

AN FPGA RECEIVER FOR CPSK SPREAD SPECTRUM SIGNALING

Sunny K. S. Chan and Victor C. M. Leung

Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, B. C., Canada, V6T 1Z4
{sunnyc, vleung}@ece.ubc.ca

Abstract

In this paper, we present the design, implementation and testing of an M-ary direct sequence spread spectrum receiver suitable for wireless home networking applications. The receiver employs a novel Code-Phase-Shift Keying (CPSK) signaling scheme, in which each of the M signaling waveforms is derived from a different phase shift of a single pseudonoise code sequence. The receiver consists of an IF demodulator and a CPSK baseband decoder, implemented using discrete components and an FPGA (field programmable gate array) chip, respectively. A modified double-dwell serial search scheme is used for code acquisition and tracking, and the carrier-phase synchronization is solved by a Costas loop in the IF demodulator and a double threshold detection scheme in the CPSK decoder. Measurements of receiver performance are presented and compared with theoretical calculations.

1. Introduction

With the emergence of home entertainment, automation, and information devices that are capable of being interconnected in home networks [1], there is increasing interest in the use of wireless transmissions in home networking [2]. Consumer wireless networking devices typically work in the licence-free instrumentation, scientific and medical (ISM) bands and employ spread spectrum (SS) signaling [3] in compliance with U.S. FCC Part 15 rules. Commercially available radio transceivers employing conventional direct sequence (DS) SS signaling scheme provide a data throughput up to 2 Mbps, limited by the available frequency bandwidth. The processing gain, defined by the ratio of bandwidth after and before the data signal is spread by a code sequence, is an important parameter affecting the performance of a SS system. Higher processing gain results in higher interference immunity; however, the data throughput is decreased correspondingly. One way to improve the throughput is to use an M-ary SS scheme in which different pseudonoise (PN) sequences are used to encode several data bits for transmissions [4]. However, this scheme has a drawback in that the PN codes may interfere with each other resulting in degraded bit error rate

(BER) performance. Although orthogonal maximal length PN sequences can be used, the number of available sequences for any given code length is limited. Another drawback is the need of extra hardware to generate different PN codes. To minimize these drawbacks, a novel M-ary SS signalling scheme, known as code-phase-shift keying (CPSK) has been proposed [5]. Previous performance evaluations [5] and implementation using digital signal processing (DSP) techniques [6] have confirmed the CPSK scheme to be practical while enhancing data throughput relative to conventional DS/SS schemes. However, the cost of building a CPSK transceiver (or modem) using DSP boards is too expensive and the data throughput too low if the signaling method is to be commercialized for consumer applications. Thus, an alternative approach to build a cost-reduced CPSK modem is necessary.

In this paper, we present the design, implementation and testing of a CPSK receiver consisting of an IF demodulator with a Costas loop and an FPGA-based CPSK decoder. An FPGA chip is used to minimize the number of logic components and ease system reconfiguration for different symbol sizes and PN code lengths. To evaluate the receiver performance, extensive symbol error rate (SER) measurements in the presence of additive white Gaussian noise (AWGN) and jamming from another CPSK transmitter have been conducted, and the resulting BER vs. signal to noise ratio (SNR) curves are compared with those of an ideal receiver.

An outline of this paper is as follows. Following this introductory section, section 2 gives an overview of the CPSK signaling scheme, the transmitter and receiver structure, and the theoretical SER in AWGN. Section 3 describes in details the implementation of the CPSK receiver, concentrating on the design of the baseband CPSK decoder and its implementation in an FPGA chip. Performance measurements in AWGN and with CPSK jamming are presented in section 4. Concluding remarks are given in section 5.

2. The CPSK Signaling Scheme

A CPSK transmitter is shown in Figure 1. The data are grouped into k -bit data symbols each represented by an

integer m , $0 \leq m \leq M - 1$. Each symbol selects one of the $M = 2^k$ code sequences $p_m(t) = p(t - m_c T_c)$ which is phase-shifted by $m_c = m(L+1)/M$ chips, where L is the length of the common PN sequence with chip interval T_c . Different code-phase-shifted sequences are derived from a PN code generator with a code chip delay line. The selected sequence is then up-converted for transmission at carrier frequency ω_c with the transmitted signal:

$$s(t) = A p_m(t) \cos(\omega_c t) \quad (1)$$

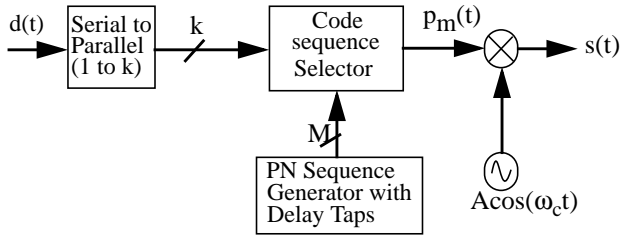


Figure 1: The CPSK Transmitter

The received signal $r(t) = s(t) + j(t) + n(t)$ at the input of the receiver with jamming and noise is fed to a bank of M correlators as shown in Figure 2. The decision device locates the correlator which gives the largest output and selects the corresponding symbol for parallel-to-serial decoding into the output data. The SER in AWGN with zero mean and two sided spectral density $N_0/2$ is

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \left\{ 1 - \left(\frac{1}{2} \operatorname{erfc} \left(\frac{y}{\sqrt{2}} \right) \right)^{M-1} \right\} \exp \left(-\frac{(y - \sqrt{2(E_s/N_0)})^2}{2} \right) dy \quad (2)$$

