PART IV

HARDWARE IMPLEMENTATION

CHAPTER NINE

IMPLEMENTATION OF THE DSSS TRANSMITTER

9.1 INTRODUCTION

The success by which the theoretical analysis and simulation of the DSSS transmitter have been performed and the promising results obtained in Chapter 7, motivated the implementation of the transmitter in hardware. The same approach that has been followed with the theoretical analysis and simulation, was applied in the implementation of the hardware DSSS transmitter. This chapter gives the design approach and hardware transmitter structures used, as well as results obtained with this transmitter module. The transmitter module was the first prototype version built to illustrate the practical implementation of complex spreading sequences. The results obtained with the hardware transmitter module correlated very well with the theory and simulations. The system was also upgraded to a more advanced version using FPGA technology and will also be described in detail in this chapter. In Chapter 10 the final FPGA-based complex DSSS modem is presented, which is a great improvement in terms of technology relative to the structures described in this chapter.

9.2 HARDWARE DESCRIPTION

In contrast with DSSS communication systems using binary spreading sequences [6, 40], the spreading sequences employed in the DSSS systems described in this dissertation are complex, resulting in an increased implementation complexity when discrete logic components and memory devices are utilised. Memory devices are needed to implement the complex spreading sequences (CSS), by storing sampled pre-filtered replicas of the complex spreading sequences. The generation of complex spreading sequences is subsequently accomplished by reading the sequences from memory in real time. In contrast, in a conventional binary DSSS system, the binary Gold or Kasami spreading sequences are generated using simple shift register devices with feedback. It is nevertheless shown in Chapter 10 that the complexity problem experienced with CSS can be largely overcome by using reprogrammable FPGA technology.

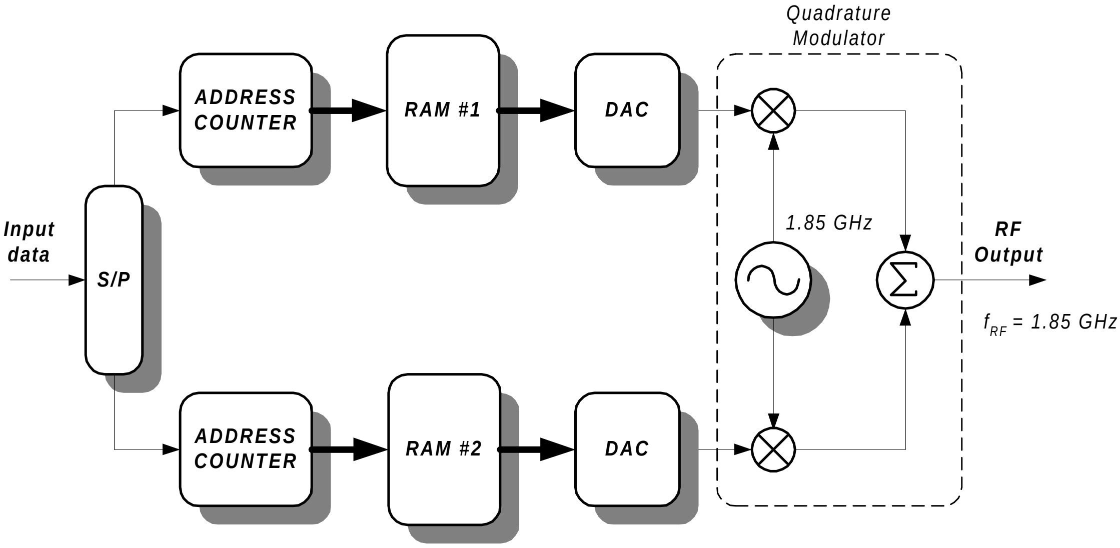


FIGURE 9.1: Block diagram of the hardware implementation of the complex DSSS transmitter

The transmitter, illustrated in Figure 9.1, consists of a serial-to-parallel converter producing two parallel data streams at the symbol rate (half the bit rate) for the case of the dual channel QPSK configuration and two identical data streams at the bit rate for the balanced QPSK configuration. These data streams are inputs to two address counters and are used as the most significant bits in the addresses supplied to two blocks of SRAMs. The data bits are used to select between different RAM blocks for the representation of ”$1s$” and ”$0s$”, where different replicas of the spreading sequences are stored. Thus, different replicas of the spreading sequences are read out of the RAMs, representing ”$1s$” and ”$0s$”, respectively, corresponding to the occurrence of ”$1s$” and ”$0s$” on the two branches, to accomplish the direct sequence spreading process. The digital outputs of the RAMs are converted to analog signals by means of digital-to-analog converters (DACs) to form the baseband in-phase and quadrature-phase branches. These baseband signals are fed into a quadrature modulator to be modulated onto cosine and sine RF carriers (at typically 1.85

GHz), whereafter they are combined with a zero-degree combiner to produce the final output signal of the complex DSSS transmitter. Figure 9.2 shows the complex DSSS transmitter hardware implementation.

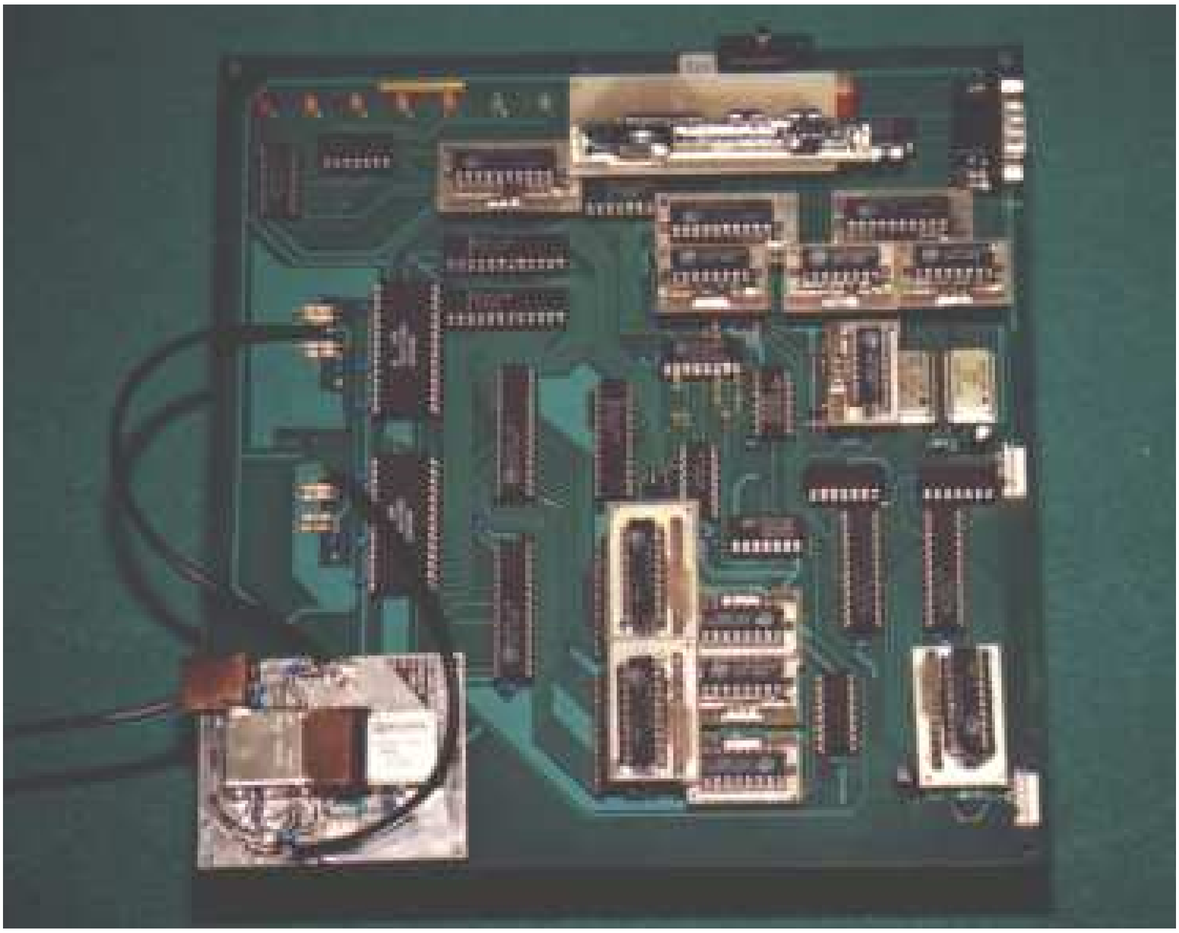


FIGURE 9.2: Complex DSSS transmitter hardware (First hardware version).

System control is achieved by means of a computer (PC) connected to a micro-controller (DS2250T). The board setup is done in software on the PC and then down-loaded to the micro-controller. The complex spreading sequences, used in the spreading process, are also initially down-loaded from the PC to the SRAMs of the transmitter board under the control of the micro-controller. Figure 9.3 shows a typical hardware setup for down-loading the setup configuration and spreading sequences. The length of the spreading sequence is chosen to equal the length of one data bit in the balanced QPSK configuration, or equal to one symbol in the dual channel QPSK configuration. The sampling clock on the board is used to clock the address counters, SRAMs and DACs, while the sampling clock is divided by a factor $S = spc * L$ to produce the data symbol clock, where spc represents the number of samples per chip in the sampled system and L the spreading sequence length. This symbol clock

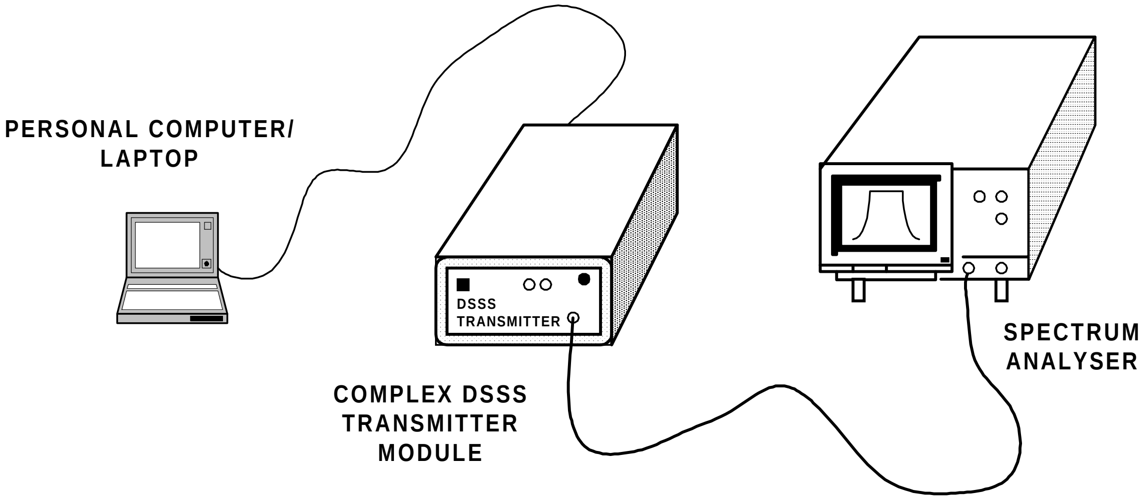


FIGURE 9.3: Hardware during down-loading of setup configuration and spreading sequences (First hardware version).

is used to clock the data latch in the input stage. It is also necessary to produce a double frequency clock to clock the data into the serial-to-parallel (S/P) converter in the case of the dual channel QPSK configuration, whereafter the clock is divided by two to clock the parallel I and Q symbol streams after S/P conversion of the input data. The complex DSSS transmitter unit, used in this setup, can be seen in Figure 9.4.

