Model-based Software-defined Radio(SDR) Design Using FPGA

Anton S. Rodriguez, Michael C. Mensinger Jr., In Soo Ahn, and Yufeng Lu ,
Department of Electrical and Computer Engineering
Bradley University, Peoria, Illinois 61625

Abstract— **A software-defined radio (SDR) allows for digital communication systems to easily adopt more sophisticated coding and modulation technologies, which is extremely important in meeting the ever-increasing demands of the wireless communication industry. In this study, an SDR has been constructed using Xilinx system generator tools and implemented on the Virtex-II Field Programmable Gate Array (FPGA). The Xilinx system generator allows for fixed-point logic system design, hardware/software co-simulation, and board implementation through a model-based approach. In the SDR, Quadrature Phase Shift Keying (QPSK) is chosen as the modulation scheme for demonstration. Through a wireless link, the QPSK signal experiences amplitude attenuation, frequency and phase shifts causing rotation of the signal constellation. A phase-locked loop (PLL) circuit is designed for carrier and symbol synchronization for recovery of data from the received signal. In order to resolve the phase ambiguity of the QPSK system, a differential encoding scheme is also implemented.**

Keywords- Software-defined radio, QPSK, FPGA ;

I. INTRODUCTION

The software defined radios (SDR) has been proposed decades ago to implement the whole communication and digital signal processing system after the antenna picks up the signal. Recent technological advances in very large scale integrated (VLSI) circuit, digital signal processing (DSP) processor, graphic processing unit (GPU), and reconfigurable devices have enabled the SDR for a major push in communication system design and implementation [1-4]. The SDR provides a versatile wireless communication solution for a wide range of applications, including cellular telephones, global positioning system receivers, and military-grade communication radios. Since all hardware is physically programmed using software, re-design becomes relatively simple, making the SDR very cost-effective. Field Programmable Gate Array (FPGA) is widely used in embedded applications such as automotive, communications, industrial automation, motor control, medical imaging etc. In SDR applications, FPGA is an important alternative due to its reconfigurability. Without requiring hardware change-out, the use of FPGA type devices expands the product life by updating data stream files. FPGAs have grown to a level of holding an entire system on a single chip, while allowing in-platform testing and debugging of the system. Furthermore, it offers an opportunity of utilizing hardware/software co-design to develop a high performance system for different applications by incorporating processors (hardware core processor or software core processor), on-chip busses, memory, and hardware accelerators for specific software functions.

In conventional approaches, the digital designs on FPGA are implemented by using hardware description languages, which requires prior knowledge and experience in the tools such as VHDL and Verilog. In our preliminary investigation [5], it has been shown that the FPGA-based embedded system can be efficiently designed using Xilinx Embedded Development Kit (EDK) tools, C programming and VHDL together. The EDK tools enable the designer to focus on the design in system level, instead of struggling in the details of programming. However there is still a long learning curve to be familiar with these new tools. MATLAB/Simulink is an environment for model-based design, which covers almost all industrial and scientific areas. It is worthwhile to bridge over MATLAB/Simulink and FPGA design.

In this study, the Xilinx digital signal processing (DSP) tool: system generator is explored. With a graphical environment based on MATLAB/Simulink and a pre-designed block set of DSP cores, the system generator can be used to model a DSP system, which includes core DSP algorithms and synchronous interfaces to external peripherals and system control unit[6-10]. As a demonstration, a model-based SDR system is designed and implemented using FPGA.

This paper is organized as follows: Section II reviews the proposed SDR design. Section III discusses methods and procedures of the SDR design. Section IV covers the experimental setup and design validation. Section V concludes the paper.

II. QPSK DESIGN REVIEW

Quadrature phase shift keying (QPSK) is one of leading modulation schemes used in 3G wireless communications, WiMAX, WiFi, and the coming 4G wireless communication. It sends a pair of bits per symbol, increasing data rate by a factor of two.

