - used on slow i2c bus
NBits     .equ 29h        ; # bits read so far
NBytes    .equ NBits+1    ; # bytes in buffer
lastkey   .equ NBytes+1   ; last key was?
keytemp   .equ lastkey+1  ; used to build the key bit by bit
keybuff   .equ keytemp+1  ; store the chars here
INMAX     .equ 8          ; Size of keyboard buffer
KEYCLK    .equ p1.5       ; keyboard clock signal on ext int 0
KEYDAT    .equ 82h        ; keyboard data line
EDGEINT   .equ 08ah
.org 0            ; reset vector
.org 0003h        ; external Interrupt 0
.org 0bh          ; counter/timer 0
.org 013h         ; ext int 1
.org 01bh         ; timer 1 - i2c timeout
.org 023h         ; i2c interupt
.org 48h
ajmp $            ; main routine waiting for key presses
.org 50h
start:
      mov r0,#0ffh       ; power supply settling time
      mov r1,#0ffh
      mov r2,#04h
dly1: djnz r0,$
      djnz r1,dly1
      djnz r2,dly1
reset:
      mov p0,#0ffh
      mov p1,#0ffh
      mov p3,#0ffh
mov sp,#STACK       ; initialise stack pointer
      setb EDGEINT        ; make ext int 0 edge activated
clr Shift         ; clear keyboard shift flag
mov r0,#NBits       ; clear the input buffers
      mov r1,#10h
      mov acc,#0
clrlp: mov @r0,a          ; do the clearing
      inc r0
      djnz r1,clrlp
;     mov xmtdat,#'.'     ; transmitt buffer filled with
;     mov xmtdat+1,#0ffh  ; $ff when empty
mov flags,#0
      mov i2ctime,#0
restart:
mov  I2CFG,#80h+CTVAL ;enable slave functions
      mov  I2CON,#BIDLE   ;place in idle state
mov ien0,#91h       ;enable external & IIC interrupts
main:
      mov a,NBytes
      jz empty          ; if data in keybuff copy to i2c xmt buffer
mov ien0,#0h        ;disable all interrupts temporarily
      mov xmtdat,keybuff
      mov NBytes,#0       ; clear keyboard buffer full flag
      mov ien0,#91h       ;enable external & IIC interrupts
empty:
jnb recvd,notread   ; recvd flag tells 751 to clear I2C xmt buffer
mov acc,xmtdat      ; when i2c master sucessfully reads the data
      orl a,#80h          ; from the 87c751, the master writes any data
      mov xmtdat,acc      ; back which will set the MSB of the data
```

```
                    ; buffer. This is necessary to syncronise
                    ; the two processors.
      clr recvd          ; reset i2c received flag
notread:
mov a,lastkey      ; detect alt key for special functions
      clr c
      subb a,#11h
      jnz notalt

      nop              ; alt code goes here
notalt:
      mov a,NBytes       ;
      clr c
      subb a,#INMAX      ; limit the input buffer to INMAX
      jc notdone
; if data is buffered then buffer exceed
                    ; code goes here
notdone:
sjmp main          ; go back to start
kbd:
      push psw           ; save regs during ISR
      push acc
mov acc,NBits      ; NBits = the bit number next expected
                    ; from the keyboard
      cjne a,#0,bit1_8   ; if not bit 0 then maybe bit 1 to 8
bit0:                ; discard bit 0 - Start bit
      sjmp bump
bit1_8:
      cjne a,#9,$+3      ; CY flag is set if acc < 9
      jnc bit9
mov c,KEYDAT       ; read data for keyboard data line
      mov a,keytemp      ; data arrives least sig bit 1st
      rr a             ; hence old value is rotated and new
      mov a.7,c          ; bit is or'ed to the msb
      mov keytemp,acc
      sjmp bump
bit9:
      cjne a,#9,bit10
sjmp bump          ; parity check code would go here
bit10:
      mov acc,keytemp    ; get next key
cjne a,#12h,notls  ; is it the left shift char?
mov acc,lastkey
      cjne a,#0f0h,makels ; if last key was $f0 then shift is released
clr Shift          ; Next keys will be unshifted
      mov lastkey,#12h   ; Copy left shift key to last key
      sjmp tidy
makels:
      setb Shift         ; Next Keys will be shifted
      mov lastkey,#12h   ; Copy left shift key to last key
      sjmp tidy
notls:
;     mov acc,keytemp    ; get next key
      cjne a,#14h,notctrl ; is it a control char?
nop              ; control state goes here
      sjmp tidy
notctrl:
cjne a,#0f0h,notbreak ; is current key is $f0 then break code
mov lastkey,a      ; if so record break code in last key
      sjmp tidy          ; but don't store in the buffer
```

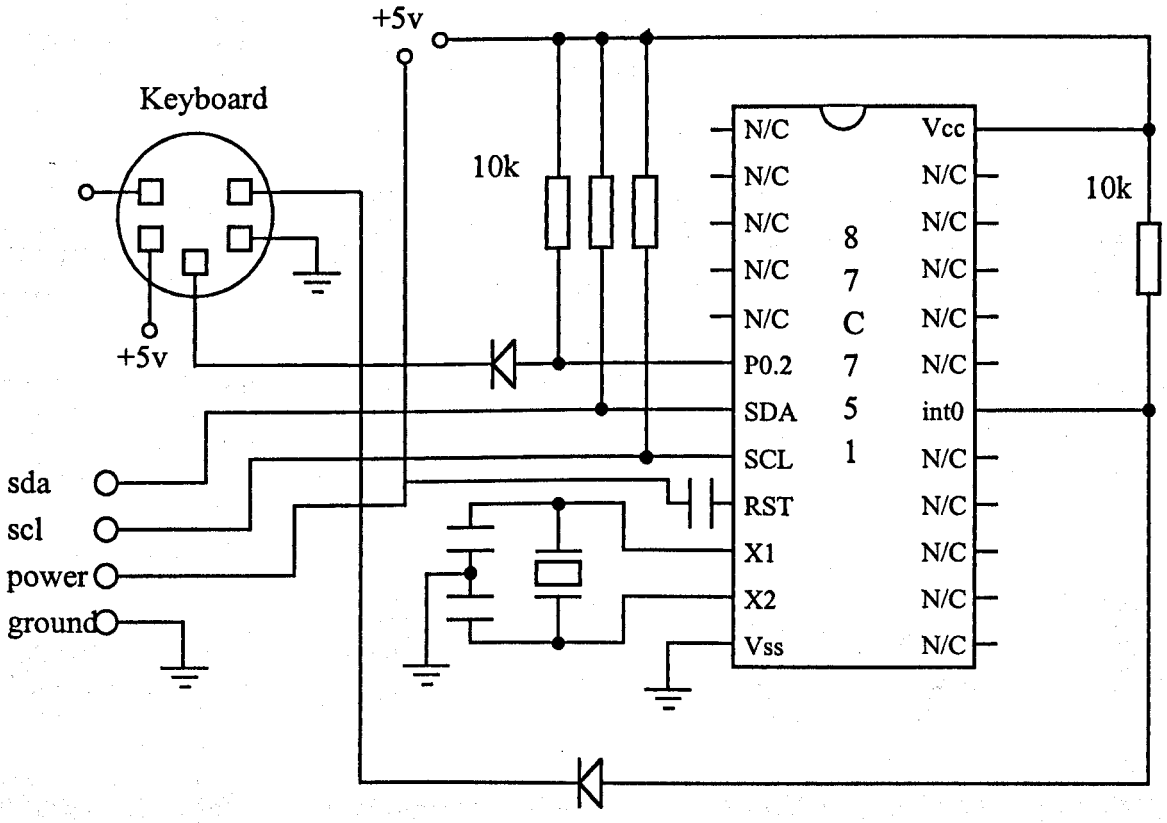
```
notbreak:
mov acc,lastkey     ; if last key was $f0 then
     cjne a,#0f0h,not_f0  ; ignore the next scan code
mov lastkey,#0      ; which is a break code
     sjmp tidy
not_f0:
#ifdef buffered
push 0              ; r0 will be used as an indirect pointer
                    ; so save it
     mov acc,#keybuff    ; copy data into keyboard
     add a,NBytes
     mov r0,a
mov a,keytemp       ; get current key
     mov lastkey,a       ; & copy to lastkey
push dph            ; dp required to point to translation
     push dpl            ; tables. Since in ISR save dp contents
jb Shift,shifted    ; if in the unshifted state use the unshift table
     mov dptr,#unshift
     sjmp skip1
shifted:
     mov dptr,#shift     ; else use the shift table
skip1 :
     movc a,@a+dptr      ; translate the char in Acc to Ascii
pop dpl             ; retsore the data pointer
     pop dph
cjne a,#0,Not0      ; If data is zero discard
; sjmp NoSave       ; discard code goes here
Not0:
     mov r0,#keybuff     ; Save Ascii value into keyboard buffer
     mov @r0,a           ; buffered keyboard entry
     inc NBytes
NoSave
     pop 0               ; restore r0
#endif
#define unbuffered 1
#ifdef unbuffered
     mov a,keytemp       ; get current key
     mov lastkey,a       ; & copy to lastkey
push dph            ; dp required to point to translation
     push dpl            ; tables. Since in ISR save dp contents
jb Shift,shifted    ; if in the unshifted state use the unshift table
     mov dptr,#unshift
     sjmp skip1
shifted:
     mov dptr,#shift     ; else use the shift table
skip1 :
     movc a,@a+dptr      ; translate the char in Acc to Ascii
pop dpl             ; retsore the data pointer
     pop dph
cjne a,#0,Not0      ; If data is zero discard
; sjmp tidy         ; discard code goes here
Not0:
     mov keybuff,a       ; store in kbd buffer
     mov NBytes,#1       ; mark byte read
tidy:
     mov NBits,#0        ; clear flags ready for next keyboard char
     mov keytemp,#0
     sjmp intdone
bump:
     inc NBits           ; inc the number of bits read so far
```

