Digital Audio Broadcasting (Eureka 147) : Orthogonal Frequency Division Multiplexing (OFDM) : Quarternary Phase Shift Keying (QPSK) with Guardband Interval Modulation for Digital Audio Broadcasting (DAB)

van der Plaats, J.C.

Award date:
1989

5536

FACULTY OF ELECTRICAL ENGINEERING
EINDHOVEN UNIVERSITY OF TECHNOLOGY
TELECOMMUNICATION DIVISION

DIGITAL AUDIO BROADCASTING

(EUREKA 147)

Orthogonal Frequency Division Multiplexing (OFDM)
Quarternary Phase Shift Keying (QPSK)
with Guardband Interval Modulation
for Digital Audio Broadcasting (DAB)

by J. C. van der Plaats

Report of graduation work carried out from August 1988 to August 1989 at the Nederlandse Philips Bedrijven, Consumer Electronics, Advanced Developement Centre, Broadcasting Laboratory.

Professor : prof. ir. J. van der Plaats
Supervisors: ir. C. R. de Graaf (Philips)
ir. A. P. Verlijsdonk

ABSTRACT

This report treats the building of a Digital Audio Broadcasting (DAB) transmitter simulator at the Nederlandse Philips Bedrijven, Consumer Electronics, Advanced Developement Centre, Broadcasting Laboratory. This simulator is capable of generating a real-time Orthogonal Frequency Division Multiplexed Quarternary Phase Shift Keying (OFDM/QPSK) signal with guardband interval according to the four different systems proposed within the DAB project (Eureka 147). This OFDM/QPSK with guardband interval modulation technique, proposed by the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT), is very well suited to be used in the mobile transmission channel. A detailed theoretical and practical analysis of the OFDM/QPSK with guardband interval signal generation has been made. This analysis has been used for the DAB transmitter simulator design. Two of the four OFDM/QPSK with guardband interval modulation systems, generated by the simulator, are specially designed to share the same frequency channels already allocated to geographically separated co-channel television transmitters. For these two systems (System-3 and System-4) the minimal TV signal to DAB interference power-ratio $(S_{tv}I_{dab}R)_{min}$ for invisible TV picture distortion has been measured. This power-ratio is an important parameter for the DAB frequency allocation planning. For both systems the same value of $(S_{tv}I_{dab}R)_{min}$ has been found:

$$(S_{tv}D_{dab}R)_{min} = 54 \pm 4 \text{ dB.}$$

The inaccuracy in this value is mainly caused by the subjective TV picture distortion judgement.

GLOSSARY OF ABBREVIATIONS

ADC	Analog to Digital Converter
CCETT	Centre Commun d'Etudes de Télédiffusion et Télécommunications
CD	Compact Disc
COFDM	Coded Orthogonal Frequency Division Multiplexing
CSRS	Cyclotomatically Shortened Reed-Solomon
DAB	Digital Audio Broadcasting
DAC	Digital to Analog Converter
DBS	Direct Broadcasting Satellite
DC	Direct Current
DEQPSK	Differentially Encoded Quarternary Phase Shift Keying
DFT	Discrete Fourier Transform
DQPSK	Differential Quarternary Phase Shift Keying
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FM	Frequency Modulation
FSK	Frequency Shift Keying
IC	Integrated Circuit
IDFT	Inverse Discrete Fourier Transform
IF	Intermediate Frequency
I/O	Input/Output
MASCAM	Masking-pattern Adapted Sub-band Coding And Multiplexing
OFDM	Orthogonal Frequency Division Multiplexing
PC	Personal Computer
PCB	Printed Circuit Board
PPI	Programmable Parallel Interface
PSK	Phase Shift Keying
QPSK	Quarternary Phase Shift Keying
RAM	Random Access Memory
RDAT	Rotary head Digital Audio Tape
TV	Television
VHF	Very High Frequency
UHF	Ultra High Frequency

GLOSSARY OF SYMBOLS

B_d	Doppler spread
C_{tv}	unmodulated TV vision carrier power
C_{dab}	unmodulated DAB carrier power
F_k	amplitude compensation factor for carrier k
i	the signalling interval number
I	oversampling ratio
k	the carrier number
M	the number of complex points in the DFT or IDFT processing
N	total number of carriers (emitted + virtual)
$S_{dab}I_{tv}R$	DAB signal to TV interference unmodulated power-ratio
$S_{tv}I_{dab}R$	TV signal to DAB interference unmodulated power-ratio
T	sampling period
T_m	multipath spread or delay spread
T_s	useful signal duration
T'_s	total signal duration
Δ	guardband interval duration
$(\Delta f)_c$	coherence bandwidth
$(\Delta t)_c$	coherence time
ϕ_k	the phase of carrier k when $\theta_{i,k} = 0$
$\theta_{i,k}$	the modulation phase of carrier k in signalling interval i
σ	standard deviation
ξ_k	indicates if carrier k is emitted ($\xi_k = 1$) or not ($\xi_k = 0$)

TABLE OF CONTENTS

1. INTRODUCTION

The introduction of digital sound storage media, such as Compact Disc (CD) and Rotary head Digital Audio Tape (RDAT), in the domestic consumer market has lead to wider public appreciation of high-quality sound. In order not to loose the attractiveness of their services, sound radio broadcasters must adjust their services to the new standard for high-quality sound set by these new storage media. The VHF/FM services being used now can still provide moderate stereophonic sound service to fixed home receivers (using a directional antenna mounted at house-roof height). But the sound quality of these services will generally be poor in the case of mobile or portable reception, especially in urban, suburban or mountainous rural areas. This low sound quality is caused by shadowing (e.g. large buildings between transmitter and receiver antenna), multipath propagation (caused by wave reflections e.g. against mountains), time variance (caused by moving environment and/or moving receiver) and the use of an omni-directional receiving antenna.

The solution to the problem of raising the sound quality of sound radio broadcasting services is to develop an entirely new digital sound radio broadcasting service with a complete digital sound program chain from studio to receiver. In Federal Republic of Germany such a system has already been developed for use in the 12 GHz Direct Broadcasting Satellite (DBS) services band [1]. However, this system is only suitable for fixed home reception as a consequence of the high carrier frequency and the modulation scheme (DEQPSK) used. Therefore it will not satisfy all the needs of the future sound radio audience, which will expect to be able to receive a stereophonic sound service in vehicles on the move and on portable receivers, in addition to fixed home reception. The increasing importance of traffic information and the increasing use of cable radio in the case of fixed home reception are two other aspects that demonstrate the importance of mobile reception for sound radio broadcasting over the air. Therefore it is necessary to develop a new digital sound radio broadcasting system designed to meet all the reception requirements.

The European Eureka "Digital Audio Broadcasting (DAB)" joint research and development project is committed to developing such a new sound radio system and its enabling technology. The major starting points for this project were:

- the use of terrestrial transmitters,
- the sound quality delivered must be comparable to that of Compact Disc,
- the spectral requirement of the system must be comparable to that of the existing FM system,
- the system must be suited to fixed, portable and mobile reception,
- price setting for the consumer market.

This report treats the building of a DAB transmitter simulator, that is capable of generating a real-time Orthogonal Frequency Division Multiplexed Quarternary Phase Shift Keying (OFDM/QPSK) signal with guardband interval. This OFDM/QPSK with guardband interval modulation technique, proposed by the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT) within the Eureka DAB project, is very well suited to be used in the mobile transmission channel. Chapter 2 gives an introduction to the DAB transmission system. This chapter in fact is a summary of the literature study that was made during this project. Chapter 3 gives a detailed theoretical and practical analysis of the OFDM/QPSK with guardband interval modulation using digital signal processing techniques, including the Inverse Fast Fourier Transform (IFFT). This analysis was made in order to achieve a proper and flexible DAB transmitter simulator design. In Chapter 4 the DAB transmitter simulator, which was built during this project, is described. The hardware, the software and the use of this simulator are treated. Two OFDM/QPSK with guardband interval modulation systems, generated by the simulator, are specially designed to share the same frequency channels already allocated to geographically separated co-channel TV transmitters. For these two systems the minimal TV signal to DAB interference unmodulated power-ratio for no visible TV picture distortion has been measured. The measurement method and the results are described in Chapter 5.

2. INTRODUCTION TO THE DAB TRANSMISSION SYSTEM

2.1. Introduction

This chapter gives a brief introduction to the development of a new digital transmission system suitable for fixed, portable and mobile reception. The development of such a transmission system is a necessary step for the European Eureka DAB project to be successful.

2.2. Source coding

The first problem to be solved in the DAB transmission system development is the source coding. Digitizing a monophonic sound signal using a 16 bit Analog to Digital Converter (ADC) and a 48 kHz sample frequency gives a bit-rate of 768 kbit/s. In order to make most efficient use of the scarce radio frequency spectrum, it is necessary to reduce this bit rate to a minimum. However, the full subjective sound quality of the original studio signal should be conserved. The bit-rate reduction will be too small if only the redundancy is taken out of the signal. However, further bit-rate reduction is possible by taking irrelevant information, based on the perception characteristics of the human ear, out of the signal.

Transform coding and Sub-band coding [2] are two different source coding techniques, both taking irrelevance out of the signal. Recent progress in these sound coding compression techniques has shown that it is possible to reduce the bit-rate of the sound signal to around 100 kbit/s. The bit-rate reduction factor and the complexity of the circuits are the major weighting factors in the evaluation of the different sound coding techniques. The sensitivity of each source coding system to transmission errors can also be important for channel coding and modulation schemes, whereby the error performance gradually degrades when the noise level is increased. However, when a power and bandwidth efficient channel coding and modulation scheme near the Shannon limit is used, this sensitivity is not important. This is due to the fact that such a channel coding and modulation scheme is characterized by an abrupt curve for the bit error probability as a function of the signal-to-noise ratio per bit. This gives virtual error-free transmission and therefore perfect service if the signal-to-noise ratio per bit is above some threshold. Under this threshold almost no information can be transmitted which will totally interrupt the service.

2.3. Modulation and channel coding

The second problem is to find a modulation and channel coding scheme adapted to the hostile transmission properties of the mobile radio channel. The mobile radio channel is characterized by multipath propagation and time variance. Therefore, the channel transfer function H(f,t) is time variant and random. This is illustrated in Figure 2.1.

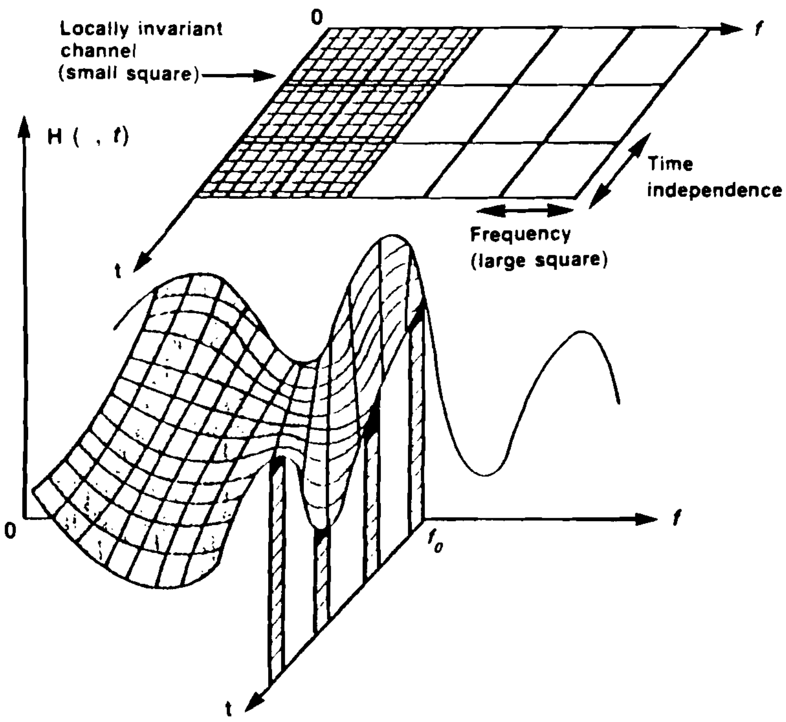


Figure 2.1. Frequency-time response for the mobile radio channel [3].

The main statistical properties of a mobile radio channel are its multipath spread (or delay spread) T_m and its Doppler spread B_d [4]. The multipath spread is mainly dependent on the environment of the receiver and the Doppler spread is mainly determined by the speed of the moving receiver, the speed of nearby moving objects and on the carrier frequency used.

The reciprocal of the multipath spread T_m is a measure of the coherence bandwidth $(\Delta f)_c$ of the channel:

$$(\Delta f)_c \approx \frac{1}{T_m} \tag{2.1}$$

and the reciprocal of the Doppler spread B_d is a measure of the coherence time $(\Delta t)_c$ of the channel:

$$(\Delta t)_c \approx \frac{1}{B_d} \ . \tag{2.2}$$

The coherence bandwidth and the coherence time both give information on how the mobile channel affects the transmitted digitally modulated signal. When signals with a bandwidth greater than the coherence bandwidth of the channel are used, the frequency components of the signals are subjected to different gains and phase shifts. In this case the channel is said to be frequency-selective. On the contrary all frequency components are equally affected by the channel, if signals are used with a bandwidth much smaller than the coherence bandwidth of the channel. Then the channel is said to be frequency-nonselective.

Besides frequency-selectivity there is another type of distortion caused by the time variations in the channel characteristics. This type of distortion is evidenced as a variation in the received signal strength and has been termed fading. In the least favourable case this signal strength is Rayleigh distributed [5]. Consider the use of signals with a duration much smaller than the coherence time of the channel. Then the attenuation and phase shift for every frequency component of the signal will be essentially fixed for at least one signaling interval. The channel is said to be slowly-fading in this case. On the other hand if signals with a duration much larger than the coherence time of the channel are used, the channel characteristics will change during one signaling interval and the channel is said to be fast-fading.

Consider the use of a conventional digital modulation scheme without channel coding (e.g. PSK, QPSK, FSK) over a multipath time variant channel. In urban areas the multipath spread is found to be of the order of several microseconds and the Doppler spread will be of the order of

100 Hz. In order to transmit at least one stereophonic sound service a bit-rate of approximately 250 kbit/s is needed. Therefore the channel can generally be modeled by the frequency-selective slowly-fading Rayleigh channel. Due to the frequency-selective nature of the channel it will introduce inter-symbol interference. This inter-symbol interference will generally limit the bit error probability to a minimal value, depending on the modulation scheme used and the ratio $T_r = T_m / T_s$ between the multipath spread T_m and the duration T_s of a modulated symbol. This is illustrated for PSK in Figure 2.2 for different values of T_r.

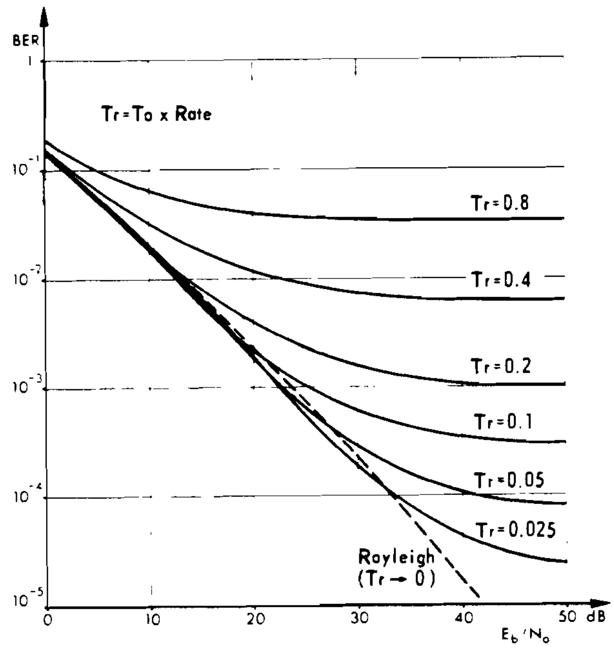


Figure 2.2. Bit error probability for different values of T_r as a function of the signal-to-noise ratio per bit for PSK in a selective Rayleigh channel [5].

The greater the ratio T_r, the higher is the minimum error probability. When T_s becomes large compared with T_m, the error probability curve tends to that of the frequency-nonselective slowly-fading Rayleigh channel. In order to get a bit error probability of 10^{-9} (virtual error-free channel), in the case of a frequency-nonselective slowly-fading Rayleigh channel with additive white Gaussian noise, a for DAB unacceptable high transmitting power is needed.

Therefore, if a conventional modulation scheme without channel coding is used over the mobile radio channel, it will be impossible (frequency-selectivity) or power inefficient to get a bit error probability of 10^{-9}. Thus, these modulation schemes are unacceptable for the DAB system.

A well-known solution to the frequency-selective fading channel is to employ spread spectrum techniques, which permit the temporal separation of the signals corresponding to the different transmission paths [6]. The main weakness of these techniques is the low spectral efficiency attainable, usually below 0.25 bits/s/hertz. This efficiency is unacceptable for broadcasting, where severe restrictions on available spectrum apply.

A very promising modulation scheme for DAB is the use of Orthogonal Frequency Division Multiplexing Differential Quarternary Phase Shift Keying (OFDM/DQPSK) with a guardband interval [7]. This modulation scheme allows the problem of inter-symbol interference to be solved by splitting the wide-band highly-frequency-selective channel into a large number of small-band low-frequency-selective sub-channels. Increasing the number of sub-channels will asymptotically eliminate the inter-symbol interference caused by frequency-selectivity, but it will introduce the problem of fast fading. Another solution, used for the DAB transmission system, consists of preceding each signalling interval by a guardband interval which absorbs the inter-symbol interference. This eliminates the problem of channel frequency-selectivity.

The introduction of this guardband interval reduces the power efficiency and the spectral efficiency compared to those of the original OFDM/DQPSK. However, this disadvantage is largely compensated by the system advantages in a mobile transmission channel.

For OFDM/DQPSK with a guardband interval, there remains the problem of fading. The amplitude of each of the sub-channels will follow a Rayleigh law or, if there is a direct path, a Rice-Nakagami law. But if the total signal bandwidth is much greater than the coherence bandwidth, it is very unlikely that all the carriers will fade simultaneously. Therefore it will always be possible to transmit information.

The error performance of OFDM/DQPSK with guardband interval in a frequency-selective slowly-fading Rayleigh channel is equal to that of DQPSK in a frequency-nonselective slowly-fading Rayleigh channel. It is clear that this error performance is not suitable for the DAB transmission system either. However, with the use of channel coding it is possible to improve this error performance while reducing the effective spectral efficiency. If the useful signal duration $T_s = 4 \cdot \Delta$, where Δ is the guardband duration and the number of sub-channels is large, OFDM/QPSK with a guardband interval has a basic spectral efficiency of 1.6 bits/s/hertz. Figure 2.3 [7] shows the error performance for an uncoded system and for a system which uses a convolutional code with efficiency $\frac{1}{2}$ combined with soft decision maximum likelihood decoding (Viterbi decoder). This figure makes clear that a good error performance can be achieved for an acceptable spectral efficiency of 0.8 bits/s/hertz.

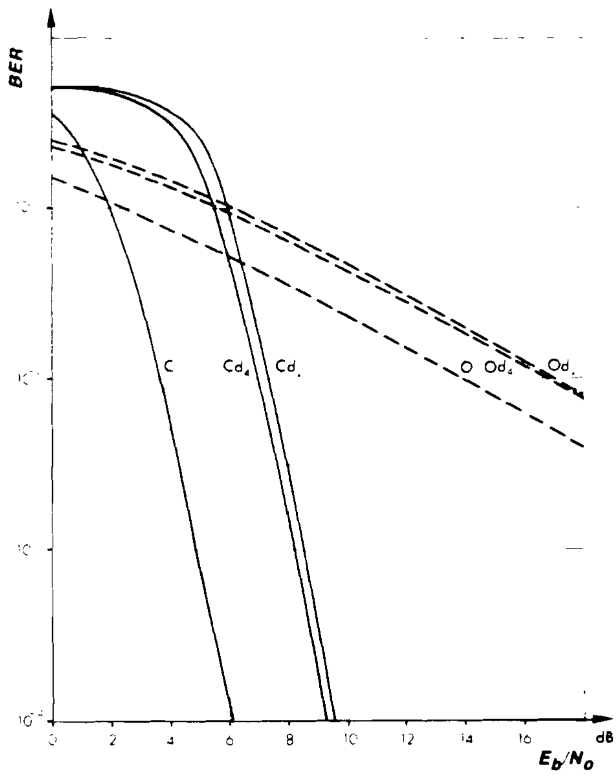


Figure 2.3. Comparison of the bit error-ratio performances of OFDM/PSK, OFDM/QPSK, OFDM/DPSK (d_2) and OFDM/DQPSK (d_4) in a Rayleigh channel without (O) and with the use of a convolutional code (C) with efficiency $\frac{1}{2}$ [7].

Figure 2.4. [7] shows the error performance for a concatenated code, where the inner convolutional code is equal to that of Figure 2.3 and the outer code is a Cyclotomatically Shortened Reed-Solomon (CSRS) code with efficiency $\frac{18}{21}$ giving a total coding efficiency of $\frac{9}{21}$. This method of channel coding clearly gives a better error performance than the system of Figure 2.3 for a slightly decreased spectral efficiency of ≈ 0.69 bits/s/hertz. It gives a virtual error-free channel when the mean signal-to-noise ratio per useful bit exceeds ≈ 7.5 dB.

The nature of the mobile radio channel tends to create dependency between successive decision variables. The results from Figure 2.3 and 2.4 are based on the assumption that a time and frequency interleaving arrangement effectively guarantees the conditions of independence required by the Viterbi decoder.

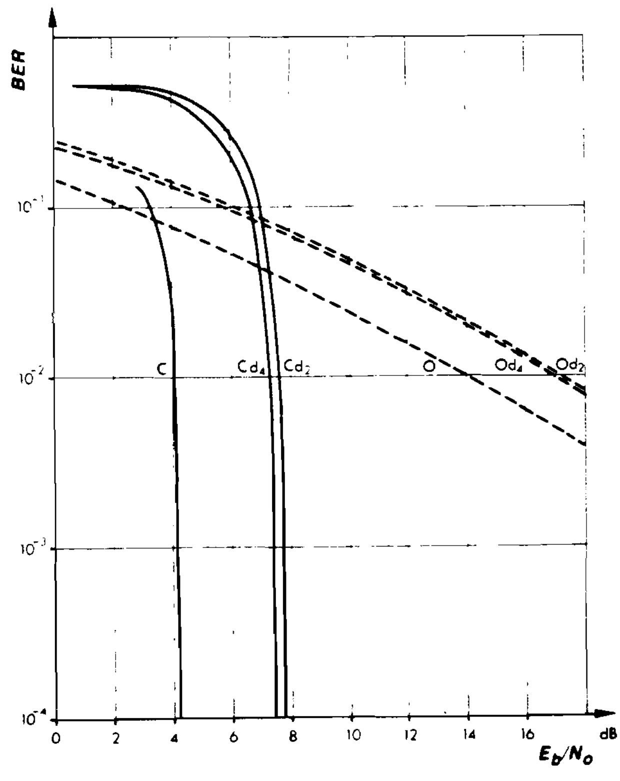


Figure 2.4. Comparison of the bit error-ratio performances of OFDM/PSK, OFDM/QPSK, OFDM/DPSK (d_2) and OFDM/DQPSK (d_4) in a Rayleigh channel without (O) and with the use of a concatenated code (C) with efficiency $\frac{9}{21}$ [7].

The choice of differential over coherent demodulation is based on the expectation that carrier recovery will be very difficult after transmission over the mobile radio channel.

The use of OFDM instead of FDM has the advantage that the modulation and the demodulation can be realized using digital signal processing techniques. Modulation is done by calculating one Inverse Discrete Fourier Transform (IDFT) over N complex points within each total signalling interval duration $T'_s = T_s + \Delta$, whereby N is greater than the number of carriers used. Demodulation consists of one N complex point Discrete Fourier Transform (DFT) every signalling interval.

A demonstration (Geneva, september 1988) [8] of a system called COFDM/MASCAM based on sub-band source coding, convolutional channel coding and OFDM/DQPSK modulation with guardband interval, has shown the technical feasibility of such a system and the high mobile reception quality attainable with such a system.

2.4. Frequency planning

At a time when there is growing interest in increasing the choice of sound radio services available, it is of great concern that there is, as yet, no future radio spectrum capacity available for accommodating a new sound broadcasting system. Spectrum considerations for COFDM have therefore been spread fairly broadly and possible options include the broadcasting bands I, III, IV and V, part of the aeronautical radio-navigation band immediately above band II and a potential new upper UHF band located somewhere between 1 and 2 GHz, particularly suitable for satellite delivery.

One of the major problems in considering the existing television bands is that it will be very difficult to gain any common channels across Europe and in many cases possibly none at all, unless a system can be engineered to share with television at relatively short separation distances.

A minimum block of bandwidth is required to provide the necessary independence of multipath fading but typically up to 16 high-quality stereophonic programs may be accommodated in a 8 MHz wide band. However, in some of the possible bands only 4 - 8 MHz of spectrum might be available and this gives very limited options for planning where different services are desired in adjacent geographical areas, e.g. in the border regions between neighbouring countries. Conventional area coverage terrestrial planning requires some 10 - 20 times the amount of the basic spectrum required for a single transmitter.

Nevertheless, national coverage may be obtained using a satellite of modest power in the 0.5 - 2 GHz frequency range with a theoretical minimum requirement of 7 times the basic spectrum block for regular coverage beams conveying different services. Alternatively and interestingly, a single frequency network of terrestrial transmitters may be used in the 50 - 250 MHz frequency range. Because the receiver is able to cope with multipath signals, it does not, in principle, know which transmitter(s) the signals have come from. Furthermore, any holes in the coverage area may be filled in by providing a local low-power transmitter on the same frequencies as those in the network. This novel feature may also find application in providing re-broadcast service within buildings and other heavily shielded areas.

A further important factor is the significant reduction in interference protection required by these digital systems and their reduced potential to interfere with other radio communication services because of their uniform spectral characteristics. This offers better prospects of squeezing this broadcasting service into an already congested radio frequency spectrum. For example, local coverage might be obtainable by using terrestrial transmitters sharing the same spectrum already allocated to television services. By judicious choice of the sub-channel carrier spacing, to be related to television line frequency, it may be possible to interleave directly with geographically separated co-channel television transmitters, possibly using only part of the 8 MHz television channel with a reduced number of digital sound programs.

Within the DAB project, four different OFDM modulation schemes have been proposed. These four modulation schemes are named System-1, System-2, System-3 and System-4. System-2, the reference system, is designed for conventional area coverage terrestrial planning. This reference system is the one used at the Geneva demonstration. System-1 is a modulation scheme suitable for a single frequency network. Because a very large guardband interval is needed to let the signals from different transmitters combine constructively, this system is characterized by a very long total signalling interval and a very large number of carriers. System-3 and System-4 are both intended to be used for sharing the spectrum with television broadcasting. Both systems are characterized by a comb-shaped frequency power spectrum, with a spacing between the peaks of 15625 Hz, which equals the television line frequency. A detailed description of the four OFDM modulation schemes is given in Appendix A.

2.5. Draft DAB transmission system

In this section a draft DAB transmission system is described. This draft system, designed by the Centre Commun d'Etudes de Télédiffusion et Télécommunications (CCETT) in France, is an extension of the system used at the Geneva demonstration. Figure 2.5 shows the transmitter.

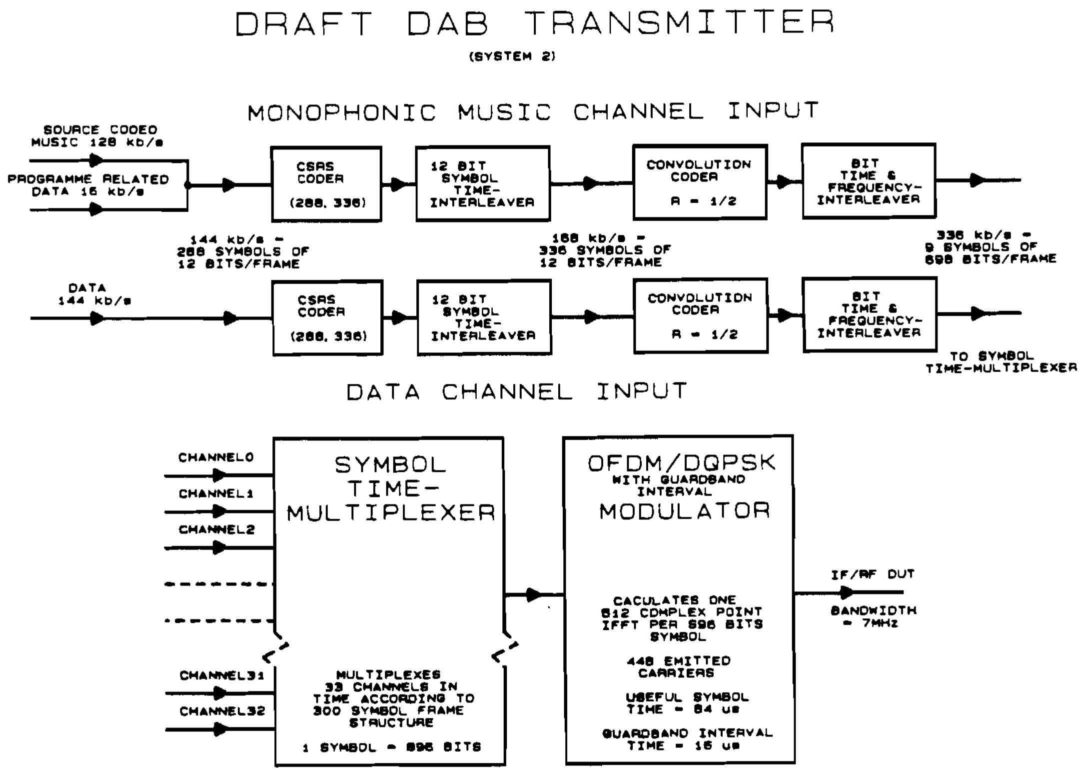


Figure 2.5. The draft DAB transmitter.

