

Application of ICA in Collision Resolution for Passive RFID Communication

Yuan Sun, Peter J. Hawrylak, *Member, IEEE*, and Marlin H. Mickle, *Life Fellow, IEEE*

Abstract—In ISO 18000-6c passive RFID (EPCglobal Gen2) communication, a tag collision is caused when multiple tags respond to the reader simultaneously. This research proposes a means to resolve the tag collision using the well known statistical signal processing technique Independent Component Analysis (ICA) in real time as specified by the protocol. The hardware fixed point algorithm was optimized for FPGA implementation and achieved satisfactory resolution results on both stationary and moving tags.

Index Terms— Passive, RFID, Collision, Independent Component Analysis (ICA).

I. INTRODUCTION

According to ISO 18000-6c Passive RFID standard [1], the communication between the reader and tag follows a reader-talk-first mode. After receiving an effective reader command, the tag in the field directly modulates the received 915MHz (UHF) reader carrier wave with its baseband data and backscatters the modulated signal to the reader. Although this feature makes the passive RFID tags easy and inexpensive to deploy and maintain compared to active tags, the passive RFID tags are subject to critical communication timing constraints: a reader which fully conforms to the standard, must realize the tag inventory round including a three step handshake as shown in Figure 1 in order to make the tag transit into the data access states, which allows for tag memory access. At Step.1 of the handshake, the reader sends a Query command; the tag chooses a random time slot to backscatter its 16-bit random number (RN16) after receiving an effective Query command, and then transits from the Arbitrate state to the Reply state. In Step.2, the interrogator decodes the tag backscattered random number and attaches it to the command header of an acknowledge command ACK, and then sends out the ACK command within a specified turn-around time. The tag receives this ACK command and responds with its ID protected by a 16-bit CRC. The tag then transits into the Acknowledged state. At Step.3, the reader receives the tag response and sends a Req_RN command with the previous tag backscattered RN16 and CRC over the command within the same turn-around time in Step.2 to notify the tag is entering into the memory access state. The tag then backscatters a Handle (16 bit number). The effective

turn-around time between each of the three steps lies in the range of from 31.25 μ s in the worst case when the tag back link frequency (BLF) reaches 640KHz to 0.5ms in the best case when BLF is as low as 40KHz, which implies that the reader circuitry shall complete the signal decoding and command assembly work in real time.

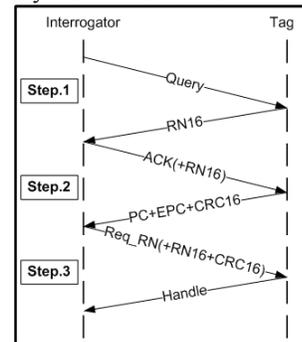


Fig.1. the 3-Step Handshake Inventory Round

The collisions in UHF passive RFID communications lie in two major categories: the reader collision and the tag collision. Reader collisions occur in a reader intensive environment when the signals from two or more readers overlap in time and frequency. Tag collisions occur in a tag intensive environment when multiple tags are present in the reader transmitting field. In such situations, tags may respond to the reader Query command simultaneously causing the reader to fail to decode the received signal, which is the result of collision. This paper focuses on the resolution of tag collisions. Currently, to address the tag collision problem, multiple anti-collision protocols enabling the passive RFID tags to take turns in transmitting to a reader have been developed. Generally, there are two types of anti-collision protocols in common use based on time division multiple access (TDMA): One is the dynamic framed slotted Aloha, the other is the Binary Tree scheme [2]. ISO 18000-6c RFID systems use the dynamic framed slotted Aloha. The dynamic framed slotted Aloha is a probabilistic method which can decrease the probability of collision occurrences significantly, but collisions cannot be completely avoided.

