Parallel and Pipelined Hardware Implementation of Radar Signal Processing for an FMCW Multi-channel Radar

Eugin Hyun[1], Sang-Dong Kim[1], Jun-Hyeok Choi[2], Dong-Jin Yeom[3], Jong-Hun Lee[1]
[1]Robotics Division, Daegu Gyeongbuk Institute of Science & Technology,
50-1, Sang-Ri, Hyeonpung-Myeon, Dalseong-Gun, Daegu, 711-873, Korea
[2]ISR Center, LIGNex1,
207, Mabuk-ro, Giheung-gu, Yongin-si, Gyeonggi-do, Korea
[3]Agency for Defence Development,
Bugyuseong daero 488, beon gil, Yuseong, Daejeon, Korea
jhlee@dgist.ac.kr

***Abstract*—Ramp-sequence based frequency modulated continuous wave (FMCW) radar is effective in detecting the range and velocity of a target. However, because the target detection algorithm is based on a two-step fast Fourier transform (FFT) over several pulse-repetition intervals (PRIs), a significant amount of data must be processed in order to detect the range and velocity of target. In specific cases, when multiple channels must be supported in order to estimate the angle position of a target, even more hardware resources and memory, as well as longer processing times, are required. In this paper, a field programmable gate array (FPGA) based radar detection algorithm with a parallel and pipelined architecture is implemented in order to support the multi-channel processing of the algorithm, which includes range and Doppler processing, digital beam forming (DBF), and constant false alarm rate (CFAR) detection. In order to effectively support the parallel and pipelined architecture, we propose a data-routing-schemed DBF and fine-grained DBF architecture. The results from implementation of the proposed hardware resources and processing times are also presented. The implemented radar sensor is installed on an experimental vehicle and is demonstrated in the field.**

***Index Terms*—FMCW radar, vehicle radar, FPGA implementation, pipelining, parallel, detection algorithm.**

I. INTRODUCTION

In recent years, frequency-modulated continuous-wave (FMCW) radars have been used in vehicle applications. In FMCW radars, linear ramps are generated and transmitted. The frequency difference between a transmitted signal and its received reflection is used to detect the target range and velocity. The return signal, which has a different frequency, is called a ‘beat-signal’. Because the bandwidth of the received beat-signal decreases by less than a dozen MHz regardless of the transmitted bandwidth, the complexity of the signal processing can be reduced, compared with that of conventional pulse Doppler radar [1], [2].

In a radar system, the target range and velocity must be measured simultaneously with high accuracy. However, FMCW radar possesses ambiguities related to the separation of the range and velocity, which become more serious under multi-target situations [2]. In general, there are two approaches to resolving these range-velocity ambiguities.

In the first approach, slow ramps with different slopes are generated [3], [4]. In this algorithm, because the range and velocity are detected using a combination of several beat-frequency, an effective pairing algorithm is required for unique beat-frequency combination. Moreover, in order to obtain enhanced detection of moving targets, additional algorithms may be required (e.g., the moving target indication (MTI) algorithm, the clutter cancellation algorithm, or the ghost target suppression algorithm). However, these data association based algorithms have fundamental limitations related to the occurrence of ghost targets.

The second approach is the ramp-sequence based FMCW radar [4]–[6]. The basic concept is illustrated in Fig. 1. Here, Fig. 1(a) represents the signal shape in the frequency-time domain. The shape of the frequency sweep is saw-toothed, with the solid line representing the transmitted signal and the dotted line representing the received signal. Figure 1(b) presents a received beat-signal from a single target. The range information can be expressed as a frequency spectrum of each beat-signal; the Doppler-frequency appears as phase information over the all ramps in slow time domain.

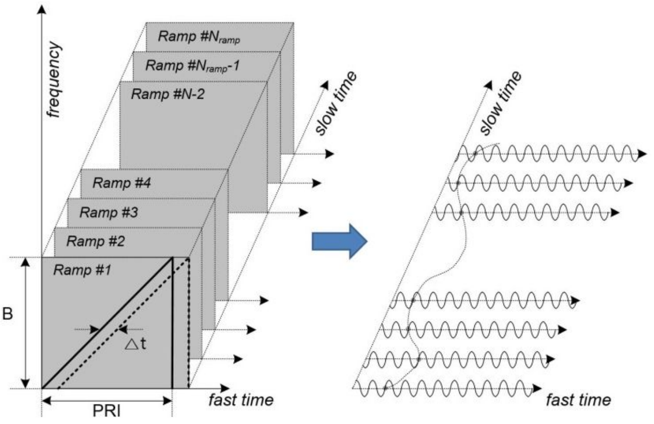
In this method, two-step fast Fourier transform (FFT) processing is used to detect the range and velocity. In this approach, because the clutter is diminished in all range-bins with a zero Doppler, stationary targets, including clutter, can easily be suppressed and moving targets can be easily distinguished.