The transmitter is capable of transmitting 33 channels of 144 kbit/s each. One of these channels is used to transmit data (144 kbit/s) and the other 32 channels are used to transmit a monophonic source coded music signal (128 kbit/s) together with program related data (16 kbit/s). Two monophonic channels can be used to transmit one stereophonic music signal. Therefore 16 stereophonic programs can be transmitted simultaneously.

In order to reduce the receiver complexity, the channel coding and modulation is done separately for every channel. This enables separate reception of one or more channels without having to decode the other channels. A concatenated code with coding efficiency $\frac{9}{21}$ is used. The inner code is a convolutional code with efficiency $\frac{1}{2}$ and the outer code is a CSRS code with efficiency $\frac{18}{21}$.

The system uses a frame structure for the transmitted signal. This frame structure is needed for program selection, synchronization, phase reference and channel noise measurement. The signalling frame structure is shown in Figure 2.6.

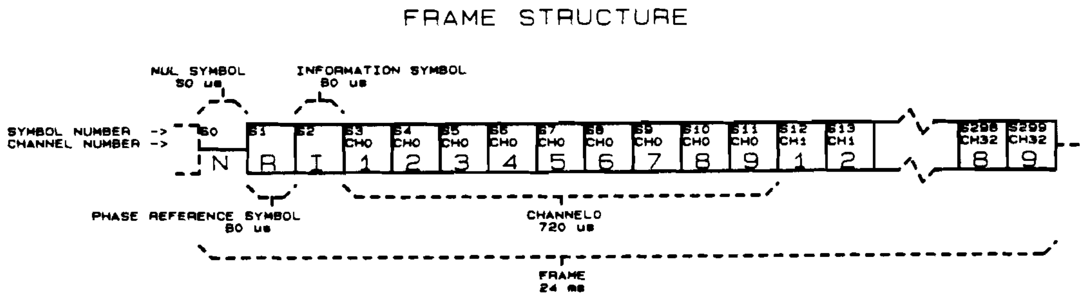


Figure 2.6. The signalling frame structure.

The frame has a length of 300 signalling intervals. Each signalling interval contains a guardband interval of 16 μs and a useful signal of 64 μs. Therefore the total frame duration equals 24 ms. The signal within each signalling interval is an OFDM/DQPSK modulated symbol of 896 bits (according to System-2, see appendix A). For each of the 33 channels, 9 successive symbols of 896 bits are reserved within the frame. The remaining three signalling intervals are reserved for the null symbol, the phase reference symbol and the information symbol.

The first signalling interval is used by the null symbol. During the null symbol signalling interval no signal is transmitted. This null symbol is used to estimate the channel noise of each frequency multiplexed sub-channel. Since the signal of the null symbol introduces a dip in the received energy, it can be used for course frame synchronization in the receiver.

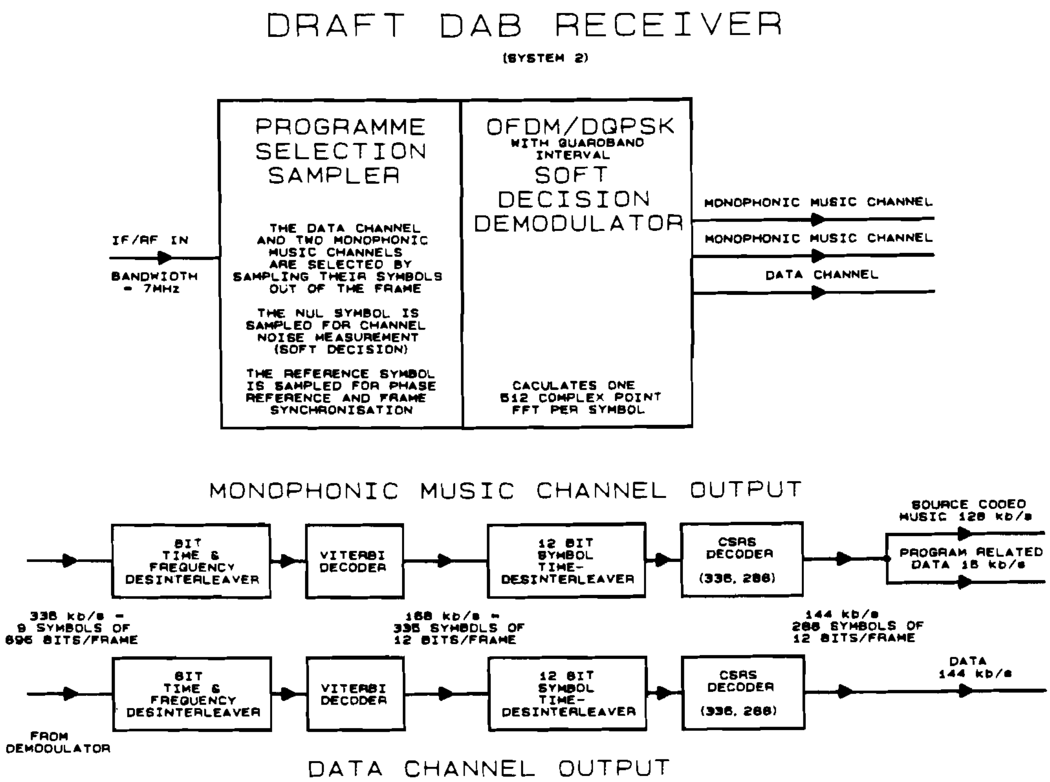


Figure 2.7. The draft DAB receiver.

The second signalling interval is used by the phase reference symbol. The signal of the phase reference symbol is used as a reference for the differential demodulation within each frequency multiplexed sub-channel. The reference symbol signal is also used for accurate frame synchronization in the receiver. The reference signal is totally known by the receiver and its bandwidth is much greater than the reciprocal of the signal duration. Therefore, it can be used to estimate the channel's impulse response with a time resolution much smaller than the symbol duration. This impulse response estimation is highly noise resistant. Analysis of this impulse response can be used to synchronize the receiver to the signal received via the path with the lowest attenuation.

The third signalling interval is used by an information symbol. This information symbol contains static data on e.g. the transmitter, alternative frequencies etc.

Figure 2.7 shows the receiver for this system. This receiver is capable of processing the data channel and two monophonic channels (or one stereophonic channel) simultaneously. Besides the null symbol, the phase reference symbol and the information symbol, only the signals belonging to the symbols of the selected music channels and the data channel are sampled, stored and processed. A 512 complex point Fast Fourier Transform (FFT) algorithm takes care of the matched filtering for each frequency multiplexed sub-channel.

3. OFDM/(D)QPSK MODULATION WITH GUARDBAND INTERVAL

3.1. Introduction

When using the OFDM/(D)QPSK signalling scheme with a useful signal duration T_s and guardband interval Δ, the modulator has to generate the bandpass signal s(t)

$$s(t) = \left\{ \sum_{i=-\infty}^{\infty} \sum_{k=0}^{N-1} A_{i,k}\Phi'_{i,k,f_0}(t) - B_{i,k}\Theta'_{i,k,f_0}(t) \right\} * h_0(t), \qquad (3.1)$$

in which

$$\Phi'_{i,k,f}(t) = \begin{cases} \cos 2\pi(f + \frac{k}{T_s})t, & iT'_s - \Delta \leqslant t < (i+1)T'_s - \Delta \\ 0 & , \text{otherwise}, \end{cases} \qquad (3.2)$$

$$\Theta'_{i,k,f}(t) = \begin{cases} \sin 2\pi(f + \frac{k}{T_s})t, & iT'_s - \Delta \leqslant t < (i+1)T'_s - \Delta \\ 0 & , \text{otherwise}, \end{cases} \qquad (3.3)$$

$$A_{i,k} = \begin{cases} \sqrt{2}\cos(\theta_{i,k} + \phi_k), & \text{for the emitted carriers} \\ 0 & , \text{for the virtual carriers}, \end{cases} \qquad (3.4)$$

$$B_{i,k} = \begin{cases} \sqrt{2}\sin(\theta_{i,k} + \phi_k), & \text{for the emitted carriers} \\ 0 & , \text{for the virtual carriers}, \end{cases} \qquad (3.5)$$

$$T'_s = T_s + \Delta, \qquad (3.6)$$

where

- i : is the signalling interval number, $i \varepsilon \{-\infty, \infty\}$,
- k : is the carrier number, $k \varepsilon \{0, 1, \ldots, N-1\}$,
- $\theta_{i,k}$: is the modulation phase of carrier k in signalling interval i, $\theta_{i,k} \varepsilon \{-\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi\}$,
- ϕ_k : is the phase of carrier k when $\theta_{i,k} = 0$,

and where $h_0(t)$ is the impulse response of a bandpass filter with Fourier transform $H_0(f)$

$$H_0(f) = \begin{cases} 1, & \text{for } f_0 \leqslant |f| \leqslant f_0 + N/T_s \\ 0, & \text{otherwise.} \end{cases} \qquad (3.7)$$

The phase $\theta_{i,k}$ for each emitted carrier and for each signalling interval is determined by two data bits in the case of QPSK. In the case of DQPSK the phase $\theta_{i,k}$ is determined by two data bits and the phase $\theta_{i-1,k}$ of carrier k in the previous signalling interval.

The draft DAB transmission system, introduced in chapter 2, uses a frame structure with a null and a reference symbol. In order to generate the signal of the null symbol, A_k and B_k are set to zero for all k. The reference symbol is characterized by known modulation phases θ_k for all emitted carriers. An easy choice for the modulation phases of the reference signal is $\theta_k = 0$.

The bandpass filter $H_0(f)$ is necessary to eliminate interference in neighbouring channels. This filter can be seen as part of the radio channel. The bandlimited nature of this filter will cause the impulse response of the radio channel to be unlimited in time. So the use of a guardband interval Δ, when the multipath delay's are limited to Δ, will generally not totally eliminate the inter-symbol interference. This inter-symbol interference, caused by the bandpass filter, only is negligible when the symbol rate $1/T_s'$ is much smaller than the bandwidth N/T_s of the filter.

From a technological point of view, it is difficult to imagine the OFDM/(D)QPSK with guardband interval modulator to be built with analog mixers and oscillators if the number of emitted carriers is large. However, when some restrictions to N, Δ and f_0 are satisfied and some distortion in the generated signal is accepted, it is possible and relatively easy to generate the signal s(t) for large N using digital signal processing techniques including the Inverse Fast Fourier Transform (IFFT) [9, 10].

3.2. Theoretical analysis

This section treats the generation of the signal s(t) using the Inverse Discrete Fourier Transform (IDFT) over M complex points

$$M = I \cdot N \tag{3.8}$$

in which I is the oversampling ratio. The following restrictions are made

- N is a power of 2 equal to or greater than 4, $N = 2^{\alpha}$, $\alpha = 2, 3, \ldots,$
- I is a power of 2 equal to or greater than 1, $I = 2^{\beta}$, $\beta = 0, 1, \ldots,$
- the guardband time Δ is an integer multiple of T_s/N, $\Delta = \gamma T_s/N$, $\gamma = 1, 2, \ldots,$
- the carrier with frequency f_0 is virtual, therefore $A_{i,0} = B_{i,0} = 0$ for all i,
- the frequency f_0 is greater than $(M - N)/2T_s$.

In the first place two sequences of M real numbers for every i indicated as $\bar{A}_i$ and $\bar{B}_i$ are formed. The individual real numbers of $\bar{A}_i$ and $\bar{B}_i$ are respectively indicated as $\bar{A}_{i,m}$ and $\bar{B}_{i,m}$ with $m = 0, 1, \ldots, M-1$. The construction of $\bar{A}_i$ and $\bar{B}_i$ goes as follows

$$\begin{aligned}
\bar{A}_{i,m} &= 0 && , \text{ for } \quad 0 \leqslant m < \frac{M}{2} - \frac{M}{2I}, \\
\bar{A}_{i,m} &= A_{i,(m + \frac{M}{2I} - \frac{M}{2})} && , \text{ for } \quad \frac{M}{2} - \frac{M}{2I} \leqslant m < \frac{M}{2} + \frac{M}{2I}, \\
\bar{A}_{i,m} &= 0 && , \text{ for } \quad \frac{M}{2} + \frac{M}{2I} \leqslant m < M, \\
\bar{B}_{i,m} &= 0 && , \text{ for } \quad 0 \leqslant m < \frac{M}{2} - \frac{M}{2I}, \\
\bar{B}_{i,m} &= B_{i,(m + \frac{M}{2I} - \frac{M}{2})} && , \text{ for } \quad \frac{M}{2} - \frac{M}{2I} \leqslant m < \frac{M}{2} + \frac{M}{2I}, \\
\bar{B}_{i,m} &= 0 && , \text{ for } \quad \frac{M}{2} + \frac{M}{2I} \leqslant m < M.
\end{aligned} \tag{3.9}$$

An example of this construction for N = 4 and I = 4 is given below

$$\bar{A}_i = (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ A_{i,0},\ A_{i,1},\ A_{i,2},\ A_{i,3},\ 0,\ 0,\ 0,\ 0,\ 0,\ 0),$$

$$\bar{B}_i = (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ B_{i,0},\ B_{i,1},\ B_{i,2},\ B_{i,3},\ 0,\ 0,\ 0,\ 0,\ 0,\ 0).$$

It is clear that $\bar{A}_i = A_i$ and $\bar{B}_i = B_i$ when I = 1. With the use of $\bar{A}_i$ and $\bar{B}_i$ two new sequences a_i and b_i of M real numbers are calculated according to

$$a_{i,n} + jb_{i,n} = (-1)^n \cdot M \cdot \left\{ \mathrm{IDFT}_M(\bar{A}_{i,m} + j\bar{B}_{i,m}) \right\}_n,$$

$$n = 0,\ 1,\ \ldots,\ M-1, \tag{3.10}$$

where $\mathrm{IDFT}_M()$ stands for an Inverse Discrete Fourier Transform over M complex points. This gives

$$a_{i,n} + jb_{i,n} = (-1)^n \cdot \sum_{m=0}^{M-1} (\bar{A}_{i,m} + j\bar{B}_{i,m}) \cdot e^{\frac{j2\pi nm}{M}}, \text{ with } (-1)^n = e^{-jn\pi} \Rightarrow$$

$$a_{i,n} + jb_{i,n} = \sum_{m=0}^{M-1} (\bar{A}_{i,m} + j\bar{B}_{i,m}) \cdot e^{\frac{j2\pi n(m - M/2)}{M}}, \text{ with } m - \frac{M}{2} \rightarrow m \Rightarrow$$

$$a_{i,n} + jb_{i,n} = \sum_{m=-M/2}^{M/2-1} (\bar{A}_{i,m+M/2} + j\bar{B}_{i,m+M/2}) \cdot e^{\frac{j2\pi nm}{M}}, \text{ with Eq. 3.9} \Rightarrow$$

$$a_{i,n} + jb_{i,n} = \sum_{m=-N/2+1}^{N/2-1} (A_{i,m+N/2} + jB_{i,m+N/2}) \cdot e^{\frac{j2\pi nm}{IN}},$$

$$n = 0,\ 1,\ \ldots,\ M-1. \tag{3.11}$$

Therefore,

$$a_{i,n} = \mathrm{Re}\left\{ \sum_{m=-N/2+1}^{N/2-1} (A_{i,m+N/2} + jB_{i,m+N/2}) \cdot e^{\frac{j2\pi nm}{IN}} \right\} \Rightarrow$$

$$a_{i,n} = \sum_{m=-N/2+1}^{N/2-1} \left\{ A_{i,m+N/2} \cdot \cos(\frac{2\pi nm}{IN}) - B_{i,m+N/2} \cdot \sin(\frac{2\pi nm}{IN}) \right\} \Rightarrow$$

$$a_{i,n} = A_{i,N/2} + \sum_{m=1}^{N/2-1} \left\{ (A_{i,N/2+m} + A_{i,N/2-m})\cos(\frac{2\pi nm}{IN}) - (B_{i,N/2+m} - B_{i,N/2-m})\sin(\frac{2\pi nm}{IN}) \right\},$$

$$n = 0, 1, \ldots, M-1, \tag{3.12}$$

and

$$b_{i,n} = \mathrm{Im}\left\{ \sum_{m=-N/2+1}^{N/2-1} (A_{i,m+N/2} + jB_{i,m+N/2})\cdot e^{\frac{j2\pi nm}{IN}} \right\} \Rightarrow$$

$$b_{i,n} = \sum_{m=-N/2+1}^{N/2-1} \left\{ A_{i,m+N/2}\cdot\sin(\frac{2\pi nm}{IN}) + B_{i,m+N/2}\cdot\cos(\frac{2\pi nm}{IN}) \right\} \Rightarrow$$

$$b_{i,n} = B_{i,N/2} + \sum_{m=1}^{N/2-1} \left\{ (A_{i,N/2+m} - A_{i,N/2-m})\sin(\frac{2\pi nm}{IN}) + (B_{i,N/2+m} + B_{i,N/2-m})\cos(\frac{2\pi nm}{IN}) \right\},$$

$$n = 0, 1, \ldots, M-1. \tag{3.13}$$

After calculating $a_{i,n}$ and $b_{i,n}$ the following time discrete signals with sampling period T are constructed

$$\tilde{a}(t) = \sum_{i=-\infty}^{\infty} \left\{ \sum_{n=-d}^{M-1} a_{i,((n))}\cdot\delta(t - nT - iT_s') \right\}, \tag{3.14}$$

$$\tilde{b}(t) = \sum_{i=-\infty}^{\infty} \left\{ \sum_{n=-d}^{M-1} b_{i,((n))}\cdot\delta(t - nT - iT_s') \right\}, \tag{3.15}$$

in which

$$((n)) = n \text{ modulo } M = n + rM,\ 0 \leqslant n + rM < M,\ r \text{ an integer} \tag{3.16}$$

$$T = \frac{T_s}{IN} = \frac{T_s}{M}, \tag{3.17}$$

$$d = \frac{\Delta}{T} = \frac{\Delta IN}{T_s} = \gamma I. \tag{3.18}$$

Consider the continuous time signals a(t) and b(t)

$$a(t) = \sum_{i=-\infty}^{\infty} \left\{ A_{i,N/2} \cdot \Phi'_{i,0,0}(t) + \sum_{m=1}^{N/2-1} \left\{ (A_{i,N/2+m} + A_{i,N/2-m}) \Phi'_{i,m,0}(t) - (B_{i,N/2+m} - B_{i,N/2-m}) \Theta'_{i,m,0}(t) \right\} \right\}, \tag{3.19}$$

$$b(t) = \sum_{i=-\infty}^{\infty} \left\{ B_{i,N/2} \cdot \Theta'_{i,0,0}(t) + \sum_{m=1}^{N/2-1} \left\{ (A_{i,N/2+m} - A_{i,N/2-m}) \Theta'_{i,m,0}(t) + (B_{i,N/2+m} + B_{i,N/2-m}) \Phi'_{i,m,0}(t) \right\} \right\}, \tag{3.20}$$

and the lowpass filter with transfer function $H_i(f)$

$$H_i(f) = \begin{cases} 1, & \text{for } |f| \leqslant 1/2T \\ 0, & \text{otherwise,} \end{cases} \tag{3.21}$$

and impulse response $h_i(t)$

$$h_i(t) = \frac{\sin(\pi t/T)}{\pi t} = \frac{1}{T} \cdot \mathrm{sinc}(t/T). \tag{3.22}$$

Now the desired signal s(t) according to Equation 3.1 can be rewritten as

$$s(t) = \left\{ (a(t) * h_i(t)) \cdot \cos(2\pi(f_0 + \frac{N}{2T_s})t) - (b(t) * h_i(t)) \cdot \sin(2\pi(f_0 + \frac{N}{2T_s})t) \right\} * h_0(t). \tag{3.23}$$

From Equations 3.12, 3.13, 3.14, 3.15, 3.19 and 3.20 it can be recognized that the constructed signals $\tilde{a}(t)$ and $\tilde{b}(t)$ are equal to the sampled versions of the signals a(t) and b(t) with sampling period T

$$\tilde{a}(t) = \sum_{n=-\infty}^{\infty} a(nT)\cdot\delta(t - nT), \tag{3.24}$$

$$\tilde{b}(t) = \sum_{n=-\infty}^{\infty} b(nT)\cdot\delta(t - nT). \tag{3.25}$$

Therefore, if it is possible to reconstruct a(t) and b(t) from their sampled versions $\tilde{a}(t)$ and $\tilde{b}(t)$, then it is possible to generate s(t). However, the bandwidth of both a(t) and b(t) is unlimited (random data), as shown by the power spectral density G(f) of both a(t) and b(t)

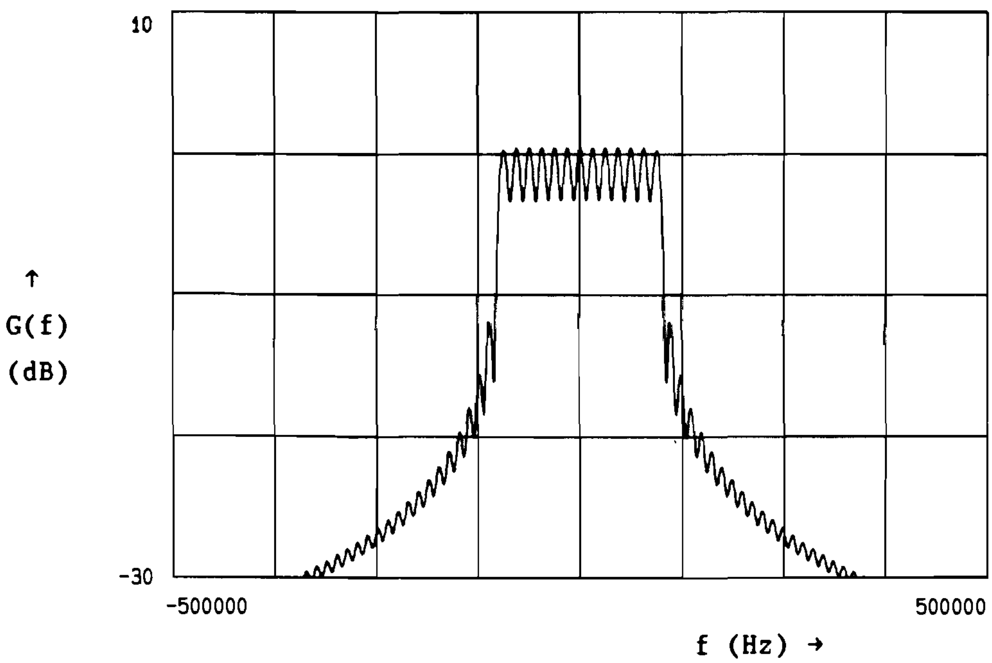
$$\begin{aligned} G(f) = T_s'\cdot\xi_{N/2}\cdot\mathrm{sinc}^2(fT_s') + \frac{1}{2}\cdot T_s'\cdot\sum_{m=1}^{N/2-1}\Big\{ & \xi_{N/2-m}\cdot\mathrm{sinc}^2((f - \frac{m}{T_s})T_s') \\ & + \xi_{N/2+m}\cdot\mathrm{sinc}^2((f - \frac{m}{T_s})T_s') + \xi_{N/2-m}\cdot\mathrm{sinc}^2((f + \frac{m}{T_s})T_s') \\ & + \xi_{N/2+m}\cdot\mathrm{sinc}^2((f + \frac{m}{T_s})T_s') \Big\}, \end{aligned} \tag{3.26}$$

where

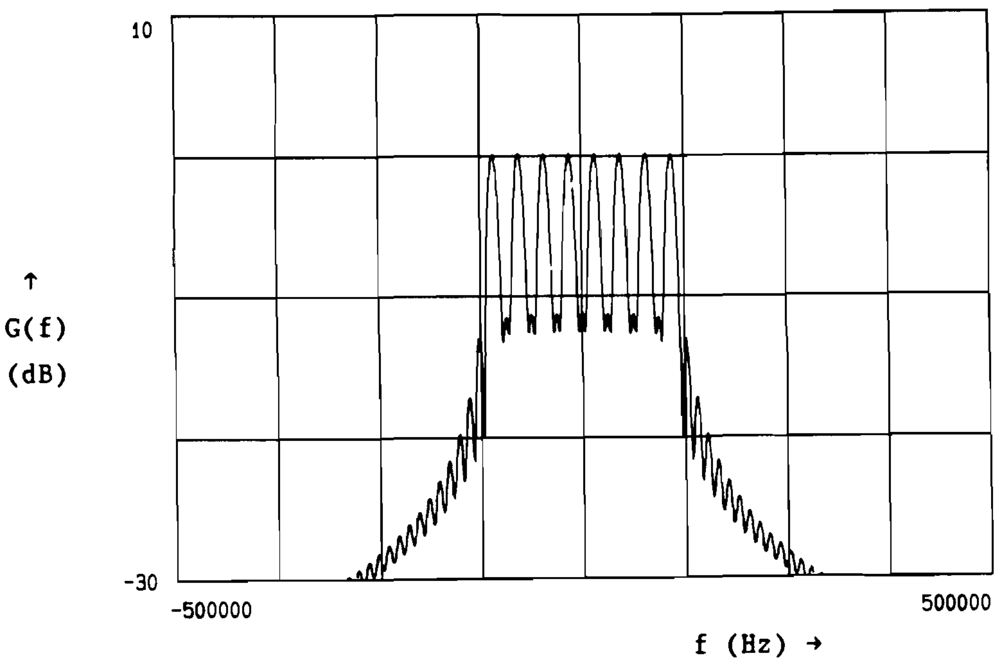
$$\xi_k = \begin{cases} 1, & \text{for the values of k belonging to the emitted carriers,} \\ 0, & \text{for the values of k belonging to the virtual carriers.} \end{cases} \tag{3.27}$$

Figure 3.1 shows two typical examples of the power spectral density G(f). From Equations 3.24, 3.25, 3.26 and 3.27 it can be derived that the power spectral density $\tilde{G}(f)$ of both $\tilde{a}(t)$ and $\tilde{b}(t)$ is given by

$$\tilde{G}(f) = \sum_{p=-\infty}^{\infty} G(f - p\cdot\frac{IN}{T_s}). \tag{3.28}$$

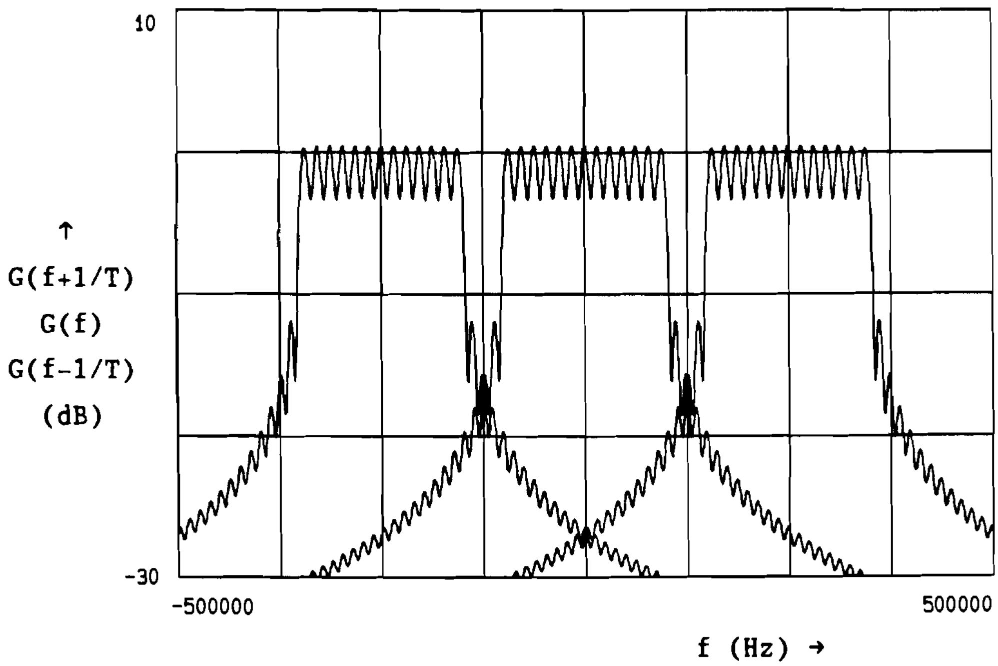


(a)

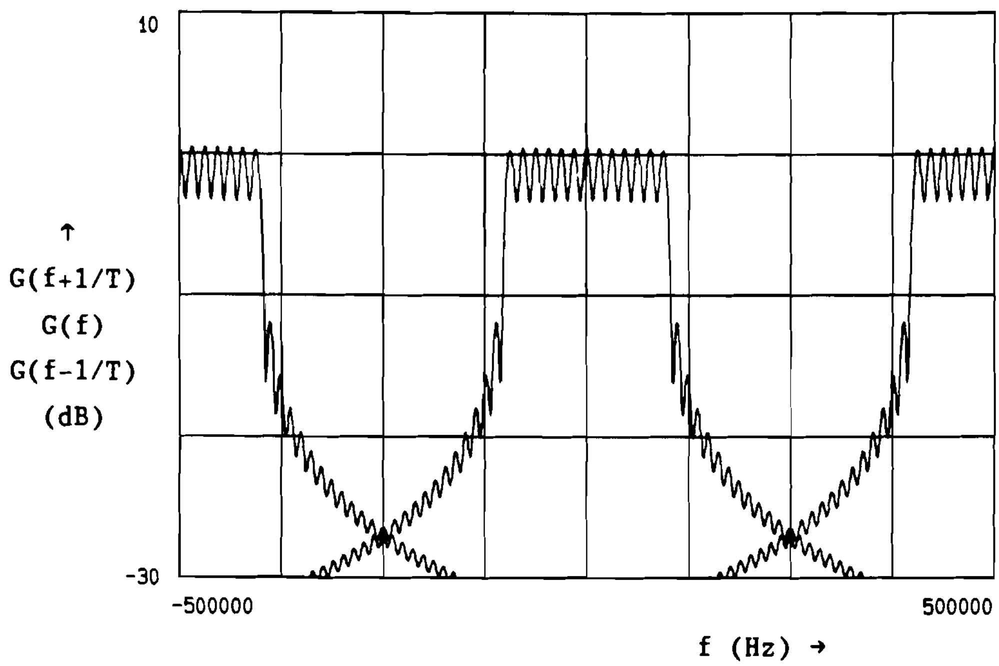


(b)

Figure 3.1. The power spectral density G(f) of both a(t) and b(t) for T_s = 64 μs, Δ = 16 μs, N = 16,
(a) ξ_k = 1 for k = 2, 3, ..., N-2, ξ_k = 0 for k = 0, 1, N-1,
(b) ξ_k = 1 for k = 1, 3, 5, 7, 9, 11, 13, 15,
ξ_k = 0 for k = 0, 2, 4, 6, 8, 10, 12, 14.