As a type of source separation method based on signal statistical characteristics in wide use, the Independent Component Analysis (ICA) provides a promising solution for online passive RFID communication collision resolution. ICA is a method for finding underlying factors or components from multivariate statistical data, and it looks for components that are both statistically independent, and with nongaussian distribution. Because the responses from colliding tags are all 16-bit random numbers and are driven by an independent clock signal on each tag, they are

Yuan Sun is with the Electrical and Computer Engineering Dept., University of Pittsburgh, USA (e-mail: yus22@pitt.edu).

Peter J. Hawrylak is with the Electrical and Computer Engineering Dept., University of Pittsburgh USA (email: pjhst23@pitt.edu@pitt.edu).

Marlin H. Mickle is with the Electrical and Computer Engineering Dept., University of Pittsburgh USA (e-mail: mickle@pitt.edu).

statistically independent. Because the baseband binary tag responses consist of only two separate logic values 0 and 1 (or +1 and -1), they are intrinsically super-Gaussian distributed characterized by two distinct peaks located near the two logic values in probability density function (pdf) curve. Therefore, the collision signal satisfies with the prerequisites of the ICA method.

The remainder of the paper is organized as follows: In section II, the architecture of the data acquisition platform and signal baseband is presented. Section III introduces the characteristics of the observed collision signal. Section IV introduces the online ICA algorithm used in this application. Section V presents the FPGA implementation of the collision resolution algorithm and corresponding experimental results. Finally, concluding remarks are provided in section VI.

II. DATA ACQUISITION PLATFORM

The communication between the reader and tag is realized by a reconfigurable software defined radio data acquisition platform. Both the software and hardware of the platform are programmed and controlled by National Instruments (NI) LabVIEW® software. The platform includes three frequency bands: the digital baseband running at the RFID command symbol rate, the digital/analog intermediate frequency band (IF), and the RF front end running at UHF. The baseband is realized in a Xilinx Virtex-II Pro XC2VP30 FPGA performing interrogator command and tag response processing in the digital domain, while the IF band performs the first stage ASK modulation and demodulation. The baseband and the IF band are together in the NI PCI-5640R Software Defined Radio transceiver board.

The architecture of the platform hardware is shown in Figure 2. The 2-way signal flow includes the transmitter side and the receiver side. On the transmitter side, the software running on a host PC selects and sends out user specified RFID commands to the FPGA baseband. The FPGA assembles the received binary commands according to the standard and then passes the data into IF band. The IF stage includes an Analog Devices AD9857 14-bit quadrature digital upconverter and an AD6654 14-bit downconverter. With the built-in Numerical Controlled Oscillator (NCO), the IF upconverter modulates the baseband data at 25MHz. The 25MHz IF ASK signal is then sent into the RF front end tuned to 915MHz for RF stage modulation. The RF front end consists of an NI 5610 RF upconverter and an NI5600 RF downconverter, which are connected to the host PC by NI PXI bus. On the receiver side, the tag backscattered signals (after receiving effective interrogator commands) pass through the 2-stage ASK demodulation in RF and IF bands, and then enter into the FPGA baseband. For comparison purposes, an Agilent E4443A real time spectrum analyzer is employed as an auxiliary monitor for the RF communication process. The FPGA baseband decodes the received signals, and then sends the received signals as well as the decoded binary data after decode back to the host PC for display and analysis.