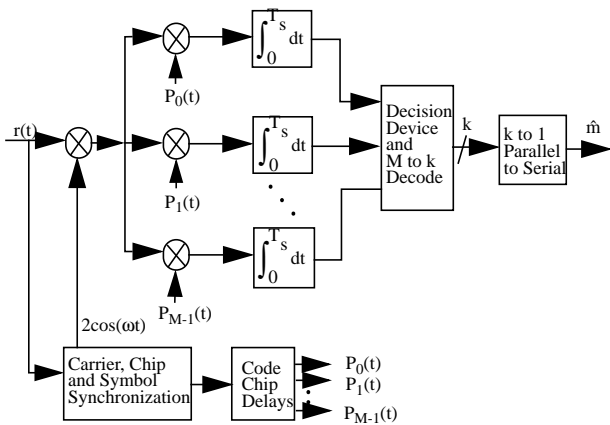


Figure 2: The CPSK Receiver

The detailed calculation of the above error probability can be found in [5], which shows that the performance of CPSK is similar to M-FSK, in that power efficiency

increases with M . When $k \geq 3$, CPSK requires a lower E_b/N_0 (bit-energy to noise-density ratio) than BPSK at any given BER. For example, at $\text{BER} = 10^{-5}$, $E_b/N_0 = 9.5$ dB for BPSK-DS/SS while $E_b/N_0 = 8.3$ dB, 7.3 dB, 6.7 dB, and 6 dB with $k = 3, 4, 5$, and 6, respectively for CPSK. Therefore, at $k = 6$, the power saving for CPSK is 3.5 dB. Moreover, the bandwidth of CPSK is dependent on the spreading gain but not M , and the bandwidth efficiency can be increased by increasing M (i.e., the number of different code-phases), as long as the code length is long enough to accommodate M different code phases.

3. CPSK Receiver Implementation

A CPSK receiver was implemented for $M = 8$, PN code length = 127, and a PN chip rate of 1 Mcps (this chip rate limitation was imposed by the available transmitter). The receiver implementation consists of an IF demodulator assembled with discrete components, and a CPSK digital decoder implemented in a commercial FPGA chip. To enable the received data to be read into a PC, a PC interface consisting of some simple logic circuits and some FIFO memory was also implemented. As with conventional DS/SS systems, carrier phase and PN code phase synchronization are two important issues to be addressed in the design of the CPSK receiver. Carrier phase synchronization is solved by employing a Costas loop [7] in the IF demodulator to track the carrier phase and limit its uncertainty to $\pm 180^\circ$. The uncertainty is then resolved by using double thresholds in the CPSK decoder. A modified double dwell serial search scheme with a digital tracking loop is employed in the CPSK decoder to achieve PN code phase synchronization [4][8]. The design of the IF demodulator and the CPSK decoder are discussed in more detail below.

3.1 The IF Demodulator

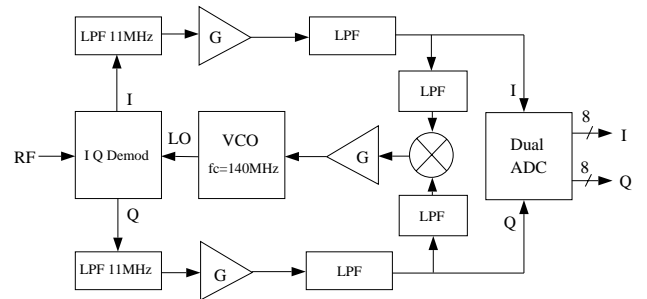


Figure 3: The IF Demodulator

The block diagram for the IF demodulator is shown in Figure 3. The received RF signal is demodulated into In-phase (I) and Quadrature-phase (Q) components which are then filtered by an 11 MHz low pass filter to suppress harmonics

above 11 MHz. The cut-off frequency is selected to allow the chip rate of the PN code to be increased up to 4 Mcps in the future. The filtered signals are amplified and level shifted to the range between 0 and 2V as required by the analog-to-digital converter (ADC). A second stage first order low pass active filter is combined with the gain stage to provide dominant pole cutoff at an optimal cutoff frequency calculated to maximize the SNR [9].

The feedback path of the Costas loop consists of a four quadrant multiplier and a voltage controlled oscillator (VCO). The low-passed I and Q signals are filtered and fed into the multiplier. The analog multiplier output is gain adjusted and level shifted to provide a control signal to the VCO which provides the Local Oscillator (LO) signal with a nominal frequency of 140 MHz for demodulating the received IF signal. The demodulated I and Q signals are also fed into a dual 8-bit ADC. Note that the I and Q channels must carry the same baseband signal for the Costas loop to work properly. The CPSK decoder provides a sampling clock at twice the chip rate to sample the demodulated I and Q signals and convert them into digital data for decoding. Only the I channel ADC output is used by the CPSK decoder. The detailed circuit of the IF demodulator can be found in [10].

3.2 The CPSK Baseband Decoder

The CPSK decoder receives the digital data from the I channel ADC output of the IF demodulator and decodes the signal using the scheme shown in Figure 2. Although the decoding process is done digitally, the master clock needs to be adjusted by some analog circuit in order to maintain continuous tracking of the PN code in the received signal. Therefore, the CPSK decoder is composed of a digital section implemented in a commercial FPGA, and an analog section implemented with discrete components. The FPGA is programmed using a commercial computer-aided design (CAD) system. The block diagram of the CPSK decoder is shown in Figure 4.

The decoder is designed to decode CPSK signals with a PN sequence length of 127 and word (symbol) size of 3. Therefore a 127-stage shift register fed by a PN code generator is used to provide PN codes with different phase shifts. A bank of 8 correlators is used to detect the 8 different code phases corresponding to a 3-bit word. The correlator outputs are then fed into 8 threshold devices for changing the state of the system between code acquisition and code tracking, according to a search lock logic module as described in Section 3.2.4, and a decision device to locate the correlator which gives the largest output. During acquisition, the phase of the local PN sequence is shifted half a chip every PN code cycle by the clock shift module. When the system is in tracking mode, a pair of early and late correlators and a 16-bit subtractor are used to deter-

mine the phase difference of the incoming signal and the local PN code. The difference is converted by a digital to analog converter (DAC) to a control voltage for the VCO to adjust the master clock phase continuously. The details of the design of each sub-module of the decoder can be found in [10]. Some highlights of the design of the decoder are described in the following sub-sections.