(a)



(b)

Figure 3.2. The power spectra G(f + 1/T), G(f), G(f - 1/T) for T_s = 64 μs, Δ = 16 μs, N = 16, ξ_k = 1 for k = 2, 3, ..., N-2, ξ_k = 0 for k = 0, 1, N-1,

(a) I = 1

(b) I = 2

Because the bandwidth of a(t) and b(t) is unlimited, it is impossible to reconstruct a(t) and b(t) from $\tilde{a}(t)$ and $\tilde{b}(t)$ without aliasing distortion. This aliasing distortion only tends to zero when the oversampling ratio I goes to infinity which is illustrated in Figure 3.2. It can also be reduced by introducing virtual carriers for the values of k close to zero and close to N. Consequently it is clear that using the Inverse Discrete Fourier Transform in generating the OFDM/(D)QPSK signal gives the signal $\tilde{s}(t)$ according to

$$\begin{aligned}\tilde{s}(t) &= \Big\{ (\tilde{a}(t)*h_i(t))\cdot\cos(2\pi(f_0+\frac{N}{2T_s})t) \\ &\qquad - (\tilde{b}(t)*h_i(t))\cdot\sin(2\pi(f_0+\frac{N}{2T_s})t) \Big\} * h_0(t) \\ &= s(t) + \Big\{ \varepsilon_a(t)\cdot\cos(2\pi(f_0+\frac{N}{2T_s})t) \\ &\qquad - \varepsilon_b(t)\cdot\sin(2\pi(f_0+\frac{N}{2T_s})t) \Big\} * h_0(t),\end{aligned} \tag{3.29}$$

where

$$\varepsilon_a(t) = \tilde{a}(t)*h_i(t) - a(t)*h_i(t), \tag{3.30}$$

$$\varepsilon_b(t) = \tilde{b}(t)*h_i(t) - b(t)*h_i(t). \tag{3.31}$$

This signal $\tilde{s}(t)$ is equal to the sum of the desired signal s(t) and a interfering signal caused by aliasing. Only when the oversampling ratio goes to infinity, this interfering signal will fade

$$\lim_{I\to\infty} \tilde{s}(t) = s(t). \tag{3.32}$$

3.3. Practical considerations

In the previous section it has been theoretically treated how to generate the OFDM/(D)QPSK signal with useful signal duration T_s and guardband interval Δ. In this section attention is paid to the practical realization of such a modulator and the problems associated with it. Figure 3.3 shows the block diagram of the OFDM/(D)QPSK modulator.

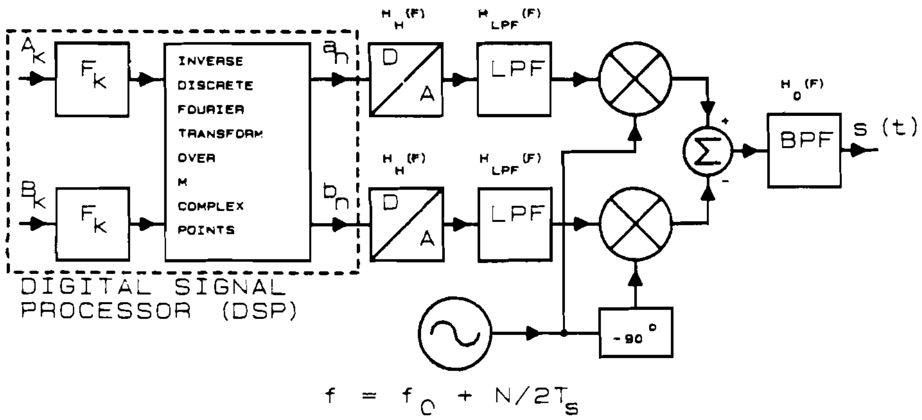


Figure 3.3. The block diagram of the OFDM/(D)QPSK modulator.

The first problem is that the analog interpolating filter, with transfer function $H_i(t)$ according to Equation 3.21, can not be realized. In practice this interpolating filter is formed by the cascade interconnection of the zero-order hold in the Digital to Analog Converter (DAC), with transfer function $H_h(f)$, and the lowpass filter following the DAC, with transfer function $H_{lpf}(f)$. The transfer function $H_h(f)$ is given by

$$H_h(f) = \mathrm{sinc}(fT). \tag{3.33}$$

The frequency components of both $\tilde{a}(t) * h_i(t)$ and $\tilde{b}(t) * h_i(t)$ between $N/2T_s$ and $IN/2T_s$ do not contribute to the signal $\tilde{s}(t)$ due to the bandpass filter after the quadrature modulator. Therefore, it is sufficient to make a lowpass filter with transfer function $H_{lpf}(f)$ which approximately satisfies the following conditions

$$|H_{lpf}(f)| = \frac{1}{\mathrm{sinc}(fT)} \quad , \text{ for } |f| \leq N/2T_s, \tag{3.34}$$

$$|H_{lpf}(f)| = 0 \quad , \text{ for } |f| > IN/2T_s, \tag{3.35}$$

$$\arg(H_{lpf}(f)) = \text{constant} \cdot f \quad , \text{ for } |f| \leq N/2T_s. \tag{3.36}$$

These conditions make clear that increasing the oversampling ratio facilitates the design of the lowpass filter. The best choice to satisfy Equation 3.36 is a Bessel lowpass filter with 3 dB cut-off frequency somewhere between $N/2T_s$ and $IN/2T_s$. The order of this filter is mainly determined by the chosen cut-off frequency and the condition as stated in Equation 3.35. In general this filter will not satisfy Equation 3.34, therefore the magnitude of the transfer function of the analog interpolating filter will not be constant for $|f| \leq N/2T_s$. However, this non flat frequency response can be compensated in the digital part of the modulator. To do this, the values of $A_{i,k}$ and $B_{i,k}$ are multiplied by F_k

$$F_k = \frac{1}{\left|H_h\left(\frac{|k - N/2|}{T_s}\right)\right|} \cdot \frac{1}{\left|H_{lpf}\left(\frac{|k - N/2|}{T_s}\right)\right|}. \tag{3.37}$$

These multiplications are done in the first stage of the modulator as shown in Figure 3.3. This first stage together with the second stage containing the IDFT processing are realized using a digital signal processor.

The second practical problem is the quantisation noise. There are five parameters who influence the quantisation noise power in the signal $\tilde{s}(t)$. The first parameter is the real number representation in the digital signal processor. In general more bits give less quantisation noise power and for the same number of bits a floating point representation will generally give less quantisation noise power than a fixed point representation. The second parameter is the algorithm used to calculate $a_{i,n}$ and $b_{i,n}$. The quantisation noise power is minimized by minimizing the number of round off operations. The third parameter is the number of (linear) quantisation levels used by the Digital to Analog Converters. More levels (bits) will give less quantisation noise power. The fourth parameter is the ratio between the clipping level of the Digital to Analog Converter and the standard deviation of both $a_{i,n}$ and $b_{i,n}$. This standard deviation is given by

$$\sigma_a = \sqrt{E[(a_{i,n})^2] - (E[a_{i,n}])^2} = \sqrt{E[(a_{i,n})^2]} = \sqrt{\sum_{k=0}^{N-1} F_k \xi_k}, \tag{3.38}$$

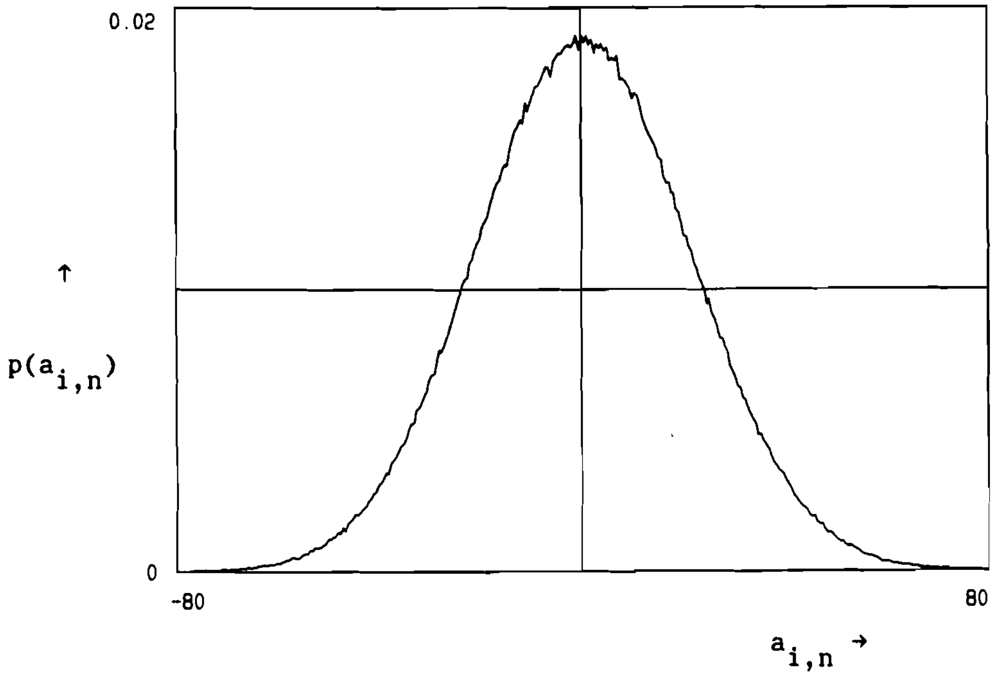
$$\sigma_b = \sigma_a. \tag{3.39}$$

Decreasing the clipping level proportionally to this standard deviation will decrease the quantisation noise power. However, this will increase the clipping noise power as soon as the clipping level becomes smaller than the maximum value of both $a_{i,n}$ and $b_{i,n}$. An upper bound for this maximum value is given by

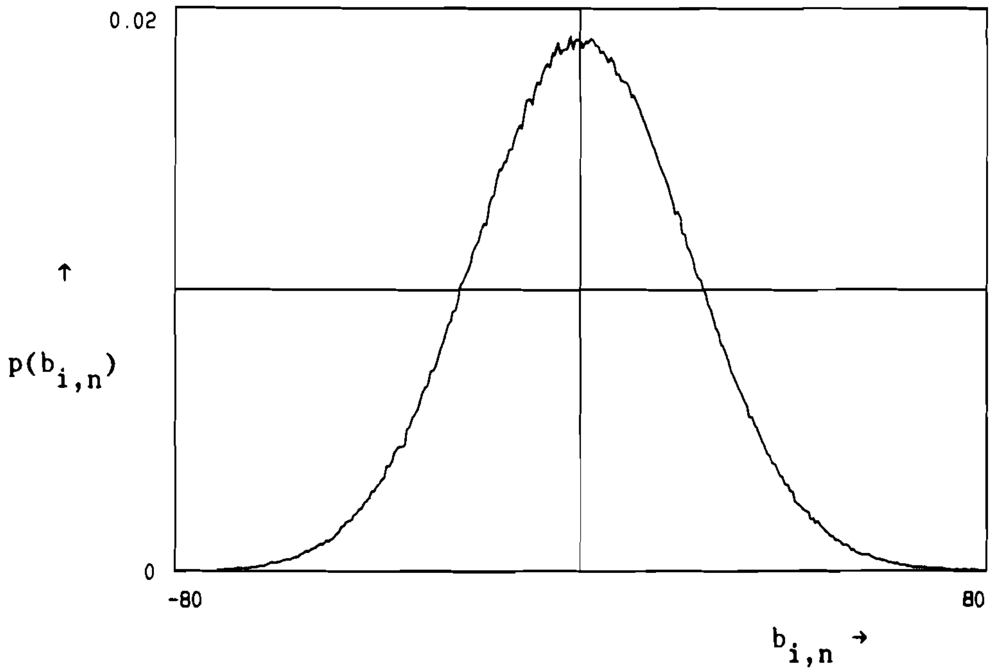
$$a_{i,n} \leqslant \sum_{k=0}^{N-1} F_k \xi_k, \tag{3.40}$$

$$b_{i,n} \leqslant \sum_{k=0}^{N-1} F_k \xi_k. \tag{3.41}$$

If the distribution of both $a_{i,n}$ and $b_{i,n}$ is known, then it is possible to maximize the signal to clipping and quantisation noise ratio. However the question is if this will minimize the bit error probability. In practice, if the number of carriers is large, setting the clipping level to four times the standard deviation will give a good performance. When the number of emitted carriers is large, the distribution of both $a_{i,n}$ and $b_{i,n}$ can be approximated by a normal distribution with zero mean and standard deviation as given by Equation 3.38. In Figure 3.4 and Table 3.1 the results are shown of a Monte Carlo simulation used to calculate an estimation of the distribution of both $a_{i,n}$ and $b_{i,n}$.



(a)



(b)

Figure 3.4. Probability density function of both $a_{i,n}$ and $b_{i,n}$ calculated over 3000 random data symbols for $N = 512$, $I = 1$, $\phi_k = \pi \cdot k^2/N$, $F_k = 1$, emitted carriers for $32 \leqslant k \leqslant 255$ and $257 \leqslant k \leqslant 480$ and virtual carriers for $0 \leqslant k \leqslant 31$, $k = 256$ and $481 \leqslant k \leqslant 511$.

Table 3.1. Statistical data belonging to Figure 3.4.

minimum value $a_{i,n}$	=	-98,243
maximum value $a_{i,n}$	=	111,651
mean value $a_{i,n}$	=	0,000
standard deviation $a_{i,n}$	=	21,165
probability that $a_{i,n}$ > 80	=	$1,439 \cdot 10^{-4}$
minimum value $b_{i,n}$	=	-101,787
maximum value $b_{i,n}$	=	107,559
mean value $b_{i,n}$	=	0,000
standard deviation $b_{i,n}$	=	21,160
probability that $b_{i,n}$ > 80	=	$1.465 \cdot 10^{-4}$

During this simulation 3000 symbols containing random data were calculated given the parameters below

$N = 512$,
$I = 1$,
$\phi_k = \frac{\pi \cdot k^2}{N}$,
$F_k = 1$,
emitted carriers for $32 \leqslant k \leqslant 255$ and $257 \leqslant k \leqslant 480$ and
virtual carriers for $0 \leqslant k \leqslant 31$, $k = 256$, $481 \leqslant k \leqslant 511$.

The fifth and last parameter, which has influence on the quantisation noise power, is the oversampling ratio. Increasing the oversampling ratio will decrease the quantisation noise power in $\tilde{s}(t)$.

The previous section made clear that increasing the oversampling ratio decreases the aliasing distortion in the generated OFDM/(D)QPSK signal. Besides this advantage, increasing the oversampling ratio has two more practical advantages as described in this section. It is obvious that a price has to be paid for all these advantages. This price is expressed in more computing power and faster Digital to Analog Converters. If no advantage is taken of the large number of zero's in $\bar{A}_{i,m}$ and $\bar{B}_{i,m}$ when $I > 1$, then the computation of $a_{i,n}$ and $b_{i,n}$ is done using a M point Inverse Fast Fourier Transform requiring $\frac{M}{2}\cdot\log_2(M)$ complex multiplications and $M\cdot\log_2(M)$ complex additions. However, if the large number of zero's is taken into account, some computing power can be saved. This can be shown by rewriting Equation 3.11

$$a_{i,Ig+h} + jb_{i,Ig+h} = \sum_{k=0}^{N-1} (A_{i,k} + jB_{i,k})\cdot e^{\frac{j2\pi(Ig + h)(k - N/2)}{IN}}$$

$$= \sum_{k=0}^{N-1} (A_{i,k} + jB_{i,k})\cdot e^{\frac{j2\pi gk}{N}}\cdot e^{-j\pi g}\cdot e^{\frac{j2\pi hk}{IN}}\cdot e^{\frac{-j\pi h}{I}}$$

$$= (-1)^g\cdot \sum_{k=0}^{N-1} \left\{(A_{i,k} + jB_{i,k})\cdot e^{\frac{j2\pi hk}{IN}}\cdot e^{\frac{-j\pi h}{I}}\right\}\cdot e^{\frac{j2\pi gk}{N}},$$

$$g = 0, 1, .., N-1;\ h = 0, 1, .., I-1;\ k = 0, 1, .., N-1. \qquad (3.42)$$

If parameter h is fixed, then N values of both $a_{i,n}$ and $b_{i,n}$ can be computed using a N point IFFT

$$a_{i,Ig+h} + jb_{i,Ig+h}\Big|_{h\ \text{fixed}} = (-1)^g\cdot N\cdot \mathrm{IFFT}_N\left\{(A_{i,k} + jB_{i,k})\cdot e^{\frac{j\pi(2hk - hN)}{IN}}\right\}.$$

$$(3.43)$$

This takes $N\cdot\log_2(N)$ complex additions and $\frac{N}{2}\cdot\log_2(N)$ complex multiplications if h is equal to zero or $\frac{N}{2}\cdot\log_2(N) + N$ complex multiplications if h is not equal to zero. This computation must be repeated for I times to obtain all values of $a_{i,n}$ and $b_{i,n}$. Therefore totally $\frac{M}{2}\cdot\log_2(N) + M - N$ complex multiplications and $M\cdot\log_2(N)$ complex additions are needed.

Another practical problem is the choice of the signal of the reference symbol. For the reference signal it is easy to set all modulation angles θ_k to zero. However, this will generally lead to impulse shaped baseband signals a(t) and b(t). This is illustrated in Figure 3.5 for System-2 (see appendix A), $\theta_k = 0$ and $\phi_k = 0$. Such a signal is undesirable because it will be highly distorted if the clipping level is set to four times the standard deviation σ. The solution to this problem is to set ϕ_k equal to $\pi \cdot k^2/N$ for the emitted carriers. For this choice of ϕ_k the reference symbol will approximately be a sine-sweep signal. The amplitude of both a(t) and b(t) is smaller than four times the standard deviation σ for this choice of ϕ_k. Therefore, this signal choice makes the generation of an undistorted reference symbol possible. Figure 3.6 shows this by means of the baseband reference signals a(t) and b(t) for System-2, $\theta_k = 0$ and $\phi_k = \pi \cdot k^2/N$.

Figure 3.7 shows a typical example of the baseband signals belonging to a data symbol for System-2, θ_k randomly chosen from $\{ 0, \frac{\pi}{2}, \pi, \frac{3\cdot\pi}{2} \}$, $\phi_k = \pi \cdot k^2/N$. This figure is added in order to compare it with the other baseband signal figures. In Figure 3.5, 3.6 and 3.7, the standard deviation σ is equal to the square root of 448 ($\sigma = 21.17$).

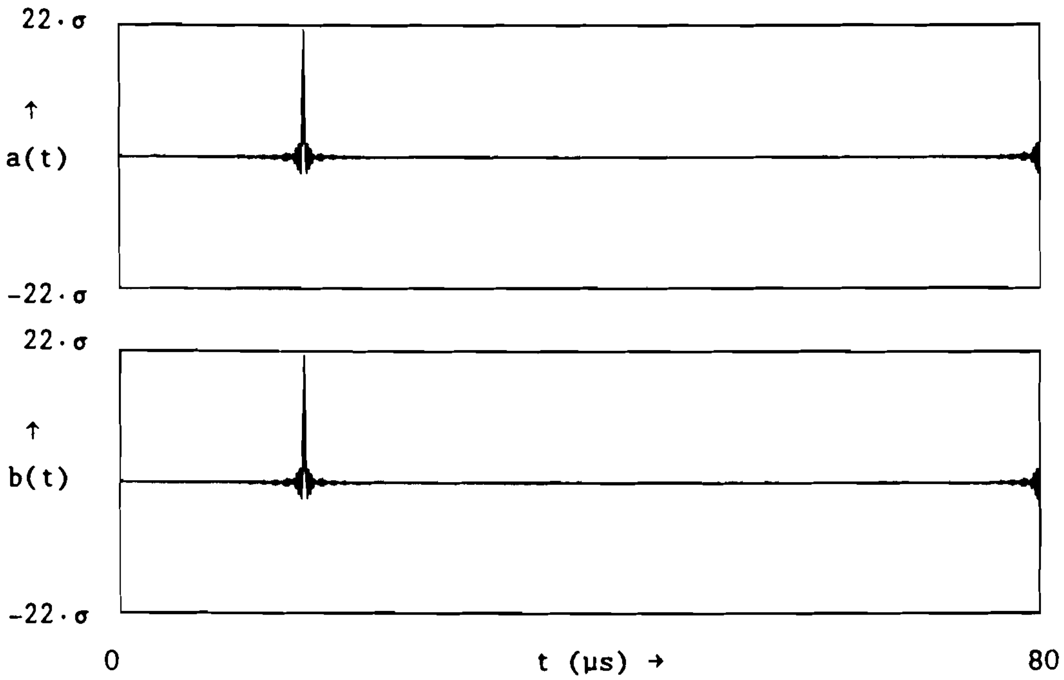


Figure 3.5. Calculated baseband OFDM/DQPSK signals a(t) and b(t) with guardband interval of System-2 for $\theta_k = 0$ and $\phi_k = 0$.

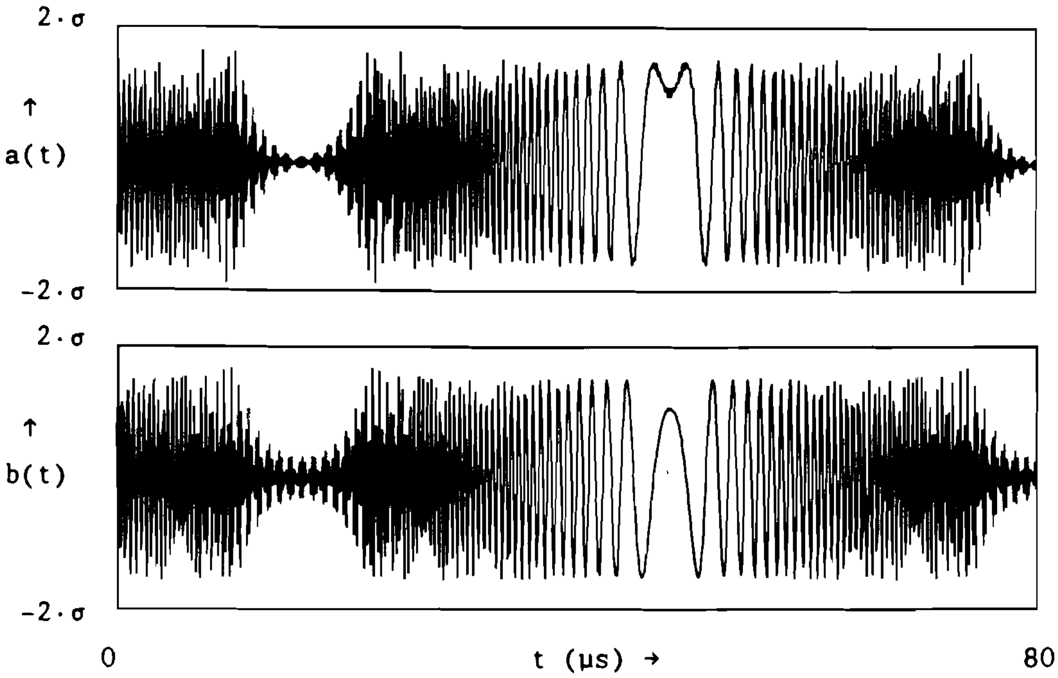


Figure 3.6. Calculated baseband OFDM/DQPSK signals a(t) and b(t) with guardband interval of System-2 for $\theta_k = 0$ and $\phi_k = \pi \cdot k^2/N$.

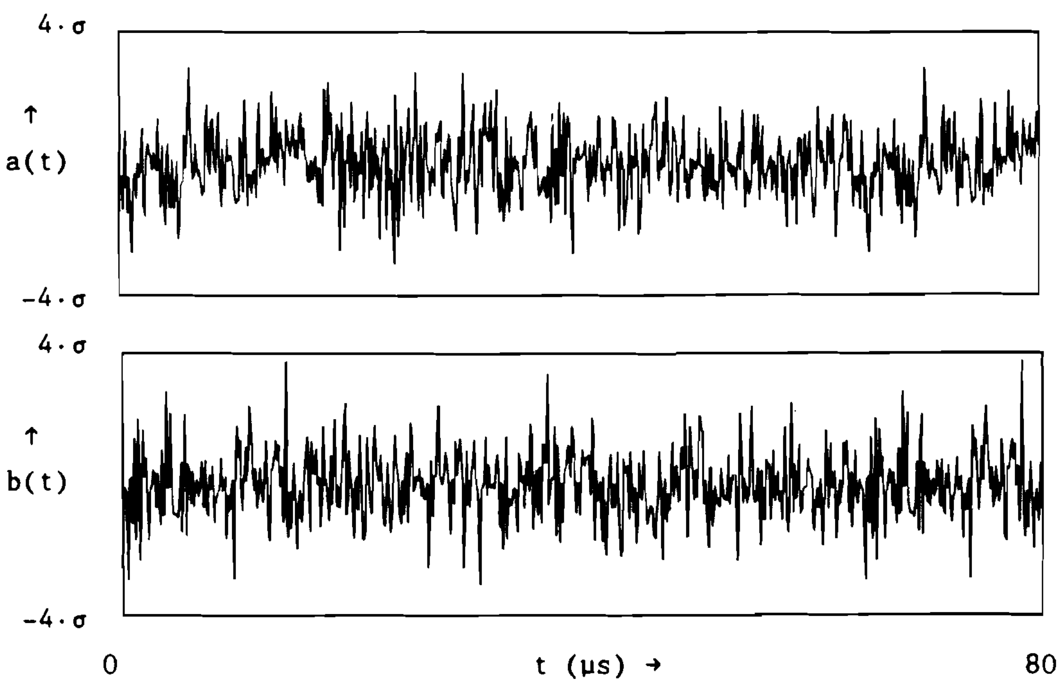


Figure 3.7. Calculated baseband OFDM/DQPSK signals a(t) and b(t) with guardband interval of System-2 for $\phi_k = \pi \cdot k\ /N$ and θ_k randomly chosen from $\{0, \frac{\pi}{2}, \pi, \frac{3 \cdot \pi}{2}\}$.

4. DAB TRANSMITTER SIMULATOR

4.1. Introduction

This chapter describes the DAB transmitter simulator that was built during this project. This DAB transmitter simulator is capable of generating a real-time OFDM/(D)QPSK signal according to the specifications of System-1 up to System-4. The signal consists of a periodically repeated frame. This frame contains a reference symbol, one or more (depending on the used system) pseudo random data symbols and a null symbol. For test purposes it is also possible to suppress all phase and amplitude modulations. The DAB transmitter simulator can be used to determine the OFDM/(D)QPSK signal interference on co- and neighbour-channel TV or other signals. The DAB transmitter simulator is built in such a way that it can be extended with a receiver part, giving a DAB transceiver simulator. With such a DAB transceiver simulator it will be possible to measure bit error rates in various channels.

4.2. Simulator structure

The DAB transmitter simulator structure is illustrated in figure 4.1. It is built-up around a Personal Computer (PHILIPS P3200). The Personal Computer is used for the digital OFDM/(D)QPSK (with guardband interval) signal generation as described in chapter 3.

Since the processing power of the Personal Computer is too low to generate a real-time signal, it is only used to calculate the two discrete baseband signals $a_{i,n}$ and $b_{i,n}$ (real and imaginary part) of one frame with a duration of 3.84 ms. The processing time needed for this calculation is much greater than those 3.84 ms. In order to get a real-time signal, an external sampler is used.

When the discrete signals are calculated, the 8 bit quantized samples are stored in the 30720 bytes random access memory (RAM) of the external sampler. As soon as the samples are stored, the external sampler starts reading the samples of both signals with a sample frequency of 8 MHz. These samples of both discrete signals are sent to two Digital to Analog Converters both followed by an interpolating filter. In this way the discrete-time baseband signals are transformed to the continuous-time signals a(t) and b(t). The sampler reads the memory cyclically. This gives the periodically repeated frame.

In the next part of the simulator, the two signals a(t) and b(t) are mixed in quadrature on a Intermediate Frequency (IF) of 36.2 MHz. Finally this IF OFDM/(D)QPSK signal is filtered by a bandpass filter with a bandwidth of 7 MHz. This gives the wanted OFDM/(D)QPSK signal s(t).

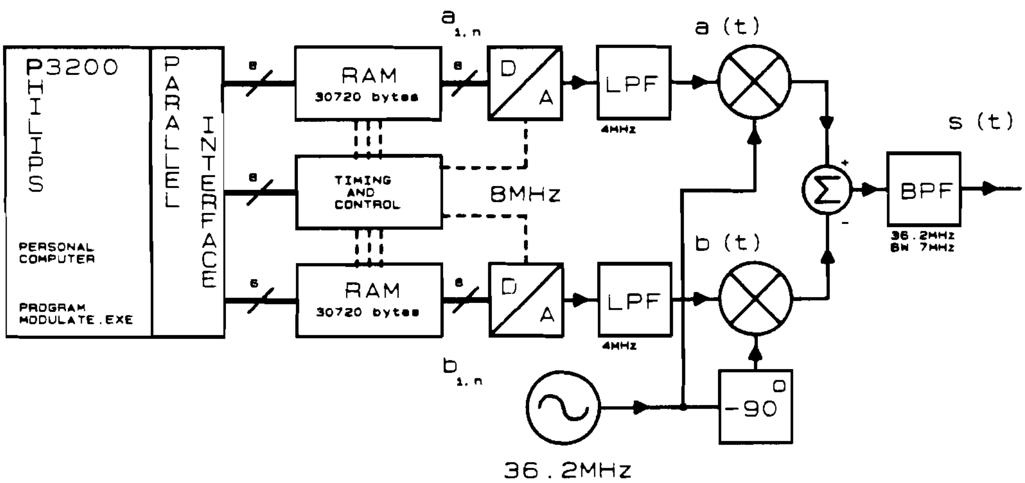


Figure 4.1. The DAB transmitter simulator structure.