However, because numerous ramps must be generated when using this method, substantial computational effort is required. In particular, to implement the multi-channel FMCW radar required to support angle estimation, the total computational complexity increases significantly. Therefore, there is a need for a 3D FFT-based signal processor designed to meet the required processing demands and to reduce the

Manuscript received July 16, 2014; accepted December 30, 2014.
This work was supported by the DGIST R&D Program of the Ministry of Science, ICT and Technology of Korea (15-RS-01). This work was also supported by Applied Research Project of ADD and LIGNex1.

required hardware resources.

Recently, with the needs of many radar applications outstripping the processing capabilities of digital signal processors, the use of field programmable gate arrays (FPGAs) has become an attractive solution toward better compact dedicated processors [7], [8]. That is, FPGAs provide a good combination of high speed implementation features, along with flexibility. For parallel construction, all functions that must be executed at the same time should operate with independent data. Pipelined processing requires that the function block be able to access data in the same direction.



a) b)

Fig. 1. Basic concept of the ramp-sequence based FMCW radar: (a) Transmitted signal and received signal in the frequency-time domain, and (b) Beat-signal for a single moving target in the slow time domain. B is the bandwidth, PRI is the pulse repetition interval, and Δt is the two-way delay time of the received signal reflected from a single target.

In this paper, the design and implementation of an effective parallel and pipelined hardware architecture intended to support a 3D FFT based radar signal processing algorithm is reported. In Section II, an overview of the target detection algorithm is presented. Issues about the parallel and pipeline processing, and the design of the radar signal processing, are discussed in Section III. In Section IV, the results from the FPGA implementation and experiments are given. The conclusions from our study are presented in Section V.

II. Detection Algorithm Overview

Figure 2 describes schematic of the radar signal processing algorithm implemented for a ramp-sequence based FMCW radar in this paper. The corresponding data flow of the 3D map is also presented. The signal processing procedure is divided into four steps: range processing, digital beam forming (angle processing), Doppler processing, and detection. Here, Q is the number of digitalized samples in a ramp, P is the number of ramps in a frame time, L is the number of received channels, M is the number of detected range bins, K is the number of detected angle bins, and N is the number of detected Doppler bins.

The received beat-signal can be expressed as in (1). Here, $q = 0 \sim Q – 1$, $p = 0 \sim P – 1$, and $l = 0 \sim L – 1$. The beat-signal is composed of the range, Doppler, and angle information of a target, where A is the signal amplitude, f_r is the range beat frequency, f_d is the Doppler-frequency, θ is the target angle position, T is the PRI, d is the distance between receiving antennas, and λ is the wavelength

$$S(q,p,l) = \mathrm{A} \times e^{j2® f_r q} \times e^{j2® \frac{d sin¯}{a} l} \times e^{j2® f_d Tp} S(q,p,l) =$$

$$= \mathrm{A} \times e^{j2® f_r q} \times e^{j2® \frac{d sin¯}{a} l} \times e^{j2® f_d Tp}. \qquad (1)$$

In (1), there are three terms that must be processed. The first exponential term, the range beat-frequency, is obtained using (2). Here, $m = 0 \sim 2{\cdot}M – 1$ and $w(q)$ is the window function.

Prior to FFT processing, the window is applied in order to suppress the side-lobe, and then the windowed signal is transformed using the $2{\cdot}M$-point FFT into the frequency domain within each PRI. In this FFT process, because a negative range beat frequency is not necessary to detect the target range, the FFT point is equal to $2{\cdot}M$, and the number of range bins is M for each ramp

$$X(m,p,l) = \sum_{q=0}^{Q-1} S(q,p,l) \times w(q) \times e^{-j2® \frac{q}{Q} m}. \qquad (2)$$

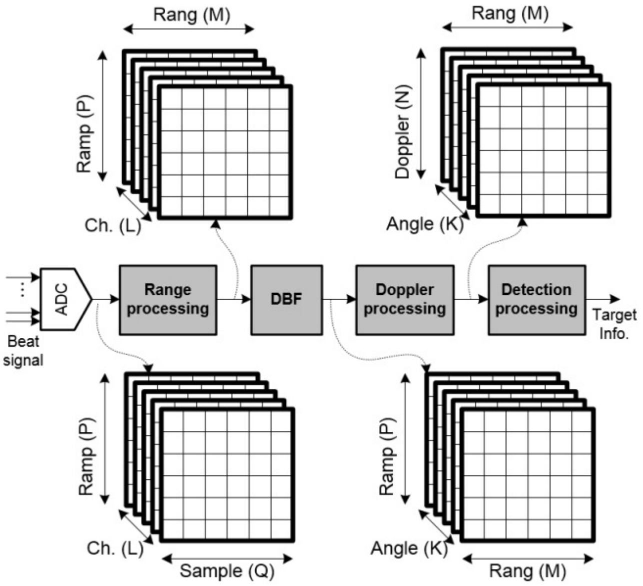


Fig. 2. Schematic of the implemented signal processing algorithm and corresponding data flow of 3D map.

