An Ultra Low Cost Software Defined Radio Laboratory for Education and Research

Yair Linn*, Member IEEE

Abstract - In this paper we present an ultra-low-cost SDR (Software Defined Radio) laboratory that is based on a commercial-off-the-shelf FPGA (Field Programmable Gate Array) development board that is both inexpensive and available worldwide. The total cost of the laboratory is under USD$200, but yet includes complete transmission, channel emulation, reception (coherent and noncoherent) and probing capabilities. Over 15 different modulation types are currently supported. The same, unmodified FPGA board can be used for various undergraduate and graduate courses in digital logic, networking, embedded systems, and microprocessor design, and hence the incremental cost of using the same board for the proposed laboratory can thus theoretically approach $0. The laboratory is aimed primarily at facilitating senior undergraduate and graduate courses and projects, and allows students to get first-hand practical experience in SDR wireless modulation and demodulation techniques. The laboratory is nonetheless powerful enough to allow for SDR research projects. In this context, the laboratory is also particularly useful for universities in developing countries, where budgets are extremely limited as compared to those available in developed countries.

1. Introduction

The great Athenian philosopher Plato (427 BC - 347 BC) is credited with coining the expression "[…] necessity […] is the mother of invention"[1 p. 190]. That quote perhaps embodies the story of this paper, which was born more out of circumstance and necessity than of forethought and planning.

Having graduated in 2007 with a Ph.D. in Electrical and Computer Engineering at the University of British Columbia, Canada, the author accepted a visiting professorship at the Universidad Pontificia Bolivariana (UPB) in Bucaramanga, Colombia. The aim of this academic exchange, as is usually the case, was to enhance both parties' understanding of the other. Indeed, the differences in budgets and facilities can be quite striking, and the lessons learned quite humbling. It suffices to give as an example that the *entire* annual research budget of the electronic engineering faculty in the Colombian university was $20,000, which is the same order of magnitude as the budget of a single professor in any medium-sized university in North America. Government funding assistance in Colombia, though theoretically available, is practically almost impossible to obtain. Generally speaking, in some Latin American universities these problems are often compounded by an antiquated university cultural mindset which deemphasizes research funding in favor of administrative, political, or ceremonial activities which have little academic value.

Needless to say, the aforementioned budgetary constraints posed a significant challenge when the author wished to continue research in SDR. While software simulations could be done, there is no true substitute to hardware implementations and laboratory experimental verification. The only budget that could be allocated, at great effort, was approximately $1200, for the purchase of six Spartan-3A Starter Kit boards [2] (each costing less than $200) for a senior undergraduate course in FPGA design techniques taught by the author. It seemed like an impossible task, and a foolhardy endeavour, to attempt to construct a fully functional SDR laboratory using such an inexpensive board. And yet, surprisingly, this is exactly what transpired. Faced with this absurdly low development budget, this constraint forced the author to employ, and sometimes invent, extremely efficient Verilog programming techniques to cram the SDR algorithms inside the FPGA, resulting in several theoretical and practical breakthroughs, some of which have already yielded academic publications [3-5].

In this paper we shall describe the SDR laboratory developed by the author at UPB Bucaramanga. As noted, it is based upon a commercially available Spartan-3A Starter Kit board [2] which costs less than US$200. The only additional equipment required is a computer to interface to the board, and, optionally, a keyboard and VGA monitor to transform the lab into a completely autonomous entity. From the author's experience, the laboratory presented here is fully suitable to accompany undergraduate and graduate courses on SDR, and is powerful enough to be useful in academic research in SDR. Moreover, since the same unmodified FPGA board can be also used in courses for digital design, computer architecture, networking, and embedded systems, the investment needed for the laboratory can be reduced to a pittance, by spreading the cost among various courses and research projects. As seen from the authors' first-hand experience, this is essential in order to allow for SDR teaching and research in developing countries, where resources are significantly limited as compared to those available in developed nations.

2. Physical Platform

The laboratory's FPGA board is based on a 700,000-gate Spartan-3A Xilinx FPGA. A photograph of the board is in Fig. 1. In addition to the FPGA, the board contains the following:

• 50 MHz oscillator	• Stereo audio output
• 133 MHz oscillator	• 100-pin expansion connector
• External oscillator input	• 2x6-pin expansion connectors
• Two 16 Mbit serial FLASH	• 8 LEDS
• 32 Mbit parallel FLASH	• Rotary knob/push-button
• 4-channel D/A converter	• 4 push-buttons
• 2-channel A/D converter	• 2-line x 16-character LCD
• Analog amplifiers	• 4 switches
• Ethernet port	• PS/2 keyboard connector
• VGA connector	• Two RS-232 ports

The board also has a USB port, but unfortunately it is not available for user applications. For additional details on the FPGA board the reader is directed to [2]. The abundance of peripheral components, the FPGA's decent logic capacity, and the expansion capabilities make this board very suitable for courses in FPGA

*The author has a Ph.D. from the University of British Columbia, Canada. He is currently a Visiting Professor at the Universidad Pontificia Bolivariana in Bucaramanga, Colombia, Tel: +57(7)6796220, e-mail: yairlinn@gmail.com.

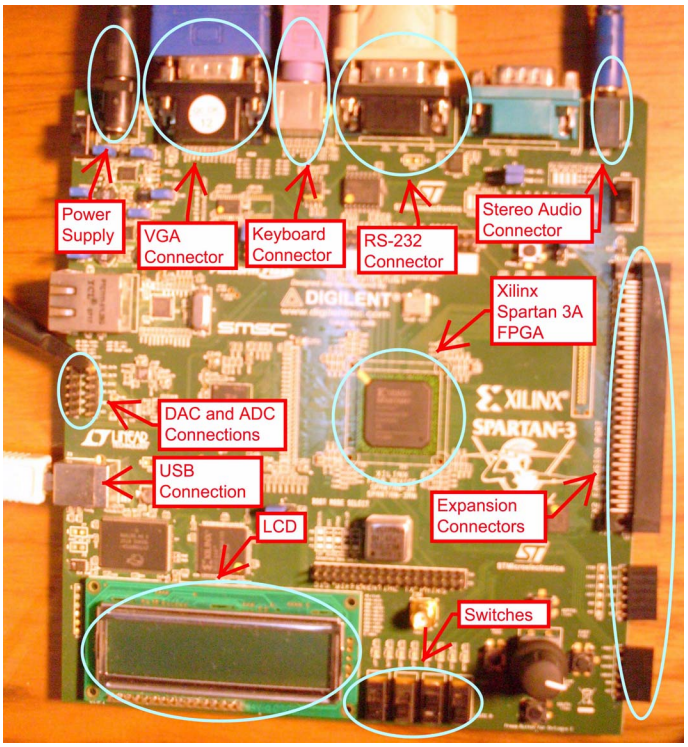

Fig. 1 –Spartan 3A Starter Kit upon which the SDR lab is implemented.

design, digital logic, computer architecture, networking, and more. For example, the author has used this board to teach an advanced undergraduate FPGA design course that was mentioned earlier.

For the laboratory presented here, the FPGA and some of the peripheral devices were used in order to form a transmission, channel emulation, and receiver system which comprise a complete communications system. Other components on the board are used for input, output, probing, and control, in order to enhance the learning experience. This is now explained in detail.

3. Implementation and Structure

The laboratory is implemented via a configuration file that is loaded into the FPGA. Additionally, the FLASH memories on the board are loaded with data for lookup tables, data, and parameters that are used in the transmission, reception, and channel emulation.

3.1. Internal Structure

A simplified functional diagram for the SDR implementation within the FPGA is shown in Fig. 2. Essentially, the laboratory includes a data sequence generator, a modulator, a channel emulator (Gaussian noise addition, slow fading can also emulated), and a demodulator (can be configured as coherent or differential). The demodulator includes BER (Bit Error Rate) and SNR (Signal to Noise Ratio) measurement circuits. The modulation/demodulation combinations currently supported are BPSK, QPSK (4-QAM), 8-PSK, 16-PSK, DBPSK, DQPSK, D8PSK, D16PSK, QAM-16, QAM-64, QAM-256, $\pi/4$-QPSK, $\pi/8$-8PSK, $\pi/4$-DQPSK, and $\pi/8$-D8PSK. It is emphasized that all of the above modulations are contained in a single FPGA configuration. That is, no FPGA reconfiguration is necessary in order to change the modulation, but rather only a user command – i.e. it is a true SDR. Work is ongoing to extend the modulations even further to MSK, OQPSK, and FSK, and to include a root-raised-cosine pulse-shaping option (currently the baseband pulse shape is rectangular).