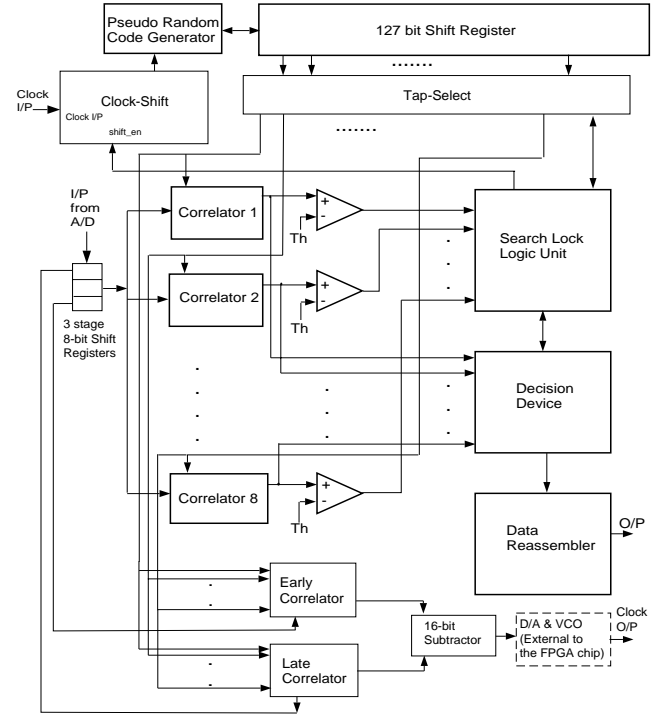


Figure 4: The CPSK Decoder

3.2.1 Digital Correlator with Dual Threshold Detector

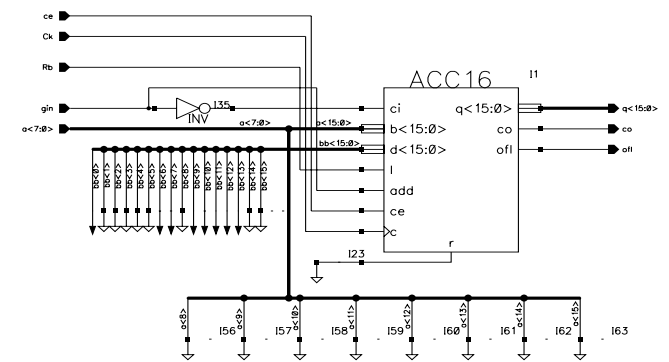


Figure 5: The Digital Correlator

Since a PN sequence consists of only +1 and -1 values, the digital correlation of the received digitized signal from the IF demodulator and the PN sequence is implemented by a 16-bit accumulator with increment and decrement capabilities. To save the available resource in the FPGA chip, the circuit is simplified by adding offset values to the accumu-

lator prior to correlation so that the handling of negative numbers is avoided. Consequently, the same offset is added to the threshold detector for proper detection. The correlator circuit is shown in Figure 5. The local PN sequence and the received signal are fed into gin and a<7:0>, respectively. A dual threshold detector is used to determine when the received signal is in synchronization with the local PN sequence. The major component in the detector is a 16-bit digital comparator which compares the correlated values with two threshold values, thu and thl, as calculated in [10]. Since the received signal from the IF demodulator has a 180° phase uncertainty, the correlation with the local PN sequence will give a value higher than thu or lower than thl when the signal is in synchronization with the local PN sequence, as shown in Figure 6.

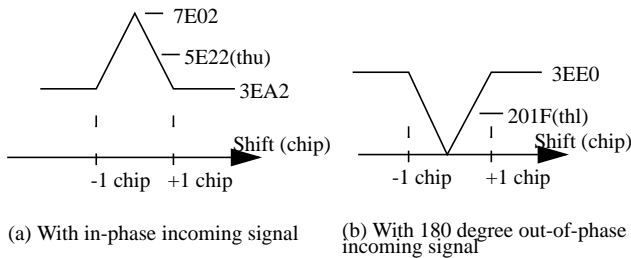


Figure 6: Correlation with Dual Thresholds

3.2.2 PN Code Generator with Shift Register

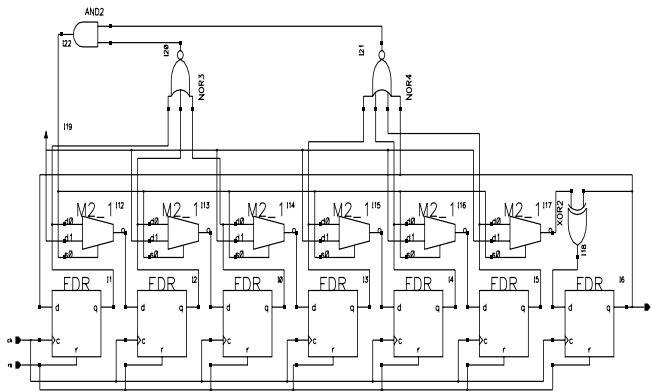


Figure 7: The 127 Chip PN Code Generator

To generate a 127-chip PN sequence, a 7-stage linear feedback shift register consisting of a delay line with taps that are modulo-2 summed and fed back to the first stage of the shift register is used. From [4], 7 different configurations of a 127-chip PN code generator are available. The one used in the CPSK decoder is one of the simplest type in that only the first and the last taps of the delay line are fed back to the first stage. To minimize the delay of the modulo 2 summers, a modular type generator is designed as shown

in Figure 7. By feeding the clock signal to input ck, the PN code is generated at the output q. Rs is the reset input to set q to zero when the decoder is reset.

In order to obtain different phase shifted versions of the local PN code, a 127 stage shift register is connected to the PN code generator to store the entire PN sequence and the required phase shifted PN codes are tapped from the outputs of appropriate shift register stages.

3.2.3 Tap Select Module The need of the tap select module is due to the fact that the phase shift between the first and the last code sequences in the signaling constellation is 15 chips while the other adjacent pairs of sequences are separated by 16 chips. When the CPSK decoder is in the acquisition state (state aq1 in Figure 9, as described in section 3.2.4), the received signal is fed to each of the 8 correlators to despread the signal with 8 different phase shifted versions of the local PN code that are tapped from the 127 stage shift register, such that the phase shift between the PN codes fed into any adjacent pair of the correlators is 16 chips. When the system first changes state to the tracking mode, the address or the location of the correlator which gives the largest correlation is stored as the reference. When the reference correlator is not correlator 1, the taps from the 127 bits shift register needs to be adjusted to make sure that the PN codes fed to the reference correlator and the one placed before it are separated by 15 chips while the codes fed to any other 2 adjacent correlators are phase shifted by 16 chips.