Using analytical complex spreading sequences, generated as described in Chapter 3, a SSB output signal is obtained, resulting in a 50% saving in bandwidth compared to DSB binary phase shift keying (BPSK). Since the latter two systems support identical data throughput, a 3dB processing gain (PG) advantage is obtained with the SSB system relative to the BPSK system. The 3dB PG can either be achieved by doubling the data rate of the SSB relative to the BPSK system, or by using double length spreading sequences for the same data rate and identical bandwidths. However, compared to the DSB dual channel (QPSK) transmitter configuration, the balanced quadrature configuration used to obtain the SSB result leads to a reduction of data throughput rate by a factor of two. The 50% bandwidth advantage of the balanced SSB system is thus neutralised by the factor of two reduction in data throughput rate relative to the dual channel (QPSK) system, resulting in 0dB PG advantage overall, i.e., the two systems potentially have identical performance. Any differences in performance will therefore be due to differences in cross-correlation properties between the two families of spreading sequences employed in each case, namely the analytically bandlimited complex (ABC) sequences used in the SSB case, and the generalised chirp-like (GCL) sequences used in the DSB case (refer [25] for details). It should be finally noted

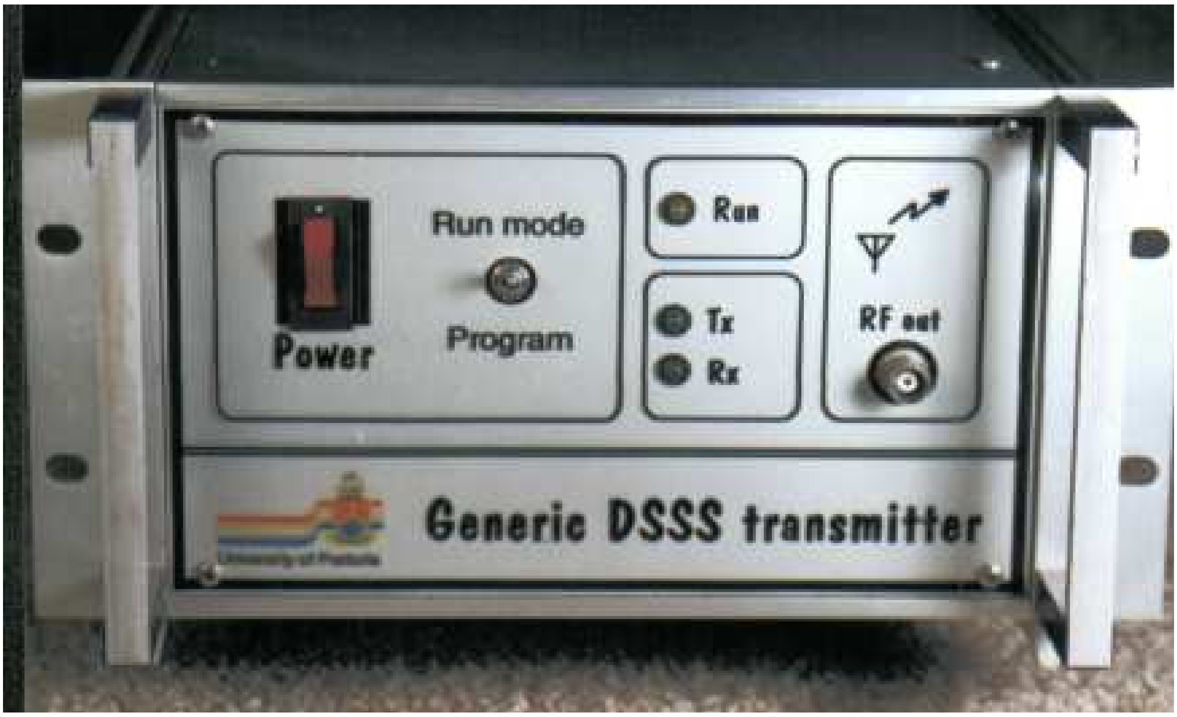


FIGURE 9.4: The complex DSSS transmitter unit (First hardware version).

that the output signals produced by all the aforementioned systems exhibit perfectly constant envelopes (CE), which is a unique result in the SSB case. The CE SSB feature is achieved through a unique combination of a balanced QPSK structure and the use of ABC CSS.

The transmitter not only facilitates the use of binary or complex spreading sequences, but the sequences can be pre-filtered before they are down-loaded to the transmitter board. This feature makes it possible to use any type of filtering which can be implemented in software. For example in the case of binary spreading sequences, filters with different roll-off factors, like the Nyquist filter, can be easily implemented. The latter is not always that simple to implement in practice, and the results are more ideal as would ever be the case for conventional lumped hardware systems. The results illustrating this concept can be seen in section 9.3 of this chapter.

The balanced QPSK and dual channel QPSK DSSS transmitter configurations can, with minor modifications, be extended to also generate $\pi/4$-QPSK, 8-PSK or 7x1-PSK modulation formats [21, 41–43]. The signal constellations for these modulation configurations are shown in Figure 9.5. When either $\pi/4$-QPSK, QPSK, 8-PSK or 7x1-PSK mode is selected, the incoming data is firstly serial-to-parallel converted. This is followed by combinational logic (precoder) to Gray and differentially encode the serial-to-parallel converted data into symbols which are then appropriately mapped onto the selected M-PSK signal constellation in the following way: The output of the precoder is used to control logic which generates the starting address of a pair of spreading sequences. These spreading sequences are weighted by scale factors corresponding with the X and Y -coordinates of the signal constellation point, onto which the symbol to be transmitted, is to be mapped. The mapping is accomplished by reading the selected pair of spreading sequences from the SRAMs under control of the micro-controller.

The real and imaginary parts of the spreading sequence can be pre-modulated onto cosine and sine carriers at an intermediate frequency, f_{IF}, (usually chosen to be equal to the chip frequency, f_{chip}), before they are stored in RAM. This is done by multiplying the four times over-sampled real and imaginary components of the complex spreading sequence by repetitive $(1, 0, -1, 0)$ and $(0, 1, 0, -1)$ sequences, respectively, constituting the cosine and sine modulation processes, [44–46]. This modulation method produces an IF output signal at a quarter of the sampling frequency. No unwanted spectral components are generated by means of this modulation method and thus less filtering is required.

Figure 9.6 shows version 2 of the DSSS transmitter module, which also includes parts of the receiver.

9.2.1 TRANSMITTER SPECIFICATIONS

In this section f_{sample} denotes the sampling frequency at which the system is clocked, f_{chip} the chip frequency, spc the number of samples per chip and L the length of the spreading sequence. For the chirp-like sequences the (DSB) transmission bandwidth on the carrier is

$$\begin{aligned} B_T &= \frac{f_{sample}}{spc} \\ &= f_{chip} \quad [Hz] \end{aligned} \tag{9.1}$$

It should be noted that the spectral shaping obtainable with the NLI-RU filtered GCL CSS approaches that of a Nyquist filter with roll-off factor $\alpha = 0$ for sufficiently long sequence lengths L. Since Nyquist filters with $\alpha = 0$ are hard to realise in practice, realistic Nyquist bandwidths equal that defined in Equation 9.1, multiplied by the factor $(1 + \alpha)$, giving the NLI-RU filtering process an α bandwidth advantage compared to conventional Nyquist applications.

The data rate, f_b, depends on the available spreading bandwidth B_T, the chosen family of spreading sequences and the spreading sequence length, L, within a specific family, the maximum spreading sequence length being limited to $L_{max} = 1024$ for this system.

$$f_b = \frac{f_{chip}}{L} \quad [bps] \tag{9.2}$$

The primary system specifications are summarized in Table 9.1.

PARAMETER	SPECIFICATION
Modulation Technique	Balanced/Dual channel QPSK DSSS
Spreading Sequences	Families of binary (Gold/Kasami) or complex (GCL)
Sequence Lengths	L (selectable up to 1023 - see Tables 9.2 and 9.3)
Data rate	f_b (selectable - see Tables 9.2 and 9.3)
Processing Gain	$PG = 10logL$ (see Tables 9.2 and 9.3)
Chip rate	$f_{chip} = 12.5Mchips/s$
Samples per chip	$spc = 4$
Sampling frequency	$f_{sample} = 50MSPS$
RF Carrier frequency	$f_{RF} = 1.85GHz$
Transmission bandwidth	$B_T = 12.5MHz(DSB) or B_T = 6.25MHz(SSB)$

TABLE 9.1: Complex DSSS transmitter specifications.

The typical data rates and processing gains achievable by the DSSS transmitter module is given in Tables 9.2 and 9.3 (other options are also available and can be obtained through simple software adjustments).

CODE LENGTH	PROCESSING GAIN [dB]	DATA RATE [kbps]
63	17.99	198.41
127	21.04	98.43
255	24.07	49.02
511	27.08	24.46
1023	30.10	12.22

Table 9.2: PGs and data rates obtainable with Gold and Kasami (binary) spreading sequences at a chip rate of $f_{chip} = 12.5Mchips/s$.

CODE LENGTH	PROCESSING GAIN [dB]	DATA RATE [kbps]
49	16.90	255.10
121	30.83	82.64
169	22.28	73.96
289	24.61	43.25
361	25.58	34.63
529	27.23	23.63
841	29.25	14.86
961	29.83	13.00

Table 9.3: PGs and data rates obtainable with General Chirp-Like (GCL) spreading sequences at a chip rate of $f_{chip} = 12.5Mchips/s$

9.3 PRACTICAL HARDWARE RESULTS

For comparative purposes, Figure 9.7 depicts the spectrum of the output of the DSSS transmitter when using Nyquist filtered binary (Gold) spreading sequences of length 511, modulated onto a 1.85 GHz RF carrier. Figure 9.8 shows the output spectrum of the DSSS transmitter, modulated onto a 1.85 GHz RF carrier, for the case where DSB root-of-unity filtered complex spreading sequences (RU-CSS) of length 529 were used for spreading. Figure 9.9 shows the spectrum of the output of the DSSS transmitter when non-linearly-interpolated root-of-unity filtered complex spreading sequences (NLI-RU-CSS, $N = 529$), [4], are used. Note that, the bandwidth of the output of the balanced SSB DSSS transmitter, using ABC NLI-RU-CSS, is only half of that produced by the dual channel DSB DSSS transmitter, using GCL NLI-RU-CSS. The data throughput rate of the former system is however two times lower than the latter system, resulting in identical processing gains, as explained in section 9.2.

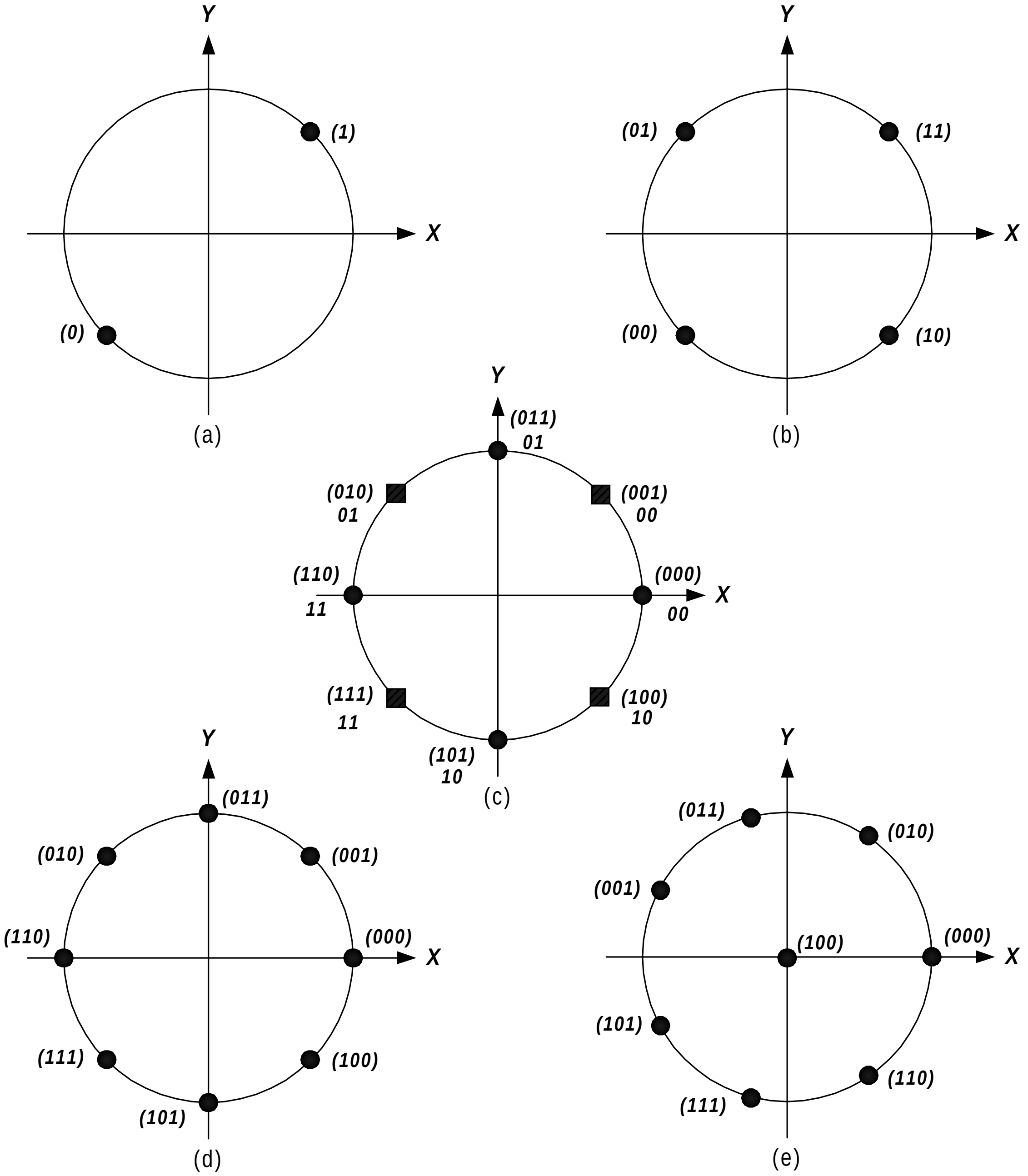


FIGURE 9.5: Signal constellations for the different modulation schemes: (a) Balanced QPSK, (b) Dual channel QPSK, (c) $\pi/4 - QPSK$, (d) 8-PSK and (e) 7x1-PSK.