4.3. Simulator hardware

The main part of the simulator hardware is the PHILIPS P3200 Personal Computer. This computer is IBM PC/AT compatible and is built-up around a 80286 microprocessor from Intel. In order to enhance the floating point processing power a 80287 math coprocessor (Intel) was added to the system. The rest of this paragraph only describes the hardware added to the Personal Computer.

During this project, five Printed Circuit Boards (PCB) with hardware were added to the Personal Computer:

- the Personal Computer/DAB transceiver interface,
- the data input/output board of the transceiver,
- the transceiver output board for synchronization signals,
- the transmitter baseband board,
- the transmitter intermediate frequency board.

In order to make a complete DAB transceiver, a receiver baseband and a receiver intermediate frequency board should be added to the system.

The Personal Computer/DAB transceiver interface is an expansion card for the PC which gives 24 bit parallel I/O capabilities to the PC. The schematic of the circuit on the card is given in appendix B under the name "PC/TRANSCEIVER INTERFACE". This parallel I/O circuit is built around a 8255 Programmable Parallel Interface (PPI) Integrated Circuit (IC) from Intel. This IC belongs to the same family as the 80286 and the 80287. The switches on this board can be used to set the base address, of the four 8 bit registers of the 8255 PPI, in the I/O address space of the 80286. The DAB transmitter simulator software assumes that this base address is set to 768 (= 300 hexadecimal).

Table 4.1. PPI 8255 port C signal definition used for Personal Computer/DAB transceiver interface.

PC0 - check transceiver +5V power supply input
PC1 - check connection between PC and transceiver input
PC2 - read receiver RAM acknowledge input*
PC3 - write transmitter RAM acknowledge input
PC4 - read receiver RAM strobe output*
PC5 - write transmitter RAM strobe output
PC6 - read receiver RAM request output*
PC7 - write transmitter RAM request output

* not used during this project

Port A and port B of the 8255 PPI are used for input or output of the 8 bit quantized samples of respectively the real and imaginary part of the time-discrete signal. Since no receiving part was built, only the output mode of port A and B is used during this project. Port C is used for handshake and control signals. Table 4.1 gives the used PPI 8255 port C signal definition.

The four other boards, which will be described hereafter, are built on a single Euroboard. The four boards are assembled in a 19" Eurorack system. The +5V and -5V supplies are delivered by two switched mode power supplies which are also placed in the 19" Eurorack system. The 96 pin Euroconnectors are used for the transceiver bus. The description of this transceiver bus is given in appendix B.

The input/output board of the DAB transmitter simulator hardware is used to get a proper connection between the 24 bit parallel interface in the PC and the DAB transceiver bus. The schematic of this board is given in appendix B with the name "TRANSCEIVER I/O BOARD". The main function of this board is to disable the "write transmitter RAM request" signal when the "read receiver RAM acknowledge" signal is active and to disable the "read receiver RAM request" when the "write transmitter RAM acknowledge" is active. This gives a hardware protection against a possible transceiver data bus collision.

The transceiver output board for synchronization signals is used to get buffered synchronization signals. The schematic of this board is given in appendix B under the name "SYNC OUTPUT BOARD". This board gives the 8 MHz sampling clock signal and the frame start pulse signal. The used buffers are specially suited to drive a 50Ω load.

The transmitter baseband board takes care of generating the baseband OFDM/(D)QPSK signals by cyclically reading the signals which are written into its RAM by the PC. The output signals of this board are sample-and-hold signals produced by the Digital to Analog Converters (DAC) on this board (the DC of this signal is blocked). The schematic of this board is given in Appendix B under the name "TRANSMITTER BASEBAND BOARD". This schematic consists of one root-sheet and four sub-sheets. This board contains the function blocks "timing and control", "RAM" and "D/A" as given in Figure 4.1. The outputs of the Digital to Analog Converters are capable of driving a 75Ω load.

The transmitter baseband circuit starts transmitting as soon as the last samples are written into its memory. If the PC wants to write new samples into the RAM, it has to set the WTRREQ signal on the transceiver bus. When this signal is high, the transmitter stops transmitting as soon as the last samples of the frame have been read and send to the DAC's. At this moment, the WTRACK signal goes high in order to signal to the PC, that it can start to write the samples of a new frame into the transmitter RAM. When WTRACK is active, the PC has to reset the WTRREQ signal and place the first real and the first imaginary sample on the databus. After placing the first samples on the databus, the PC has to generate a strobe signal. This strobe signal consists of a positive going edge on the WTRSTR signal line. The effect of this strobe signal is that the samples are written into the first RAM location and that the RAM address counter is advanced to the next RAM location. By repeatedly placing samples on the databus and generating a strobe signal, the total frame can be written into the transmitter RAM. For all this time the WTRACK signal stays active high. As soon as the last sample is written into the RAM, the WTRACK signal goes low and the transmitter starts reading the samples of the new frame.

The switches on the transmitter baseband board are used to set the frame length in number of samples. The maximum length is 32768 samples. The default setting for the written software (program MODULATE) is 30720 samples. For the OFDM/(D)QPSK with guardband interval modulation systems 1 to 4 (appendix A) this length is respectively equal to 3, 48, 24 and 12 signalling intervals.

The last board of the DAB transmitter simulator is the transmitter intermediate frequency board. The schematic of this board is given in appendix B in the root-sheet with the name "TRANSMITTER IF BOARD". This root-sheet has three sub-sheets. On this board there is a 36.2 MHz crystal oscillator, which is used local oscillator for the quadrature modulation of the real part and imaginary part OFDM/(D)QPSK baseband signals. Two balanced diode mixers , one two-way 90^{o} power splitter/combiner and one two-way 0^{o} power splitter/combiner are used to build the quadrature mixer. Furthermore there are two lowpass filters and one bandpass filter on this board. The lowpass filters, together with the sample-and-hold circuits of the DAC's on the baseband board, are used as interpolating filters. The bandpass filter is used to filter out the unwanted harmonic signals introduced by the quadrature mixer. The inputs for the real part and imaginary part baseband signals have a 75Ω input impedance. The IF output is capable of driving a 50Ω load.

4.4. Simulator software

As mentioned before, the Personal Computer is used for the calculation of the discrete OFDM/(D)QPSK signals (with guardband interval). During this project a program with the name "MODULATE" has been written for this purpose. This program has been written in the C programming language. The used C compiler was "TURBO C 1.5" from Borland. The source code of the written program is given in appendix C. The many comments in this source code listing, will make it easy to understand the operation of the program.

After starting the program "MODULATE", the user has to choose for one of the four OFDM/(D)QPSK with guardband interval modulation systems as given in appendix A. This is shown in Figure 4.2 which gives the opening menu of the program.

OFDM/QPSK MODULATOR (with guardband interval)

Which system do you want?

system selection key	1	2	3	4
system name	system 1	system 2	system 3	system 4
#carriers (emitted + virtual).........	8192	512	512	512
#emitted carriers	7168	448	224	112
carrier spacing (Hz)	976.5625	15625	7812.5	3906.25
minimal modulated carrier spacing (Hz)	976.5625	15625	15625	15625
sampling frequency (MHz)	8	8	8	8
oversampling ratio	1	1	2	4
nominal bandwidth (MHz)	7	7	3.5	1.75
bitrate for used frame (Mbit/s)	3.733334	10.73334	2.566667	0.583334
bitrate for infinite frame (Mbit/s) ..	11.2	11.2	2.8	0.7
useful signal duration (microsec)	1024	64	128	256
guardband interval (microsec)	256	16	32	64
total signal duration (microsec)	1280	80	160	320
#modulated symbols per frame	3	48	24	12

Type system selection key to enter your choice: -

Figure 4.2. The opening menu of the program "MODULATE".

When the user has made a choice for System-2, System-3 or System-4, the program asks if all phase and amplitude modulations must be suppressed. A positive answer to this question, results in filling the transmitter RAM with reference signals without guardband interval. This signal is equal to the sum of all sine waves of the emitted carriers for the chosen system. For System-1 it is not possible to suppress all modulations because it is not possible to fill the transmitter RAM with a multiple integer of reference symbols without guardband interval. The generation of unmodulated carriers is useful in testing the interpolating lowpass filters and the IF bandpass filter. It can also be used to determine the amplitude compensation factors F_k. In the program "MODULATE", these factors F_k are only used to compensate for the sin(x)/x filtering. This filtering is introduced by the sample-and-hold circuits in the DAC's.

If the user has chosen not to suppress the modulations, the program calculates a frame that contains one reference symbol followed by one or more random data symbols which are again followed by one null symbol. The random data, used to generate the random data symbols, are also generated by the program. These generated random data are stored in a file called "ANGLES.DAT" (for detailed description see appendix C). For the systems 1 up to 4 respectively 1, 46, 22 and 10 random data symbols are calculated.

Figure 4.2 gives some interesting information on the DAB OFDM/(D)QPSK modulator simulator. From this figure can be seen that the oversampling ratio for System-3 is equal to 2 and that it is equal to 4 for System-4. Due to this oversampling, it is possible to work with a constant sample frequency in the hardware of the simulator.

4.5. Generated signal measurement

In this paragraph the results of some measurements on the OFDM/(D)QPSK signals generated by the simulator are presented. These measurements are done in the time and frequency domain. The Figures 4.3 and 4.4 show the measured reference symbol baseband signals of respectively System-2 and System-3. These signals are measured with a digitizing oscilloscope at the outputs of the Digital to Analog Converters.

The frequency domain measurements are presented in the Figures 4.5, 4.6 and 4.7. These figures respectively show the measured power spectra of System-2, System-3 and System-4. Each figure shows the overall generated spectrum and a detailed measurement within that overall spectrum.

All spectral measurements, especially the detailed ones, clearly show the virtual center carrier (k = N/2). The detailed spectral measurements show the comb shaped frequency spectrum of System-3 and System-4. From the spurious spectral components below 32 MHz and above 40 MHz in the overall spectrum of System-2 (Figure 4.5), it can be seen that the signal was generated digitally with a 8 MHz sample frequency. In the same spectrum it is possible to see the fourth and fifth harmonic of the sample clock oscillator.

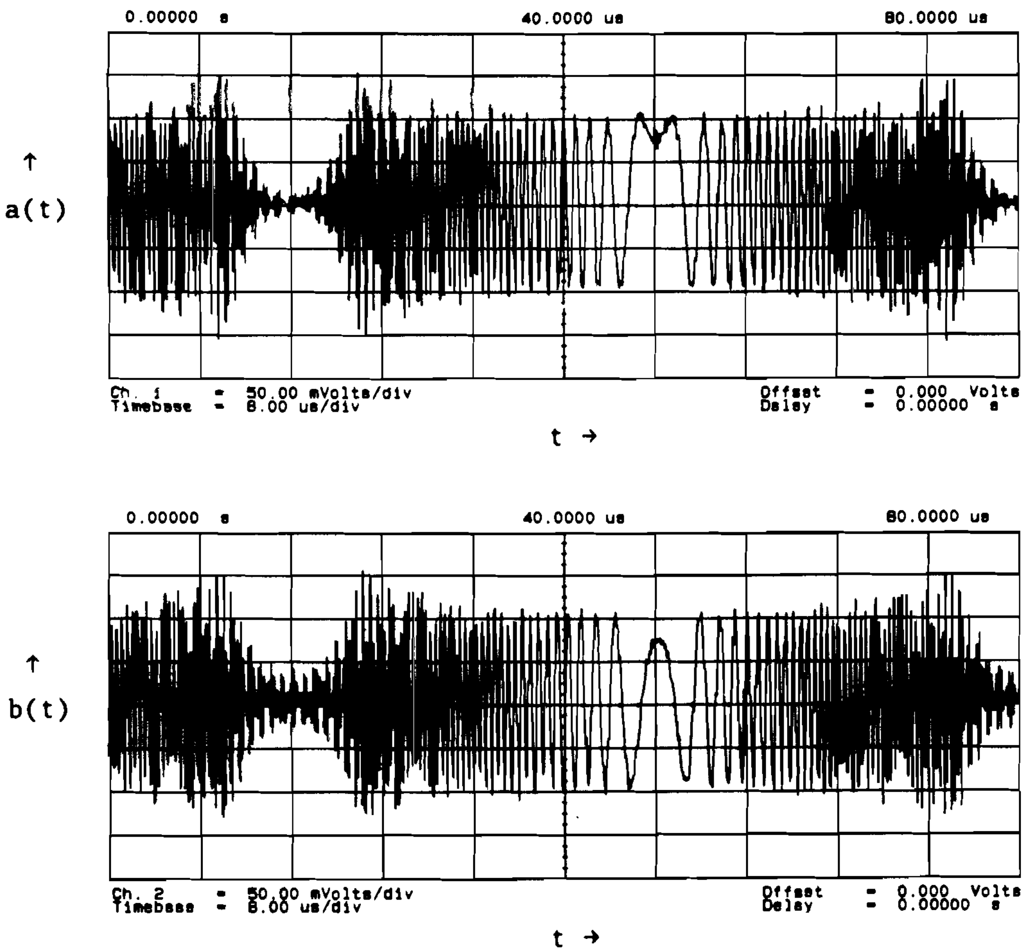


Figure 4.3. Measured reference symbol baseband signals a(t) and b(t) of System-2.

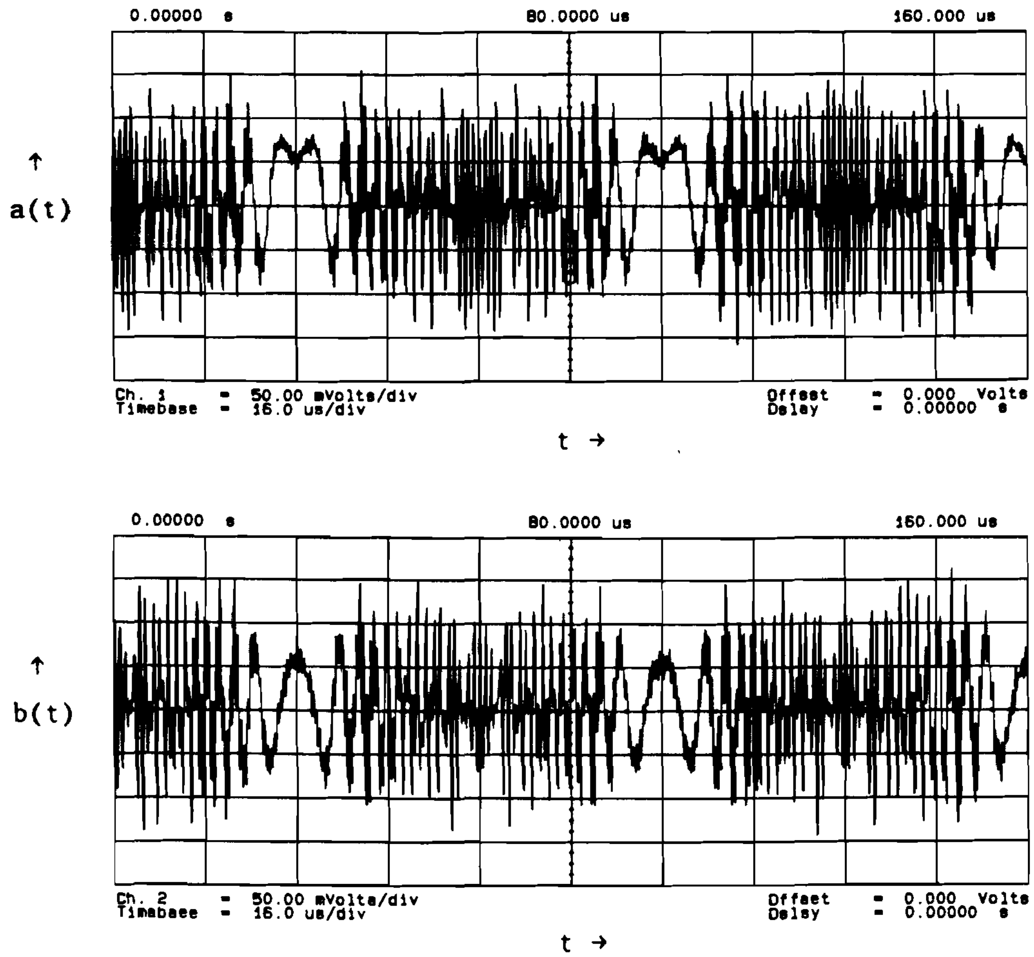


Figure 4.4. Measured reference symbol baseband signals a(t) and b(t) of System-3.

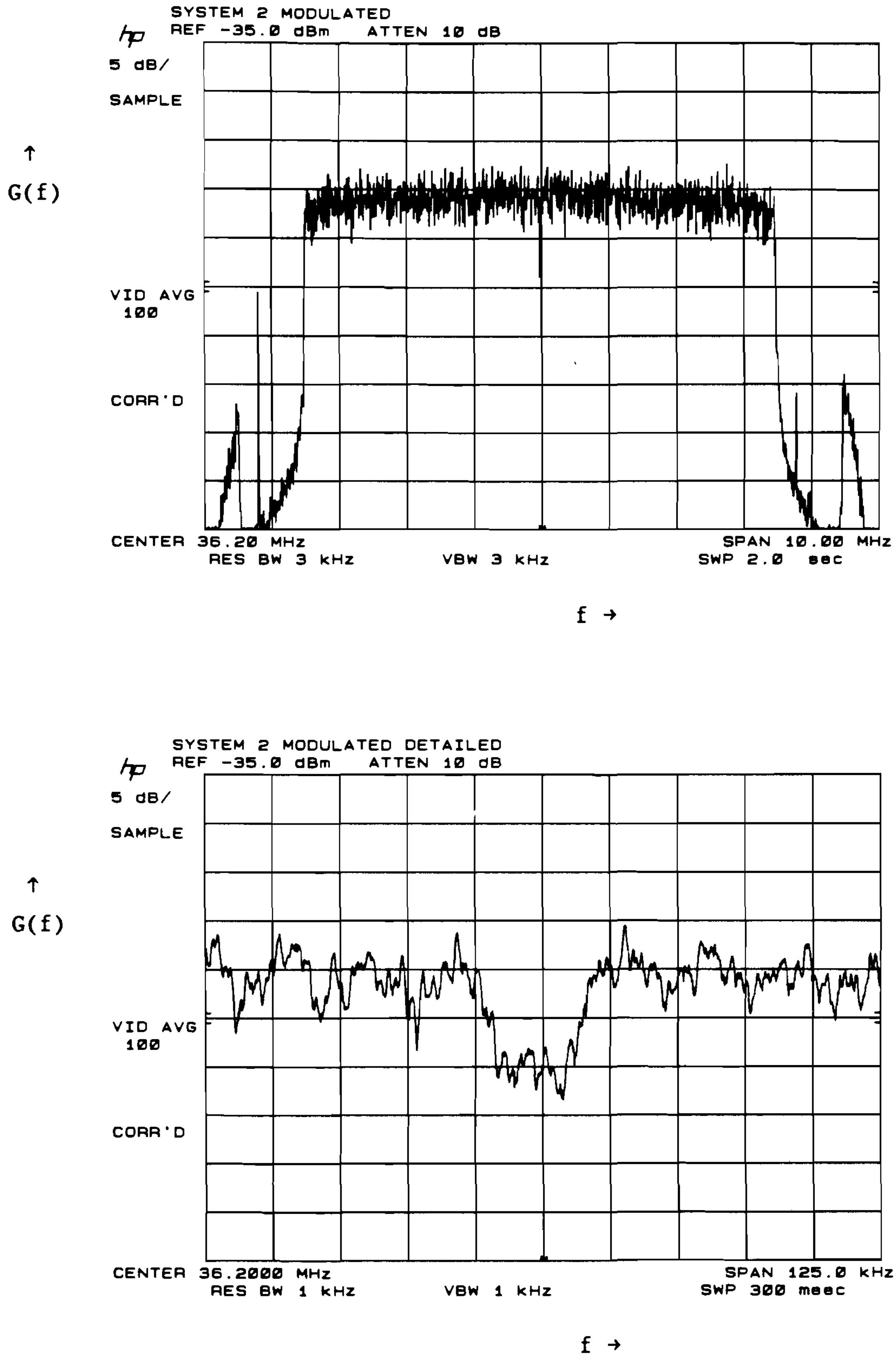
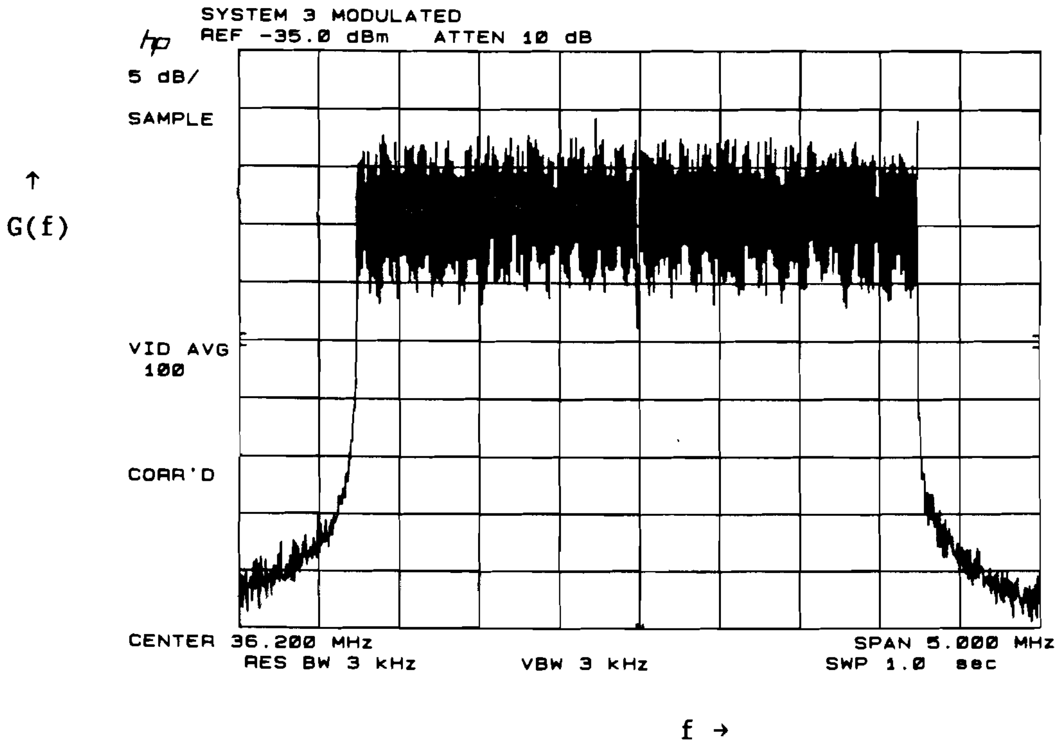


Figure 4.5. Measured IF power spectrum of System-2.



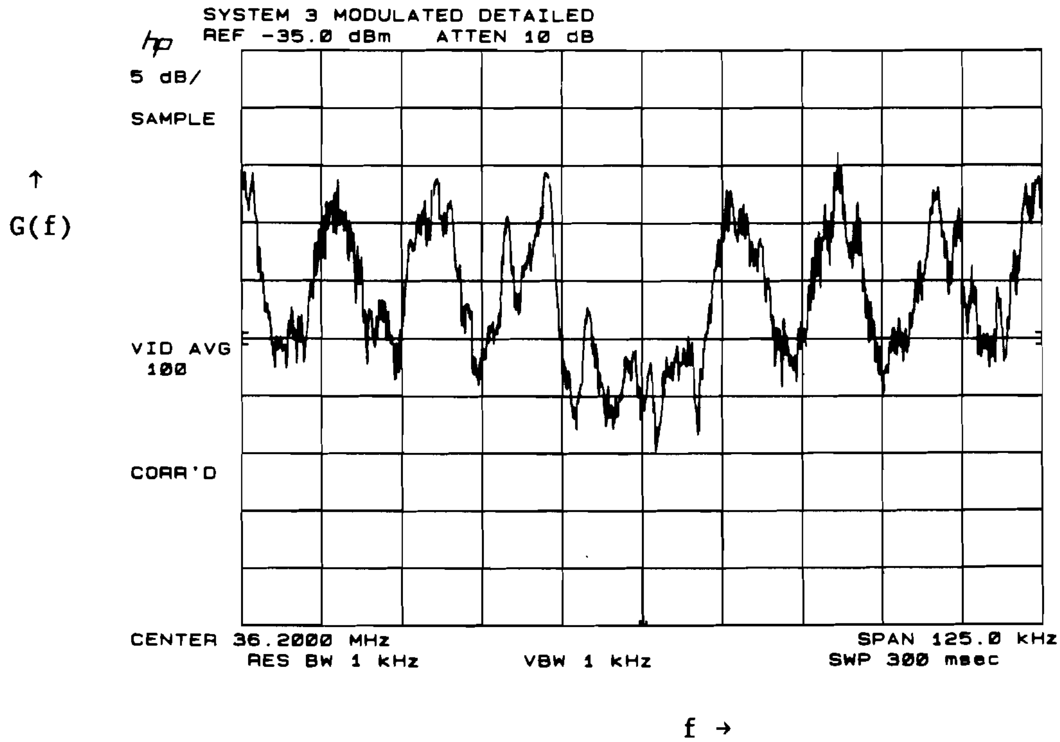
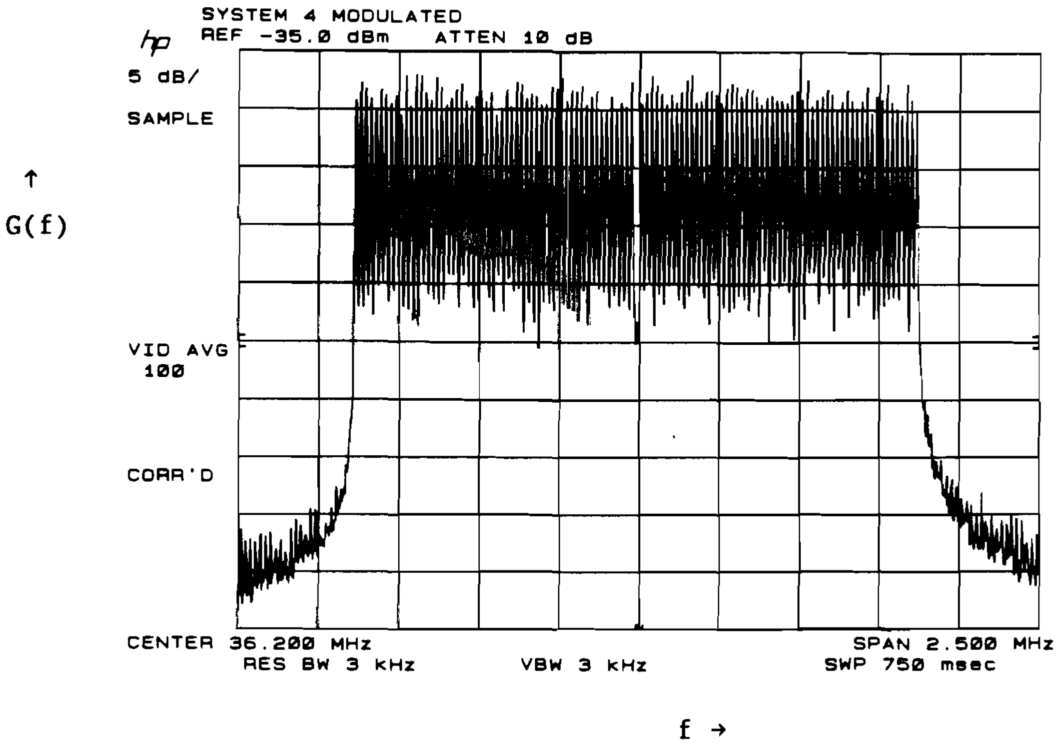


Figure 4.6. Measured IF power spectrum of System-3.



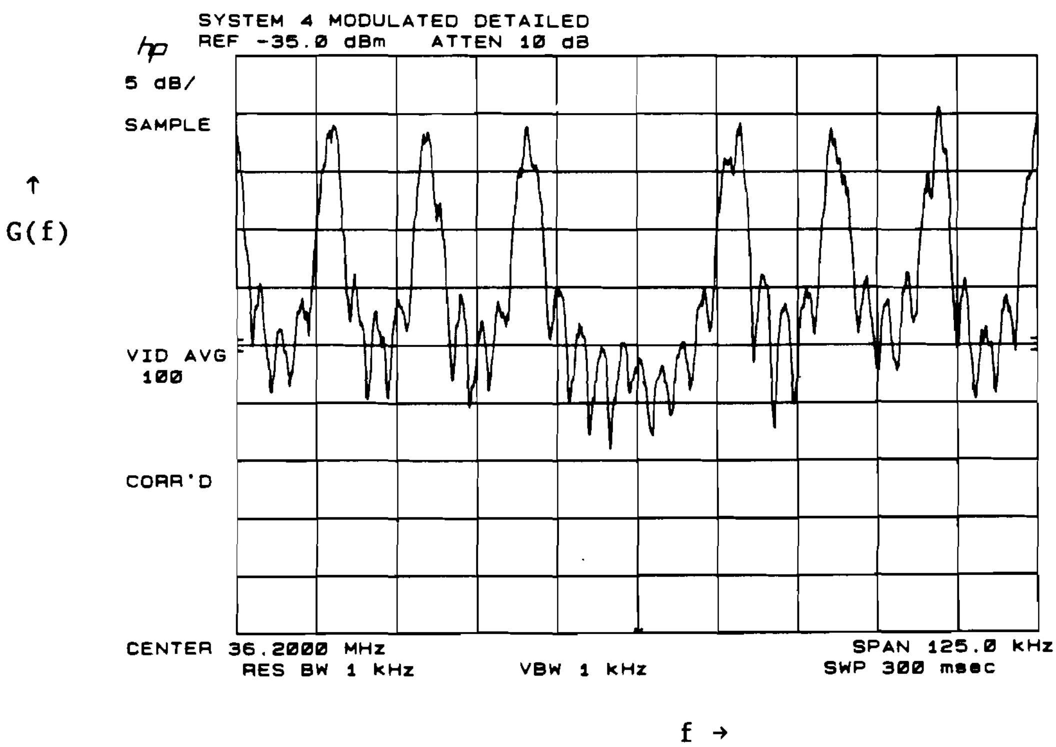


Figure 4.7. Measured IF power spectrum of modulation System-4.

