AN FPGA-BASED ZIGBEE RECEIVER ON THE HARRIS SOFTWARE DEFINED RADIO SIP

Tingting Meng, Chen Zhang, Peter Athanas
The Bradley Department of Electrical and Computer Engineering, Virginia Tech,
Blacksburg, VA, USA 24061
Email: {tmeng, zhangc, athanas}@vt.edu

ABSTRACT

A software defined ZigBee receiver has been designed, implemented and tested with a commercially available off-the-shelf ZigBee transceiver. This system is based on the software defined system-in-package (SIP) platform by Harris Corporation (Melbourne, Florida, USA). This platform is equipped with a high-resolution analog-to-digital converter (ADC) and four Xilinx Virtex-4 FPGAs. An all-digital phase-locked loop (PLL) and ZigBee demodulator have been created to operate inside the FPGAs. The ZigBee medium access control layer is also implemented inside an FPGA to allow the system to recover the commercial ZigBee signals. A simple RF front-end is composed of commercially available off-the-shelf components chosen for this system. This work permits a large portion of the RF and IF hardware chain, as well as the physical layer and medium access control layer, to be reconfigured inside the FPGAs. A streaming video source emanating from the XBee ZigBee transceiver is demonstrated on the SIP in real time.

1. INTRODUCTION

Software defined radio (SDR) provides communications devices with the ability to change the transmitted and received waveform without physically modifying the hardware. Typical implementations of SDR systems include a general-purpose processor (GPP), digital signal processor (DSP) and field programmable gate arrays (FPGAs) [1]-[3]. Since the FPGAs are capable of offloading either the GPP or DSP or both, a SDR system can be implemented using suitable radio frequency (RF) and intermediate frequency (IF) stages in conjunction with the FPGAs. However, the receiver depends upon the analog RF front-end and the processor for the MAC sublayer implementation. It is a challenging task to map the analog RF front-end structures and the MAC sublayer control to the FPGAs.

ZigBee [4] is a low data rate, low power, low cost, wireless networking protocol based on the *IEEE 802.15.4* standard [5] for wireless personal area networks (WPAN). Since this technology is simpler and cheaper than other wireless personal area networks, such as *IEEE 802.15.1*/Bluetooth, ZigBee is becoming the standard approach for wireless sensors and embedded applications on the industrial, scientific and medical radio bands. Nowadays, more and more attention has been placed on ZigBee.

In this paper, a software defined ZigBee receiver with a digital IF down-converting stage and corresponding digital filters was built in a single Xilinx Virtex-4 FPGA. A novel all-digital PLL was also designed and implemented inside the FPGA. Unlike the contemporary digital and all-digital phase-locked loops (PLL) have been reported recently [6]-[9], the presented PLL did not utilize a phase detector or a loop filter. Instead, a chip balance detector based on three accumulators was deployed to achieve the same function of the phase detector and the loop filter. The MAC sublayer of the ZigBee receiver was also modeled by using VHDL and is capable of being reconfigured easily inside the FPGA. The performance of the ZigBee receiver was validated in part of an experiment featuring a video link between this ZigBee receiver on Harris SIP and a commercial ZigBee transceiver, XBee [10] manufactured by MaxStream.

The rest of the paper is organized as follows. Section 2 describes the system hardware used for this SDR platform and the design of the analog RF front end for this receiver. Section 3 presents the implementation of the PHY layer and MAC sublayer of the ZigBee receiver. The set-up and the performance of the video link experiment are discussed in Section 4. Section 5 concludes the open issues in this research and the future work on this Harris system-in-package.

2. SDR PLATFORM HARDWARE

The SDR ZigBee receiver is divided into two separate sections: a *2.4GHz* analog RF front-end chain and the Harris SIP.