Table I – Summary of Current SDR Lab Parameters

Parameter Name	Value	Notation
Modulations/ demodulations	BPSK, QPSK (4-QAM), 8-PSK, 16-PSK, DBPSK, DQPSK, D8PSK, D16PSK, QAM-16, QAM-64, QAM-256, $\pi/4$-QPSK, $\pi/8$-8PSK, $\pi/4$-DQPSK, $\pi/8$-D8PSK	
Symbol Coding	Differential coding, Gray mapping	
Demodulation	Coherent or Differential	
Carrier Frequency	5 KHz	f_c
Symbol Rate	625 Hz	1/T
Sampling Rate	100 KHz	f_s

The current parameters of the laboratory are shown in Table I. Though these parameters are completely changeable by the user, and could be chosen to be much higher, the choice of such relatively low rates has advantages from a laboratory and teaching perspective. First, the low carrier frequency (5 KHz) while certainly well below the maximum rate achievable with the FPGA, provides a signal that is well suited for measurement with low-cost spectrum analyzers and oscilloscopes that use a computer's audio input (which has filter which cuts off signals above 20 KHz). This allows a student to use the laboratory in his or her own home, without the necessity of having access to a spectrum analyzer or oscilloscope (which, especially in developing countries, are scarce). Very powerful spectrum analysis and oscilloscope freeware programs that use the PC's audio input, such as Visual Analyzer (www.sillanumsoft.org) should be quite adequate for most academic needs. Thus, choosing low operating frequencies allows the student or researcher to use very low-cost measurement apparatus, which is essential if the student is to be allowed to take the laboratory home for autonomous study, or if the university has limited resources, as is the case in developing countries.

3.2. Carrier and Symbol Synchronization PLLs

The laboratory supports coherent and differential demodulation for various modulation schemes[1]. Therefore, a carrier PLL and a symbol PLL are implemented within the FPGA, where the carrier PLL is only applicable for the coherent demodulation modes. The receiver structure generally follows the all-digital receiver structure detailed in [7 Chap. 2-5]. More specifically, the receiver architecture follows the Linn architecture detailed in [5], which will now be briefly discussed in the context of this receiver.

The symbol timing recovery PLL is based on [8], and linear interpolation is chosen for the resampler in order to conserve FPGA resources (although it should be noted that a parabolic Farrow interpolator [9] was implemented successfully and incorporated in the receiver for experimentation purposes). The post-matched-filter sampling rate is user selectable between the values of 10 samples/symbol, 5 samples/symbol, and 2.5 samples/symbol, all of which are adequate in order to minimize degradation due to the linear interpolation used, as detailed in [9 Table V]. The timing error detector is that of [10], whilst the symbol timing recovery PLL lock detector is that of [11, 12].

The carrier synchronization PLL's phase detector is based on the structures investigated in [13]. The carrier lock detector is the Linn-Peleg detector taken from [14, 15]. There are two on-board real-time SNR estimators: the SNR estimator of [14, 16], which

[1] Note that the transmitted symbols are always differentially encoded even if in the receiver they are demodulated coherently, in order to resolve the inherent carrier ambiguity [6 p. 203-205] of the carrier recovery PLL.

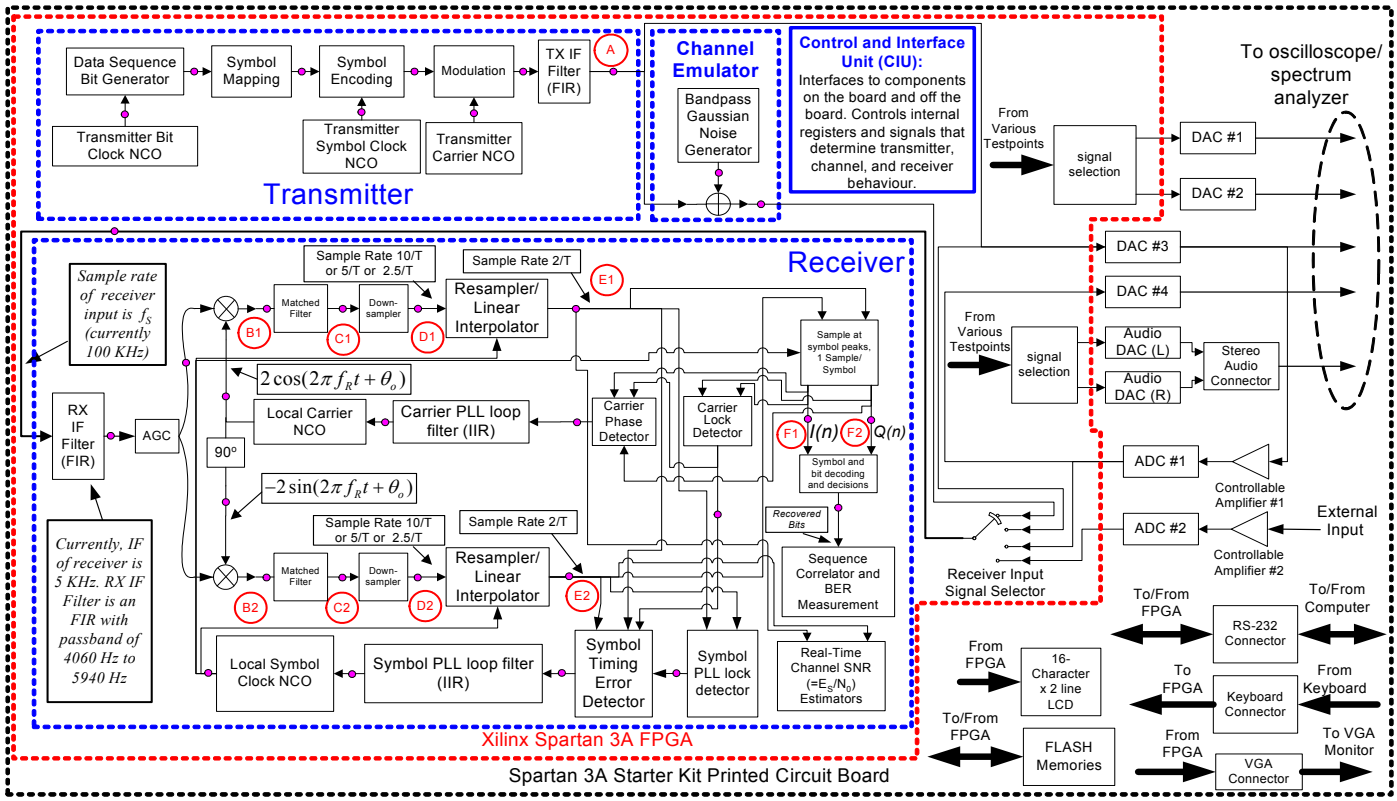

Fig. 2 – Simplified diagram of the SDR laboratory. Small magenta filled-in dots in the various paths represent some of the possible test points that can be fed out to the DACs. The points labeled (A), (B1), (B2), (C1), (C2), (D1), (D2), (E1), (E2), (F1), and (F2), are discussed later in the paper. ADC = Analog to Digital Converter. AGC = Automatic Gain Control. DAC = Digital to Analog Converter. FIR = Finite Impulse Response. IF =Intermediate Frequency. IIR = Infinite Impulse Response. NCO = Numerically Controlled Oscillator. SNR = Signal to Noise Ratio (=E_S/N_0, the channel symbol SNR). Note that a second AGC loop (not shown) controls the signal levels after the matched filters in order to further minimize quantization effects.

requires carrier lock and hence is suitable only for coherent demodulation, and the one proposed by Linn in [17], which does not require carrier synchronization and is thus suitable for SNR estimation for the differential demodulation modes. The architecture of the PLLs' loop filters is detailed [18, 19], and the design methodology of the PLLs is detailed in [20]. Many of the structures used in the carrier and symbol PLLs use lookup tables which are stored either on internal FPGA BlockRam, internal FPGA distributed ROM, or in the FLASH memories on the board (which are read in real-time by FSMs (Finite State Machines) inside the FPGA).