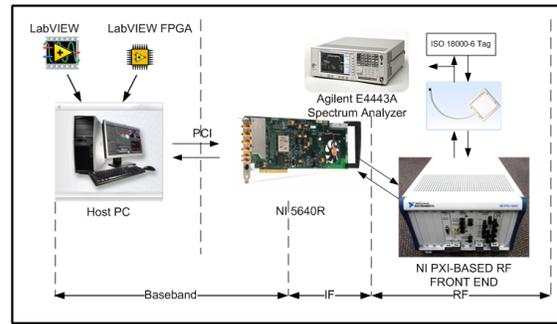


Fig.2. Platform Hardware Architecture

The FPGA baseband consists of four parts to realize the ISO 18000-6c standard: the ASK modulator, the real-time DSP unit, the signal decode module, and the processing unit for the previously mentioned three step real time handshake. On the transmitter side, the FPGA baseband receives the binary command from the host PC control panel, and then modulates the data stream using ASK after performing encoding. The PIE encoded data is sent into two separate quadrature channels I and Q in the IF stage. The DSB-ASK and SSB-ASK modulation manner can be selected by controlling the strobe of the Q channel. In the transmitting channels, the ASK modulated signal is usually subject to a potential bandwidth limit caused by the filters in the IF stage. The real-time DSP unit is responsible for guarantying the transmitting/receiving signal quality and realizing the collision resolution algorithm as will be discussed in Section V. At the first stage of the IF upconverter (i.e. AD9857), there exists an interpolation filter with cutoff frequency of 40% of the incoming data sampling rates. However, the square wave like command bit stream generated by the FPGA contains a theoretically unlimited bandwidth. If it is passed into the IF upconverter without bandwidth limiting, the so called Gibbs phenomenon occurs, resulting in significant over/under shoot at the data edges which degrades the output signal. The over/under shoots of the output command usually exceed the maximum tolerance specified in ISO 18000-6c standard, and make the outputting command signal invalid for the tag in test. To improve the reader transmitting signal quality, a low pass filter is realized in the FPGA baseband to limit the bandwidth of the outputting baseband signals below the cut off frequency of the interpolation filter in IF upconverter. With the LabVIEW Digital Filter Design (DFD) toolkit, the tap coefficients for an 8-order Bessel FIR low pass filter are generated as shown in Figure 3. Comparing the baseband outputs, it can be shown that the under/over shoots in command signals are significantly quenched after the smoothing.

Unlike the reader, which uses a fixed baseband frequency, the tag BLF can vary from 64kHz to 640kHz. Therefore, improving the quality of the received baseband signal requires adapting tap coefficients for different typical BLFs if using an FIR or IIR filter. As an alternative, a 10 order fixed point median filter using bubble sorting is realized in the FPGA. As shown in Figure 4, the SNR of the received tag response is significantly improved using the median filter, which thereby decreases the bit error rate of the decoding and accelerates the convergence of ICA algorithm. A trade off can also be made between the complexity of the median filter

and its effect as shown in Figure 5. Normally, a median filter with 4 orders which guarantees a 3dB SNR improvement over the original signal is enough for the application.