2.1. RF Front-end

The RF front-end is composed of an antenna, a RF band-

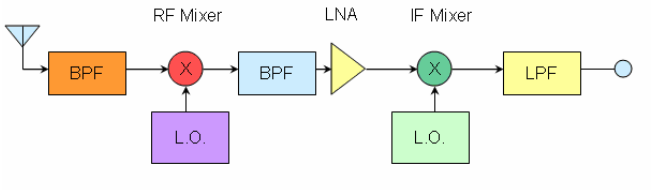

Figure 1 Block diagram of the RF front-end of ZigBee receiver.

pass filter, a low noise amplifier, a IF band-pass filter, a low-pass filter, two frequency mixers, and two local oscillators. Significant time, flexibility and resource savings was achieved by using commercially available off-the-shelf components. Figure 1 shows the block diagram of the RF front-end of the ZigBee receiver.

The band-pass filter with center frequency *2.45GHz* and *200MHz* bandwidth is used as a pre-selective filter. The RF signal is down-converted to *70MHz* first IF stage and filtered by a band-pass filter with a *70MHz* center frequency and *2MHz* bandwidth. The low noise amplifier enhances the *70MHz* IF signal for the next IF stage. In the second IF stage, the signal is down-converted to *6MHz* and fed into the high-resolution ADC.

2.2. Harris Software Defined SIP Board

The Harris SIP is a software defined radio featuring, among other things, a high-resolution analog to digital converter with *14*-bit of resolution, four Xilinx *XC4VLX60* Virtex-*4* FPGAs [11], and a Texas Instruments DaVinci processor. The processor on SIP is capable of configure and software-defined the whole system. The recovered data can be delivered to a computer through a wired Ethernet port on the SIP. Figure 2 shows the block diagram of the Harris SIP board.

3. ZIGBEE RECEIVER IMPLEMENTION

The *IEEE 802.15.4* standard employs an offset quadrature phase shift keying (OQPSK) modulation with a half-sine pulse shaping for signals transmitted in the *2.4GHz* ISM frequency band. A direct sequence spread spectrum (DSSS) coding with processing gain of eight is obtained by using sixteen quasi-orthogonal codes, each composed by thirty-two chips and encoding *4*–bit data. The chip rate is *2Mcps* and the raw data rate obtained is *250Kbps*. Each *32*-bit code is divided into two *16*-bit sub-codes that are separately modulated and mapped to the in-phase and quadrature channel [12][13]. The carrier frequency locates at channel 3 of the 16 channels equally distributed between *2405MHz* and *2490MHz*. The bandwidth of each channel is *3MHz*.

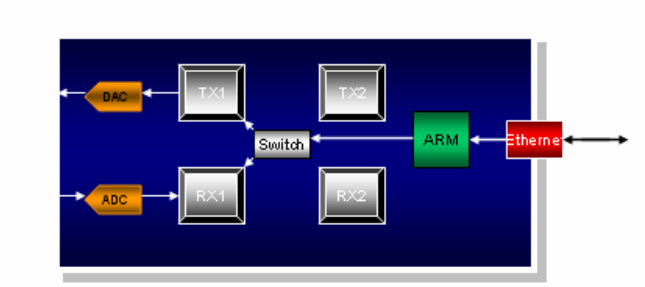

Figure 2 Block diagram of Harris SIP board.

OQPSK is a variant of phase-shift keying modulation using four different values of the phase to transmit. By offsetting the timing of the odd and even bits in one bit-period, or half a symbol-period, the in-phase and quadrature components will never change at the same time. In the constellation diagram, it can be seen that the phase-shift is limited to no more than *90* degree at a time rather than the *180* degree phase-shift for quadrature phase shift keying. The DSSS modulation technique multiplies the data being transmitted by a "noise" signal. This noise signal is a pseudorandom sequence of *1* and *-1* values, at a frequency much higher than that of the original signal, thereby spreading the energy of the original signal into a much wider band. The XBee RF module is a ZigBee/ *IEEE 802.15.4* compliant solution that satisfies the unique needs of low-cost, low-power wireless sensor networks.