QPSK modulation encodes data based on the phase angle between two waveforms. The two waveforms are represented as an in-phase component, $I(t)$, and an out-of-phase component, $Q(t)$. This information is modulated with a carrier

signal and transmitted. The mathematical representation of the transmitted signal, $s(\mathrm{t})$, is

$$\begin{aligned} s(\mathrm{t}) &= \mathrm{Re}\left[\left(I(t)+jQ(t)\right)e^{j2\pi f_c t}\right] \\ &= \mathrm{I}(t)\cos(2\pi \mathrm{f}_c t) - Q(t)\sin(2\pi \mathrm{f}_c t) \\ &= \sqrt{\mathrm{I}^2(t)+Q^2(t)}\cos\left(2\pi\, \mathrm{f}_c t + \phi(t)\right) \end{aligned} \quad (1)$$

where f_c denotes the carrier frequency and the vectors I and Q carry one bit information each. $\phi(t) = \tan^{-1}\left(\frac{Q(t)}{I(t)}\right)$ denotes the phase of the transmitted signal $s(\mathrm{t})$.

Once received, the signal is demodulated and the data can be extracted. The data can be represented as a constellation on an x-y plane in terms of symbols. The possible decoded symbols will be at radians of $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, and $\frac{7\pi}{4}$ as shown in Figure 1.

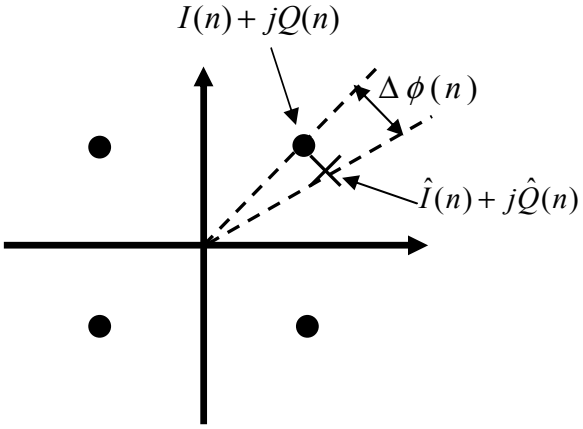

Figure 1 Constellation plot of QPSK system

A typical problem at the receiver end in wireless communication is carrier synchronization, or the synchronization of the oscillator at the receiver with the oscillator at the transmitter [11]. In order to do so, a phase-locked loop circuit must be appended to the receiver [12-13]. This provides the local oscillator at the receiver with a frequency adjustment. However, once this correction is made, a static phase error called phase ambiguity will still exist. In QPSK systems [14], the phase ambiguity could be any multiple of $\frac{\pi}{2}$. In order to properly decode transmitted data, a differential coding scheme is implemented which corrects any phase ambiguity.

Each of these constellation points, or symbols, represents two bits of information that are decoded based on their position in the constellation. Other modulation schemes, such as 8PSK. 16 Quadrature Amplitude Modulation (16QAM) and 64QAM increase data rate even further, sending 3, 4, or 6 bits information per symbol. In general, the more number of bits transmitted per symbol, a higher data rate is provided for a given bandwidth. Although the bits-per-symbol ratio has increased, the degree of accuracy needed for accurate transmission is also heightened, requiring a more complex demodulation scheme. Meanwhile, the constellation becomes more crowded in order to represent more symbols and the system is more vulnerable to inter-symbol interference (ISI) and channel impairments.

During wireless transmission, a signal travels from the transmitter to the receiver via a number of different paths, called multi-path effect as exhibited in Figure 2. Point A represents the transmitter and point B the receiver. The sum of these signals at point B will result in constructive and/or destructive interferences. In the QPSK application, the multi-path and relative motion between the transmitter and receiver cause amplitude attenuation, Doppler frequency shift, and phase shift in the constellation data points, as seen in Figure 1. The Doppler frequency shift and coarse carrier synchronization cause the QPSK constellation to rotate and thus, decoding of the data is a real challenge. In order to lock onto the correct QPSK constellation and correctly recover the transmitted data, coarse and fine carrier synchronization is required.