```
intdone:
      pop acc
      pop psw
      reti
i2cint:                 ; i2c interrupt entry point
      setb i2c_busy     ; semaphore on xmtdata buffer
push psw          ; save registers used in ISR
      push acc
      push 0            ; R0 no bank switching
clr  EI2          ; make i2c ISR interruptable
      acall clrint      ; execute a reti
slave:
      mov i2ctime,#3    ; set up i2c timeout watchdog 30 mS
mov I2CON,#BCARL+BCSTP+BCSTR+BCXA
                  ; clear start status
jnb ATN,$         ; wait for next data bit
      mov bitcnt,#7
acall recvb2      ; get remainder of slave address
      mov adrrcvd,a
      clr a.0           ; mask read/write bit to check address
      cjne a,#IICADD,goidle ; idle again if not for us
jb rwflag,read    ; test for read or write
mov r0,#rcvdat    ; r0 points to data bufferer
      mov bytecnt,#MAXBYTES
rcvloop:
      acall sendack     ; acknowledge the address
      acall rcvbyte     ; wait for the next data byte
      jnb DRDY,exitwr   ; end of frame
      mov @r0,a
      inc r0
      djnz bytecnt,rcvloop
acall sendack     ; ack last byte
      acall rcvbyte     ; get but discard next one
      mov I2DAT,#80h    ; send neg ack
      jnb ATN,$         ; wait till gone
exitwr:
setb recvd
      sjmp msgend
read:
      mov r0,#xmtdat    ; r0 points to data buffer
      mov bytecnt,#MAXBYTES
      acall sendack     ; acknowledge address
txloop:
      mov a,@r0         ; get next data byte
      inc r0            ; bump buffer pointer
      acall xmitbyte    ; transmit the byte to the I2C
      jb noack,exitrd   ; if not acknowledged the exit
      djnz bytecnt,txloop
exitrd: sjmp msgend
msgend:
      jnb ATN,$         ; wait for stop or repeated start
      jb STR,slave      ; if repeat start do again
goidle:
      mov i2ctime,#0    ; stop i2c timeout
      mov I2CON,#BCSTP+BCXA+BCDR+BCARL+BIDLE
pop 0             ;restore state before i2c ISR
      pop acc
      pop psw
setb EI2
      setb sent_flag    ;flag to say data has been sent
```

```
      clr i2c_busy        ;flag denotes exiting I2C routine
ret
xmitbyte:                 ; I2C data transmit  ( enter with data in acc )
      mov bitcnt,#8
xmitbit:
      mov I2DAT,a
      rl a
      jnb ATN,$
      djnz bitcnt,xmitbit
      mov I2CON,#BCDR+BCXA ; switch to rcv mode
      jnb ATN,$           ; wait for ack
      mov flags,I2DAT     ; save ack bit
      ret
rdack:                    ; rdack   - receives data byte then sends ack
      acall rcvbyte       ; I2C data recv routine  - data returned in a
sendack:
      mov I2DAT,#0        ; I2C ack = data low when clock pulsed high
      jnb ATN,$
      ret
rcvbyte:                  ; rcvbyte receives byte - data returned in Acc
      mov bitcnt,#8
recvb2: clr a
rbit:  orl a,I2DAT
      rl a
      jnb ATN,$
      jnb DRDY,rbex       ; exit if not a data bit
      djnz bitcnt,rbit
mov c,RDAT          ; get last bit - do not clear ATN
      rlc a               ; shift into byte
rbex:
      ret
timerI:
      mov ien0,#0               fixup:
      setb CLRTI                mov I2CFG,#0        ; turn off i2c
mov I2CON,#BCXA+BCDR+BCARL+BCSTR+BCSTP
acall clrint ajmp reset           ; restart program
clrint: reti
badint: reti
#include attable.h
.end
```

Appendix 6

I^2C Interface using the 8XC750/751 and a standard keyboard



PC/AT keyboard gives a start bit, 8 data bits with the LSB transmitted first and a stop bit. When a key is pressed the keyboard transmits a make code and when the key is released a break code. These are basically the same but the break has a 0F0 (hex) on the front. The ASCII keyboards are better because control/shift keys give unique codes, on the PC/AT they give two codes which need to be interpreted together, care is needed. The software reproduced overleaf will capture and convert to ASCII the data from the PC keyboard, the look up tables will need modification depending upon the keyboard type, the code is then output as an I2C message

Software to drive 751 +keyboard

```
BTIR    EQU   10h          ; TIRUN bit.
BMRQ    EQU   40h          ; MASTRQ bit.
BCXA    EQU   80h          ; CXA bit.
BIDLE   EQU   40h          ; IDLE bit.
BCDR    EQU   20h          ; CDR bit.
BCARL   EQU   10h          ; CARL bit.
BCSTR   EQU   08h          ; CSTR bit.
BCSTP   EQU   04h          ; CSTP bit.
BXSTR   EQU   02h          ; XSTR bit.
BXSTP   EQU   01h          ; XSTP bit
SGO     EQU   10h      ; Started Slave message processing.
SRCVD   EQU   11h      ; as a slave, received a new message
SRLNG   EQU   12h      ; received as slave a message which is
STXED   EQU   13h
SRERR   EQU   14h      ; bus error detected when operating as a slave.
MGO     EQU   20h      ; Started Master message processing.
MRCVED  EQU   21h
MTXED   EQU
MTXNAK  EQU
MTXNOSLV EQU   TIMOUT   EQU   30h      ; TIMERI Timed out.
NOTSTR  EQU   32h      ; Master did not recognize Start.
MASCMD  DATA  20h
SUBADD  BIT   MASCMD.0
RPSTRT  BIT   MASCMD.1
SETMRQ  BIT   MASCMD.2
DSEG  AT 24h
MSGSTAT: DS   1        ; I2C communications status.
MYADDR:  DS   1        ; Address of this I2C node.
DESTADRW: DS  1        ; Destination address + R/W  (for Master).
DESSUBAD: DS  1        ; Destination subaddress.
MASTCNT: DS   TITOCNT: DS   1        ; Timer I bus watchdog timeouts counter.
StackSave: DS   MasBuf:  DS   4        ; Master receive/transmit buffer, 8 bytes.
SRcvBuf: DS   4        ; Slave receive buffer, 8 bytes.
STxBuf:  DS   4        ; Slave transmit buffer, 8 bytes.
RBufLen  EQU  4h       ; The length of SRcvBuf
TogLED   BIT  P1.
ErrLED   BIT  P1.1     ; Error indication.
OnLED    BIT  P1.3     ;
APPFLAGS  DATA  21h
 TRQFLAG  BIT  APPFLAGS.0
SErrFLAG BIT  APPFLAGS.1
FAILCNT: DS   1
TOGCNT:  DS
CSEG
AJMP  Reset            ;Reset vector at address 0.
ORG   1Bh              ; Timer I (I2C timeout) interrupt.
TimerI:  SETB  CLRTI
      AJMP  TIISR          ; Go to Interrupt Service Routine.
ORG   23h
I2CISR:  CLR   EI2     ; Disable I2C interrupt.
      ACALL  XRETI     ; Allow other interrupts to occur.
      PUSH   PSW
      PUSH   ACC
      MOV    A,R0
      PUSH   ACC
      MOV    A,R1
      PUSH   ACC
      MOV    A,R2
```

```
        PUSH   ACC

        MOV    StackSave, SP
        CLR    TIRUN
        SETB   TIRUN
JB      STP,NoGo
        JNB    MASTER, GoSlave
        MOV    MSGSTAT,#MGO
        JB     STR,GoMaster
NoGo:   MOV    MSGSTAT,#NOTSTR
        AJMP   Dismiss          ; Not a valid Start.
XRETI:  RETI
GoSlave: MOV   MSGSTAT,#SGO
AddrRcv: ACALL ClsRcv8
        JNB    DRDY,
STstRW: MOV    C,ACC.0          ; Save R/W~ bit in carry.
        CLR    ACC. JC
SRcv2:  CJNE   A,MYADDR,
        MOV    R1,#SRcvBuf      ; Set receive buffer address.
        MOV    R2,#RbufLen+1    ;
        SJMP   SRcv3
SRcvSto: MOV   @R1,A            ; Store the byte
        Inc    R1               ; Step address.
SRcv3:  ACALL  AckRcv8
        JNB    DRDY,SRcvEnd     ; Exit loop -end reception.
        DJNZ   R2,SRcvSto       ; Go to store byte if buffer not full.
MOV     MSGSTAT,#
        ACALL  SLnRCvdR         ; Handle new data - slave event routine.
        SJMP   GoIdle
SRcvEnd: CJNE  R0,#7,
MOV     MSGSTAT,#
        MOV    A,R1
        CLR    C
        SUBB   A,#SRcvBuf       ; number of bytes in ACC
        ACALL  SRCvdR           ; Handle new data - slave event routine.
        SJMP   SMsgEnd
SlvTx:  NOP
STx2:   CJNE   A,MYADDR,GoIdle  ; Not for us.
        MOV    I2DAT,#0         ; Acknowledge the address.
        JNB    ATN,$            ; Wait for attention flag.
        JNB    DRDY,SMsgEnd
        MOV    R1,#STxBuf       ; Start address of transmit buffer.
STxlp:  MOV    A,@R1            ; Get byte from buffer
        INC    R1
        ACALL  XmByte
        JNB    DRDY,SMsgEnd     ; Byte Tx not completed.
        JNB    RDAT,STxlp       ; Byte acknowledge, proceed trans.
        MOV    I2CON,#BCDR+BIDLE ; Master Nak'ed for msg end.
        MOV    MSGSTAT,#STXED
        ACALL  STXedR           ; Slave transmitted event routine.
        AJMP   Dismiss
SRcvErr: MOV   MSGSTAT,#SRERR   ; Flag bus/protocol error
        ACALL  SRErrR           ; Slave error event routine.
        SJMP   SMsgEnd
StxErr: MOV    MSGSTAT,#SRERR   ; Flag bus/protocol error
        ACALL  SRErrR
SMsgEnd: JB    MASTER,SMsgEnd2
        JB     STR,GoSlave      ; If it was a Start, be Slave
SMsgEnd2:
        AJMP   Dismiss
```