5. DAB SYSTEM-3 AND SYSTEM-4 SIGNAL INTERFERENCE ON CO-CHANNEL TV SIGNALS

5.1. Introduction

The OFDM/(D)QPSK (with guardband interval) modulation systems 3 and 4 are specially designed to share the same radio frequency channels already allocated to television transmitters. The emitted carrier spacing of these systems is equal to the television line frequency. In between those emitted carriers, there is 1 virtual carrier for System-3 and there are 3 virtual carriers for System-4. Due to these virtual carriers, both systems are characterized by a comb shaped frequency power spectrum with a spacing between the peaks equal to the TV line frequency. Since the power of television signals is also concentrated around peaks with a spacing equal to the line frequency, it is possible to interleave the power spectrum of System-3 or System-4 with that of a geographically separated co-channel television transmitter. The expectation is that this interleaving will result in low mutual interference. If this is the case, it might be possible to transmit a DAB System-3 or System-4 signal in a VHF or UHF TV channel, from a transmitter that is not allowed, for interference reasons, to transmit a TV signal in that channel.

This chapter describes a measurement to determine the minimal received TV signal to interleaved DAB interference (System-3 or System-4) power-ratio, whereby distortion is invisible on the demodulated video signal.

5.2. The measurement system

In order to determine the distortion caused by interleaved DAB System-3 or System-4 interference on co-channel TV signals, a measurement system has been built according to the diagram in Figure 5.1.

A TV test pattern generator is used as the TV signal source. This test pattern generator gives an intermediate frequency TV signal with a vision carrier frequency equal to 38.9 MHz and a sound carrier frequency equal to 33.4 MHz. The power of the sound carrier is equal to 1/10 of the vision carrier power. The intermediate frequency TV signal is translated to channel E-5 with the use of a mixer and a signal generator. The frequency of the generated sine wave is equal to 214.15 MHz. This results in a TV signal with a vision carrier of 175.25 MHz and a sound carrier of 179.75 MHz.

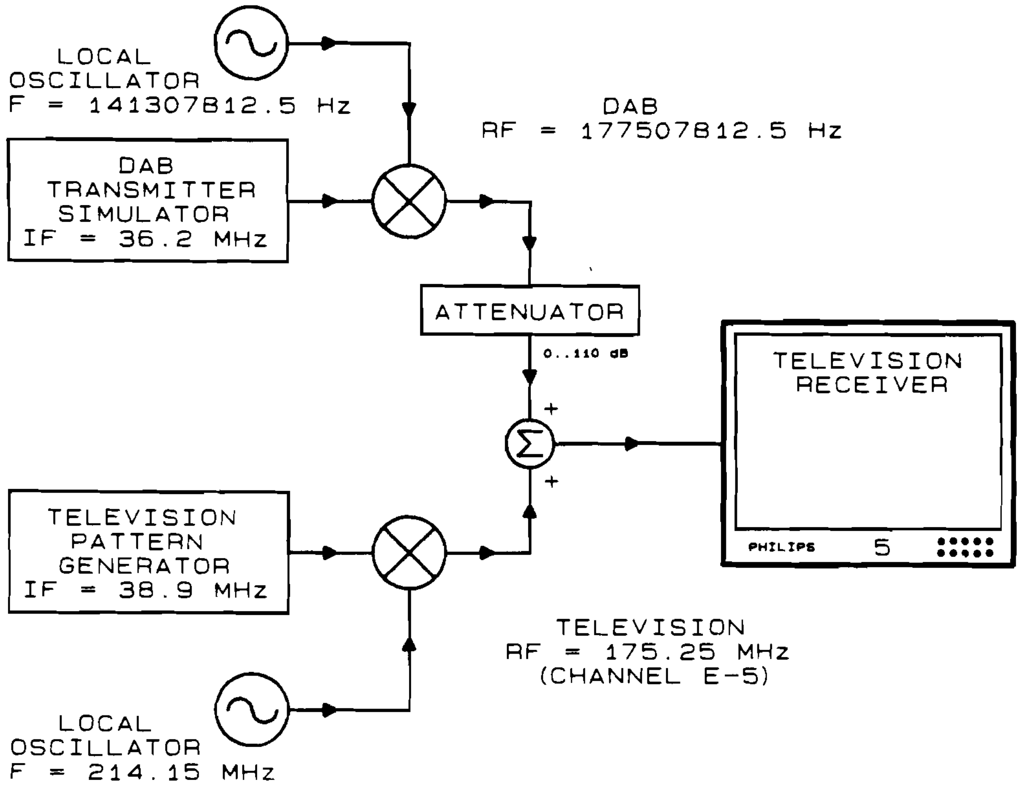


Figure 5.1. The measurement system to measure the co-channel interference of a DAB System-3 or System-4 signal on a TV signal.

The DAB transmitter simulator described in the previous chapter is used as the DAB System-3 or System-4 signal source. The intermediate frequency of the virtual center carrier with number 256 (N/2) is equal to 36.2 MHz for this simulator. A signal generator and a mixer are used to translate the output spectrum of the simulator to the VHF TV channel E-5. The frequency of the sine wave generated by the signal generator is variable. Therefore, it is possible to place the DAB signal spectrum at any position in the TV channel. However, for the co-channel interference measurement, only one frequency (= position) is used. This frequency is chosen as a result of a preliminary experiment. This preliminary experiment will be described in the next paragraph. For this position, the signal generator generates a sine wave signal of 141307812.5 Hz.

Therefore, the radio frequency of the virtual center carrier will be equal to 177507812.5 Hz in this case.

The DAB signal is lead through a variable attenuator before it is added to the TV signal. With this attenuator it is possible to change the interleaved DAB signal power relatively to the TV signal power. A 66 cm colour television receiver is used to demodulate the composed signal and to make it visible. The distortion in the TV picture caused by the DAB signal is judged at one meter distance from the TV screen.

5.3. Co-channel interference measurement

As a preparation for the co-channel interference measurement a preliminary experiment is done. This experiment had the purpose to find the DAB spectrum position within the TV channel which gives minimal distortion in the TV picture. In order to find this position, the TV/DAB signal power-ratio was set to such a level that distortion was clearly visible at any position of the DAB spectrum. After setting the fixed TV/DAB signal power-ratio, the distortion caused by the DAB signal was judged as a function of the DAB spectrum position within the TV channel. This experiment was done for DAB System-3 and System-4 with modulated and unmodulated carriers. Because of the rather subjective TV picture judgement, only qualitative results of this preliminary experiment are given.

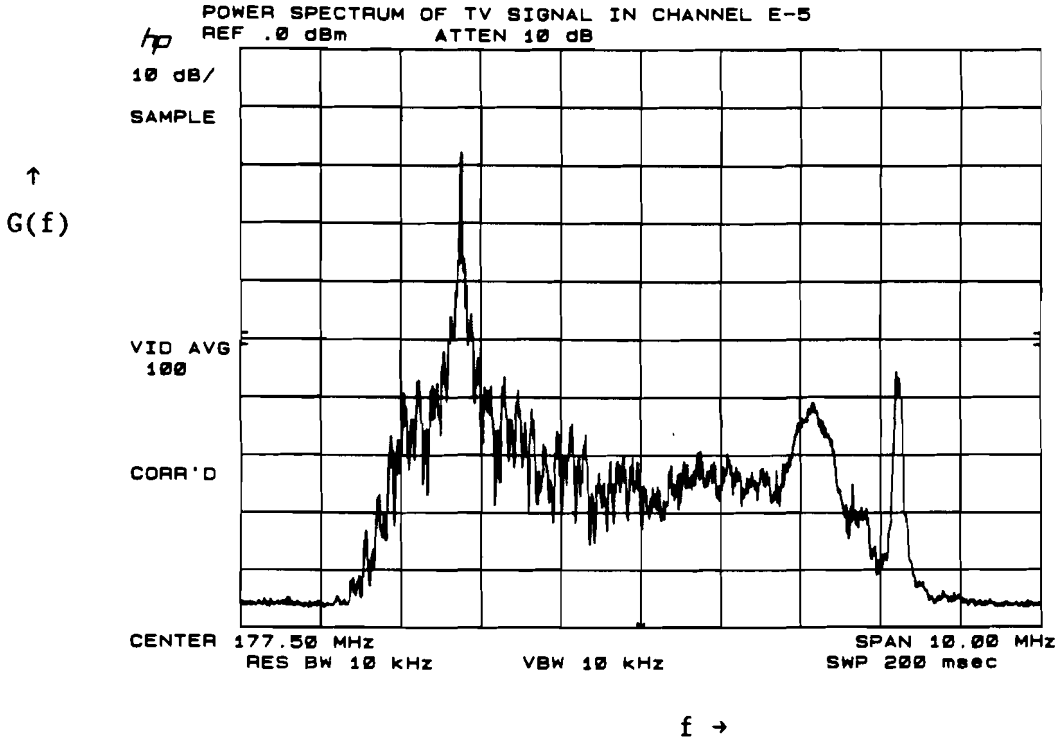
For both systems, high TV picture distortion is found when the DAB carriers are laid on the TV spectrum peaks and low TV picture distortion is found when the DAB carriers are interleaved with the TV spectrum peaks. Unmodulated carriers always give more distortion than modulated carriers. For the positions where the DAB spectrum is interleaved with the TV spectrum, minimal TV picture distortion is found when the DAB spectrum lays just outside the chrominance part in the luminance part of the TV spectrum. For this position there is only luminance distortion. When the DAB spectrum is moved from this minimal distortion position to an interleaved position in the chrominance spectrum part of the TV signal, very inconvenient chrominance distortion is introduced. When the DAB spectrum is moved closer to the TV vision carrier, the luminance distortion at the interleaved positions will slowly increase.

The co-channel interference measurement, to be described now, is done in order to get quantitative results on the TV picture distortion, caused by the interleaved DAB System-3 or System-4 interference on co-channel TV signals. Measured is the minimal TV signal to DAB interference unmodulated power-ratio $(S_{tv}I_{dab}R)_{min}$, whereby TV picture distortion is invisible. The values of $(S_{tv}I_{dab}R)_{min}$ measured for System-3 and System-4 are important parameters. They can be used to calculate the maximum DAB System-3 or System-4 transmitter power, whereby there will be no picture distortion in the service area of a geographically separated TV transmitter that uses the same TV channel.

The measurement is done in the following way. The modulated DAB System-3 or System-4 signal is added to the modulated TV signal in channel E-5. For both systems the frequency of the virtual center carrier is set to 177507812.5 Hz. For both systems this position gives approximately minimal distortion. With the use of the attenuator, the power of the interleaved DAB signal is decreased relatively to the TV signal power until absolutely no more TV picture distortion is visible. At this point, the modulation of both the TV and DAB signal is switched off. This results in unmodulated TV and DAB carriers. A spectrum analyser is then used to measure the unmodulated TV vision carrier power C_{tv} and the unmodulated DAB carrier power C_{dab}. With these measured parameters, the minimal TV signal to DAB interference unmodulated power-ratio $(S_{tv}I_{dab}R)_{min}$ for no TV picture distortion is calculated according to:

$$S_{tv}I_{dab}R = \frac{C_{tv}}{\sum_{k=0}^{N-1} \xi_k \cdot C_{dab}}. \tag{5.1}$$

The Figures 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7 on the following pages illustrate the co-channel interference measurement. All these figures are power spectra, measured with a spectrum analyzer in the E-5 VHF channel. Figure 5.2 shows the TV test pattern signal spectrum. The global spectrum clearly shows the position of the vision carrier and the sound carrier. Also the chrominance part of the spectrum, around 179.75 MHz, can be distinguished. The detailed spectrum clearly shows the power concentration around peaks. These peaks lay on the vision carrier frequency plus a multiple integer of the line frequency.



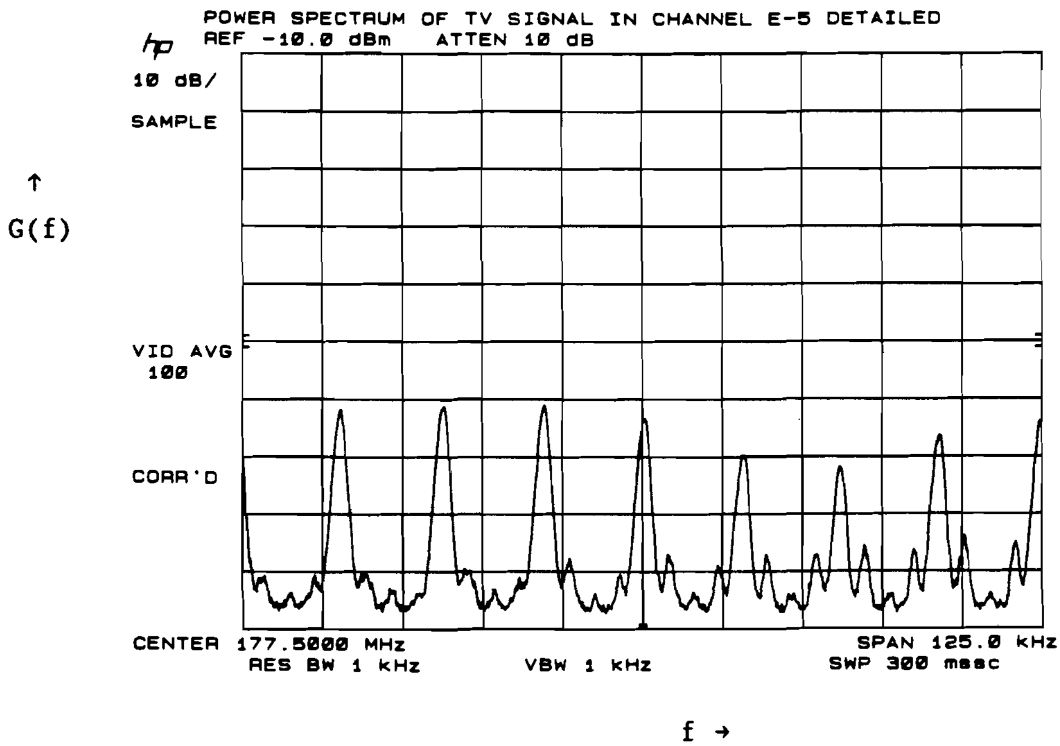
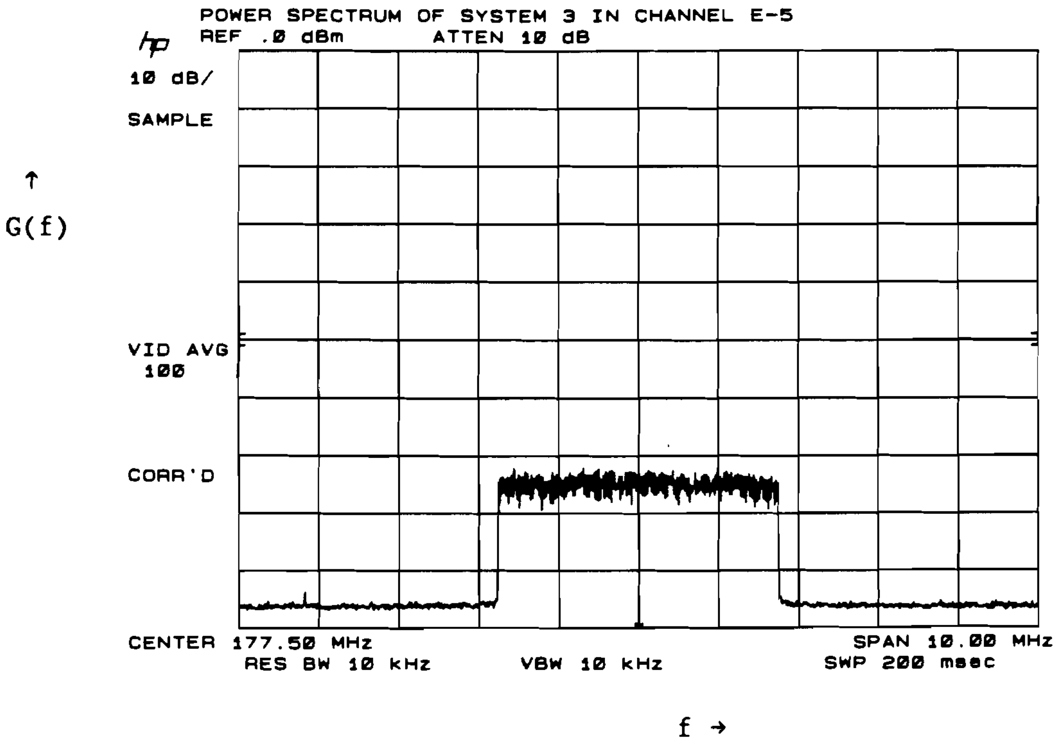


Figure 5.2. Measured RF power spectrum of test pattern TV signal in VHF channel E-5.



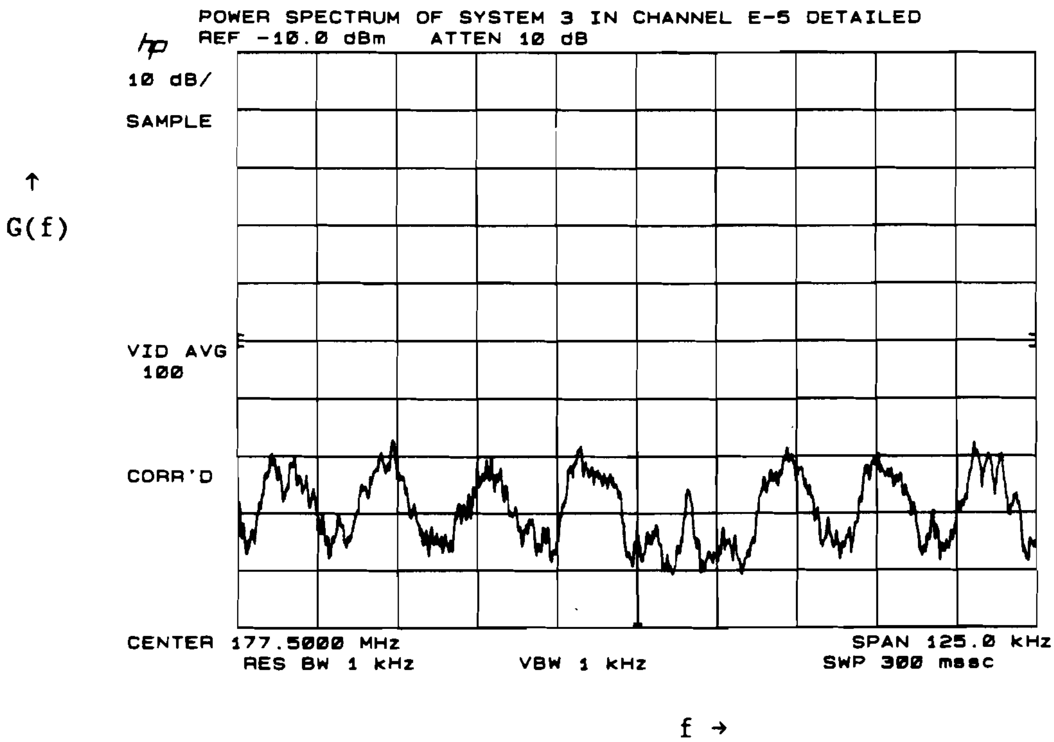
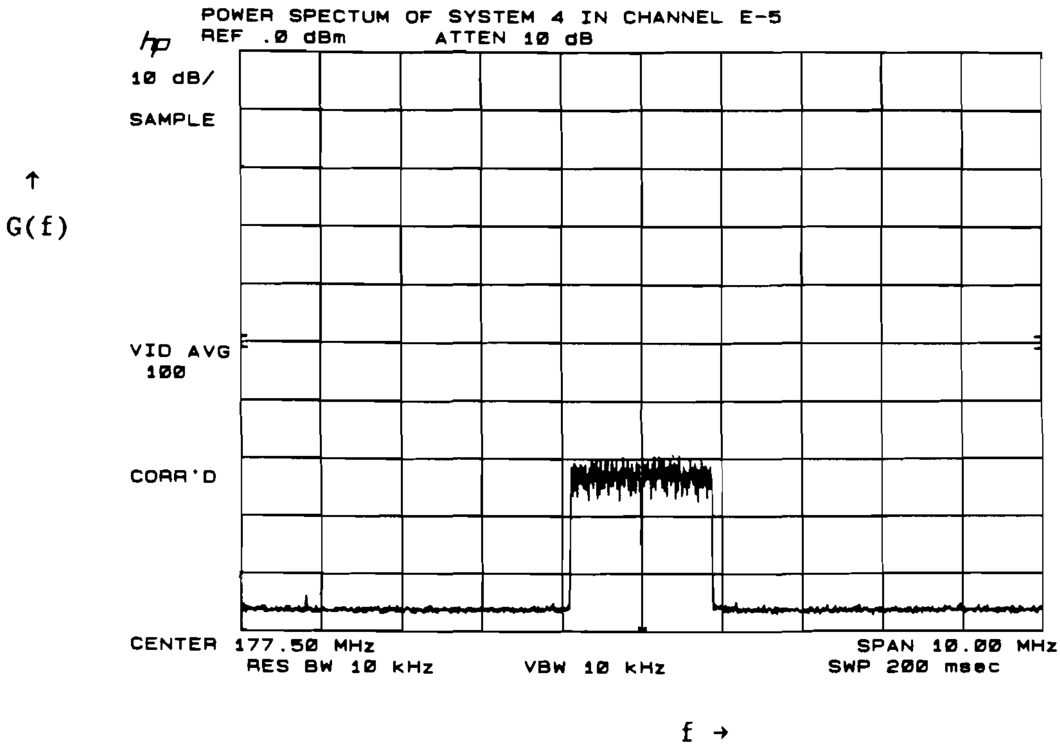


Figure 5.3. Measured RF power spectrum of DAB System-3 signal in VHF channel E-5.



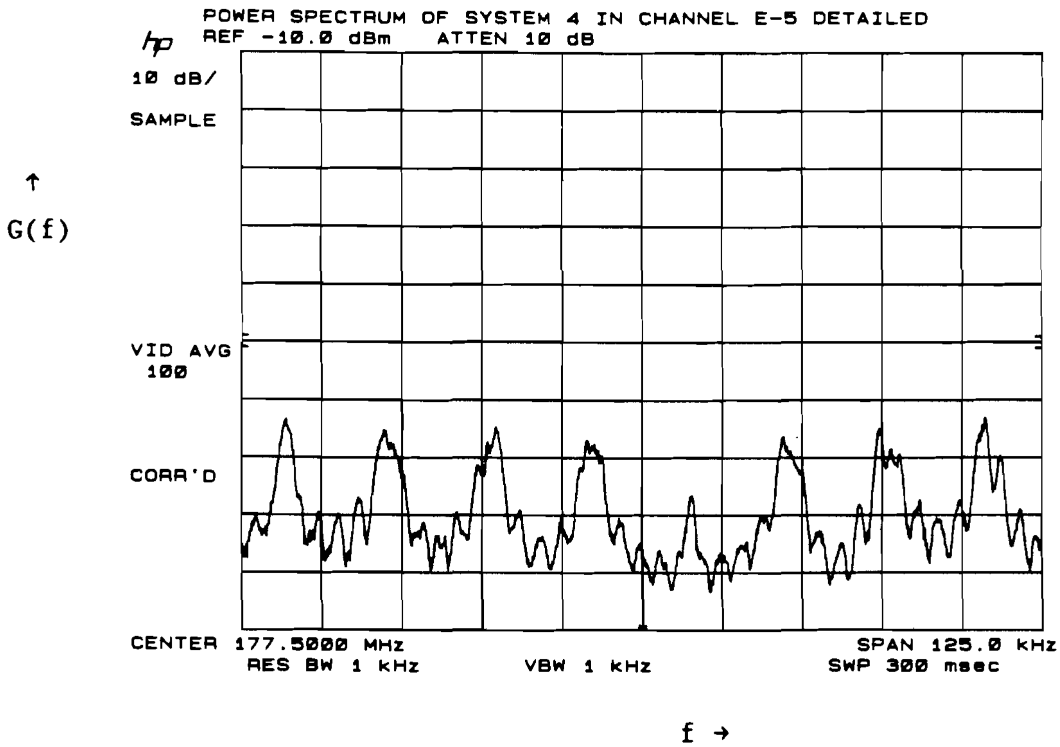


Figure 5.4. Measured RF power spectrum of DAB System-4 signal in VHF channel E-5.

The measured RF spectra of the added DAB Systems-3 and System-4 signals are respectively shown in Figure 5.3. and 5.4. The detailed spectra of these figures show a local oscillator breakthrough signal at the position of the virtual center carrier. This breakthrough signal clearly gives the position of the DAB signal within the frequency domain.

The Figures 5.5 and 5.6 show the spectra of the TV signal plus the interleaved signal of respectively System-3 and System-4. These figures clearly show the frequency interleaving of the DAB signal with the TV signal. The DAB signal power levels shown in these spectra still give visible TV picture distortion.

Figure 5.7 shows the TV and DAB carriers, with the modulation switched off, at the level where no more TV picture distortion is visible for both systems. These spectra give the values of C_{tv} and C_{dab} for both systems at such levels that just no more TV picture distortion is visible. The measured levels are:

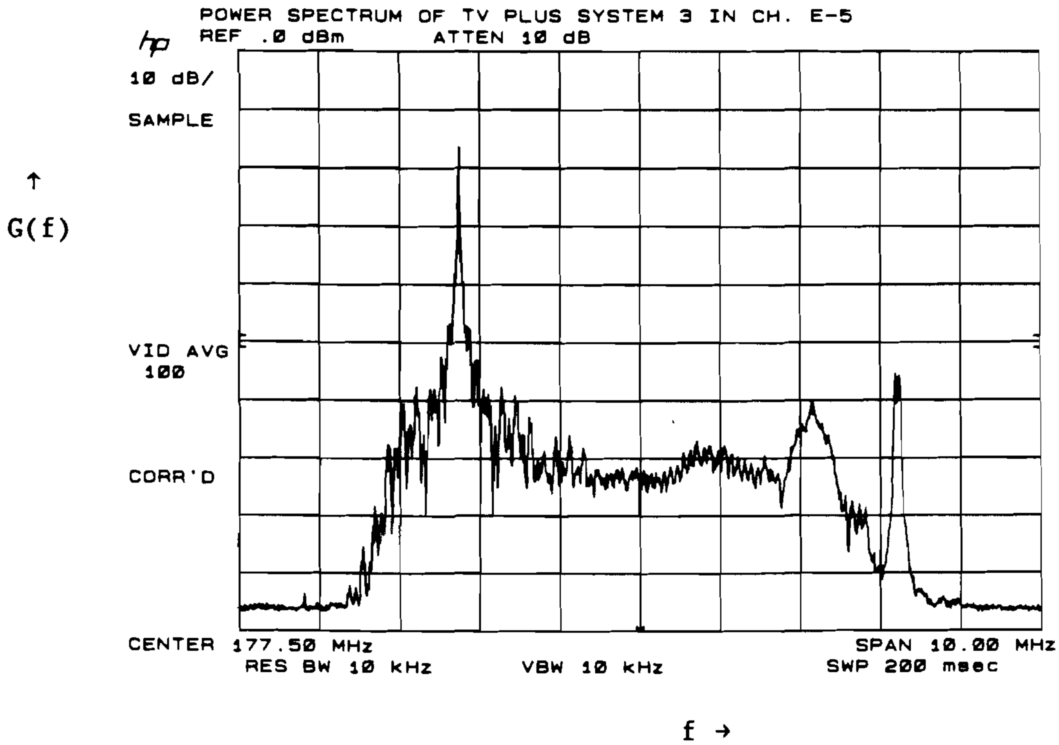
C_{tv} = -10.0 ± 0.5 dBm,

C_{dab}(System-3) = -78 ± 3 dBm,

C_{dab}(System-4) = -75 ± 3 dBm.

The inaccuracy in the C_{dab} levels are introduced to account for the subjective TV picture distortion judgement. These measured values are used to calculate the minimal TV signal to DAB interference unmodulated power-ratio $(S_{tv}I_{dab}R)_{min}$ according to Equation 5.1. For both systems this calculation gives the same result:

$(S_{tv}I_{dab}R)_{min}$ = 54 ± 4 dB.