3.3. Channel Emulation

The channel emulation part of the laboratory is based on a novel method of bandpass Gaussian noise process generation that is detailed in [3]. Accurate Signal-to-Noise Ratios (SNRs) can thus be generated on the FPGA board without the need for external noise sources. Paper [3] is an example one of the research breakthroughs that were made possible to devise, develop, and test by using this laboratory.

3.4. Probing and DAC/ADC Interface

The filled-in small magenta circles on the various paths in Fig. 2 are some test points that can be channeled to the various on-board DAC (Digital to Analog Converters) for observation in a spectrum analyzer or oscilloscope. This allows the user to fully observe, in real-time, the internal signals in the transmitter, channel, and receiver.

Since a controllable bandpass noise process is generated on the card, and since the modulated signal amplitude is also user-controllable, then a wide range of SNRs can be generated on-board without the need to transmit the signal over the air; hence, no external circuitry (such as amplifiers or antennas) are needed to generate SNRs. Nonetheless, if desired, the transmitted signal is routed to DAC #3; and optionally this signal can be fed-back through an on-board controllable amplifier and ADC on the card. If a more elaborate setup is desired, the signal can be upconverted to RF, passed to a transmit antenna and then received by a receiving antenna, downconverted back to IF, and then connected to the on-board controllable amplifier and then to the ADC. Two cards can also be connected to each other as a TX-RX symmetric pair. However, again, such complicated setups are unnecessary since a wide variety of SNRs can be generated using the on-board noise and signal amplitude controls, i.e. one card can serve as the entire TX-RX chain without need to pass to the analog domain.

It is worth noting that the low-speed stereo audio DAC (see Fig. 2, the Left and Right audio channels) has the great advantage that the voltage levels and connector form make it ideal for connecting to a PC's audio input. High quality PC-Based spectrum-analyzers and oscilloscopes, which are (as mentioned earlier) available either freely or for a modest sum, can then be used for signal measurements.

3.5. Command and Control of the FPGA card

The FPGA card can be controlled through various means. A hyperterminal connection via the RS-232 port is sufficient, though requires low-level knowledge of the FPGA's internals in order to

change the SDR's parameters. A Matlab-based GUI (Graphical User Interface) is being developed in order to make the control more user-friendly. Signals can be routed not only to the DACs, but also to the Matlab interface via 4 FIFOs inside the FPGA that work as a 4-channel simultaneous capture circuitry; the transfer from the FIFOs to the Matlab GUI is done via the RS-232 port (as is all contact between the Matlab GUI and the card).

Another control method is directly via a keyboard connected to the on-board PS/2 connector. This is a simple, text-based interface, which is equal to the RS-232 command structure except that the commands are entered by the keyboard and not through the serial port. The user can switch at will between RS-232 and keyboard-based command entry.

Output from the card can be displayed at the following: (1) via the Matlab GUI on the controlling computer's screen (including graphical information obtained via the FIFOs); (2) text data on a hyperterminal on the controlling computer screen; (3) text data on a simple monochrome character-based VGA display connected directly via the on-board VGA connector; (4) very basic information on the on-board 2-line LCD.

Therefore, the card can be used in a completely autonomous manner, whereby control is achieved via direct connection to the card's keyboard port and VGA port. More robust and used friendly control is achieved through the Matlab interface connected via the RS-232 connection.

The VGA port, RS-232 protocol, and LCD are controlled by three separate embedded soft-core PicoBlaze microcontrollers (see www.xilinx.com/picoblaze) in the FPGA, which are extremely area-efficient yet sufficiently powerful embedded microcontrollers.

3.6. Data Sources

A variety of transmitted bit sequences can be chosen by the user, including Pseudo-Random Bit Sequences (PRBS) generated by Linear Feedback Shift Registers [21]. Such sequences are ideal for BER measurement in order to characterize the communication link's performance [21]. Various long and short PRBS sequences are available, with the tradeoff being that a longer PRBS sequence approximates random data better, but results in a longer time for the BER measurement circuitry to achieve lock. Also, user-chosen patterns can be transmitted.

As a didactic tool, for example, we can transmit the sequence ('01000111'). This sequence is quite short and it cannot approximate random data to any sufficient degree; however, this sequence allows the user to easily visualize errors using the VGA connection or the hyperterminal connection, and thus it is very useful as a didactic tool. Such usage is shown in Fig. 3.

4. Academic Teaching Example

In this section we give examples of typical usage of the lab in a university setting in the teaching of wireless communications. As a representative case we show how QPSK transmission and reception can be observed using the laboratory. The baseband complex modulation signal is $m(t) \triangleq \sum_{n=-\infty}^{\infty} \frac{1}{\sqrt{2}}\left(a_n + j \cdot b_n\right) p(t-nT)$, where $1/T$ is the symbol rate, $p(t) = \{1 \quad -\frac{T}{2} \le t \le \frac{T}{2},\ 0 \quad \text{otherwise}\}$, and the differentially gray coded data symbols are (a_n, b_n) with $a_n, b_n \in \{-1, 1\}$. The modulated signal is

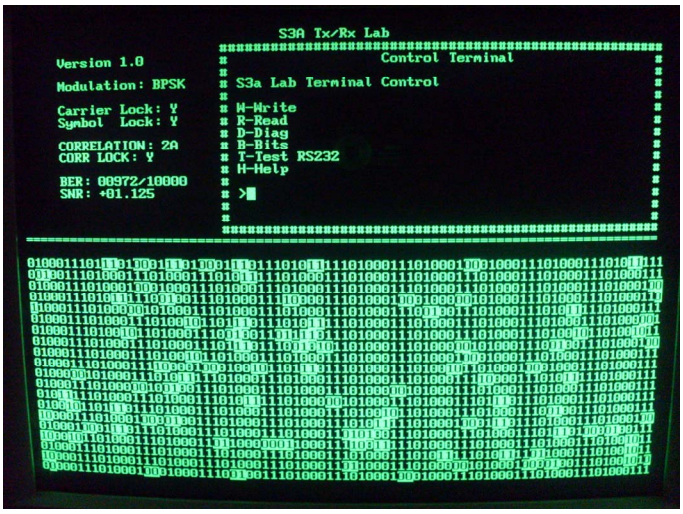

Fig. 3 – Screenshot of the control terminal with an SNR of 1.125 dB and the transmitted signal is the word 01000111. On the screen's top half we see on the left various receiver status metrics and on the right the control terminal. The screen's bottom part is a running display of the received bits, with erroneous bits denoted with inverse color. In this example, differentially encoded coherently demodulated BPSK is the modulation. Note that errors tend to occur in *pairs* of bits, which is expected due to the differential encoding and decoding [22 Sec. 4.2].

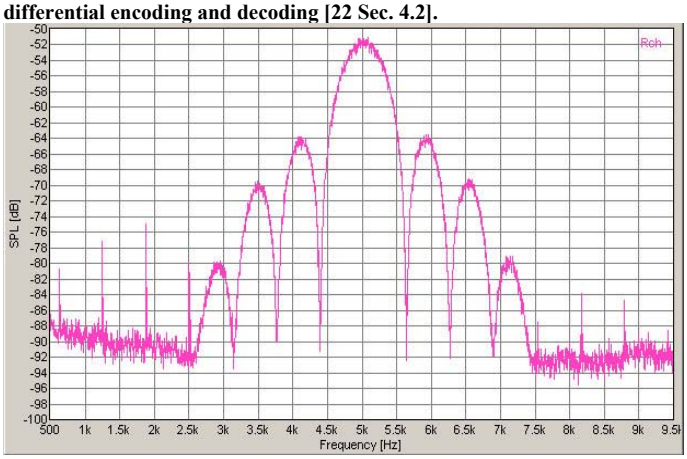

Fig. 4 – Modulated QPSK signal after transmission filter.