```
GoIdle:
        AJMP   Dismiss
GoMaster:
MOV     R1,#MasBuf        ; Master buffer address
        MOV     R2,MASTCNT        ; # of bytes, to send or rcv
        MOV     A,DESTADRW        ; Destination address (including
                                  ; R/W~ byte).
        JB      SUBADD,GoMas2     ; Branch if subaddress is needed.
ACALL   XmAddr
JNB     DRDY,GM2
        JNB     ARL,GM3
GM2:    AJMP    AdTxArl
GM3:    JB      RDAT,Noslave      ; No Ack for address transmission.
        JB      ACC.0, MRcv       ; Check R/W~ bit
        AJMP    MTx
GoMas2: NOP                       ; Subaddress needed. Address in ACC.
        CLR     ACC.0             ; Force a Write bit with address.
        ACALL   XmAddr
        JNB     DRDY,GM4
        JNB     ARL,GM5
GM4:    AJMP    AdTxArl
GM5:    JB      RDAT,Noslave      ; No Ack for address transmission.
        MOV     A,DESSUBAD
        ACALL   XmByte            ; Transmit subaddress.
        JNB     DRDY,SMsgEnd2     ; Arbitration loss (by Start or Stop)
        JB      ARL,SMsgEnd2      ; Arbitration loss occured.
        JB      RDAT,NoAck        ; Subadress transmission was not ack'ed.
        MOV     A,DESTADRW        ; Reload ACC with address.
        JNB     ACC.0, MTx
MOV     I2CON,#BCDR+BXSTR  ; Send Repeated Start.
        JNB     ATN,$
        MOV     I2CON,#.
        JNB     ATN,$             ; expecting an STR.
        JNB     ARL,GM6
        AJMP    MArlEnd           ; oops - lost arbitration.
GM6:    ACALL   XmAddr
        JNB     DRDY,GM7
        JNB     ARL,GM8
GM7:    AJMP    AdTxArl
GM8:    JB      RDAT,Noslave      ; No Ack - the slave disappeared.
        SJMP    MRcv
MTx:    NOP
MTxLoop: MOV    A,@R1             ; Get byte from buffer.
        INC     R1                ; Step the address.
        ACALL   XmByte
        JNB     DRDY,SMsgEnd2     ; Arbitration loss (by Start or Stop)
        JB      ARL,SMsgEnd2      ; Arbitration loss.
        JB      RDAT,NoAck
        DJNZ    R2,MTxLoop        ; Loop if more bytes to send.
MOV     MSGSTAT.
        SJMP    MTxStop
NoSlave: MOV    MSGSTAT,#MTXNOSLV
        SJMP    MTxStop
NoAck:  MOV     MSGSTAT,#MTXNAK
        SJMP    MTxStop
MRcv:   ACALL   ClaRcv8           ; Receive a byte.
        SJMP    MRcv2
MRcvLoop: ACALL AckRcv8
MRcv2:  JNB     DRDY,MArl         ; Other's Start or Stop.
        MOV     @R1,A             ; Store received byte.
```

```
        INC   R1              ; Advance address.
        DJNZ  R2,MRcvLoop
MOV   I2DAT,#80h
        JNB   ATN,$
        JNB   DRDY,MArl
        MOV   MSGSTAT,#MRCVED
        SJMP  MTxStop         ; Go to send Stop or Repeated Start.
MTxStop: JNB  RPSTRT,MTxStop2
        MOV   I2CON,#BCDR+BXSTR  ; Send Repeated Start.
        SJMP  MTxStop3
MTxStop2: MOV  C,SETMRQ       ; Set new Master Request if demanded
        MOV   MASTRQ,C        ; by SETMRQ bit of MASCMD.
MOV   I2CON,#BCDR+BXSTP  ; Request the HW to send a Stop.
MTxStop3: JNB  ATN,$          ; Wait for Attention
        MOV   I2CON,#BCDR
        JNB   ATN,$           ; Wait for ARL, STP or STR.
        JB    ARL,MarlEnd
ACALL  MastNext
JNB   MASTRQ,MMsgEnd
STR,MMsgEnd
        AJMP  GoMaster
MMsgEnd:                      ; End of Master messages,
        SJMP  Dismiss
MArl:
JNB   STR,MArl2
        AJMP  GoSlave
Marl2:
        AJMP  Dismiss
MArlEnd:
        SETB  MASTRQ
        AJMP  MArl
AdTxArl: JB   STR,MArl        ; Non-synchronous Start or Stop.
        JB    STP,MArl
CJNE   R0,#0,AdTxArl2
DEC   A               ;
        SJMP  AdAr3
AdTxArl2:
        RR    A               ; Realign partially Tx'ed ACC
        MOV   R1,A            ; and save itin R1
        MOV   A,R0            ; Pointer for lookup table
        MOV   DPTR,#MaskTable
        MOVC  A,@A+DPTR
        ANL   A,R1
MOV   I2CON,#BCXA+BCARL
        ACALL  RBit3
        JB    DRDY,AdAr3      ; Around if received address OK
        AJMP  SMsgEnd
AdAr3:   AJMP  STstRW
MaskTable: DB   0ffh,7Eh,3Eh,1Eh,0Eh,06h,02h,00h, ; 0ffh is dummy
Dismiss: ACALL  I2CDONE
MOV   I2CON,#BCARL+BCSTP+BCDR+BCXA+BIDLE
        CLR   TIRUN
        POP   ACC
        MOV   R2,A
        POP   ACC
        MOV   R1,A
        POP   ACC
        MOV   R0,A
        POP   ACC
        POP   PSW
```

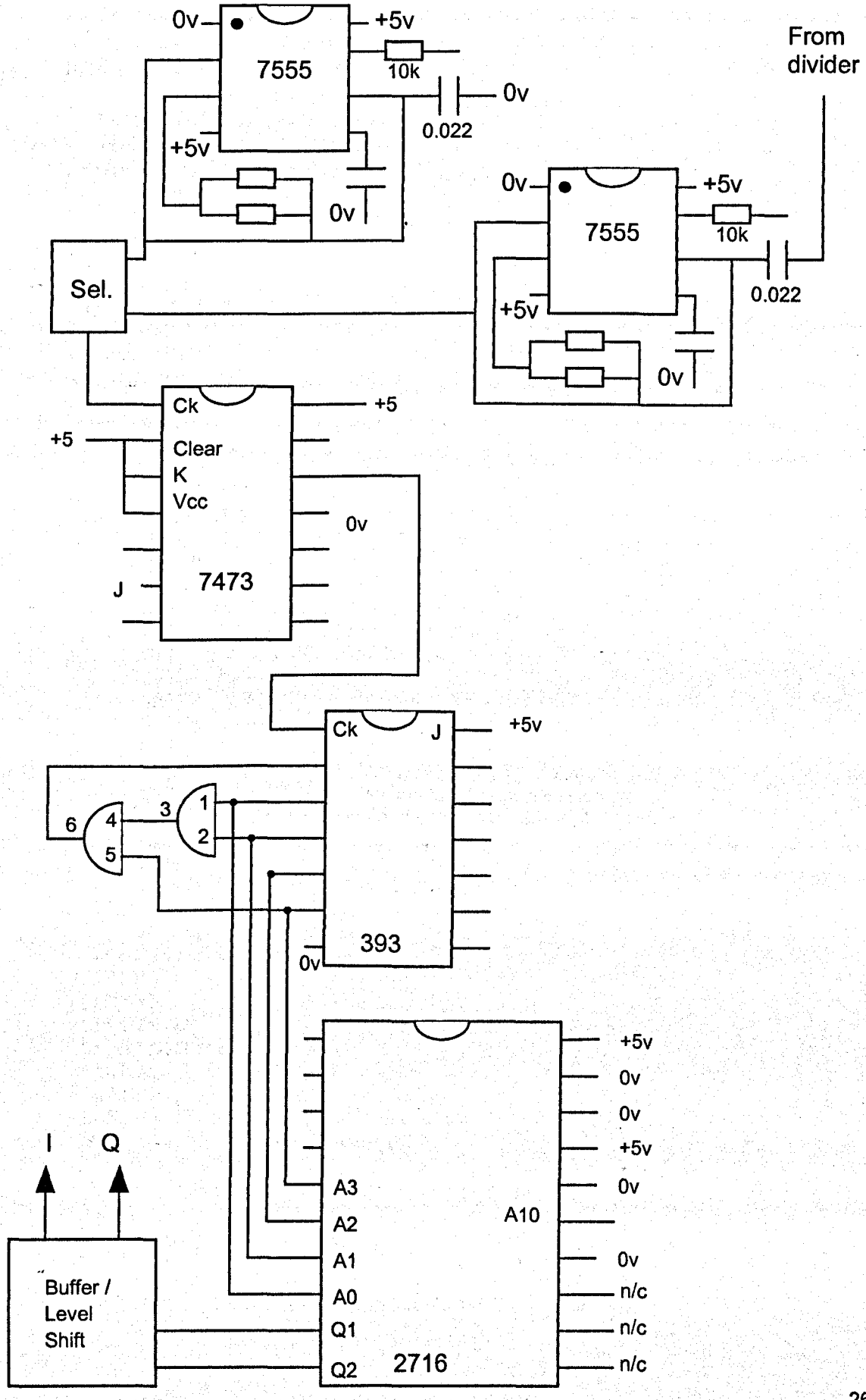
```
        SETB   EI2

        RET             ; Return from I2C interrupt Service Routine
XmAddr: MOV    I2DAT,A          ; Send first bit, clears DRDY.
        MOV    I2CON,#BCARL+BCSTR+BCSTP
        MOV    R0,#8            ; Set R0 as bit counter
        SJMP   XmBit2
XmByte: MOV    R0,#8
XmBit:  MOV    I2DAT,A          ; Send the first bit.
XmBit2: RL     A                ; Get next bit.
        JNB    ATN,$            ; Wait for bit sent.
        JNB    DRDY,XmBex       ; Should be data ready.
        DJNZ   R0,XmBit         ; Repeat until all bits sent.
        MOV    I2CON,#BCDR+BCXA ; Switch to receive mode.
        JNB    ATN
XmBex:  RET
ClsRcv8: MOV   I2CON,#BCARL+BCSTR+BCSTP+BCXA
        JNB    ATN,$
        JNB    DRDY,RCVex
        SJMP   Rcv8
AckRcv8: MOV   I2DAT,#0         ; Send Ack (low)
        JNB    ATN,$
        JNB    DRDY,RCVerr      ; Bus exception - exit.
ClaRcv8: MOV   I2CON,#BCDR+BCXA
        JNB    ATN,$
Rcv8:   MOV    R0,#
        CLR    A                ; Init received byte to 0.
RBit:   ORL    A,I2DAT          ; Get bit, clear ATN.
RBit2:  RL     A                ; Shift data.
        JNB    ATN,$            ; Wait for next bit.
        JNB    DRDY,RCVex       ; Exit if not a data bit (could be Start/
RBit3:  DJNZ   R0,RBit          ; Repeat until 7 bits are in.
        MOV    C,RDAT           ; Get last bit, don't clear ATN.
        RLC    A                ; Form full data byte.
RCVex:  RET
RCVerr: MOV    R0,#9            ; Return non legitimate bit count
        RET
TIISR:  CLR    MASTRQ           ; "Manual" reset.
        MOV    I2CON,#BXSTP     ;
        MOV    I2CON,#BCXA+BCDR+BCARL+BCSTR+BCSTP
TI1:    MOV    MSGSTAT,#TIMOUT  ; Status Flag for Main.
TI2:    MOV    A,#0FFh
        CJNE   A,TITOCNT,TI3    ; Increment TITOCNT, saturating
        SJMP   TI4              ; at FFh.
TI3:    INC    TITOCNT
TI4:    ACALL  RECOVER
SETB    CLRTI           ; Clear TI interrupt flag.
        ACALL  XRETI
        MOV    SP,StackSave
        AJMP   Dismiss
RECOVER: CLR   EA
        CLR    MASTRQ           ; "Manual" reset.
        MOV    I2CON,#BCXA+BIDLE+BCDR+BCARL+BCSTR+BCSTP
        CLR    SLAVEN   ; Non I2C TimerI mode
        SETB   TIRUN
        MOV    R1,#0ffh
DLY5:   NOP
        NOP
        NOP
        DJNZ   R1,DLY5
```