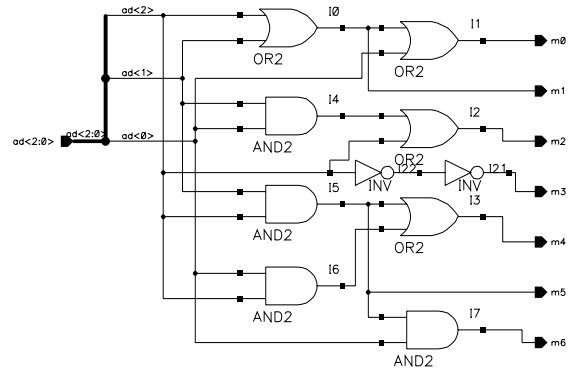


Figure 8: The Tap Select Module

As shown in Figure 8, outputs m0 to m6 of the tap select circuit are used to select the taps from the shift register. Inputs ad<2:0> come from the register that stores the address of the reference correlator. Since the phase difference between 2 adjacent PN codes is either 15 or 16 chips, 2:1 multiplexers controlled by m0 to m6 are connected to the shift register to select the codes with the correct phase shifts.

3.2.4 Search Lock Logic Module The search lock logic is implemented using a modified double dwell serial

search scheme as shown in Figure 9. The 4 possible states of the decoder are aq1, aq2, lock1, and lock2. The system starts in the first dwell state (aq1). The local PN code slides half a chip every cycle under the control of the clock shift module (section 3.2.5), until one of the 8 correlator outputs exceeds the predetermined threshold ($hit1 = 1$). The system then switches to the second dwell state (aq2) and correlates for another cycle. If, again, the output of the same correlator exceeds the threshold ($hit2 = 1$), the decoder is said to be synchronized and the system advances to the tracking state (lock1). Otherwise, the system switches back to the first dwell state (aq1), the acquisition process is repeated, and the clock shifts half a chip again after each PN cycle.

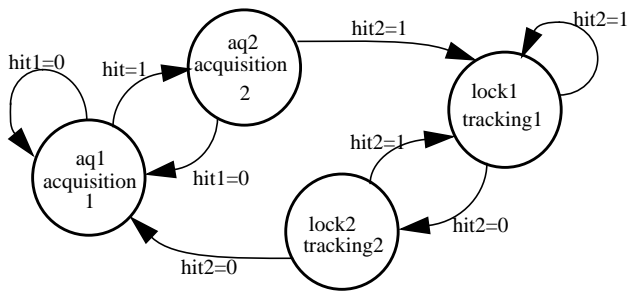


Figure 9: The Modified Double Dwell Serial Search Scheme

In the tracking mode (lock1), the master clock phase is adjusted continuously by the VCO and the threshold detector keeps on monitoring the outputs from the correlators. If none of them exceeds the threshold ($hit 2 = 0$), the system will go to the second lock state (lock2) and continue to decode data. The system will go back to lock1 if the next correlation exceeds the threshold ($hit2 = 1$). Otherwise, tracking is considered to be lost and the system switches back to the very first acquisition state (aq1). The state lock2 is inserted to reduce the chance of losing tracking in a very noisy environment.

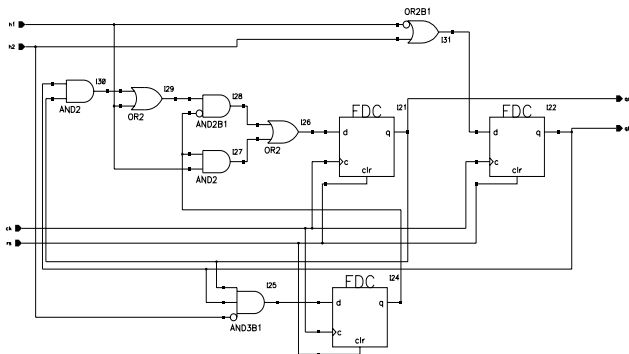


Figure 10: The Search Lock Logic Circuit

The schematic of the search lock logic is shown in Figure 10 and the output states are represented by the values of qa and qb as shown in Table 1.

Table 1: Representation of the Search Lock States

Search Lock State	qa qb
System Reset	0 0
aq1	0 1
aq2	1 0
lock1 or lock2	1 1

3.2.5 Clock Shift Module During the acquisition state (aq1), the local PN code needs to be advanced or retarded half a chip once every correlation cycle (127 chip duration) in order to slide the phase of the local PN code until one of its 8 phase-shifted versions matches with the phase of the incoming signal to within half a chip. This is done by shifting the phase of the clock signal for the PN generator by half a chip using the clock shift module. This module also provides timing signals for doing data comparison, storage, state change, and data clear in the decoder.

To achieve the half-chip shifting, a clock signal with rate equal to twice the required chip rate is fed to the clock shift circuit and an 8-bit counter is used to keep track of each correlation cycle. Input shift_en (as shown in Figure 11) determines whether the code phase is half-shifted. The resulting clock signal with half-chip shifting capability is obtained at output clk_out.

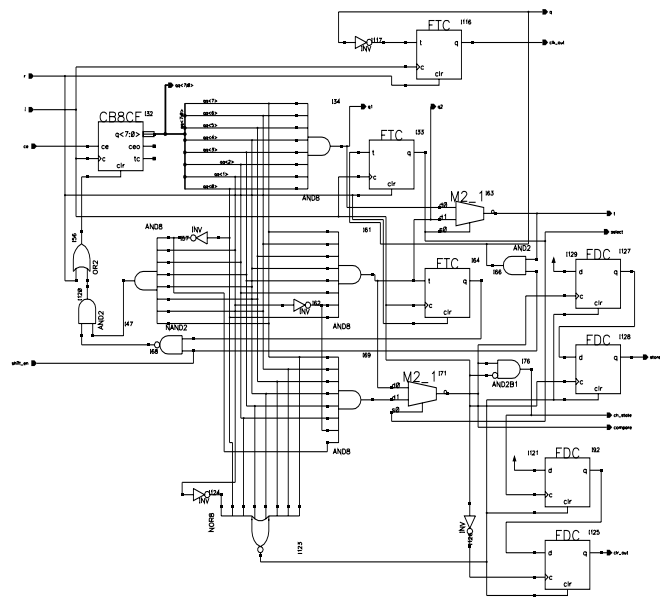


Figure 11: The Clock Shift Circuit

At the end of each correlation cycle, several tasks must be done sequentially before the data can be decoded correctly. The signals used to direct the tasks include compare, ch_state, store, and clr_out as shown in Figure 11. Compare is used to switch the thresholds for detecting out-of-phase incoming signal. Ch-state is used to change the state