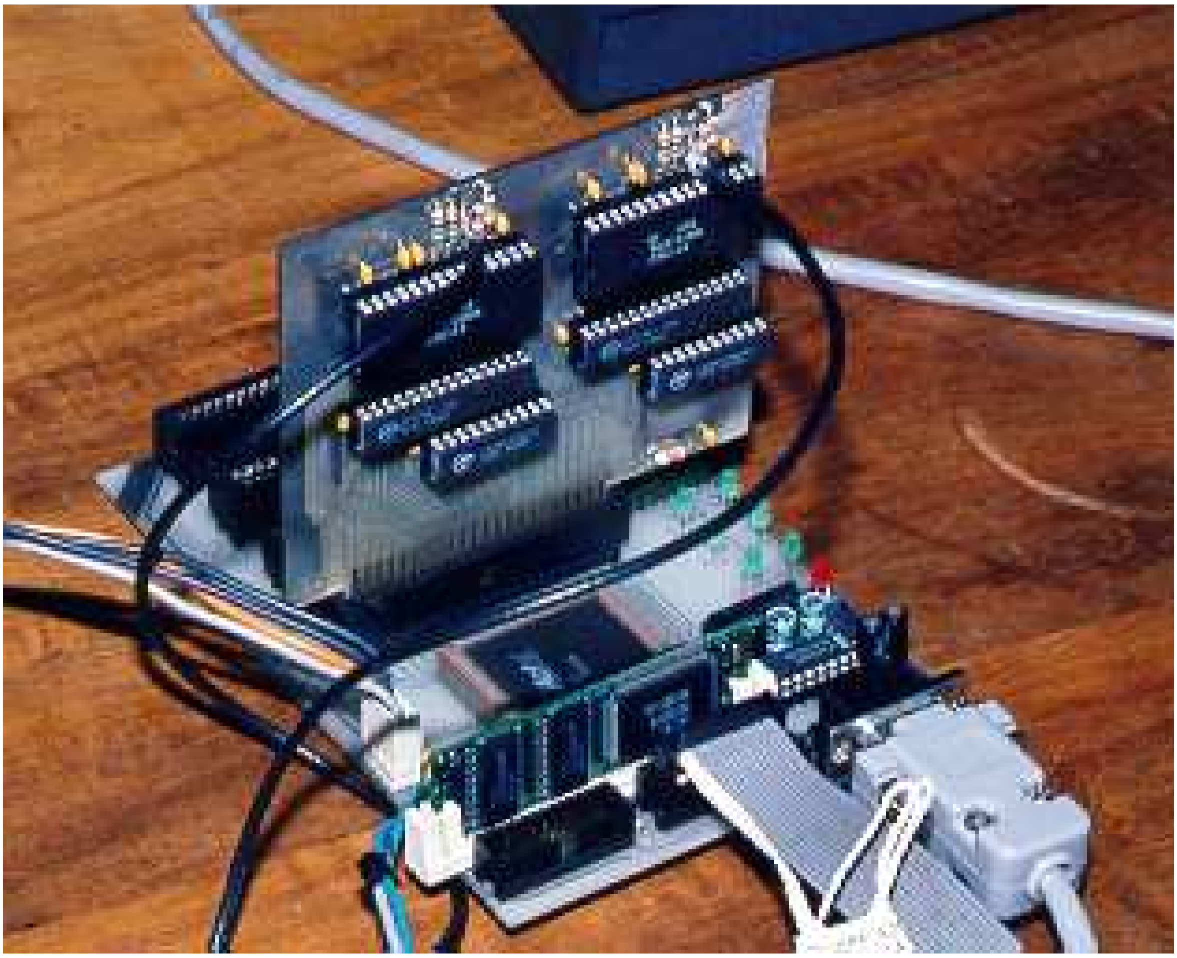


FIGURE 9.6: Second hardware version of the DSSS transmitter, including parts of the receiver.

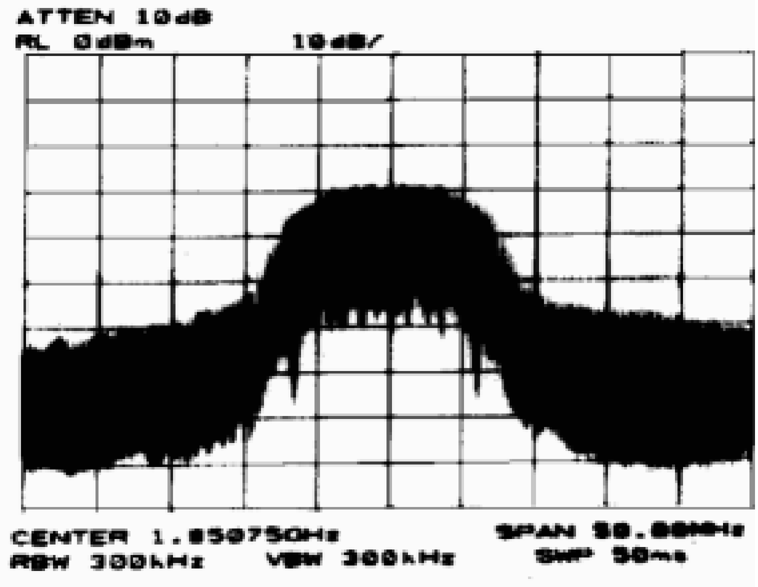


FIGURE 9.7: Spectrum of the output of the DSSS transmitter using binary (Gold) spreading sequences of length 511.

9.4 TYPICAL APPLICATIONS

As an example of the versatility and applicability of the proposed DSSS transmitter, it may be mentioned that the hardware prototype has been used in a channel sounding application to measure the delay-spread profile of various communication channels, including DECT and GSM (i.e., at different carrier frequencies). Other applications include various types of channel sounding, accurate distance and signal path delay measurements, radar applications and more.

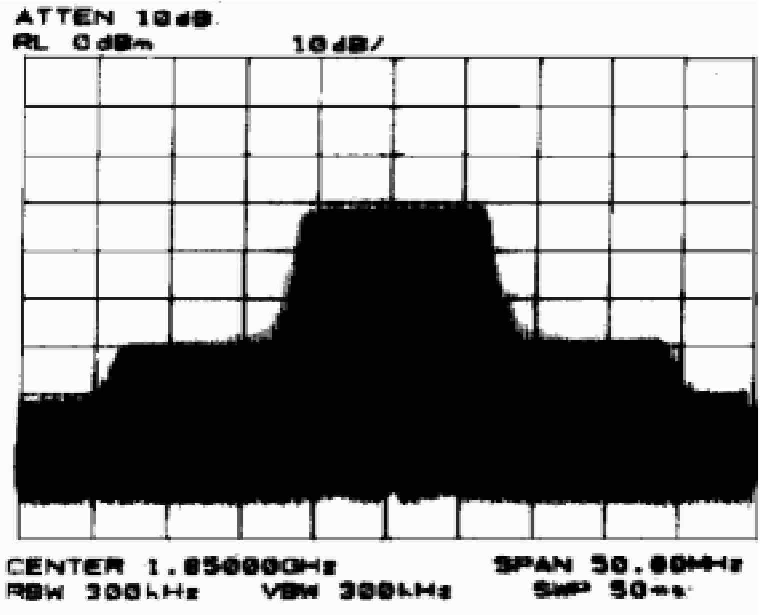


FIGURE 9.8: Spectrum of the output of the DSSS transmitter, modulated onto a $1.85\ GHz$ RF carrier, for the case where DSB root-of-unity filtered complex spreading sequences (RU-CSS) of length 529 were used.

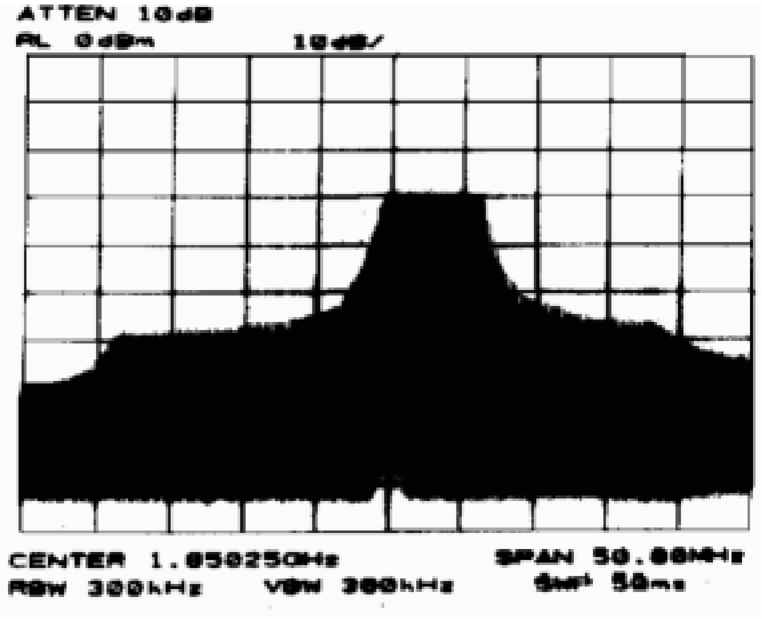
[htb] 

FIGURE 9.9: Spectrum of the output of the DSSS transmitter, modulated onto a $1.85\ GHz$ RF carrier, for the case where SSB non-linearly-interpolated root-of-unity filtered complex spreading sequences (NLI-RU-CSS) of length 529 were used.

CHAPTER TEN

IMPLEMENTATION OF THE GENERIC FPGA-BASED COMPLEX DSSS MODEM

10.1 INTRODUCTION

The results of the previous chapters serve to illustration the power and value of a simulation platform. The simulation results, presented in Chapter 8, is of inestimable value and will be used as a baseline in the implementation of the DSSS modem. The aforementioned theoretical structures and schemes were implemented in hardware by utilizing Altera's FPGA technology. In the initial design of the complex DSSS transmitter and receiver, most of the baseband processing was done by means of Altera's FPGA technology in VHDL. The rest of the system, and particularly the IF and RF subsystems, was realised with analog components. IF sampling principles were applied at the receiver and implemented on FPGA using VHDL programming software, resulting in a number of advantages compared to analog down-conversion and carrier tracking. The hardware results of the complex DSSS communication system are presented in this chapter. Complete system specifications are presented, as well as a brief description of the operation of the DSSS system.

10.2 HARDWARE DESCRIPTION

10.2.1 DSSS Modem Module

10.2.1.1 Hardware Design of DSSS Modem, Version One

The block diagram of the first hardware designed version of the complete DSSS modem is depicted in Figure 10.2. In this version of the DSSS modem, most of the IF, RF, carrier recovery, down-conversion, filtering, etc. were done by means of external hardware components, while the baseband processing is done inside the FPGA. On the transmit side the digital-to-analog converted DAC) baseband in-phase and quadrature data branches are modulated onto a $70\ MHz$ IF carrier frequency by means of a quadrature modulator. The IF is then up-converted to an RF frequency of $2.4415\ GHz$, which falls within one of the Industrial, Scientific and medical (ISM) frequency bands. On the receive side the received signal is first bandpass filtered and amplified by means of a low noise amplifier (LNA), whereafter it is again band pass filtered. The $2.4415\ GHz$ RF signal is down-converted to a $70\ MHz$ IF frequency, bandpass filtered and automatic gain controlled (AGC). This 70 MHz IF signal is quadrature down-converted to provide baseband in-phase and quadrature branches, which are analog-to-digital converted. A quadrature down-converter with a voltage controlled oscillator (VCO) and external tank circuit, was used for the final down-conversion stage as well as for part of the carrier recovery loop. The phase detector, etc. parts of the carrier recovery loop were done inside the baseband processing block in FPGA. A DAC is used for conversion of the carrier recovery error signal to an analog signal to control the external VCO in the carrier recovery loop. The data spreading and despreading are done inside the FPGA as part of the baseband processing. The data source can be a voice codec or any other external device that generates data to be transmitted, as illustrated in Figure 10.2. Hardware boards developed to implement the first few versions of the DSSS communication system are shown in Figure 10.1.

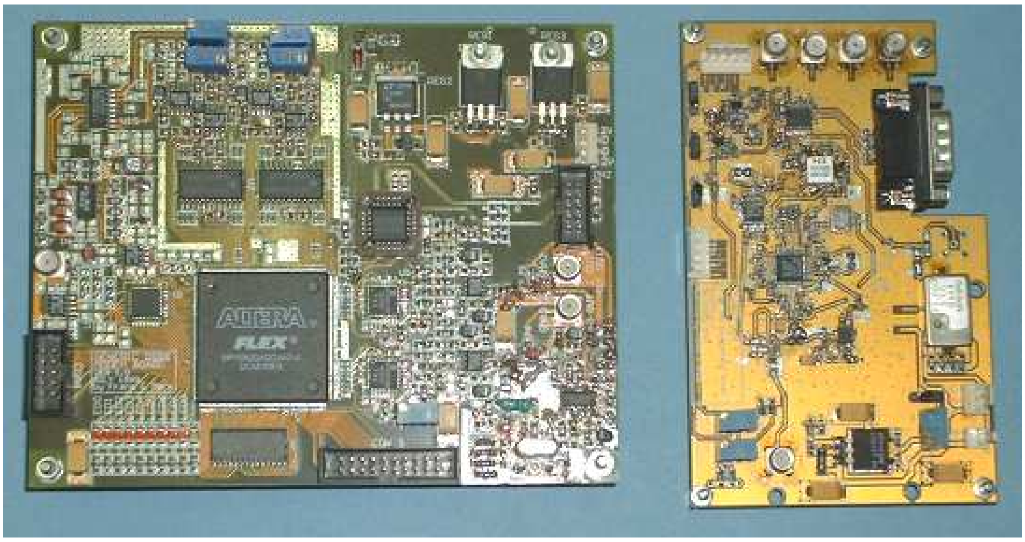


FIGURE 10.1: Hardware boards of the first design version of the DSSS modem. The FPGA board with the baseband processing on the left and the IF/RF board on the right.