```
        CLR   TIRUN
        SETB  CLRTI

        SETB  SCL      ; Issue clocks to help release other devices.
        SETB  SDA
        MOV   R1,#08h
RC7:    CLR   SCL
        DB    0,0,0,0,0
        SETB  SCL
        DB    0,0,0,0,0
        DJNZ  R1,RC7
        CLR   SCL
        DB    0,0
        CLR   SDA
        DB    0,0
        SETB  SCL
        DB    0,0,0,0,0
        SETB  SDA
        DB    0,0,0,0,0  ; Issue a Stop.
Rex:    MOV   I2CON,#BCXA+BCDR+BCARL+BCSTR+BCSTP ; clear flags
        SETB  EA
        RET
Reset:  MOV   SP,#07h          ;Set stack location.
        CLR   A
        MOV   DPTR,#R_CTVAL
        MOVC  A,@A+DPTR
        MOV   I2CFG,A
        CLR   A
        MOV   DPTR,#R_MYADDR
        MOVC  A,@A+DPTR        ; Get this node's address from ROM table
        MOV   MYADDR,A         ; into MYADDR RAM location.
CLR   OnLED
Reset2: CLR   ErrLED           ; Flash LED.
        ACALL LDELAY
        SETB  ErrLED
        CLR   SErrFLAG
        CLR   TRQFLAG
        MOV   FAILCNT,#50h
        SETB  TogLED
        MOV   TOGCNT,#
        MOV   I2CON,#BIDLE     ; Slave will idle till next Start.
        SETB  SLAVEN           ; Enable slave operation.
SETB  ETI              ; Enable timer I interrupts.
        SETB  EI2              ; Enable I2C port interrupts.
        SETB  EA               ; Enable global interrupts.
MOV   MASCMD,#0h       ; "Regular" master transmissions.
        MOV   DPTR,#PongADDR
        CLR   A
        MOVC  A,@A+DPTR
        MOV   DESTADRW,A
        MOV   MASTCNT,#01h     ; Message length - a single byte.
PPSTART:
        MOV   MasBuf,#00h
PP2:
        SETB  TRQFLAG
        SETB  MASTRQ
        MOV   R1,#0ffh
PP22:   JNB   TRQFLAG,PP3      ; Transmitted OK
        DJNZ  R1,PP22
MFAIL1: DJNZ  FAILCNT,PP2
```

```
        ACALL  RECOVER
        SJMP   Reset2
PP3:    MOV    R0,#0ffh        ; Software timeout loop count.
PP31:   MOV    R1,#0ffh
PP32:   JB     TRQFLAG,PP2     ; Rcvd ok as slave, go transmit.
        JB     SErrFLAG,PP5
        DJNZ   R1,PP32
        DJNZ   R0,PP31
PPTO:   ACALL  RECOVER         ; Software timeout.
        AJMP   Reset2
PP5:    CLR    ErrLED          ; Receive error.
        ACALL  LDELAY
        SETB   ErrLED
        CLR    SErrFLAG
        AJMP   PPSTART
LDELAY: MOV    R2,#030h
LDELAY1: MOV   R1,#0ffh
        DJNZ   R1,$
        DJNZ   R2,LDELAY1
        RET
SRcvdR: NOP
        MOV    A,SRcvBuf
        JNZ    SR2
        MOV    MasBuf,#01h     ; It was ping-pong reset value
        SJMP   SR3
SR2:    INC    MasBuf          ; The expected data.
        CJNE   A,MasBuf,ErrSR
        INC    MasBuf
DJNZ    TOGCNT,SR3
        CPL    TogLED          ; Toggle output
        MOV    TOGCNT,#050h    ;
SR3:    CLR    SErrFLAG
        SETB   TRQFLAG         ; Request main to transmit
        RET
ErrSR:  SETB   SErrFLAG
        RET
SLnRcvdR:
STXedR:
SRErrR: JMP    ErrSR
MastNext:
        MOV    A,MSGSTAT
        CJNE   A,#MTXED,MN1
        MOV    FAILCNT,#50h
        CLR    TRQFLAG
        RET
MN1:
        RET
I2CDONE:
        MOV    A,MSGSTAT
        CJNE   A,#NOTSTR,I2CD1
        DJNZ   FAILCNT,I2CD1
        MOV    FAILCNT,#01h    ; Too many "illegal" i2c interrupts
        CLR    EI2             ; - shut off.
I2CD1:  RET
R_MYADDR: DB   4Eh     ; This node's address
R_CTVAL:  DB   02h     ; CT1, CT0 bit values
PongADDR: DB   4Ah
END
```

Appendix 8

Code generation (test)



Appendix 9

Filter programs for the TMS320C50, the first is an audio bandpass 1kHz to 2kHz pass program with the on board audio interface chip (DSK board) initialised. The second is the pulse shaping filter, 20 tap F.I.R. with roll off of 0.35.

```
;   Blackman filter windowing was used
;   when calculating the filter's coefficients.
; -> AIC
          .mmregs
          .ds   0f00h
TA    .word 16 ;
RA    .word 16 ;
TAp   .word 1 ; not used
RAp   .word 1 ; " "
TB    .word 31;
RB    .word 31;
AIC_CTR .word 08h ;
Output  .word 0
Temp   .word 0;
.include "bandpass.flt"
.ps   080ah      ;
rint:  B     RECEIVE     ;0A; Serial port receive interrupt RINT.
xint:  B     TRANSMIT    ;0C; Serial port transmit interrupt XINT.
; TMS32C05X INIT                    *
          .ps   0a00h
          .entry
          SETC   INTM          ; Disable interrupts
          LDP    #0            ; Set data page pointer
          OPL    #0834h,PMST   ; OR the status register
          LACC   #0
          SAMM   CWSR          ; Set software wait state to 0
          SAMM   PDWSR         ;
          SETC   SXM
          SPLK   #022h,IMR     ; Using XINT syn TX & RX
          CALL   AICINIT       ; initialize AIC and enable interrupts
          SPLK   #12h,IMR ; Select serial port receive interrupt
          CLRC   OVM
          SPM 0
          CLRC   INTM          ; Enable interrupts
for  IDLE
          B     for
;------- end of main program ----------;
RECEIVE:
  LDP #XN
  CLRC INTM
  LAMM DRR ; get input in ACC
  SACL XN
; start FIR
  LAR  ar0,#XNLAST    ; FIR data value in data
  MAR  *,ar0          ; Make AR1 current.
  RPT  #taps-1
  MACD #coefftab,*-   ; Coefficients stored in program memory.
  APAC
  SACH Output,1
  LACC Output
  SFL
  AND  #0fffch ; and ls 2 bits of top 16 bits, for DAC.
```

```
  SAMM DXR
  RETE
TRANSMIT:
  RETE
* aic initialization data
AICINIT:
        SPLK #01h,PRD
        SPLK #20h,TCR
        MAR *,AR0
        LACC #0008h
        SACL SPC
        LACC #00c8h
        SACL SPC
        LACC #080h
        SACH DXR
        SACL GREG
        LAR AR0,#0FFFFh
        RPT #10000 ;
        LACC *,0,AR0 ;
        SACH GREG
        LDP #TA ;
        SETC SXM ;
        LACC TA,9 ;
        ADD RA,2 ;
        CALL AIC_2ND ;
        LDP #TB
        LACC TB,9
        ADD RB,2 ;
        ADD #02h ;
        CALL AIC_2ND ;
        LDP #AIC_CTR
        LACC AIC_CTR,2
        ADD #03h ;
        ;LACC #11100000b
        CALL AIC_2ND ;
        RET ;
        .end

AIC_2ND:
LDP #0
SACH DXR ;
CLRC INTM
IDLE
ADD #6h,15 ;
SACH DXR ;
IDLE
SACL DXR ;
IDLE
LACL #0 ;
SACL DXR
IDLE
SETC INTM
RET ;

*       Pulse shaping Filter routine

*   Output : Filtered/Pulse shaped data o/p, the variables are "ikf","qkf"

*      Job : To shape the incoming diff. enc. and mapped data
*    Cycles : 19 max
*
*            Filter Characteristics
*
*      Sampling Frequency : 48 Khz (i/p is 12 Kbaud/s,
*           Filter type : Parks Mcllelan window FIR
*       Percent Roll-off : 35%
*            Alpha value :
*          Filter length : 20 (interpolated to a factor of 4, therefore 20/4=5
* Coefficient Quantization : 16 bits
* Note: The Filter characteristics are the same for both I & Q filters.
*      Hence the same filter coefficients are used for both,
*      though the filter i/p I & Q delays are different.
* Note : The first 3 interrupt routines would call Mod_fltr
*       and the last interrupt routine would call Modfltr to do filter
*       delays too
```

```
Mod_fltr
************ Mod I-arm Filter routine ************
*
rpt    #coefnum       ; repeat coefnum+1=6 times
                      ; the following instruction
mads   *-             ; multiply & accum., coefficients
apac                  ; form result
       sach   ikf,1          ; save (scaled) filtered output in ikf
********** End Mod I-arm Filter routine **********
************ Mod Q-arm Filter routine ************
rptz   #coefnum       ; repeat coefnum+1=6 times
                      ; the following instruction
       mads   *-          ; multiply, accum.
apac                  ; form result
       sach   qkf,1       ; save (scaled) filtered  output in qkf

       call   modout      ; call modout in io.asm to output
                          ; mod o/p to 2 files named ik.dat & qk.dat
*      lacc   ikf         ; load ikf
*      and    #mask       ; make lower two LSBs 00 so that
*                         ; AIC registers are not tampered
*      samm   dxr         ; send o/p in data transmit register
*      lacc   qkf         ; load qkf
*      and    #mask       ; make lower two LSBs 00 so that
*                         ; AIC registers are not tampered
*      samm   dxr         ; send o/p in data transmit register
ret                   ; return
Modfltr
rpt    #coefnum       ; repeat coefnum+1=6 times
                      ; the following instruction
       madd   *-          ; multiply, accum. AND  delay
                          ; (perform convolution)
       apac               ; form result
       sach   ikf,1       ; save output
********** End Mod I-arm Filter routine **********
************ Mod Q-arm Filter routine ************
rptz   #coefnum       ; repeat coefnum+1=15 times
                      ; the following instruction
       madd   *-          ; multiply, accum. AND  delay
                          ; (perform convolution)
       apac               ; form result
       sach   qkf,1       ; save output
call   modout         ; call modout
********** End Mod Q-arm Filter routine **********
*      lacc   ikf         ; load ikf
*      and    #mask       ; make lower two LSBs 00 so that
*                         ; AIC registers are not tampered
*      samm   dxr         ; send o/p in data transmit register
*      lacc   qkf         ; load qkf
*      and    #mask       ; make lower two LSBs 00 so that
*                         ; AIC registers are not tampered
*      samm   dxr         ; send o/p in data transmit register
ret                   ; return
****    End Modulator Pulse Shaping Filter Module    ****