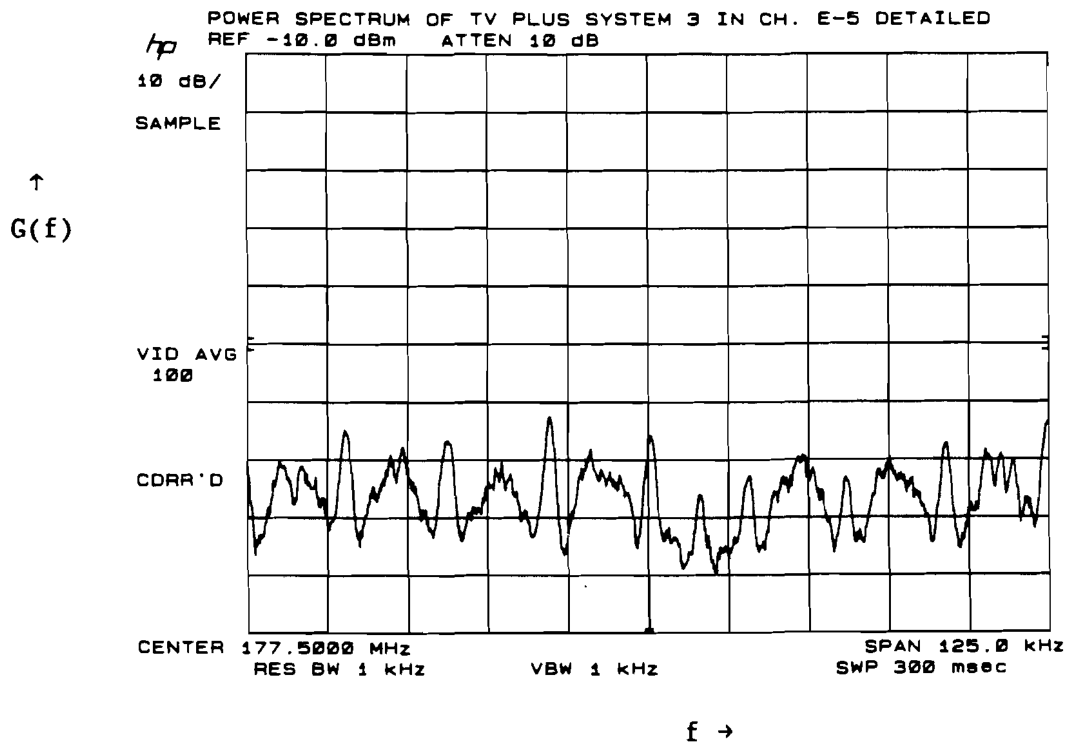
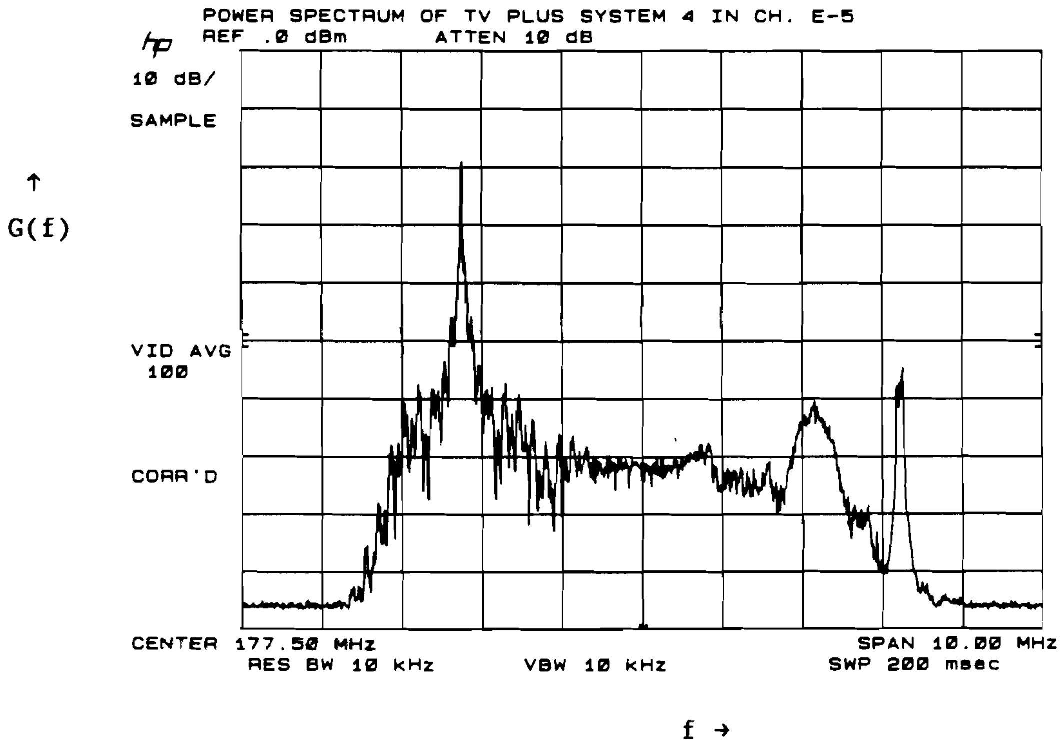


Figure 5.5. Measured RF power spectrum of test pattern TV signal plus interleaved DAB System-3 signal in VHF channel E-5.



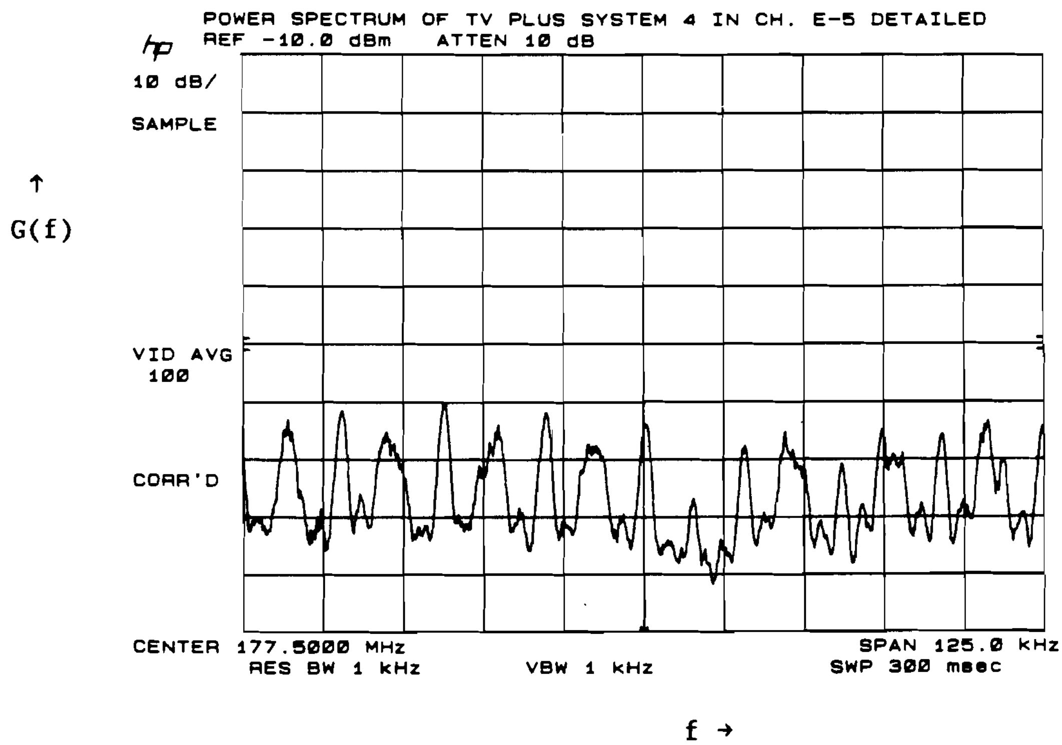
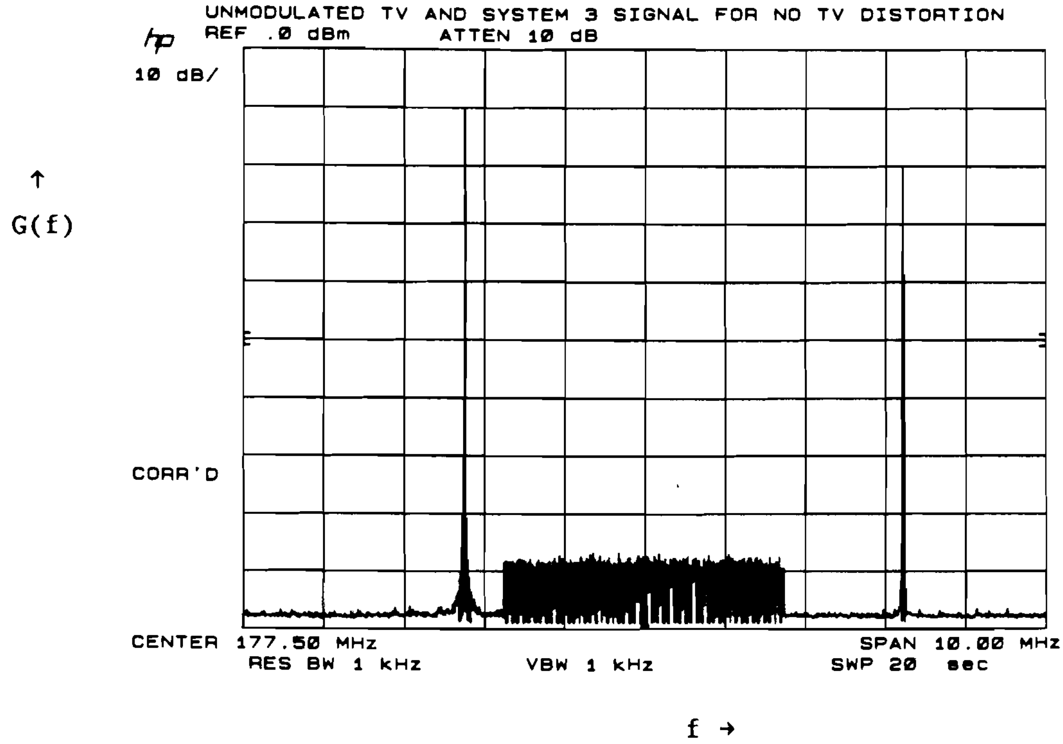


Figure 5.6. Measured RF power spectrum of test pattern TV signal plus interleaved DAB System-4 signal in VHF channel E-5.



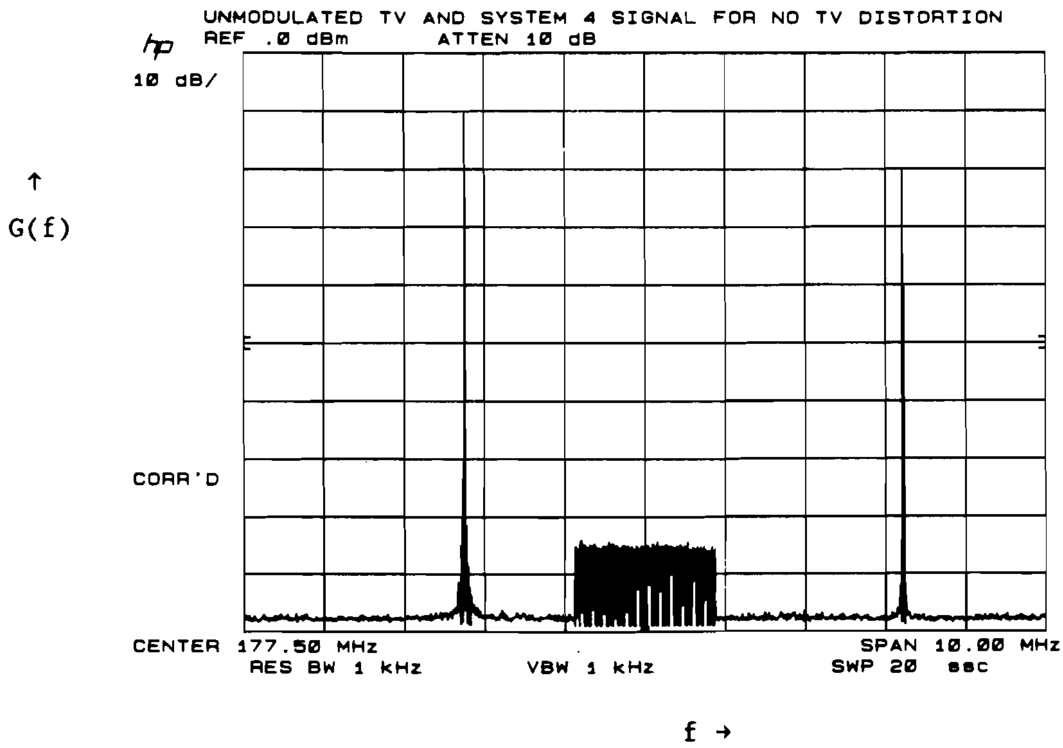


Figure 5.7. Measured RF power spectra of TV and DAB carriers, with the modulation switched off, at the level where no more TV picture distortion is visible for DAB System-3 and System-4.

6. CONCLUSIONS

The DAB transmitter simulator, that has been built during this project, has shown to be capable of generating real-time OFDM/(D)QPSK with guardband interval signals, according to the four systems proposed by the CCETT within the Eureka DAB project. The signals generated by this simulator consist of a periodically repeated frame containing one reference symbol, one or more random data symbols and one null symbol.

The generated signals of System-1 and System-2 are suffering from spurious spectral components. These unwanted spectral components are caused by the low quality interpolation filters. The generated System-3 and System-4 signals do not have these unwanted spectral components, because of the oversampling that compensates for the low quality of the analog interpolating filters.

For System-3 and System-4, both intended for sharing with TV broadcasting, the minimal TV signal to DAB interference unmodulated power-ratio $(S_{tv}I_{dab}R)_{min}$ for invisible TV picture distortion is measured. For both systems the same value of $(S_{tv}I_{dab}R)_{min}$ has been measured:

$$(S_{tv}I_{dab}R)_{min} = 54 \pm 4 \text{ dB.}$$

The inaccuracy in this value is mainly caused by the subjective TV picture distortion judgement.

The value of $(S_{tv}I_{dab}R)_{min}$ measured for both System-3 and System-4 is an important parameter. It can be used to calculate the maximum DAB System-3 or System-4 transmitter power, whereby there will be no picture distortion in the service area of a geographically separated TV transmitter that uses the same TV channel. Now consider a DAB System-3 or System-4 transmitter radiating with that calculated maximum power. For this transmitter it is still not possible to define a DAB service area that is not disturbed by the co-channel TV signal. This can only be done if the minimal DAB signal to TV interference power-ratio

$(S_{dab}I_{tv}R)_{min}$ is known, whereby the DAB bit error-ratio is still acceptable. In order to measure $(S_{dab}I_{tv}R)_{min}$ for System-3 and System-4, a DAB System-3 and System-4 OFDM/(D)QPSK with guardband interval demodulator is needed to enable bit error-rate measurements.

If a receiver part (hardware and software) is added to the existing DAB transmitter simulator, it would be possible to measure the $(S_{dab}I_{tv}R)_{min}$ for System-3 and System-4. However, it would still not be possible to do accurate measurements (interference on and from neighbour-channel TV signals) with System-1 and System-2 due to the low generated signal quality.

For further research, it is advised to build a new and complete DAB transceiver simulator. The sample frequency, of the OFDM/(D)QPSK with guardband interval modulator in the new transmitter, should minimally be 16 MHz in order to get high quality System-1 and System-2 signals (oversampling ratio = 2). It is also advised to increase the frame length in order to increase the randomness of the transmitted signal. This is especially needed for the System-1 signal, generated with the existing simulator, that contains only one random data symbol. With such a complete transceiver, it will be possible to do bit error-rate measurements for all systems in different channels. It will also enable all sorts of mutual interference measurements between DAB and TV signals. Such measurements are very important for the frequency planning of a potential DAB system.

REFERENCES

[1] P. Treytl (Editor),
Digital sound broadcasting by satellite, second revised version.
Bonn: Federal Ministry of Research and Technology.

[2] G. Theile, G. Stoll and M. Link,
"Low bit-rate coding of high-quality audio signals - An introduction to the MASCAM system",
EBU Review - Technical, No. 230, pp. 158-181, August 1988.

[3] R. J. Levey (Editor),
Advanced digital techniques for UHF satellite sound broadcasting (collected papers on concepts for sound broadcasting into the 21st century).
Geneva: EBU, 1988.

[4] J. G. Proakis,
Digital communications.
Singapore: McGraw-Hill, 1983.

[5] D. Pommier and Yi Wu,
"Interleaving or spectrum-spreading in digital radio intented for vehicles",
EBU Review - Technical, No. 217, pp. 128-142, June 1986.

[6] G. L. Turin,
"Introduction to spread spectrum anti-multipath techniques and their application to urban digital radio",
Proceedings of the IEEE, Vol. 68, No. 3, pp. 328-353, March 1980.

[7] M. Alard and R. Lassalle,
"Principles of modulation and channel coding for digital broadcasting for mobile receivers",
EBU Review - Technical, No. 224, pp. 168-190, August 1987.

[8] Ch. Dosch, P. A. Ratliff and D. Pommier,
"First public demonstrations of COFDM/MASCAM. A milestone for the future of radio broadcasting",
EBU Review - Technical, No. 232, pp. 275-283, December 1988.

[9] S. B. Weinstein and P. M. Ebert,
"Data transmission by frequency-division multiplexing using the discrete Fourier transform",
IEEE Trans. Commun. Technol., Vol. COM-19, No. 5, pp. 628-634, October 1971.

[10] L. J. Cimini,
"Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing",
IEEE Trans. Commun., Vol. COM-33, No. 7, pp. 665-675, July 1985.

A. DAB MODULATION SYSTEMS

A.1. System-1

MODULATION SCHEME .... OFDM/DQPSK with guardband interval

useful signal duration T_s .... 1024 µs
guardband interval Δ .... 256 µs
total signal duration T'_s .... 1280 µs

number of carriers (emitted + virtual) ... 8192
number of emitted carriers 7168
number of virtual carriers 1024
emitted carrier numbers:

{512, 513, 514, ..., 4095} and
{4097, 4098, 4099, ..., 7680}

minimal carrier spacing .... 976.5625 Hz
minimal emitted carrier spacing .... 976.5625 Hz
bandwidth .... 7 MHz
frequency allocation range .... 0 - 125 MHz

number of programmes 16
planning method:

single frequency network giving 16 national programmes per frequency allocation

A.2. System-2

MODULATION SCHEME .... OFDM/DQPSK with guardband interval

useful signal duration T_s .... 64 μs
guardband interval Δ .... 16 μs
total signal duration T_s' .... 80 μs

number of carriers (emitted + virtual) ... 512
number of emitted carriers 448
number of virtual carriers 64
emitted carrier numbers:

{32, 33, 34, ..., 255} and
{257, 258, 259, ..., 480}

minimal carrier spacing .... 15625 Hz
minimal emitted carrier spacing .... 15625 Hz
bandwidth .... 7 MHz
frequency allocation range .... 0 - 2 GHz

number of programmes 16
planning method:

conventional giving 16 local programmes per frequency allocation

A.3. System-3

MODULATION SCHEME .... OFDM/DQPSK with guardband interval

useful signal duration T_s .... 128 μs
guardband interval Δ .... 32 μs
total signal duration T'_s .... 160 μs

number of carriers (emitted + virtual) ... 512
number of emitted carriers 224
number of virtual carriers 288
emitted carrier numbers:
{32, 34, 36, ..., 254} and
{258, 260, 262, ..., 480}

minimal carrier spacing .... 7812.5 Hz
minimal emitted carrier spacing .... 15625 Hz
bandwidth .... 3.5 MHz
frequency allocation range .... 0 - 1 GHz

number of programmes 4
planning method:
sharing with TV broadcasting giving 4 local programmes per frequency allocation

A.4. System-4

MODULATION SCHEME .... OFDM/DQPSK with guardband interval

useful signal duration T_s .... 256 µs
guardband interval Δ .... 64 µs
total signal duration T'_s .... 320 µs

number of carriers (emitted + virtual) ... 512
number of emitted carriers 112
number of virtual carriers 400
emitted carrier numbers:

{32, 36, 40, ..., 252} and
{260, 264, 268, ..., 480}

minimal carrier spacing .... 3906.25 Hz
minimal emitted carrier spacing .... 15625 Hz
bandwidth .... 1.75 MHz
frequency allocation range .... 0 - 500 MHz

number of programmes 1
planning method:

sharing with TV broadcasting giving 1 local programmes per frequency allocation

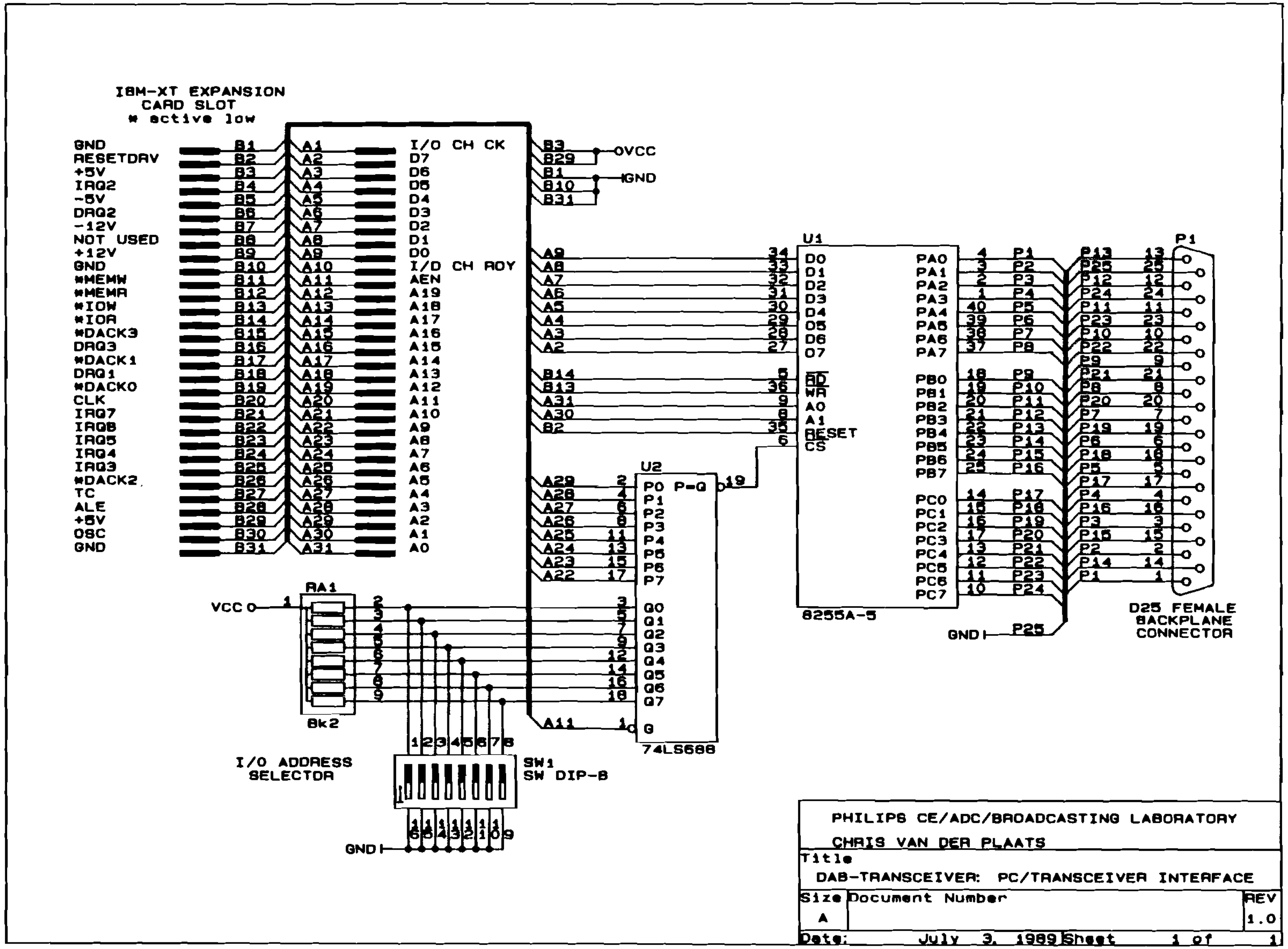
B. DAB-TRANSCEIVER SCHEMATICS

B.1. Transceiver bus definition

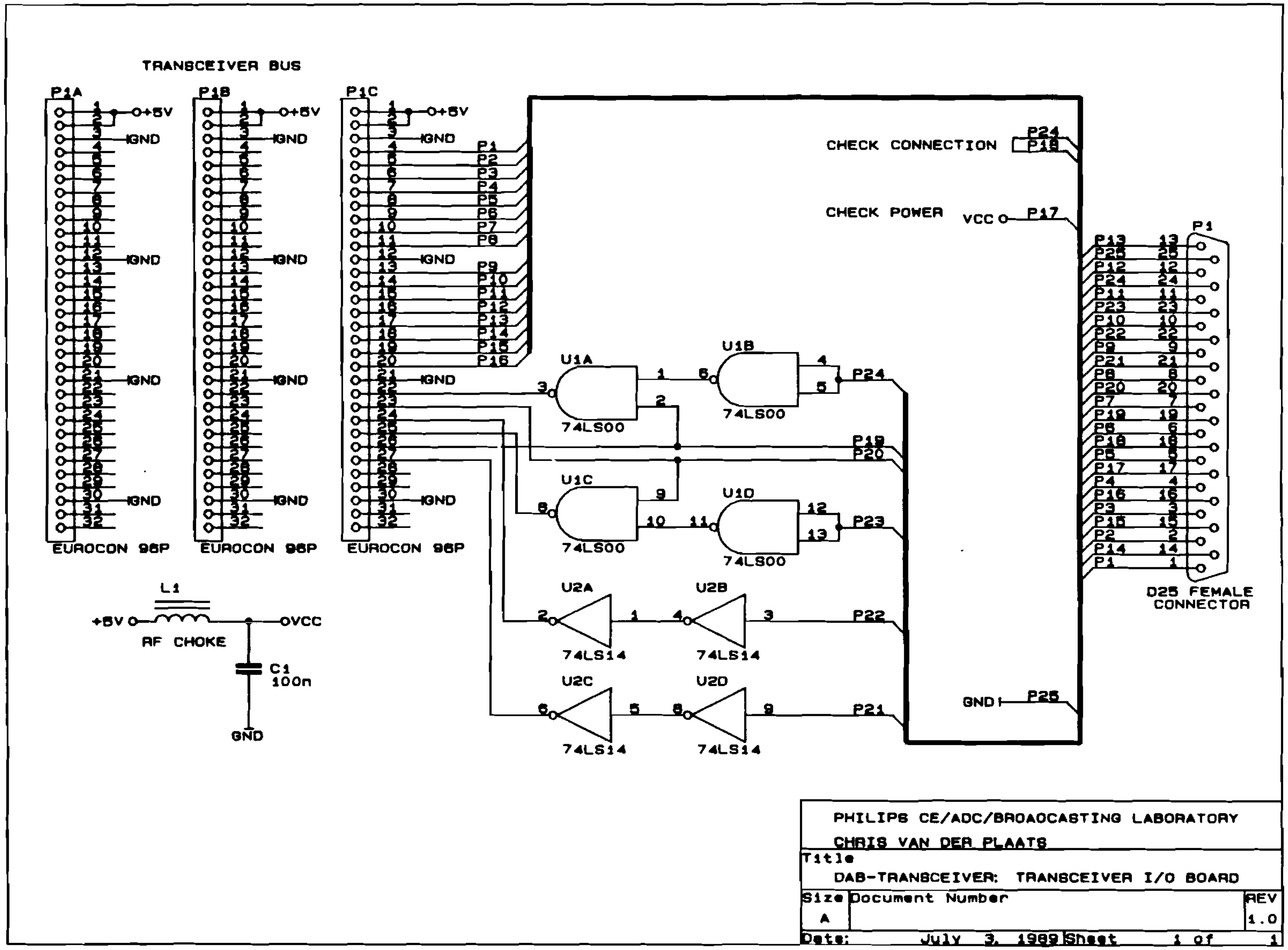
96 PIN EUROCONNECTOR

1A	1B	1C	+5V	- +5V POWER SUPPLY
2A	2B	2C	+5V	- +5V POWER SUPPLY
3A	3B	3C	GND	- GROUND
4A	4B	4C	RD0	LSB
5A	5B	5C	RD1	
6A	6B	6C	RD2	
7A	7B	7C	RD3	8 BIT BIDIRECTIONAL DATABUS
8A	8B	8C	RD4	FOR THE REAL SIGNAL PART
9A	9B	9C	RD5	
10A	10B	10C	RD6	
11A	11B	11C	RD7	MSB
12A	12B	12C	GND	- GROUND
13A	13B	13C	ID0	LSB
14A	14B	14C	ID1	
15A	15B	15C	ID2	
16A	16B	16C	ID3	8 BIT BIDIRECTIONAL DATABUS
17A	17B	17C	ID4	FOR THE IMAGINARY SIGNAL PART
18A	18B	18C	ID5	
19A	19B	19C	ID6	
20A	20B	20C	ID7	MSB
21A	21B	21C	GND	- GROUND
22A	22B	22C	WTRREQ	- WRITE TRANSMITTER RAM REQUEST
23A	23B	23C	WTRACK	- WRITE TRANSMITTER RAM ACKNOWLEDGE
24A	24B	24C	WTRSTR	- WRITE TRANSMITTER RAM STROBE
25A	25B	25C	RRRREQ	- READ RECEIVER RAM REQUEST
26A	26B	26C	RRRACK	- READ RECEIVER RAM ACKNOWLEDGE
27A	27B	27C	RRRSTR	- READ RECEIVER RAM STROBE
28A	28B	28C	2*CLK	- 16 MHz CLOCK FROM TRANSMITTER
29A	29B	29C	FSYNC	- FRAME SYNC PULSE FROM TRANSMITTER
30A	30B	30C	GND	- GROUND
31A	31B	31C	-5V	- -5V POWER SUPPLY
32A	32B	32C	-5V	- -5V POWER SUPPLY

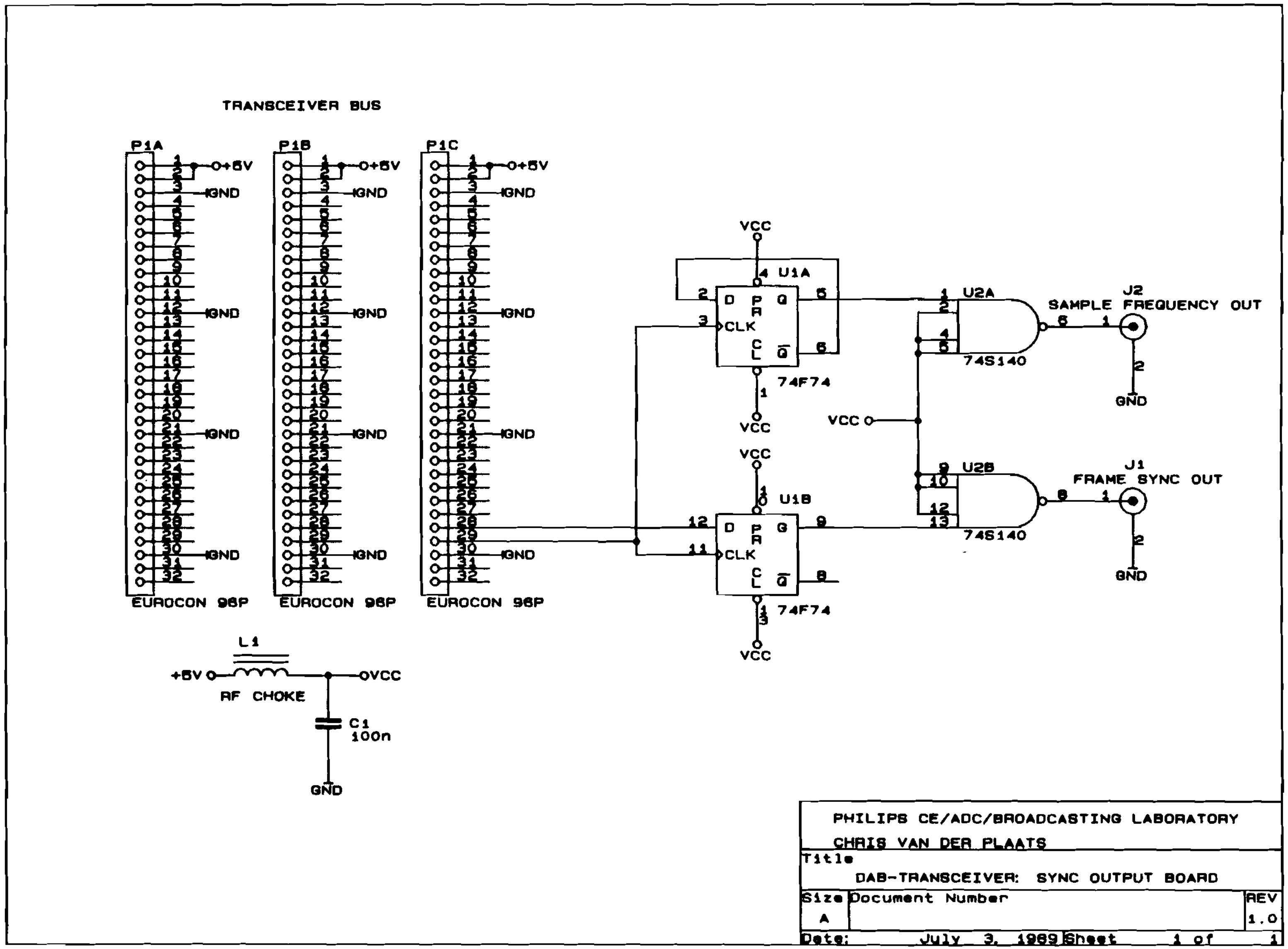
B.2. Schematic of the "PC/TRANSCEIVER INTERFACE"