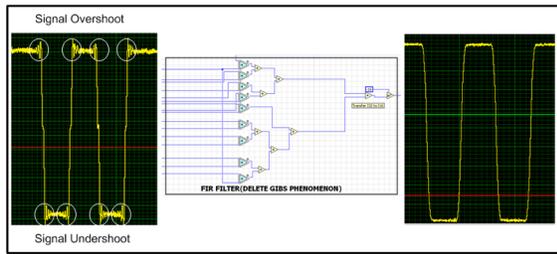


Fig.3. Signal Smoothing with 8-order FIR Filter

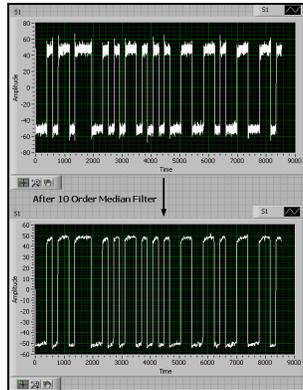


Fig.4. the effect of a 10-order median filter

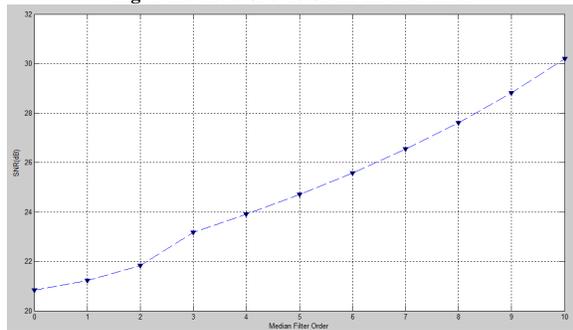


Fig.5. Median order vs. Processed Signal SNR

III. COLLISION SIGNAL CHARACTERISTICS

According to the ISO 18000-6c standard [1], tags shall encode the backscattered data as either FM0 baseband or Miller modulation of a subcarrier at the data rate (BLF). The reader commands the encoding choice, and both FM0 and Miller are bi-phase space encoding technology. FM0 inverts the baseband phase at each symbol boundary; a data-0 has an additional mid-symbol phase inversion. Baseband Miller inverts its phase between two data 0's in sequence. Baseband Miller also places a phase inversion in the middle of a data 1 symbol. Generally, the symbol of both FM0 and Miller can be categorized into formations as shown in Figure 4. Formation 0 features an edge transition in the middle of the symbol, while there is no edge transition in the Formation 1 symbol.

By configuring the command parameter, the reader sends a Query command with its Q (6c random number range) field equaling zero which forces two tags in the field to respond their RN16 simultaneously. At the receiver side, an acquired collision signal is captured as shown in Figure 7. The

collision signal is the result of the linear superposition of tag responses, The two tag responses received are likely different in magnitude because of different propagating paths and the effect that the PLL in the receiver locks onto the relatively stronger transmitting carrier wave rather than the tag backscatter carrier wave. Therefore, according to permutation, four possible voltage levels can be obtained from their linear superposition as can be observed. A phase shift caused by different propagate paths and the tags RF front end circuitry physical difference can also be observed in the collision signal, and the phase shift value is not fixed due to the tag BLF deviation. As can be seen, the initial phase shift is minimal, but it accumulates as time increases. Multiple tag collisions follows the same linear additive model as 2-tag situation, but the increased number of voltage levels and the phase shift relationship in the signal complicates the waveform.

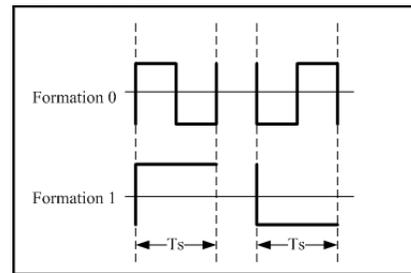


Fig.6. Generalized Tag Baseband Formation

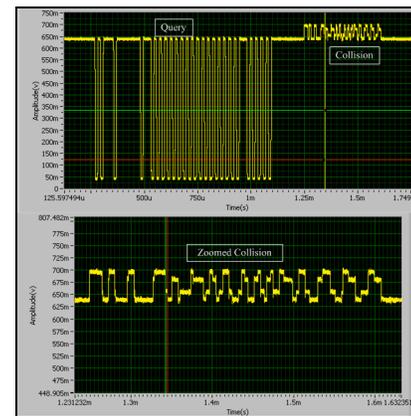


Fig.7. Two-Tag Collision Signal

IV. RESOLUTION ALGORITHM

The intrinsic source independency and the super-Gaussian statistical distribution of the tag responses make ICA an appropriate candidate for resolving multiple tag collisions. According to the ICA mixing model as described in Eq. 1, the collision can be represented as a vector X , where each vector variable corresponds to a received mixture from one channel; the source can be represented as a vector S , where each vector variable corresponds to a source signal (individual tag response). The mixing matrix is represented as A . Assume the number of the conflicting tags (the size of the S vector in Eq. 1) is M , and the number of receiving channels (the size of the X vector in Eq.2) is N , ICA requires that N shall be at least equal to M , i.e., there shall be no less captured mixtures than the mixing sources. Therefore, in

order to resolve M tag collisions, at least M receiving channels are required. If the number of the receiving channels is more than the number of colliding tags, a Principle Component Analysis (PCA) can be performed to extract the N most significant components from the mixtures in order to make the mixing matrix A in Eq.2 square (because only a square matrix has inverse). ICA is then performed to find the inverse of A (an N by M matrix) in an iterative manner until the algorithm converges.