The second term of (1) is the target angle term. This information can be extracted through digital beam forming (DBF), which is an advanced approach for steering receiving phased array antennas in order to estimate the angle. Using the data of the same single range-bin over all channels, DBF is conducted through windowing and L-point FFT in the angle-index direction. This process is presented in (3), where $k = 0 \sim K – 1$ and $w(l)$ is the window function

$$Y(m,p,k) = \sum_{l=0}^{L-1} X(m,p,l) \times w(l) \times e^{-j2® \frac{l}{L} k}. \qquad (3)$$

The final exponential term in (1) is the target Doppler term; the value is extracted based on (4), where $n = 0 \sim N – 1$ and $w(p)$ is the window function. The target Doppler-frequency spectrum is estimated using N-point FFT processing together with windowing inside the single range-bin and over a sequence of adjacent ramp signals in the PRI-index direction.

While only the positive frequency spectrum is required in the range-FFT process, all frequency spectra in the Doppler FFT, over the positive and negative Doppler frequencies are required in order to recognize whether the target is approaching or retreating

$$Z(m,n,k)=\sum_{p=0}^{P-1} Y(m,p,k)\times \ddot{}\,(p)\times w(p)\times e^{-j2\circledR \frac{p}{P}n}. \qquad (4)$$

Using the three-step processing in (2) to (4), a 3D map consisting of the range, Doppler, and angle information can be completed. In order to determine whether each cell of the 3D cube is a target or clutter, a final step of detection processing is conducted using the cell average-constant false alarm rate (CA-CFAR) detector in the range direction.

When using conventional CA-CFAR [9], comparison of the power spectrum density (PSD) of the current cell and the decision threshold, makes it possible to determine whether or not the current cell is the target. The decision threshold is calculated by averaging and scaling the PSD of the neighbouring cells of the current cell. The CA-CFAR procedure is conducted for the entire range cells using a sliding window.

III. Parallel and Pipelined Issue and Design

A. Architecture Design

In order to enhance the efficiency of the processing time, first, it is necessary for the pipelined architecture to effectively support the algorithms presented in Fig. 3. Here, input data of all blocks are composed of real and imaginary numbers, excepting for range processing with only real ADC data.

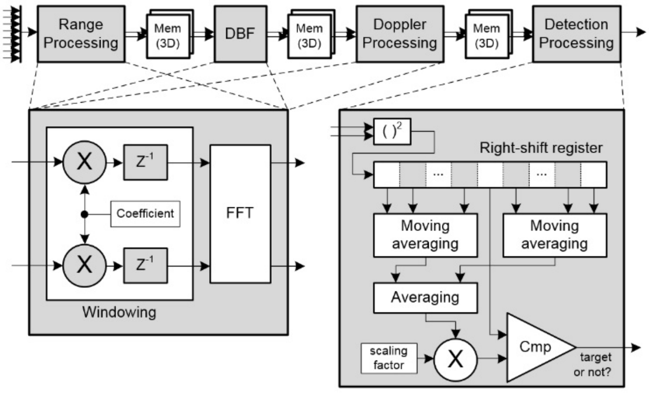


Fig. 3. Designed typical pipelined and non-parallel signal processing architecture.

In the typical architecture, the four pipelined processing groups with different directions of data flow are separated, and a 3D DPM (Dual-Port Memory) block is inserted between processing groups. The typical architecture based on DPM allows the windowing and FFT for the range processing to be conducted in the pipeline in the time direction. Similarly, the DBF consisting of windowing and FFT can be internally pipelined over the channel. The Doppler estimation based on windowing and FFT can be internally pipelined over the ramp. Finally, the CFAR can be processed in the range-bin direction. In each processing part, the corresponding computation is repeated until completion is achieved for all the 3D data presented in Fig. 2.

In this paper, in order to support the pipeline, the windowing algorithm is designed with a streaming architecture using Xilinx LogiCORETM IP Multiplier v11.2 (Xilinx, USA). In order to optimize the processing time, each FFT is also implemented using Xilinx LogiCORETM IP Fast Fourier Transform v7.0, which is based on Radix-2 Burst I/O architecture with a streaming 16-bit input and output. Moreover, the CA-CFAR, which is based on sliding-windowing, is implemented using the shifter registers and a moving averaging scheme for the full pipelined process, without a wait step between each calculation.

Compared to the typical architecture, in order to accelerate the computation time, parallel processing is required. To support parallel processing, all functions to be executed simultaneously must operate using independent data. In the range, Doppler, and CFAR detection, the parallel processing architecture is made possible by processing the data from the eight channels independently. However, in order to support the angle processing in parallel, a data exchange function is required before and after the DBF in the signal processing architecture described in Fig. 3.

That is because the DBF needs all FFT results from all channels of the same range-bin in order to estimate the angle-position of the target.

In this paper, we propose a data routing based pipelined and parallel architecture such as that shown in Fig. 4. In this architecture, we assume that the number of receive channels is 8 and the number of Doppler bins is 128.