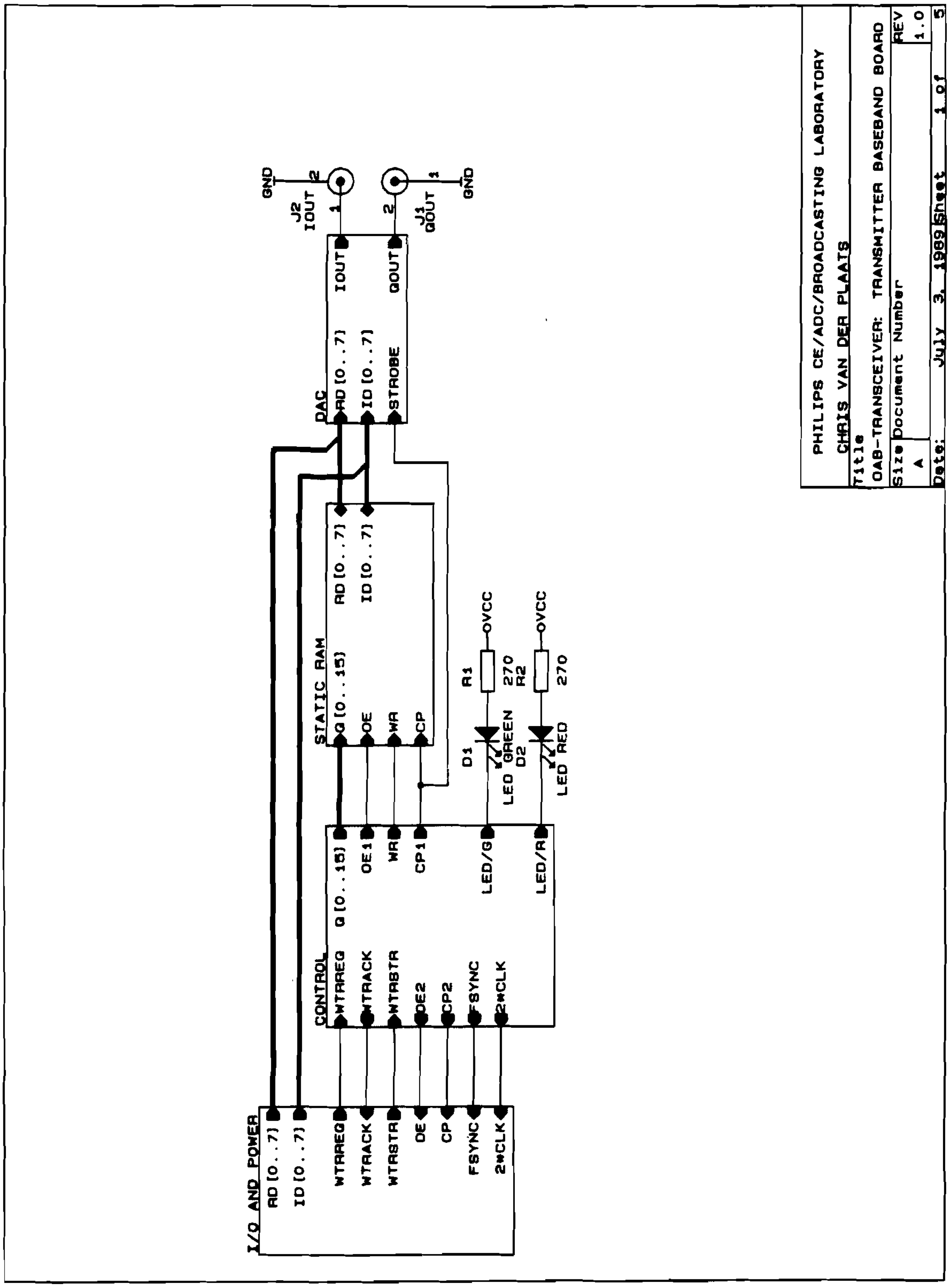
B.3. Schematic of the "TRANSCEIVER I/O BOARD"



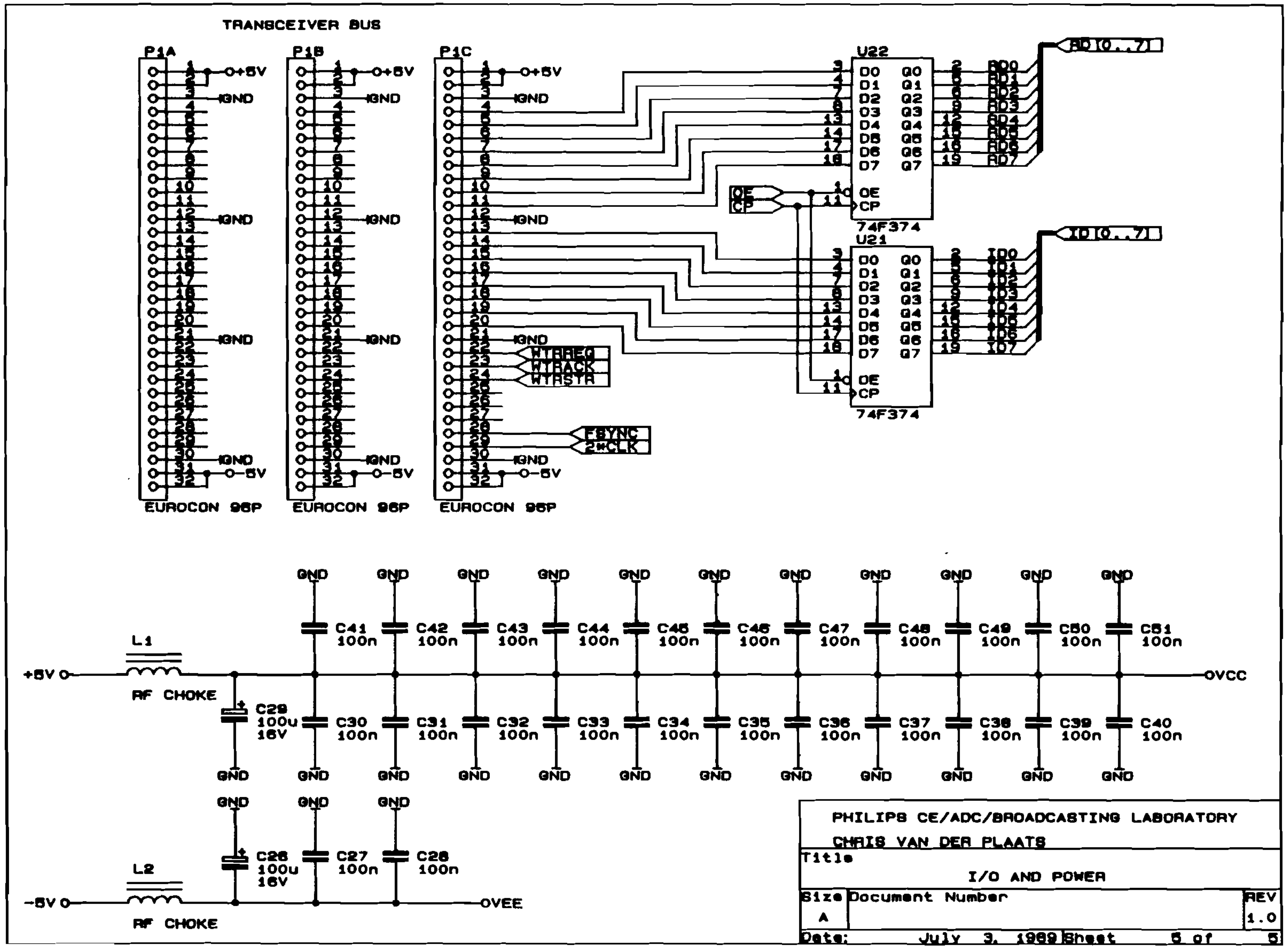
B.4. Schematic of the "SYNC OUTPUT BOARD"



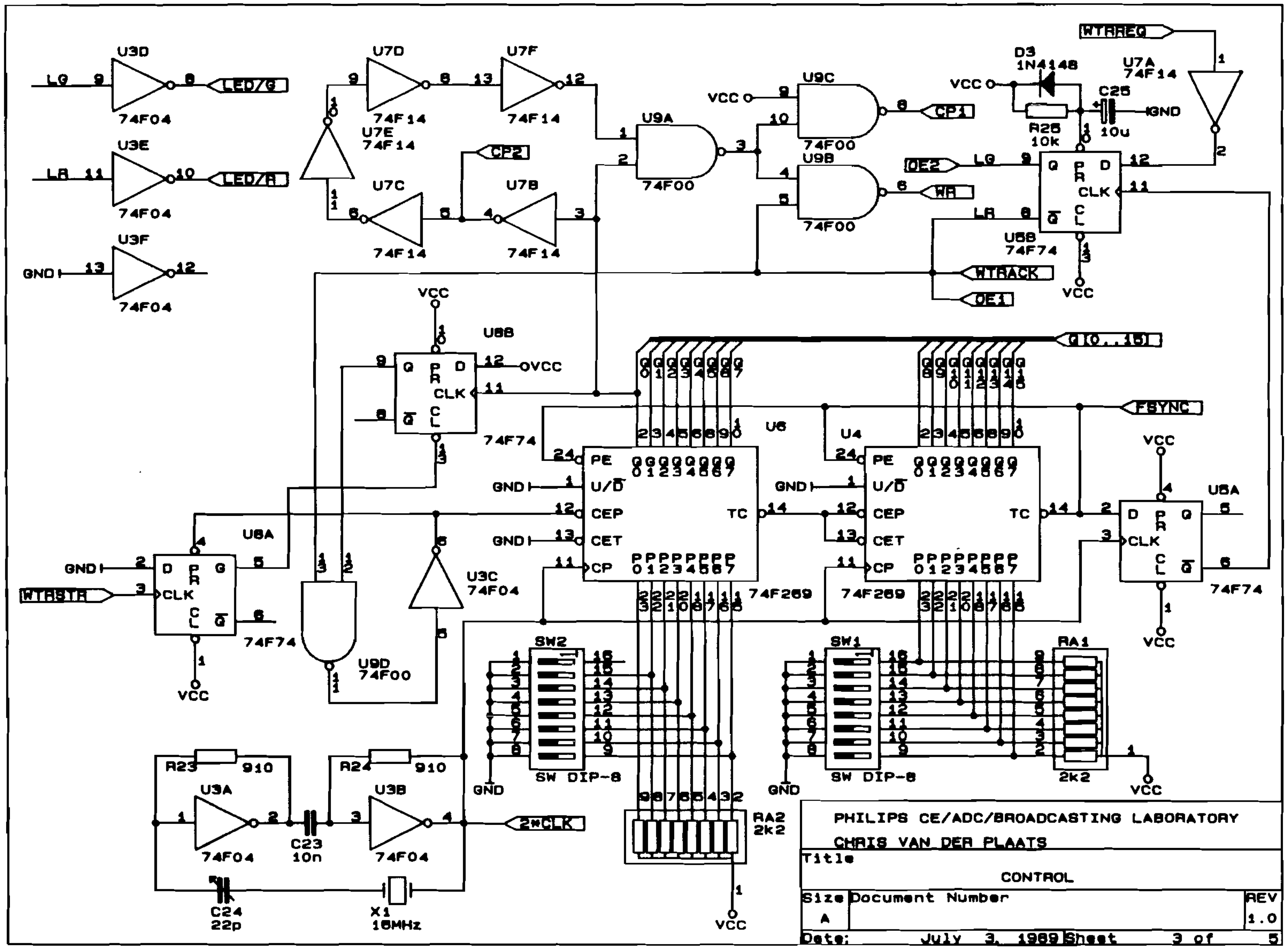
B.5. Schematic of the "TRANSMITTER BASEBAND BOARD"



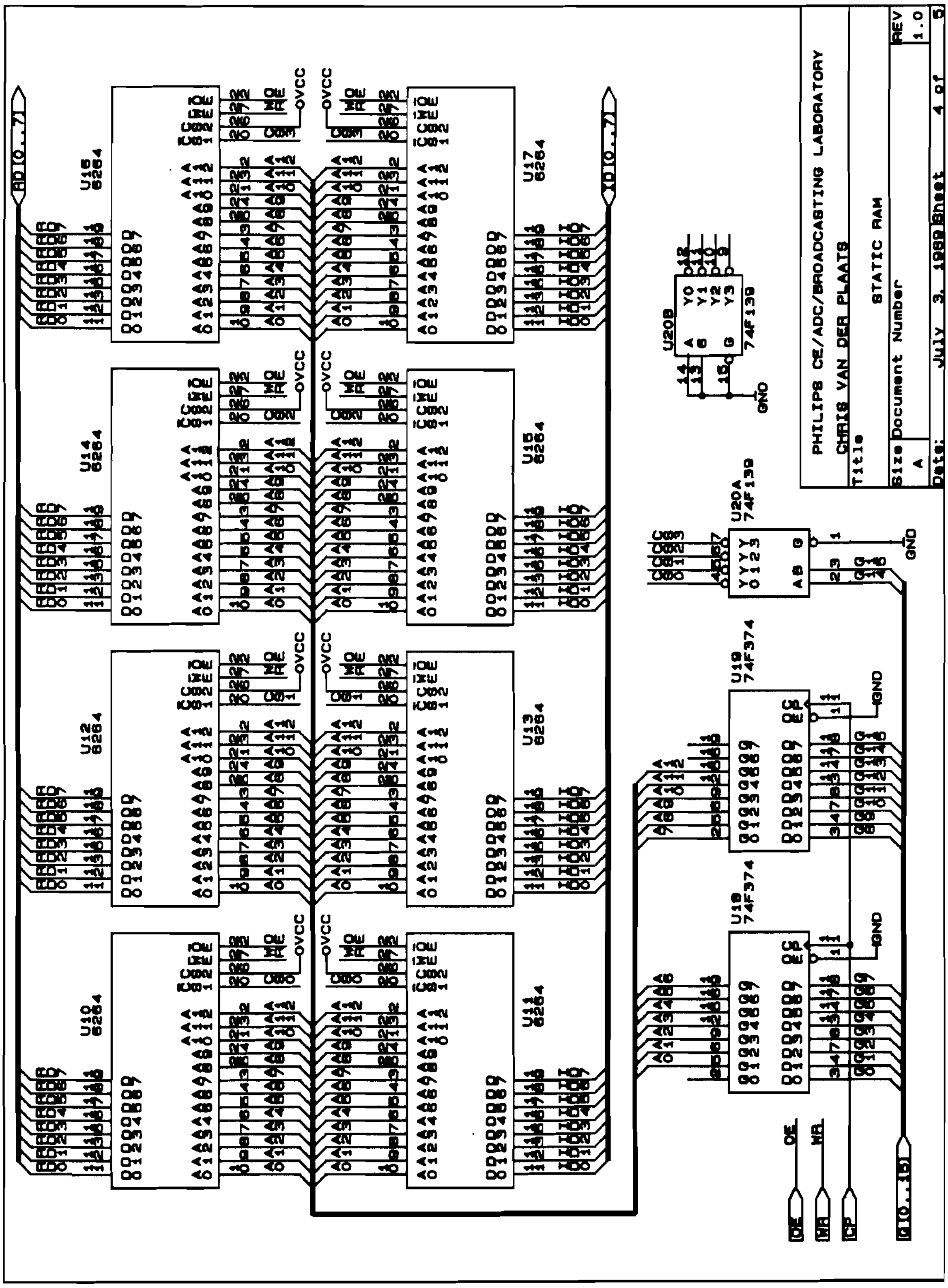
This schematic has four sub-sheets: "I/O and POWER", "CONTROL", "STATIC RAM" and "DAC".



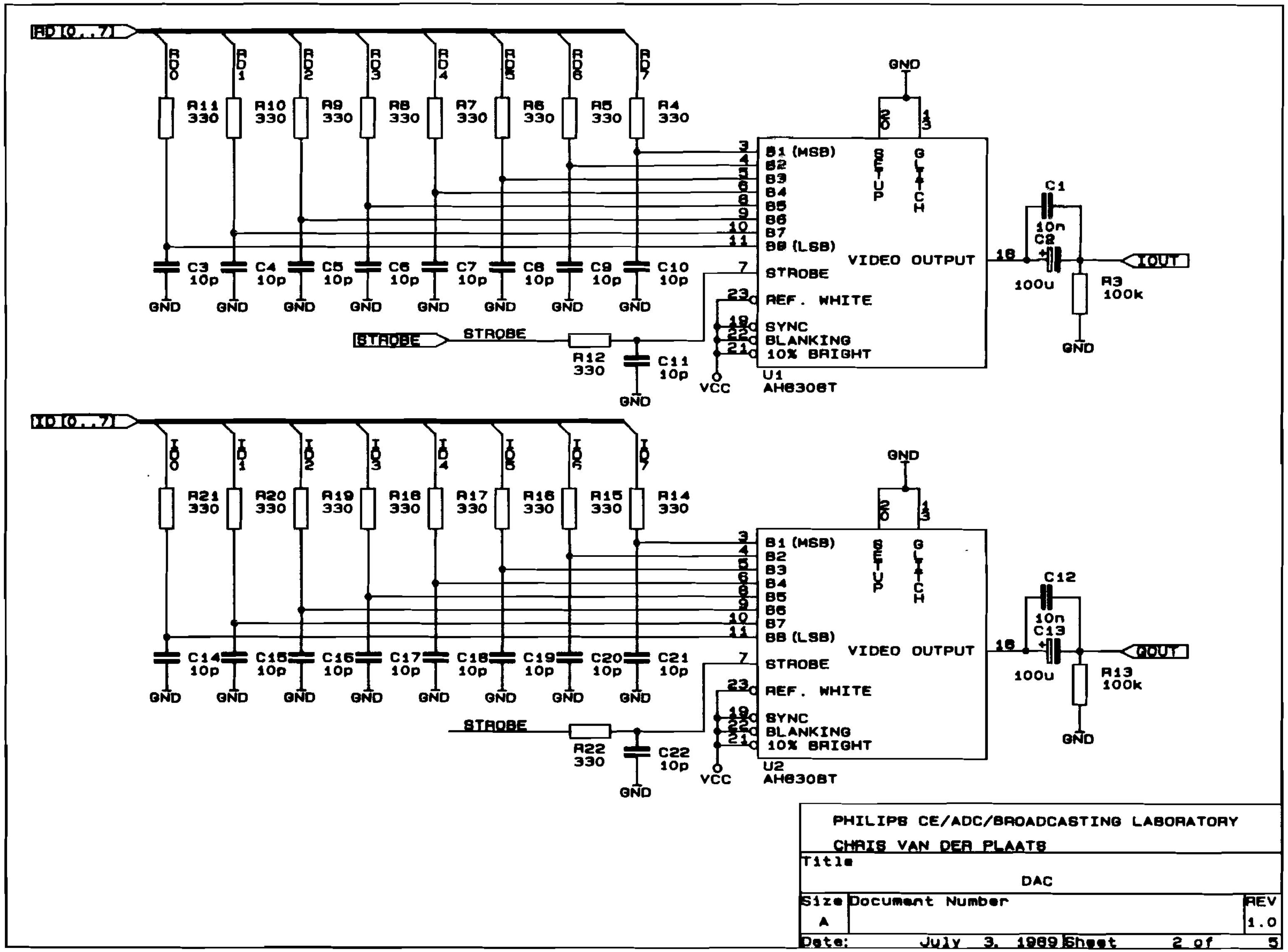
Sub-sheet "I/O and POWER".



Sub-sheet "CONTROL".

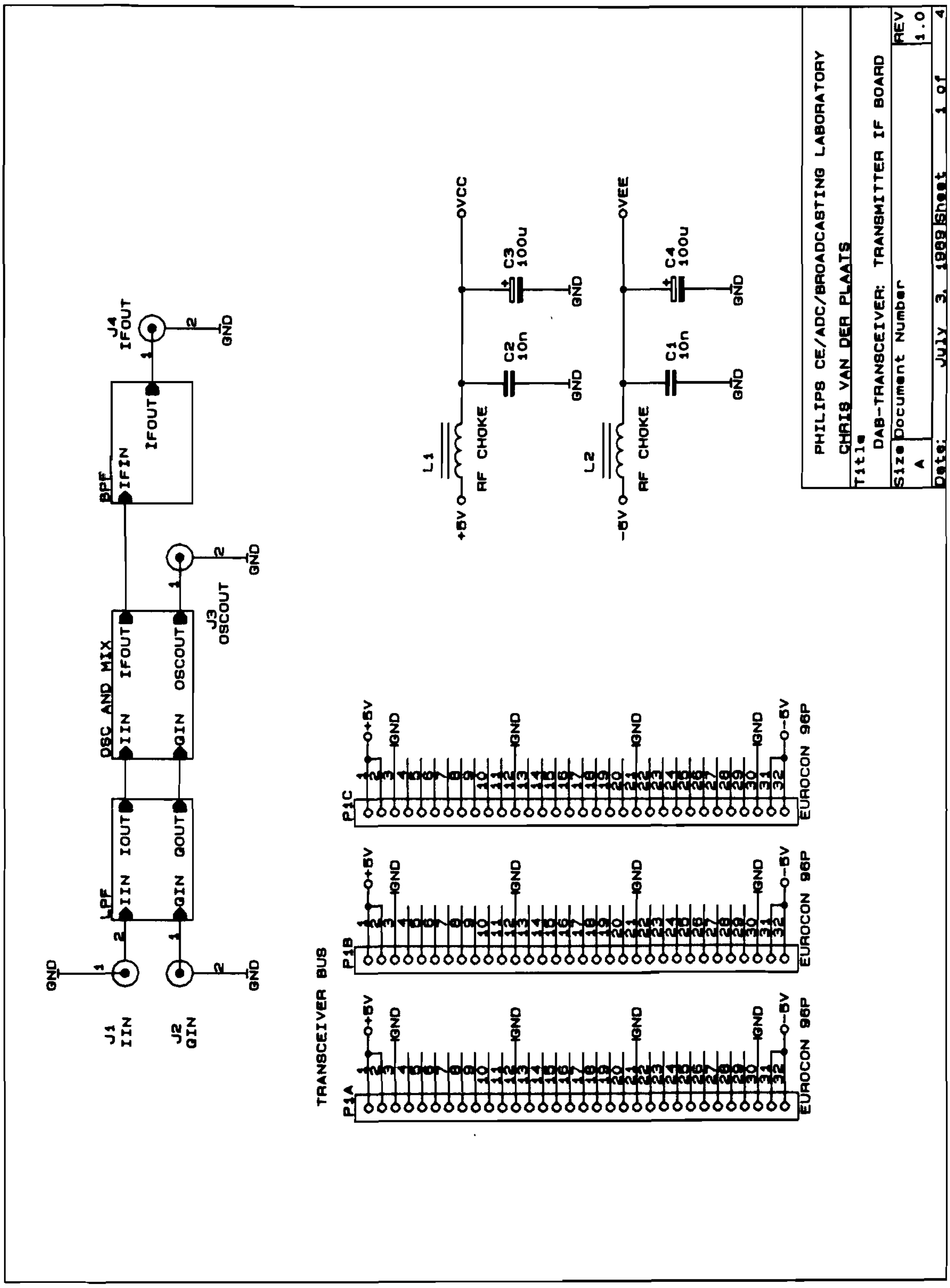


Sub-sheet "STATIC RAM".

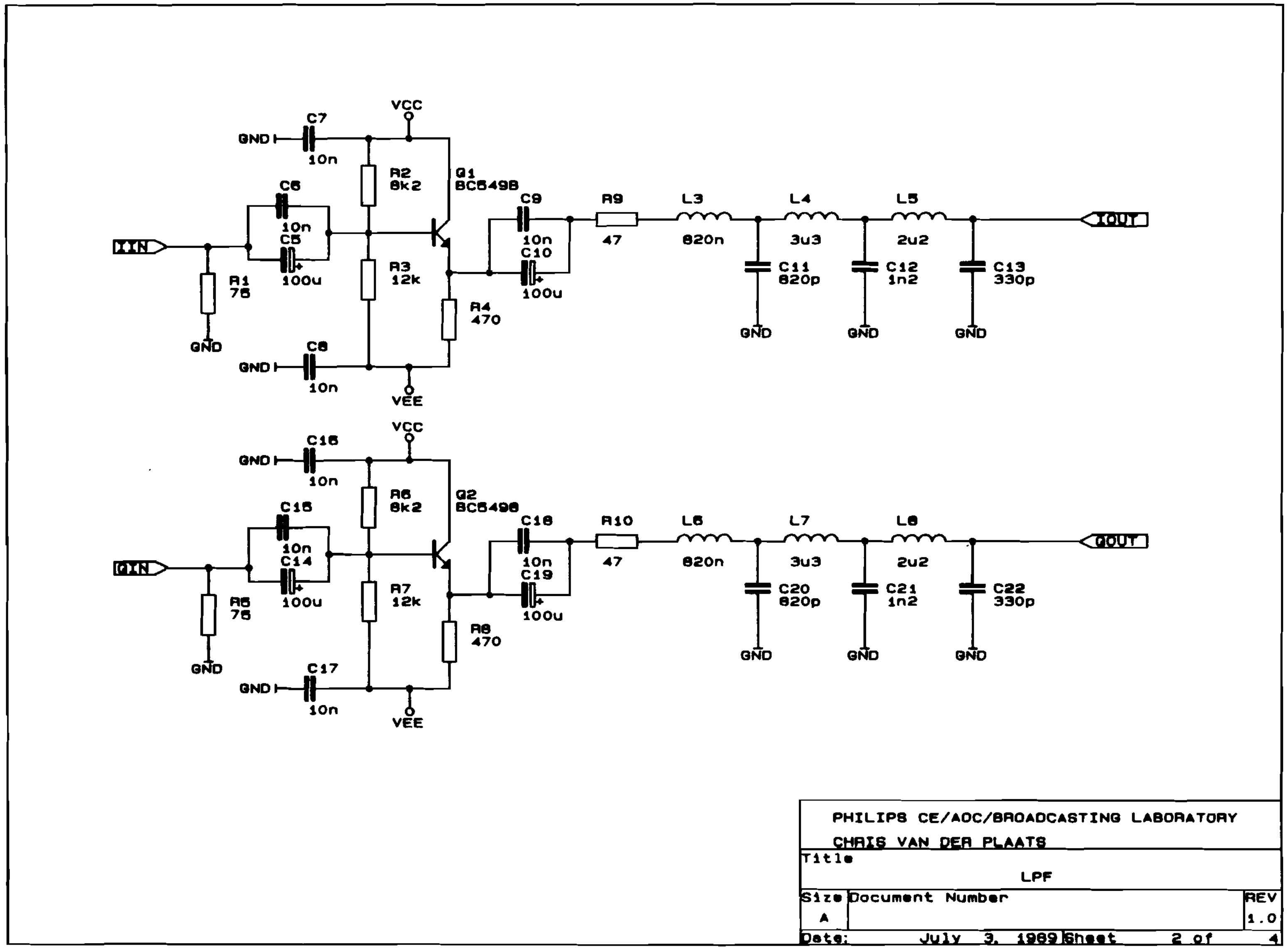


Sub-sheet "DAC".