$$X = AS \quad (1)$$

(where $X = (x_1, x_2, \dots, x_m)'$, $S = (s_1, s_2, \dots, s_n)'$)

Various practical methods for employing the ICA model can be used for resolving the tag collision signal. Among the available methods, one approach based on minimizing Entropy of the collision signal named FastICA[3] features medium computation load with reasonable separation quality. It is thus selected to be used in the ICA model for multiple tag collision resolution, which requires online real time signal processing. In information theory, the Entropy H of a signal y (as described in Eq. 2) is a measure of non-Gaussianity. (Where, $P_y(y)$ is the pdf function of signal y.)

$$H(y) = - \int P_y(y) \log P_y(y) dy \quad (2)$$

According to information theory, a Gaussian signal has the largest Entropy among all random signals of equal variance. In probability theory, the central limit theorem (CLT) states conditions under which the sum of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed [4]. According to the ICA model in Eq. 1, the mixture signal X (multiple tag collision signal) is the linear superposition of the source signal (the response of each tag), which implies that the distribution of the mixture signal approaches to normal distribution (more Gaussian) more than the source signals. Therefore, the ICA algorithm resolves the collision signal by minimizing its Entropy (Gaussianity), because once the source signals are recovered, they are with minimum Entropy compared to the collision signal. To obtain a measure of non-Gaussianity that is zero for a Gaussian variable and always nonnegative, a normalized version of differential entropy, called negentropy J as defined in Eq. 3 is used where y_{gauss} is a Gaussian random vector. Negentropy is always nonnegative, and it is zero if and only if signal y has a Gaussian distribution.

$$J(y) = H(y_{\text{gauss}}) - H(y) \quad (3)$$

Although Entropy and Negentropy can be used to measure the Gaussianity, the integral in the calculation hinders computation efficiency. As an effective approximation, the Negentropy of a signal can be calculated without involving the integral as shown in Eq.4, where G is a nonquadratic function:

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \quad (4)$$

(where $G(u) = (1/3)u^3$)

The FastICA algorithm performs a Gaussian-Newton optimization when maximizing the Negentropy. Following

Eq. 1, the source can be recovered from the collision signal X by left-multiplying the collision with a separating matrix W as shown in Eq. 5, which is the inverse of the mixing matrix X. Table 1 lists the FastICA algorithm using Negentropy maximization.

$$S = WX; \quad (5)$$

(where $X = (x_1, x_2, \dots, x_n)'$, $S = (s_1, s_2, \dots, s_n)'$)

Table 1. FastICA Algorithm

1. Preprocessing the collision signal X by removing the mean
2. Whiten the data to Z, ($Z=(z_1, z_2, \dots, z_n)'$)
3. Initialize the w' vectors (each row) in the separating matrix W
4. $w = E\{Zg(w'Z)\} - E\{g'(w'Z)\}w$, where g is the first derivative of the G function in Eq.5.
5. Let $w = w/\|w\|$
6. If not converged, go back to step 4.

V. FPGA REALIZATION

A. Experiment Setup and Algorithm Realization

As required by the ICA model, to resolve a 2-tag collision, two independent receiving channels are deployed as shown in Figure 8. As shown, two RF front ends are connected to the FPGA baseband to capture the collision signal. After the data acquisition, the data is passed through the median filter in the FPGA as introduced in Section II.

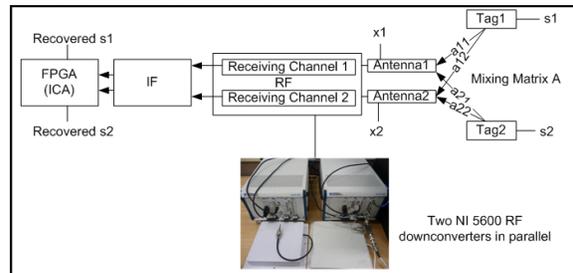


Fig.8. 2-Tag Situation Experiment Setup

The data acquisition resolution (incoming data sampling frequency) is 25MHz (0.04μs data point spacing) to keep the timing information included in the tag response accurate. Before the FPGA fixed point ICA of the processing, a Host PC floating point ICA algorithm simulation is performed. The resolution result (for BLF=64kHz) is shown in Figure 9.

