Hardware Implementation of Fast Detection Anti-collision Algorithm For RFID System

Jahariah Sampe[1,3], Masuri Othman[2,3] , Iskandar Baharin[1]

[1] *Universiti Kuala Lumpur (UniKL), 1016 Jalan Sultan Ismail, 50250 Kuala Lumpur, Malaysia*
[2] *Institute of Microengineering and Nanoelectronic (IMEN), Universiti Kebangsaan Malaysia, 43600 Selangor, Malaysia*
[3]*Department of Electrical, electronics and System, Faculty of Engineering, Universiti Kebangsaan Malaysia, 43600 Malaysia*

This paper presents a proposed hardware implementation of Fast Detection Anti-collision Algorithm ((FDACA) for Radio Frequency Identification (RFID) system. Our proposed FDACA is base on the deterministic anti-collision technique. Novelty of the proposed FDACA is fast identification by reducing the number of iterations during the identification process. In this technique, powered tags were divided into group of four for every Read cycle. Meanwhile, the proposed FDACA also reads the identification (ID) bits at once regardless of its length. Furthermore the proposed FDACA does not require the tags to remember the instructions from the reader during the communication process in which the tags were treated as *address carrying devices* only. As a result simple, small, low cost and memoryless tags can be produced. The proposed FDACA system was designed using Verilog HDL. The system was simulated using Modelsim XE II and synthesized using Xilinx synthesis technology (XST). The system has been successfully implemented in hardware using Field Programmable Grid Array (FPGA) board model Virtex II Xc2v250. Finally the output waveforms from the FPGA have been displayed on the Tektronix Logic Analyzer model TLA 5201 for real time verification. The results show that the proposed FDACA can identify tags without error until 100MHz and more.

1. Introduction

The RFID system is consists of three main components as shown in Figure 1; reader or interrogator, tag or transponder and software or data management module. The reader is to read data from and write data to tags by broadcasting the RF signals. The tags are to store data or unique Identification (ID) numbers and basically attached to the objects to be identified.

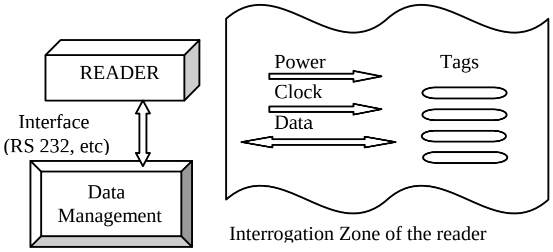

Figure 1: RFID System

A collision occurs when multiple tags in the reading zone (magnetic/electromagnetic field) of the reader simultaneously respond to the reader's commands. The reader will receive mixture of signals and this would lead to wrongly identify the tags. In order for reader to communicate with these multiple tags, anti-collision technique is required to coordinate the communication between reader and tag. Anti-collision algorithm circuit employs one of the main parts of the control unit (ASIC module) of the reader(1). Depending on the algorithm, the anti-collision algorithm circuit may consist of a random number generator, counters, flip-flops, gates and etc.

Anti collision techniques can be classified into two; deterministic or reader-driven techniques and stochastic/probability or transponder-driven techniques. In the deterministic techniques, the reader (Master) controls the communication between tags (Slaves) and in the stochastic techniques, tags (Masters) control the communication (1). The common stochastic technique is based on aloha algorithm such as Aloha, slotted Aloha and frame slotted aloha. In this technique, tags avoid collision by responding to reader command at random intervals. If collision occurs, the affected tags will wait for another random interval before responding again (8) (11).

The common deterministic technique is based on Tree Algorithm such as Binary Tree and Query Tree Algorithms(1)-(5). In this technique, tags avoid collision by employing unique ID numbers and reader sends command to multiple tags for the next bit of their ID. If the reader detects a collision, it will send response bit indicating which tags should continue with the protocol and which should quit. Each choice of bit represents choosing a branch in a tree and the leaves correspond to tag's ID. Advantages of tree-based technique over aloha-based technique are tree-based does not cause the tag starvation and has efficient broadcasting because only bits of an ID are used to identify the tags(1). However, the identification time of Binary Tree algorithm is depends on two parameters, the number of tags simultaneously exists in the interrogation zone and the length of tag's ID. If either one of these parameters is increased the identification time will increase. In this technique also the tags are required to remember the previous instructions from the reader during the communication process.