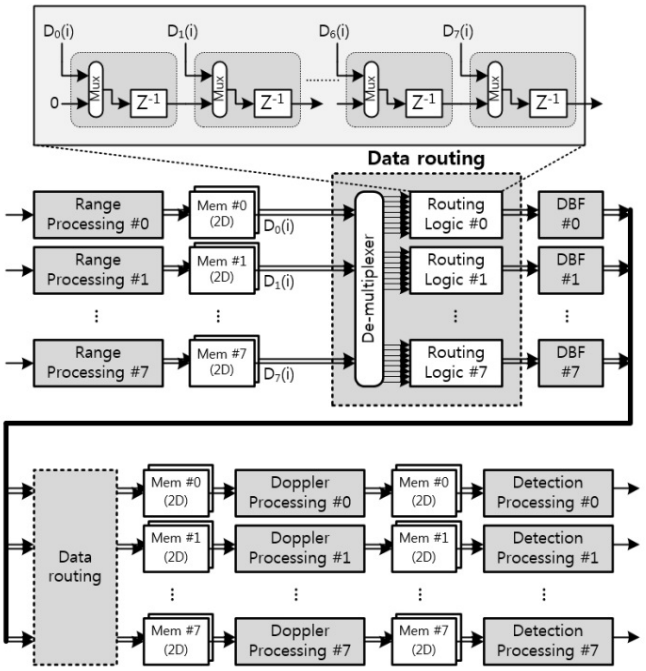


Fig. 4. Structure of the data routing based on the de-multiplexer and shift registers prior to the parallel and pipelined DBF.

First, for the pipeline processing of the windowing and FFT functions in the DBF block, multi-channel 2D data are simultaneously read from 8 memory blocks and are fed into each DBF block. Moreover, for parallel data conduction, the data are distributed to each DBF block in the proper order by a routing scheme.

In order to support the streaming data flow described in Fig. 5, the data routing logic routine is implemented prior to the DBF using a de-multiplexer with eight ports and eight-step shift registers with data-loading functions. Here, $D_0(i)$–$D_7(i)$ are the range FFT results determined using the ith range-bin index of each channel.

The scheme for DBF processing based on the de-multiplexer and the shift registers is as follows:

1. Data distribution to each DBF block using the de-multiplexer.
   – $D_0(0)$~$D_7(0)$ → DBF #0 block,
   – …,
   – $D_0(8)$–$D_7(8)$ → DBF #7 block.
2. Waiting period inserting until the processing of DBF Block #0 is completed.
3. Data saving to each memory block using multiplexer.
4. Above procedure is repeated until the eight DBF blocks finish calculations for m = 120 ~ 127, respectively.

For the structure shown in Fig. 4, even though the windowing and FFT are carried out in a pipeline, waiting time should be inserted. If the DBF processing can be performed without bubble time, and if the DBF function can be pipelined together with the range processing or the Doppler processing, more of the total algorithm calculation time can be saved. To this end, an architecture based on fine-grained pipelined DBF with a full IO streaming structure is designed.

Based on (3), the DBF equation is re-described as $Y_k(m,n)=\sum_{i=0}^{7} X_i(m,n)\cdot{}^{-}{}_{k}(i)$. Here, $X_i(m,n)$ indicates the range processing results in the ith channel and the $Y_k(m,n)$ and DBF results for the kth angle. Moreover, $\theta_k(l)$ is the DBF coefficient for the kth angle and expressed as $w(i)\cdot e^{-j2\text{®}\frac{k}{8}i}$. For example, for the angle-bin #0, the DBF result is expressed as $Y_0(m,n)=X_0(m,n)\cdot{}^{-}{}_{0}(0)+\ldots+X_7(m,n)\cdot{}^{-}{}_{7}(0)$. Therefore, $Y_0(m,n)$ can be estimated using only eight complex matrix multipliers. Here, because $\theta_0(0)$~ $\theta_0(7)$ can be stored in the look-up table (LUT) with the pre-calculated values, the constant values can be used.

This approach is able to simultaneously estimate eight angle-bins. In the proposed fine-grained pipelined DBF architecture, since the outputs of the range-FFT processing are directly fed into eight DBF blocks without memory or data routing, a fully pipelined process is possible. Figure 5 shows the newly designed fine-grained pipelined and parallel DBF architecture.

A processing structure for each angle-bin is composed of eight complex multipliers and one binary-tree adder.

In Fig. 6, the detailed architecture used to calculate angle-bin 0, i.e., $Y_0(m,n)$, in the proposed DBF block is presented. In this architecture, eight parallel complex-multipliers (CMs), seven complex adders, and 15 complex registers are used to achieve the fine-grained pipelined DBF architecture with parallel processing.

Unlike the pipelined architecture using the data routing scheme, two data routing blocks and one memory block are not necessary in the fine-grained pipelined DBF architecture. Thus, the total pipelined group can be decreased to three blocks, and the memory accessing time can also be reduced.

