

IMPLEMENTATION OF A VERILOG-BASED DIGITAL RECEIVER FOR 2.4 GHz ZIGBEE APPLICATIONS ON FPGA

RAFIDAH AHMAD*, OTHMAN SIDEK,
SHUKRI KORAKKOTTIL KUNHI MOHD

Collaborative Microelectronic Design Excellence Centre (CEDEC),
Universiti Sains Malaysia, Engineering Campus,
14300 Nibong Tebal, Seberang Perai Selatan, Penang, Malaysia
*Corresponding Author: rafidah.ahmad@usm.my

Abstract

This paper presents the implementation of a digital receiver for 2.4 GHz Zigbee IEEE 802.15.4 applications on a Spartan3E XC3S500E field programmable gate array (FPGA). The proposed digital receiver comprises an offset quadrature phase shift keying (OQPSK) demodulator, chip synchronization, and a de-spreading block. A new design method that uses Verilog hardware description language (HDL) code through Xilinx ISE version 12 was developed to design these blocks. These blocks were integrated into one top module for optimization. Simulation and measurement were conducted to verify the functionality of the receiver. Implementation results show that the receiver design matched the theoretical expectation. The implementation configuration required up to 22% less slices, flip-flops (FFs), and look-up tables (LUTs) than that in previous research. The clock frequencies used were as low as 250 kHz and 2 MHz.

Keywords: Verilog, Digital receiver, Zigbee, IEEE 802.15.4 standard, FPGA.

1. Introduction

Nowadays, Zigbee has attracted increasing research interest. Zigbee is a low-data-rate, low-power, and low-cost wireless networking protocol based on the IEEE 802.15.4 standard for wireless personal area networks (WPANs) [1]. The Zigbee protocol stack is built on top of IEEE 802.15.4, which defines the media access control (MAC) and physical (PHY) layers. It supports the frequency bands 868 MHz for European countries, 915 MHz for the United States, and 2.4 GHz for worldwide. A Zigbee end device can operate for months or even years without battery replacement. The maximum data rate is 250 kbps, and up to 65,000 nodes

can be connected in a network. The transmission range is 10–100 m, based on the environment [2]. Li and Zhang [3] mentioned more than 100 well-known global coalitions of hardware and software companies committed to the development of Zigbee technology for various applications.

This research focuses on development of a Zigbee-based digital receiver through Verilog hardware description language (HDL) code as a new design method instead of using a very high speed integrated circuit HDL (VHDL) code. HDL is used to describe the logic functionality of a circuit and the behavioral aspects of a circuit function. Verilog is easy to write, and easy to read and understand because it is similar to C. It is also easier to learn than VHDL. Spartan3E field programmable gate array (FPGA) is used as an implementation platform to verify the design. FPGA is a well-known prototype platform that can be reprogrammed and reconfigured for different designs without exchanging the hardware. Any design implemented on FPGA does not require the layout stage, design rules check (DRC), layout versus schematic (LVS), and input output (IO) buffer. Therefore, design time and overall cost excluding the mask cost, fabrication cost, package cost, and the test cost, are reduced. Using the advantages of Verilog and FPGA, this work provides evidence on the faster simulation of a digital receiver and the reduced design area. The next section presents detailed information on this paper including a comparison of previous research in terms of design approach, clock frequency, and configuration obtained during implementation [i.e., slice, flip-flop (FF), and look-up-table (LUT)].

This paper is organized as follows. Section 2 provides the background of Zigbee, FPGA implementation, and digital receiver. Section 3 presents a literature review on previous research. Section 4 describes the design methodology, and Section 5 presents the results and discussion of the simulation and the measurement for the digital receiver. Finally, Section 6 concludes the paper.

2. Background

2.1. Zigbee

The development of Zigbee standard and products is conducted by Zigbee Alliance, which consists of more than 270 companies including Freescale, Ember, Mitsubishi, Philips, Honeywell, and Texas Instruments [4]. The development is based on different applications, including smart energy, commercial building automation, home automation, health monitoring, industrial monitoring, and telecommunication. Temperature, pressure, flow, humidity, vibration, machine condition, and operating equipment can be monitored and controlled within a Zigbee network.

Figure 1 shows Zigbee wireless networking protocol layers. The protocol layers are based on the open system interconnect (OSI) basic reference model. These layers allow a layer affected by change to be replaced or modified rather than change the entire protocol [5]. Figure 1 shows that the MAC and PHY layers are defined by IEEE 802.15.4 standard [6]. Meanwhile, the networking, application, and the security layers of the protocol are defined by Zigbee standard. The Zigbee wireless networking protocol combines Zigbee and IEEE 802.15.4 standards. Therefore, any Zigbee-compliant device conforms to the IEEE 802.15.4 as well.

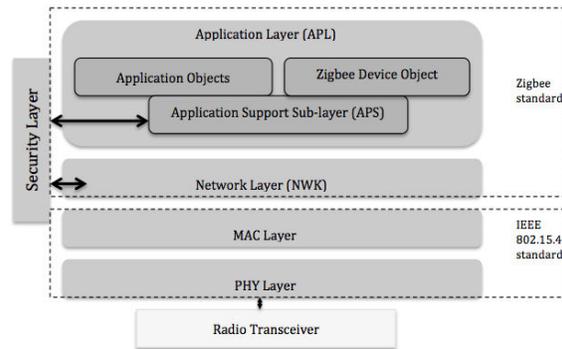


Fig. 1. Zigbee Wireless Networking Protocol Layers.

Devices in Zigbee wireless network perform three roles: (1) coordinator – principle controller of the personal area network (PAN), (2) router – relay of messages, and (3) end device – does not act as a coordinator. Zigbee has the least memory size, cheapest product, and least processing capabilities and features. Zigbee also possesses three network topologies: star, mesh, and tree topologies. In the star topology, every device in the network can communicate only with the coordinator. In the mesh topology, any device is allowed to communicate with another device directly or by taking advantage of the routing-capable device. Therefore, the reliability of wireless connections could be increased because the mesh network can create and modify routes dynamically. In the tree topology, a Zigbee coordinator acts as the root of the tree, where a coordinator or router can act as a parent device and accept association from other devices in the network. An end device can act as a child only because it is not capable of routing.

