American J. of Engineering and Applied Sciences 1 (2): 136-140, 2008
ISSN 1941-7020

Hardware Implementation of Higher Throughput Anti-Collision Algorithm for Radio Frequency Identification System

Jahariah Sampe and Masuri Othman
Department of Electrical, Electronics and System, Faculty of Engineering,
National University of Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: This paper presents a proposed hardware implementation of Higher Throughput Anti-Collision Algorithm (HTACA) for Radio Frequency Identification (RFID) system. Our proposed HTACA combines a pipeline and a deterministic anti-collision technique in order to enhance its performances. These downlink and uplink operations are concurrently executed during the identification process. The downlink operation is purposely pre-selected a group of tags base on its Object Class. Therefore a packet format of the tag consists of an object class bits (OC) and an Identification bits (ID). Meanwhile, the uplink operation is performed a Fast Detection anti-collision technique. This technique is novel in terms of faster identification time by reducing the number of iterations during the identification process. It also reads the ID at once regardless of its length. The proposed system is designed using Verilog HDL. The system is 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. The output waveforms from the FPGA have been tested on the Tektronix Logic Analyzer model TLA 5201 for real time verification. From the result, it shows that the proposed HTACA system enables to identify the tags without error at the maximum operating frequency of 180 MHz. As a result, the maximum throughput of this hardware implemented system is 180 Megatags sec^{-1}.

Key words: Pipeline, deterministic, downlink and uplink, object class, real time verification

INTRODUCTION

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

There are two types of communication involve in the RFID system. First, the Reader transmits data to the tags (downlink) and the tags simultaneously receive these transmitted data stream. This communication process is called broadcasting. The second type of communication is the tags transmit data simultaneously to the reader (Uplink) and the reader will receive these data streams by using multi-access procedure. This procedure is called anti-collision techniques. In the RFID systems, the Time Division Multiplexing (TDM) procedures are the largest group of anti-collision procedures and a binary tree is one of the examples[1-4]. In the binary tree algorithm, the identification process will first search the smallest tag's ID until the largest one follows the Binary Tree sequence. Since this algorithm is a deterministic anti-collision technique, the reader will control the communication between the Tags. Therefore enable production of tag with simple, small, low cost and low power features. Meanwhile from the work has been done using Matlab simulation, it shows that there are two limitations of the Binary Tree algorithm. Its identification time is dependent on two parameters; the number of tags simultaneously exists in the interrogation zone and the length of tag's

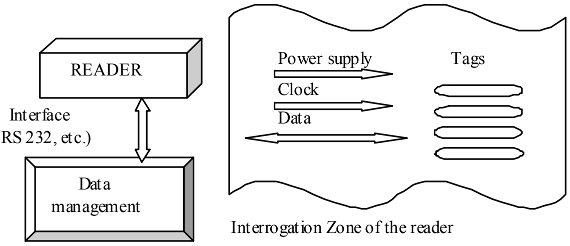


Fig. 1: RFID system

Corresponding Author: Jahariah Sampe, Department of Electrical, Electronic and System, Faculty of Engineering, National University of Malaysia, 43600 Bangi, Selangor, Malaysia

ID. If either one of these parameters is increased the identification time will increase. This algorithm also requires the tags to remember the previous instructions from the reader during the communication process.

Pipeline technique is used to enhance the synchronous system performance by increasing the system throughput. In this technique, the system is partitioned into a linear array of stages and each is executed concurrently. In addition, the technique does not affect the latency of the system, but it increases the system throughput. Hence, the maximum throughput occurs when the data are produced and consumed at the same clock cycle[9].

MATERIALS AND METHODS

The proposed HTACA system is divided into two subsystems; downlink and uplink. Both of these subsystems are executed concurrently using the pipeline technique in order to increase the system throughput. In addition, instruction pipelines are also applied to every module in the subsystems. The maximum throughput of the HTACA system is achieved when the throughput is equal to the inverse of the identification time of the tag.

The first step in the tag identification process is the reader need to preselect a group of tags from many existing tags in the interrogation zone of the reader. In the proposed HTACA, the downlink operation involves preselect a group of tags base on its Object Class. The tags with the same Object class will be grouped in the same register. Therefore, the packet format of the tag consists of the Object Class bits (OC) and the Identification bits (ID). The example of these packets are shown in Table 1 which consists of 3 bit OC and their correspond 8 bit ID.