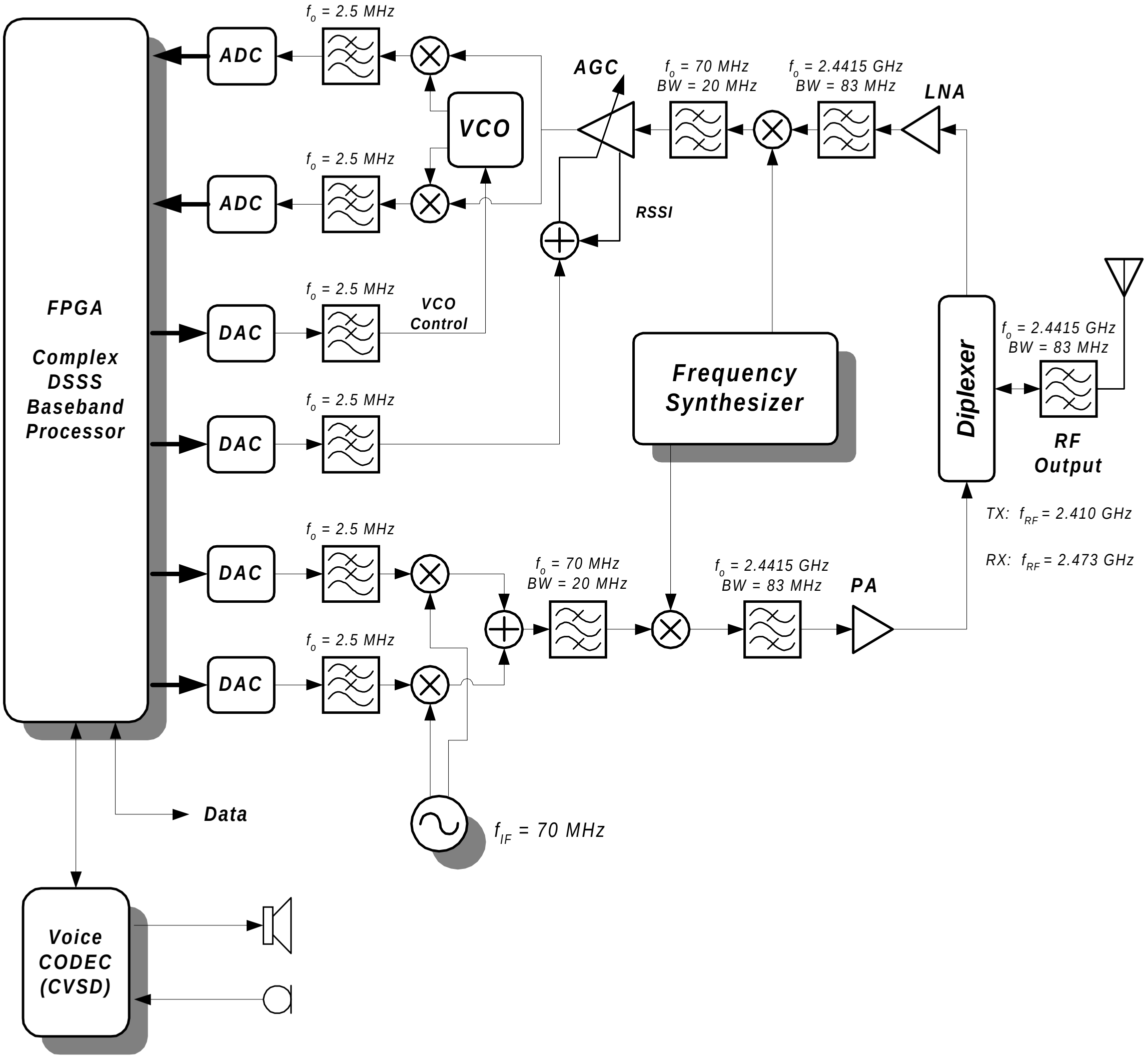


FIGURE 10.2: Block diagram of the first designed hardware implementation of the complex DSSS modem structure.

10.2.1.2 Hardware Design of the DSSS Modem Employing CSS: Final Version

The final design and implementation of the DSSS modem, consisting of the transmitter and receiver blocks, are shown in Figures 10.3 and 10.4, respectively. The hardware setup of the final DSSS modem is depicted in Figure 10.7.

Two generic FPGA based DSP platforms were developed on which the DSSS modem was implemented. The FPGA development boards each consist of four $20K600E$ FPGA chips. The FPGA development platform has a system clock of $80\ MHz$. This system clock is used to clock all the FPGAs, on the board as well as the DACs and ADCs needed to establish communication to and from the DSSS modem.

The DSSS modem structure consists of a voice CODEC, complex DSSS baseband processor, IF section and RF section. The complex DSSS baseband processor contains the DSSS transmitter core software, receiver, code acquisition, code tracking, IF sampling, numerically controlled oscillator (NCO), carrier phase tracking, automatic gain control (AGC), interfacing with the data source and voice CODEC as well as the overall setup and control of the DSSS modem module. The interfacing with the IF section and control signals is done by means of ADC and DAC circuitry.

10.2.1.3 DSSS Modem Transmitter Module

The DSSS transmitter, shown in Figure 10.3, accepts binary data as input. These data bits, are firstly differentially encoded and then spread using composite complex spreading sequences. The spreading block, differential encoding, spreading code generator, code control and clock generators forms part of the DSSS baseband processor. This baseband processor runs at a system / sample clock speed of $80\ MHz$. The baseband in-phase and quadrature phase branches, generated at the transmitter, are converted to analog signals by means of two DACs. These analog signals are then lowpass filtered, modulated onto an $850\ MHz$ cosine and sine RF frequency and summed, by means of a quadrature modulator. The lowpass filtering following the DACs does not perform shaping, but eliminates unwanted frequency components due to sampling. This RF output is bandpass filtered, amplified and finally transmitted via an antenna.

10.2.1.4 DSSS Modem Receiver Part

With reference to Figure 10.4, the signal received from the antenna is amplified by a low noise amplifier (LNA), bandpass filtered at a RF centre frequency of $850\ MHz$

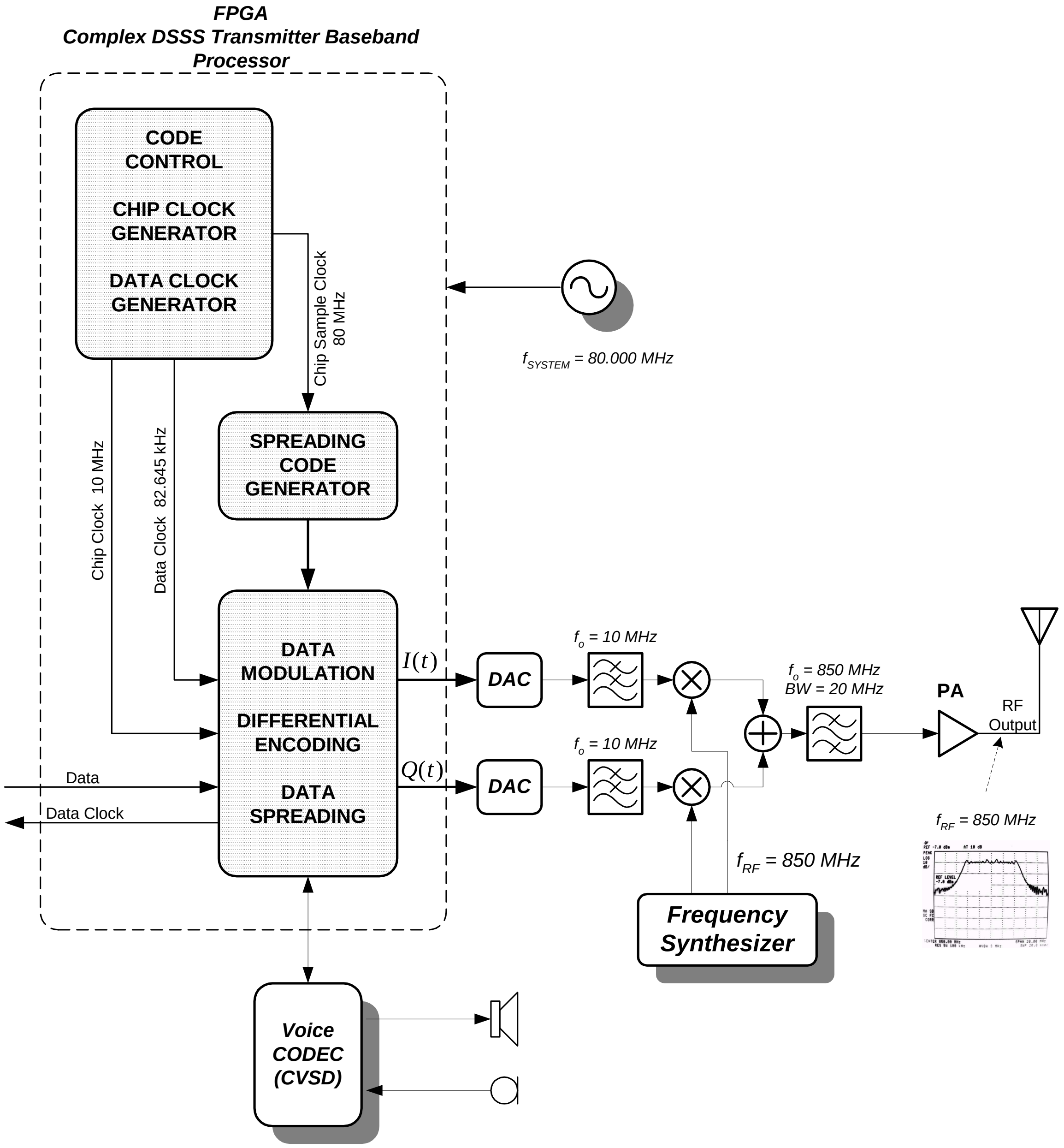


FIGURE 10.3: Block diagram of the final hardware implementation of the transmitter part of the DSSS modem employing CSS.

down-converted to $70\ MHz$ IF frequency. The $70\ MHz$ IF signal is passed through a bandpass SAW filter to eliminate all noise and interference power outside the relevant spreaded data bandwidth. The $70\ MHz$ IF is again down-converted to a secondary IF frequency of $16\ MHz$, which is low enough to be IF-sampled at a sampling rate of 80 $Msps$. Automatic gain control (AGC) is also performed on this IF-signal to ensure full range quantization on the ADC input. The control signal of the AGC is generated by using the measured average voltage level of the sampled IF signal, which is compared to a pre-specified reference level, in order to control the gain of the AGC amplifier. This enables a full range voltage input signal to be delivered to the ADC.

In the complex DSSS baseband processor FPGA the sampled $16\ MHz$ IF signal is down-converted to a baseband in-phase and quadrature branch, by means of a quadrature demodulation block. The quadrature local oscillator (LO) is provided by a numerically controlled oscillator (NCO), which forms part of a decision-directed Costas carrier-recovery-loop (DD-CCRL). A decision-directed complex delay-locked-loop (DD-CDLL) operates in parallel with the DD-CCRL, to track the received and locally generated spreading codes to within one chip period. The punctual complex spreading codes generated by the DD-CDLL are used in the DD-CCRL in the despreading process of the data. The recovered quadrature carriers, provided by the DD-CCRL, are used in the DD-CDLL to demodulate the incoming IF signal into two quadrature baseband branches, despreaded by the unique composite signals, to produce an error signal as described in Chapters 5 and 6. The combination of these two synchronization blocks, operating as jointly as one combined unit, in effect achieves a dual diversity gain, and provides superior performance compared to two independent tracking loops, controlled by two separate tracking error signals, as illustrated by Proakis in [21] p.333. An illustration of how the periodic autocorrelation function is affect by the combined code tracking and carrier recovery operation is illustrated in paragraph 6.6 Figure 6.10.

The spreading code acquisition operates in parallel with the combined synchronization block. The function of the acquisition process is to ensure alignment of the received and locally generated spreading codes within one chip period. This coarse code synchronization forms part of the initial synchronization phase. After this initial acquisition phase, the DD-CDLL swithes over into closed-loop operation mode for fine and accurate code tracking.