The IEEE 802.15.4 developed the MAC and PHY layer frame formats, as shown in Fig. 2. The preamble field is used by the transceiver to obtain chip and symbol synchronization with an incoming message [6]. The length of the preamble is 4 bytes (32 bits). The start of the frame delimiter (SFD) field indicates the end of the synchronization header (SHR) and the start of the PHY header (PHR). The length of SFD is 1 byte (8 bits). The frame length field is 7 bits in length and specifies the total number of octets contained in the PHY service data unit (PSDU). The PSDU field carries the data of the PHY packet. The maximum size of this field is 127 bytes (1,016 bits). Together, the SHR, PHR, and PSDU form the PHY protocol data unit (PPDU). The IEEE 802.15.4 standard defines four frame structures, including beacon, data, acknowledgment, and MAC command frames. A coordinator uses the beacon frame to transmit beacons. The data frame transfers data, and the acknowledgment frame confirms successful frame reception. Finally, the MAC command frame handles all MAC peer entity control transfers. However, this paper only uses the acknowledgment frame.

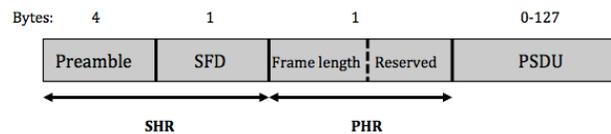


Fig. 2. General MAC and PHY Layer Frame Format of IEEE 802.15.4.

2.2. FPGA implementation

FPGAs were substantially improved in terms of sophistication and volume of production in the 1990s. FPGAs were primarily used in telecommunications and networking before they were used for consumer, automotive, and industrial applications [7]. In 2010, incorporating the ARM processor-based platform into a 28 nm FPGA family enables system architects and embedded software developers to apply a combination of serial and parallel processing. Therefore, the challenges in designing today's embedded systems can be addressed. The application of a combination of serial and parallel processing is required to meet ever-growing demands for highly complex functions. When embedded designers design in a familiar ARM environment, they benefit from the time-to-market advantages of an FPGA platform compared with the use of traditional design cycles associated with ASICs [8-12].

There are many FPGA manufactures such as Xilinx, Altera, Lucent, Actel, Lattice/Vantis, Quicklogic, and Atmel [13]. In [14], Xilinx claims that several market and technology dynamics are changing the ASIC/FPGA paradigm:

- Integrated circuit costs are rising aggressively.
- ASIC complexity lengthens development time.
- R&D resources and headcount are decreasing.
- Revenue losses for slow time-to-market are increasing.
- Financial constraints in a poor economy are driving low-cost technology.

These trends make FPGA a better alternative than ASIC for a larger number of higher-volume applications than they have been historically used for, to which the company attributes the growing number of FPGA design starts [14].

FPGA architecture consists of programmable logic components known as logic blocks (also called configurable logic block (CLB)), as well as a hierarchy of reconfigurable interconnects that allow the blocks to be wired together. Logic blocks can be configured to perform complex combinational functions or merely simple logic gates such as AND and XOR. These logic blocks include a few logical cells called slice. A typical cell consists of an LUT, full adder (FA), and a FF.

To define the behavior of FPGA, the user provides a schematic or an HDL design. Design is conventionally done through schematic capture. However, with the great progress in the development and the worldwide use of electronics in the 1990s, design houses began using HDL code. Millions of transistors can be fit through HDL onto a single IC chip within a short time. HDL can be used in several different ways: (1) synthesis of logic circuits known as a synthesizable code, (2) verification of a circuit known as behavioral code, and (3) netlist representation of a synthesized circuit called structural code. Thereby, the circuits need to be designed with HDL code before implementation process. HDL code can be either Verilog or VHDL. This paper used Verilog to design the digital receiver for Zigbee applications. Verilog is a descriptive language that describes the relationship between signals in a circuit [13].

Verilog declares the design inside a module. Editing a design with Verilog can be shorter and simpler than that described by schematics. The following are the advantages of Verilog over VHDL. (1) It is easily learned –Verilog does not

require prior experience with HDLs. (2) Easy use –Verilog can be easily used for specific design requirements, even by a first-time user. (3) Design integration – The designs done in Verilog are completed fast in the aspect of gates/manweek. (4) Design time – With Verilog, a netlist can be generated faster and simulated in a short time. These advantages make Verilog design able to be produced in the market immediately.

Spartan3E XC3S500E, a product from Xilinx, was used in this paper to implement a Zigbee-based digital receiver. The Spartan3E family with 1.2 V is available at a low cost and integrates many architectural features associated with high-end programmable logic. The combination of low cost and these features makes the Spartan3E family an ideal replacement for ASIC. Furthermore, the Spartan3E family is based on IBM and UMC advanced 90 nm with eight-layer metal process technology. Xilinx uses a 90 nm technology to decrease costs to under USD20 for a one-million-gate FPGA; this value represents cost savings as high as 80% compared with other competitive offerings [15]. The architecture of this family comprises five fundamental programmable functional elements: CLBs, input/output blocks (IOBs), embedded block RAM, multiplier blocks, and digital clock manager (DCM) blocks.

2.3. Digital Receiver

Sixteen channels based on IEEE 802.15.4 standard are available with an ample channel spacing of 5 MHz for 2.4 GHz-band applications. A direct-sequence spread spectrum (DSSS) with a digital spreading function representing pseudo-random noise (PN) chip sequences is employed [16], as shown in Table 1. The DSSS is used to improve the performance of the receivers in a multipath environment [17]. The transmitted signal in most practical scenarios may find several different paths to the receiver because of reflections, diffractions, and scatterings. In DSSS, every 4 bits of octet of a PPDU are grouped together to form a symbol. This symbol is mapped to a 32-bit chip represented by $c_0c_1c_2c_3\dots c_{31}$, which is also known as the PN sequence. A total of 16 different symbols and 32-bit chips are found.