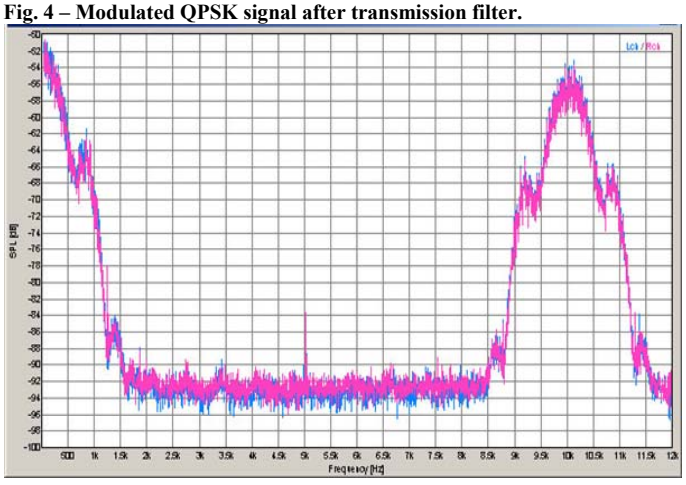

Fig. 5 – Spectrum of *I* channel (blue) and *Q* channel (cyan) at the output of the I-*Q* demodulator (before the matched filter) when the carrier is locked. Symbol SNR is 12 dB (the spectrums essentially overlap).

$s_m(t) \triangleq \mathrm{Re}[m(t)\exp(j2\pi f_c t)]$ with f_c being the carrier frequency. As per Table I, with current lab parameters we have $T = 1/625 = 1.6$ mSec and $f_c = 5000$ Hz. The modulated post TX IF filter QPSK signal spectrum is shown in Fig. 4, which was obtained by channeling point (A) in Fig. 2 through an audio DAC to a PC-based spectrum analyzer. The time-domain QPSK signal can also be observed via an oscilloscope (omitted due to space constraints).

At the output of the receiver's IF filter the waveform is: $s_R(t) \triangleq s_m(t) \otimes f_T(t) \otimes f_{IF}(t) + n(t)$ where $f_T(t)$ and $f_{IF}(t)$ are the impulse response of the transmission and receiver IF filters, "$\otimes$" is convolution, and *n(t)* represents the bandpass noise. Note that the receiver's IF filter is narrower than the TX filter and only lets the main lobe and one half of the first sidelobe to pass, so that ~95% of the received input signal power is passed to the receiver.

Continuing the signal chain at the receiver, the received IF signal is passed through the *I-Q* demodulator. The output of the *I-Q* demodulator is present in Fig. 2 at points (B1) and (B2). In coherent QPSK mode with the receiver's carrier locked, in Fig. 2 we have $f_R = f_c$ and $\theta_o = 0$. If the equations are developed (omitted here due to space constraints) it can be shown (e.g. [23 Chap. 2], [22 Chap. 4]) that the *I* and *Q* channel (points (B1) and (B2) in Fig. 2) will contain the recovered *I* and *Q* baseband data spectrum and also a modulated signal at <u>double</u> the carrier frequency (10 KHz). The spectrums of these signals are shown in Fig. 5, while time-domain oscilloscope screenshots are shown in Fig. 6. On a large timescale (Fig. 6, top), we can see the demodulated, pre-matched filter rectangular shape silhouette of the transmitted bits. As we zoom in (Fig. 6, middle and bottom) we can clearly see the 10 KHz double-carrier component. Note how these graphs coincide with the theory (e.g. [23 Fig. 2.11]). Already, students can see with these spectrums and time-domain graphs how the concepts of modulation, filtering, noise, and *I-Q* demodulation manifest themselves in the real world, and can acquire valuable experience in measuring and interpreting spectrum and oscilloscope measurements.

The matched filters in the *I* and *Q* arms will eliminate the double-carrier-frequency terms, and the *I* and *Q* arm after the matched filter will be composed of triangular pulses (which is the post-matched-filter pulse shape). This can be seen in Fig. 7, obtained by channeling points (C1) and (C2) through DACs to an oscilloscope. In that figure, a time-lapse mode of the oscilloscope is used, so that the so-called eye-diagram can be discerned.

Fig. 8 contains the post-matched-filter post-downsampling recovered baseband symbol stream, sampled (for this example) at 2.5/*T* complex samples per symbol. This is shown in Fig. 8, obtained by channeling points (D1) and (D2) through DACs to an oscilloscope. The signals are then interpolated and resampled at a rate of 2/*T* for the symbol timing recovery loop [8-12]. These resampled signals are shown in Fig. 9, obtained by channeling points (E1) and (E2) through DACs to an oscilloscope. Note that even though a noiseless signal enters the receiver for the graphs in Fig. 8 and Fig. 9, some noise is seen there, due to a combination of: (1) quantization noise; (2) jitter in the carrier and symbol PLLs (due primarily to quantization noise and oscillator phase noise); (3) DAC imperfections; (4) oscilloscope measurement imperfections.

If we downsample the *I* and *Q* signals further to a rate of 1/T, as

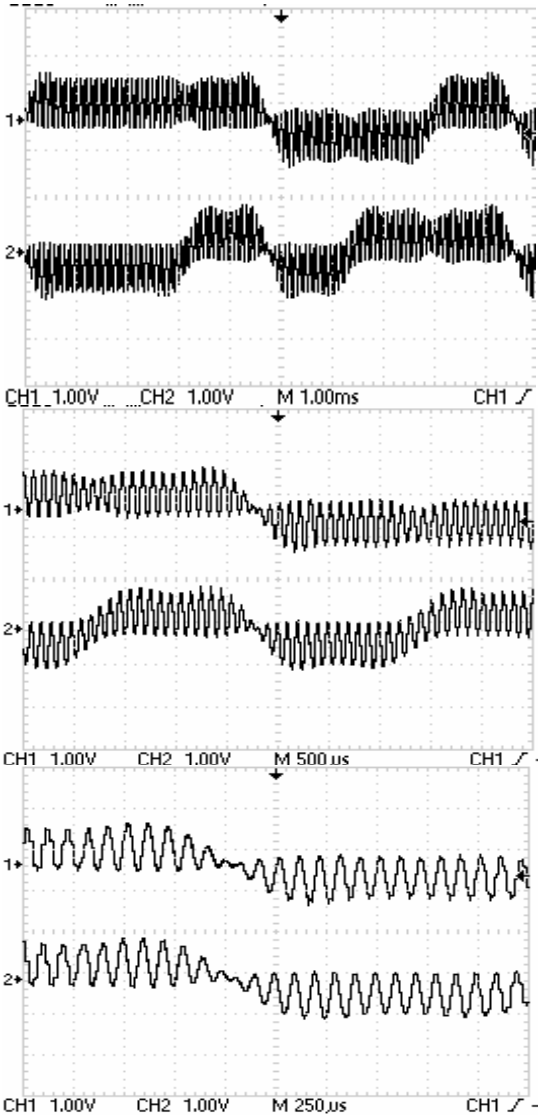

Fig. 6 – *I* and *Q* time domain waveforms of a received QPSK signal after the *I-Q* demodulator. Carrier is locked, no noise is added (i.e. SNR = ∞). Top: 1 mSec/div. Middle: 500 µSec/div. Bottom: 250 µSec/div. *I* is scope channel 1, while *Q* is scope channel 2.

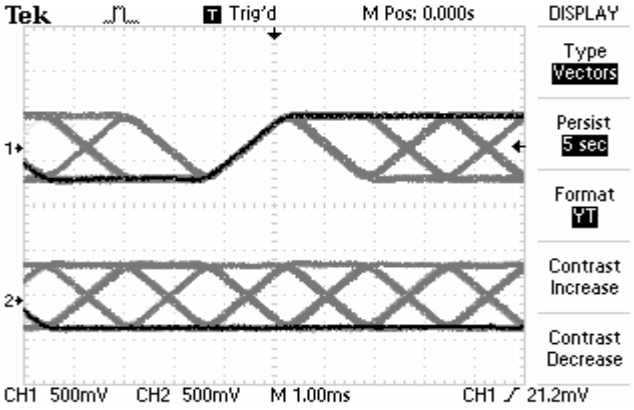

Fig. 7 – Post-matched filtered QPSK waveforms after matched filtering before downsampling. Time-lapse nature of the measurement allows the eye diagram to be observed. *I* is scope channel 1, while *Q* is scope channel 2.

needed for the carrier loop, where the samples are at the symbol peaks, then we should see the well-known *I-Q* graph for QPSK if the (I,Q) signals are fed to an oscilloscope in X-Y mode. This is shown in Fig. 10, obtained by channeling points (F1) and (F2) in Fig. 2 through DACs to an oscilloscope.

In the above paragraphs we have thus seen how the complete modulation and demodulation process can be observed by channeling the appropriate signals to the DACs and then to spectrum analyzers or oscilloscopes.