****       Demodulator Matched  Filter Module        ****
*       Matched Filter routine
*    Input : 24 Kbaud/s mod filtered o/p, the values are loaded in appropriate
*            filter i/p locations before this module is called
*   Output : Filtered mod data o/p, the filtered ik sample is in ACCB &
```

```
*           filtered qk sample is in Acc.
*     Job : To filter the incoming I & Q demod input data.
*   Cycles : 52 max
* Note: The Filter characteristics are the same for both I & Q filters.
*       Hence the same filter coefficients are used for both,
*       though the filter i/p I & Q delays are different.
************ Demod I-arm Filter routine ************
*  Fiter coefficient locations : External Program Memory
*  Filter delays (past input samples) locations : Internal Data Memory (DARAM)
*  Implementation : FIR convolution
DMD_fltr

     rptz    #coeffnm      ; repeat coeffnm+1=20 times
                           ; the following instruction
     macd    #dmdcoff,*-   ; multiply, accum. AND  delay
                           ; (perform convolution)
     apac                  ; form result
     sacb                  ; store result in Acc into Acc Buffer (ACCB)

********** End Demod I-arm Filter routine **********

************ Demod Q-arm Filter routine ************
*
*  Fiter coefficient locations : External Program Memory
*  Filter delays (past input samples) locations : Internal Data Memory (DARAM)
*  Implementation : FIR convolution
*
*********************************************

     rptz    #coeffnm      ; repeat coeffnm+1=20 times
                           ; the following instruction
     macd    #dmdcoff,*-   ; multiply, accum. AND  delay
                           ; (perform convolution)
     retd                  ; return delayed
     apac                  ; form result, Acc has result
     sbrk    #coeffnm      ; ar5=ar5-coeffnm(19) so that it points to the
                           ; Ik demod filter i/p location