Table 1. DSSS Method [6].

Data symbol (decimal)	Data symbol (binary) ($b_0 b_1 b_2 b_3$)	Chip values ($c_0 c_1 \dots c_{30} c_{31}$)
0	0000	11011001110000110101001000101110
1	1000	11101101100111000011010100100010
2	0100	00101110110110011100001101010010
3	1100	00100010111011011001110000110101
4	0010	01010010001011101101100111000011
5	1010	00110101001000101110110110011100
6	0110	11000011010100100010111011011001
7	1110	10011100001101010010001011101101
8	0001	10001100100101100000011101111011
9	1001	10111000110010010110000001110111
10	0101	01111011100011001001011000000111
11	1101	01110111101110001100100101100000
12	0011	00000111011110111000110010010110
13	1011	01100000011101111011100011001001
14	0111	10010110000001110111101110001100
15	1111	11001001011000000111011110111000

This paper focuses on the acknowledgment frame based on [6], as shown in Fig. 3. This frame is the simplest MAC frame format and does not carry any MAC payload. This frame is constructed from a MAC header (MHR) and a MAC footer (MFR). The frame control field and direct sequence number (DSN) form the MHR. The MFR is composed of a 16-bit frame check sequence (FCS). The FCS is based on the International Telecommunication Union (ITU) cyclic redundancy check (CRC) to detect possible errors in the data packet [18]. Both MHR and MFR are also known as PSDU, which becomes the PHY payload. The SHR, PHR, and PSDU form the physical packet known as the PPDU. An acknowledgment frame has 11 bytes (88 bits).

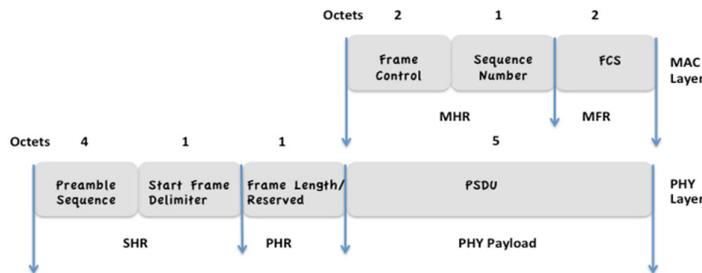


Fig. 3. Acknowledgment Frame.

The proposed digital receiver for Zigbee applications is designed to complete the digital transceiver, as shown in Fig. 4. The acknowledgment frames, which originate from the MAC sub-layer, are inserted into the CRC block. Every 4 bits are mapped into one data symbol in the bit-to-symbol block. The symbol-to-chip block performs the DSSS, where each symbol is mapped into a 32-chip PN sequence. Once completed, these chips are processed through an offset quadrature phase shift keying (OQPSK) modulator with half sine pulse-shaping block to reduce inter-symbol interference [19]. The OQPSK modulator is an improved version of the QPSK [20]. The significant difference between OQPSK and QPSK is that OQPSK processes the in-phase (I-phase) signal with the quadrature-phase (Q-phase) signal, which is delayed by half a cycle. This delay is important to avoid sudden phase-shift changes [21]. The modulated even chips are called the I-phase, and the modulated odd chip bits are called the Q-phase. Figure 5(a) shows that in forming the offset between I-phase and Q-phase chip modulation in the input data, the Q-phase chips shall be delayed by T_c with respect to the I-phase chips. T_c is the inverse of the chip rate. The chip rate is nominally 2 Mchip/s, which is 32 times the symbol rate [1].

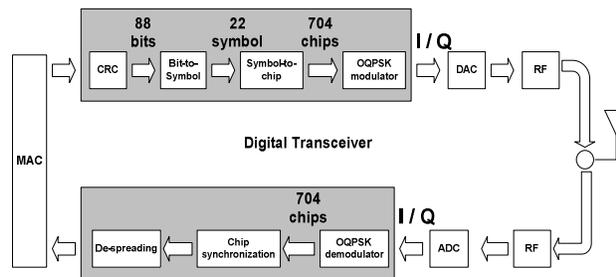


Fig. 4. Detailed Schematic of the Proposed Digital Zigbee Transceiver.

Bits or symbols are typically transmitted in the form of individual pulses of energy in the digital transmitter; this transmission is synonymous with the concept of binary information [22]. Therefore, IEEE 802.15.4 standard requires half-sine pulse-shaping to modify the shape of each binary pulse from the OQPSK modulator into half of a sinusoidal signal [5] before the pulse enters the digital-to-analog converter (DAC). The pulse-shaping block can improve the transmission efficiency by controlling the spectrum of the transmitted signal. This process is done by concentrating most of the signal power within the bandwidth of the information signal [23]. This block is described in IEEE 802.15.4 as

$$p(t) = \begin{cases} \sin\left(\frac{\pi t}{2T_c}\right), & 0 \leq t \leq 2T_c \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $2T_c$ is the width of the pulse. Figure 5(b) shows a sample of the baseband chip sequence with a half-sine pulse-shaping based on [24]. The resultant signal from the pulse-shaping block is converted to an analog signal before it is transmitted by the radio frequency (RF) transmitter. Then, the signal is received by the OQPSK demodulator, followed by chip synchronization and de-spreading blocks to regain the original data bits.

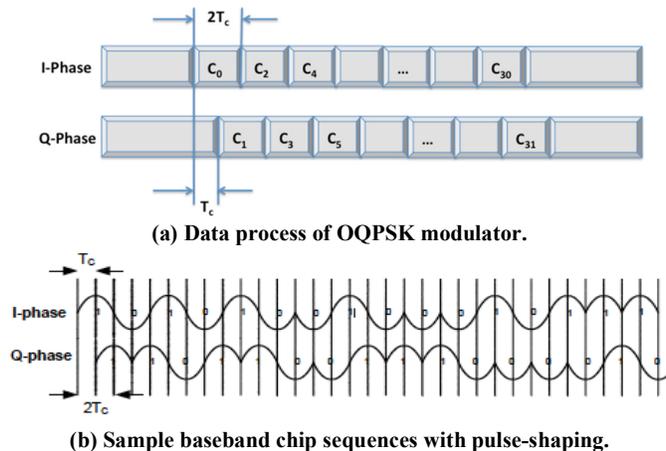


Fig. 5. The OQPSK Modulator with Half-Sine Pulse-Shaping.