The uplink operation performs a Fast Detection Anti-Collision Algorithm (FDACA) which bases on a Binary Tree anti-collision algorithm. The proposed primary FDACA is novel in terms of faster identification time by reducing the number of iterations needed to identify one tag. The powered tags are divided into a 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 primary FDACA does not depend on the length of the tag's ID. Instead of sending and receiving the ID bit by bit, the FDACA will read all the ID bits at once regardless of its length. This is performed by using the word-by-word multiplexing or byte interleaving. Meanwhile, this algorithm also does not require the tags to remember the previous instructions from the reader during the identification process. In order to increase the throughput of the primary FDACA system, the number of input/output lines should be increased. But if the number of input/output lines increases, the maximum data rates will decrease and vice versa. Therefore, to increase the number of lines per Read cycle without decreasing the maximum data rate, two primary FDACA modules are multiplexed to form 8 input/output lines. Hence, during the identification process, the FDACA system will identify eight tag's IDs simultaneously in one Read cycle. The FDACA will first identify the smallest ID bits and finally the largest one follows the Binary Tree sequences.

HTACA modules: The HTACA system consists of two main modules; the Downlink and Uplink. Each of these main modules consists of sub-modules. All of these modules are synchronized to a system clock. The HTACA system has been designed using Verilog HDL. The system has also been synthesized using Xilinx Synthesis Technology (XST).

The Downlink module consists of four sub modules, Packet-generator, two Clock-divider, Detect-OC-ID and Registers modules are shown in Fig. 2. The Packet-generator module will randomly generate 255 different tags as shown in Table 1. Then the 255 different tags are fed into the Detect-OC-ID module, eight tags simultaneously at every reselect clock cycle. The Detect OC-ID module will detect the OC and the ID of these tags. Then the respective IDs of these tags are put into the specific registers according to its OC. Therefore, the tags' ID with the same OC value will be grouped together in the same register. Finally, the outputs of this subsystem are fed into the Uplink subsystem. The output consist only the tags ID and eight IDs are fed simultaneously at every Preselect clock cycle. In this subsystem, all operations are synchronized by the Preselect clock. The Preselect clock is equal to eight times of the System clock cycle or equal to one Read cycle.

Table 1: Example of packet format for the tags

Object Class bits	Identification bits (Hex)
0	A0 until BF
1	EO until FF
2	60 until 7F
3	00 until 1F
4	40 until 5F
5	C0 until DF
6	20 until 3F
7	80 until 9F

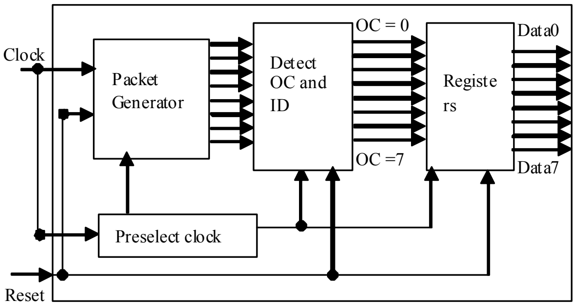


Fig. 2: Modules of downlink system

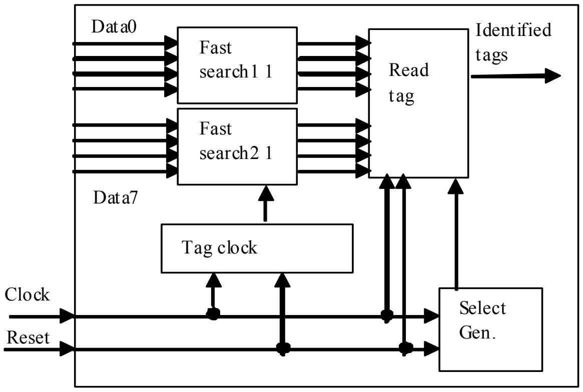


Fig. 3: Modules of uplink system

The Uplink modules are explained in detail as in[10]. This module consists of five submodules, two Fast-search, Read-tag, Clock-divider and Select-generator modules are shown in Fig. 3. The Fast-search module is a heart of the Uplink subsystem. It is based on the TDM operation. In this module the tags' ID will be manipulated in such a way that there is no collision between them. For every cycle of the tag clock, 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. This search process is performed by two ICs which use the 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 the Tag clock. All these loading processes are through the high speed multiplexing line. The operations are also synchronized by the Tag clock which is equal to four times of the system clock. Two Fast-search modules are multiplexed to form 8 input/output lines in order to increase the Uplink throughput. The Read-tag Module will display the eight 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. As a result, eight tag's IDs will be displayed in one Read cycle.

RESULTS AND DISCUSSION

Simulation output: Verilog HDL codes for the HTACA modules have been successfully simulated and verified using ModelSim XE II/Starter 5.7g tool. Every HTACA module has been tested on its individual testbench. These modules have also been simulated and verified at each level of simulation categories. The following will discuss the output waveforms from the Behavioral simulation for the selected modules of the HTACA system.