of the search lock logic according to hit1 and hit2 (Figure 9). Hit1 is set when any of the threshold device output is 1 indicating that a high correlation is detected. Hit2 is set under one of the two conditions: 1) When the search lock state is in aq1, hit1 = 1, and the address of the reference correlator is equal to the address of the correlator which gives the highest correlation output at the end of the present correlation cycle. 2) When the search lock state is in either lock1 or lock2 and hit1=1. Store is used to store the reference address or location of the correlator that gives the highest correlation during acquisition mode and store the decoded data during the tracking or lock mode. Finally, the values stored in the correlators are cleared by the clr_out signal before the start of another correlation cycle. These tasks are completed in the last 2 chips (or 4 clock cycles) of each correlation cycle. Since the last 2 chips of each correlation cycle are not correlated, each correlated value is roughly decreased by 2/127 or 1.6% and this can be compensated by lowering upper threshold thu and raising the lower threshold thl by the same amount. Three outputs, namely q, t, and select, are used for testing purposes and are not connected externally.

3.2.6 Tracking Loop When the system first changes to the lock state (or tracking mode), the phase difference between the reference local PN code and that of the incoming signal may still be as large as half a chip. Also, both chip rates are not synchronized resulted in drifting of chips. A tracking loop is necessary to adjust the system clock to align the reference PN code phase with that of the incoming signal [8].

As shown in Figure 4, a pair of early and late correlators are used to calculate the correlation of the half-chip advanced and retarded versions of the incoming signal with the local PN code. The two versions of the incoming signal are obtained as follows. The incoming signal is sampled at twice the chip rate of the PN code and the samples through the ADC are fed into an 8-bit-3-stage shift register. Then, the late, punctual, and early versions of the incoming signal are obtained by the outputs from the first, second, and last stage of the shift register, respectively. The outputs from the pair of correlators are then subtracted, converted by a DAC, and fed to the master clock VCO to adjust its phase.

Since each symbol is decoded at the end of every correlation cycle, it is impossible to choose which of the eight phase shifted PN sequences at the beginning of each correlation cycle to correlate with the early and late versions of the incoming signal for evaluating the correct phase error. One way to get the correct phase error is to store the incoming signal for the duration of one PN sequence cycle. This approach would need $2 \times 127 \times 8 = 2032$ flip flops, an amount much larger than that available in the FPGA chip employed. To overcome this problem, a novel approach is

developed. We note that if the local PN code input to the pair of early and late correlators is fed any of the phase-shifted PN sequences other than the one corresponding to the currently received symbol, the phase error from the correlator pair is 0. Thus by summing the values of the 8 PN sequences and correlating the sum with the half-chip advanced and retarded versions of the incoming signal, the correct phase error is obtained to adjust the master clock.

A 4-input adder (Figure 12) is designed to sum up the values of the 8 PN sequences in two groups. Since each code value is either 1 or 0 (corresponding to actual value of +1 or -1, respectively), all the inputs to the 4-bit adder (a0, a1, a2, and a3) have a size of 1 bit and the maximum output is 4. To sum up all 8 PN code values, the outputs of two 4-input adders are summed by a 3-bit adder to yield a 3-bit number (y2 y1 y0). In the schematic, however, a 4-bit adder is used in place of a 3-bit adder because the 4-bit adder is provided by the CAD system as a macro. In this case it will be trimmed to a 3-bit adder automatically when the circuit is compiled for programming the FPGA.

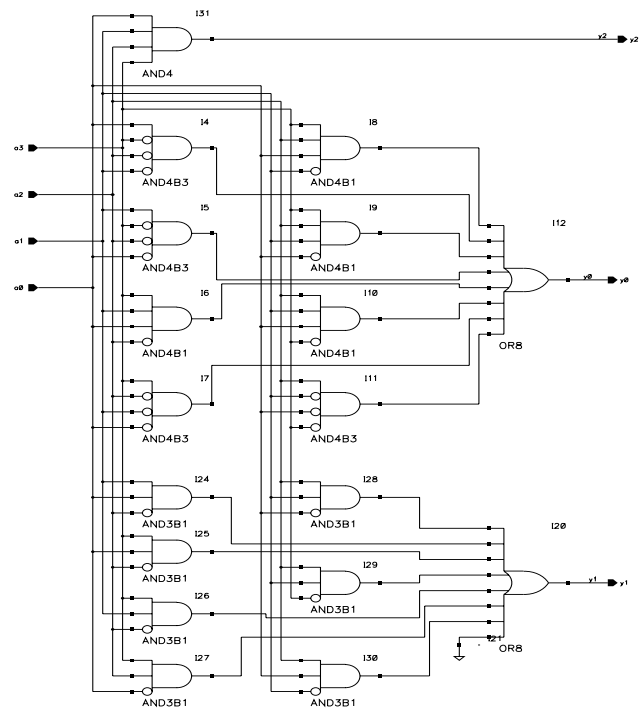


Figure 12: The 4-input Adder

The designs of the early and late correlators are similar to the ones used for decoding data except that the incoming signal is correlated with the summed value of the 8 PN sequences. A close examination of the PN codes reveals that the only possible values of the sum of the 8 PN sequences for any particular chip are 7, 6, 5, 4, 3, 2, 1 (corresponding to actual values of +6, +4, +2, 0, -2, -4, -6). Therefore, a module called 4_4_decode is designed to

select the correct multiplicand to the incoming signal for each chip. As shown in Figure 13, {a2 a1 a0} is the 3-bit input for the summed value of the 8 PN codes and s is the phase uncertainty of the incoming signal. The first output of the module, a0, is used to select whether the accumulation is an increment or decrement. The next four outputs, a1, a2, a3, a4, are used to activate the multiplicand to be multiplied to the incoming signal.