However, the large data size (in the worst case when BLF=64kHz, the length of the tag responded RN16 can be over 9000 fixed point vectors) which burdens the FPGA based fixed point ICA to converge in real time. Therefore a tradeoff between the ICA training sample size and resolution accuracy needs be made to reduce the computation load. Realizing the fact that ICA was originally designed for source separation in application featuring a much more complicated formation (in certain time variant cases) than the simple square wave like a passive RFID signal, it is intuitive to select only part of the entire acquired collision signal to train the ICA algorithm. A sample of 1024 2-variable vectors. Each component corresponds to one channel. is chosen for the application when the BLF=64kHz, which corresponds to

40.96 μ s covering approximately 3 data bits in the collision signal(already containing adequate signal mixing information).

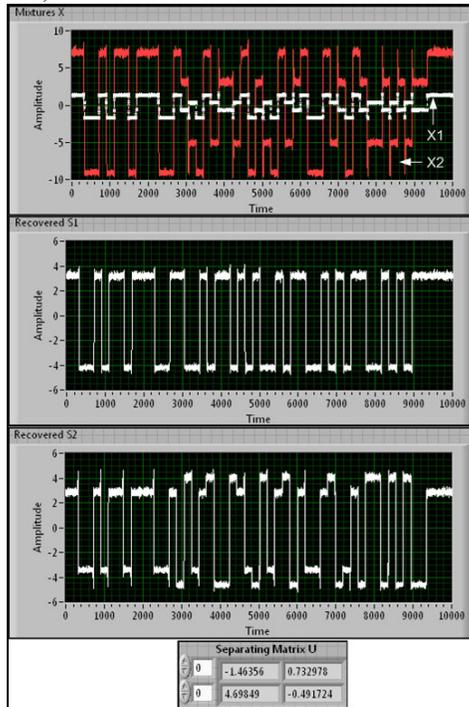


Fig.9. 2-Tag Collision Resolution Simulation Result

In addition, because the sampling frequency is 25MHz and the fastest tag BLF is only around 640kHz, a decimation factor of up to 16 can be imposed on the selected collision data segment to satisfy the Nyquist Theorem. Therefore, the original over 8000 data points of training sample is significantly reduced to only 32 fixed point data points. Figure 10 shows the FPGA based ICA resolution result when the tag BLF=64kHz. Due to the FPGA capacity limitation, only one tag data resolution circuitry is deployed in the hardware. Because the resolution is successful, which means the algorithm has already converges to the direction of one weight in the separating matrix, another weight pointing to the direction of the second tag response can be simply obtained by finding the quadrature direction of the first weight. The algorithm converges in 2 iterations, which corresponds to 22 μ s real time reported from the FPGA timer. After the ICA algorithm converges to the separating matrix, only 32 fixed point 2-point dot product calculations (2 for each data bit in the 16-bit RN16, with one for the first half bit and the other for the second half) need to be completed to decode the recovered data (“0” features the difference of the two halves, while “1” includes the same halves.). The 32 multiplications are realized by 32 parallel on-chip multiplication and accumulation units, and consumes 1 μ s as reported by the FPGA timer. The FPGA on chip resource utilization summary for 2-tag ICA resolution is listed in Table 2. As shown, the design fits the device well.

As an extension to multiple tags (more than 2) collision resolution, 3-tag collision signal resolution in FPGA is implemented as well. The three collision signals from each of the receiving channels are artificially synthesized by mixing a real 2-tag collision and a real individual tag response.

Figure 11 shows the resolution result when BLF=64kHz.