3.1. PHY Layer Implementation

A ZigBee receiver architecture has been designed to take the maximum advantages of the processing power of the Harris SDR hardware platform. This architecture processes the carrier synchronization, IF down-conversion, filtering, quadrature demodulation, chip synchronization and dispreading inside a FPGA. A simplified block diagram of

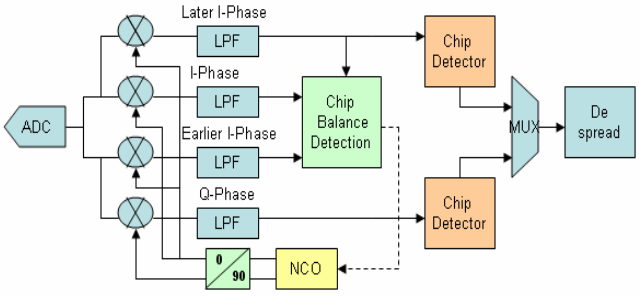

Figure 3 Block diagram of the ZigBee receiver structure on Harris SIP.

the ZigBee receiver is shown in Figure 3. The *14*-bit digitized IF signals from the ADC is down-converted to the in-phase, earlier in-phase, later in-phase and quadrature base-band signals. The three in-phase signals with different time offset are directed into a chip balance detection module to determine the phase offset between the carrier and the local oscillator. This module controls the numerically controlled oscillator (NCO) to remove the phase offset. The in-phase and quadrature signals are sent to chip detectors to achieve the chip synchronization. The de-spreading module recovers the bits after obtaining the correct chip synchronization.

3.2. Data Packets Processing Implementation

The *IEEE 802.15.4* PHY and MAC layer frame structures have been designed to keep the complexity to a minimum and to make packets sufficiently robust for transmission on a noisy channel. The data frame for the *IEEE 802.15.4* has been designed to recover the transmitted source data from the demodulated ZigBee data with layer-specific headers and footers. Figure 4 shows the structure of the data frame [5][14]. It consists of PHY layer header, MAC sublayer header, data payload and a frame check sequence. The PHY layer header contains a preamble sequence of bytes for packet detection and chip synchronization, a frame delimiter byte for indicating the start of the packet and symbol synchronization, and a frame length byte for informing the length of the packet. The MAC sublayer header includes frame control bytes, a sequence number byte and addressing field bytes. The data payload is the data to be sent. The frame check sequence bytes are for cyclic redundancy check.

After a preamble sequence is received, a matched filter is used for detecting the frame delimiter byte, *"11100101"*. Once the frame delimiter byte is detected, the next received byte, frame length byte, is use to set a packet receiving count-down counter. This counter controls the length of the MAC sublayer packet. Since only non-encrypted data frames are transmitted for the real-time video steam, the frame control bytes and the sequence number byte can be ignored. Another matched filter is used to determine if the packet is sent to this specific receiver. The next received one hundred bytes, the maximum length of the data payload, are sent to a remote computer through Ethernet. Since the video data does not require high accuracy, the cyclic redundancy check can also be omitted. The whole design of the PHY layer and MAC sublayer of the ZigBee receiver was implemented in one Xilinx Virtex-*4* LX*60* FPGA on the Harris software defined SIP board, which occupies *3,047* slices out of *26,624* (*11%*), *3,659* flip-flops out of *53,248* (*6%*), *4,125* *4*-input LUTs out of *53,248* (*7%*) and *24* multipliers out of *64* (*37%*) . Due to the tight timeline of this project, certain optimizations have been omitted. The design can be further improved in several aspects, which will be discussed in the next section.

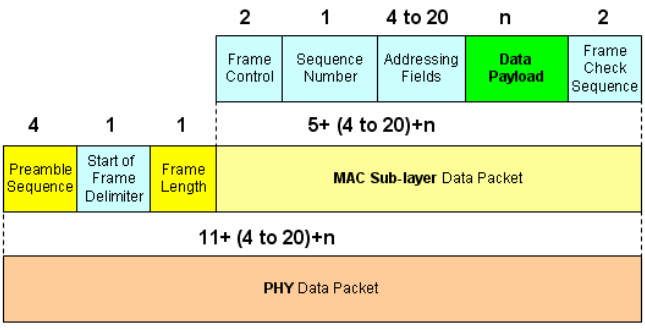

Figure 4 Structure of the ZigBee data frame.