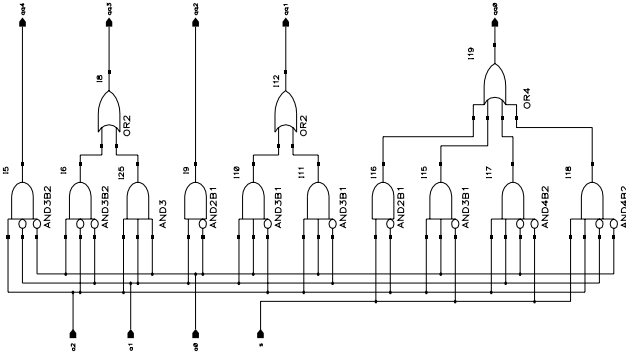


Figure 13: The 4-4-Decode Circuit

The schematic of the early correlator is shown in Figure 14. The late correlator is identical to the early counterpart except that the offset of the accumulator in the early correlator is much higher than that in the late correlator to eliminate the handling of negative numbers when taking the phase error value by subtracting the output of the late correlator from that of the early correlator. This phase error is fed into the analog section of the decoder for adjusting the master clock of the system.

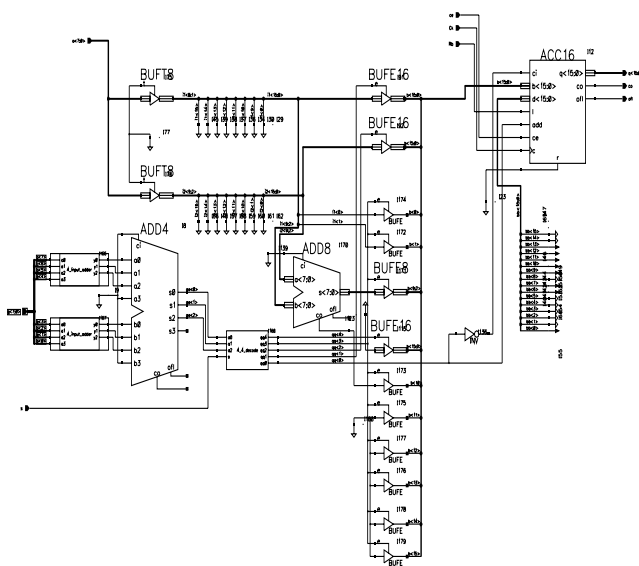


Figure 14: The Early Correlator

Finally, the full schematic of the CPSK decoder, without the analog section, is shown in Figure 15.

3.2.7 The Analog Section The major components in the analog section include an 8-bit DAC and a VCO. As shown in Figure 16, the phase error from the pair of early and later correlators are converted to an analog voltage to control the frequency of the VCO. The output of the VCO provides the master clock signal. Two potentiometers, R5 and R7, are used to adjust the control range and the center frequency generated by the VCO.

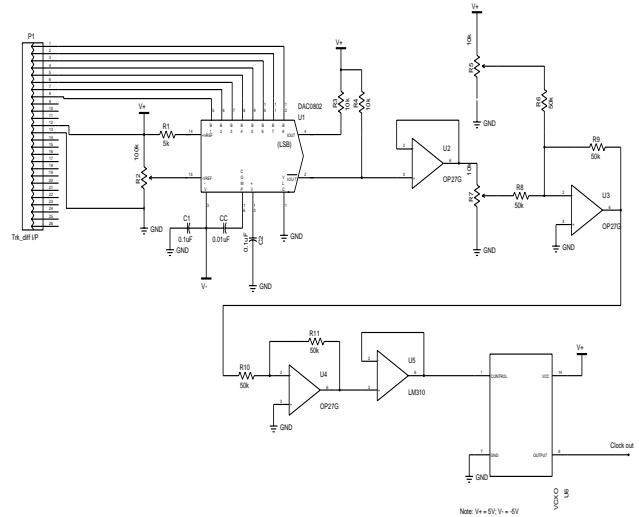


Figure 16: The Analog Section of the CPSK Decoder

The frequency range of the clock signal is calculated as follows. Assume that the nominal clock frequency is 16 MHz and the length of the PN sequence is 127 chips. To enable the system to go through the acquisition state to enter the lock state successfully (which requires at least the duration of 3 PN cycles), the clock signal should be stable enough so that the drifting is less than 1 chip in this duration. Thus the maximum drift of the clock signal with respect to the nominal value is $\pm 1/3$ chip in each PN cycle. At nominal frequency, the duration of a PN cycle is:

$$T_d = \frac{1}{16M} \times 127 \times 2 = 15.875\mu s \quad (3)$$

If the drift is $1/3$ chip less in 1 PN cycle, the chip duration becomes:

$$T_c = \frac{T_d}{127 - 1/3} = 125.329ns \quad (4)$$

Thus the frequency required to have 1 chip deducted is:

$$f_l = \frac{1}{T_c/2} = 15.958MHz \quad (5)$$

Similarly the frequency required to have 1 chip added to a PN cycle is:

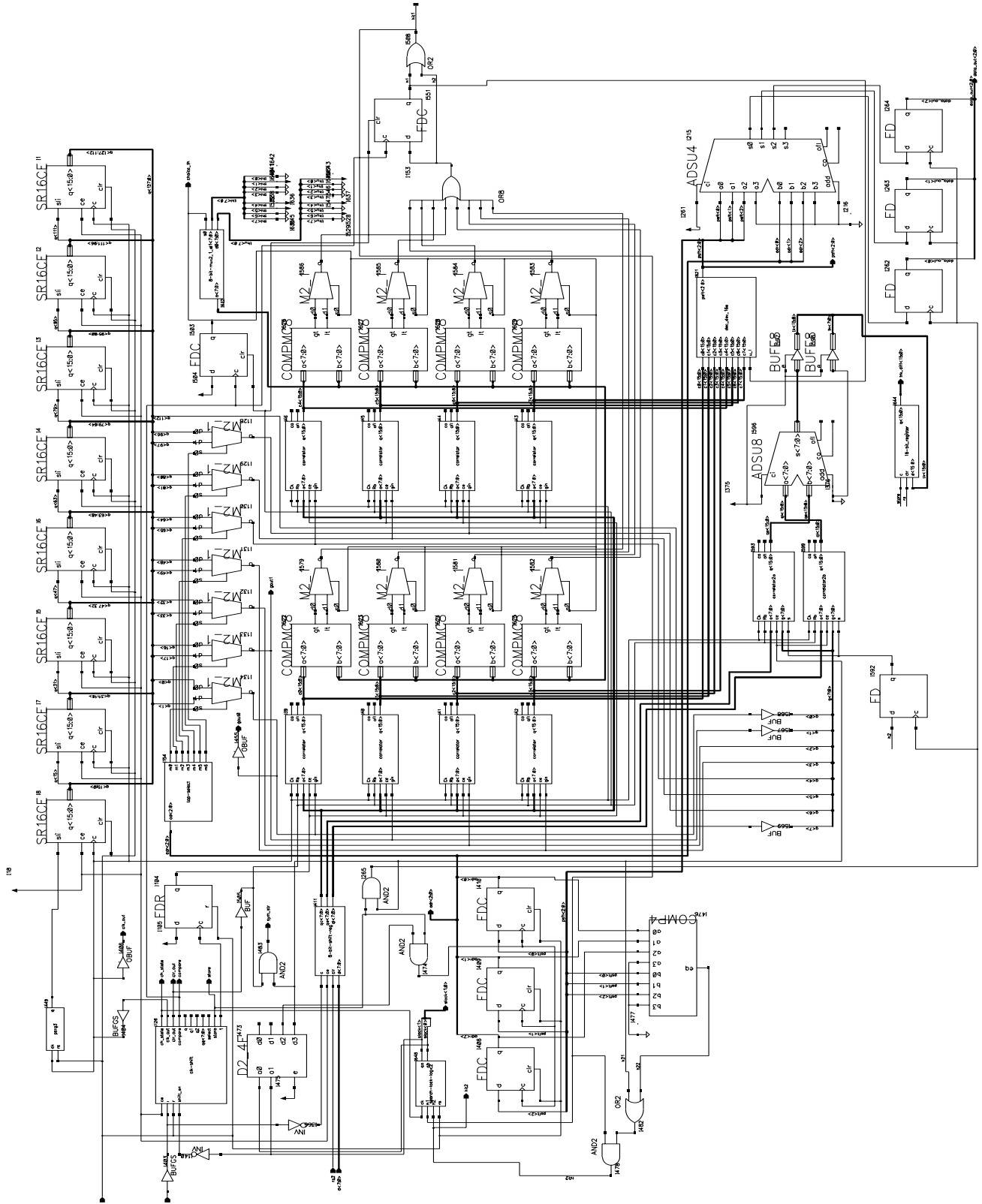


Figure 15: The CPSK Decoder

$$f_u = \frac{1}{T_c'/2} = \frac{1}{\frac{1}{16M} \times \frac{127}{127 + 1/3}} = 16.042\text{MHz} \quad (6)$$

Therefore, the maximum frequency control range is (15.958MHz, 16.042MHz).

It is important that the oscillator is stable enough such that the range of frequency variations is less than the range calculated above in order for the acquisition mode to work properly. Therefore, a VCO with frequency stability of 100 ppm and a control range of 200 ppm is employed to generate the master clock signal.

4. Performance Measurements

4.1 The Measurement Setup

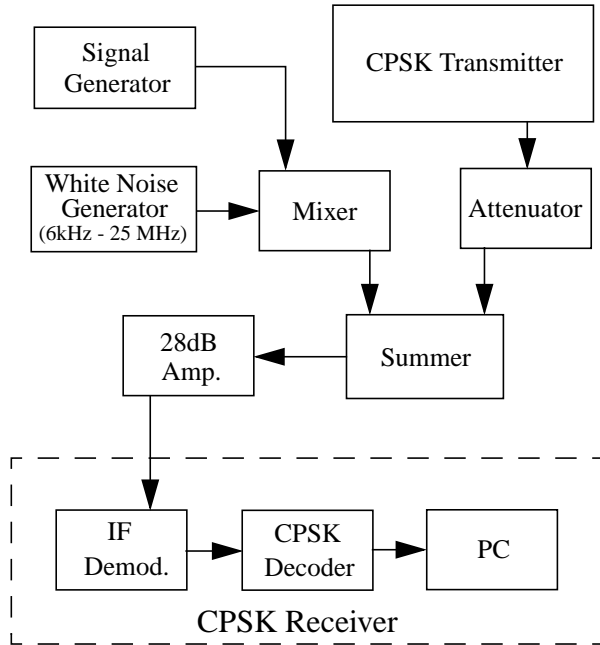


Figure 17: Experimental Setup

The setup for measuring the performance of the CPSK receiver is shown in Figure 17. The SNR of the signal is measured with a spectrum analyzer. To measure the BER or SER, pseudorandom data patterns are fed to the transmitter. The receiver output is read into a PC via the PC interface. The PC then operates like a BER analyzer to count the number of symbols that do not match the pre-defined pattern. To convert the SER to BER, the following equation is used:

$$BER = SER \times \left(\frac{2^{(M-1)}}{2^M - 1} \right) \quad (7)$$

To evaluate the performance with jamming from another transmitter, some modification on the setup is needed. An additional summer/splitter is inserted into the path between the 28 dB amplifier and the IF demodulator to mix in the signal from another CPSK transmitter.

4.2 BER Performance in AWGN

The measured BER performance of the receiver under AWGN is compared in Figure 18 with theoretical values calculated from (2). The error bar at each data point indicates the 95% confidence interval of the measurement. When the BER is high (in the order of 10^{-3}), the deviation of the measured values from the corresponding theoretical value is around 1.75 dB. As the SNR increases, the performance degradation gradually increases to about 2.2 dB at a BER of about 10^{-6} . The contributions to the degradation include the quantization of the received signal, the use of digital filters in the correlator, using an 8-bit DAC to control the VCO that supplies the master clock, and the instability of the Costas loop in the presence of noise.