Finally, BER and channel symbol SNR measurement circuits are implemented in the FPGA, and by noting their values (for example, by using the VGA interface (see Fig. 3) or by using the RS-232 interface), we arrive at BER-vs.-SNR curves, which is an essential skill for the wireless engineer to master. An example of such a graph made using this laboratory is given in Fig. 11. In that figure, we plot the BER for differentially coded Gray-mapped M-PSK which is coherently demodulated, commonly known as DEMPSK (Differentially Encoded (coherent) M-PSK) [22 Chap. 4]. The theoretical BER formulas can be found in [22 Chap. 4, 24 Sec. 5.2.7].

5. SDR Research using the Lab

Although the lab is primarily geared towards teaching and projects for undergraduate and graduate students, it nonetheless is sufficiently potent for it to be used in academic research. For example, to the author's knowledge the receiver in this laboratory is the first documented hardware implementation of some of the structures in [3-5, 10-14, 16-20, 25-27]. Moreover, it can serve as a platform for investigating the performance of these and other structures. It could be argued that it is a true feat to be able to insert into a Spartan 3A, a relatively inexpensive and small FPGA, such complicated SDR circuitry, including transmission and reception (for multiple modulations), and channel emulation modules, and that this in itself is a significant research achievement. Moreover, since the FPGA is reconfigurable, the laboratory could potentially permit other researchers to use it in order to rapidly implement and characterized their own structures in hardware. Thus, the laboratory is useful as a research tool as well as a teaching platform.

6. Miscellaneous Results and Experiments

In this section we briefly show various graphs obtained during work by the author on the lab. The graphs, along with brief explanatory captions, are shown after the references, in Fig. 12-Fig. 28. It can be seen that a wide variety of experiments can be carried out using the laboratory. Nothing is needed except a standard oscilloscope and a (computer-audio based) spectrum analyzer. Such experiments include, but are not limited to:

- Investigating the nonlinear behavior of the synchronization PLLs, including pull-in and cycle-slip behavior.
- Investigation of the cross-interaction of the various PLLs and AGC control loops.
- Investigating the interaction between PLL performance and BER.
- Investigating and comparing waveforms throughout the transmitter's and receiver's signal chains for various modulations and signal-to-noise ratios.
- Optimization of receiver and PLL parameters.
- Eye-diagram measurements.
- BER and SER performance comparisons between modulations.

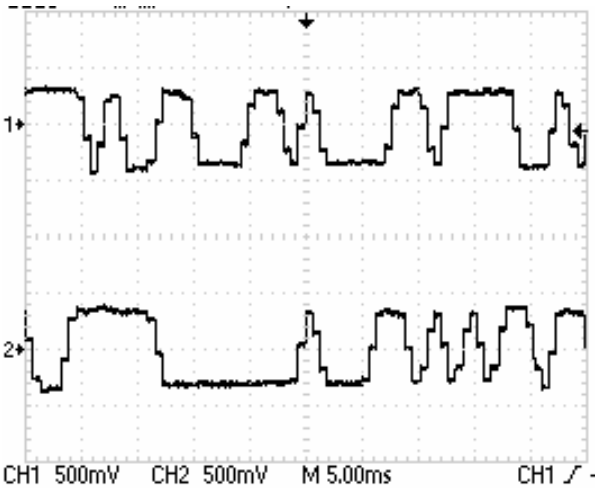

Fig. 8 – Post-matched-filter *I-Q* waveforms after downsampling, sampling rate is 2.5 complex samples/symbol. Carrier is in lock. *I* is scope channel 1, while *Q* is scope channel 2.

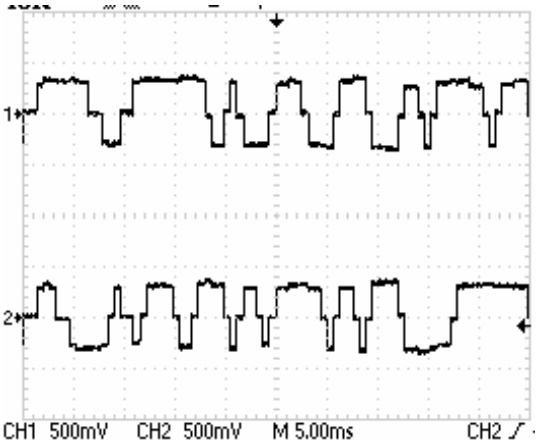

Fig. 9 – Post resampler/interpolator *I* and *Q* waveforms for QPSK when the carrier and symbol PLLs are locked, noiseless input to the receiver (SNR = ∞). The channels are sampled at a rate of 2/T. As can be seen, for each successive pair of samples, one sample is on the peak of the symbol, while the other is midway between the current symbol and then next one. Stair-like waveform is due to the fact that the DAC is of the ZOH (Zero Order Hold) type. *I* is scope channel 1, while *Q* is scope channel 2.

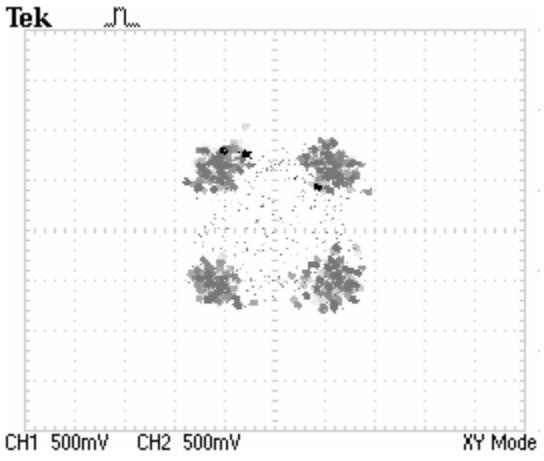

Fig. 10 – QPSK demodulated *I-Q* graph for a symbol SNR of 12 dB.

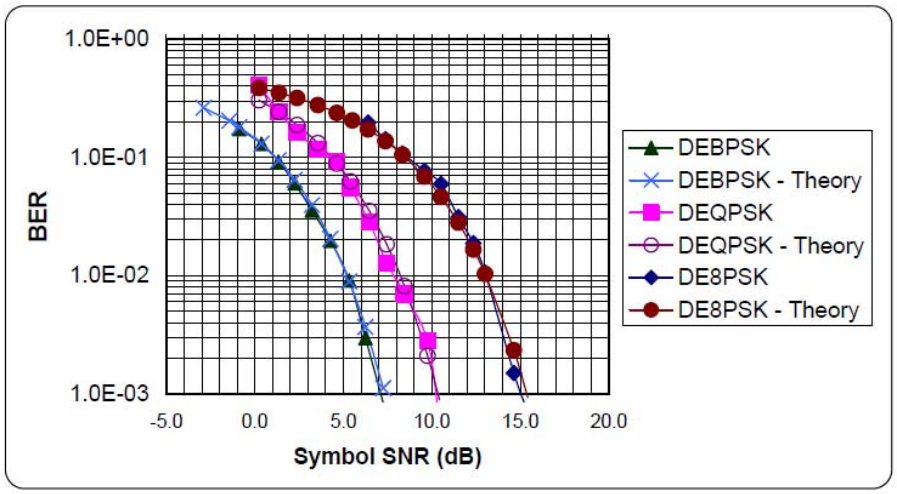

Fig. 11 - Theoretical vs. Measured results for DEMPSK reception. Graph is of BER versus E_S/N_0.

- Invention and investigation of new modulation and demodulation schemes. These could include exotic constellations that have limited practical use, but could be valuable as a theoretic teaching tool (e.g. for BER calculations, synchronization technique exercises, etc.). For example, we show the constellations $\pi/4$ - BPSK, $\pi/4$ - QAM-64: see Fig. 27-Fig. 28.
- Investigation of cognitive radio algorithms.
- Experimental reinforcement of various theoretical concepts learned in class.

All of these experiments can be carried out with a very minimal budget and even within a student's home as take-home experiments. Hence, this laboratory can be an important part of digital communications curricula for senior undergraduate and graduate work in universities around the world.

7. Conclusions

This paper presented a ultra low cost SDR laboratory based upon an inexpensive FPGA-centered board which costs under US$200. The cost of the laboratory can be further reduced if the cost of this general-purpose FPGA card is spread out among several courses or projects for which this board is suited. The laboratory allowed easy probing and control of internal signals and parameters and is thus useful for teaching and research of SDR, especially in a university setting. In particular, the low cost of the laboratory makes it ideal for universities in the developing world. The FPGA card is a commercial product which is available worldwide, as are the necessary configuration files (via contacting the author). Hence, the SDR laboratory described here is ready for immediate deployment in universities around the world.