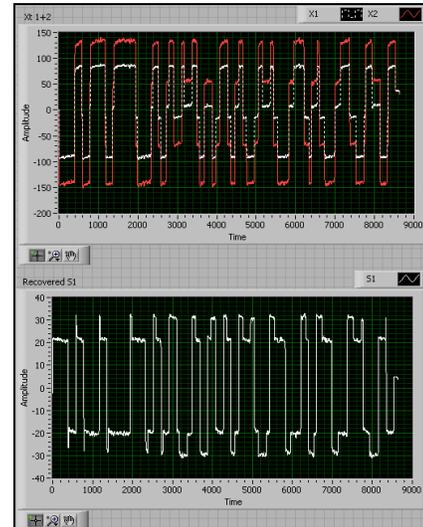


Fig.10. 2-Tag Collision Resolution in FPGA

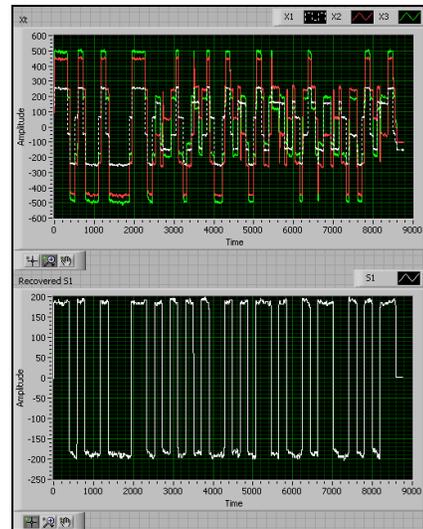


Fig.11. 3-Tag Collision Resolution in FPGA

Similarly, the ICA converges in 2 iterations corresponding to 36 μ s real time. The data bit recovery takes 32 fixed point 3-point dot product calculations consuming 1.28 μ s. The FPGA on chip resources utilization summary is listed in Table 3 for 3-tag ICA resolution. The total FPGA slice usage significantly increases for 53% from 2-tag design to 89% in 3-tag design.

Table 2. FPGA resources utilization for 2-tag ICA

(Xilinx Virtex-II Pro XC2VP30 FPGA)		
Number of BUFGMUXs	7 out of 16	43%
Number of External IOBs	403 out of 556	72%
Number of LOCed IOBs	403 out of 403	100%
Number of MULT18X18s	48 out of 136	35%
Number of RAMB16s	24 out of 136	17%
Number of SLICES	7262 out of 13696	53%

Table 3. FPGA resources utilization for 2-tag ICA

(Xilinx Virtex-II Pro XC2VP30 FPGA)		
Number of BUFGMUXs	7 out of 16	43%
Number of External IOBs	403 out of 556	72%
Number of LOCed IOBs	403 out of 403	100%
Number of MULT18X18s	56 out of 136	41%
Number of RAMB16s	32 out of 136	23%
Number of SLICES	12211 out of 13696	89%

B. Data Whitening

As a preprocessing step in the FastICA algorithm as described in Table 1, the whitening of the data samples can significantly simplify the ICA computation and thus accelerate the convergence [3]. The whitening process is as follows:

1. Calculate the covariance matrix of the training sample.
2. Calculate the eigenvalues and eigenvectors of the covariance matrix.
3. Form the whitening matrix as $V=E/D^{1/2}$, where D is a diagonal matrix with all the eigenvalues on the diagonal, and E is the matrix consisting of all the eigenvectors.
4. Multiply the training data with the whitening matrix.