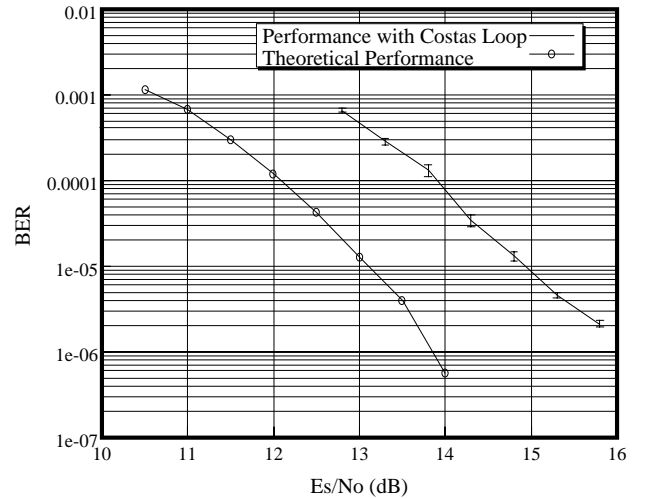


Figure 18: Performance of the Receiver in AWGN

4.3 BER Performance with CPSK Jamming

The BER performance is measured when the transmitter is sending pre-defined data patterns while the jammer is sending a purely zero data pattern. The carriers are generated by 2 separate RF generators. This makes the jamming a realistic situation where the transmitter and jammers are not synchronized. Four sets of measurements at SNRs of 12.8, 13.8, 14.8, and 15.8 dB were taken with different jamming levels and the respective BER curves are plotted in Figure 19. In general, the BER increases with the jamming level and as the SNR decreases, the system is more tolerant to the jamming signal. When the jamming signal is absent, the BER is around 2.5×10^{-6} at an SNR of 15.8 dB as shown in Figure 19. With the introduction of the jammer

with a jamming to signal ratio (JSR) of -8.5 dB, the BER is raised to 2.5×10^{-3} . Any further increase in JSR causes the decoder to lose synchronization. The substantial degradation of BER performance in the presence of CPSK jamming highlights the importance of using a good medium access control (MAC) protocol such as CSMA/CA [11] to mitigate interference.

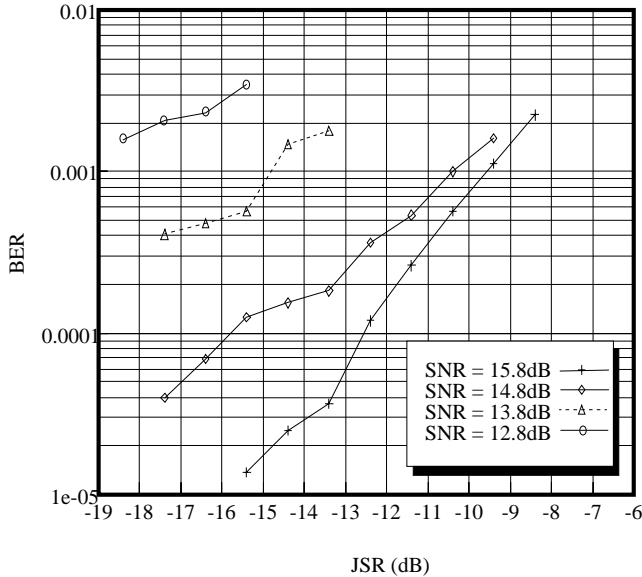


Figure 19: Performance with CPSK Jamming

5. Conclusions

We have presented the design, implementation and testing of a CPSK receiver consisting of an IF demodulator implemented with discrete components, and a CPSK decoder implemented in a FPGA chip. The implementation is highly efficient in that it consumes 98% of the function generators and 40% of the flip flops in the selected FPGA. The carrier phase synchronization problem is solved using a Costas loop in the demodulator followed by a double threshold detector in the CPSK decoder. A modified double dwell serial search scheme is employed for code synchronization, including acquisition and tracking. The decoder can be easily modified to decode data with different symbol size and PN code length by re-programming the FPGA chip. In the present design, to facilitate testing with the available transmitters, the decoder system clock frequency is limited to 2 MHz for a PN chip rate of 1 Mcps, which limits the data throughput to 23.62 Kbps. However, the data throughput can be boosted up to 6 Mbps using a data symbol size of 3 and PN code length of 15, with the FPGA running at its highest possible clock frequency of 60 MHz to give a PN chip rate of 30 Mcps.

Acknowledgment

This work was supported by the Science Council of B.C. through a Technology B.C. grant.

References

- [1] P.M. Corcoran, F. Papai and A. Zoldi, "User interface technologies for home appliances and networks", *IEEE Trans. Consumer Electronics*, vol. 44, pp. 679-685, Aug. 1998.
- [2] K.J. Negus, J. Waters, J. Tourrilhes, C. Romans, J. Lansford and S. Hui, "HomeRF and SWAP: Wireless networking for the connected home", *ACM Mobile Computing and Communications Review*, vol. 2, no. 4, pp. 28-37, Oct. 1998.
- [3] D.T. Magill, F.D. Natali and G.P. Edwards, "Spread-spectrum technology for commercial applications", *IEEE Proc.*, vol. 82, pp. 572-578, April 1994.
- [4] R. C. Dixon, *Spread Spectrum Systems with Commercial Applications*, 3rd. ed., New York, NY: John Wiley & Sons, Inc., 1994.
- [5] A.Y.C. Wong and V.C.M. Leung, "Code-phase-shift keying: a power and bandwidth efficient spread spectrum signaling technique for wireless local area network applications", in *Proc. IEEE CCECE'97*, St. John's, NF, pp. 478-481, May 1997.
- [6] R. Link and V.C.M. Leung, "Design of a DSP-based code phase shift keying modem for wireless local area network applications", in *Proc. IEEE VTC'97*, Phoenix, AZ, pp. 1069-1073, May 1997.
- [7] L. W. Couch, *Digital and Analog Communication Systems*, 4th ed., New York, NY: Macmillan Publishing Co., 1990.
- [8] M. K. Simon, *Spread Spectrum Communication, Vols. 1,2,3*, Computer Science Press, 1985
- [9] H Wang, "Spread spectrum modem IF receiver", Technical Manual, Board Revision 1.0, University of British Columbia, 1995.
- [10] S. Chan, "Design and implementation of a code-phase-shift keying spread spectrum receiver employing a FPGA baseband decoder," M.A.Sc. thesis, University of British Columbia, 1997.
- [11] K.-W. Chen, "Medium access control of wireless LANs for mobile computing", *IEEE Network*, vol. 8, no. 5, pp. 50-63, Sept./Oct. 1994.