B.6. Schematic of the "TRANSMITTER IF BOARD"

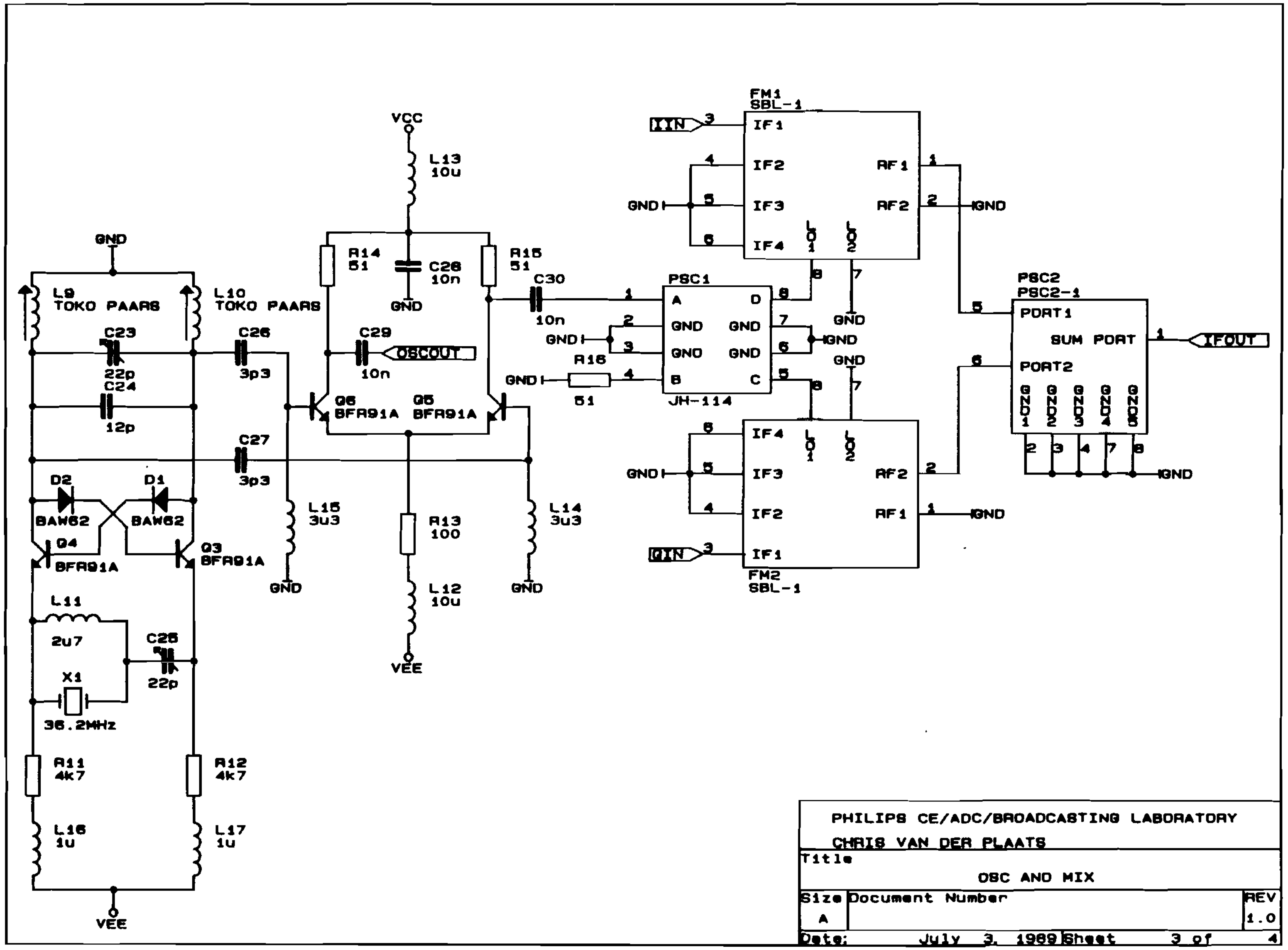


This schematic has three sub-sheets: "LPF", "OSC AND MIX" and "BPF".

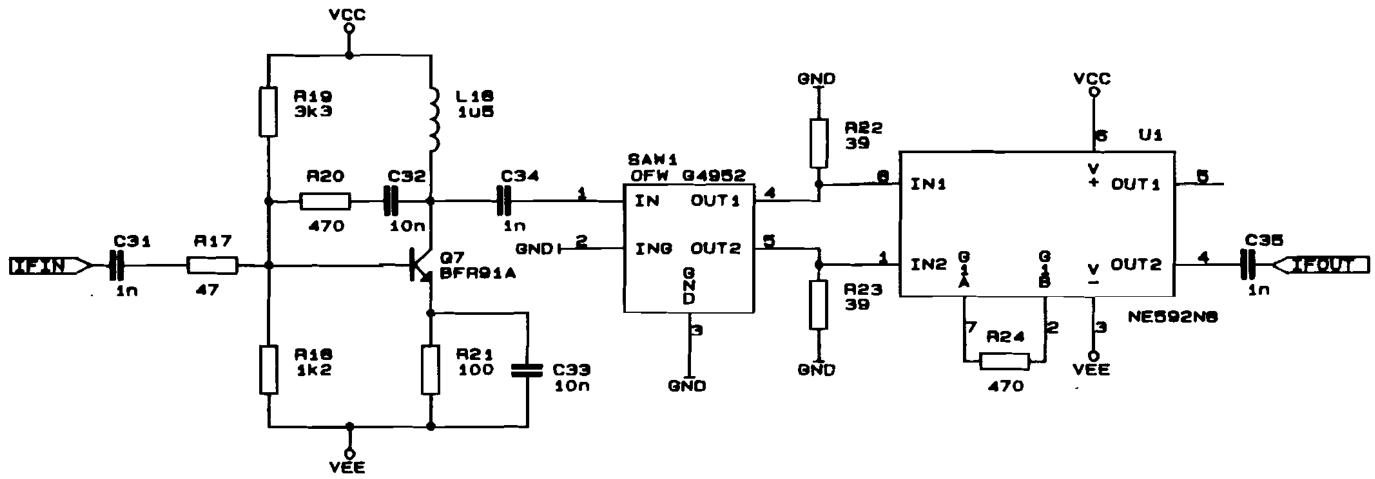


Sub-sheet "LPF".

Sub-sheet "OSC AND MIX".



Sub-sheet "BPF".



PHILIPS CE/ADC/BROADCASTING LABORATORY		
CHRIS VAN DER PLAATS		
Title: BPF		
Size: A	Document Number	REV: 1.0
Date: July 3, 1989	Sheet 4 of 4	

C. SOURCE CODE OF PROGRAM MODULATE

```
/* TURBO C source code of program MODULATE.EXE */
/* filename: MODULATE.C */

/* conditional compilation preprocessor commands */

#if !defined(__TURBOC__)
#error Wrong compiler. Use TURBOC!
#elif !((__TURBOC__ == 0x0150) || (__TURBOC__ == 0x018d))
#error Wrong compiler version. Use TURBOC 1.5 or TURBOC 2.0!
#elif defined __TINY__
#error Tiny memory model. Use compact, large or huge memory model!
#elif defined __SMALL__
#error Small memory model. Use compact, large or huge memory model!
#elif defined __MEDIUM__
#error Medium memory model. Use compact, large or huge memory model!
#elif defined(__STDC__)
#error ANSI keywords only option is on. Switch it to off!
#elif defined(__PASCAL__)
#error Calling convention is switched to Pascal. Switch it to C!
#endif

/* include following library header files */

#include<math.h>
#include<time.h>
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<dos.h>

/* constant definitions */

#define RAM         30720  /* number  of  samples which can be stored in
                              the transmitter RAM */
#define ZERO        128    /* zero level of the used D/A converters */
#define PPI_PORT_A  0x0300 /* PPI 8255 port A I/O address */
```

```
#define PPI_PORT_B  0x0301 /* PPI 8255 port B I/O address */
#define PPI_PORT_C  0x0302 /* PPI 8255 port C I/O address */
#define PPI_CONTROL 0x0303 /* PPI 8255 control port I/O address */

/* type definitions */

typedef unsigned char byte;
typedef unsigned int word;
typedef struct {byte m_power; byte i_power;} sysdef;

/* function prototypes */

void  make_sin_table       (word M, float *sin_table);
void  make_sinc_table      (word M, float *sinc_table);
void  make_save_mod_angles (word C, word F, byte *mod_angles,
                            char *filename);
void  shape_spectrum       (word symbol, word C, word F, word I,
                            word M, word N, byte *mod_angles,
                            float *re_spectrum, float *im_spectrum,
                            float *sinc_table);
void  ifft                 (word M, float *(*re_spectrum),
                            float *(*im_spectrum), float *(*re_signal),
                            float *(*im_signal), float *sin_table);
void  modulate             (word M, float *re_signal, float *im_signal);
float clip                 (float clipvalue, float sample);
void  set_transmitter      (void);
void  transmit             (byte *re_ram, byte *im_ram);

/* main function */

void main()
{

   /* variables and pointers */

   byte   *re_ram;       /* pointer to the array in which the 8-bit
                            quantized and clipped I-channel samples of
                            the frame are stored */
```

```
byte   *im_ram;         /* pointer  to  the  array  in  which the 8-bit
                           quantized and clipped Q-channel  samples  of
                           the frame are stored */
byte   *mod_angles;     /* pointer to the array in which the modulation
                           angles  of  all  emitted carriers and of all
                           frame signalling intervals are stored */
byte   key;             /* key number variable */
word   C;               /* number of emitted carriers */
word   F;               /* number of modulated symbols per frame */
word   G;               /* number of samples in guardband interval */
word   I;               /* oversampling ratio */
word   M;               /* number of samples in useful signal */
word   N;               /* number of emitted + virtual carriers */
word   sysnum;          /* system number variable */
word   symbol;          /* symbol number variable */
word   sample;          /* sample number variable */
float  *re_spectrum;    /* pointer  to the array in which the real part
                           of the DFT of the signal is stored */
float  *im_spectrum;    /* pointer  to the array in which the imaginary
                           part of the DFT of the signal is stored */
float  *re_signal;      /* pointer  to the array in which the I-channel
                           samples of the signal are stored */
float  *im_signal;      /* pointer  to the array in which the Q-channel
                           samples of the signal are stored */
float  *sin_table;      /* pointer  to the array in which sin(x) values
                           are  stored  used  by   function   ifft   to
                           calculate a M-point IDFT */
float  *sinc_table;     /* pointer to the array in  which  x  /  sin(x)
                           values   are   stored   used   by   function
                           shape_spectrum  to  compensate  for  sample
                           and hold circuit in D/A converter */
sysdef system[] = {{13, 0}, { 9, 0}, {10, 1}, {11, 2}};
                        /* array    with   systemdefinitions   of   the
                           four  used  systems;  a  system  is  totally
                           defined by log2(M) and log2(I) */

/* give information on systems on screen and ask for choice */
```

```
clrscr ();
cprintf ("OFDM/QPSK MODULATOR (with guardband interval)\n\n\r");
cprintf ("Which system do you want?\n\n\r");
cprintf ("system selection key ................  1       "
         " 2     . 3         4\n\n\r");
cprintf ("system name .........................  system 1"
         " system 2  system 3  system 4\n\r");
cprintf ("#carriers (emitted + virtual).........  8192    "
         " 512       512       512\n\r");
cprintf ("#emitted carriers ...................  7168    "
         " 448       224       112\n\r");
cprintf ("carrier spacing (Hz) ................  976.5625"
         " 15625     7812.5    3906.25\n\r");
cprintf ("minimal modulated carrier spacing (Hz)  976.5625"
         " 15625     15625     15625\n\r");
cprintf ("sampling frequency (MHz) ............  8       "
         " 8         8         8\n\r");
cprintf ("oversampling ratio ..................  1       "
         " 1         2         4\n\r");
cprintf ("nominal bandwidth (MHz) .............  7       "
         " 7         3.5       1.75\n\r");
cprintf ("bitrate for used frame (Mbit/s) ......  3.733334"
         " 10.73334  2.566667  0.583334\n\r");
cprintf ("bitrate for infinite frame (Mbit/s) ..  11.2    "
         " 11.2      2.8       0.7\n\r");
cprintf ("useful signal duration (microsec) ....  1024    "
         " 64        128       256\n\r");
cprintf ("guardband interval (microsec) ........  256     "
         " 16        32        64\n\r");
cprintf ("total signal duration (microsec) .....  1280    "
         " 80        160       320\n\r");
cprintf ("#modulated symbols per frame .........  3       "
         " 48        24        12\n\n\r");
do
{
   cprintf ("\rType system selection key to enter your choice: ");
   key = getche ();
   sysnum = 0;
```

```
   if (key == 50) sysnum = 1;
   if (key == 51) sysnum = 2;
   if (key == 52) sysnum = 3;
}
while ((key != 49) && (key != 50) && (key != 51) && (key != 52));

      /* ask if all modulations should be suppressed; this option  is
   not possible for system 1 due to the RAM length */

if (sysnum != 0)
{
   clrscr ();
   cprintf ("OFDM/QPSK MODULATOR (with guardband interval)\n\n\r");
   do
   {
      cprintf ("\rSuppress all phase and amplitude modulations"
               " (Y/N)? ");
      key = getche ();
   }
   while ((key != 'Y') && (key != 'y') &&
          (key != 'N') && (key != 'n'));
}
else key = 'N';

/* calculate parameters for the chosen system and check  for  illegal
   system definition */

clrscr ();
cprintf ("OFDM/QPSK MODULATOR (with guardband interval)\n\n\r");
cprintf ("SYSTEM %d ", sysnum + 1);
if ((key == 'y') || (key == 'Y')) cprintf ("- UNMODULATED CARRIERS");
cprintf ("\n\n\r");
if ((system[sysnum].m_power >= 4) && (system[sysnum].m_power <= 13))
{
   M = 1 << system[sysnum].m_power;
   G = M >> 2;
   F = RAM / (M + G);
}
```

```
else
{
   cprintf ("Illegal value in system[%d].m_power!",sysnum);
   exit (1);
}
if ((system[sysnum].i_power >= 0) &&
    (system[sysnum].i_power <= system[sysnum].m_power - 4))
{
   I = 1 << system[sysnum].i_power;
   N = 1 << (system[sysnum].m_power - system[sysnum].i_power);
   C = (N - (N >> 3)) / I;
}
else
{
   cprintf ("Illegal value in system[%d].i_power!",sysnum);
   exit (1);
}

/* allocate memory for used arrays and check for memory overflow */

re_ram      = (byte *)  calloc (RAM        , sizeof(byte));
im_ram      = (byte *)  calloc (RAM        , sizeof(byte));
mod_angles  = (byte *)  calloc (C * (F - 2), sizeof(byte));
re_spectrum = (float *) calloc (M          , sizeof(float));
im_spectrum = (float *) calloc (M          , sizeof(float));
re_signal   = (float *) calloc (M          , sizeof(float));
im_signal   = (float *) calloc (M          , sizeof(float));
sin_table   = (float *) calloc (M          , sizeof(float));
sinc_table  = (float *) calloc (M          , sizeof(float));
if ((re_ram      == NULL) || (im_ram      == NULL) ||
    (mod_angles  == NULL) || (re_spectrum == NULL) ||
    (im_spectrum == NULL) || (re_signal   == NULL) ||
    (im_signal   == NULL) || (sin_table   == NULL) ||
    (sinc_table  == NULL))
{
   cprintf("Out of memory error in function main !\n\n\r");
   exit(1);
}
```

```
/* calculate  sine  table which is used by function ifft to calculate
   a M-point IDFT */

cprintf("Calculating sine table\n\n\r");
make_sin_table (M, sin_table);

/* calculate  sinc  table which is used by function shape_spectrum to
   compensate for the sample and hold cicuit  in  the  D/A  converter
   in the transmitter */

cprintf("Calculating sinc table\n\n\r");
make_sinc_table (M, sinc_table);

/* test for modulation suppression */

if ((key == 'N') || (key == 'n'))
{
   /* get  C  * (F - 2) random modulation angles and save them in the
      file "ANGLES.DAT" so that they can be  used  to  calculate  bit
      error  rates  when  the  received  phase  angles are known; the
      random angles are denoted by 0, 1, 2 or 3 indicating  0,  pi/2,
      pi  and 3*pi/2 and they are saved sequentially from low carrier
      number to high carrier number and from  low  symbol  number  to
      high symbol number */

   cprintf("Making and saving modulation angles in file"
           " \"ANGLES.DAT\"\n\n\r");
   make_save_mod_angles (C, F, mod_angles, "angles.dat");

   /* make frame consisting of one reference symbol F - 2 random data
      symbols and one null symbol */

   cprintf("Processing frame of %u symbols\n\n\r", F);
   for (symbol = 0; symbol < F; symbol++)
   {
      cprintf("\rProcessing symbol number %2u ", symbol);

      /* shape the DFT of the next signal in the frame */
```

```
        shape_spectrum(symbol, C, F, I, M, N, mod_angles,
                       re_spectrum, im_spectrum, sinc_table);

        /* calculate signal of symbol by taking the M-point IDFT of the
           shaped discrete spectrum */

        ifft(M, &re_spectrum, &im_spectrum, &re_signal, &im_signal,
             sin_table);

        /* modulate signal with half the sample frequency */

        modulate(M, re_signal, im_signal);

        /* place 8 bit quantized and clipped guardband  signal  in  the
           frame;  the  standard  deviation of the signal is set to one
           fourth of the clipping level */

        for (sample = 0; sample < G; sample++)
        {
           re_ram[symbol * (M + G) + sample] =
              ZERO + clip (127, 32 * re_signal[sample + M - G]);
           im_ram[symbol * (M + G) + sample] =
              ZERO + clip (127, 32 * im_signal[sample + M - G]);
        }

        /* place 8 bit quantized and clipped signal in the  frame;  the
           standard deviation of the signal is set to one fourth of the
           clipping level */

        for (sample = G; sample < M + G; sample++)
        {
           re_ram[symbol * (M + G) + sample] =
              ZERO + clip (127, 32 * re_signal[sample - G]);
           im_ram[symbol * (M + G) + sample] =
              ZERO + clip (127, 32 * im_signal[sample - G]);
        }
     }
  }
```

```
else
{

   /* write "UNMODULATED CARRIERS" to file angles.dat */

   FILE *fp;
   fp = fopen ("angles.dat", "w");
   fprintf(fp, "UNMODULATED CARRIERS");
   fclose(fp);

   cprintf ("Processing unmodulated carriers ");

   /* shape DFT of modulated reference symbol */

   shape_spectrum (0, C, F, I, M, N, mod_angles, re_spectrum,
                   im_spectrum, sinc_table);

   /* calculate  signal  of  symbol by taking the M-point IDFT of the
      shaped discrete spectrum */

   ifft(M, &re_spectrum, &im_spectrum, &re_signal, &im_signal,
        sin_table);

   /* modulate signal with half the sample frequency */

   modulate(M, re_signal, im_signal);

   for (symbol = 0; symbol < RAM / M; symbol++)
      for (sample = 0; sample < M; sample++)
      {
         re_ram[symbol * M + sample] =
            ZERO + clip (127, 32 * re_signal[sample]);
         im_ram[symbol * M + sample] =
            ZERO + clip (127, 32 * im_signal[sample]);
      }
}

/* initialize transmitter */
```

```
   cprintf("\n\n\rInitializing transmitter\n\n\r");
   set_transmitter();

   /* write signals to transmitter RAM */

   cprintf("Writing samples of signals to transmitter\n\n\r");
   transmit(re_ram, im_ram);

   /* free allocated memory */

   free(sin_table);
   free(sinc_table);
   free(re_spectrum);
   free(im_spectrum);
   free(re_signal);
   free(im_signal);
   free(re_ram);
   free(im_ram);
   free(mod_angles);

   /* end of program */

   cprintf("Ready\n\n\r");
}

/* functions */

void make_sin_table (word M, float *sin_table)

/* the function make_sin_table fills the array sin_table with the M
   values sin (2 * pi * i / M) with 0 < i < M, i = 0, 1, 2, 3, ... */

{
   word i;
   float x;
   x = 2 * M_PI / M;
   for (i = 0; i < M; i++) sin_table[i] = sin (i * x);
}
```

```
void make_sinc_table (word M, float *sinc_table)

/* the function make_sinc_table fills the array sinc_table  with  the  M
   values x / sin(x) with x = pi * (M / 2 - i) / M, 0 < i < M, i = 0, 1,
   2, 3, ... */

{
   word i;
   float x;
   for(i = 1; i < (M >> 1); i++)
   {
      x = M_PI * ((M >> 1) - i) / M;
      sinc_table[i] = x / sin (x);
      sinc_table[M - i] = sinc_table[i];
   }
   sinc_table[0] = M_PI_2 / sin (M_PI_2);
   sinc_table[M >> 1] = 1;
}

void make_save_mod_angles (word C, word F, byte *mod_angles,
                           char *filename)

/* the function make_save_mod_angles generates a sequence of C * (F - 2)
   numbers randomly chosen from the set {0, 1, 2, 3}; these numbers  are
   stored  in the array mod_angles and in the file filename; the numbers
   are used by the function shape_spectrum to generate random modulation
   angles equal to number * pi/2 */

{
   word i;
   FILE *fp;
   randomize ();
   fp = fopen (filename, "w");
   for (i = 0; i < C * (F - 2); i++)
   {
      mod_angles[i] = random(4);
      fprintf(fp, "%u ", mod_angles[i]);
   }
```

```
   fclose(fp);
}

void shape_spectrum(word symbol, word C, word F, word I, word M,
                    word N, byte *mod_angles, float *re_spectrum,
                    float *im_spectrum, float *sinc_table)

/* the function shape_spectrum assembles the DFT of a signal; the real
   part of the DFT is stored in the array re_spectrum and the imaginary
   part of the DFT is stored in the array im_spectrum; if symbol = 0 it
   generates the DFT of the signal belonging to the reference symbol
   (all modulation angles equal to zero); if symbol = F - 1 it generates
   the DFT of the signal belonging to the null symbol; for the other
   values of symbol it generates the DFT of the signal belonging to a
   random symbol using the random numbers in the array mod_angles; the
   random numbers from the set {0, 1, 2, 3} are multiplied with pi/2 to
   obtain the modulation angle; for every symbol C carriers are
   modulated and emitted, the other carriers are set to zero; if k is
   the carrier number, the carrier is emited for N/16 <= k < N/2 and for
   N/2 < k <= N - N/16; the zero phase of each emitted carrier is set to
   pi*k*k/N; the standard deviation of the OFDM signal after IFFT
   processing is equal to a constant multiplied by the square root of
   the number of emitted carriers C, therefore the DFT is multiplied
   with 1/sqrt(C) in order to get signals with equal standard deviation
   for each system; furthermore the amplitude of each carrier is
   multiplied with a x/sin(x) factor from the sinc_table array to
   compensate for the zero-order hold circuit in the DAC */

{
   word i, k, counter;
   double scale, angle;
   scale = sqrt(2) / sqrt(C);
   for (i = 0; i < M; i++)
   {
      re_spectrum[i] = 0;
      im_spectrum[i] = 0;
   }
```

```
/* reference symbol */

if (symbol == 0)
{
   counter = 0;
   for (i = (M >> 1) - (N >> 1) + (N >> 4); i < (M >> 1); i += I)
   {
      k = i - (M >> 1) + (N >> 1);
      angle = M_PI * k * k / N;
      re_spectrum[i] = scale * sinc_table[i] * cos (angle);
      im_spectrum[i] = scale * sinc_table[i] * sin (angle);
      counter++;
   }
   for (i = (M >> 1) + I; i <= (M >> 1) + (N >> 1) - (N >> 4);
        i += I)
   {
      k = i - (M >> 1) + (N >> 1);
      angle = M_PI * k * k / N;
      re_spectrum[i] = scale * sinc_table[i] * cos (angle);
      im_spectrum[i] = scale * sinc_table[i] * sin (angle);
      counter++;
   }
}

/* random data symbols */

if ((symbol > 0) && (symbol < F - 1))
{
   counter = 0;
   for (i = (M >> 1) - (N >> 1) + (N >> 4); i < (M >> 1); i += I)
   {
      k = i - (M >> 1) + (N >> 1);
      angle = M_PI * k * k / N +
              M_PI_2 * mod_angles[(symbol - 1) * C + counter];
      re_spectrum[i] = scale * sinc_table[i] * cos (angle);
      im_spectrum[i] = scale * sinc_table[i] * sin (angle);
      counter++;
   }
```

```
    for (i = (M >> 1) + I; i <= (M >> 1) + (N >> 1) - (N >> 4);
         i += I)
    {
       k = i - (M >> 1) + (N >> 1);
       angle = M_PI * k * k / N +
               M_PI_2 * mod_angles[(symbol - 1) * C + counter];
       re_spectrum[i] = scale * sinc_table[i] * cos (angle);
       im_spectrum[i] = scale * sinc_table[i] * sin (angle);
       counter++;
    }
  }
}
```

**void ifft (word M, float *(*re_spectrum), float *(*im_spectrum),
float *(*re_signal), float *(*im_signal), float *sin_table)**

```
/* the  function ifft caculates the M complex point IDFT of spectrum and
   multiplies it with the factor M; the result is stored in signal;  the
   real  part  of  spectrum  is  given  in the array re_spectrum and the
   imaginary  part  is  given  in  the  array  im_spectrum;  the  arrays
   re_signal and im_signal form the real and imaginary part of signal; M
   must be a power of 2 for this function; the array sin_table  is  used
   as a sine lookup table */

{
   word i, j, k, l;
   float *buffer_ptr;

   /* bit reversal */

   for (i = 0; i < M; i++)
   {
      for (k = M >> 1, j = 0, l = i; k > 0; k >>= 1)
      {
         j = (l & 1) | (j << 1);
         l >>= 1;
      }
      (*re_signal)[i] = (*re_spectrum)[j];
```

```
      (*im_signal)[i] = (*im_spectrum)[j];
   }

   /* butterfly calculation */

   for (i = M >> 1; i > 0; i >>= 1)
   {
      for (j = 0, k = 0, l = 0; j < M;
           j++, k = (k + 2) % M, l = (j - (j % i)))
      {
         (*re_spectrum)[j] = (*re_signal)[k] +
            (*re_signal)[k + 1] * sin_table[(l + (M >> 2)) % M] -
            (*im_signal)[k + 1] * sin_table[l];
         (*im_spectrum)[j] = (*im_signal)[k] +
            (*re_signal)[k + 1] * sin_table[l] +
            (*im_signal)[k + 1] * sin_table[(l + (M >> 2)) % M];
      }
      buffer_ptr = *re_signal;
      *re_signal = *re_spectrum;
      *re_spectrum = buffer_ptr;
      buffer_ptr = *im_signal;
      *im_signal = *im_spectrum;
      *im_spectrum = buffer_ptr;
   }
}

void modulate(word M, float *re_signal, float *im_signal)
{
   word i;
   for (i = 1; i < M; i += 2)
   {
      re_signal[i] = -re_signal[i];
      im_signal[i] = -im_signal[i];
   }
}

float clip (float clipvalue, float sample)
```

```
/* the function clip returns the clipped float value of sample */

{
   if(sample > clipvalue) sample = clipvalue;
   if(sample < -clipvalue) sample = -clipvalue;
   return(sample);
}
```

void set_transmitter (void)

```
/* the function set_transmitter initializes and clears  the  transmitter
   and it checks for hardware errors */

{
   word i;

   /* program PPI 8255 to mode 0, port A output, port B  output  port  C
      upper output and port C lower input */

   outportb(PPI_CONTROL, 0x81);

   /* reset WRITE TRANSMITTER RAM REQUEST bit */

   outportb(PPI_CONTROL, 0x0e);

   /* check for connection */

   if (inportb(PPI_PORT_C) & 2)
   {
      cprintf("Hardware error: "
              "no connection between PC and transmitter!\n\n\r");
      exit(1);
   }

   /* set WRITE TRANSMITTER RAM REQUEST bit */

   outportb(PPI_CONTROL, 0x0f);
```

```
/* check for connection */

if (!(inportb(PPI_PORT_C) & 2))
{
   cprintf("Hardware error: "
           "no connection between PC and transmitter!\n\n\r");
   exit(1);
}

/* check transmitter +5V power */

if (!(inportb(PPI_PORT_C) & 1))
{
   cprintf("Hardware error: power failure in transmitter!\n\n\r");
   exit(1);
}

/* wait for acknowledge by checking WRITE TRANSMITTER RAM ACKNOWLEDGE
   bit */

while(!(inportb(PPI_PORT_C) & 8));

/* reset WRITE TRANSMITTER RAM REQUEST  bit */

outportb(PPI_CONTROL, 0x0e);

/* bring transmitter to transmitting state */

while(inportb(PPI_PORT_C) & 8)
{

   /* reset  and  set  WRITE  TRANSMITTER  RAM  STROBE bit to advance
      address counter */

   outportb(PPI_CONTROL, 0x0a);
   outportb(PPI_CONTROL, 0x0b);
}
```

```
/* request for writing transmitter RAM by setting the WRITE
   TRANSMITTER RAM REQUEST bit */

outportb(PPI_CONTROL, 0x0f);

/* wait for acknowledge by checking WRITE TRANSMITTER RAM ACKNOWLEDGE
   bit */

while (!(inportb(PPI_PORT_C) & 8));

/* reset WRITE TRANSMITTER RAM REQUEST bit */

outportb(PPI_CONTROL, 0x0e);

/* clear transmitter memory by writing ZERO value to all memory
   locations */

for (i = 0; inportb(PPI_PORT_C) & 8; i++)
{

   /* write ZERO to I-channel RAM location */

   outportb(PPI_PORT_A, ZERO);

   /* write ZERO to Q-channel RAM location */

   outportb(PPI_PORT_B, ZERO);

   /* reset and set WRITE TRANSMITTER RAM STROBE bit to advance
      address counter */

   outportb(PPI_CONTROL, 0x0a);
   outportb(PPI_CONTROL, 0x0b);
}

/* check for bad strobe signal */

if (i != RAM)
```

```
    {
       cprintf("Hardware error: bad strobe signal!\n\n\r");
       exit(1);
    }
}

void transmit (byte *re_ram, byte *im_ram)

/* the  function  transmit  writes the array re_ram to the I-channel ram
   and the array im_ram to the Q-channel ram of the transmitter */

{
   word i;

   /* request   for   writing  transmitter  RAM  by  setting  the  WRITE
      TRANSMITTER RAM REQUEST bit */

   outportb(PPI_CONTROL, 0x0f);

   /* wait for acknowledge by checking WRITE TRANSMITTER RAM ACKNOWLEDGE
      bit */

   while (!(inportb(PPI_PORT_C) & 8));

   /* reset WRITE TRANSMITTER RAM REQUEST bit */

   outportb(PPI_CONTROL, 0x0e);

   /* write frame signal samples to transmitter memory */

   for (i = 0; inportb(PPI_PORT_C) & 8; i++)
   {

       /* write next I-channel sample */

       outportb(PPI_PORT_A, re_ram[i]);

       /* write next Q-channle sample */
```

```
        outportb(PPI_PORT_B, im_ram[i]);

        /* reset and set WRITE TRANSMITTER RAM STROBE bit to advance
           address counter */

        outportb(PPI_CONTROL, 0x0a);
        outportb(PPI_CONTROL, 0x0b);
    }

    /* check for bad strobe signal */

    if (i != RAM)
    {
       cprintf("Hardware error: bad strobe signal!\n\n\r");
       exit(1);
    }
}
```