3. Literature Review

Various studies design digital receivers using different methodologies, such as Matlab, VHDL, and schematic. However, Matlab can only be used for modeling and simulation. Schematic is not practical when the circuit is complex because the method needs a long design timeframe. Behavioral modeling of digital design goes through HDL, which is more time-efficient than other methods. Most important the HDL code can be simulated and implemented directly on FPGA as a prototyping device, or on an ASIC.

Di Stefano et al. [25] designed a simple receiver architecture using Matlab, Simulink, and VHDL. The architecture involves only four blocks, and the design

was implemented on Spartan3-200 FPGA. A 1 MHz low-pass filter reduces signal noise and co-channel interferences. The noncoherent OQPSK demodulator functions as a phase detector. The correlator and bit-timing block retains the original code, and it acquires timing and phase reference during the reception of each frame preamble. The frame-processing block then decodes the data field before sending it to the MAC or the host. Zigbee configuration requires less than 200 slices at 22 MHz sampling frequency.

Meng et al. [26] proposed a digital receiver architecture for Zigbee applications. The architecture comprises six blocks: carrier synchronizer, IF downconverter, filter, quadrature demodulator, chip synchronizer, and despreading. The IF downconverter changes IF signals from the ADC to the I-phase, earlier I-phase, later I-phase, and Q-phase signals. The filter block determines the phase offset between the carrier and the local oscillator. The quadrature demodulator then removes the phase offset. The chip synchronizer synchronizes the I-Q signals. The despreading block recovers the bits after the correct chip synchronization is obtained. This receiver was designed with VHDL and implemented on Xilinx Virtex-4 LX60 FPGA. The configuration obtained 3,047 slices out of 26,624; 3,659 FFs out of 53,248; 4,125 four-input LUTs out of 53,248; and 24 multipliers out of 64.

Kim et al. [27] successfully designed and fabricated a low-complexity demodulation scheme for a Zigbee receiver with 0.18 μm CMOS standard cell library. Multiple active correlators for demodulation are replaced with a matched filter-based cross-correlator. The demodulator shares correlators with a synchronization unit, which requires only a few additional control units. This correlator sharing reduces the total complexity. The proposed receiver reduced 36 nontrivial multipliers to 6; 338 adders to 259; and 59 k to 27 k logic gates over the existing hardware architecture from IEEE Std.802.15.4 [28].

Chen and Ma [29] designed and fabricated a different architecture of Zigbee receivers using TSMC 0.18 μm technology. The receiver works in three steps: packet detection, synchronization, and data recovery. At the packet-detection stage, other blocks are turned off until the packet is detected. At synchronization, the carrier frequency synchronization block is turned on to estimate the frequency error by preamble, and the phase compensation block works for phase rotation. The despreading block collects the packet information. Finally, at the data-recovery stage, the symbol-to-bit block recovers the data bit stream for media access control (MAC). The phase-tracking block begins to track the phase error. In the present paper, the packet error rate (PER) can achieve 0.01 at an SNR lower than 5 dB. This design has a chip area of 1.63 $\text{mm}^2 \times 1.63 \text{mm}^2$ (including the transmitter and the receiver) with a gate count of 78 k.

Bernier et al. [30] designed an ultra low-power Zigbee digital receiver using 130 nm CMOS technology. The 4 MSps 3-bit I/Q streams are half-sine-filtered before they are feeded to the synchronization block and the decoding module. The synchronization block recovers the symbol clock using a code-matched filter and a recursive channel filter. The code-matched filter is implemented by eight partial correlation banks. The recursive channel filter increases the SNR by averaging the samples over symbol periods. The SFD then locks the symbol clock, and the required symbol correlations are performed for each symbol period. The sample stream is decimated by a factor of 2, which means that a single complex sample

per chip is retained. At 1.2 V, the design drains 5.4 mW in receiver active modes and achieves 1% PER for a -81 dBm input power.

Using six blocks with Matlab and CoWare's signal processing designer, Zhang et al. [31] designed a digital receiver. The chip recoverer in the proposed architecture regains the two chip streams from the I-Q input signals. The chip recoverer is controlled by a chip synchronizer to align the chips. A received-signal strength indicator (RSSI) detects the presence of a valid input signal. The I-Q channel detector identifies the Q-channel chip stream between the two chip streams and locates the bit stream head. I-channel chip data are ignored to simplify the receiver implementation. Only Q-channel chip data are used to extract the bit data with a bit-synchronizer module. Finally, the bit data are processed into symbol data by a symbol recoverer. The receiver was implemented on an element computing array (CXI ECA-64) platform, which delivers faster reconfiguration time and higher-computational-density FPGAs with similar computational power. The final results show that 84% logical operations used, with 18% of the memory units implementing delay functions and shift registers.

Wang et al. [21, 32] designed another Zigbee digital receiver for 2.4 GHz band. The authors implemented the receiver on FPGA and fabricated it using 0.18 μm CMOS technology. The architecture, which involves eight blocks, is quite complicated. The packet detector discriminates whether the incoming signal is data or noise. The phase difference detector determines the phase difference of each sample data. The downsampling block finds the maximum phase difference and performs the downsampling. Meanwhile, the frequency-offset compensation computes and compensates the frequency offset. The noncoherent demodulator uses minimum shift-keying (MSK) scheme to perform demodulation. MSK is assumed to be similar to OQPSK. The preamble removal block acquires and removes the preamble from the PPDU packet. The despreading block despreads each chip PN sequence to the symbol data. The confirmed SFD block acquires the PSDU length and notifies the MAC layer of the PSDU obtained from the receiver. All process corners (0 $^{\circ}\text{C}$, +100 $^{\circ}\text{C}$) and (SS, TT, FF) models were simulated to verify the design. The fabrication results shows that the PER is less than 1% at 8 MHz system clock.