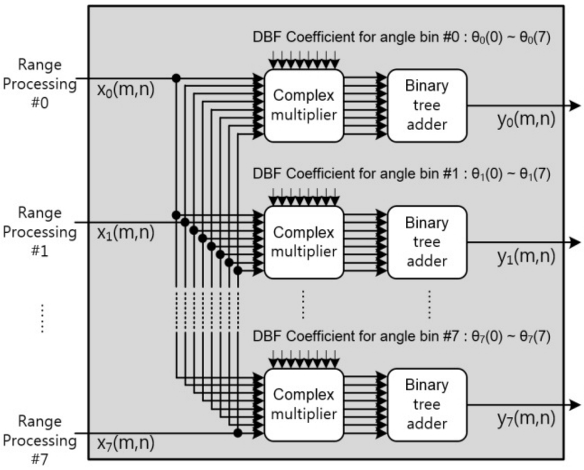


Fig. 5. Newly designed fine-grained pipelined DBF architecture with parallel processing.

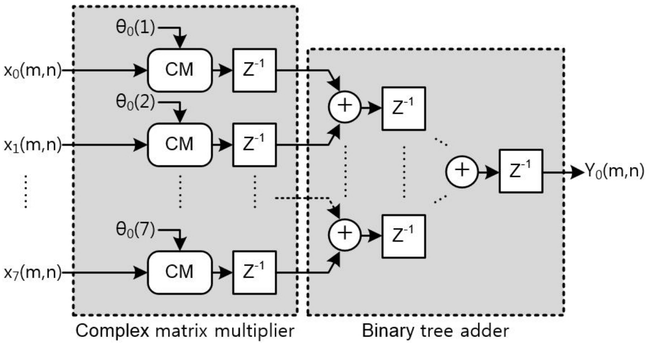


Fig. 6. Detailed depiction of the newly designed DBF architecture with a full pipeline path and parallel processing for calculation of angle-bin #0.

B. Time Complexity Comparison

In that case, the designed radar parameter values Q (sample number), P (ramp number), L (channel number), M (range bin size), K (angle bin size), and N (Doppler bins size) are set at 165, 118, 8, 128, 8, and 128 for this paper. Moreover, the window size G for CFAR detection is designed to have a value of 16. Based on these parameters, the processing time complexity of each algorithm is analysed as follows, without regard to the memory accessing time.

First, for range-processing, the windowing consumes the $1+Q$ clock for all samples in one ramp. Because the FFT point is $2{\cdot}M$, the time complexity is $2{\cdot}M{\cdot}log(2{\cdot}M)$. Therefore, the pipeline based range-processing requires $1+2{\cdot}M{\cdot}log(2{\cdot}M)$ clocks and the processing should be repeated P times for all ramps.

Similarly, the consumed clocks during Doppler processing can be estimated. Next, in the CFAR composed of the root-square, binary adder, scaling factor, and comparison operation, the consumed time complexity can be estimated as $4+log(G)\ +M$, and conduction is performed for every Doppler bin N.

In the pipelined DBF that uses the data routing scheme, the processing complexity of the DBF is $1+K{\cdot}log(K)$. The DBF should be repeated $(M/8){\times}P$ times.

Finally, the fine-grained DBF architecture is considered. In this case, because this architecture is full pipelined and parallel processing is performed, the total processing time can be reduced. The processing clock of the proposed DBF can be estimated as $1+log(K)+M$ because the complex multiplier and the binary adder are used; the total data length is M. Moreover, the proposed DBF is pipelined with range processing, the two processing can be recalculated together in $2 + (2{\cdot}M){\cdot}log(2{\cdot}M) + log(K) - M$ clocks. The procedure is completed rapidly every P for all ramps.

Table I presents a comparison of the time complexity results for the typical and the proposed architectures. Compared with that of the pipelined and non-parallel architecture, the time complexity of the pipelined architecture using the data routing scheme and the fine-grained DBF is reduced by more than 85 %. The fine-grained structure has a reduction ratio of 15 % compared with that of the data routing scheme.

TABLE I. COMPARISON OF TIME-COMPLEXITY RESULTS.

Algorithms	**Typical architecture**		**Proposed architecture**	
	Non-pipelined and non-parallel architecture	**Pipelined and non-parallel architecture**	**Data routing based pipelined and parallel architecture**	**Fine-grained pipelined and parallel architecture**
Range processing	2,090,016	1,934,256	241,782	227,622
DBF	498,432	377,600	47,200	
Doppler processing	1,161,216	919,552	114,944	114,944
Detection processing	139,264	139,264	17,408	17,408
Total	3,888,928	3,370,672	421,334	359,974

IV. IMPLEMENTATION AND EXPERIMENTAL RESULTS

A. *Hardware Implementation Results*

A block diagram of the Virtex-5 FPGA based radar signal processing firmware structure is presented in Fig. 7. The primary data processing path begins with eight parallel signals received from the transceiver (TRx) module. The received signal is sampled at the ADC clock, and the serial bit streams are then de-serialized into 14-bit words. After deserialization, the 14-bit words are saved into the dual port memory for synchronizing with the Tx trigger, which is generated by the digital direct synthesizer (DDS) controller, which is responsible for transmit wave generation.

Eight algorithm blocks are processed in-parallel and in pipeline, and one major control block manages these jobs. For each algorithm, the size of the input and output data set is designed to be a 16-bit signed fixed-point, except for the CFAR detection output. The final target detection information is transferred to the digital signal processor (DSP) and then resent to the host computer through the Ethernet. The host computer operates the radar signal processing module and corrects the target detection information.