4. EXPERIMENT AND FUTURE WORK

An experiment designed to validate the functionality of the FPGA-based ZigBee receiver is shown in Figure 5. A commercial ZigBee device, XBee shown in Figure 6, was connected with a camera through *RS-232*. The streaming video taken by the camera was fed to the XBee as the transmitting source data. XBee transmitted the ZigBee signals through the antenna over the wireless channel. The FPGA-based ZigBee receiver received the signals; recovered the transmitted data; and presents them to a remote computer through Ethernet. Figure 7 shows the recovered video signals. It can be observed that some errors occurred at the data recovering process. These errors are mainly caused by the carrier frequency jitter of the XBee signal that moves out of the tracking range of the receiver PLL. A rotating chip detection structure was deployed to reduce the impact of this kind of errors. As the result, the

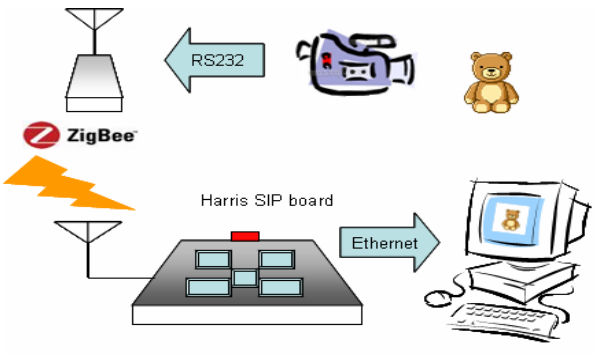

Figure 5 Experimental setup of the ZigBee SDR link.

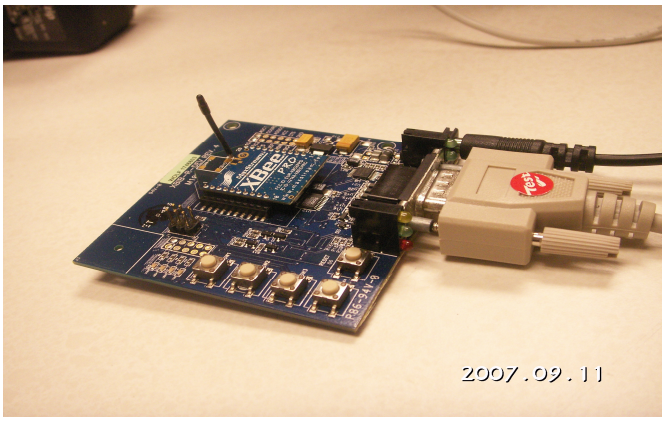

Figure 6 MaxStream XBee transceiver.

number of required slices was increased from *6%* to *11% of 26,624* slices. To reduce the chip utility, one of the future improvements is to increase the frequency tracking range of the all-digital PLL. Another improvement in the future could be the filter structure. For simplicity, all filters in this receiver are in full parallel structure and run at *48MHz* in the current design. A pipelined filter structure can reduce the number of multipliers from twenty-four to four by reusing the high-speed multipliers.

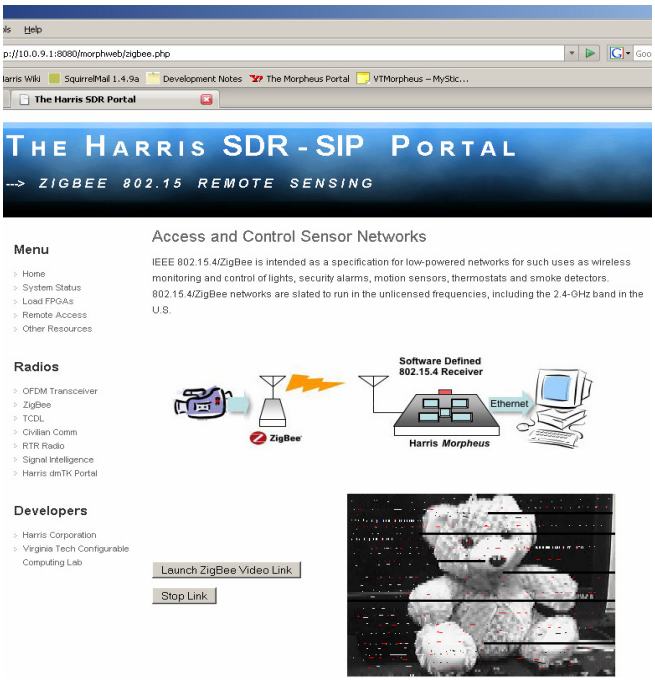

Figure 7 Snapshot of the webpage of received streaming video.