Table 2 summarizes and compares the design methodologies and measurement results of all seven papers for the past six years. The comparison shows that [26] possesses complete measurement results in terms of slice, LUT, and multiplexer usages. However, the clock frequency used is the highest among these studies. Therefore, signal integrity may be lost because of the rise and fall times of the output signals. These signals decrease because the devices are designed to operate fast and use small silicon manufacturing process. Hence, the frequency clock used in this paper is reduced to avoid loss of signal integrity.

Table 2. Comparison of Measurement Results in Previous Works.

References	[25]	[26]	[27]	[29]	[30]	[31]	[32]
Implementation	FPGA	FPGA	ASIC	ASIC	ASIC	CXI ECA-64	FPGA, ASIC
Clock frequency (MHz)	22	48	NA	4	8	NA	8
Slice (%)	10	11	-54*	(78 k)	(30 k)	NA	NA
LUT (%)	NA	8	NA	NA	NA	84	NA
Multiplexer (%)	NA	38	-83*	NA	NA	25	NA

Note: * Reduced over the existing hardware architecture.

4. Design Methodology

The design methodology for the Zigbee digital receiver is described in detail. The methodology is divided into three parts: OQPSK demodulator, synchronization block, and de-spreading block. These blocks were modeled with Verilog through Xilinx ISE. The block's modules were combined as sub-modules before synthesis and simulation. Finally, the Verilog module was implemented on Spartan3E FPGA for real environment verification.

4.1. OQPSK demodulator

The proposed design methodology for the OQPSK demodulator is as follows:

- The input data comprise 704 chips per acknowledgment frame, with 352 chips each for the I-phase (even chips) and the Q-phase (odd chips). The number of data chips is calculated based on (2) [17]. The input frequency is very low at 2 MHz.

$$[88 \text{ bits} \div 4 \text{ symbols}] \times 32 \text{ chips} = 704 \text{ chips} \quad (2)$$

- With the same frequency, each chip of input data is processed based on

$$\begin{aligned} \text{output_data}[2k - 1] &= \text{I_phase}[2k - 2] \\ \text{output_data}[2k] &= \text{Q_phase}[2k - 1] \end{aligned} \quad (3)$$

where $1 \leq k \leq 352$.

Based on this equation, each even chip of output data is registered as C_0, C_2, \dots, C_{704} , and each odd chip is registered as C_1, C_3, \dots, C_{703} , with a total 352 chips each for the I-phase and the Q-phase. These data chips will be the input for the next block in the Zigbee receiver.

Figure 6(a) shows the OQPSK demodulator structure. The $\text{data_in}[0]$ and $\text{data_in}[1]$ represent the even and odd input signals, respectively. The load_demod , reset_demod , shift_demod , and process_demod are the pins of the input signals. Meanwhile, clk is the clock frequency of 2 MHz. The data_out represents the output signal.

4.2. Chip synchronization

The proposed design methodology of the chip synchronization block is described as follows:

- The input data for this block comprise 704 chips. Every 32 chips of the input data are mapped into 1 symbol data with the DSSS, as shown in Table 1. At a frequency of 250 kHz and 2 MHz, the numbers of the symbols produced as output data are calculated as follows:

$$704 \text{ chips} / 32 = 22 \text{ symbols} \quad (4)$$

- The basic process of this block is described briefly in Eq. (5), where the last 32 input chips are the most significant bits (MSBs) of the output data as symbol 22th. These output symbol data will be the input data for the de-spreading block.

$$\text{chip} [\{704 - 32m\} : \{673-32m\}] \Rightarrow \text{symbol} [22-m] \quad (5)$$

where $0 \leq m \leq 21$.

Figure 6(b) shows the schematic block of chip synchronization. *clk1* and *clk2* are the clock frequencies of 2 MHz and 250 kHz, respectively. The other input ports are *enable_out*, *input_chip*, *load_chip*, *reset_chip*, and *shift_out*. The *symbol1_out(3:0)* until *symbol22_out(3:0)* are the output ports for this block.

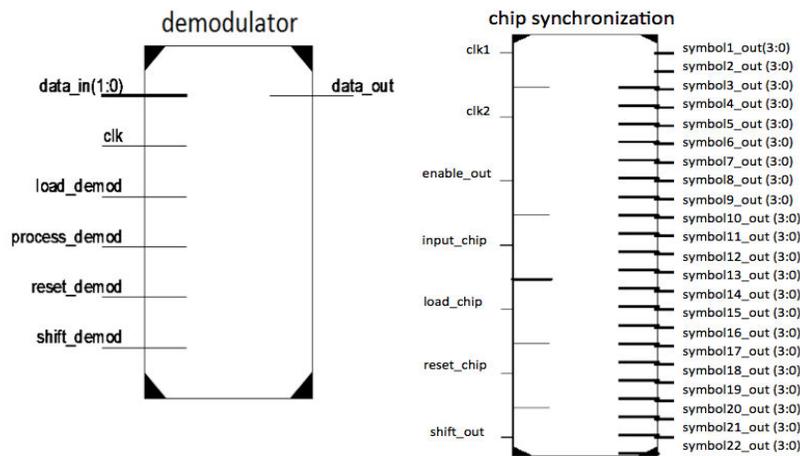
4.3. De-spreading

The proposed design methodology for the de-spreading block is described as follows:

- The input data comprise 22 symbols. For each symbol, the input data are mapped into 4 bit data. The logic value of these bits is based on Table 1.
- With a frequency of 250 kHz, the total number of output bits produced is obtained from the following equation. These output data form the PPDU packet for the acknowledgment frame.

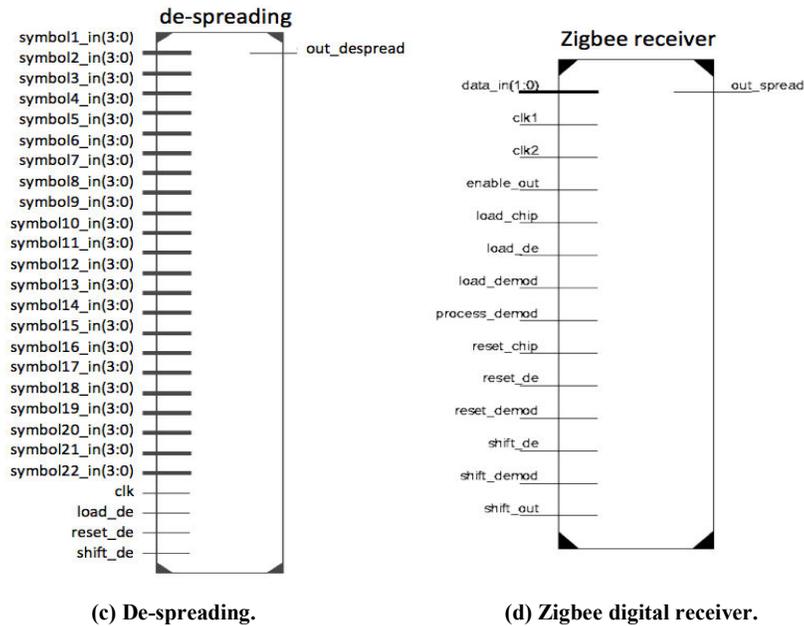
$$22 \text{ symbols} \times 4 = 88 \text{ bits} \quad (6)$$

Figure 6(c) shows the schematic of the de-spreading block. The frequency clock represented by *clk* is 250 kHz. The output data in bits are sent to *out_despread* port serially within 352,000 ns. The input ports comprise *clk*, *load_de*, *reset_de*, *shift_de*, and 22 ports of *symbol_in(3:0)*.



(a) OQPSK demodulator.

(b) Chip synchronization.



(c) De-spreading.

(d) Zigbee digital receiver.

Fig. 6. Schematic of the Blocks.

4.4. Digital Receiver

The three blocks, namely, the OQPSK demodulator, chip synchronization, and de-spreading blocks, are integrated into one top Verilog module to form a digital receiver for Zigbee applications. The schematic of this block is shown Fig. 6(d). The two frequency clocks used are *clk1* and *clk2* with 2 MHz and 250 kHz, respectively. The block has 12 input ports and one output port, where the data are processed serially. The *out_spread* has a total of 88 bits. Meanwhile, Fig. 7 depicts part of the Verilog code for this block. The top module was named *zigbee_receiver*.

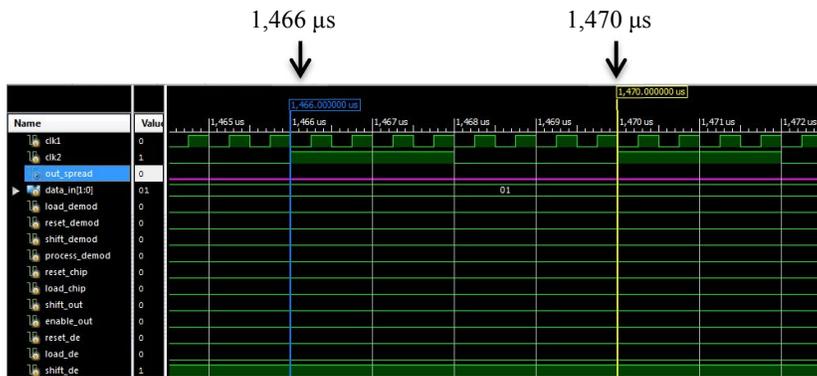
```
//top_module
module zigbee_receiver (data_in, load_demod, reset_demod, shift_demod, process_demod,
reset_chip, load_chip, shift_out, enable_out, reset_de, load_de, shift_de, clk1, clk2,
out_spread);
input [1:0] data_in;
input load_demod, reset_demod, shift_demod, process_demod, reset_chip, load_chip,
shift_out, enable_out, reset_de, load_de, shift_de, clk1, clk2;
output out_spread;
wire in_chip;
wire [3:0] symbol1, symbol2, symbol3, symbol4, symbol5, symbol6, symbol7, symbol8,
symbol9, symbol10, symbol11, symbol12, symbol13, symbol14, symbol15, symbol16, symbol17,
symbol18, symbol19, symbol20, symbol21, symbol22;
```

Fig. 7. Part of the Verilog Module for Zigbee Digital Receiver.

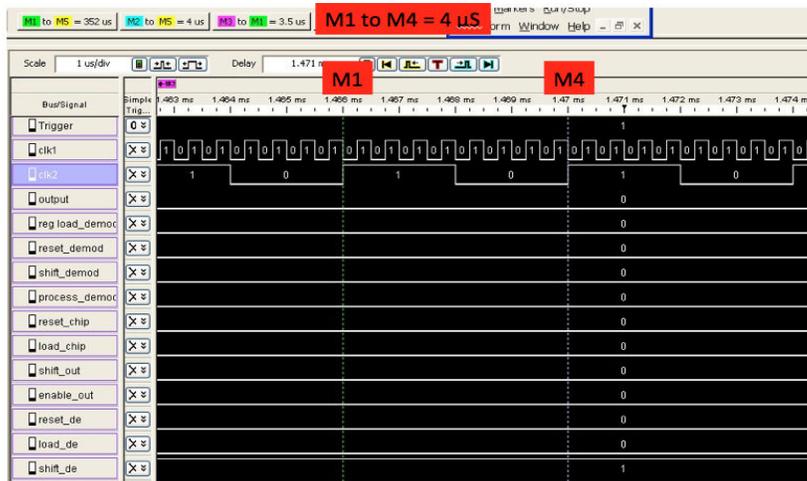
5. Results and Discussion

Simulation and implementation were performed for the digital receiver. The output waveforms were compared to verify their functionality. The simulation waveform in Fig. 8(a) shows that the first single bit of output data (*out_spread*) is starting at 1,466 μ s and ending at 1,470 μ s. Thus, 1 bit data length is 4 μ s with logic '0'. This also means that the data rate of Zigbee digital receiver is totally 250 kbps.