The numerically controlled chip clock generator (NCCCG), controlled by the DD-CDLL, drives the complex spreading code generator. The complex spreading code generator produces the two punctual combinations of complex spreading sequences, used in the data

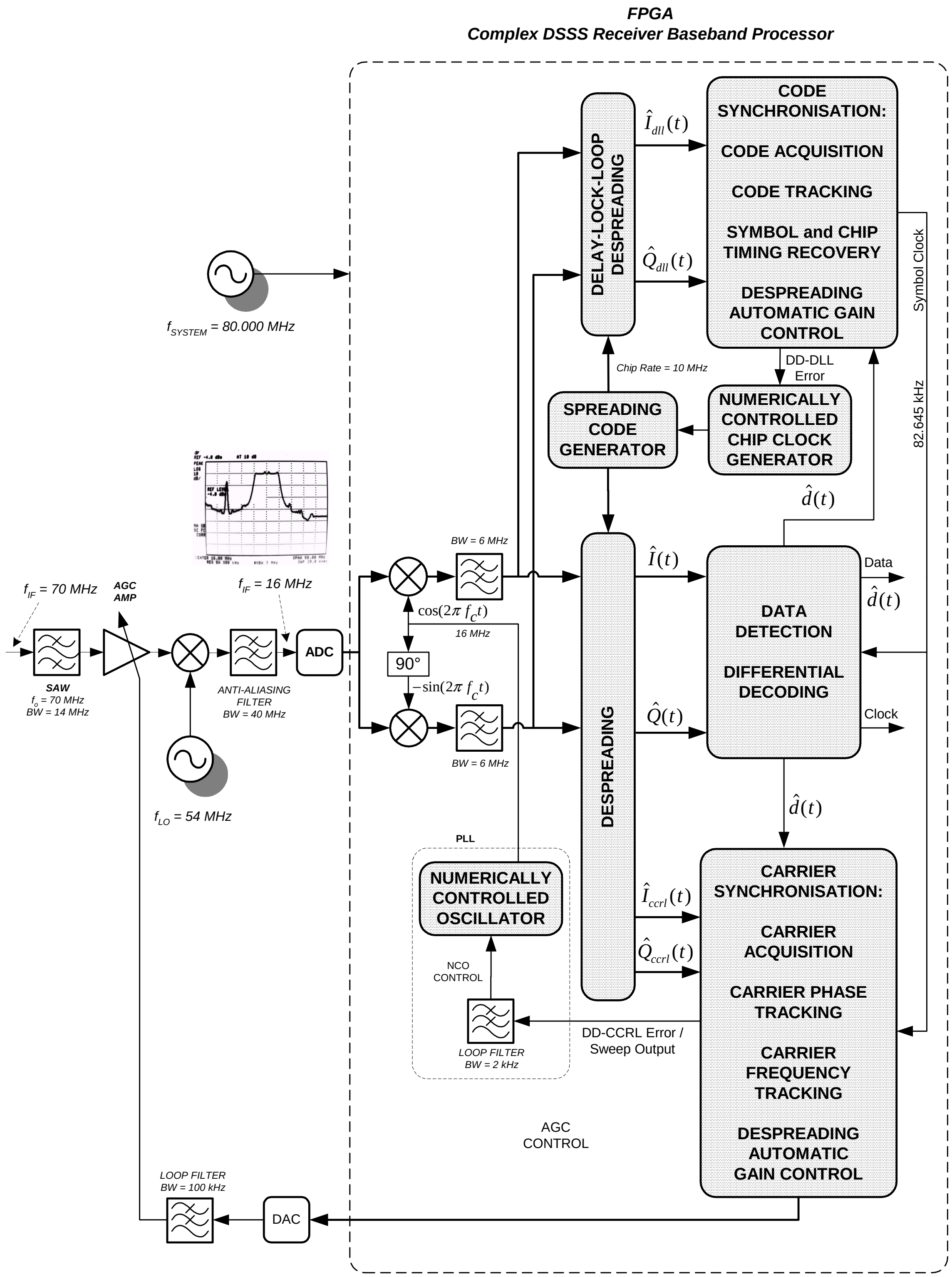


FIGURE 10.4: Block diagram of the hardware implementation of the final receiver of the DSSS modem employing CSS.

despreading process. It also produces the composite pair of combinations of complex spreading sequences, used by the DD-CDLL to generate the control error signal for controlling the NCCCG. The DD-CDLL automatically recovers the symbol timing by fine tracking the spreading code within one chip period.

10.2.2 Voice CODEC Module

A voice CODEC, as depicted in Figure 10.5, digitally encodes the incoming voice signal from a microphone, which is automatic gain controlled. It also decodes the received data and provides an analog signal, which is amplified to drive a speaker.

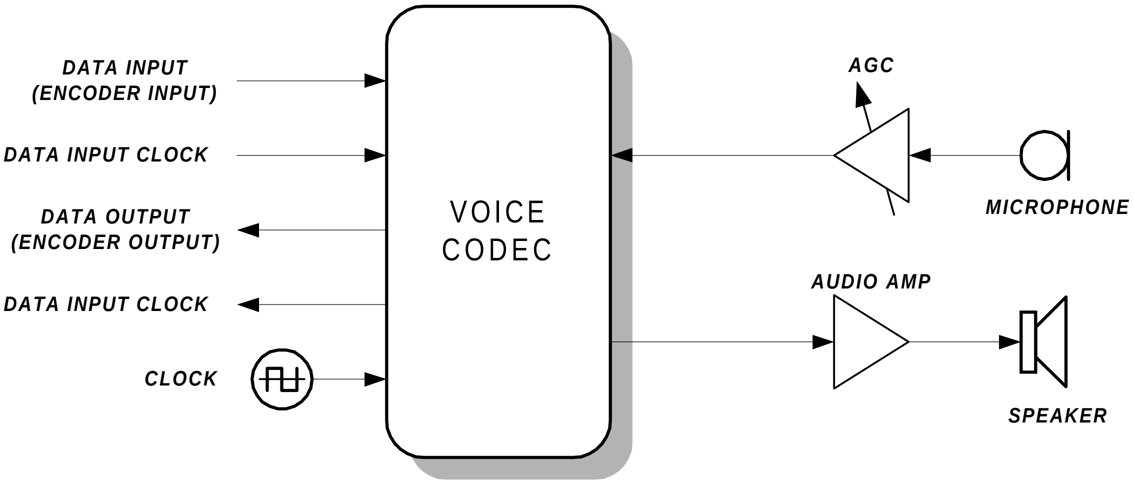


FIGURE 10.5: Block diagram of the audio front-end.

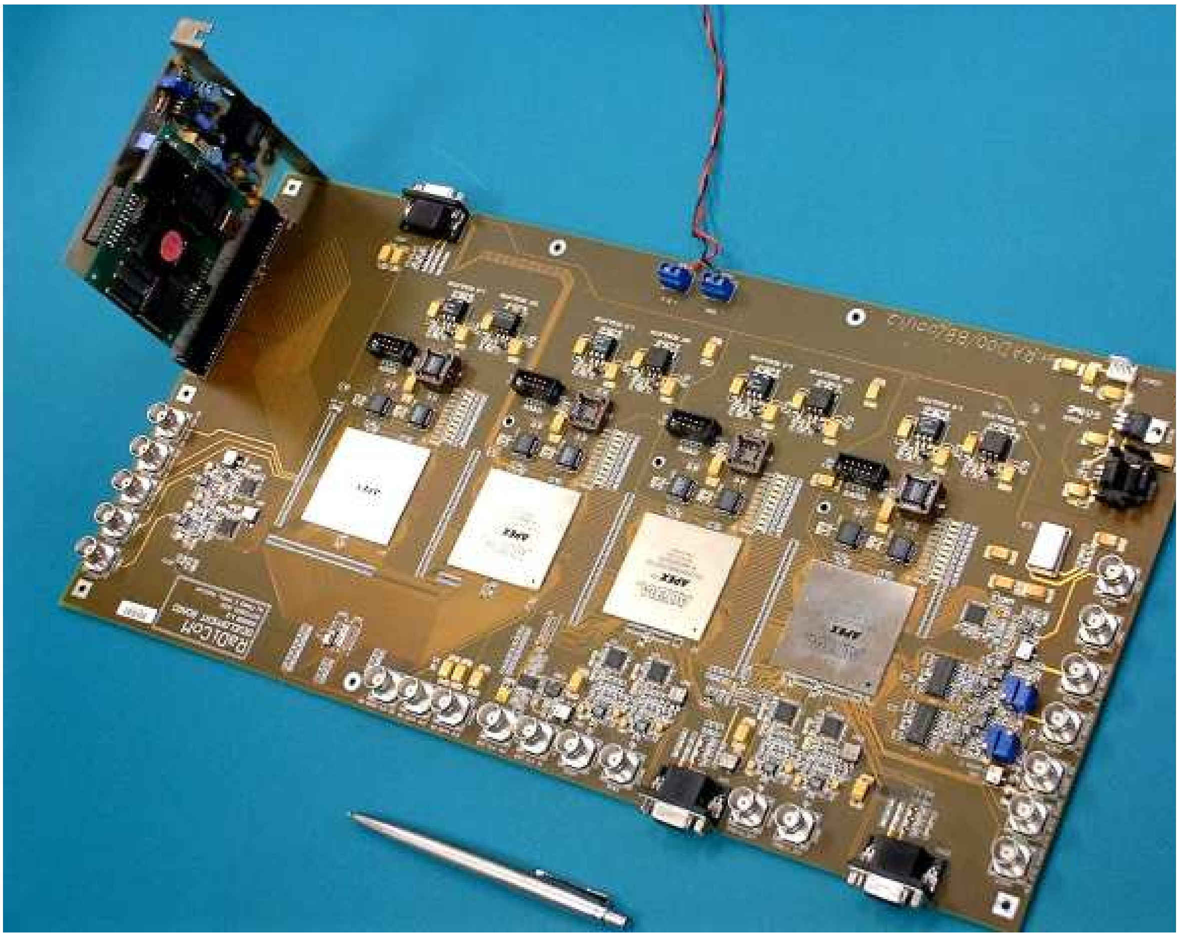


FIGURE 10.6: FPGA development board used as development platform for the DSSS modem employing complex spreading sequences.

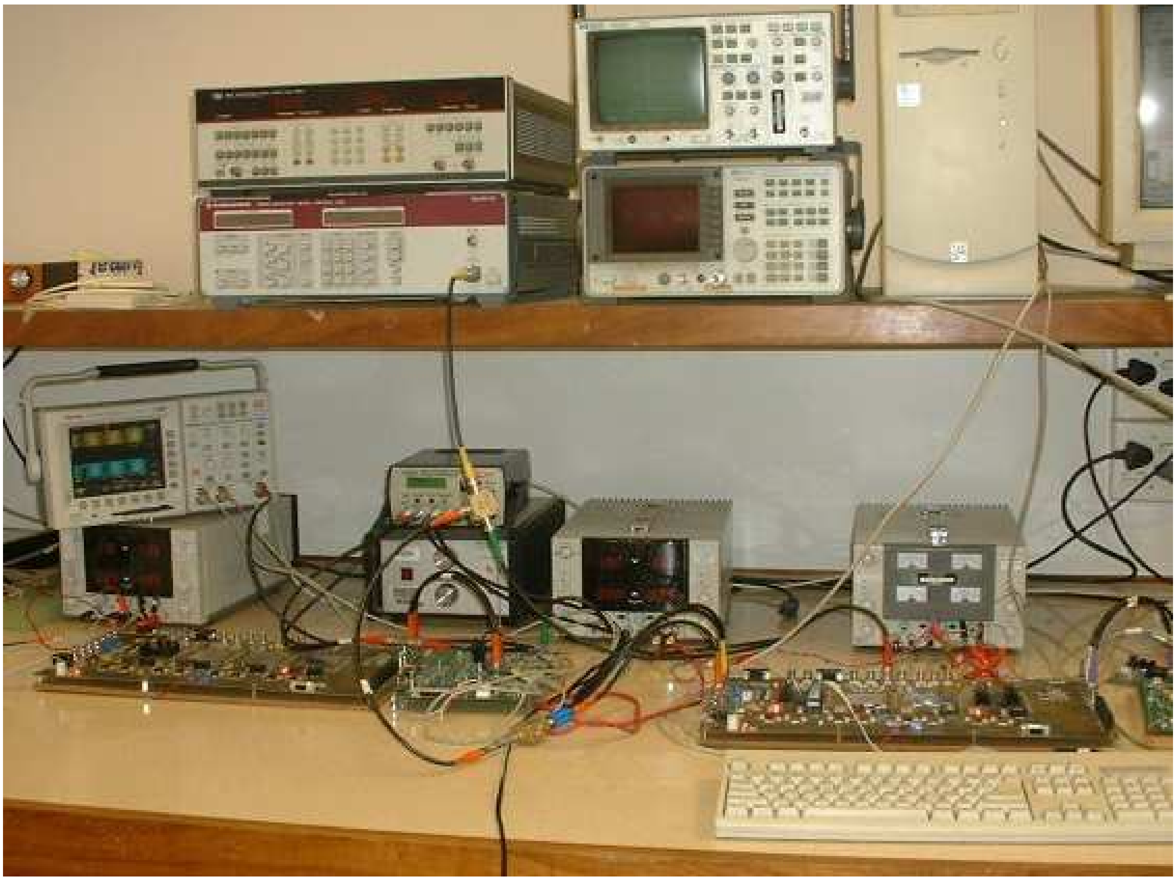


FIGURE 10.7: The hardware setup of the final DSSS modem employing CSS.

10.3 SYSTEM SPECIFICATIONS

The system specifications for the DSSS modem are given in Table 10.1. This table includes both the first design version specifications and the final design system specifications for the DSSS modem employing CSS. The system can easily be software configured to establish a system with different spreading sequence length, processing gain (PG) , data rate, transmission bandwidth, balanced or dual mode DSSS QPSK modulation, system sample frequency, etc. Some of these options are summarized in Tables 10.2, 10.3 and 10.4.