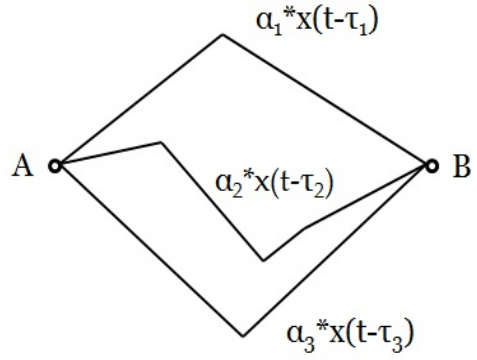

Figure 2 Multipath effect

Communication systems require local oscillators at both the transmitter and the receiver. The frequency and phase offset between these oscillators is inevitable due to various channel imperfections and transmission delay. Due to these imperfections, phase error is introduced in the received signal causing an incorrect representation of the transmitted data. In order to recover the data, a coherent detection is required and is achieved through a digital phase-locked loop (PLL). The coherent detection requires estimation of the instantaneous phase offset of every data point in the constellation grid with respect to where it should be. That information is then fed to a direct digital synthesizer which generates coherent sine and cosine carriers to correct the phase error. Upon completing carrier synchronization, data can be decoded. This is a crucial component of the SDR for successful demodulation of data.

There are phase ambiguity problems in QPSK systems. Figure 3 shows the in-phase channel data at both the transmitter and the receiver. The signal is inverted which will result in either a $\frac{\pi}{2}$ or π radians phase ambiguity as shown in

Figure 4. A possible solution to phase ambiguity is to attach a known sequence to the data stream or employ differential encoding.

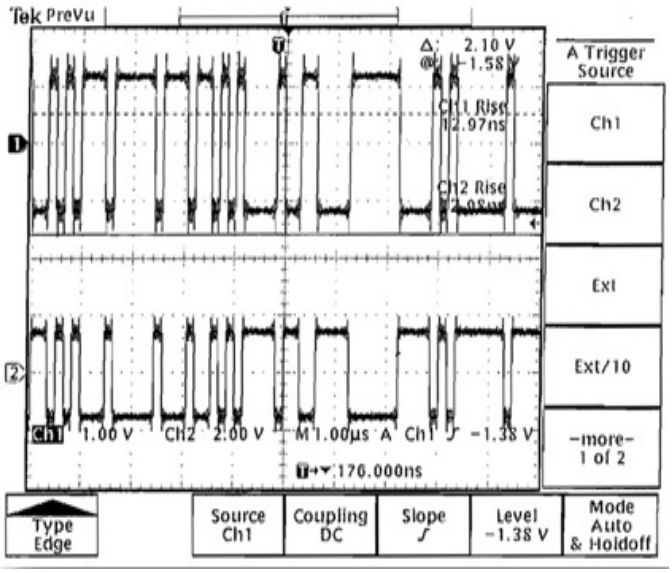

Figure 3 In-phase channel data at the transmitter (top) and the Receiver (bottom)

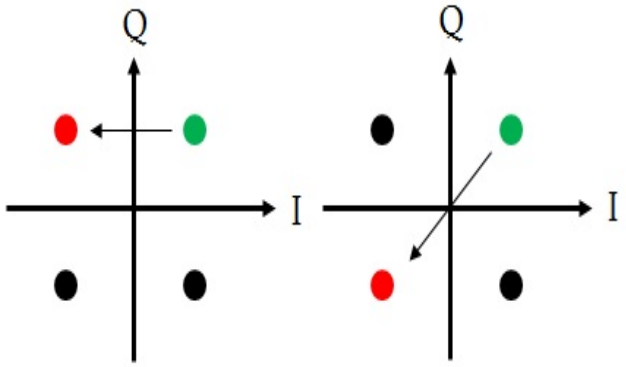

Figure 4 Possible phase rotation for the signals in Figure 3.

III. QPSK System In System Generator

The entire QPSK communication system is created in the MATLAB/Simulink environment using Xilinx System Generator. The overall system diagram of the QPSK system is shown in Figure 5. It can be seen that the system mainly includes five parts: symbol generator, interpolator, upconverter, downconverter, carrier recovery block, and decimator.