Acknowledgments

The author would like to thank Prof. Fabio Guzmán Andrés Figueroa of the Universidad Pontificia Bolivariana (UPB) in Medellín, Colombia for reading the draft manuscript and providing valuable comments. As well, the author thanks Profs. Raúl Restrepo, Alex Monclou, Jhon Jairo Padilla, and Claudia Rueda of the UPB Bucramanga and Prof. Sayra M. Cristancho from the University of Toronto for their logistical and technical assistance.

References

[1] H. Gomes, *Quality Quotes*: Irwin Professional Publishing, 1996.

[2] Xilinx Inc., "Spartan 3A Starter Kit HW-SPAR3A-SK-UNI-G," at www.xilinx.com, accessed May 2009

[3] Y. Linn, "Generation of Bandpass Gaussian Noise with Applications to Complete Built In Self Test in Wireless Communications Receivers," in *Proc. IEEE LATINCOM*, Medellín, Colombia, Sept. 10-11, 2009.

[4] Y. Linn, "Bridging the Wireless Communications Education Digital Divide: a Low-Cost Laboratory for Wireless Communications Teaching and Research," in *Proc. Latin America Workshop on Communications*, Medellín, Colombia, Sep. 8-11, 2009.

[5] Y. Linn, "A New Architecture for Coherent M-PSK Receivers," in *Proc. IEEE COMCAS'09*, Tel-Aviv, Israel, Nov. 9-11, 2009.

[6] U. Mengali and A. N. D'Andrea, *Synchronization techniques for digital receivers*. NY: Plenum Press, 1997.

[7] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital communication receivers: synchronization, channel estimation, and signal processing*. NY: Wiley, 1998.

[8] F. M. Gardner, "Interpolation in digital modems. I. Fundamentals," *IEEE Trans. Commun.*, vol. 41, no. 3, pp. 501-507, Mar. 1993.

[9] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems. II. Implementation and performance," *IEEE Trans. Commun.*, vol. 41, no. 6, pp. 998-1008, Jun. 1993.

[10] Y. Linn, "A new NDA timing error detector for BPSK and QPSK with an efficient hardware implementation for ASIC-based and FPGA-based wireless receivers," in *Proc. 2004 IEEE Intl. Symp. on Circuits and Systems (ISCAS'04)*, Vancouver, BC, Canada, May 23-26, 2004, pp. IV:465-468.

[11] Y. Linn, "A self-normalizing symbol synchronization lock detector for QPSK and BPSK," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 347-353, Feb. 2006.

[12] Y. Linn, "A symbol synchronization lock detector and SNR estimator for QPSK, with application to BPSK," in *Proc. 3rd IASTED International Conference on Wireless and Optical Communications (IASTED WOC'03)*, Banff, AB, Canada, Jul. 14-16, 2003, pp. 506-514.

[13] Y. Linn, "Robust M-PSK phase detectors for carrier synchronization PLLs in coherent receivers: theory and simulations," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1794-1805, Jun. 2009.

[14] Y. Linn and N. Peleg, "A family of self-normalizing carrier lock detectors and Es/N0 estimators for M-PSK and other phase modulation schemes," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1659-1668, Sep. 2004.

[15] Y. Linn, "Simple and Exact Closed-Form Expressions for the Expectation of the Linn-Peleg M-PSK Lock Detector," in *Proc. 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'07)*, Victoria, BC, Canada, Aug. 22-24, 2007.

[16] Y. Linn, "Quantitative analysis of a new method for real-time generation of SNR estimates for digital phase modulation signals," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1984-1988, Nov. 2004.

[17] Y. Linn, "A Carrier-Independent Non-Data-Aided Real-Time SNR Estimator for M-PSK and D-MPSK Suitable for FPGAs and ASICs,"

IEEE Trans. Circuits and Systems I, vol. 56, no. 7, pp. 1525-1538, Jul. 2009.

[18] Y. Linn, "Efficient Loop Filter Design in FPGAs for Phase Lock Loops in High-Datarate Wireless Receivers – Theory and Case Study," in *Proc. 6th Annual Wireless Telecommunications Symposium (WTS 2007)*, Pomona, CA, Apr. 26-28, 2007.

[19] Y. Linn, "Efficient Structures for PLL Loop Filter Design in FPGAs in High-Datarate Wireless Receivers – Theory and Case Study," in *Wireless Technology: Applications, Management, and Security*, S. Powell and J. P. Shim, Eds.: Springer, 2009.

[20] Y. Linn, "A Methodical Approach to Hybrid PLL Design for High-Speed Wireless Communications," in *Proc. 8th IEEE Wireless and Microwave Technology Conf. (WAMICON 2006)*, Clearwater, FL, Dec. 4-5, 2006.

[21] R. N. Mutagi, "Pseudo noise sequences for engineers," *Electronics & Communication Engineering Journal*, vol. 8, no. 2, pp. 79-87, Apr. 1996.

[22] F. Xiong, *Digital modulation techniques*. Boston: Artech House, 2000.

[23] B. A. Fette, *RF & wireless technologies*. Boston: Newnes, 2008.

[24] J. G. Proakis, *Digital communications*, 4th ed. Boston: McGraw-Hill, 2001.

[25] Y. Linn, "Synchronization, Phase Detection, Lock Detection, and SNR Estimation in Coherent M-PSK Receivers," Ph.D. Electrical and Computer Engineering, University of British Columbia, July 2007.

[26] Y. Linn, *Synchronization in Coherent M-PSK Receivers: Carrier Synchronization, Phase Detection, Lock Detection, and SNR Estimation*. Saarbruecken, Germany: VDM Verlag, 2008.

[27] K. E. Baddour and N. C. Beaulieu, "Autoregressive models for fading channel simulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1650-1662, Jul. 2005.

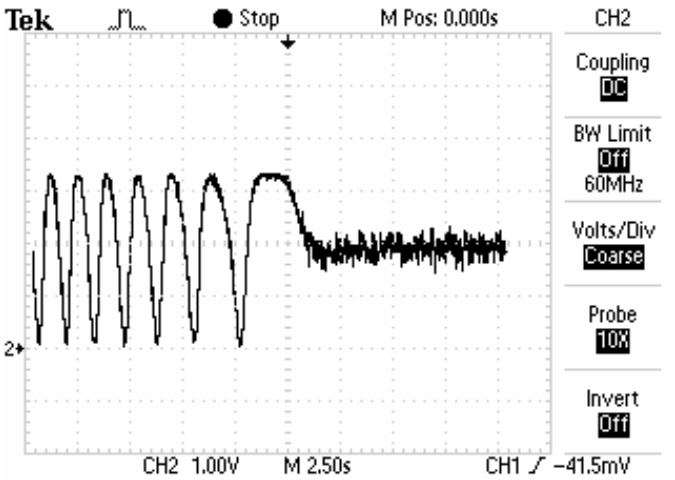

Fig. 12 – Pull-in process of the carrier PLL for BPSK. Cycle slips can clearly be seen, followed by lock-in and tracking.

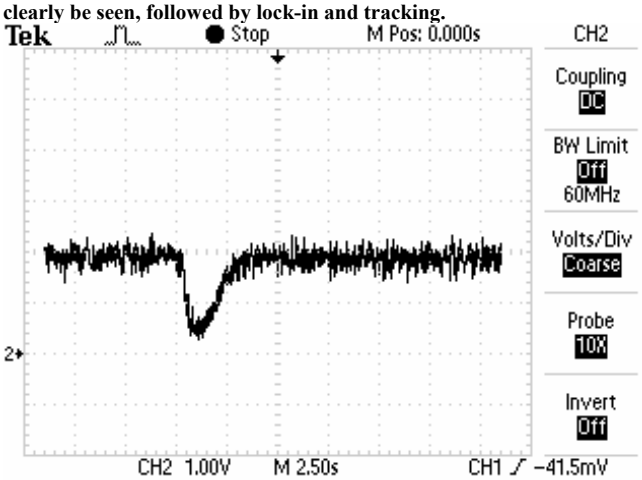

Fig. 13 – Carrier loop tracking and response to a small frequency step without loss of lock, for BPSK.