In this paper, FMCW radar signal processing algorithms are implemented on a Xilinx Virtex-5 XC5VLX330 with sufficient internal resources. A Texas Instruments TMS3206455 DSP (TI, USA), which supports Gigabit Ethernet, is selected because it has sufficient internal memory, and because it also has a high quality processing clock. Currently, the role of the DSP is only that of a bridge controller between the FPGA and the radar operator; however, in the future, high-resolution angle-estimation algorithms and tracking algorithms will be implemented.

The implementation summary for the proposed signal processing system is presented in Table II. Here, one slice contains four LUTs and four flip-flops. One DSP48E device for fast calculations consists of a multiplier, an adder, and an accumulator. One block random access memory (RAM) is 36 Kbits in size [10].

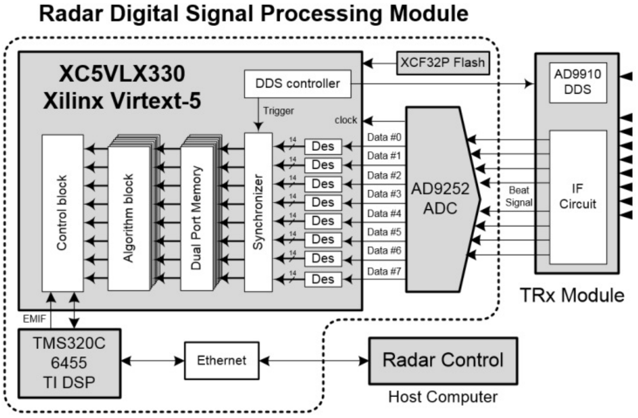


Fig. 7. Virtex-5 FPGA based radar signal processing structure with 8 receive channels.

Compared to the typical non-parallel architecture, in the designed pipelined and parallel architecture, while the slice registers and slice LUTs are more consumed, the required memory resources are similar. This is because the data received on all channels from the ADC are simultaneously saved into the necessary memory space.

The total processing time is estimated in the DSP by measuring the time from the request of the radar start command to the reception of all detected target information. From Table III, it can be seen that the total processing time is approximately 12.69 ms. The ADC data logging consumes 3.9 ms at the sampling frequency of 5 MHz; the algorithm is conducted using 8.79 ms at 50 MHz. However, because the data transfer time from the FPGA to the host computer through the DSP is not considered, the real operating time may be longer.

TABLE II. IMPLEMENTATION SUMMARY FOR THE PROPOSED SYSTEM WITH FINE-GRAINED PIPELINED AND PARALLEL DBF.

Device	Xilinx Virtex-5 XC5VLX330
Maximum frequency	103.036 MHz
Used slices register/Total	19,501/51,840 (38 %)
Used slices LUTs/Total	27,967/51,840 (54 %)
Used embedded memory/Total	244/288 (85 %)
Used DSP48Es/Total	168/192 (88 %)
Used I/O/Total	178/1200 (14 %)
Used system clock	50 MHz

TABLE III. TOTAL PROCESSING TIME FOR THE NEW ARCHITECTURE WITH FINE-GRAINED PIPELINED DBF.

Function	**Processing Time (ms)**
ADC data logging	3.9
Algorithm processing	8.79
Total	12.69

B. *Experimental Results*

The radar system is set up to evaluate the signal-processing module together with the developed transceiver module, including antennas such as shown in Fig. 8.

In Figure 8, the TRx module is developed based on a single transmit channel and multiple receive channels for each antenna. For the generation of frequency modulated waves, a DDS AD9910 (Analog Device, USA) is employed. This device is controlled by the signal-processing module. A single horn antenna is used for transmitting. Eight antennas of the same type, with half wavelength inter element spacing, are used for the multiple receiving channels. A PC, rather than a radar operator, is used to control the radar and to monitor the detection results.

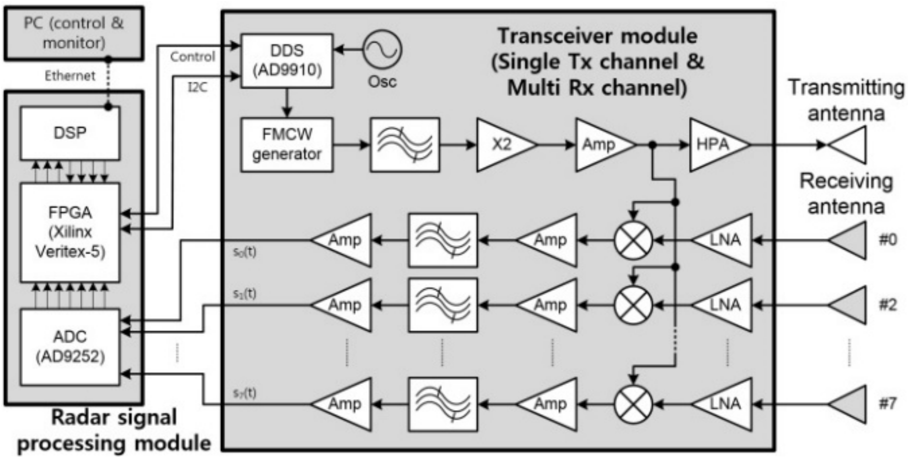


Fig. 8. Experimental setup for the radar-signal-processing module along with the TRx module, including antennas.