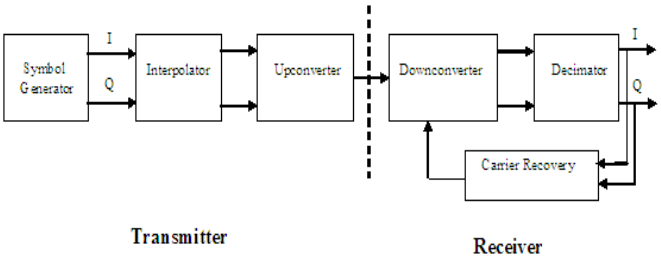

Figure 5 Block diagram of QPSK system

In the symbol generation, a read-only memory is used to allocate bit streams for the I and Q channels for testing purpose. For the implementation, an image data is used. Each pixel is split into consecutive two-bit symbols. This is completed using time division multiplexers to stream the symbols sequentially, and time division de-multiplexers to extract the corresponding symbols.

A common signal distortion in communication system is inter-symbol interference (ISI) among adjacent symbols. The ISI degrades system performance unless properly corrected using channel equalization at the receiver. The ISI makes communication system more sensitive to a synchronization error and more vulnerable to channel noises. Raised cosine signaling is a type of pulse shaping filter which has been widely used in practice to reduce detrimental ISI.

A simulation example of raised cosine filtering is shown in Figure 6. The raised cosine filter is implemented by using a distributed arithmetic FIR followed by data format converting blocks (i.e., Cast blocks). To further improve the system performance in terms of hardware resource and speed, a ployphase structure could be used for multirate signal processing.

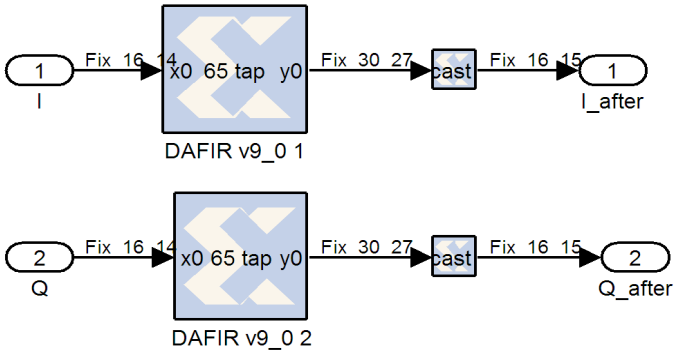

Figure 6 Raised cosine filter design

For simulation, a pulse train is used the filter input whose spectrum is shown as the top plot in Figure 7. The spectrum of filter output (the bottom plot in Figure 7) demonstrates that the bandwidth efficiency is greatly improved after applying a raised cosine filter.

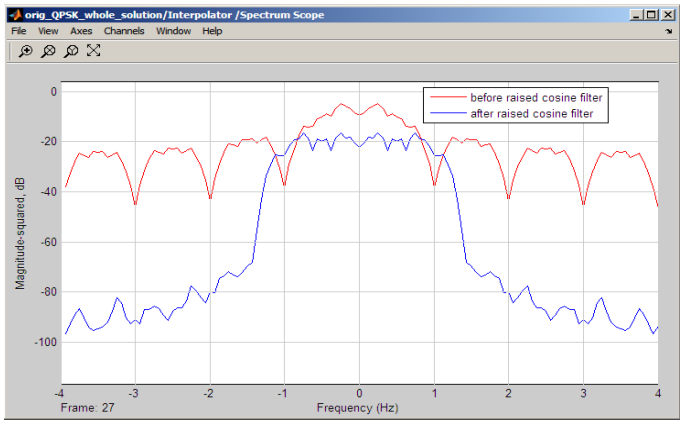

Figure 7 Simulation results before and after raised cosine filter

After being interpolated, the data is modulated by 12.5 MHz sine and cosine carriers and then sent through a high-speed ADC device and an RF antenna.

The received signal from a high-speed DAC device is down-converted and decimated for demodulation. The decimator utilizes the same FIR filter structure as the interpolator.