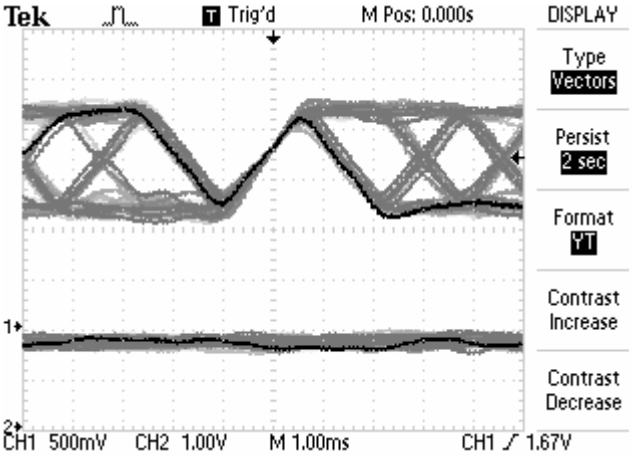

Fig. 14 - Eye Diagram for BPSK, high SNR (shown are the *I* and *Q* channels, at the receiver, after matched filter, before downsampling).

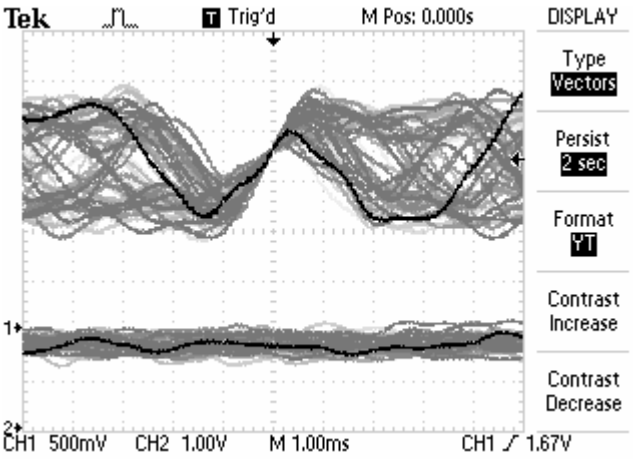

Fig. 15 - Eye Diagram for BPSK, low SNR (shown are the *I* and *Q* channels, at the receiver, after matched filter, before downsampling).

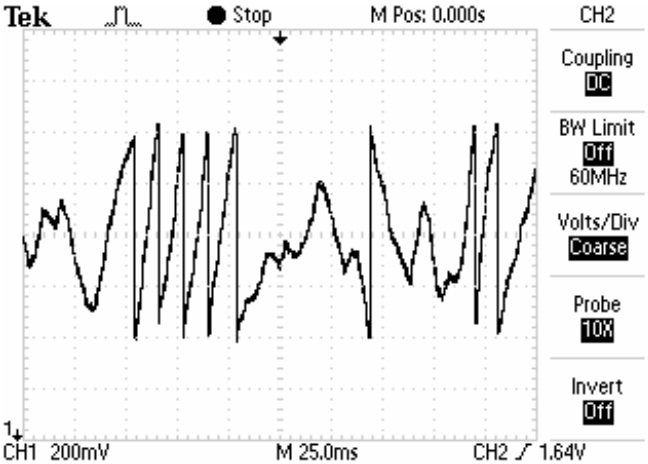

Fig. 16 – Value of the interpolator's control variable μ as function of time, for an interpolation rate of 10 samples/symbol. See [8, 9].

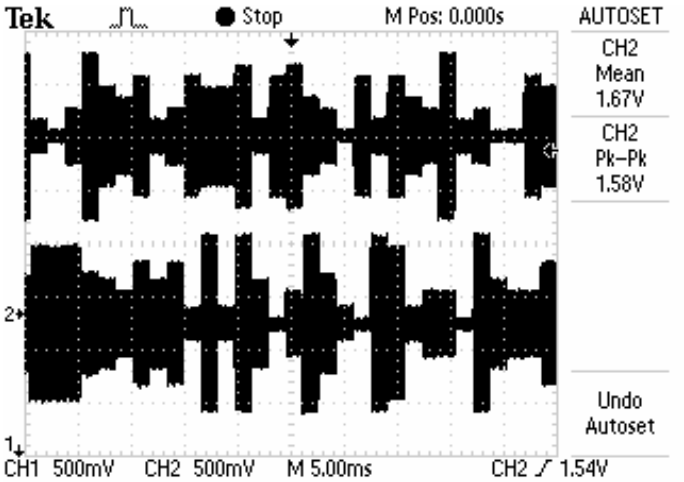

Fig. 17 - *I* and *Q* arms before summation in QAM-256 transmitter (zoom-out view)

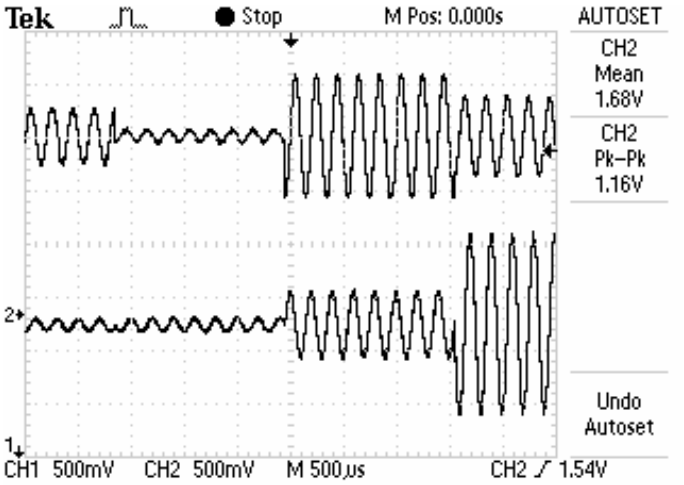

Fig. 18 - *I* and *Q* channel in transmitter before addition in a QAM-256 transmitter (zoom in view).

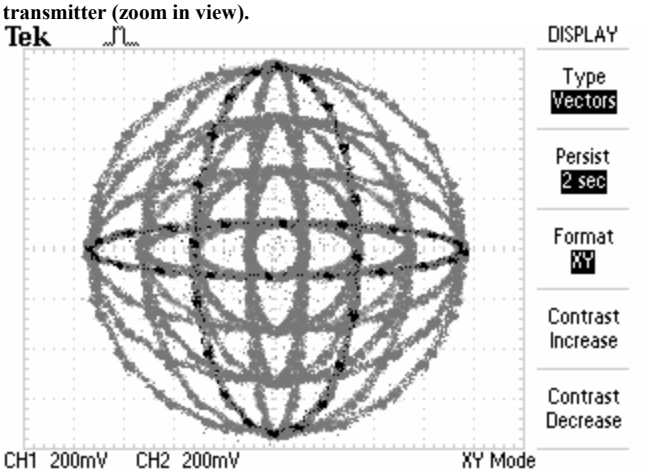

Fig. 19 - XY Graph of *I* arm vs. *Q* arm in QAM-64 transmitter.

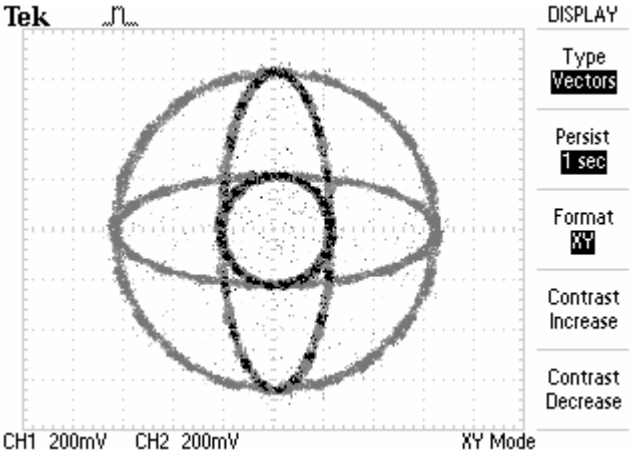

Fig. 20 - XY Graph of *I* arm vs. *Q* arm in QAM-64 transmitter.