2. FDACA Architecture

Based on Binary Tree Anti-collision Algorithm, the Fast Detection Anti-collision Algorithm (FDACA) is proposed. Novelty of the proposed FDACA is to reduce the identification time by reducing the number of iterations

ISBN 978-979-16338-0-2

needed to identify one tag. The powered tags are divided into group of four for every read cycle in order to reduce the number of iterations during the identification process. In addition, the identification time of the proposed FDACA is not depends on the length of the tag's ID. Instead of sending and receiving ID bit by bit, the FDACA will read all the ID bits at once regardless of its length. Meanwhile, this algorithm also does not require the tags to remember the previous instructions from the reader during the identification process. The reader transmits the read command to the tags and the tags will simultaneously backscatter its' ID bits. As a result the tag is treated as a *address carrying device* i.e. the tag only carries its identification bits. Therefore, memory-less tag which exhibits very low power consumption can be produced[(4)].

During the identification process, the FDACA will identify four tag's IDs simultaneously in one read cycle. The FDACA will firstly identify the smallest ID bits and finally the largest one, follows the two levels of Binary Tree with maximum four leaves. The hardware implemented of FDACA system is designed using top-down approach as shown in Figure 2.

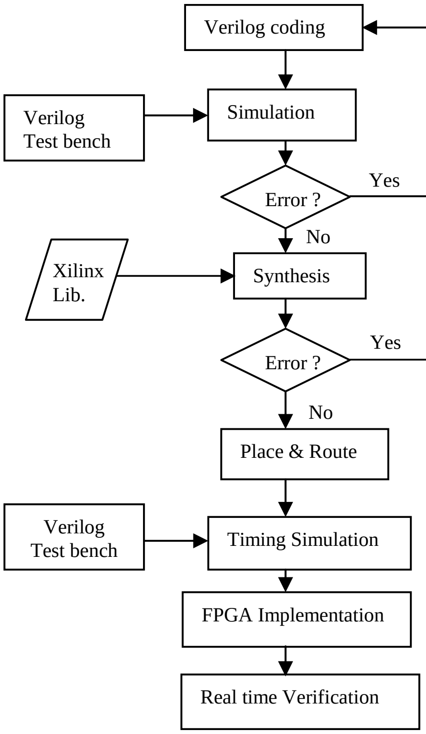

Figure 2: Design Flow Chart of FDACA

2.1 FDACA Modules

The proposed Fast Detection Anti-collision Algorithm (FDACA) is consists of one Top-module and five sub-modules so called FDACA system. The FDACA system was designed using Verilog HDL. The system also was simulated using Modelsim XE II and synthesized using Xilinx synthesis technology (XST). The FDACA's Top module consists of Data-generator module, Clock-divider module, Fast-search module, Select-generator module and Read-tag module as shown in Figure 3.

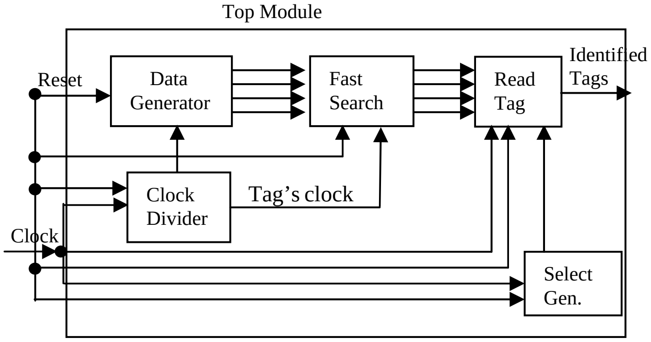

Figure 3: Modules of FDACA system

Data-generator module is to generate random tag's ID bits with any lengths for examples eight bits ID and sixteen bits ID. For every cycle of tag clock, four tag's IDs will be simultaneously generated for identification.

Clock-divider Module is to provide clock for Data-generator and Fast-search Modules and called as Tag clock. Function of this module is to generate clock with period (T_{clk_tg}) is equal to four times period of system clock (T_{clk}) as represented by Eq. (1). One cycle (a period) of tag clock is used to identify four tags simultaneously and also called one read cycle.

$$T_{clk\_tg} = 4 \text{ x } T_{clk} \qquad (1)$$

Fast-search module is the heart of FDACA system, in this module the tags' ID will be manipulated in such a way there is no collision between them. For every read cycle, the search process will start by loading the four tag's IDs into four arrays of register. Then the comparators will compare these tag's IDs. The search process is performed by two ICs which are used conditional statements (*if-else*) for the left and the right branches of the tree respectively. The four identified tag's IDs are loaded simultaneously into the output registers at the negative edge of tag clock.