Figure 4 shows the generated tags inside the Packet-generator module. These tags are loaded into the Detect-OC-ID module, eight tags for every Preselect clock cycle. For example at the first Preselect clock cycle, the tags 276_{16}, $5D2_{16}$, $45F_{16}$, $0B2_{16}$, 631_{16}, $1F1_{16}$, $79F_{16}$ and 318_{16} are loaded into the Detect-OC-ID module.

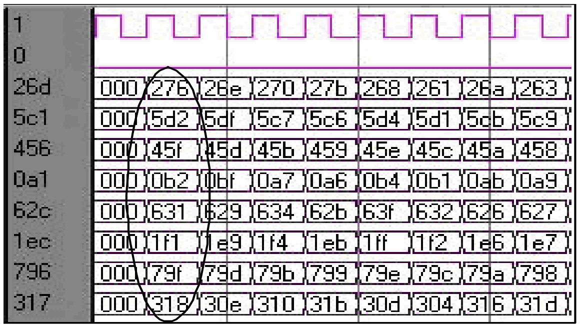

1
0
26d 000 276 26e 270 27b 268 261 26a 263
5c1 000 5d2 5df 5c7 5c6 5d4 5d1 5cb 5c9
456 000 45f 45d 45b 459 45e 45c 45a 458
0a1 000 0b2 0bf 0a7 0a6 0b4 0b1 0ab 0a9
62c 000 631 629 634 62b 63f 632 626 627
1ec 000 1f1 1e9 1f4 1eb 1ff 1f2 1e6 1e7
796 000 79f 79d 79b 799 79e 79c 79a 798
317 000 318 30e 310 31b 30d 304 316 31d

Fig. 4: Outputs of the packets generator

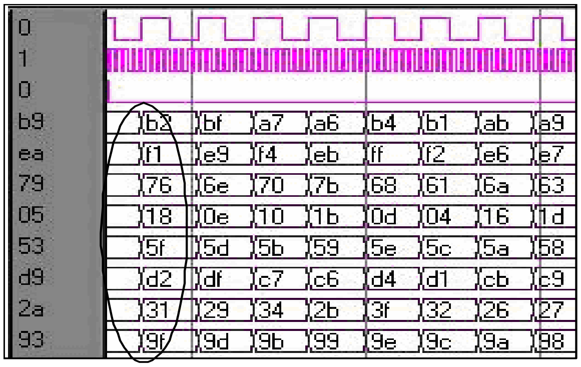
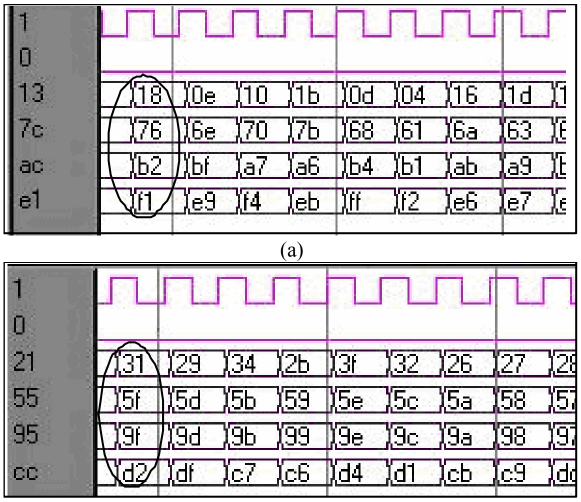

0
1
0
b9 b2 bf a7 a6 b4 b1 ab a9
ea f1 e9 f4 eb ff f2 e6 e7
79 76 6e 70 7b 68 61 6a 63
05 18 0e 10 1b 0d 04 16 1d
53 5f 5d 5b 59 5e 5c 5a 58
d9 d2 df c7 c6 d4 d1 cb c9
2a 31 29 34 2b 3f 32 26 27
93 9f 9d 9b 99 9e 9c 9a 98

Fig. 5: Outputs from downlink module

In Detect-OC-ID module, the tags' ID will be put into the specific register according to its OC value. ID with smallest OC value will be put on the first register, follow by the ID with the smaller OC value and finally ID with the largest OC value will be put on the last register. For example, for the input as marked by a circle in Fig. 4 will be arranged as $B2_{16}$, $F1_{16}$, 76_{16}, 18_{16},

$5F_{16}$, $D2_{16}$, 31_{16} and $9F_{16}$ are marked by a circle in Fig. 5. Therefore, the tags' ID with the same OC will be grouped in the same register. These IDs will be fed into the Uplink module; eight IDs are simultaneously fed at every Preselect clock cycle.