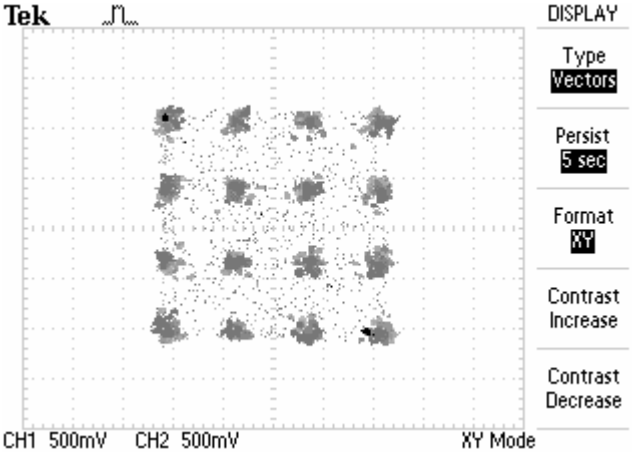

Fig. 21 - QAM-16 demodulated constellation.

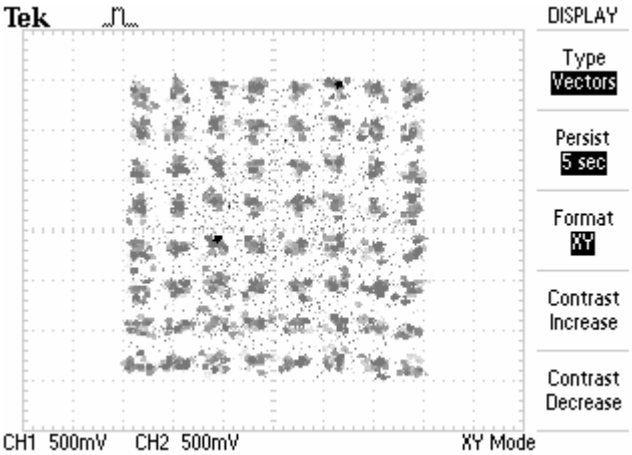

Fig. 22 - QAM-64 demodulated constellation.

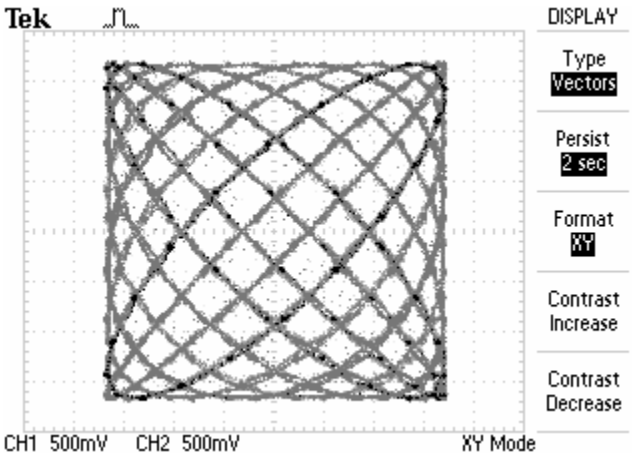

Fig. 23 - XY Graph of 16-PSK modulated carrier vs. unmodulated carrier.

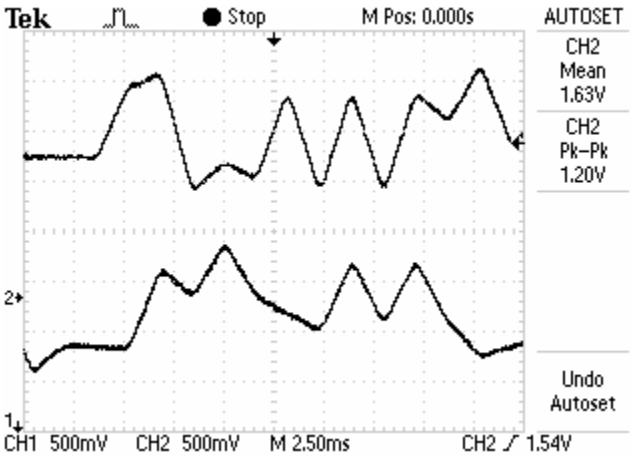

Fig. 24 - *I* and *Q* QAM-256 signal at outputs of matched filter in the receiver, before downsampling.

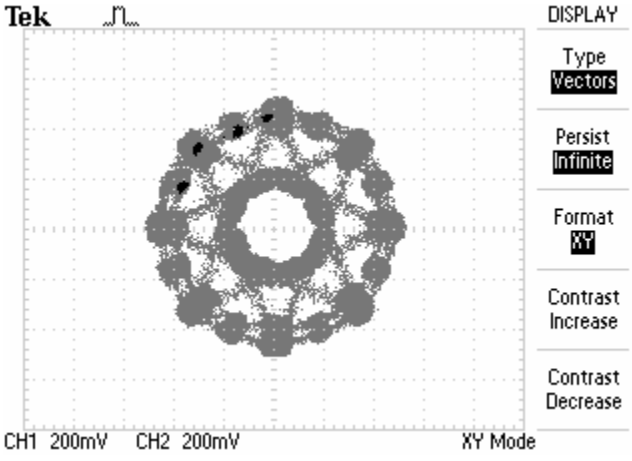

Fig. 25 - $\pi/8$-8PSK demodulated signal at receiver after resampler/interpolator, 2 samples/symbol.

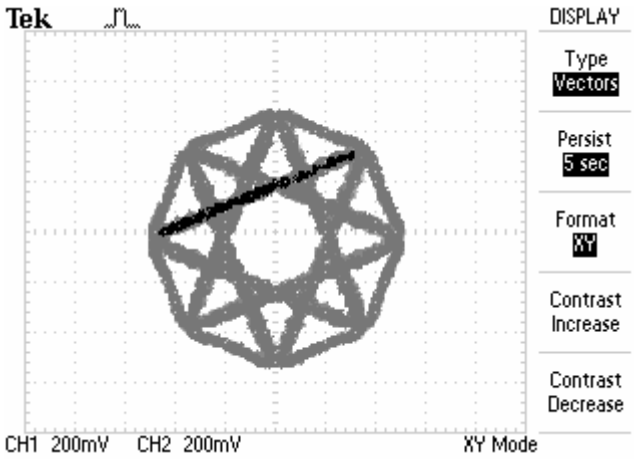

Fig. 26 - $\pi/4$ - QPSK demodulated signal at receiver after matched filter, before downsampling.

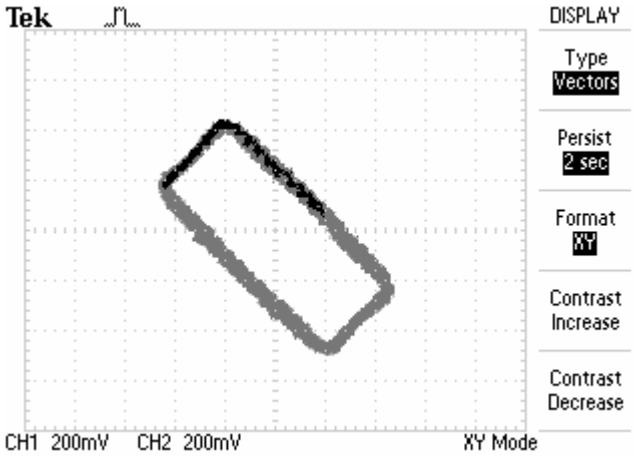

Fig. 27 - A unique (and possibly useless?) $\pi/4$ - BPSK demodulated signal at receiver after matched filter, before downsampling. Note that carrier and timing synchronization was achieved for this rather exotic modulation.

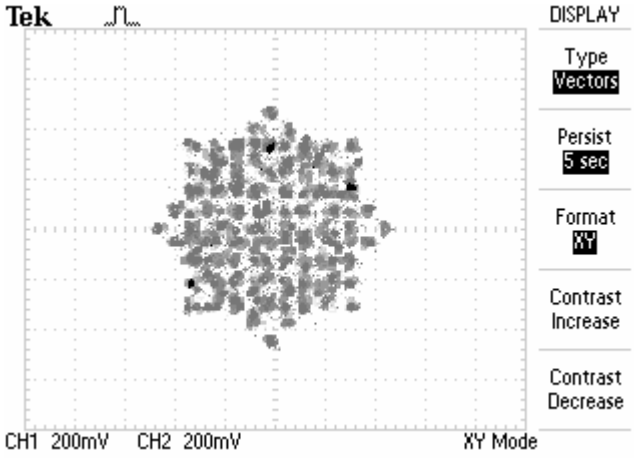

Fig. 28 - A $\pi/4$ - QAM-64 demodulated constellation at receiver output. Note that carrier and timing synchronization was achieved for this rather exotic modulation.