Read-tag Module will display the four identified tags from the Fast-search module serially, one tag for every cycle of system clock start from the smallest tag's ID until the largest one. So for every cycle of tag clock, four tag's IDs will be displayed. Select generator module is to generate select signals for Read-tag module. Four states will be generated in every four cycles of system clock which are equal to one cycle of tag clock.

Top module is used to implement the FDACA system in hardware using FPGA. Function of this module is very similar to Read-tag Module but modified to suit with the hardware requirements. Logic diagram of the synthesized Top module is shown in Figure 4.

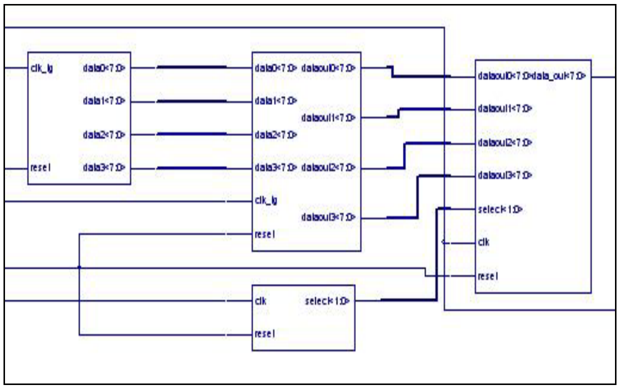

Figure 4: Logic Diagram of Synthesized Top Module

3. Simulation Results

Verilog HDL codes of FDACA system have been successfully simulated and verified using ModelSim XE II/Starter 5.7g tool. Every FDACA module has been tested on their individual test bench. The FDACA modules also have been simulated and verified at each level of simulation categories as listed at the following.

i. Behavioral Model Simulation
ii. Post-Translate Verilog Model Simulation
iii. Post-map Verilog Model Simulation
iv. Post Place and Route Verilog Model Simulation

For Top module simulation, the first three categories have given the best simulation results without error at the minimum period of 4ns or maximum frequency of 250 MHz. The fourth simulation category has given the best simulation result without glitches/jitters at the minimum period of 8ns or maximum frequency of 125 MHz. The following will discuss the output waveforms from the Behavioral simulation for the selected modules of the FDACA system.

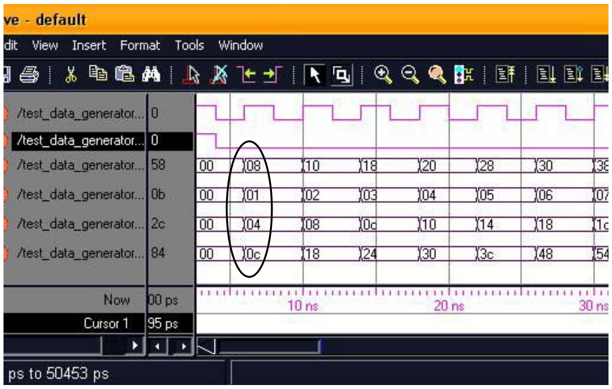
ve - default
dit View Insert Format Tools Window
/test_data_generator... 0
/test_data_generator... 0
/test_data_generator... 58
/test_data_generator... 0b
/test_data_generator... 2c
/test_data_generator... 84
00 08 10 18 20 28 30 38
00 01 02 03 04 05 06 07
00 04 08 0c 10 14 18 1c
00 0c 18 24 30 3c 48 54
Now 00 ps
Cursor 1 95 ps
10 ns
20 ns
30 ns
ps to 50453 ps

Figure 5: Output waveform from Data-generator module

In Data-generator module, four tag's IDs will be simultaneously generated at every positive edge of tag clock. For example from Figure 5, tag's IDs 08_{16}, 01_{16}, 04_{16} and $0C_{16}$ are simultaneously generated for identification. These tag's IDs will be identified in Fast-search Module.

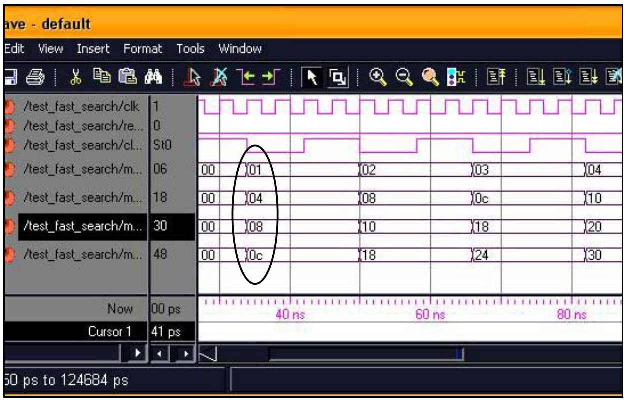
ave - default
Edit View Insert Format Tools Window
/test_fast_search/clk 1
/test_fast_search/re... 0
/test_fast_search/cl... St0
/test_fast_search/m... 06
/test_fast_search/m... 18
/test_fast_search/m... 30
/test_fast_search/m... 48
00 01 02 03 04
00 04 08 0c 10
00 08 10 18 20
00 0c 18 24 30
Now 00 ps
Cursor 1 41 ps
40 ns
60 ns
80 ns
50 ps to 124684 ps