********** End Demod Q-arm Filter routine **********
```

Appendix 10

Maxim Data

19-3455; Rev 0; 12/05

EVALUATION KIT AVAILABLE

MAXIM

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

MAX2450

General Description

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

Pulling the CMOS-compatible ENABLE pin low shuts down the MAX2450 and reduces the supply current to less than 1µA. To minimize spurious feedback, the MAX2450's internal oscillator is set at twice the IF via external tuning components. The oscillator and associated phase shifters produce differential signals exhibiting low amplitude and phase imbalance, yielding modulator sideband rejection of 36dB. The MAX2450 comes in 20-pin SO and QSOP packages.

Applications

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

Features

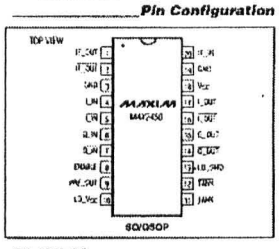
- Combines Quadrature Modulator and Demodulator
- Integrated Quadrature Phase Shifters
- On-Chip Oscillator (Requires External Tuning Circuit)
- On-Chip Divide-by-8 Prescaler
- Modulator Input Bandwidth Up to 15MHz
- Demodulator Output Bandwidth Up to 8MHz
- 51dB Demodulator Voltage Conversion Gain
-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

Ordering Information

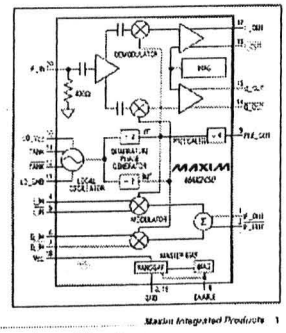
PART	TEMP. RANGE	PIN-PACKAGE
MAX2450CWP	0°C to +70°C	20 Wide SO
MAX2450CEP	0°C to +70°C	20 QSOP*

*Contact factory for availability.

Pin Configuration



Functional Diagram



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

MAX2450

AC ELECTRICAL CHARACTERISTICS (continued)

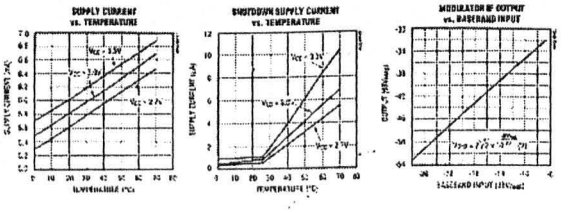
(MAX2450 EV kit using MAX2450CWP, VCC = LO_VCC = ENABLE = 3.0V, fLO = 160MHz, fLO_IN = fLO_IN/2 = 80MHz, VLO_IN = VLO_IN/2 = 1.2Vp-p, fIF_IN = 70.1MHz, VIF_IN = 7.85mVp-p, TA = +25°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
MODULATOR						
Allowable Differential Input Voltage	VI_IN, VQ_IN	(Note 1)			1.35	Vp-p
Input Common-Mode Voltage Range			1.25		1.75	V
I and Q Signal -3dB Bandwidth	BWMOD			15		MHz
IF Differential Output Voltage	VIF_OUT, IF_OUT	VI_IN = VQ_IN = 1.2Vp-p, RL = 200Ω differential, CL = 5pF differential		85		mVp-p
IF Output (LO) Level	LOIF	VI_IN = 1.2Vp-p composite (Note 4)		-50		dBc
IF Output (RF) Level	RFIF	VI_IN = 1.2Vp-p composite (Note 4)		-60		dBc
Sideband Rejection				36		dBc
Carrier Suppression at Modulator Output				30		dBc

Note 1: Guaranteed by design and test.
Note 2: fIF_IN = 2 times at 70.1MHz and 70.15MHz, VIF_IN = 1.41mVp-p per tone.
Note 3: The frequency range can be extended for signal dividers, but has not been characterized. At higher frequencies, the modulator IF output amplitude may decrease and additional harmonics may increase.
Note 4: QLO_IN, IF pins are terminated in a 200Ω resistor. fLO_IN = 2 times at 552MHz and 600MHz.

Typical Operating Characteristics

(MAX2450 EV Kit using MAX2450CWP, VCC = LO_VCC = ENABLE = 3.0V, fLO = 160MHz, fLO_IN = fLO_IN/2 = 80MHz, VLO_IN = VQ_IN/QLO_IN = 1.2Vp-p, fIF_IN = 70.1MHz, VIF_IN = 2.82mVp-p, TA = +25°C, unless otherwise noted.)



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

MAX2450

ABSOLUTE MAXIMUM RATINGS

VCC, LO_VCC to GND ..... -0.3V to +4.5V
ENABLE, TANK, TANK, I_IN, I_IN, Q_IN, Q_IN to GND ..... -0.3V to (VCC + 0.3V)
IF_IN to GND ..... -1.5V to +1.5V

Continuous Power Dissipation (TA = +70°C)
Wide SO (derate 10.00mW/°C above +70°C) 800mW
QSOP (derate 9.1mW/°C above +70°C) 727mW
Operating Temperature Range ..... 0°C to +70°C
Storage Temperature Range ..... -65°C to +165°C
Lead Temperature (soldering, 10sec) ..... +300°C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(VCC = LO_VCC = TANK = 2.7V to 3.3V, ENABLE = VCC, fLO = 160MHz, fLO_IN = fLO_IN/2, VLO_IN = VQ_IN/QLO_IN = 1.2Vp-p, fIF_IN = 70.1MHz, VIF_IN = 7.85mVp-p, TA = 0°C to +70°C, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage Range	VCC, LO_VCC		2.7		3.3	V
Supply Current	ICC(ON)			5.9	6.9	mA
Shutdown Supply Current	ICC(OFF)	ENABLE = 0.4V		2	20	µA
Enable/Disable Time	tENBTP			10		µs
ENABLE Bias Current	IEN	ENABLE = VCC		1	3	µA
ENABLE High Voltage	VENH		VCC - 0.4			V
ENABLE Low Voltage	VENL				0.4	V
I_IN, I_IN, Q_IN, Q_IN Self-Bias DC Voltage Level	VI_IN, VQ_IN		1.25	1.5	1.75	V
Modulator Differential Input Impedance	ZI_IN, ZQ_IN		35	44		kΩ
IF_OUT, IF_OUT DC Bias Voltage	VIF_OUT, IF_OUT			VCC - 1.5		V
Demodulator IF Input Impedance	ZIF_IN		320	400	160	Ω
Demodulator I and Q Baseband DC Offset				±11	±50	mV
I_OUT, I_OUT, Q_OUT, Q_OUT DC Bias Voltage Level	VI_OUT, VQ_OUT			1.2		V

AC ELECTRICAL CHARACTERISTICS

(MAX2450 EV kit using MAX2450CWP, VCC = LO_VCC = ENABLE = 3.0V, fLO = 160MHz, fLO_IN = fLO_IN/2 = 80MHz, VLO_IN = VLO_IN/2 = 1.2Vp-p, fIF_IN = 70.1MHz, VIF_IN = 7.85mVp-p, TA = +25°C, unless otherwise noted.)

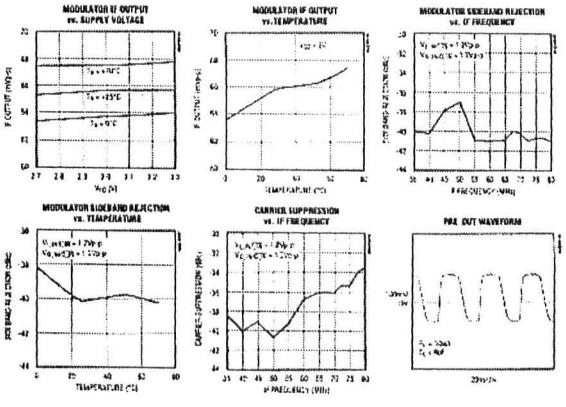
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DEMODULATOR						
I and Q Amplitude Balance				< ±0.45		dB
I and Q Phase Accuracy				< ±1.5		degrees
Voltage Conversion Gain				51		dB
Allowable I and Q Voltage Swing		(Note 1)			1.35	Vp-p
Noise Figure	NF			19		dB
I and Q (1MHz) Level	IM3(Q)	(Note 2)		-44		dBc

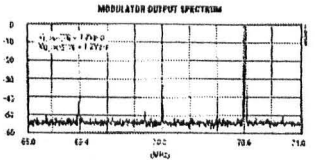
3V, Ultra-Low-Power Quadrature Modulator/Demodulator

MAX2450

Typical Operating Characteristics (continued)

(MAX2450 EV Kit using MAX2450CWP, VCC = LO_VCC = ENABLE = 3.0V, fLO = 160MHz, fLO_IN = fLO_IN/2 = 80MHz, VLO_IN = VQ_IN/QLO_IN = 1.2Vp-p, fIF_IN = 70.1MHz, VIF_IN = 2.82mVp-p, TA = +25°C, unless otherwise noted.)

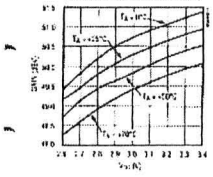




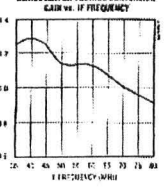
Typical Operating Characteristics (continued)

(MAX2450 EV Kit using MAX2450CWP, V_{CC} = LO_V_{CC} = ENABLE = 3.0V, f_{LO} = 1400MHz, f_{LO_IN} = f_{LO} / 8 = 935MHz, V_{LO_IN} = V_{LO_GND} = 1.7V_{p-p}, f_{IF_IN} = 70.1MHz, V_{IF_IN} = 2.83mV_{p-p}, T_A = +25°C, unless otherwise noted.)

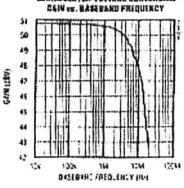
DEMODULATOR VOLTAGE CONVERSION GAIN vs. TEMPERATURE AND SUPPLY



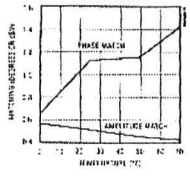
DEMODULATOR VOLTAGE CONVERSION GAIN vs. IF FREQUENCY



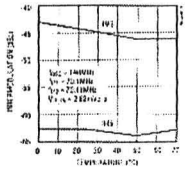
DEMODULATOR VOLTAGE CONVERSION GAIN vs. BASEBAND FREQUENCY



DEMODULATOR I/Q PHASE AND AMPLITUDE MISMATCH vs. TEMPERATURE



DEMODULATOR INTERMOD POWER vs. TEMPERATURE



Pin Description

PIN	NAME	FUNCTION
1	I_OUT	Modulator I Output
2	IF_OUT	Modulator + Inverting Output
3, 10	GND	Ground
4	I_IN	Baseband I Voltage Input
5	I_IN	Baseband I Voltage Inverting Input
6	Q_IN	Baseband Q Voltage Input
7	Q_IN	Baseband Q Voltage Inverting Input
8	ENABLE	Enable Control, active-high
9	PRE_OUT	Local-Oscillator, Divide-by-8, Prescaler Output
10	LO_VCC	Local-Oscillator Supply. Bypass separately from VCC
11	TANK	Local-Oscillator Resonant Tank Input (Figure 4)