(a)

(b)

Fig. 6: Output of two multiplexed fast-search modules. (a): 1st Fast-search module, (b): 2nd Fast-search module

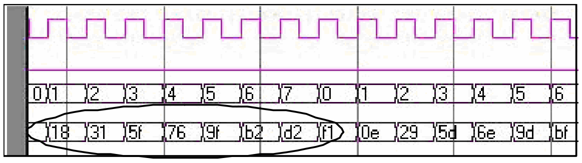


Fig. 7: Output of read-tag module

The output data from the Downlink subsystem will be divided into two, four input data will go to the First Fast-search and another four will go to the second Fast-search module. Figure 6a shows the output data of the first Fast-search module. For example the identified tags of the first Tag clock cycle are 18_{16}, 76_{16}, $B2_{16}$ and $F1_{16}$ is marked by a circle. Meanwhile, Fig. 6b shows the output data of the second Fast-search module. In this module the identified tags at the first Tag clock cycle are 31_{16}, $5F_{16}$, $9F_{16}$ and $D2_{16}$ is marked by a circle.

Figure 7 shows the output data of the Read-tag Module which display the tag's IDs serially. For every Read cycle eight tags are identified which equal to one tag for every System clock cycle. For example, for the first selected group of the tags as shown in Fig. 4, the tags are identified as 18_{16}, 31_{16}, $5F_{16}$, 76_{16}, $9F_{16}$, $B2_{16}$, $D2_{16}$ and $F1_{16}$ is marked by a circle.

Hardware Output: A development of anti-collision algorithm in hardware is to achieve the high performance features such as area, latency, cycle time and throughput. Hence, the Higher Throughput Anti-Collision Algorithm (HTACA) system has been implemented in hardware using FPGA model Virtex II Xc2v250. The output waveforms from the FPGA have been displayed using the Tektronix Logic Analyzer model TLA 5201 for real time verification. From the results, it shows that the HTACA can still identify the tags at the operating frequency of 180 MHz without errors.

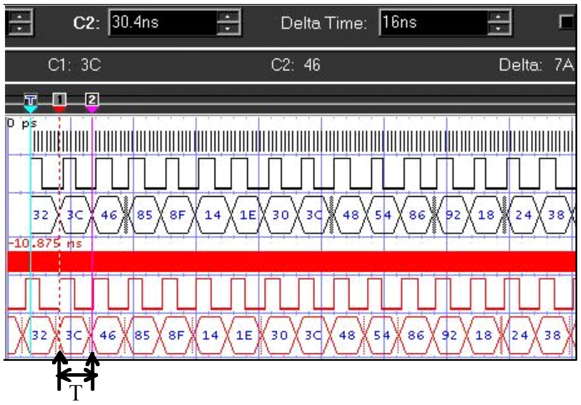

C2: 30.4ns
Delta Time: 16ns
C1: 3C
C2: 46
Delta: 7A

Fig. 8: 60 MHz output waveform

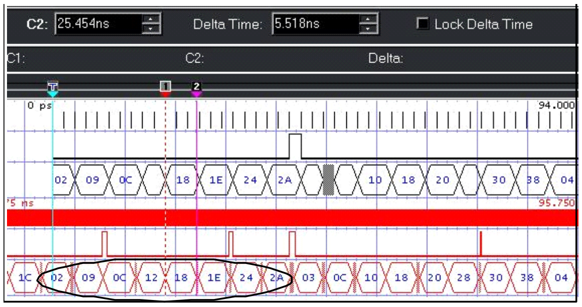

C2: 25.454ns
Delta Time: 5.518ns
□ Lock Delta Time
C1:
C2:
Delta:

Fig. 9: 180 MHz output waveform

Figure 8 shows the tag's IDs have been verified using Logic Analyzer for operating frequency of 60 MHz. From the output, it shows that the identification time (*T*) of the tag is equal to one System clock cycle as displayed by the Delta Time. For one Read cycle, eight tags have been identified from the smallest ID to the largest ID value. For example for the Read cycle as marked by a circle, the identified tags are 14_{16}, $1E_{16}$, 30_{16}, $3C_{16}$, 48_{16}, 54_{16}, 86_{16} and 92_{16} accordingly.