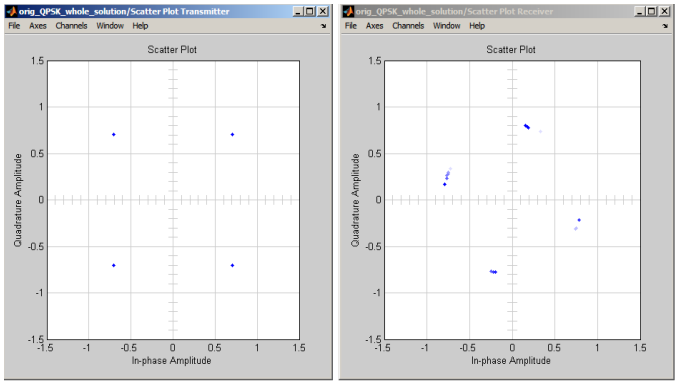

Figure 8. Simulation result of QPSK (Left: constellation plot from the transmitter; Right: constellation plot from the receiver)

Figure 8 shows a simulated result. It can be seen that the transmitted data can not be recovered correctly due the frequency offset of the local oscillator at the receiver.

The synchronization of the carrier frequency is the most important component of the communication system. In a wireless communication channel, various channel imperfections like ISI and Rayleigh fading due to multi-path effect exist. Without a synchronized local oscillator, data cannot be extracted correctly from the received QPSK signal. a phase-locked loop (PLL) is used to regenerate carrier frequency and correct for the channel imperfections introduced during transmission.

The QPSK waveform carries phase information that represents data. By performing the following computation

$$\sin(\Delta\phi) = \frac{\hat{I}(n)Y(n) - \hat{Q}(n)X(n)}{\sqrt{\hat{I}^2(n) + \hat{Q}^2(n)}\ \sqrt{X^2(n) + Y^2(n)}} \qquad (2)$$

where $\Delta\phi$ denotes the phase error, $X(n)$ and $Y(n)$ are the outputs from decimators, the phase information can be extracted. $\hat{I}(n)$ and $\hat{Q}(n)$ are the estimated in-phase and quadrature hard-decoded data. The phase error $\Delta\phi$ (see Figure 1) is calculated and then used to adjust the frequency and phase of the local oscillator. The transmitter and receiver frequencies become synchronized, allowing for extraction of data from the $\hat{I}(n)$ and $\hat{Q}(n)$ channels.

Equation 2 is calculated for every data sample, meaning that this must be implemented in hardware and run at the symbol rate. The phase error calculation is illustrated in Figure 9.

The estimated phase error is fed as a feedback controller for the adjustment of the local oscillator. "Sin/Cos LUT" block serves as direct digital synthesizer to regenerate the recovered carrier signals.

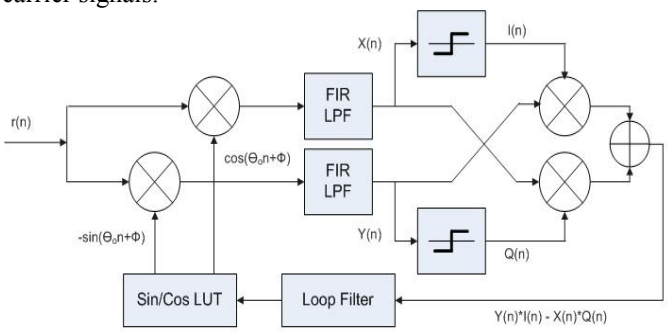

Figure 9 Phase locked loop and carrier recovery

From Figure 9, crucial to the design of the PLL is the loop filter, which provides direct control over the PLL bandwidth. The bandwidth of a PLL is the measure of the PLL's ability to track the input clock and jitter. A high bandwidth setting provides a fast lock time and tracks jitter on the reference clock source, passing it through to the PLL output, whereas a low bandwidth setting filters out reference clock jitter, but increases lock time [15].

To control phase error, proportional and integral (PI) control technique is used via a loop filter. PI control is implemented by selecting values for K_p and K_i, which provide the user with control over bandwidth and a damping factor. By optimizing these parameters, the system can achieve carrier synchronization with both a fast locking time and reduced jitter. First, the bandwidth for desired operation is chosen and then K_p and K_i values are derived using Equations 3 and 4.

$$K_p = \frac{2\sqrt{2}\theta}{1 + \sqrt{2}\theta + \theta^2} \qquad (3)$$

$$K_i = \frac{4\theta^2}{1 + \sqrt{2}\theta + \theta^2} \qquad (4)$$

where $\theta = 2\pi \times BW$ and BW is bandwidth of the filter in Hz. It can be adjusted according to the desired behavior of the

PLL with the considerations stated above. The loop filter structure in system generator is shown in Figure 10.