Obviously, the whitening requires the Eigen decomposition of the covariance matrix of the training sample. This process is even more computation hungry than the ICA itself. According to experiment, the 2-tag collision resolution result is satisfactory even without data whitening because the simplicity of RFID signal formation. However, that is not true for 3-tag collision situation: the algorithm cannot converge after large number of iterations. In the current design, this problem is circumvented by using a universal whitening matrix instead of calculating the instant whitening matrix at every reading. The eigenvalues and eigenvectors of the covariance matrix of the collision signal represent the characteristics of the mixing matrix, thus if the mixing matrix is approximately fixed each time and the tags are from the same manufacturer, the covariance matrix of the acquired collision signals from receiving channels are statistically stable. Therefore, the experiment can be carried out by fixing the position of the reader and the three tags each time, and measuring a group of covariance matrices of the data, in order to calculate the expectation. This expectation matrix is then decomposed for the eigenvalues and eigenvectors, and used thereafter to calculate the whitening matrix as a universal whitening matrix. The obtained universal whitening matrix can be used to whiten the following collision signals from target tags at the fixed positions. This process is equivalent to a calibration before ICA processing. Future work can also be developed to incorporate the direct calculation of the whitening matrix in a more advanced hardware device.

C. Tag Dynamics

In addition to acquiring collision signals from stationary tags, the collision data are also captured from moving tags. Figure 12 shows the experiment set up. One tag is attached to the antenna, while another tag is attached to a moving CO₂ cartridge boosted wooden bullet. The FPGA is triggered by a photo sensor whenever the moving tag arrives at the antenna. The speed is measured as 25miles/hour. The ICA resolves the collision signal successfully.

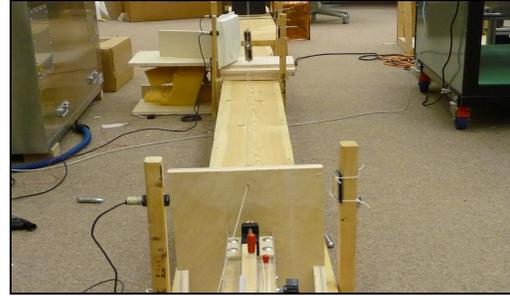


Fig. 12. Moving Tag Acquisition Setup

D. Processing Speed

Figure 13 summarized the FPGA collision resolution speed for 2-tag and 3-tag situations at typical tag BLFs from 64kHz to 682kHz. The standard required turn-around time is presented for comparison purposes. As shown, the resolution is satisfied with the required speed (time).

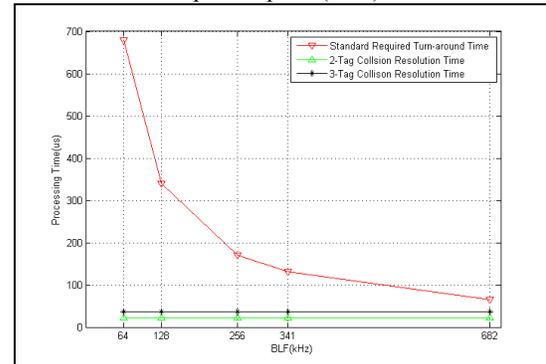


Fig. 13. Processing Speed vs. BLFs

VI. CONCLUSION

ICA presents a promising way to resolve the tag collision in ISO 18000-6c Passive RFID communication. This paper introduces the application of deploying a fixed point ICA algorithm FastICA in FPGA for multiple tags collision resolution. A tradeoff between resolution speed and effect is made to guarantee the algorithm converges in the standard specified real time. The design utilizes the FPGA resources properly, and works for collision signal acquired from both the stationary tags and moving tags.

If there are sufficient differences between the magnitudes of the two tag signals, it is possible to extract the strongest tag identification from the collision signal. The method of this paper is independent of the signal strength and provides both individual identifications from the received collision signal.

REFERENCES

- [1] EPCglobal, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Conformance Requirements V.1.0.2", 2005, <http://www.epcglobalinc.org>.
- [2] Joon Goo Lee, Seok Joong Hwang, Seon Wook Kim, "Performance Study of Anti-collision Algorithms for EPC-C1 Gen2 RFID Protocol", Proceeding of the International Conference on Information Networking, pp.523-532, Jan.2007
- [3] A. Hyvärinen, J. Karhunen and E. Oja. "Independent Component Analysis", John Wiley & Sons, New York, 2001
- [4] John A. Rice, "Mathematical Statistics and Data Analysis (Second edition)", Duxbury Press, CA, 1995