Figure 9 shows at 180 MHz, the HTACA system still enables to identify the tag's IDs correctly at every Read cycle. For examples for the first Read cycle, the identified tags are 02_{16}, 09_{16}, $0C_{16}$, 12_{16}, 18_{16}, $1E_{16}$, 24_{16} and $2A_{16}$ are marked by a circle. For the second Read cycle, the identified tags are 03_{16}, $0C_{16}$, 10_{16}, 18_{16}, 20_{16}, 28_{16}, 30_{16} and 38_{16} etc. As a result, the maximum operating frequency of the HTACA is 180 MHz which equal to the maximum frequency of the used FPGA model. Therefore, by using FPGA with higher frequency, the maximum operating frequency should more than 180 MHz. Eventhough the clock waveform cannot be displayed in this frequency, the operating frequency of the system can be verified by using the ID duration. As shown in Fig. 8, the ID duration is equal to one cycle (period) of the system clock. Therefore, the identification time (*T*) of the tag is 5.52 ns as displayed by the Delta Time.

From the Behavioral simulation result, it shows that the time to load the eight tags into the Preselect module is equal to eight cycles of the System clock. Hence one tag is loaded at every system clock cycle. As a result, the duration to load one tag into the HTACA system is the same with the duration to output the tag from the HTACA system. This duration is equal to one System clock cycle which also equal to the tag identification time (*T*). From the real verification results, it shows that the maximum operating frequency of the HTACA system is 180 MHz. Therefore, the tag identification time (*T*) is equal to 5.5 ns. Meanwhile, the maximum throughput of the HTACA system is equal to inverse of the tag identification time. As a result, the maximum throughput of the HTACA system is 180 Megatags sec^{-1} and is also known as maximum identification rate.

CONCLUSION

A proposed Higher Throughput Anti Collision Algorithm (HTACA) is presented to achieve a maximum throughput for tag identification. The throughput is maximized by combining a Pipeline technique and a deterministic anti-collision technique in the identification process. The HTACA system has been successfully implemented in hardware using FPGA model Virtex II Xc2v250. From the Behavioral simulation results, it shows that the duration to input and output a tag to and from the HTACA system is occurred in the same System clock cycle. These simulation results have been verified in real time using Tektronix Logic Analyzer model TLA 5201. The verification results show the maximum operating frequency of this hardware implemented system is 180MHz. As a result, the maximum throughput of the system is 180 Megatags sec^{-1}.

REFERENCES

1. Finkenzeller, K. and R. Waddington, 2003. RFID Handbook: Fundamental and Applications in Contactless Smart Cards and Identification. John Wiley and Sons, pp: 200-219. 2nd Edition, West Sussex, England, ISBN 0-470-84402-7
2. Zhou, F., D. Jing, C. Huang and H. Min, 2004. Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems. In Proceeding of the International Symposium on Low Power Electronics and Design, 2004, Newport Beach, California, USA, pp: 357-362. DOI: 10.1109/LPE.2004.1349366 http://ieeexplore.ieee.org/
3. Law, C., K. Lee and K.Y. Siu, 2000. 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, ISBN:1-58113-301-4 http://portal.acm.org/
4. Shih, D. H, Sun, P. L, Yen, D. C. and Shi-Ming Huang S. M, 'Taxonomy and survey of RFID anti-collision protocols', Computer Communications, Vol. 29, Issue 11, July 2006, pp. 2150-2166, DOI: 10.1016/j.comcom.2005.12.011 http://www.elsevier.com/locate/comcom
5. Choi, H.S., Cha, J.R. and Kim, J.H., 'Improved Bit-by-Bit Binary Tree Algorithm in Ubiquitous ID System', Proceeding of the International Conference on Advances in Multimedia Information Processing, 2004, LNCS 3332, pp. 696-703,DOI:10.1007/b104117 http://www.springerlink.com/
6. Victor, P., H. Nelson, N. Troy, D.C. Bill and J. David Irwin, 1995. Digital Logic Circuit Analysis and Design. Prentice Hall, Inc., pp: 789-831. 1st Edition, Englewood Cliffs, New Jersey, ISBN 0-13-463894-8
7. Sarma, S., D. Brock and D. Engels, 2001. Radio frequency identification and the electronic product code. IEEE Micro, 21: 50-54. DOI: 02721732/01 http://ieeexplore.ieee.org/
8. Michael, D.C., 2003. Advanced Digital Design with the VERILOG HDL. Pearson Education, Inc. 1st Edition, Upper Saddle River, ISBN 0-13-121974-X
9. Giovanni De Micheli, 1994. Synthesis and Optimization of Digital Circuits. Mc GrawHill, pp: 21-27. 1st Edition, Hightstown, ISBN 0-07-113271-6
10. Jahariah, S. and O. Masuri, 2008. Fast detection anti-collision algorithm for RFID system implemented on-chip. J. Applied Sci., 8(7): 1315-1319. ISSN: 1812-5654 http://www.ansijournals.com/