PARAMETER	**SPECIFICATION**
BASEBAND PROCESSOR	
Modulation technique	Balanced/dual channel QPSK DSSS
Spreading sequence	Families of complex spreading sequences
Sequence length	L (selectable - see Tables 10.2, 10.3 and 10.4)
Data rate	f_b (selectable - see Tables 10.2, 10.3 and 10.4)
Processing gain	$PG = 10logL$ (see Tables 10.2, 10.3 and 10.4)
Chip rate	$f_{chip} = 5/10/20Mchips/s$
Samples per chip	$spc = 4/8$
Sampling frequency	$f_{sample} = 20/80MHz$
Baseband bandwidth	$B_B = 2.5/5/10MHz$
RF TRANSMIT MODULE	
TX intermediate frequency	$f_{IF_{TX}} = 70/280/374MHz$
TX RF carrier frequency	$f_{RF_{TX}} = 2442/850MHz$
Transmission bandwidth	$B_T = 5/10/20MHz$
RF RECEIVE MODULE	
RX intermediate frequency	$f_{IF_{RX}} = 70/16/8MHz$
RX RF carrier frequency	$f_{RF_{RX}} = 2442/850MHz$
AGC dynamic range	$30dB$
LNA noise figure	$2\ dB$

TABLE 10.1: Complex DSSS modem specifications

	BALANCED QPSK	DUAL CHAN QPSK		
	DATA RATE (kbps)	DATA RATE (kbps)	L	PG (dB)
1	454.55	909.09	11	10.41
2	384.62	769.23	13	11.14
3	333.33	666.67	15	11.76
4	294.12	588.24	17	12.30
5	217.39	434.78	23	13.62
6	172.41	344.83	29	14.62
7	161.29	322.58	31	14.91
8	138.89	277.78	36	15.56
9	102.04	204.08	49	16.90
10	81.97	163.93	61	17.85
11	79.37	158.73	63	17.99
12	41.32	82.64	121	20.83
13	39.37	78.74	127	21.04

Table 10.2: Gross data rates for the DSSS modem with various spreading sequence lengths (L) and processing gains (PG) (Transmission bandwidth of $5MHz$; Chip rate of $f_{chip} = 5Mcps$).

	BALANCED QPSK	DUAL CHAN QPSK		
	DATA RATE (kbps)	DATA RATE (kbps)	L	PG (dB)
1	909.09	1818.18	11	10.41
2	769.23	1538.46	13	11.14
3	666.67	1333.33	15	11.76
4	588.24	1176.47	17	12.30
5	434.78	869.57	23	13.62
6	344.83	689.66	29	14.62
7	322.58	645.16	31	14.91
8	277.78	555.56	36	15.56
9	204.08	408.16	49	16.90
10	163.93	327.87	61	17.85
11	158.73	317.46	63	17.99
12	82.64	165.29	121	20.83
13	78.74	157.48	127	21.04

Table 10.3: Gross data rates for the DSSS modem with various spreading sequence lengths (L) and processing gains (PG). (Transmission bandwidth of $10MHz$; Chip rate of $f_{chip} = 10Mcps$).

	BALANCED QPSK	DUAL CHAN QPSK		
	DATA RATE (kbps)	DATA RATE (kbps)	L	PG (dB)
1	1818.18	3636.36	11	10.41
2	1538.46	3076.92	13	11.14
3	1333.33	2666.67	15	11.76
4	1176.47	2352.94	17	12.30
5	869.57	1739.13	23	13.62
6	689.66	1379.31	29	14.62
7	645.16	1290.32	31	14.91
8	555.56	1111.11	36	15.56
9	408.16	816.33	49	16.90
10	327.87	655.74	61	17.85
11	317.46	634.92	63	17.99
12	165.29	330.58	121	20.83
13	157.48	314.96	127	21.04

Table 10.4: Gross data rates for the DSSS modem with various spreading sequence lengths (L) and processing gains (PG) (Transmission bandwidth of $20MHz$; Chip rate of $f_{chip} = 20Mcps$).

10.4 HARDWARE MEASUREMENT RESULTS

Several measurement results were obtained from the hardware prototype DSSS modem (transmitter and receiver), employing CSS. These measurement results are presented and discussed in this section. The DSSS modem was configured with a CSS of length, $L = 121$, number of samples-per-chip, $spc = 8$, chip rate, $f_{chip} = 10Mcps$, a corresponding data rate of $f_{data} = 82.645kbps$ and processing gain of $PG = 20.8dB$. The measured results are presented in the following sections.

10.4.1 Complex DSSS Transmitter (Modulator)

Figure 10.8 shows the spectrum of the non-return-to-zero (NRZ) serial input data before spreading with a frequency span of $280\ kHz$. The first null of the spectrum corresponds with the data symbol rate of $82.645\ kbps$. The measured spectrum of the in-phase and quadrature spread branches are shown in Figures 10.9 and 10.10, respectively, with a frequency span of $20\ MHz$. The in-phase and quadrature spread branch signals are depicted in Figure 10.11, with a zoomed-in version in Figure 10.12, showing the polarity changes in the periodic sequences due to the data information.

The effect of data spreading is illustrated in Figures 10.13 and 10.14. In Figure 10.13 the spectrum of the unspread data is shown, while the spectrum of the spread data ($L = 121$) is depicted in Figure 10.14, both with a frequency span of $20\ MHz$.

The in-phase branch versus the quadrature branch plot is depicted in Figure 10.15. This plot is an indication of the power envelope of the final output signal of the transmitter in a two dimensional communication system. For the case of the balanced and dual channel QPSK DSSS communication systems, employing CSS, the in-phase branch vs. quadrature branch forms a perfect circle, indicating that the power envelope of the signal to be transmitted, will be constant. The constant envelope feature of the DSSS QPSK system employing CSS yields multiple advantages. The most important advantage of the system is power efficiency, which is related to most of the other benefits. These benefits include a communication system with the ability to operate close to the power amplifier $1\ dB$ compression point, and, consequently, a communication system with higher output power for a given maximum available supply power, yielding an extended coverage area. The power efficiency advantages ultimately relates to longer battery life, which again facilitates cheaper cellular handsets.

The final output spectrum of the DSSS transmitter employing CSS is shown in Figure 10.16. The $-6\ dB$ double sided bandwidth of the output is $10MHz$, which is the same as

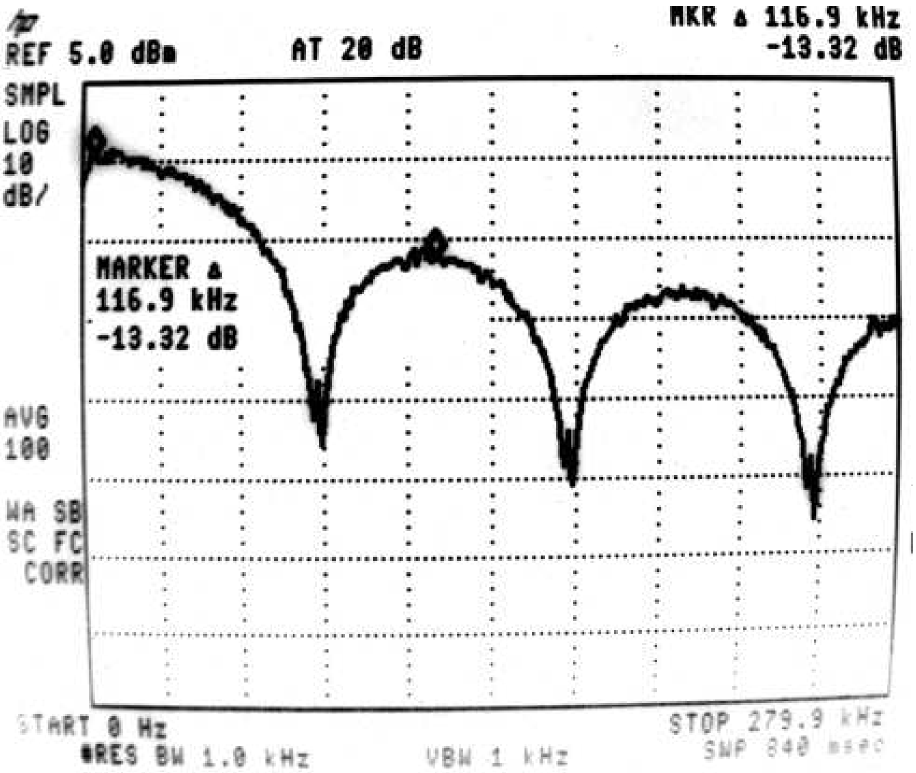


FIGURE 10.8: Spectrum of the NRZ serial input data before spreading.

the chip rate. This output spectrum is equivalent to that of a DSSS system employing binary spreading sequences filtered with a Nyquist filter with an effective roll-off factor of $\alpha = 0$ for sufficiently long spreading sequence lengths.

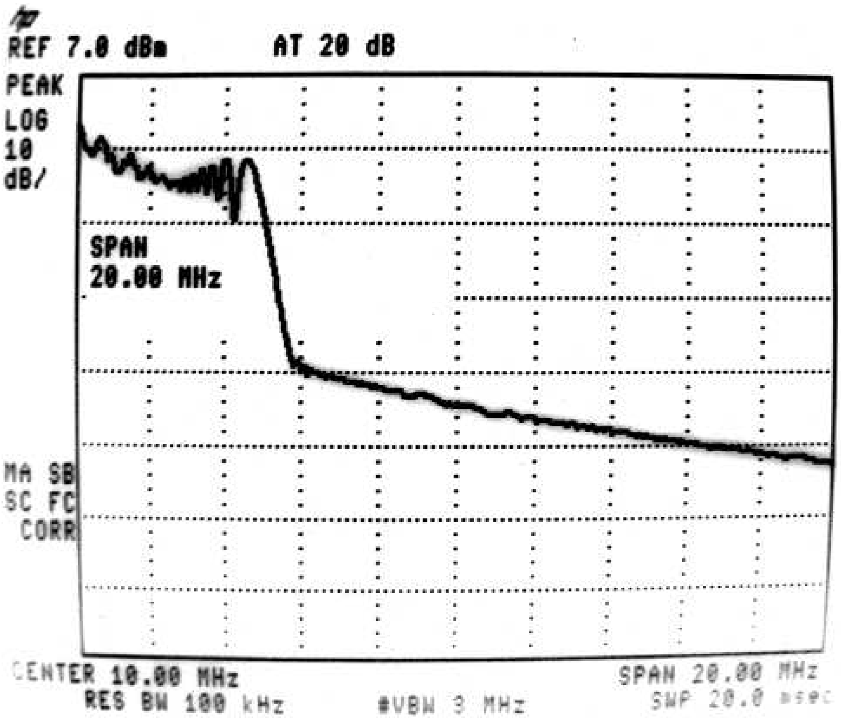


FIGURE 10.9: Measured spectrum of the in-phase branch composite complex spreading sequence in the DSSS transmitter.

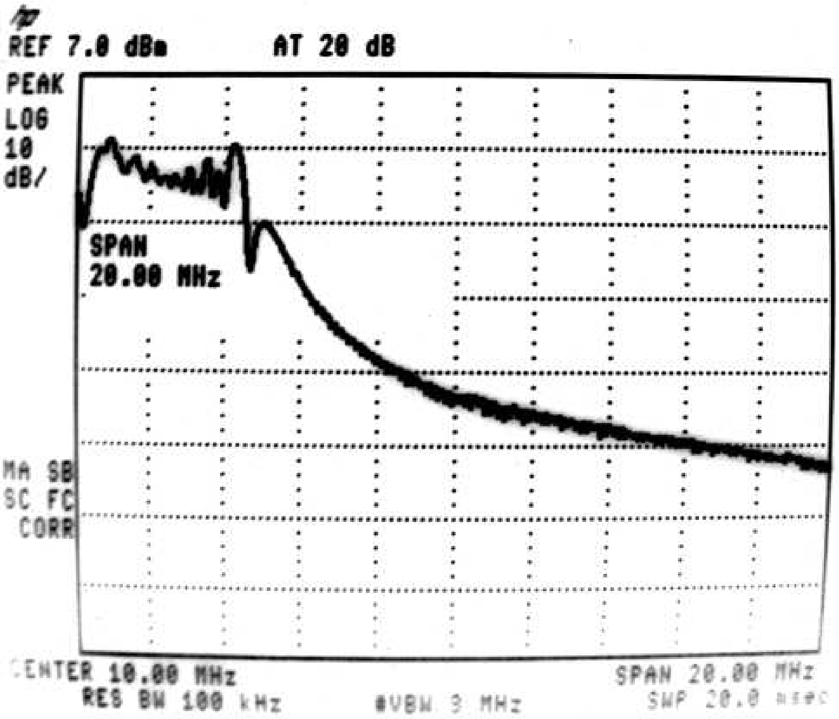


FIGURE 10.10: Measured spectrum of the quadrature branch composite complex spreading sequence in the DSSS transmitter.