12	TANK	Local-Oscillator Resonant Tank Inverting Input (Figure 4)
13	LO_GND	Local-Oscillator Ground
14	Q_OUT	Demodulator Baseband Inverting Output
15	Q_OUT	Demodulator Baseband Q Output
16	I_OUT	Demodulator In-phase Inverting Output
17	I_OUT	Demodulator In-phase Output
18	VCC	Modulator and Demodulator Supply
20	IF_IN	Demodulator IF Input

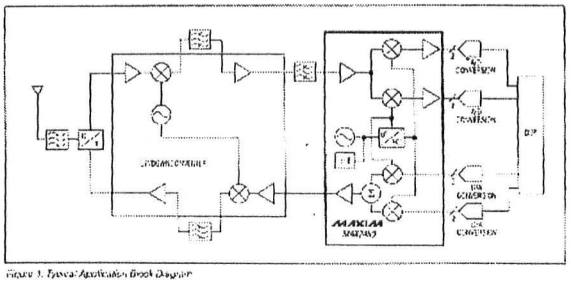


Figure 1. Typical Application Block Diagram

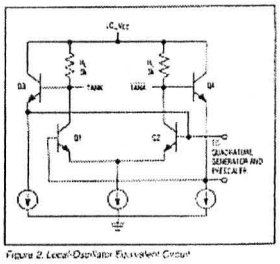


Figure 2. Local-Oscillator Equivalent Circuit

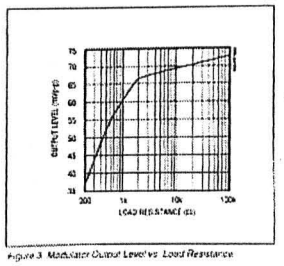


Figure 3. Modulator Output Level vs. Load Resistance

Detailed Description

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

Demodulator

The demodulator contains a single-ended-to-differential converter, two Gilbert-cell multipliers, and two fixed-gain stages. The IF signal should be AC coupled into IF_IN. Internally, IF_IN is terminated with a 450Ω resistor to GND and provides a gain of 14dB. The amplified IF signal is fed into the I and Q mixers for demodulation. The multipliers mix the IF signal with the quadrature LO signals, resulting in baseband I and Q signals. The conversion gain of the multipliers is 15dB. These signals are further amplified by 21dB by the baseband amplifiers. The baseband I and Q amplifier chains are DC coupled.

Local Oscillator

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

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

Quadrature Phase Generator

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

Prescaler

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

Modulator

The modulator accepts I and Q differential baseband signals up to 1.35V_{p-p} with frequencies up to 15MHz, and upconverts them to the IF frequency. Since these inputs are biased internally at around 1.5V, I and Q signals should be capacitively coupled into these high-impedance ports (the differential input impedance is approximately 44kΩ). The self-bias design yields very low on-chip offset, resulting in excellent carrier suppression. Alternately, a differential DAC may be connected without AC coupling, as long as a common-mode voltage range of 1.25V to 1.75V is maintained. For single-ended drive, connect I_IN and Q_IN via AC-coupling capacitors (0.1µF) to GND.

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

Master Bias

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

Applications Information

Figure 4 shows the implementation of a resonant tank circuit. The inductor, two capacitors, and a dual varactor form the oscillator's resonant circuit. In Figure 4, the oscillator frequency ranges from 1500MHz to 1600MHz.

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

$$f_0 = \frac{1}{2\pi\sqrt{L_{EQ} C_{EQ}}}$$

where

$$C_{EQ} = \frac{1}{\frac{1}{C1} + \frac{1}{C2} + C_{VAR}} + C_{STRAY}$$

and

$$L_{EQ} = L + L_{STRAY}$$

where C_{STRAY} = parasitic capacitance and L_{STRAY} = parasitic inductance.

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

$$Q = R_{EQ}\sqrt{\frac{C_{EQ}}{L_{EQ}}}$$

where R_{EQ} = 10kΩ (Figure 2).

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

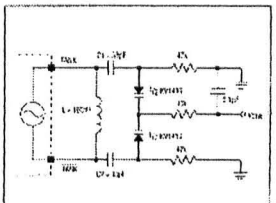
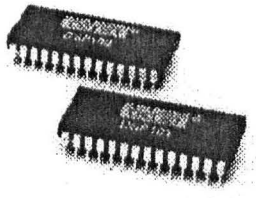


Figure 4. Typical Resonant Tank Circuit

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 (408) 737-7600

Appendix 11
Burr Brown Data

BURR-BROWN®



DSP101
DSP102

DSP-Compatible Sampling Single/Dual ANALOG-TO-DIGITAL CONVERTERS

FEATURES

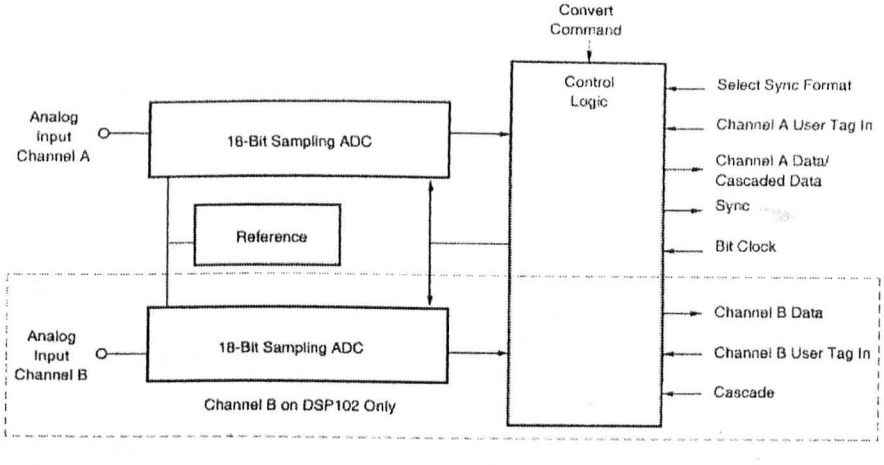
- ZERO-CHIP INTERFACE TO STANDARD DSP ICs: AD, AT&T, MOTOROLA, TI
- SINGLE CHANNEL: DSP101
- DUAL CHANNEL: DSP102
  Two Serial Outputs or Cascade to Single 32-Bit Word
- SAMPLING RATE TO 200kHz
- DYNAMIC SPECIFICATIONS:
  Signal/(Noise + Distortion) = 88dB;
  Spurious-Free Dynamic Range = 94dB;
  THD = –91dB
- SERIAL OUTPUT DATA COMPATIBLE WITH 16-, 24-, AND 32-BIT DSP IC FORMATS

DESCRIPTION

The DSP101 and DSP102 are high performance sampling analog-to-digital converters designed for simplicity of use with modern digital signal processing ICs. Both are complete with all interface logic for use directly with DSP ICs, and provide full sampling and conversion at rates up to 200kHz.

The DSP101 offers a single conversion channel, with 18 bits of serial data output, allowing the user to drive 16-bit, 24-bit, or 32-bit DSP ports. The DSP102 offers two complete conversion channels, with either two full 18-bit output ports, or a mode to cascade two 16-bit conversions into a 32-bit port as one word.

Both the DSP101 and DSP102 are packaged in standard, low-cost 28-pin plastic DIP packages. Each is offered in two performance grades to match application requirements.



International Airport Industrial Park • Mailing Address: PO Box 11400 • Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd. • Tucson, AZ 85706
Tel: (520) 746-1111 • Twx: 910-952-1111 • Cable: BBRCORP • Telex: 066-6491 • FAX: (520) 889-1510 • Immediate Product Info: (800) 548-6132

PDS-1068C

Printed in U.S.A. October, 1993

SPECIFICATIONS

ELECTRICAL

At T_A = 0°C to 70°C, ±2.75V input signal, sampling frequency (f_S) = 200kHz, $V_A+ = V_D = +5V$, $V_A- = -5V$, 16MHz external clock on OSC1, CLKOUT tied to CLKIN, 8MHzdata transfer clock on XCLK, data analysis band-limited to 20kHz, unless otherwise specified.

PARAMETER	CONDITIONS	DSP101JP DSP102JP MIN	DSP101JP DSP102JP TYP	DSP101JP DSP102JP MAX	DSP101KP DSP102KP MIN	DSP101KP DSP102KP TYP	DSP101KP DSP102KP MAX	UNITS
RESOLUTION				18			*	Bits
ANALOG INPUT								
Voltage Range			±2.75V			*		V
Impedance			1			*		kΩ
Capacitance			20			*		pF
THROUGHPUT SPEED								
Complete Cycle	Acquisition + Conversion			5			*	µs
Throughput Rate		200			*			kHz
AC ACCURACY(1)								
Signal-to-(Noise + Distortion) Ratio	f_{IN} = 1kHz	83	86		86	88		dB(2)
	f_{IN} = 1kHz (–60dB)		32			*		dB
	f_{IN} = 25kHz		82			*		dB
Total Harmonic Distortion	f_{IN} = 1kHz		–90	–86		–91	–89	dB
Spurious-Free Dynamic Range	f_{IN} = 1kHz	89	92		92	94		dB
Signal-to-Noise Ratio (SNR)	f_{IN} = 1kHz	84	88		87	89		dB
DC ACCURACY								
Gain Error				±5			*	%
Gain Error Mismatch	DSP102 Channels			±2			*	%
Integral Linearity	±2.75V Input Range		Sufficient to meet AC Accuracy Specifications					
Differential Linearity	±2.75V Input Range		Sufficient to meet AC Accuracy Specifications					
Integral Linearity Error	±0.7V Input Range		±0.003			*		%
Differential Linearity Error	±0.7V Input Range		±0.002			*		%
No Missing Codes	±0.7V Input Range		14			*		Bits
Bipolar Zero Error (3)			±2					mV
Bipolar Zero Mismatch (3)	DSP102 Channels		±2					mV
Power Supply Sensitivity	–5.25V < V_A- < –4.75V		–60			*		dB
	+4.75V < V_A+, V_D+ < +5.25V		–60			*		dB
SAMPLING DYNAMICS								
Aperture Delay			30			*		ns