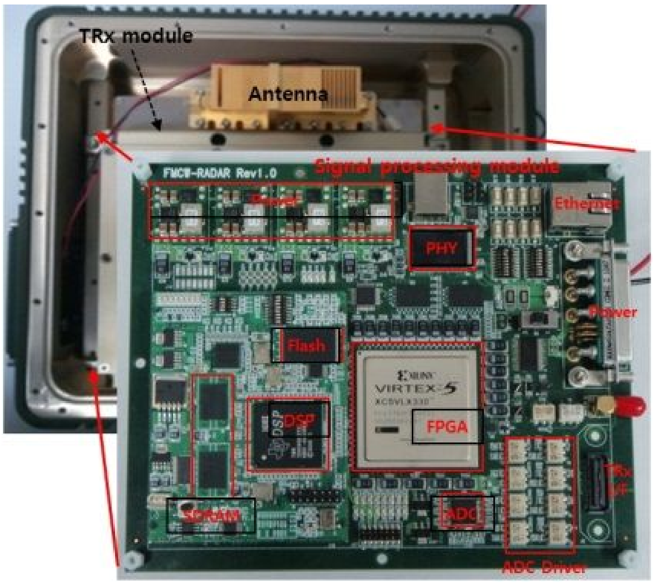
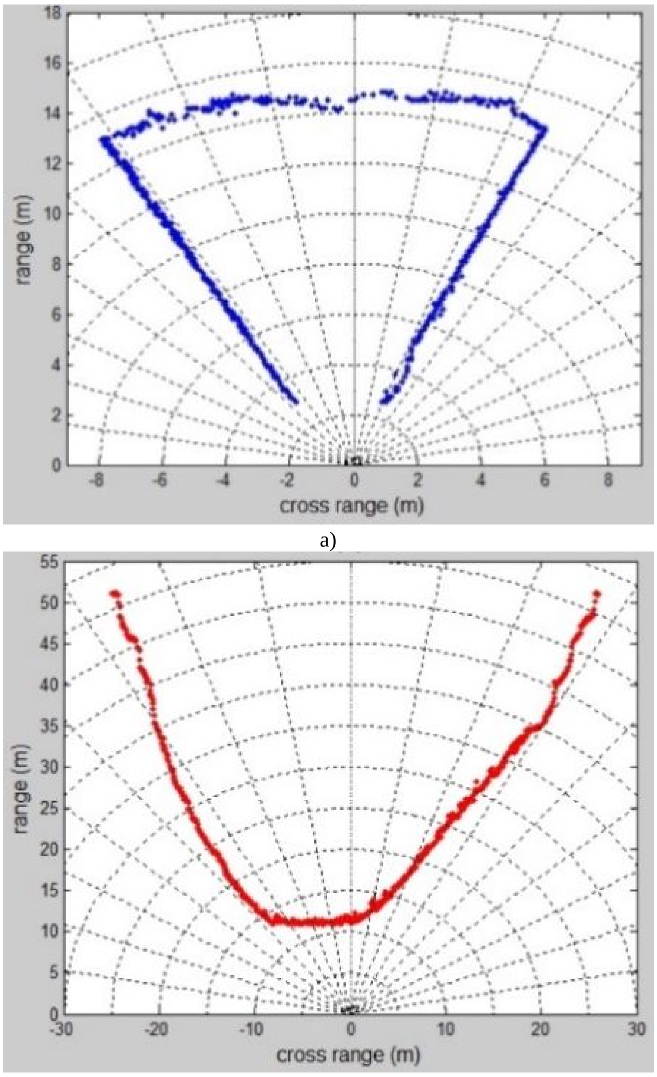


Fig. 9. New radar system integrated with the antennas, TRx module, and signal-processing module.

The signal processing module is integrated with the antenna and the transceiver module, as shown in Fig. 9. The radar system can be covered using a radome. The radar system is installed on an experimental vehicle and field testing is carried out on a real road.