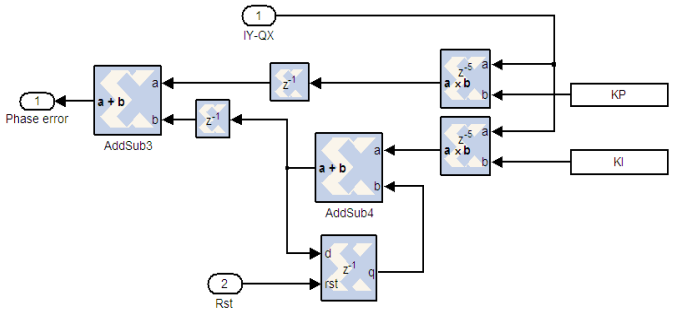

Figure 10. Loop filter structure

Differential encoding is used to compensate for the errors introduced due to phase ambiguity, an inherent problem of QPSK systems. As stated in Section II, differential encoding is a part of the data conditioning where complex data is converted to a sequence of two-bit symbols. Differential encoding is then applied to the stream of two-bit symbols and later the decoding process is applied after the $\hat{I}(n)$ and $\hat{Q}(n)$ channels have been re-combined at the receiver end. The result is a data sequence corrected of phase ambiguity. The encoding process calculates the difference between consecutive symbols and that value is transmitted. This is achieved by putting the symbol sequence through a discrete time integrator (See Figure 12). Once the data is transmitted and received, the decoding process begins by feeding the encoded symbols through a discrete time differentiator (See Figure 13).

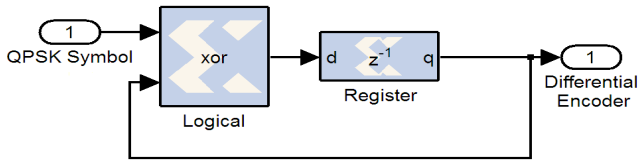

Figure 12 Differential encoder

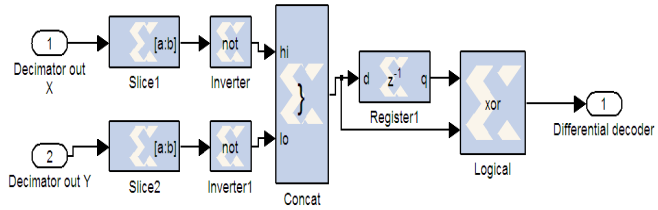

Figure 13 Differential decoder

IV. Experimental Setup And Design Validation

To implement the SDR system, a SignalWAVe board from Lyrtech. Inc. is used. It features the prototype capabilities for wireless communication, includes a complementary DSP/FPGA architecture, i.e., a TMS320C67 DSP processor operating at 225MHz and a Xilinx Virtex II family FPGA (XC2V3000), which has 3 million equivalent logic gates. Both FPGA and DSP are equipped with a total of 32 MB SDRAM. For the analog interface, the board consists of a 14-bit resolution, low power digital-to-analog converter AD9754 DAC (up to sampling rate of 125 MHz) and a 14-bit analog-to -digital converter AD6644(up to sampling rate of 65 MHz). Many applications such as GPS receiver, single side band radio have been successfully implemented on the board [16].

In our SDR design, Simulink and system generator are used to simulate the system. Then the complete system (see Figure 14) is downloaded to the FPGA board. The high speed ADC and DAC devices are used as the communication channel. The $\hat{I}(n)$ and $\hat{Q}(n)$ channels of the received signal are combined to form the QPSK constellation grid points. Without carrier synchronization, the QPSK constellation points cannot be restored, but a rotating phasor will be observed. The recovered constellation is shown in Figure 15. It confirms that the PLL tracks the carrier frequency offset and synchronizes the carrier frequencies successfully.