Figure 6: Output waveform from Fast-search module

Fast-search Module will search the tag's IDs from the smallest value to the largest one. At every negative edge of tag clock, four tag's IDs will be identified and stored into four registers. For example tag's IDs 08_{16}, 01_{16}, 04_{16} and $0C_{16}$ as marked by a circle at Figure 5 are identified as 01_{16}, 04_{16}, 08_{16}, and $0C_{16}$ accordingly as shown in Figure 6 (marked by circle).

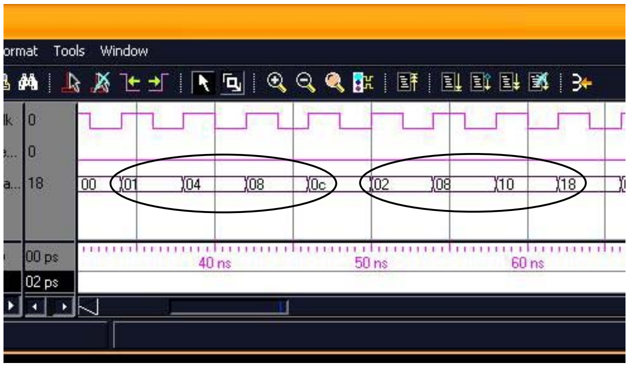

Figure 7: Output waveform from Top module

The output of Top Module as shown in Figure 7 which will display the tag's IDs serially. For example during the first cycle of tag's clock, tags with IDs 01_{16}, 04_{16}, 08_{16}, and $0C_{16}$ will be displayed serially as marked by the first circle. Then for the next cycle of tag's clock, tags with ID 02_{16}, 08_{16}, 10_{16}, and 18_{16} will be displayed serially as marked by the second circle.

4. Design Implementation and Verification

The FDACA system has been implemented in hardware using FPGA model Virtex II Xc2v250 with maximum frequency of 180MHz. The output waveforms from the FPGA have been displayed using Tektronix Logic Analyzer model TLA 5201 for real time verification as shown in Figure 8. The results show that the FDACA system still can identify tags without errors up to 110MHz and more.

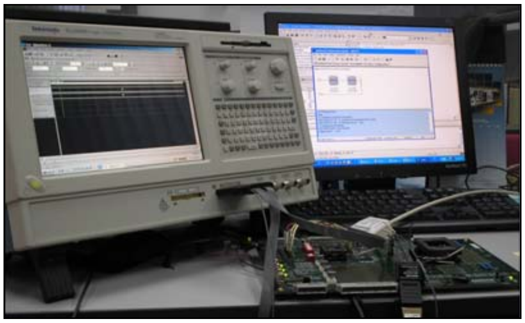

Figure 8: FPGA implementation and verification platform

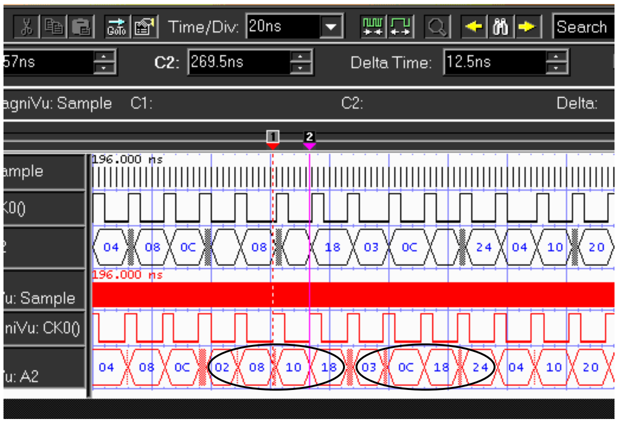

Figure 9: 80 MHz output waveform

Figure 9 shows the tag's IDs have been identified for two clock cycles of tag clock as marked by two circles. For each cycle of tag clock the tag has been identified from the smallest ID to the largest ID value. Examples for the first cycle of tag clock the identified tags are 02_{16}, 08_{16}, 10_{16} and 18_{16} accordingly as marked by the first circle. The second cycle of tag clock the identified tags are 03_{16}, $0C_{16}$, 18_{16} and 24_{16} as marked by the second circle.