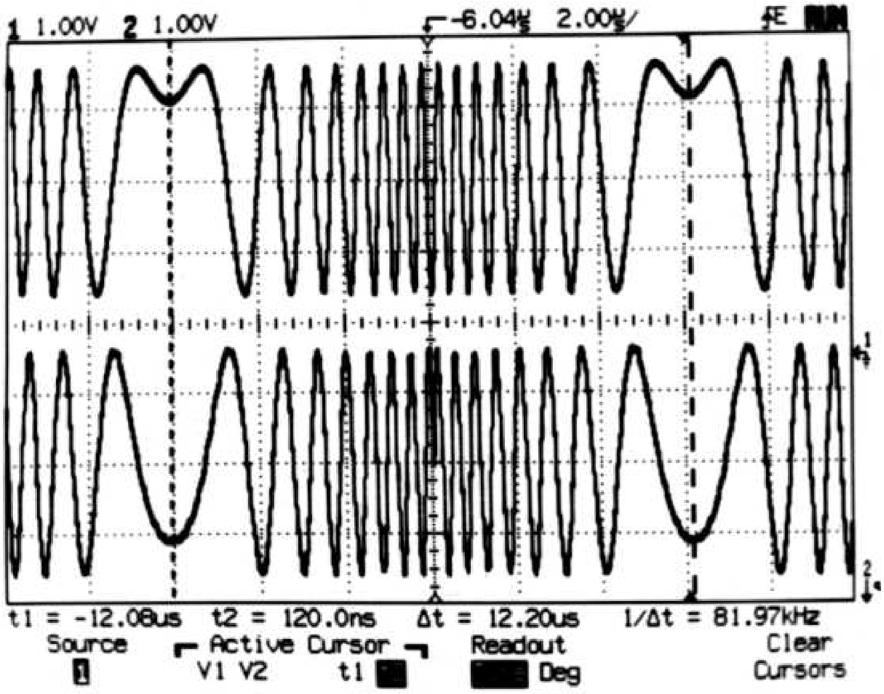


FIGURE 10.11: The two unique combinations of complex spreading sequences as implemented in the transmitter.

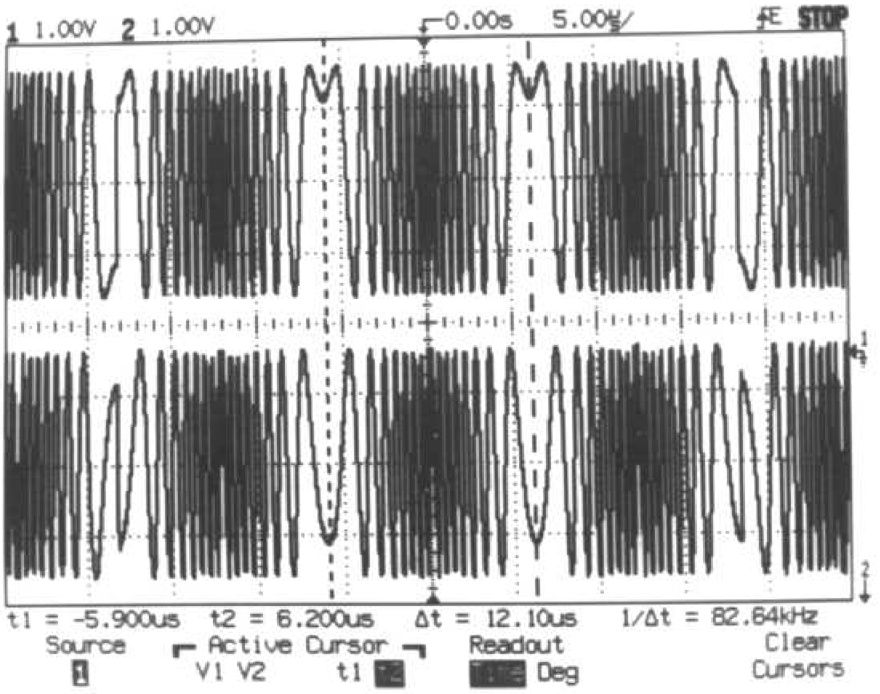


FIGURE 10.12: In-phase and quadrature phase branch signals, where the random data streams are spreaded with the unique combinations of complex spreading sequences, at the transmitter.

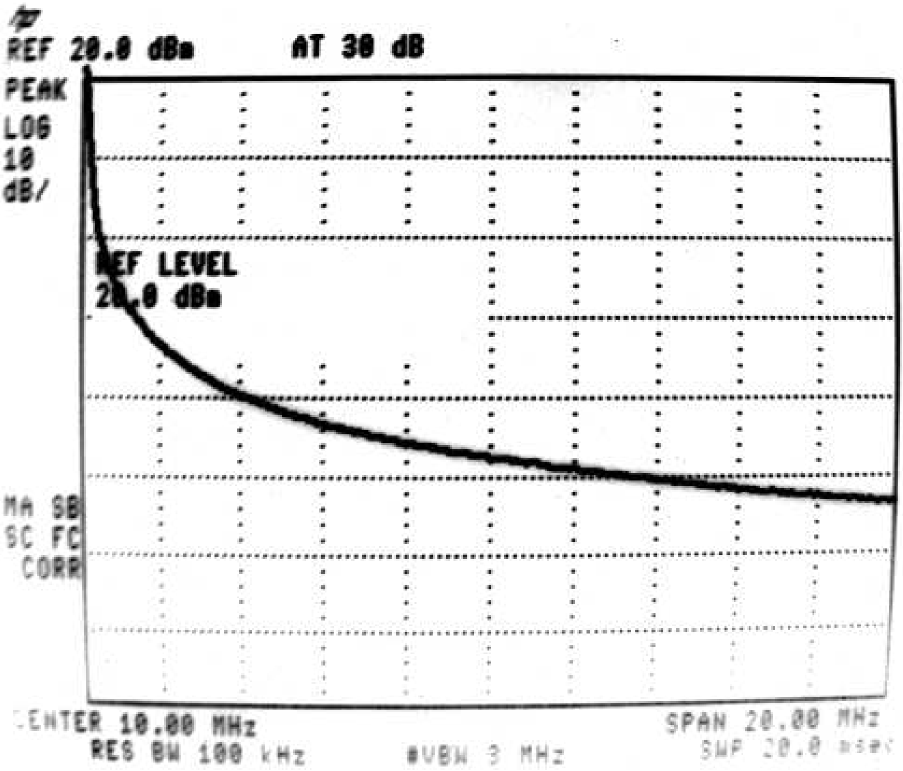


FIGURE 10.13: Spectrum of the NRZ serial input data before spreading.

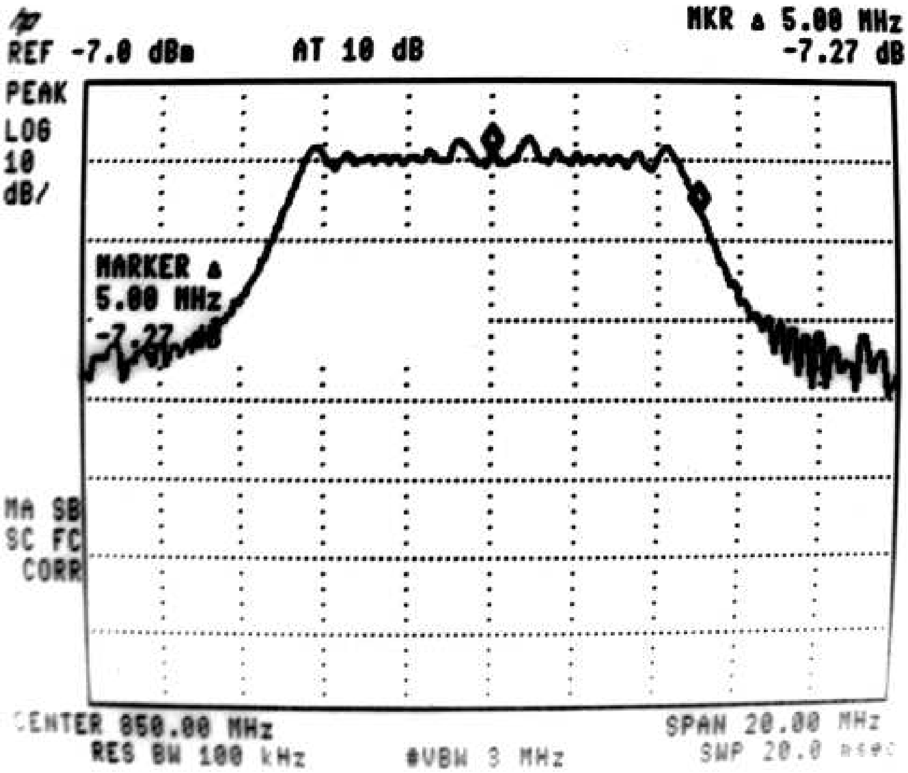


FIGURE 10.14: Measured spectrum of the quadrature branch composite complex spreading sequence in the DSSS transmitter.

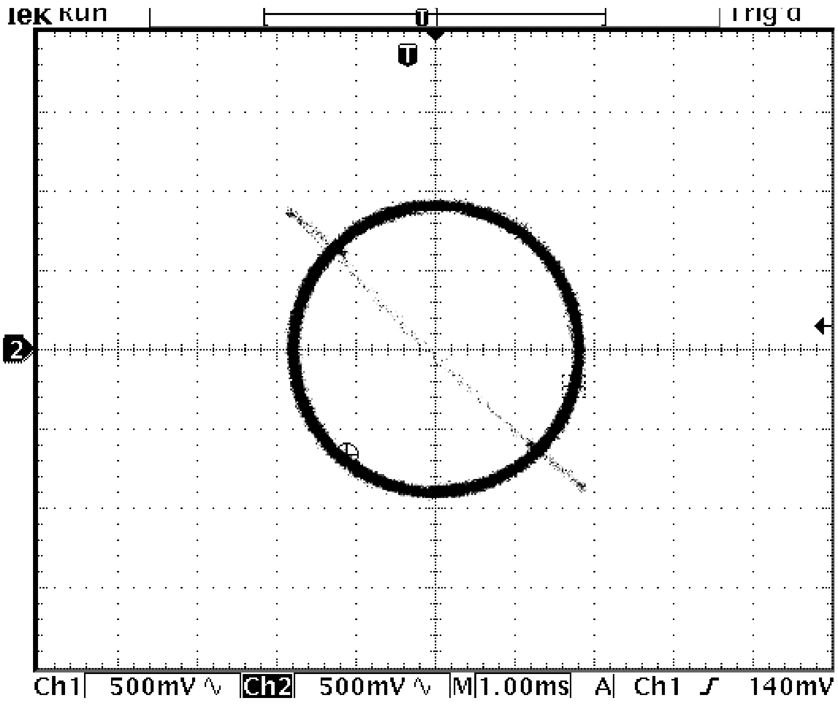


FIGURE 10.15: Measured in-phase versus quadrature branch plot at the output of the DSSS transmitter, employing CSS, to illustrate the constant envelope output.

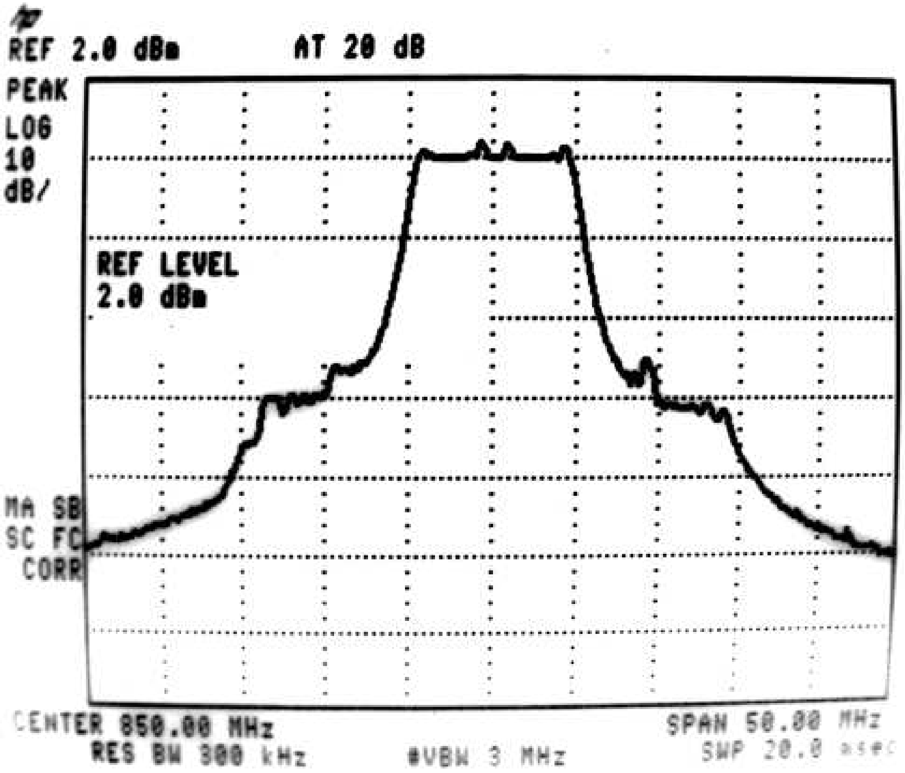


FIGURE 10.16: Final output spectrum of the DSSS transmitter employing CSS.

10.4.2 Complex DSSS Receiver (Demodulator)

The input to the receiver is down-converted to an IF of $16MHz$ before the sampling and analog-to-digital conversion process (ADC) is done. The spectrum of the signal on an IF of $16MHz$ is depicted in Figure 10.17. The composite in-phase and quadrature difference sequences used in the DD-CDLL, to determine the code error, are shown in Figure 10.18, top and bottom, respectively.

The Auto-Correlation peak, obtained from the sliding correlation process between the incoming spreading code and locally generated spreading code at the receiver, is depicted in Figure 10.19. This auto-correlation peak is produced from the sample-and-hold output after the integrate-and-dump operation in the sliding correlation despreading process. The integrate-and-dumped output is also shown in the bottom part of Figure 10.19. A zoomed-in version of this auto-correlation peak, as well as the integrate-and-dumped output, are illustrated in Figure 10.20

Another factor that influence the performance of the auto-correlation process, except for the time shift between the received and locally generated spreading sequence, is the phase error between the received and recovered carrier. The DD-CCRL is responsible for tracking the phase of the incoming carrier. Figure 10.21 shows the auto-correlation output (top) with the corresponding carrier phase error (DD-CCRL open-loop error) (bottom). A zoomed-in version of this hardware result is shown in Figure 10.22.