5. CONCLUSION

In this paper, a ZigBee receiver was implemented. The ample resources on the Harris SDR SIP made it possible to implement a software defined IF down-converting and reconfigurable PLL in the FPGA. The performance of the ZigBee receiver was determined to be comparable to that of commercial ZigBee products. As a result of this work, other wireless communications systems (such as Bluetooth, Wireless USB and *IEEE 802.11* b/g) as well as their MAC sublayers can be easily implemented on the Harris SIP. Further studies will be directed toward the use of partial reconfiguration and the use of reconfigurable modulation/demodulation elements.

6. ACKNOWLEDGEMENT

We would like to acknowledge the support from Harris Corporation that enabled the development of this research.

7. REFERENCES

[1] H. Harada, "Software defined radio prototype for W-CDMA and IEEE 802.11a wireless LAN,"IEEE VTC'04 Fall, Los Angeles, USA, Nov.2004.

[2] G. D. Jo, M. J. Sheen, S. H. Lee, and K. R. Cho, "A DSP-based reconfigurable SDR platform for 3G systems," IEICE Trans. Commun., vol.E88-B, no.2, pp.678-686, Feb. 2005.

[3] H. Shiba, T. Shono, Y. Shirato, I. Toyoda, K. Uehara, and M. Umehira, "Software defined radio prototype for PHS and IEEE 802.11 wireless LAN," IEICE Trans. Commun., vol.E85-B, no.12, pp.2703-1715, Dec. 2002.

[4] ZigBee Alliance, Network Specification, Version 1.0, Dec. 2004.

[5] IEEE Standard 802, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), Institute of Electrical and Electronic Engineers, Oct, 2003.

[6] R. B. Staszewski, D. Leipold, K. Muhammad, and P. T. Balsara, "Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS process," *IEEE Trans. Circuits Syst. II.*, vol. 50, no. 11, pp. 815–828, Nov. 2003.

[7] R. B. Staszewski, J. L. Wallberg, S. Rezeq, C.-M. Hung, O. E. Eliezer, S. K. Vemulapalli, C. Fernando, K. Maggio, R. Staszewski, N. Barton, M.-C. Lee, P. Cruise, M. Entezari, K. Muhammad, and D. Leipold, "All-digital PLL and transmitter for mobile phones," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2469–2482, Dec. 2005.

[8] J. Lin, B. Haroun, T. Foo, J.-S.Wang, B. Helmick, S. Randall, T. Mayhugh, C. Barr, and J. Kirkpatric, "A PVT tolerant 0.18 MHz to 600MHz self-calibrated digital PLL in 90 nm CMOS process," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 2004,pp. 488–541.

[9] N. D. Dalt, E. Thaller, P. Gregorius, and L. Gazsi, "A compact tripleband low-jitter digital *LC* PLL with programmable coil in 130-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1482–1490, Jul. 2005.
[10] MaxStream Co., "XBee / XBee-PRO OEM RF Modules," *datasheet*, Jan. 4, 2007.
[11] Xilinx, Inc., "Vitex-4 Family Overview," Xilinx Virtex-4 FPGA Data Sheet, Jan., 2007
[12] A.Di. Stefano, G.Fiscelli, and C.G. Giaconia, "An FPGA-Based Software Defined Radio Platform for the 2.4GHz ISM Band," *Research in Microelectronics and Electronics*, pp.73-76, Jun, 12-15, 2006.
[13] Chipcon AS SmartRF. "CC2420 - 2.4 GHz *IEEE 802*.15.4 / ZigBee RF Transceiver," CC2420 Preliminary Datasheet (rev 1.1), Mar., 2004.
[14] G. Lu, B. Krishnamachari, and C. Raghavendra, "Performance Evaluation of the IEEE 802.15.4 MAC for Low-Rate Low-Power Wireless Networks," Proc. Workshop Energy-Efficient Wireless Comm. and Networks, 2004.