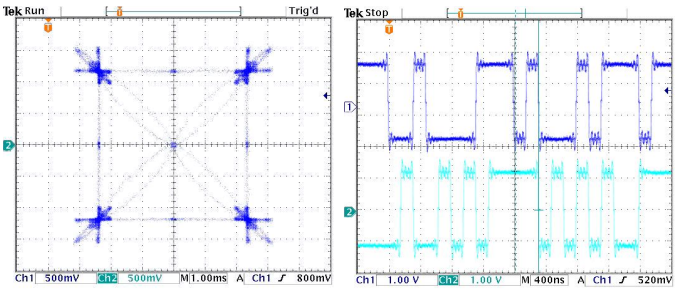

Figure 15 [From left to right] a) The constellation plot from the receiver b) the I channel and Q-channel data at the receiver.

In order to validate and evaluate overall performance of the SDR, an image file is used to test the system. To test the system, image pixels are pre- and post-processed to generate QPSK symbols and to convert QPSK symbols back to image pixels. The data conversion is handled in hardware utilizing the time division multiplexer and de-multiplexer Xilinx blocks. Figure 16 shows that the image is fully recovered at the receiver end.

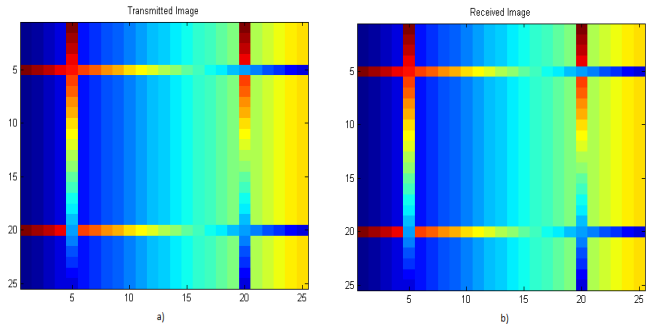

Figure 16 [From left to right] a) The transmitted image pattern b) the received image pattern.

V. CONCLUSION

The software-defined radio is a very flexible solution to the digital wireless communications industry. Xilinx provides a powerful design tool for FPGA implementation. In this work, a SDR with QPSK modulation scheme has been successfully designed using Xilinx tools. The PLL and differential coding scheme are implemented for recovery of carrier frequency offset and resolution of phase ambiguity of the QPSK system. Software simulation and hardware implementation procedures show that this type of hardware/software design methodology is well suited to a broad range of applications in communication and signal processing.

REFERENCES

[1] V. B. Alluri, J. R. Heath, and M. Lhamon, " A new multichannnel , coherent amplitude modulated, gate array technology implementation," *IEEE Transaction on Signal Processing*, vol. 58, pp. 5369-5384, October 2010.

[2] J. Mitola, " Software radios: survey, critical evaluation and future directions," *IEEE Aerospace and Electronics Systems Magazine*, vol. 8, pp. 25-36, April 1993.

[3] M. Palkovic, P. Raghavan, M. Li, A. Dejonghe, L. Van der Perre, and F. Catthoor, " Future Software-Defined Radio Platforms and Mapping Flows," *IEEE Signal Processing Magazine,* vol.27, no.2, pp.22-33, March 2010.

[4] J. Kim, H. Seungheon, and S. Choi; , "Implementation of an SDR system using graphics processing unit," *IEEE Communications Magazine* , vol.48, no.3, pp.156-162, March 2010.

[5] Y.Lu, E. Oruklu, and J. Saniie, " Fast chirplet transform with FPGA-based implementation," *IEEE Signal Processing Letters*, vol. 15, pp. 577-580, December 2008.

[6] A. T. Moreo, P. N. Lorente. F. S. Valles, J. S. Muro, " Experiences on developing computer vision hardware algorithm using Xilinx system generator," Microprocessors and Microsystems: Special Issue on FPGAs; Case Studies in Computer Vision and Image Processing,, pp. 411-419, November 2005.

[7] C. Chang, J. Wawrzynek and R. W. Brodersen, " BEE2: a high-end reconfigurable computing system," *IEEE Design & Test of Computers,* vol.22, no.2, pp. 114- 125, March-April 2005

[8] Xilinx, " System Generator 12.1 User's Manual", Xilinx. Inc., 2010.

[9] A. Jimenez-Fernandez, A. Linares-Barranco, R. Paz-Vicente, C. D. Lujan-Martinez, G. Jimenez, and A. Civit, "AER and dynamic systems co-simulation over Simulink with Xilinx System Generator,"*Proceedings of IEEE International Conference on Electronics, Circuits and Systems* , pp.1281-1284, Aug. 2008.