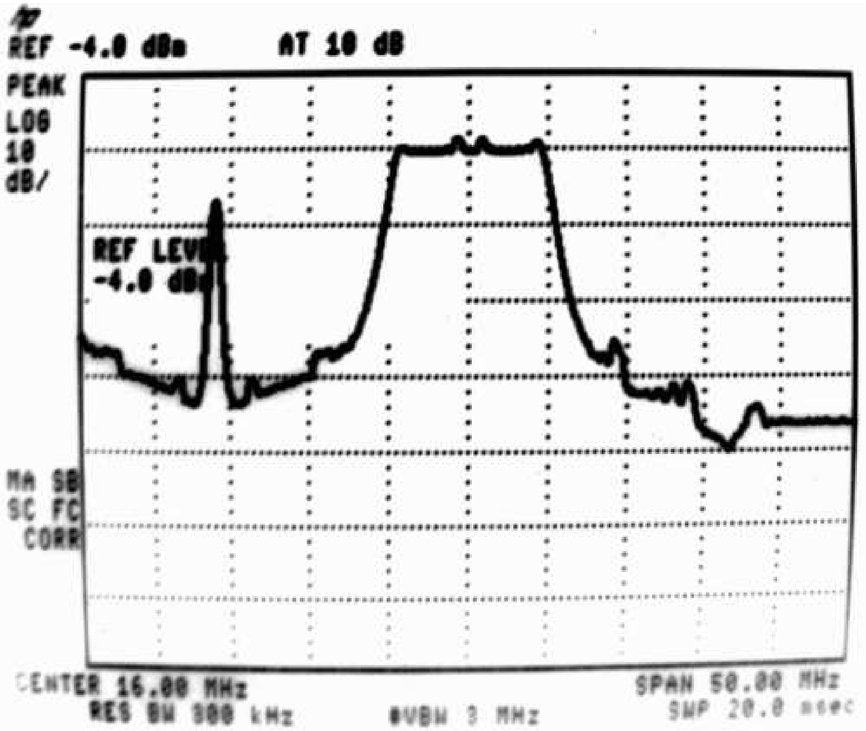


FIGURE 10.17: Spectrum of the incoming signal at the receiver on an IF of 16 MHz, as input to the ADC.

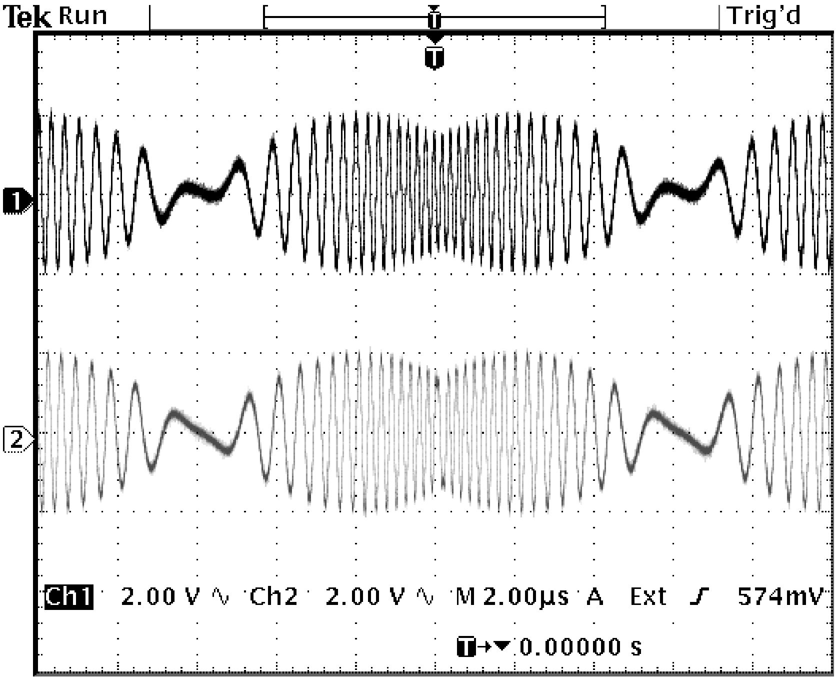


FIGURE 10.18: Composite in-phase (top) and quadrature (bottom) difference sequences used in the CDLL to determine the code error.

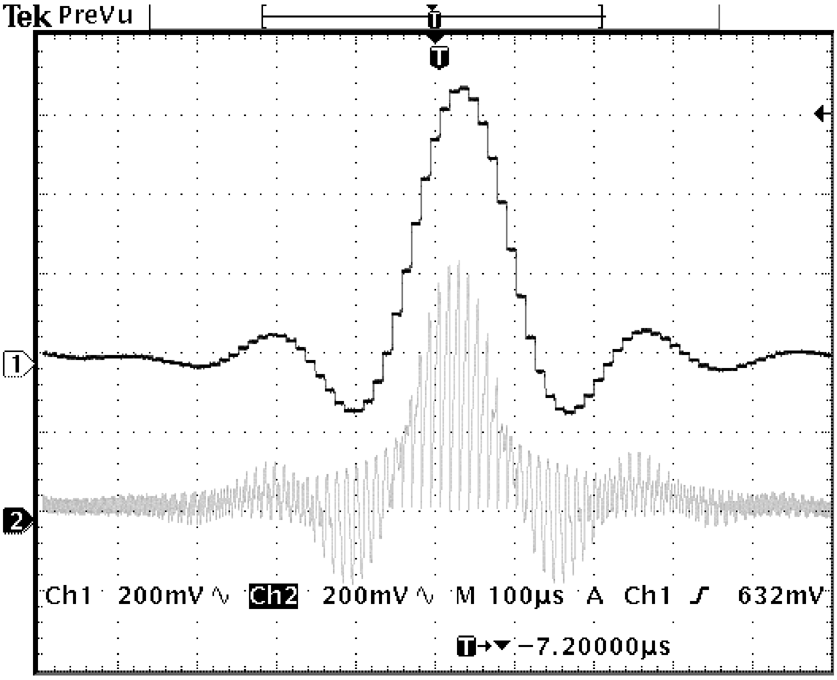


FIGURE 10.19: Auto-Correlation peak obtained (top graph) as output of sliding correlation performed between incoming spreading code and locally generated spreading code at the receiver. Integrate-and-dump output of the sliding correlation output before sample-and-hold (bottom).

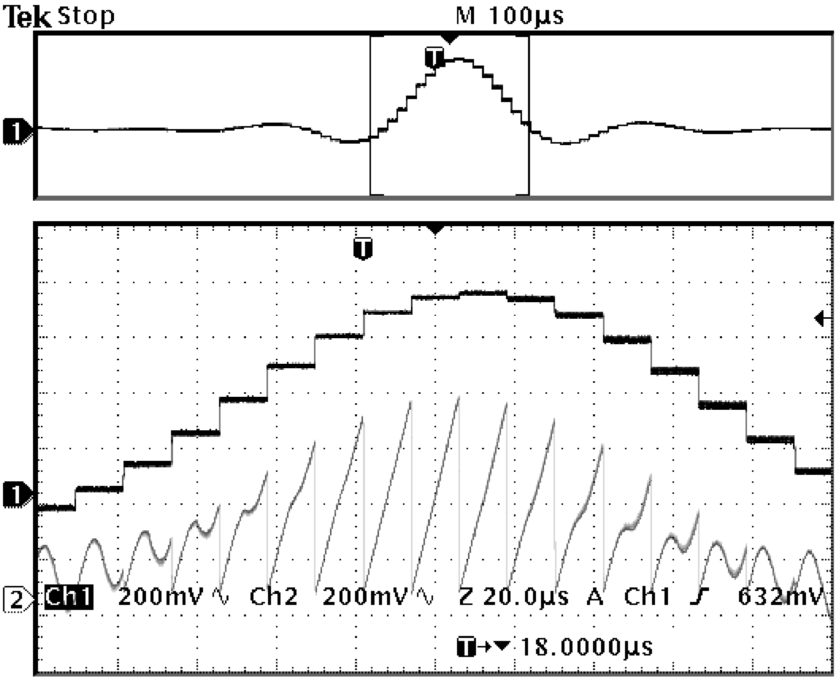


FIGURE 10.20: Auto-Correlation peak obtained (trace 1) as output of sliding correlation performed between incoming spreading code and locally generated spreading code at the receiver. Integrate-and-dump output of the sliding correlation output before sample-and-hold (trace 2).

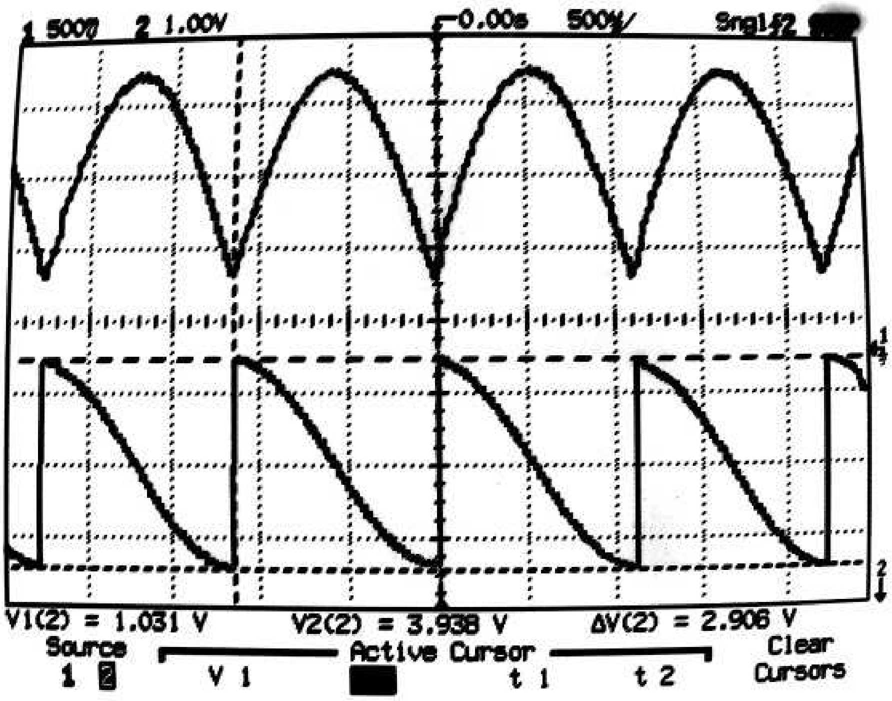


FIGURE 10.21: Auto-correlation output (top) after code acquisition has been acquired with the corresponding DD-Costas carrier recovery loop error (bottom) before carrier lock.

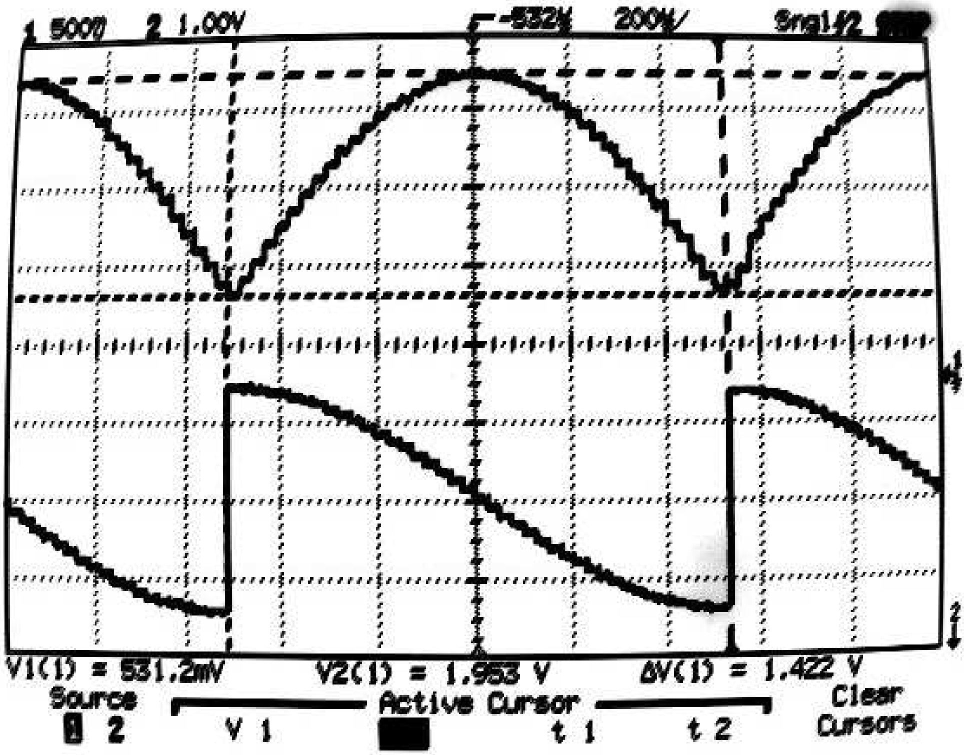


FIGURE 10.22: Zoomed in auto-correlation output (top) after code acquisition has been acquired with the corresponding DD-Costas carrier recovery loop error (bottom) before carrier lock.