Modelling and Simulation of Baseband Processor for UHF RFID Reader on FPGA

I. Ismail, A.Ibrahim

Abstract— **A baseband processor of UHF RFID reader that presented in this paper is based on International Organization for Standardization and International Electrotechnical Commission (ISO/IEC 18000-6) protocol. The protocol also known Electronic Product Code (EPC) Class-1 Generation-2 Radio Frequency Identification (RFID) protocol. The baseband processor consists of PIE encoder, FM0 decoder and Miller decoder. The behavior of the PIE encoder, FM0 decoder and Miller decoder architecture is realized by derivation of Verilog Hardware Description Language (HDL) code in Quartus II software. Utilizing the ModelSim-Altera, the encoder and decoder architecture is simulated to observe its functionality. The designing of the encoder and decoder is intended for uses in Ultra High Frequency (UHF) RFID passive interrogator.**

Index Terms— **RFID, UHF Reader, FPGA, Baseband Processor.**

I. Introduction

RADIO Frequency Identification (RFID) is an automatic identification technology that uses radio waves to transmit the identity of objects or people in the form of a unique serial number. This technology does not require either contact or line of sight for communication between reader and tag. RFID is probably the best choice for automatic identification due to contactless, wireless, multiple tag identification, high data rate, long read range, lowest cost, harsh operating environment and re-programmability of the tag.

Nowadays, RFID application is growing rapidly in many fields such as manufacturing, animal identification, logistics, transportation payment, airport baggage handling, object tracking systems, antifraud systems, auto registration and medical treatment [1-4]. Fig. 1 below illustrates the basic RFID system that consists of three components which are transponder, interrogator and the middleware.

The transponders commonly refer as RFID tags contain an electronic microchip, antenna and the encapsulating material. The microchip encoded with a unique serial number that link to entries in a database. The coil antenna uses to transmit the data as well as for communication between reader and tag. The RFID tag will be placed on the object to be identified.

The RFID reader is a device used to transmit to and receive information from the RFID tag. RFID reader has an antenna that emits the radio wave to read the RFID tag and then passes the data to a computer for processing. The reader's antenna can be external or internal. A middleware or host computer sends the instruction to the reader as well as utilizing the data obtained by the reader in some useful manner.

An Electronic Product Code (EPC) Class-1 Generation-2 Radio Frequency Identity Protocol for communications at 860 MHz to 960 MHz is the specification that defines the physical and logical requirements for a passive-backscatter, interrogator-talks-first (ITS) and Radio Frequency Identification (RFID) system [5]. The specification is also commonly known as Gen2. The Gen2 becomes ISO/IEC 18000-6 type C after the amendment of ISO/IEC 18000-6:2004/Amd 1:2006 [6].

Manuscript received April 3, 2013.
This work was supported under Dana Kecemerlangan from UiTM. Grant No: 600-RMI/DANA 5/3/RIF (37/2012).

I.Ismail Ismarani Ismail is a Senior lecturer at the Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM).

A.Ibrahim is a PhD student at Faculty of Electrical Engineering, Universiti Teknologi Mara. (email:azlina.uitm@gmail.com/zahra_lin@yahoo.com)

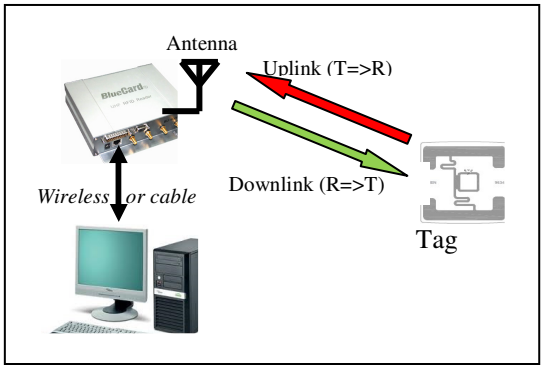

Fig. 1. Basic RFID System

In the ISO/IEC 18000-6 protocol, the reader modulates a signal in the UHF frequency range which is between 860MHz to 960MHz in order to communicate with a tag. The reader emits an RF carrier signal to power up the tag. The tag absorbs a small portion of the emitted energy and after acquiring sufficient energy from the reader, tag transmitter will send the modulated information to the reader. The reader then demodulates the received signal from the tag and decodes the signal to the binary digit for further processing [5-6].

In passive UHF (860 MHz to 960 MHz) RFID system, reading distance is limited by several factors such as propagation environment, tag characteristics and RFID reader parameter itself [7]. The performance of the reader system depends on baseband signal processor [8]. The baseband system is very important and act as Central Controller that control the communications between the reader and tags. It is can be said as a brain of the RFID reader. It is responsible for encoding PIE data, Decoding FM0 or Miller data, accessing memory, adjusting the clock generator, controlling command send to the tags and processing the tag's reflected information.

There are several schemes that can be used in designing of an RFID reader. The scheme uses are microcontroller unit (MCU) [9], Field Programmable Gate Array (FPGA) [10-11] and Digital Signal Processing (DSP) chip as a main controller [8]. There are some researcher uses combination of both FPGA and DSP schemes for their design [12-13]. FPGA technology has an advantage such as power consumption is lower, the cost is lower, more competitive with high end MCUs, can program any functions, and also designer can add custom features and change the design features easily.

The investigation of the usage of FPGA technology in RFID system has been already done in numerous research and study. Ching-Chien et al. (2008) [14] design a Miller-Modulated Subcarrier (MMS) and FM0 encoding scheme by using Verilog HDL and implemented using FPGA for UHF RFID applications. Tung et al. (2008) [15] presented physical layer encoding and decoding hardware blocks by using waveform features. The features used as an input to automatically produce an encoding and decoding hardware block. The physical layer features describe by using VHDL language that intended for RFID tags. Khan et al. (2009) [16] presented design strategy for FM0 and miller encoder based on Finite State Machine (FSM) that can be used as a core component in UHF RFID tag emulator. Li et al. (2006) [17] present analysis and simulation based on EPC Class-1 Generation-2 UHF RFID protocol.

Several works have been done on designing RFID reader baseband processor by using proprietary communication protocols between the reader and tag. Li et al. [10] presents a implementation of UHF RFID reader baseband module based on ARM processor and FPGA chip. Joon Goo et al [11] propose a multi-protocol baseband processing based on FPGA. Khannur et al. [18] presents a highly integrated UHF RFID reader IC on CMOS-based which is compatible with ISO 18000-6A/6B/6C UHF RFID protocol. Shuang et al. [20] propose a reconfigurable baseband processor architecture that compatible with ISO 18000-6 protocol suite by using CMOS process. Jing et al. [19] present an implementation of digital baseband system for UHF RFID reader that conform with ISO 18000-6C protocol by ASIC design.

This research will use FPGA scheme in order to built reader baseband processor for UHF RFID reader. It is because the implementation of UHF RFID reader baseband processor on FPGA will give benefit for easy upgrading the standard protocol and this research will be the prelude for the next researcher in order to upgrade the standard in the future and increases the RFID performance in Malaysia.

The objectives of the research are to develop a baseband processor for UHF RFID reader including PIE encoder, FM0 decoder and Miller decoder architecture of ISO/IEC 18000-6 reader protocol on Field Programmable Gate Array (FPGA). Then, observe its functionality. The baseband processor modeling was developed by

using Verilog Hardware Description Language (HDL) and simulated using ModelSim-Altera in order to check the functionality of the model.