[10] H. S. Cho, S. H. Im and G. I. Jee, " A FPGA-based Software GPS Receiver Implementation Using Simulink and Xilinx System Generator,"*Proceedings of 18th International Technical Meeting of the Satellite Division of the Institute of Navigation*, pp. 234-240, September 2005.

[11] C. Dick, F. Harris, and M. Rice, " FPGA Implementation of Carrier Synchronization for QAM Receivers," *Journal of VLSI Signal Processing*, vol. 36, pp. 57-71, 2004.

[12] J. Proakis, " Digital Communications", McGraw Hill Science, August 2000.

[13] D. R. Stephens, " Phase-locked Loops for Wireless Communication Digital and Analog Implementation," Kluwer Academic, 1998.

[14] V. K. V. R. Gari, "FPGA-based QPSK Transceiver Design", Technical Report, Department of Electrical and Computer Engineering, Bradley University, November 2008.

[15] Altera Corporation, " PLL and Clocking Glossary", Altera, 2010.

[16] D. Benson, " The Design and Implementation of a GPS Receiver Channel, An FPGA plus DSP Creats a Powerful Combination", *DSP Magazine*, October 2005.

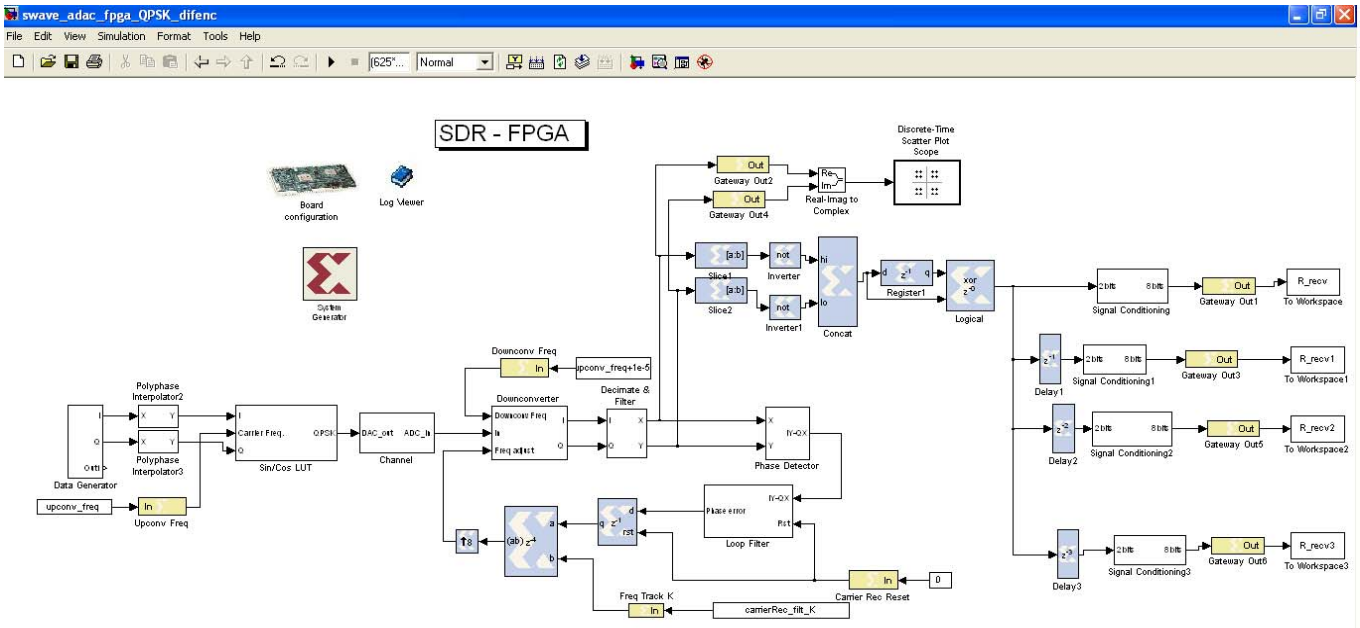

Figure 14 SDR design for the FPGA on signalWAVe board.