Meanwhile, the measurement waveform in Fig. 8(b) shows that the first single bit of output data (*output*) is also starting at 1.466 ms (M1) and ending at 1.470 ms (M4). Thus, 1 bit data length is 4 μ s also with logic '0'. The results proved that the simulation result for the first single bit is similar to the first single bit of measurement result.



(a) Simulation waveform.

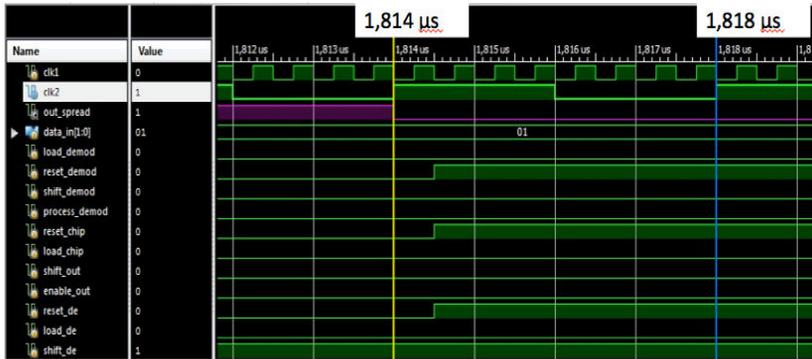


(b) Measurement waveform.

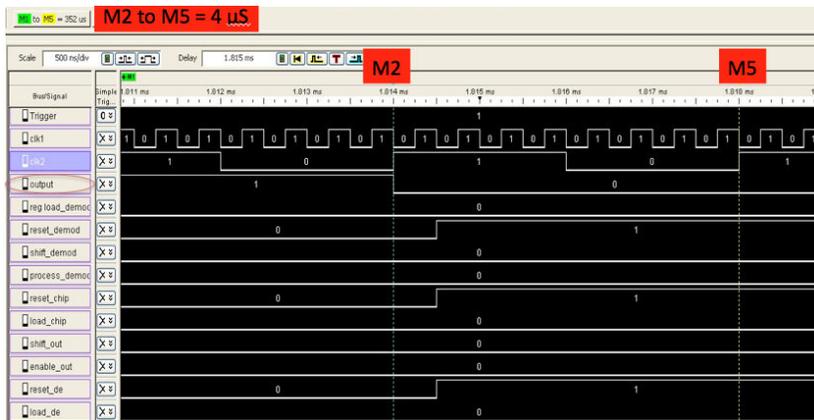
Fig. 8. Output Data for First Single Bit of Zigbee Digital Receiver.

In Fig. 9(a) the last single bit of output data (*out_spread*) starting from 1,814 μ s until 1,818 μ s. It shows that the last single bit has a length of 4 μ s with logic '0'. The last output bit is at the 88th position of overall output.

The measurement waveform in Fig. 9(b) also shows that the last single output bit (*output*) has a length of 4 μ s starting from M2 (1.814 ms) until M5 (1.818 ms). This bit is also has logic '0'. From Figs. 9(a) and (b), it is proven that the simulation and measurement results are similar.



(a) Simulation waveform.



(b) Measurement waveform.

Fig. 9. Output Data for Last Single Bit of Zigbee Digital Receiver.

Figure 10(a) shows that for the simulation waveform, the output data have a total of 88 bits represented by *out_spread*. The bit data started from 1,466 μ s and finished at 1,818 μ s. The length of the output data is 352 μ s for 88 bits. Thus, 1 bit data length is 4 μ s.

implementation result was compared with that in [26], which is summarized in Table 3. The proposed digital receiver design using Verilog reduced the use of slices by 14%, FFs by 18%, LUTs by 22%, and multiplexers by 92%. Thus, the proposed design size is smaller than the digital receiver designed by [26] using VHDL. The clock frequency used in this paper is also lower than that in [26] to avoid loss of signal integrity.

Table 3. Comparison of Implementation Results between the Proposed Design and that in [26].

References	Proposed Design	[26]
Design Approach	Verilog	VHDL
FPGA Family	Spartan3E	Virtex4
Clock Frequency	2 MHz & 250 kHz	48 MHz
Slices	2,626	3,047
FFs	2,993	3,659
LUTs	3,228	4,125
Multiplexers	2	24

6. Conclusion

This paper discusses the implementation of digital receiver for 2.4 GHz-band Zigbee applications on Spartan3E FPGA. Various digital receiver designs in the last six years were analysed in terms of performance. The study concludes that the proposed Verilog-based design reduces the design area, shortens the simulation time, and decreases the clock frequency compared with VHDL-based designs. Verilog also speeds up the design process and produces output quickly. These advantages are particularly important to engineers or designers who are on a tight deadline. Future work will implement the transmitter part using the Verilog code on FPGA. Thus, the receiver and the transmitter will be integrated to obtain a digital transceiver that can be tested over a 2.4 GHz Zigbee communication standard in a different transmission range.

Acknowledgment

This research was supported by Universiti Sains Malaysia Short-term Grant No. 304/PCEDEC/60312004.

References

1. Kluge, W.; Poegel, F.; Roller, H.; Lange, M.; Ferchland, T.; Dathe, L.; and Eggert, D. (2006). A fully integrated 2.4 GHz IEEE 802.15.4-Compliant transceiver for Zigbee applications. *IEEE Journal of Solid-State Circuits*, 41(12), 2767-2775.
2. Lee, J.-S.; Su, Y.-W.; and Shen, C.-C. (2007). A comparative study of wireless protocols: Bluetooth, UWB, Zigbee and Wi-Fi. *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 46-51.

3. Li, Y.; and Zhang, K. (2010). Research on application of Zigbee technology in flammable and explosive environment. *Scientific Research. Wireless Sensor Network*, 2(6), 467-471.
4. Elahi, A.; and Gschwender, A. (2010). *Zigbee wireless sensor and control network*. (1st Ed.) Prentice Hall, USA.
5. Farahani, S. (2008). *Zigbee wireless networks and transceivers*. Newnes, USA.
6. IEEE Std.802.15.4 (2006). Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs).
7. Maxfield, C. (2004). *The design warrior's guide to FPGAs. Devices, Tools and Flows*. (1st Ed.) Newnes, USA.
8. McConnel, T. (2010). ESC-Xilinx extensible processing platform combines best of serial and parallel processing. *EETimes*.
9. Cheung, K. (2011). Xilinx extensible processing platform for embedded systems. Available at: <http://fpgablog.com/posts/arm-cortex-mpcore/>.
10. Nass, R. (2011). Xilinx puts ARM core into its FPGAs. Available at: <http://www.eetimes.com/electronics-products/processors/4115523/Xilinx-puts-ARM-core-into-its-FPGAs>.
11. Leibson, S. (2011). Xilinx redefines the high-end microcontroller with its ARM-based extensible processing platform - Part 1. Available at: <http://www.design-reuse.com/industryexpertblogs/23302/xilinx-arm-based-extensible-processing-platform.html>.
12. Wilson, R. (2011). Xilinx acquires ESL firm to make FPGAs easier to use. Available at: <http://www.electronicweeky.com/Articles/31/01/2011/50386/xilinx-acquires-esl-firm-to-make-fpgas-easier-to-use.htm>.
13. Ciletti, M.D. (1999). *Modeling, synthesis, and rapid prototyping with the Verilog HDL* (1st ed.). Prentice-Hall, USA.
14. Erjavec, T. (2009). Introducing the Xilinx targeted design platform: Fulfilling the programmable imperative. *White Paper*.
15. Parnell K.; and Mehta, N. (2004). *Programmable logic design quick start handbook*. Xilinx. Inc.
16. Oualkadi, A.E.; Andendorpe, L.V.; and Flandre, D. (2007). System-level analysis of O-QPSK transceiver for 2.4 GHz band IEEE 802.15.4 Zigbee Standard. *Proceedings of the 14th International Conference on Mixed Design*, 469-474.
17. Lee, W.C.Y. (1998). *Mobile communication engineering, theory and application* (2nd Ed.). McGraw-Hill, New York.
18. International Telecommunication Union (2008). Available at: <http://www.itu.int>.
19. John, N.; Anthony, C.; and Gary, L. (2003). *CMOS RFIC architectures for IEEE 802.15.4 networks*. Cadence Design Systems. USA: Columbia.
20. Fang, L.; Xizheng, K.; and Qiang, L. (2007). Design and implement of OQPSK modulator based on FPGA. *Proceedings of the Eighth International Conference on Electronic Measurement and Instruments*, 4-929 – 4-933.
21. Wang, C.C.; Huang, J.M.; Lee, L.H.; Wang, S.H.; and Li, C.P. (2007). A low-power 2.45 GHz Zigbee transceiver for wearable personal medical

- devices in WPAN. Available at: <http://ieeexplore.ieee.org/iel5/4145986/4099325/04146217.pdf>.
22. Gentile, K. (2002). The care and feeding of digital, pulse-shaping filters. Available at: <http://www.deetc.isel.ipl.pt/analisedesainai/com/bibliografia/0402Gentile50.pdf>.
 23. Crockett, L.H.; MacEwen, N.C.; Pfann, E.; and Stewart, R.W. (2005). Pulse shaping for RF communications in wireless sensor networks. *Conference Record of the Thirty-Ninth Asilomar Conference on Signals, Systems and Computers*, 442-446.
 24. Ahmad, R.; Sidek, O.; Wan Hassin, W.M.H.; Mohd, S.K.K.; and Mohamad, F. (2010). Development of a Verilog-based pulse-shaping block for Zigbee applications. *Proceedings of International Conference on Modeling, Simulation and Control (ICMSC)*, 65-68.
 25. Di Stefano, A.; Fiscelli, G.; and Giaconia, C.G. (2006). An FPGA-based software defined radio platform for the 2.4 GHz ISM band. *Research in Microelectronics and Electronics*, 73-76.
 26. Meng, T.; Zhang, C.; and Athanas, P. (2007). An FPGA-based Zigbee receiver on the Harris software defined radio SIP. *Proceedings of the SDR 07 Technical Conference and Product Exposition*, Denver, Colorado, 1-5.
 27. Kim, W.; Jung, Y.; Lee, S.; and Kim, J. (2008). Low complexity demodulation scheme for IEEE 802.15.4 LR-WPAN systems. *IEICE Electronics Express*, 5(14), 490-496.
 28. IEEE Std.802.15.4 (2003). IEEE standard for wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs).
 29. Chen, K.-H.; and Ma, H.-P. (2008). A low power Zigbee baseband processor. *Proceedings of 2008 International SoC Design Conference*, I-40 - I-43.
 30. Bernier, C.; Hameau, F.; Billiot, G.; de Foucauld, E.; Robinet, S.; Lattard, D.; Durupt, J.; Dehmas, F.; Ouvry, L.; and Vincent, P. (2008). An ultra low power SoC for 2.4 GHz IEEE802.15.4 wireless communications. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4681883&tag=1, 426-429.
 31. Zhang, C.; Athanas, P.M.; Reed, J.H.; Martin, T.L.; and Buehrer, M.R. (2008). *An ECA-based Zigbee receiver*. M.Sc. thesis, Virginia Polytechnic Institute and State University, Virginia.
 32. Wang, C.-C.; Sung, G.-N.; Huang, J.-M.; Lee, L.-H.; and Li, C.-P. (2010). A low-power 2.45 GHz WPAN modulator/demodulator. *Microelectronics Journal*, 41(2-3), 150-154.