In section II, an overview of the ISO/IEC 18000-6 RFID standard is given and characteristic of the reader to tag (R=>T) communication and tag to reader (T=>R) communication are explained. Section III explains the design methodology followed by results and discussion in section IV. The last section which is section V, some conclusion and future works is drawn.

II. System Requirements

The paper review physical layer architecture for the reader to tag and tag to reader communication. Fig. 2 shows an operational block diagram of general communications between reader and tag. The diagram consists of the reader transmitter to the tag receiver communications and tag transmitter to the reader receiver communications. The communication between tag and reader is established when the reader energizes the tag by transmits a continuous wave (CW) signal to the tag and subsequently the tag transmits the backscattered data to the reader.

In the Reader transmitter architecture consists of encoder and modulator module. Reader send data to tag by modulating an RF carrier using double-sideband amplitude shift keying (DSB-ASK), single-sideband amplitude shift keying (SSB-ASK), phase-reversal (PR-ASK) with a pulse-interval encoding (PIE) format [6]. Reader transmits and receives analog waves and then turn them into bits of digital information which are a string of zeros and one [4].

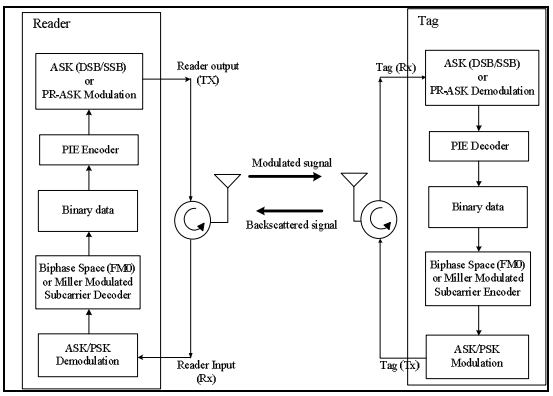

Fig. 2. Communication between reader and tags for UHF RFID systems

In the tag receiver architecture consists of demodulator and decoder module and in the tag transmitter architecture consists of encoder and modulator module. The tag demodulates the signal received from the reader antenna and decodes the signal to binary data for further processing. The binary data will be encode by using FM0 or Miller-modulated subcarrier encoding at the tag transmitter. The tag than transmit the backscattered signal to the reader. Reader will demodulate signal received from the tag antenna and decode the signal back to binary data for further processing. The operation will continue during the communication between the reader and tags.

The communications link between interrogators and tags is half-duplex, meaning that tags will not be required to demodulate interrogator commands while backscattering. Tags will not respond to a mandatory or optional command using full-duplex communications.

Data is passed to the tag by pulse the carrier wave at difference time interval to indicate the data 0 and data 1. Interrogator uses fixed data rate for the duration of sending data to tags. The data rate depends on the Tari value in the range of 6.25µs to 25µs [6]. PIE symbol that specified the Tari value for data 0 and data 1 shown in Fig. 3.

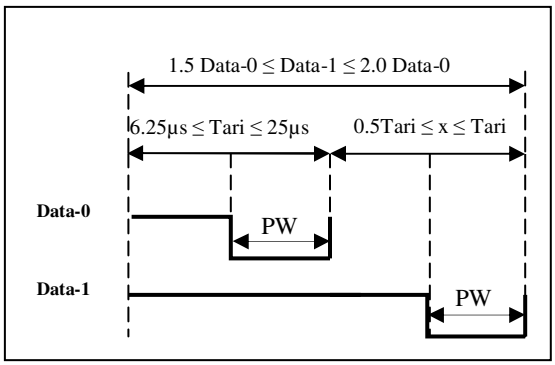

Fig. 3. PIE symbols of EPC Class-1 Generation-2 protocol

In reader to tag signalling, several timing intervals such as delimiter, tari, RTcal and TRcal need to be considered. Fig. 4 shows the relationships in time domain among the reference timing intervals when the interrogator starts the inventory round. An interrogator shall begin all readers to tag signalling with either a preamble or frame-sync. A preamble will precede a *Query* command and be the start symbol of an inventory

round. Subsequently, all other readers to tag signalling will begin with a frame-sync [6].

Reader to tag preamble consists of fixed length start with delimiter, data-0 symbol, RTcal symbol and TRcal symbol. Whereas Reader to tag frame_sync consists of all elements in preamble except TRcal symbol. An interrogator will set the value of RTcal equal to the sum of data-0 length and data-1 length.

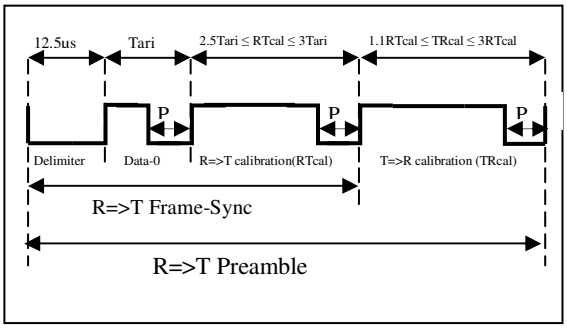

Fig.4. R=>T Preamble and R=>T Frame-Sync

The reader instructs the tag which method of data encoding to use when sending its data back. The data are sent from a tag would be in FM0 or Miller format. The reader commands the encoding choice either FM0 baseband encoding or Miller subcarrier encoding. FM0 is a bi-phase space that a transition occurs at the beginning of every symbol boundary. A data-0 is represented by an additional transition at the center of the symbol boundary and a data-1 represented by no additional transition at the center of symbol boundary. Fig. 5 shows FM0 symbols that indicate two kinds of data which is data-0 and data-1.

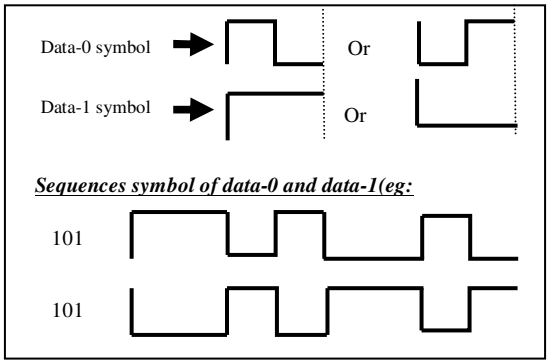

Fig.5. FM0 symbols of EPC Class-1 Generation-2 Protocol

The FM0 encoded message received from the tag will begin with one of the two preambles shown in Fig. 6. The choice depends on the value of the TRext bit in the Query command that send by the reader. The "v" indicates an FM0 violation (i.e a phase inversion should have occurred but did not).

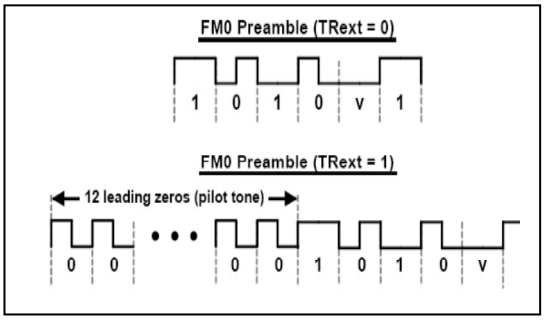

Fig. 6. FM0 T=>R preamble

Fig. 7 shows Miller modulated subcarrier symbols in sequences that indicate two kinds of data which is data-0 and data-1 for M=2 cycle per bit, M=4 cycle per bit and M=8 cycle per bit. In modulated subcarrier encoding, a transition occurs between two data-0s in sequence and also in the middle of data-1 symbol. A miller sequence can contain exactly two, four or eight subcarrier cycles per bit depending on the M value. Parameter M is a number of subcarrier cycles per symbol in the Query Command.