Aperture Jitter			100			*		ps, rms
Transient Response			1			*		µs
Overvoltage Recovery			5			*		µs
DIGITAL INPUTS								
Logic Levels (Except OSC1)								
V_{IL}	I_L = ±10µA	0		+0.8	*		*	V
V_{IH}	I_H = ±10µA	+2.4		+5	*		*	V
OSC1 Clock			74HC Compatible					
Frequency				16				MHz
Data Transfer Clock (XCLK)								
Frequency		0.1		12	*		*	MHz
Duty Cycle		40	50	60	*		*	%
Conversion Clock (CLKIN)								
Frequency		0.5		5.33	*		*	MHz
Duty Cycle		25	33	55	*	*	*	%
DIGITAL OUTPUTS								
Format			Serial; MSB first, 16/18-bit and Cascaded 32-bit Mode					
Coding			Binary Two's Complement					
Logic Levels (Except OSC2)								
V_{OL}	I_{SINK} = 4mA	0		+0.4	*		*	V
V_{OH}	I_{SOURCE} = 4mA	+2.4		+5	*		*	V
OSC2			Can only be used to drive crystal oscillator.					
Conversion Clock (CLKOUT)								
Drive Capability		±2mA			*			mA
POWER SUPPLIES								
Rated Voltage								
V_A+		+4.75	+5	+5.25	*	*	*	V
V_A-		–5.25	–5	–4.75	*	*	*	V
V_D		+4.75	+5	+5.25	*	*	*	V
Power Consumption	XCLK = OSC1 = 12MHz		250	425		*	*	mW
Supply Current	XCLK = OSC1 = 12MHz							
I_A+			30	45		*	*	mA
I_A-			–18	–25		*	*	mA
I_D			5	15		*	*	mA
TEMPERATURE RANGE								
Specification		0		+70	*		*	°C
Storage		–65		+125	*		*	°C

NOTES: (1) All dynamic specifications are based on 2048-point FFTs, using four-term Blackman-Harris window. (2) All specifications in dB are referred to a full-scale input, ±2.75Vp-p. (3) Adjustable to zero with external potentiometer.

Appendix 12

The original front end of the receiver consisted of two amplification stages. The first stage is designed for minimum noise and the second stage for maximum gain. The transistors chosen were a matched pair of Mitsubishi MGF1402 GaAs FET devices. The input, output and interconnections were designed for an impedance of 50 ohms.

The interconnection and matching calculations were done using 'Z-match', a PC based Smith chart program. The actual board dimensions were calculated using program 'microbas' on the PC.

Stability Calculations;

Stability factor

$$K = [1 - |S_{11}|^2 - |S_{22}|^2 + |S_{11}^* S_{22} - S_{12}^* S_{21}|] / 2 * |S_{12}| * |S_{12}|$$

1-0.648-0.693+(0.648*0.693-0.079*1.525) / 2*0.079*1.525

1.778

The result shows that the transistor amplifier is unconditionally stable at 1GHz with Vds=3v and Id=10mA. For optimum noise performance it is necessary to match the transistor into Γo, the optimum reflection coeficient that is quoted by the manufacturer.

$\Gamma o = 0.42 \angle -160$ with Vds and Id as above.

The gain at this reflection coefficient is less than the gain achieved at a complex conjugate match, calculated using Masons rules;

$$\text{Gain in dB} = 1 / 1 - |S_{11}|^2 * |S_{21}|^2 * 1 / 1 - |S_{22}|^2$$

$$= 1.214*2.313*1.924$$
$$= 5.403 \text{ or } 7.3 \text{ dB}$$

NB Approx only because this neglects S_{12} , first stage gain approx.7dB
The reflection coefficient was first converted to an equivalent impedance;

$$\frac{Zo(1-|\Gamma o|^2) + j2Zo.\sin\angle\Gamma o}{1+|\Gamma o|^2 - 2|\Gamma o|.\cos\angle\Gamma o} = Zsn$$

$$= \frac{50(1-|0.42|) + j2*50*0.42.\sin -160}{1+|0.42| - 2|0.42|\cos -160}$$

$$= \frac{41.48 - j14.36}{1.9657} = 20.949 - j7.303\Omega$$

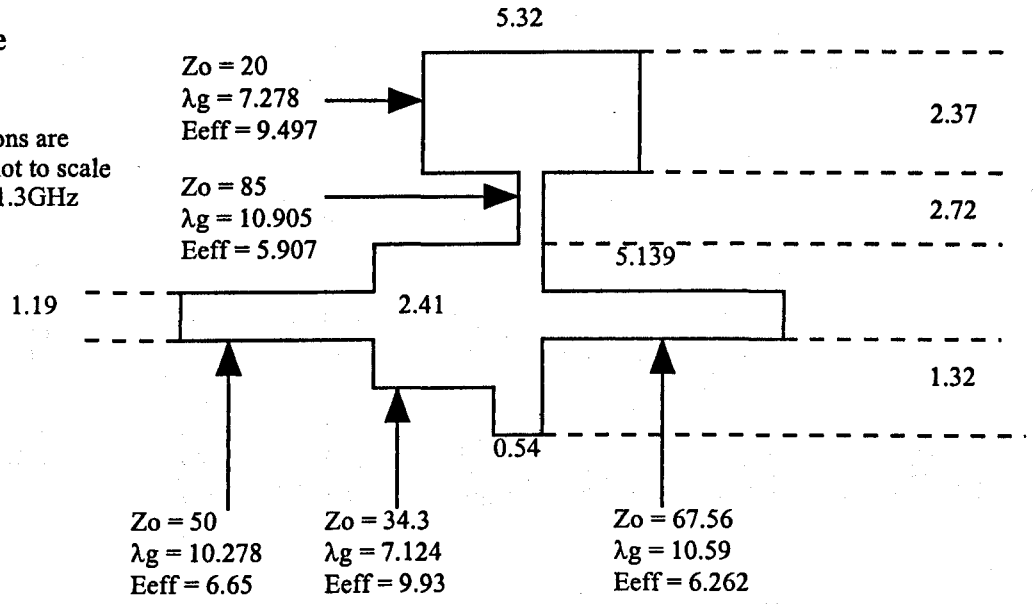
**** AC ANALYSIS TEMPERATURE = 27.000 DEG C

FREQ	VDB(4)
5.000D+08	-6.062D+00
5.236D+08	-6.066D+00
5.482D+08	-6.071D+00
5.741D+08	-6.077D+00
6.011D+08	-6.083D+00
6.295D+08	-6.090D+00
6.591D+08	-6.098D+00
6.902D+08	-6.107D+00
7.227D+08	-6.117D+00
7.568D+08	-6.128D+00
7.924D+08	-6.140D+00
8.298D+08	-6.154D+00
8.689D+08	-6.170D+00
9.099D+08	-6.187D+00
9.527D+08	-6.207D+00
9.976D+08	-6.228D+00
1.045D+09	-6.252D+00
1.094D+09	-6.279D+00
1.145D+09	-6.309D+00
1.199D+09	-6.343D+00
1.256D+09	-6.380D+00
1.315D+09	-6.423D+00
1.377D+09	-6.470D+00
1.442D+09	-6.525D+00
1.510D+09	-6.589D+00
1.581D+09	-6.666D+00
1.656D+09	-6.762D+00
1.734D+09	-6.884D+00
1.815D+09	-7.045D+00
1.901D+09	-7.264D+00
1.991D+09	-7.566D+00
2.084D+09	-7.985D+00
2.183D+09	-8.558D+00
2.285D+09	-9.320D+00
2.393D+09	-1.030D+01
2.506D+09	-1.150D+01

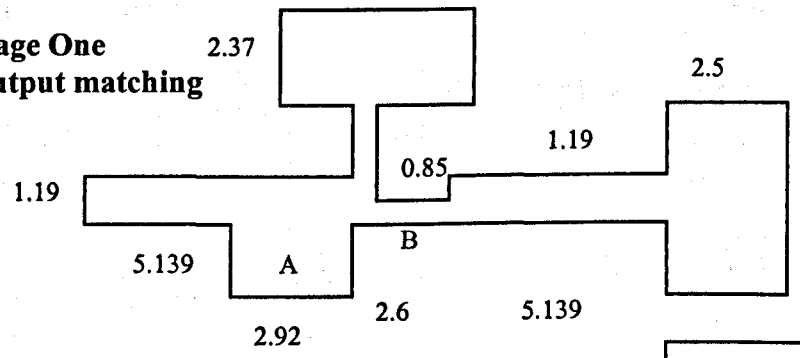
-2.000D+01 -1.500D+01 -1.000D+01 -5.000D+00

Stage One Input

All dimensions are in mm and not to scale substrate at 1.3GHz



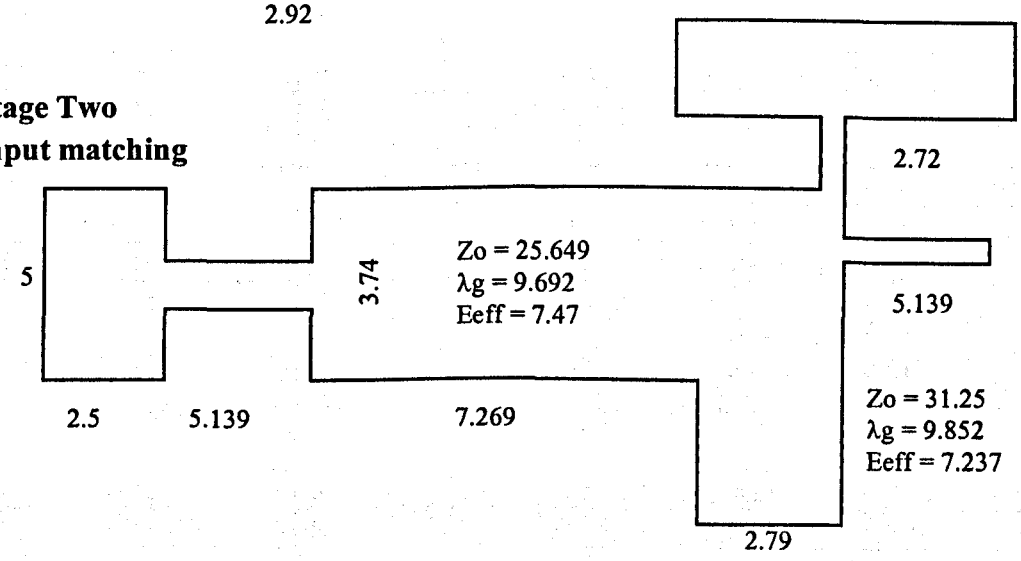
Stage One Output matching



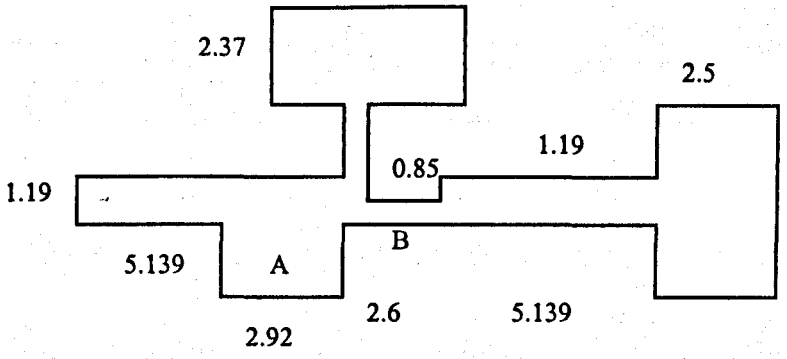
A= Zo = 30.3, λg = 9.827, Eeff = 7.274

B= Zo = 57.73, λg = 10.422, Eeff = 6.467, Else as above

Stage Two Input matching



Stage Two Output matching



A= Zo = 30.3, λg = 9.827, Eeff = 7.274

B= Zo = 57.73, λg = 10.422, Eeff = 6.467, Else as above

Appendix 13 and 14
Set-up routines

There are eight sets of registers to be initialised at the start of the decode.

i)	Global Control	@	808040
ii)	FSX/DX/CLKX	@	808042
iii)	FSR/DR/CLKR	@	808043
iv)	Data Transmit	@	808048
v)	Data Receive	@	80804c
vi)	Rx/Tx Timer control	@	808044
vii)	Rx/Tx Timer control	@	808045
viii)	Rx/Tx Timer period	@	808046

i)serial port global summary

Output	Bit No.	Acronym	Operation
0	0	RRDY	if = 1 new data is ready edge sets RINT
0	1	XRDY	if = 1 last bit shifted out sets Xint
1	2	FSXOUT	config FSX pin
0	3	XSEMPTY	Tx shift reg empty
0	4	RSRFULL	Rx overun if = 1
0	5	HS	Handshake enabled
1	6	XCLKSRCE	=1 internal Tx clock
0	7	RCLKSRCE	=1 internal Rx clock
0	8	XVAREN	Fixed rate = 0 Tx
0	9	RVAREN	Fixed rate = 0 Rx
1	10	XFSM	continuous = 0
1	11	RFSM	continuous = 0
0	12	CLKXP	check polarity, 0 = active high
0	13	CLKRP	check polarity, 0 = active high
0	14	DXP	check polarity, 0 = active high
0	15	DRP	check polarity, 0 = active high
0	16	FSXP	check polarity, 0 = active high
0	17	FSRP	check polarity, 0 = active high
1	18	XLEN	Transmit length =32
1	19	XLEN	Transmit length =32
1	20	RLEN	Receive length =32
1	21	RLEN	Receive length =32
0	22	XTINT	Tx timer interupt enabled = 1
0	23	XINT	Tx interupt enabled = 1
0	24	RTINT	Rx timer interupt enabled = 1
0	25	RINT	Rx interupt enabled = 1
1	26	XRESET	Tx reset = 0
1	27	RRESET	Rx reset = 0
0	28	RESERVED	Reserved
0	29	RESERVED	Reserved
0	30	RESERVED	Reserved
0	31	RESERVED	Reserved

The other register set ups can be obtained from the software routine in the set-up.

Appendix 15

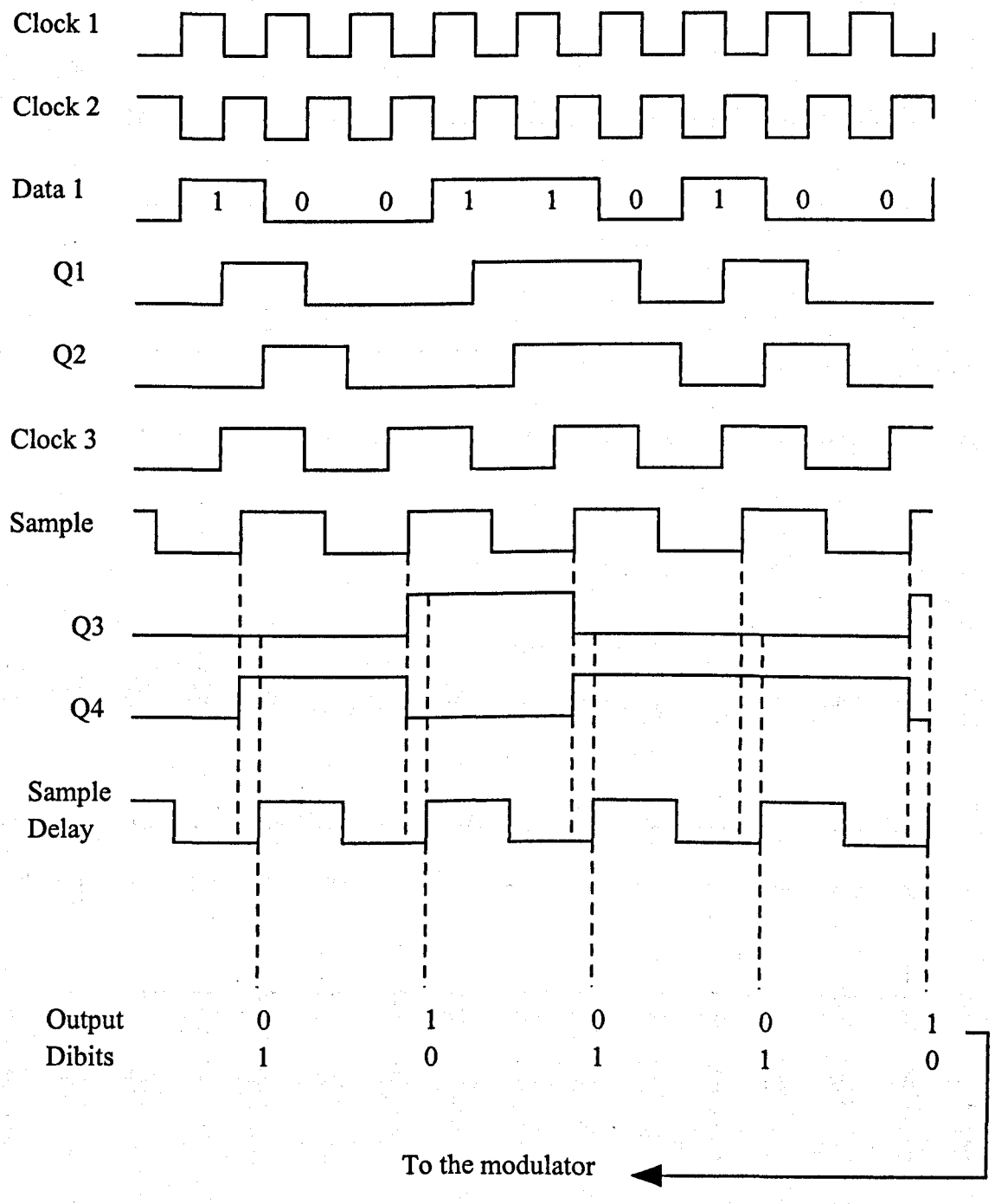
The output sequence to bypass the converter is as follows;

1	00
2	f2
3	f1
4	f0
5	f0
6	f0
7	13
8	20
9	40
a	11
b	42
c	11
d	54
e	12
f	61
10	e2
11	10
12	20
13	22
14	06
15	13
16	52
17	04
18	92
19	28
1a	63
1b	24
1c	28
1d	18
1e	00

Then repeat the sequence

Appendix 16

Timing Transmit Data



Q3 and Q4 provide the input data to the modulator, sampled on the positive going edge of delayed clock 3, delay = 0.13ms

Input data must be clocked in using clock 2, which is derived from the main synthesiser

Appendix 16

ReceiveTiming

Simplified to show general relationships