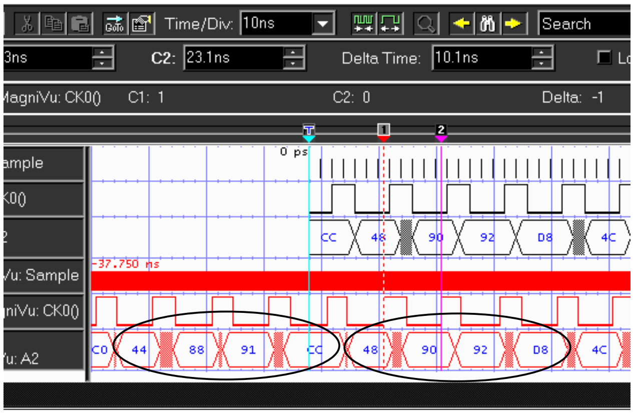

Figure 10: 100 MHz waveform

Figure 10 shows at 100 MHz, the FDACA system still can identify the tag's IDs correctly for every cycle of tag clock which is equal to four cycles of system clock. Examples for the first cycle of tag clock the identified tag's IDs are 44_{16}, 88_{16}, 91_{16} and CC_{16} accordingly. The second cycle of tag clock, the identified tags are 48_{16}, 90_{16}, 92_{16} and $D8_{16}$. Even though there are glitches/jitters on the output, the system still can identify the tag's IDs without error since the identification process occurs for one period of system clock. Meanwhile the glitches only occur at very short time not even half period of system clock.

5. Conclusion

The proposed Fast Detection Anti-collision Algorithm (FDACA) is presented to minimize the identification time for the deterministic anti-collision technique. Also the proposed FDACA architecture enables the production of tag with small, simple, low cost and low power features. With the help of four levels of simulation the proposed FDACA system has been successfully implemented in hardware using FPGA model Virtex II Xc2v250. Real time verification of hardware implemented FDACA system verified the system can identify the tags without error until 100MHz and more.

References

(1) K. Finkenzeller, and R. Waddington: *RFID Handbook: Fundamental and Applications in Contactless Smart Cards and Identification*. John Wiley & Sons, 2nd edition (2003).

(2) F. Zhou, D. Jing, C. Huang and H. Min: *Optimizing the Power Consumption of Passive Electronic Tags for Anti-collision Schemes*, In Proceedings of the 5th ASICON, Beijing, China, pp.1213-1217(2003)

(3) F. Zhou, D. Jing, C. Huang and H. Min: *Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems*, In Proc. ISLPED'04, Newport Beach, California, USA, pp. 357-362 (2004).

(4) C. Law, K. Lee and K. Y. Siu: *Efficient Memoryless Protocol for Tag Identification*, In Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachusetts, pp.75-84 (2000)

(5) MIT Auto-IDCenter. *Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag*. http://auto-id.mit.edu, (2003)

(6) J. Myung and W. Lee: *An adaptive memoryless tag anti-collision protocol for RFID networks*, IEEE INFOCOM'05, Poster Session(2005)

(7) J. Myung and W. Lee, *Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision*. IEEE COMMUNICATON LETTERS, Vol. 10, No. 3, pp.144-146 (2006)

(8) J. Zhai and G. Wang: *An anti-collision algorithm using two-functioned estimation for RFID tags*, in Proc. ICCSA'05, LNCS 3483, pp.702-711(2005)

(9) *EPCTM radio-frequency identification protocol class-1 generation-2 UHF RFID protocol for communication at 860MHz-960MHz Version 1.0.8*, EPCglobal, (2004)

(10) S. Sarma, D. Brock and D. Engels: *Radio Frequency Identification and the Electronic Product Code*, IEEE Micro, vol. 21 No. 6, pp. 50-54 (2001)

(11) S. Singh: *Low Cost Object Identification in RFID via Dynamic Markov Chain & Two Time Scale SPSA*, In proc. Of the International Symposium on Applications and the Internet Workshops (2006)

(12) Victor P. Nelson, H. Troy Nagle, Bill D. Carroll and J. David Irwin: *Digital Logic Circuit Analysis & Design*, Prentice Hall, Inc. (1995)

(13) Michael D. Ciletti: *Advanced Digital Design with the VERILOG HDL*, Pearson Education, Inc. (2003)