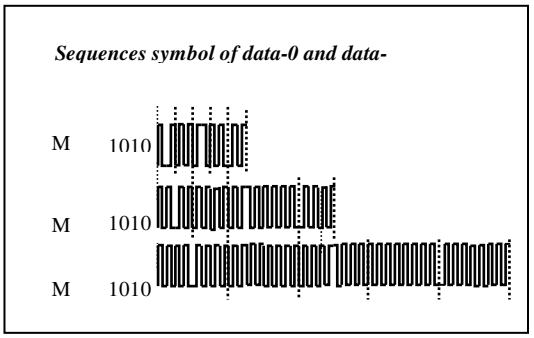

Fig.7. Miller symbols of EPC C1 Gen-2 Protocol

Same with FM0 encoding, the Miller message also begins with one of the two Tag to Reader preambles as shown in Fig. 8. The reader tells the tag which one to use by sending the Query Command that specifies the value of the TRext bit.

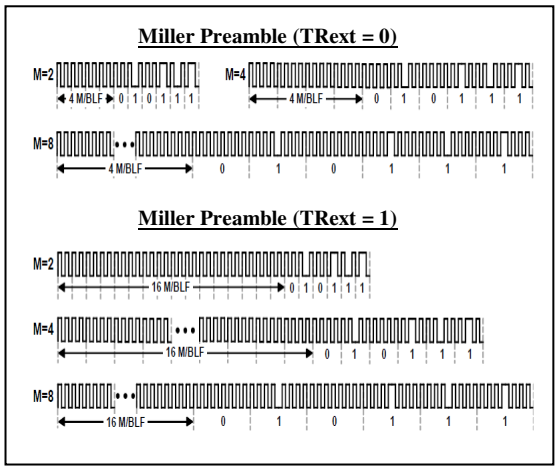

Fig.8. Subcarrier T=>R Preamble

The reader will specify a tag's Backscatter Link frequency (BLF) by sending the TRcal and a parameter called Divide Ratio (DR) in the Query Command. Equation (1) specifies the relationship between the BLF, TRcal and DR. The tags can backscatter data to reader at variety of data rate depending on the desired mode of operation sending by reader. The tag measures TRcal, divides by DR, and sends data to the reader at a rate given by equation (2) per symbol. Data rate also depands on the number of subcarrier cycles per symbol as specified by equation (3) below.

$$\text{BLF} = \frac{\text{DR}}{\text{TRcal}} \quad \text{----------} \quad (1)$$

$$\text{T} = \frac{\text{TRcal}}{\text{DR}} \quad \text{---------} \quad (2)$$

$$\text{Data Rate} = \frac{\text{BLF}}{\text{M}} \quad \text{----------} \quad (3)$$

The value of TRcal is between 1.1xRTcal to 3xRTcal. Whereby value of RTcal is depend on Tari value which is 2.5xTari to 3.0xTari. The Tari value is in the range of 6.25µs to 25µs. There are two values of DR which is 64/3(bit-1) and 8(bit-0). When the Query command sets the DR equal to bit-1, means that DR value send by reader is equal to 64/3 whereas DR value is equal to 8. The values of M setting by the reader in a Query Command also affect the tag to reader data rate. Data rate is equal to BLF divide by number of subcarrier cycles per symbol. For example, if the BLF equal to 100 KHz, FM0 provides a data rate of 100 Kbps, whereas MMS with multiplier of M=4 provides data rate equal to 25 Kbps.

The MMS offers some advantages over the FM0. In spectral terms, the energy in an MMS signal is concentrated away from the carrier, making it easier to detect in the presence of phase noise and possible interference from other readers. In the time domain, interpretation of an FM0 symbol depends on a single edge, whereas an MMS symbol provides a number of edges to locate, reducing the likelihood of a bit error [20].

III. Design Methodology

The design methodology of the research is shown in Fig. 9. This section explains the step that carried out toward achieving the objectives of the research. The behavior of the encoder and decoder architecture is realized by derivation of Verilog HDL code according to ISO/IEC 18000-6 protocol. The development of the Verilog HDL has done using Quartus II version 10.0 development tools. In programming derivation stage, design specification such as input and output element is determined.

After programming derivation and development, the Verilog HDL code is compiled and synthesized in order to check any error of the design syntax. In the compilation and synthesized stage, the Verilog HDL code will be converted to the logic gates. If the compilation and synthesis success, the Verilog HDL code then simulated in order to present the waveform that purposely to observe the output waveform designed tally with theoretical expectation. The simulation to observe its functionality was done using ModelSim-Altera version 6.5e software.

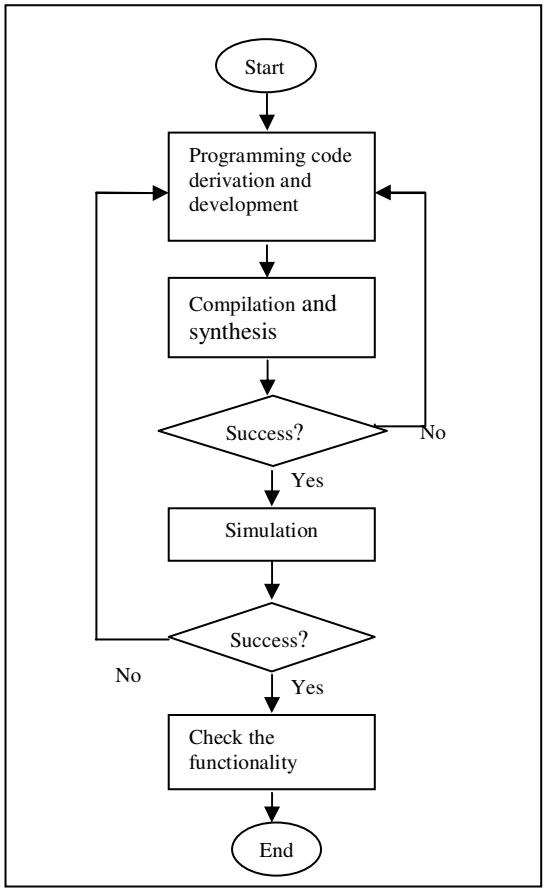

Fig. 9. Design Methodology

Fig. 10 below depicts the block diagram of the PIE encoding. The input elements are in form of time domain and digital logic data either data 0 or data 1. After through the PIE encoder, the output results are encoded data in term of symbol that express logic 1 and logic 0.

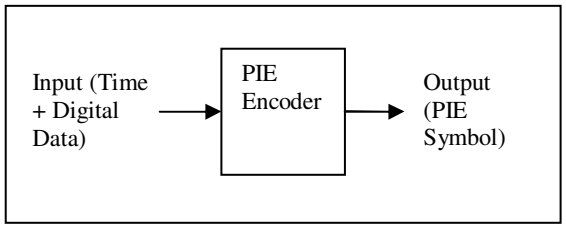

Fig. 10. Block Diagram of PIE Encoder

Fig. 11 below depicts the block diagram of the FM0 and Miller decoder. The input element is in term of FM0 and Miller symbol that express logic 1 and logic 0. The output results are decoded symbol in form of digital logic data either data 0 or data 1.

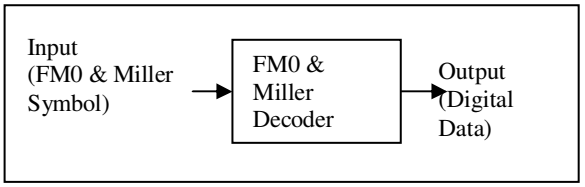

Fig.11. Block Diagram of FM0 and Miller Decoder

Fig. 12 shows the PIE encoding block that consists of an input and output element. There are six elements of input in the PIE encoder block. On the right hand side is output and left hand side is an input. The clk and rst are the basic connection in coding modeling. The RT_start and RT_preamble are input signal of the modeling. The RT_width and RT_data indicate the amount of logic data to be loaded. The only one output element is symbol_out that depend on the input condition inserted.

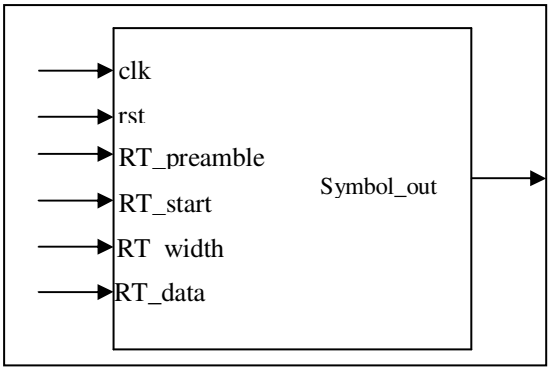

Fig. 12. PIE Encoder Block

Whereas Fig. 13 shows the decoder block that consists of an input and output element. The nine elements on the left hand side of the block is an input of FM0 and Miller decoder. Two elements on the right hand side is an output. The clk and rst are the basic connection in decoding modeling. The input signal demodin is an output symbol from the tag. The Miller and Divide Ratio is a parameter to identify data rate. The TRext is parameter to identify whether reader send a pilot tone or not. The RTcal_value, TRcal_value and Tari_value indicate the timing information. The output Rx_data displays the sequence of digital data depending on input demodin and Rx_width display their length of the data.

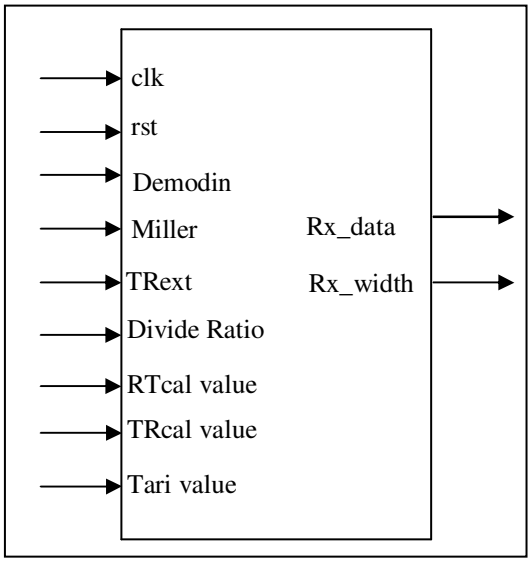

Fig. 13. FM0 and Miller Decoder Block

The Verilog design flow for the PIE encoder modeling is shown in Fig. 14. The modeling begins with the start point and initialization of input and output element including its register and wire needed. Then, the operation will start with reset setting. If the input reset is at logic 1, all the registers in the PIE encoder modeling will be clear. Otherwise, the "RT_state" State Machine will take over.

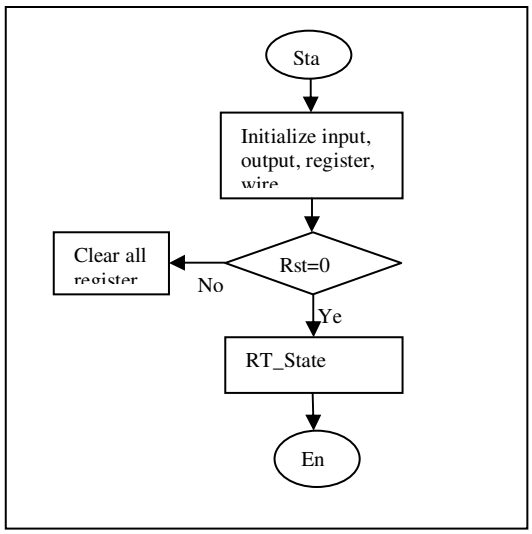

Fig.14. Verilog Modeling

Fig. 15 shows the state diagram for R=>T communications. The "RT_state" start with parameter "state_Idle", if the control signal mentions that the reader ready to start the Query command that denote the inventory round, the state will jump to preamble parameter by sequence start with "state_Delimiter" followed by "state_Data-0", "state_RTcal" and "state_TRcal".

After then inventory round success, the reader will send another query command. The subsequence Query command will use the Frame_synch parameter start with "state_Delimiter" followed by "state_Data-0" and "state_RTcal". There are no "state_TRcal" if the inventory round already success. The operations go to from one state to another state if the inputs declared in respective state achieve the value needed.

The "state_Data" consists of operations to encode the logic 0 or logic 1 into the respective PIE symbol. Either after preamble or frame synch, the state will proceed with the "state_Data" and then after complete decode the digital data; the "state_done" will come over.

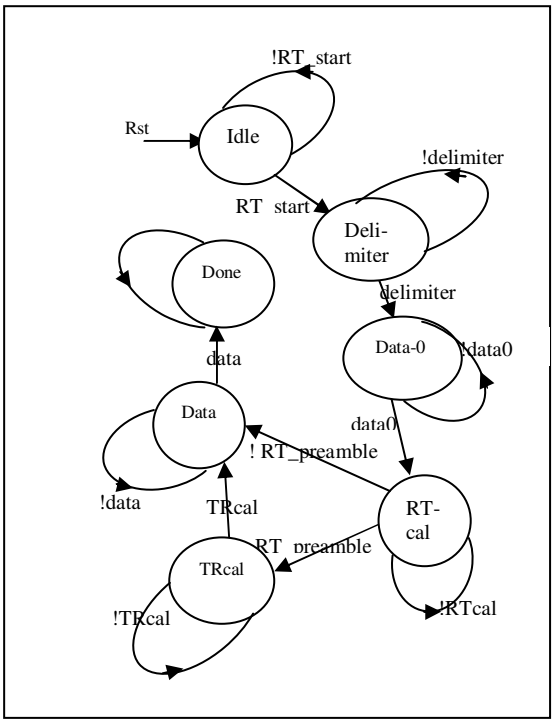

Fig. 15. State Diagram for RT_State

At the "state_data", both condition of data_end and data_tx for data0 or data1 is compare to generate the PIE symbol. The coding flow show in Fig. 16 is the mapping on how the coding made to

encoded the PIE symbol based on current data transmitted at "state_Data". The "data0_end" is assign to be equal to value of counter greater than "Tari_value" and "data0_tx" assign to counter greater than "Tari_value" minus "PW_value". "data1_end" is assign to be equal to value of counter greater than "2*Tari_value" and "data0_tx" assign to counter greater than "2*Tari_value" minus PW_value. If the "current_data" equal to logic 1, "data_tx" and "data_end" equal to "data1_tx" and "data1_end" respectively. Else "data_tx" equal to "data0_tx" and and "data_end" equal to "data0_end".

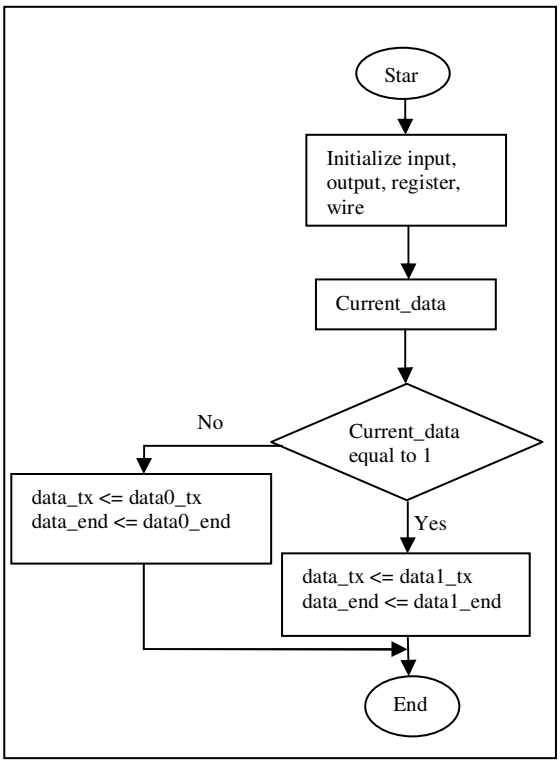

Fig. 16. Verilog Modeling to Identified Data0 or Data1

The Verilog design flow for the FM0 and Miller decoder modeling is shown in Fig. 17. The modeling begins with the start point and initialization of input and output element including its register and wire needed. Then, the operation will start with reset setting. If the input reset is at logic 1, all the registers in the FM0 and Miller decoder modeling will be clear. Otherwise, the "rx_state" case will take over with parameter state_preamble and state_bits.

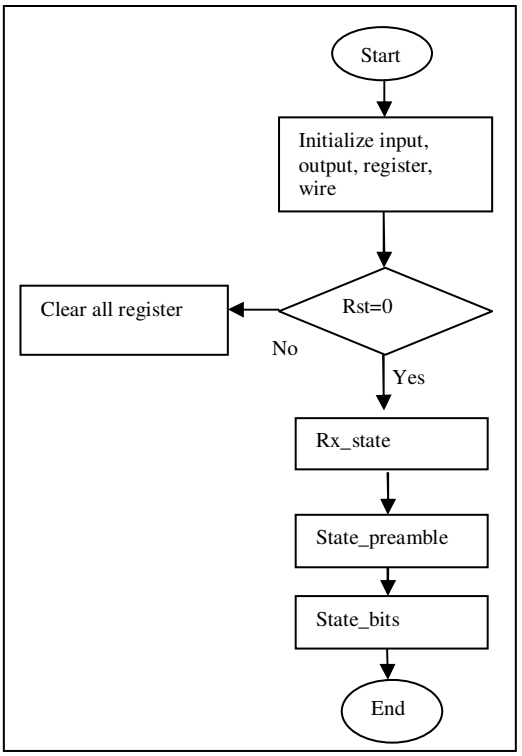

Fig.17. Verilog Modelling

The coding flow in Fig. 18 shows the mapping on how the reader sets the tag to reader frequency. The reader sets the tag to reader frequency by sending the divide ratio value whether equal to logic one or logic zero. The "TRcal3" is a ssign to be equal TRcal_value times three. The value of TRcal_value is set according to the timing information defined in the standard. If the divide ratio is set equal to logic 1, the timing is equal to "t_dr1" whereas timing is equal to "t_dr0".

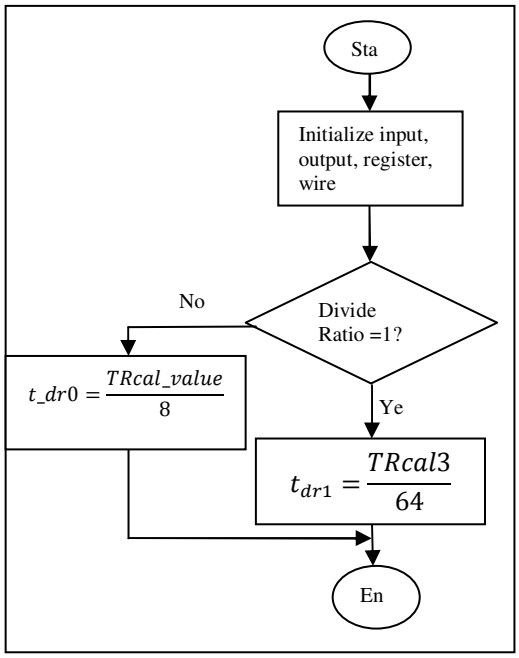

Fig. 18. Reader Sets the Tag to Reader Frequency

IV. Results And Discussions

The architecture is modeled in Verilog HDL. The Verilog HDL design is analyzed and synthesis using Quartus II software. The results from the simulation in ModelSim-Altera were compared and analyzed with theoretical expectation. Final simulation results for encoder and decoder was carried out which was designed based on its characteristic.

A. Simulation Waveform Result for Pie Encoder

A test bench was developed to get the simulation waveform results. The test bench is a virtual environment that used to verify the correctness of the model. At the test bench module, the entire input element was set based on the criteria needed to perform an operation to generate output "symbol_out". The input "RT_start" is the control signal which is indicating the modeling either to start the communication or not. The input "RT_preamble" also modeled as a control signal either to operate the signaling with R=>T Preamble or R=>T Frame-Synch.

All readers need start the R=>T signaling with the R=>T Preamble. Fig. 19 shows the input element such as reset is set to logic 0, "RT_start" is set to logic 1 and "RT_preamble" is set to logic 1. When the "RT_preamble" is set to logic 1, means that the R=>T signaling will start with a preamble and precede a Query command that denotes the start of an inventory round. From waveform, observe that the output "symbol_out" indicate the R=>T Preamble by display the parameter "Delimiter" followed by "Data0", "RTcal", "TRcal", a sequence of "Data" either data0 or data1 and "Done". Refer to "RT_state" register, binary value "001", "010", "011" and "100", refer to parameter "delimiter", "data0", "RTcal" and "TRcal" respectively. Their timing results are equal to 12500ns, 6250ns, 18750ns and 25000ns respectively.

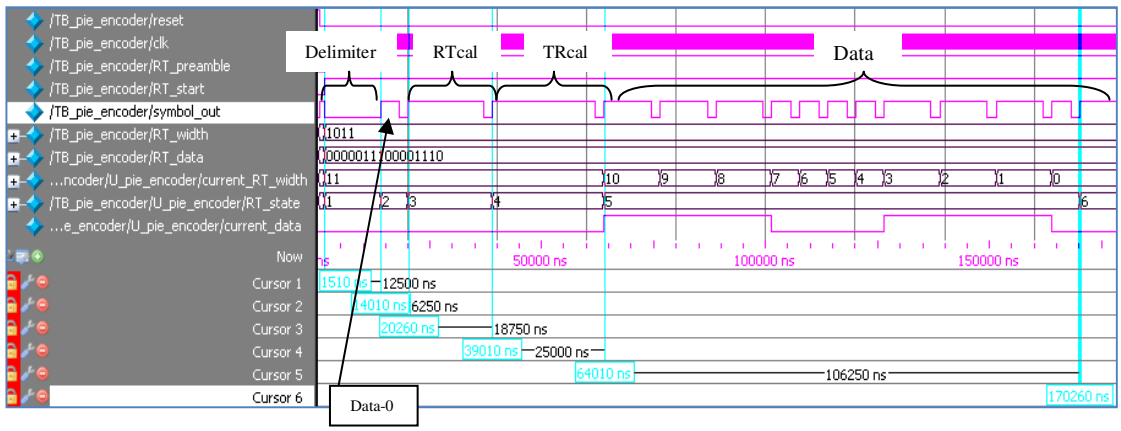

Fig. 19. R=>T Preamble

When the input reset is set to logic 0, "RT_start" is set to logic 1 and "RT_preamble" is set to logic 0, the reader will start the signaling with R=>T frame-sync as shown in Fig. 20. Observe that the

output symbol_out was displaying the parameter "Delimiter" followed by "Data0", "RTcal", a sequence of "Data" and "Done". The timing and signaling for frame-synch has the same value and structure respectively as that of the preamble except for the absence of the "TRcal_state". As discuss before, every signaling from reader to tag must begin with R=>T preamble or R=>T Frame-Sync. The first Query command must start with a preamble which denotes the starting of the inventory round and all other subsequent command starts with a Frame-Sync. At this modeling, by setting "RT_preamble" equal to 0, the first Query Command assume already start and reader send the subsequence query command that will use Frame-sync signaling.

Refer to the Fig. 18 and Fig. 19; the binary value "101" of the "RT_state" is equal to the parameter "state_Data". In that state, the results of "symbol_out" display the sequence of data-0 and data-1 symbol that has been encoded by PIE encoder. The sequences of data-0 and data-1 depend on the input value of "RT_width" and "RT_data". The "current_RT_width" will be minus by one every data bit end. Otherwise, the high signal display at "current_data" respective data-1 and low signal respective data-0.

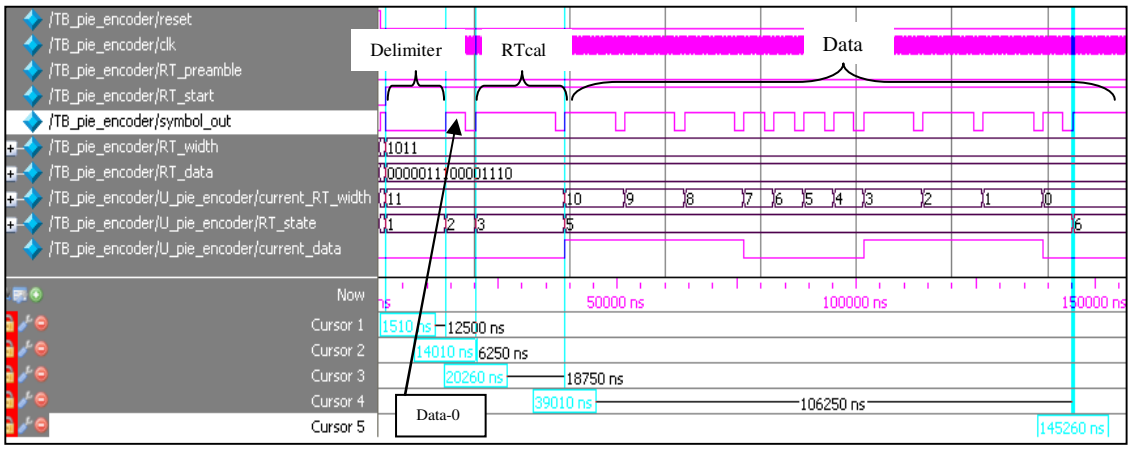

Fig. 20. R=>T Frame_Sync

Fig. 21 show more detail about the timing result of the data-0, data-1 and PW. The value of data-0 equal to tari value which is 6250ns, data-1 is 12500ns and PW is 2000ns. The simulation results show that timing value is equal to the input timing that was inserted.

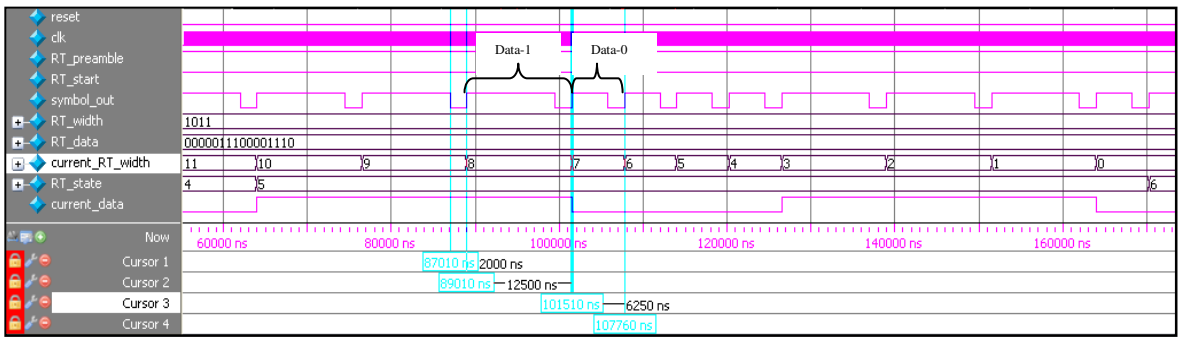

Fig. 21. Timing Result

The final result simulation waveform was observed and it shows that the output data of the PIE encoder tally with the theoretical expectation that was explained in ISO/IEC 18000-6 protocol.

B. Simulation Waveform Result for FM0 and Miller Decoder

The entire input element was set based on the criteria needed to perform an operation to decode the symbol come from tag and display at "rx_data" output. The input "demodin" represent the signal came from the tag and after demodulated by the ASK or PSK modulator at the reader receiver. The input "miller", "Trext" and "divide ratio" are set by the reader to identify the tag to reader data rate. As known, backscatter link frequency or data rate varies depending on the parameter sent by the reader to the tag and on the encoding types sends either FM0 or Miller. In this section show the results and discussion of two decoding types perform by the reader in the receiver part.

Fig. 22, Fig. 23, Fig. 24 and Fig. 25 shows the waveform results for FM0 and Miller decoder. From waveform observe that the output "rx_data" indicate decoded digit from the "demodin" symbol that was inserted as input. The binary value "00" and "01" in the row "rx_state" refer to parameter "state_preamble" and "state_bits" respectively. The timing follows the "t_dr0" or "t_dr1" values depend on the digit of divide ratio. If the divide ratio is set to zero, timing is equal to "t_dr0" whereas timing is equal to "t_dr1". The "rx_count" is a counter for input demodin and will start counting every changing of demodin. If the "rx_count" greater than "timing_div", the "rx_data" that correlate with "rx_width" (rx_data[rx_width]) will give the value of one otherwise "rx_data" equal to zero. The "t_dr0" and "t_dr1" are calculated by using equation (2).

Fig. 22 shows the waveform results of the FM0 decoder when the input "Miller", "TRcal_value", "TRext" and "divide_ratio" are set to zero, 25000ns, one and zero respectively. The value of "rx_width" (0-17) at the preamble state indicate the 18 bits of FM0 preamble when input "TRext" is set to logic one. At the bits state, the "rx_data" start display the sequence value of digit one and zero depending on input "demodin". The timing follows the "t_dr0" because of divide ratio is set to logic 0. The "rx_width" indicated the bit in the packet data. The input "demodin" for bit 1, 3, 4, and 7 is equal to symbol one. Therefore, the output "rx_data" display the value of 2, 10, 26 and 154 that correspond to 2^1, $2^1 + 2^3$, $2^1 + 2^3 + 2^4$ and $2^1 + 2^3 + 2^4 + 2^7$ respectively.

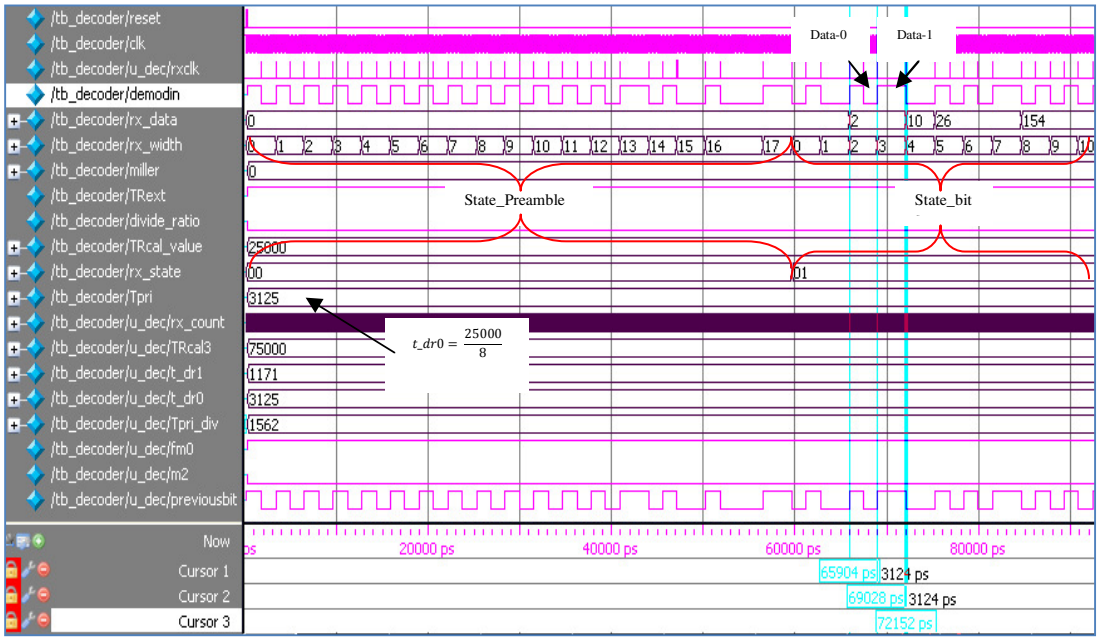

Fig. 22. FM0 Decoder

Fig. 23 shows the waveform results of the Miller decoder for two subcarrier cycle per symbol. The input "Miller" "TRcal_value", "TRext" and "divide_ratio" are set to one, 25000ns, one and one respectively. The value of "rx_width" (0-21) at the preamble state indicate the 22 bits of Miller preamble when input "TRext" is set to logic one. At the bits state, the "rx_data" start display the sequence value of digit one and zero depending on input "demodin". The timing follows the "t_dr1" because of divide ratio is set to logic 1. The input demodin for bit 2, 3 and 6 is equal to symbol one. Therefore, the output "rx_data" display the value of 4, 12 and 76 that correspond to 2^2, $2^2 + 2^3$ and $2^2 + 2^3 + 2^6$ respectively.

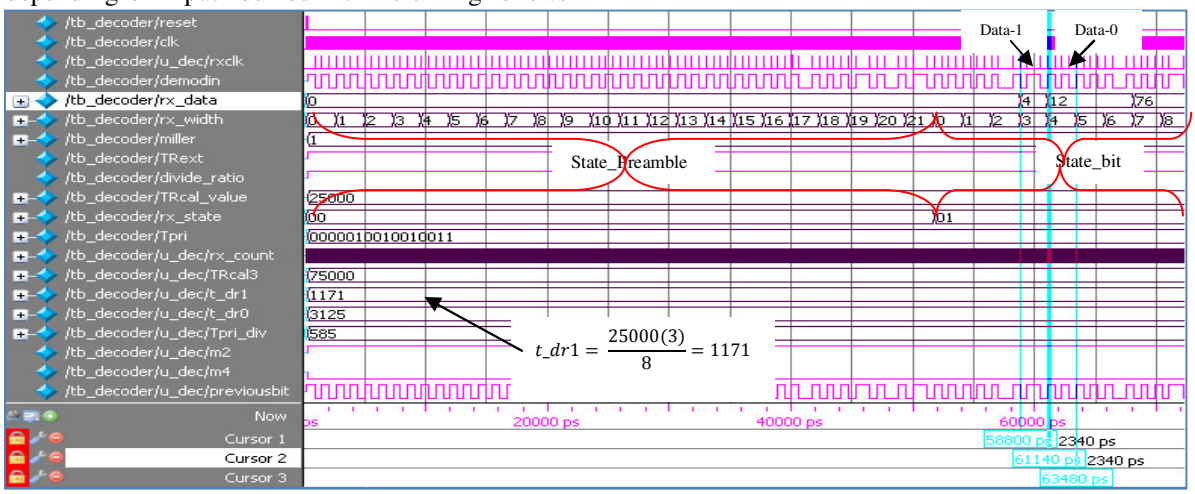

Fig. 23. Miller Decoder (2 subcarrier cycles per symbol)

Fig. 24 shows the waveform results of the Miller decoder for four subcarrier cycle per symbol. Input of "Miller", "TRcal_value", "TRext" and "divide_ratio" are set to two, 25000ns, zero and zero respectively. The values of "rx_width" (0-9) at the preamble state indicate the 10 bits of Miller preamble when input "TRext" is set to logic zero. The input demodin for bit 0 and 2 is equal to symbol one. Therefore, the output "rx_data" display the value of 1 and 5 that correspond to 2^0 and $2^0 + 2^2$ respectively.

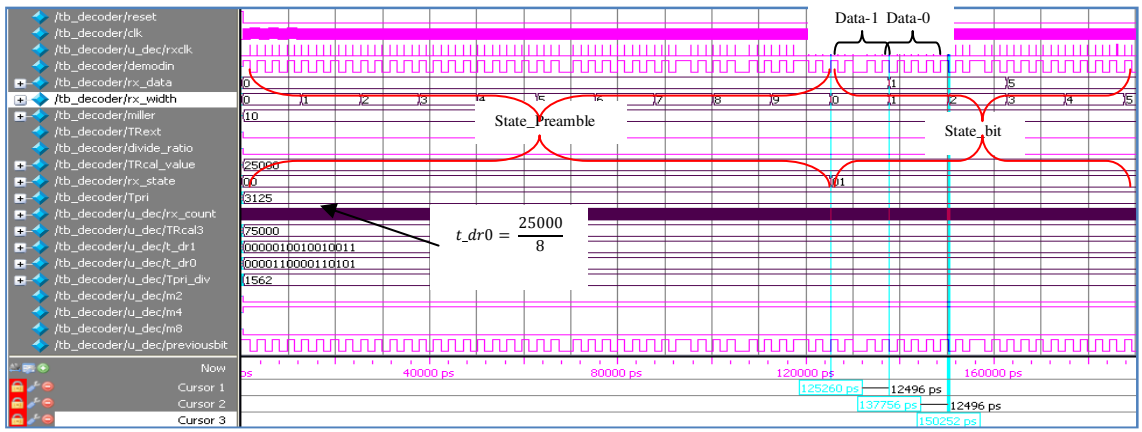

Fig. 24. Miller Decoder (4 subcarrier cycle per symbol)

Fig. 25 shows the waveform results of the Miller decoder for eight subcarrier cycle per symbol. Input of the "Miller", "TRcal_value", "TRext" and "divide_ratio" are set to three, 25000ns, zero and one respectively. The values of "rx_width" (0-9) at the preamble state indicate the bits of Miller preamble when input "TRext" is set to logic zero. At the bits state, the "rx_data" start display the sequence value of digit one and zero. The input demodin for bit 0 and 3 is equal to symbol one. Therefore, the output "rx_data" display the value of 1 and 9 that correspond to 2^0 and $2^0 + 2^3$ respectively.

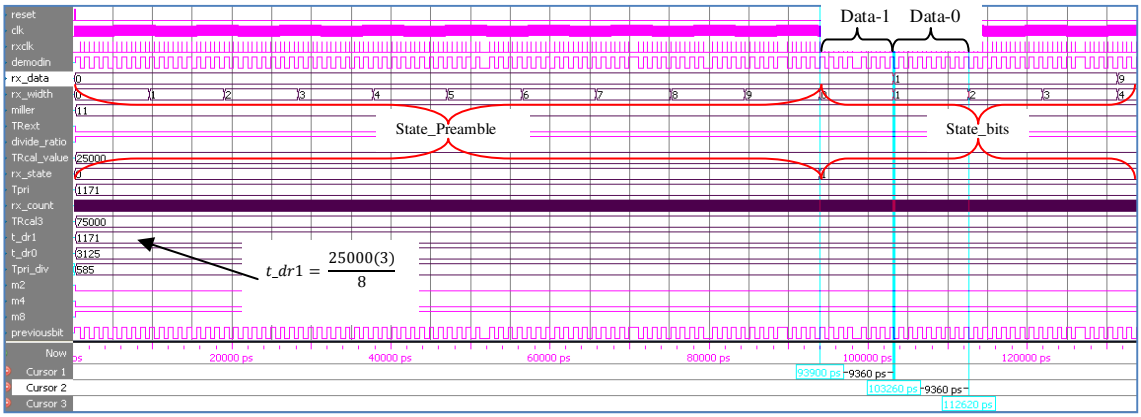

Fig. 25. Miller Decoder (8 subcarrier cycles per symbol)

The final result simulation waveform was observed and it shows that the output symbol of the PIE encoder, FM0 decoder and Miller decoder tally with the theoretical expectation that was explain in ISO/IEC 18000-6 protocol.

V. Conclusion

The result and discussion discussed in the previous section gives a clear perspective that the objectives have been successfully achieved. The entire physical layer design is develop using Verilog HDL and simulate using ModelSim-Altera. The simulation results show that the PIE encoder model follow the ISO/IEC 18000-6 protocol. The model of PIE encoder intended for UHF RFID Reader and will be integrated as a core architecture in the UHF RFID Reader design as a future work.

Acknowledgment

The work was completed with a grant from the Dana Kecemerlangan of UiTM. Grant No: 600-RMI/DANA 5/3/RIF (37/2012).

References

[1] J. Landt, "The history of RFID," *Potentials, IEEE,* vol. 24, pp. 8-11, 2005.

[2] B. Nath, *et al.*, "RFID Technology and Applications," *Pervasive Computing, IEEE,* vol. 5, pp. 22-24, 2006.

[3] R. Weinstein, "RFID: a technical overview and its application to the enterprise," *IT Professional,* vol. 7, pp. 27-33, 2005.

[4] K. Finkenzeller, "RFID Handbook," *Fundamentals and Applications in contactless smart cards, Radio Frequency Identifications and Near-Field Communication, A John Wiley and Sons, Ltd. Publication,* vol. Third Edition, 2010.

[5] "EPCglobal IncTM, EPCTM Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz -960MHz Version 1.2.0," 2008.

[6] "ISO/IEC 18000-6: Information Technology-Radio Frequency Identification for item management- Part 6: Parameter for air interface communication at 860 MHz to 960 MHz," 2006.

[7] P. V. Nikitin and K. V. S. Rao, "Performance limitations of passive UHF RFID systems," in *Antennas and Propagation Society International Symposium 2006, IEEE*, 2006, pp. 1011-1014.

[8] H. Li, *et al.*, "A new implementation of UHF RFID reader," in *TENCON 2009 - 2009 IEEE Region 10 Conference*, 2009, pp. 1-4.

[9] P. Xingdong, *et al.*, "Design and realization of a highly integrated UHF RFID reader module," in *Microwave and Millimeter Wave Technology, 2008. ICMMT 2008. International Conference on*, 2008, pp. 1503-1505.

[10] Y. Chen and F.-h. Zhang, "Design on UHF RFID reader software," in *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on*, 2009, pp. 575-578.

[11] L. Joon Goo, *et al.*, "Software architecture for a multi-protocol RFID reader on mobile devices," in *Embedded Software and Systems, 2005. Second International Conference on*, 2005, p. 8 pp.

[12] S. Ki-Hwan, *et al.*, "Implementation of the ISO/IEC 18000-6 Type C Prototype Reader," in *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, 2008, pp. 1-2.

[13] L. Liu and Q. Zhang, "Design and implementation of a long-range RFID reader," in *Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on*, 2011, pp. 297-300.

[14] Y. Ching-Chien, *et al.*, "The design of encoding architecture for UHF RFID applications," in *Microwave Conference, 2008. APMC 2008. Asia-Pacific*, 2008, pp. 1-4.

[15] S. Tung and A. K. Jones, "Physical layer design automation for RFID systems," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, 2008, pp. 1-8.

[16] M. A. Khan, *et al.*, "FSM based FM0 and Miller encoder for UHF RFID Tag Emulator," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*, 2009, pp. 1317-1322.

[17] L. Jin and T. Cheng, "Analysis and Simulation of UHF RFID System," in *Signal Processing, 2006 8th International Conference on*, 2006.

[18] P. B. Khannur, *et al.*, "A Universal UHF RFID Reader IC in 0.18-µm CMOS Technology," *Solid-State Circuits, IEEE Journal of,* vol. 43, pp. 1146-1155, 2008.

[19] L. Jing, *et al.*, "ASIC design of UHF RFID reader digital baseband," in *Microelectronics and Electronics (PrimeAsia), 2010 Asia Pacific Conference on Postgraduate Research in*, 2010, pp. 263-266.

[20] L. T. Y. Yan Zhang, Jiming chen, "RFID and Sensor Networks- Architectures, Protocols, Security, and Integrations," *Wireless Networks and Mobile Communications Series,* 2010.

Ismarani Ismail is a Senior lecturer at the Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM). She holds an Electronics Manufacturing Engineering from University of Salford, United Kingdom and Bachelor of Engineering
degrees in Electrical and Electronics Engineering from John Moores University, Liverpool, UK. Her main research interest is in Radio Frequency Identification system, electronics product development, system development and electronic and electrical manufacturing processes. She has published in a number of national and international journals. In addition to teaching, she supervises and advises Masters and Doctoral students.

Azlina Ibrahim obtained her degree of B.Eng. (Hons) Electrical Engineering from Universiti Teknologi Mara (UiTM) Shah Alam. Currently she is pursuing her PhD in the area of RFiD Technology. Her PhD work is focusing in developing UHF RFiD reader.