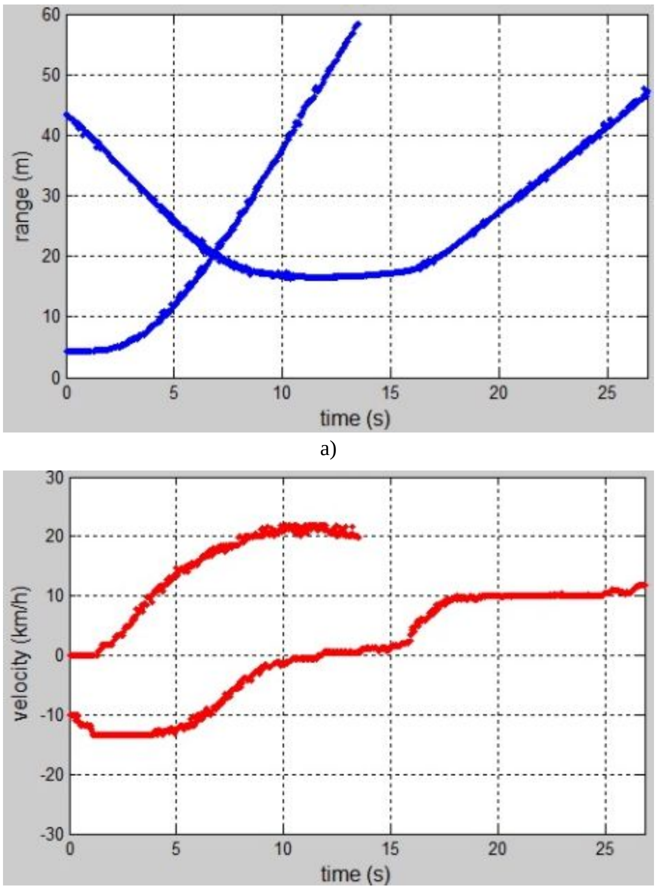
Figure 10 illustrates the measured target positions extracted from the detected distance and angle values of a human (a) and a vehicle (b) using the radar sensor. The x-axis is the cross range and the y-axis is the range. The angle grid is displayed in 10 degree steps. Figure 10(a) presents the profile of a single pedestrian, who is moved along a track of a fan-shape at a speed of approximately 4 km/h. In Figure 10(b), the detected track of a single vehicle moving at a speed of 30 km/h–40 km/h is illustrated. The target vehicle is driven along a U-shaped track beside the road.



a)

b)

Fig. 10. Target profiles for: (a) single pedestrian, and (b) one moving vehicle.



a)

b)

Fig. 11. Detection results of two moving targets over time: (a) the detected range profile, and (b) the detected velocity profile.

Figure 11 presents the multiple target detection results. Two vehicles are driving around in the central area of the field of view. In Fig. 11(a), the x-axis is the time and the y-axis is the range (m). In Fig. 11(b), the y-axis is the detected velocity (km/h).

V. CONCLUSIONS

In this paper, a Virtex-5 FPGA implementation of signal processing was presented for a multi-channel FMCW radar with a ramp-sequence waveform. The signal processing module was designed with a fully parallel architecture in order to support high speed algorithm processing for multiple receiving channels. First, a data routing scheme based DBF architecture was proposed for the pipelined and parallel implementation of the four algorithm groups, which include range processing, digital beam forming, Doppler processing, and detection algorithms. Next, in order to further reduce the processing complexity, a fine-grained DBF structure was also proposed. While the waiting time should be inserted in the data routing scheme based DBF, the fine-grained DBF structure can be fully achieved without extra pause. Moreover, since the fine-grained DBF can be made using pipeline processing together with the range processing, the total time complexity can be reduced. The target-detection ability of the proposed system was confirmed with a field experiment using the newly designed radar system.

REFERENCES

[1] M. A. Richards, *Fundamentals of radar signal processing*. McGraw-Hill, 2005.

[2] S. Miyahara, “New algorithm for multiple object detection in FM-CW radar”, *SAE Technical Paper Series*, 2004.

[3] H. Rohling, M. M. Meinecke, “Waveform design principles for automotive radar systems”, in *CIE Int. Conf.*, 2001, pp. 1–4. [Online]. Available: http://dx.doi.org/10.1109/ICR.2001.984612

[4] V. Winkler, “Range Doppler detection for automotive FMCW radar”, in *Proc. 4th IEEE European Radar Conf.*, 2007, pp. 166–169. [Online]. Available: http://dx.doi.org/10.1109/EURAD.2007.4404963

[5] M. Kronauge, C. Schroeder, H. Rohling, “Radar target detection and Doppler ambiguity resolution”, in *2010 Radar Symposium*, 2010, pp. 1–4.

[6] C. Schroeder, H. Rohling, “X-Band FMCW radar system with variable chirp duration”, *IEEE Radar Conf.*, 2010pp. 1255–1259. [Online]. Available: http://dx.doi.org/10.1109/RADAR.2010.5494425

[7] Y. Wang, Q. Liu, A. E. Fathy, “CW and pulse–Doppler radar processing based on FPGA for human sensing applications”, *IEEE Trans. Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 3097–3107, 2013.

[8] K. Sobaihi, A. Hammoudeh, D. Scammell, “Field-programmable gate array-based software-defined radio for millimetre-wave single-carrier transmission”, *IET Networks*, vol. 1, no. 4, pp. 239–248, 2012. [Online]. Available: http://dx.doi.org/10.1049/iet-net.2012.0037

[9] H. Rohling, “Radar CFAR thresholding in clutter and multiple target situations”, *IEEE Trans. Aerospace and Electronic System*, vol. AES-19, no. 4, pp. 608–621, 1983. [Online]. Available: http://dx.doi.org/10.1109/TAES.1983.309350

[10] Virtex-5 Family Overview (DS100). [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf