Volume 39, Number 4
September 2011

Published by the Association for Computing Machinery
Special Interest Group on Computer Architecture

COMPUTER ARCHITECTURE *NEWS*

SPECIAL ISSUE: HEART2

Systems and Tools I

GPU Applications

Architectures I

FPGA Applications I

Systems and Tools II

TABLE OF CONTENTS (continued on back cover)

acm
Association for
Computing Machinery

Advancing Computing as a Science & Profession

COMPUTER ARCHITECTURE NEWS (ISSN 0163-5964) is published 5 times a year (March, May, June, September, December) by the Association for Computing Machinery, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, Periodicals postage paid at New York, NY 10001 and at additional mailing offices. Send address changes to Computer Architecture News, ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701.

Subscriptions: Annual subscription cost of $19.95 is included in the SIGARCH member dues of $28.00 (for students, cost is included in $14); the non-member annual subscription cost is $54.00. SIGARCH membership information appears on the inside back cover of this issue.

Notice to Contributing Authors to SIG Newsletters: By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish your article in print on condition of acceptance by the editor
- to digitize and post your article in electronic versions of this publication
- to include your article in the ACM Digital Library and in any Digital Library related services
- to allow users to make a personal copy of the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article, and ACM will make every effort to refer requests for republication directly to you.

Submissions: All letters to the editor will be considered "for publication" unless accompanied by a request to the contrary. Except for editorial items, all sources of material appearing in COMPUTER ARCHITECTURE NEWS will be clearly identified. Items attributable to individuals will be interpreted as personal rather than organizational opinions.

Technical papers appearing in COMPUTER ARCHITECTURE NEWS are unrefereed working papers except when indicated to the contrary. Publication in COMPUTER ARCHITECTURE NEWS does not preclude publication of final papers in other ACM publications.

Formatting: All contributions to COMPUTER ARCHITECTURE NEWS should be emailed to the Editor in both Adobe Acrobat PDF format and in the original source format. All fonts must be embedded in the PDF file. Do not use TrueType Fonts. Length should not exceed eight pages. Type should be single-spaced and confined to a 7 1/2" by 9 1/2" area on each page and formatted for Letter-sized paper. Please do not add page numbers, headers, or footers to the manuscript. Two-column formatting is preferred.

Issue closing dates: Articles submitted for publication in the COMPUTER ARCHITECTURE NEWS must be submitted to the Editor by the issue closing date to be considered for publication:

Issue Date	Closing Date
March	January 5
June	April 5
September	July 5
December	October 5

Notice to Past Authors of ACM-Published Articles:
ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that was previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

Message from the Workshop Co-Chairs

It is a pleasure for us to hold the Second International Workshop on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART) in Imperial College London.

Supercomputing using cost-effective accelerators, such as high-end field-programmable gate arrays, general-purpose graphics processing units, the Cell Broadband Engine and other specialized architectures, has become a major theme in high-performance computing. The goal of this workshop is to establish a forum for reporting state-of-the-art research on high-performance computing with reconfigurable architectures and other cost-effective specialized accelerators. The workshop was affiliated to the ACM International Conference on Supercomputing in 2010. This year it becomes an independent workshop with increased number of papers and participants.

The program of the 2nd HEART Workshop includes 10 regular papers, 7 short talks, and 5 posters. They are selected from 31 submissions from 15 countries (Algeria, Brazil, China, Czech, France, Germany, India, Japan, Korea, Netherlands, Singapore, Slovakia, South Africa, Turkey, and UK). The selected presentations span a wide range of topics related to the aim of the workshop, and the two keynotes focus on supercomputing using GPUs and reconfigurable devices.

We are very grateful to the three Program Co-Chairs, Khaled Benkrid of the University of Edinburgh, Martin Herbordt of Boston University, and Yoshiki Yamaguchi of University of Tsukuba, for their hard work. We are also very grateful to the Finance Chair, Kentaro Sano, and to the Publicity Chair, Hironori Nakajo, for their perfect management in such a difficult situation after the severe disaster in Japan. The Publication Chair, Yuichiro Shibata, did a great job on arranging the connection to ACM SIGARCH Computer Architecture News (CANs), in which the proceedings of the HEART Workshop will appear. Furthermore, we are very grateful to Kuen-Hung Tsoi for local arrangements at Imperial College London. Finally, we thank the kind sponsorship of Global COE Program (Information, Electrical Engineering, and Electronics) in Keio University which helps to defray some of the costs of this workshop.

Hideharu Amano and Wayne Luk
The 2nd HEART Workshop Co-Chairs

Embedded Architecture with Hardware Accelerator for Target Recognition in Driver Assistance System

Haisheng Liu
Université de valenciennes,
le Mont-Houy, Valenciennes, France
haisheng.liu@univ-valenciennes.fr

Smail Niar
Université de valenciennes,
le Mont-Houy, Valenciennes, France
smail.niar@univ-valenciennes.fr

Yassin El-Hillali
Université de valenciennes,
le Mont-Houy, Valenciennes, France
yassin.elhillali@univ-valenciennes.fr

Atika Rivenq
Université de valenciennes,
le Mont-Houy, Valenciennes, France
atika.menhaj@univ-valenciennes.fr

ABSTRACT

This paper presents a new Radar-based recognition system, which is able to identify obstacles during a vehicle movement. Obstacles recognition gives the benefits of avoiding false alarms and allows generating alarms that take into account the identification of the obstacle in front of the vehicle. In this paper, we first identify hotspots in the target recognition application. Then, we propose an optimized version of the multiple target recognition algorithm to respect the real time constraints of the application while simplifying the underlying hardware platform. We also propose a flexible embedded architecture with hardware accelerator that supports the proposed algorithm. Using a low cost FPGA-based System-on-Chip, our system is able to detect and recognize more than 10 obstacles of different types in a time limit of 25 mSec.

Keywords

FPGA, Embedded System, Driver Assistance System, System-on-Chip

1. INTRODUCTION

Thousands of people around the world lose their lives in road accidents every year. Analyses have shown that most of the accidents are caused by driver inattention due to physical and mental fatigues. Automatic early warning onboard system has emerged as a solution to improve road safety. Such systems are called Driver Assistance Systems (DAS) in the literature. Adaptive cruise control [8], Radar-aided automatic proximity control and navigation systems are well-known examples of high-technology DAS application.

Historically, Application Specific Integrated Circuits (ASICs) [5] occupied the first place in automotive silicon use because of their cost-effective silicon solution. Increasing levels of complexity and computational demands in automotive applications forced a move to more powerful yet cost-effective processors [9]. As a promising alternative to ASICs, Field Programmable Gate Array (FPGA) solution has been proposed to implement complex automotive subsystems [6].

Recent research activities concentrate on investigating the DAS in complex driving scenarios, such as detecting pedestrians or under changing weather and lighting conditions. However, existing driver assistance systems do not support these complex applications because of their limited functionalities and cost constraints for large-scale automotive use. The programmable and flexible FPGA-based system is considered to offer a tradeoff approach of performance, flexibility and costs. Moreover, it is possible to customize hardware processing units by exploiting the high logic density of the last generations of FPGAs.

At the opposite of the existing DAS systems such as ImapCAR [3] and EyeQ2 [7], our FPGA-based system offers a high level efficiency with low cost. It can also be easily and rapidly adapted to new applications and/or new driving scenarios. In the AutoVision project [1], the authors proposed a dynamically reconfigurable architecture dedicated to video-specific processing. In contrast to the mentioned work above, our system uses an ultra wide band Radar sensor instead of a video camera. This has the advantage of long range and the system performs better in bad visibility conditions.

This paper aims to design an early warning and collision avoidance system. Firstly, the system collects data from a sensor network onboard a vehicle. These data may contain one or more potential obstacles of interest. Then, each obstacle can be identified by comparing the data with a set of predefined obstacle identifiers, called *signatures* in the rest of the paper. In this paper, we focus on the architecture design and optimization issues. The proposed architecture allows the recognition of a big number of obstacle types in a relatively short time. Here, both the words target and obstacle have the same signification.

The recognition feature is able to categorize the obstacles and thereby enhances the system performance, particularly in terms of avoiding false alarms. In fact, the alarms are generated by taking into account the obstacle's information. Let us suppose that a pedestrian and a truck have been localized 5 meters far away from the host vehicle. These two objects do not represent the same danger level and consequently the associated audio alarm should be different. To our knowledge, this feature has been rarely investigated in a Radar-based DAS up to now.

This paper is organized as follows. The principles of Radar-based system are introduced in Section 2. Our configurable architecture is given in Section 3. Finally, the conclusion and perspectives are drawn in Section 4.

2. TARGET DETECTION AND RECOGNITION ALGORITHM

In this section, the UWB (Ultra Wide Band) Radar impulse system is firstly introduced. Secondly, the corresponding detection algorithm is presented. Finally, the target recognition principles are reviewed as well as the associated computational complexity.

2.1 UWB-Radar detection system

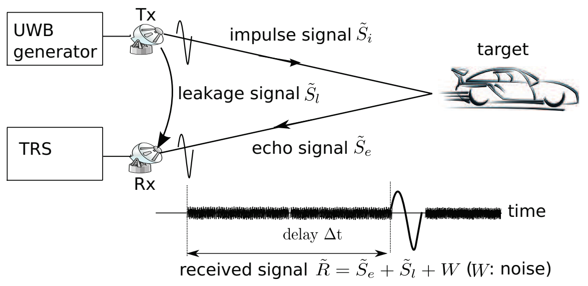
UWB generator
Tx
impulse signal $\tilde{S}_i$
target
leakage signal $\tilde{S}_l$
TRS
echo signal $\tilde{S}_e$
Rx
time
delay Δt
received signal $\tilde{R} = \tilde{S}_e + \tilde{S}_l + W$ (W: noise)

Figure 1: A simple UWB-Radar based system

An overview of a simplified UWB Radar based Target Recognition System (TRS) is given in Figure 1. The UWB generator sends periodically an impulse signal $\tilde{S}_i$ via the antenna Tx. The antenna Rx receives the sum of the echo signal $\tilde{S}_e$, the leakage $\tilde{S}_l$ and the white Gaussian noise W. This leakage corresponds to the direct transmission from the antenna Tx to Rx. The measured distance of a detected target is calculated from the transmission delay Δ_t, which corresponds to the time difference between the emission and reception signals. With regard to the impulse signal, different waveforms are possible: monocycle, Gaussian impulse or Gegenbauer function form [2].

2.2 Target detection algorithm

The general outline of data flow processing is presented in Figure 2. It is beyond the scope of this paper to discuss the impulse sending. The reference signal R' is obtained at reception by replacing the target in Figure 1 with the antenna Rx. The correlation function f_c between the signals $\tilde{R}$ and R' yields a product T_d. Its amplitude corresponds to the correlation degree between the relevant signals.

The potential targets are translated into peaks after correlation operation. Therefore, the next step consists in finding the peaks, i.e., the maximum values, in local time intervals. In case of very close targets, the peaks within a small time interval are combined and considered a single target. Both the leakage elimination and the target detection are realized by the function f_p. As shown in Figure 2, the number of targets n_b and the respective time values t_i are obtained at this stage.

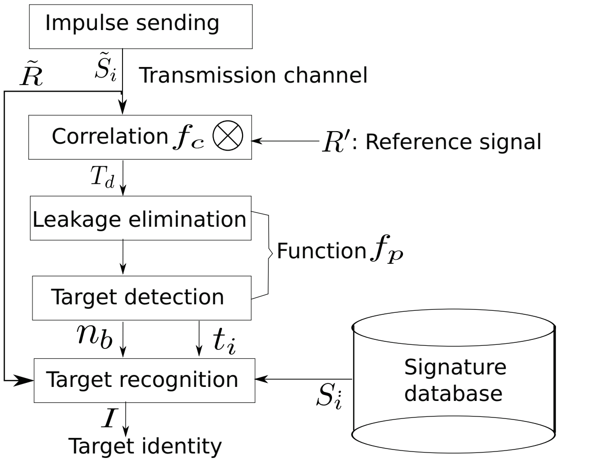

Figure 2: General outline of data flow processing

2.3 Target recognition algorithm

The target recognition is a key feature of the system. Actually, the target's signature is represented by its inherent echo signal form. The recognition algorithm is an iterative process through a pre-built signature database.

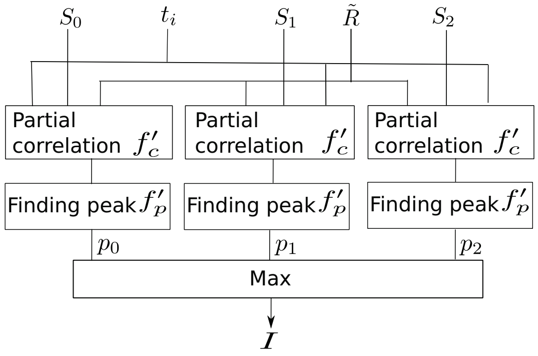

Figure 3: Diagram for the target recognition with 3 signatures

Let us define a database containing L signatures, denoted by the set $S = \{S_0, S_1, \cdots, S_{L-1}\}$. Figure 3 depicts the data processing diagram for a single target with 3 signatures. To recognize a target, the received signal $\tilde{R}$ is successively compared with the known signatures. Similarly, this comparison is realized by the correlation function. However, it is not necessary to run this operation through all the samples. For a time-known target, it is possible to perform a partial correlation, denoted by the function f'_c. Hence, a sample subset delimited by time t_i is thereby selected and the number of operations is significantly reduced. Then, the function f'_p finds the individual peaks p_i (see Figure 3). Finally, the identity I is determined in such a way that the corresponding peak has the highest value, which is in fact produced by the function Max.

2.4 Computational complexity

The computational complexity and memory loads of the proposed system play an important role in system design. These two factors must be reduced to allow a feasible realization. As explained, the complexity of our proposal depends mainly on two key values: the number of targets n_b and the complexity of correlation. Let us now investigate the correlation function. Mathematically, the correlation function $\otimes$ between two functions f and g, which contain respectively M and N elements $(M > N)$, is expressed by the following formula:

$$(f \otimes g)(n) = \sum f(k) \times g(k-n),\ n \in [0, M] \qquad (1)$$

where $f(k) = 0$ for $k \notin [0, M]$ and $g(k) = 0$ for $k \notin [0, N]$. The computational complexity is expressed in number of MAC (Multiplication Addition Cumulation) operations. According to Equation 1, the complexity is equal to the product $(N \times M)$.

For the target recognition stage, a partial convolution is proposed to avoid redundant data processing. Suppose that the signal $\tilde{R}$ is composed of M samples $(r_0, \cdots, r_{M-1})$. If a target is detected at time m, the recognition algorithm is performed over a subset $\{r_{m-L_0}, \cdots, r_{m+N+L_0}\}$ centered around m. Experimentally, the value of L_0 is set around $0.1N$. The complexity of the recognition stage is approximatively equal to N^2.

3. SYSTEM ARCHITECTURE

3.1 Timing profile

The real time constraints imposed by the specific application are the key challenge in designing a microprocessor based system. Thus, the timing profile constitutes an important step to partition tasks between hardware and software components. In our case, there are 2,048 and 128 samples respectively for the digital signals $\tilde{R}$ and R'. With regard to the microprocessor, we use a 32-bit RISC soft core processor Microblaze [10]. Considering a Microblaze processor running at 100 MHz with 8 KB cache and data instruction memory respectively, Table 1 shows the profiling results for a fixed-point representation of the algorithm.

Table 1: timing profile of the relevant functions

Stage	Detection		Recognition	
Function	f_c	f_p	f'_c	f'_p
Time (mSec)	1,450	15	93	0.16

As seen in this table, the correlation function f_c has a significant time-cost. It requires more than 1 second to complete the overall process. Additionally, the function f_p performing the leakage elimination and target detection requires a time-cost of 15 ms. It can be noted that the recognition stage has a lower time-cost. The partial correlation f'_c consumes 93 ms, whereas the finding peaks function f'_p has merely 0.16 ms. This observation shows that the correlation functions (f_c and f'_c) are highly time consuming. The corresponding time value is proportional to the correlation length N. As a result, the implementation of the application on a single Microblaze processor does not respect the real time requirements, 25 ms in our case.

3.2 Configurable architecture

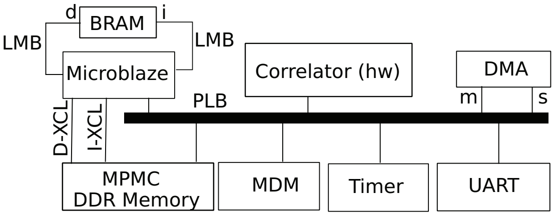

Figure 4: Configurable architecture with hardware accelerator

A configurable architecture composed of hardware and software components is given in Figure 4, in which a hardware accelerator is dedicated to the correlation functions. The key components in this architecture include the Microblaze processor, the hardware correlator and the DMA (Direct Memory Access) controller. The communication between the Microblaze processor and the peripherals is realized through the data transfer bus PLB. The peripherals include the Timer, the component UART and the external DDR memory. The Microblaze processor has data and instruction bus, as respectively marked by **d**-LMB and **i**-LMB in Figure 4.

The hardware correlator handles data processing at a high frequency, up to 370 MHz in our application. However, due to the execution delay of Microblaze processor, the data communication through software interface does not adapt to the high speed data processing, as expected by the hardware processor. The DMA technology, which automates the movement of large amounts of data without processor control, is proposed here to provide a high rate data communications. Also, the DMA controller has two PLB ports: Master and Slave, as marked respectively by **m** and **s** in Figure 4.

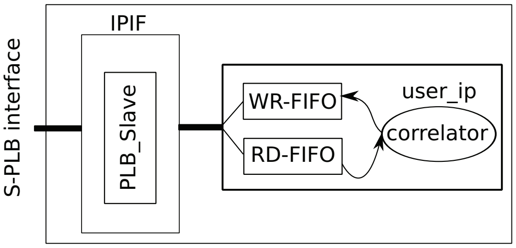

Figure 5: Hardware accelerator architecture

The hardware accelerator is considered as a peripheral device having a slave PLB interface. Figure 5 illustrates the basic blocks of the hardware accelerator architecture. The entity IPIF provides a highly adaptable and quick-to-implement interface between the PLB Bus and the hardware correlator. Depending on our requirements, the read and write FIFO buffers (WR-FIFO and RD-FIFO) are optioned in through the use of VHDL files. The corresponding size is respectively equal to 512 words of 32 bits.

3.3 System synthesis and experimental results

The first experiments with the logic synthesis system are attempts to demonstrate the feasibility of our system. The proposed architecture is implemented on the kit board ML605 [11], in which a Xilinx Virtex-6 FPGA circuit [12] is available. The logical synthesis is performed with Xilinx synthesis tools. The synthesis results are summarized in Table 2, where the component names correspond to the modules mentioned in Figure 4. The line *Others* contains the synthesis results of the following components: the local memory BRAM, the LMB bus, the clock management and the input output IPs. According to Table 2, the system architecture requires in total 14,049 Flip-Flops, 10,688 LUTs and 23 Block RAMs, whereas the hardware correlator needs 996 LUTs and 5,002 Flip-Flops.

Table 2: Logic synthesis results

Component	Flip-Flop	LUT	BRAM
System	14,049	10,688	23
Correlator	5,608	1,872	1
DMA	557	800	0
Timer	362	291	0
MDM	119	120	0
MPMC	5,052	4,201	11
UART	154	135	0
Microblaze	1,955	2,725	6
PLB	169	476	0
Others	73	68	5

Table 3: timing profile of the proposed architecture

Stage	Detection		Recognition	
Function	f_c	f_p	f'_c	f'_p
Time (mSec)	0.53	15	0.17	0.16

Table 3 shows the new timing profile based on the proposed architecture. In this table, the correlation functions f_c and f'_c requires merely 0.53 ms and 0.17 ms respectively for the detection and recognition stages. It is important to note that the functions f_p and f'_p are executed by the Microblaze processor. Then, their execution times are not different from those shown in Table 1. As a result, the total execution time is reduced to 15.53 ms for the detection stage. On the other hand, the recognition stage requires 0.33 ms for each target and each signature. Thanks to the hardware accelerator, the proposed FPGA-based architecture meets the real time requirements. As an example, considering a database including 3 signatures, the number of targets can attain up to 10 targets per Radar scan (25 mSec).

4. CONCLUSION

In the automotive industry, the Radar-based intelligent system may offer interesting detection capacity especially in bad driving conditions. In this paper, a target recognition application using correlation-based algorithm has been investigated and an embedded architecture with hardware accelerator is proposed. As future investigations, we propose to study the integration of our target recognition system with the multiple target tracking system proposed by [4]. Cooperation between these 2 approaches in a DAS allows an accurate identification of a large number of obstacles with limited hardware resources.

5. ACKNOWLEDGEMENT

The authors would like to express their gratitude to the French Research Agency and the International Campus on Safety and Intermodality in Transportation for the financial support through the research project Prima-Care.

6. REFERENCES

[1] Claus, C., Stechele, W., and Herkersdorf, A. Autovision – a run-time reconfigurable mpsoc architecture for future driver assistance systems. *Information Technology 49*, 3 (2007), 181 – 187.

[2] Elbahhar, F., Rivenq-Menhaj, A., and Rouvaen, J. M. Multi-user ultra wide band communication system based on modified gegenbauer and hermite functions. *Wireless Personal Communications 34*, 3 (August 2005), 255–277.

[3] Hiroto, H. Automotive image recognition processor imapcar. *NEC technical journal ISSN 1880-5884 1*, 5 (Dec. 2006), 24–28.

[4] Khan, J., Niar, S., Menhaj1, A., Elhillali1, Y., and Dekeyser, J. L. A mpsoc architecture for the multiple target tracking application in driver assistance system. *International Conference on Application-Specific Systems, Architectures and Processors* (2-4 July 2008), 126 – 131.

[5] Masaki, I. Asic approaches for vision-based vehicle guidance. *IEICE Transactions on Electronics E76-C*, 12 (Dec. 1993), 1735–1743.

[6] Saad, J., Baghdadi, A., and Bodereau, F. Fpga-based radar signal processing for automotive driver assistance system. *IEEE/IFIP International Symposium on Rapid System Prototyping* (23-26 June 2009), 196 – 199.

[7] Stein, G. P., Rushinek, E., Hayun, G., and Shashua, A. A computer vision system on a chip: a case study from the automotive domain. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (20-26 June 2005), 130 – 130.

[8] Vahidi, A., and Eskandarian, A. Research advances in intelligent collision avoidance and adaptive cruise control. In *IEEE Transactions on Intelligent Transportation Systems* (2003), vol. 4, pp. 143–153.

[9] Valera, M., and Velastin, S. A. Intelligent distributed surveillance systems: A review. In *IEEE Proceedings - Vision, Image and Signal Processing* (8 April 2005), vol. 152, pp. 192 – 204.

[10] Xilinx. Xilinx microblaze processor reference guide, ug081 (v11.0). *Embedded Development Kit EDK 12.1* (Sep. 14 2000).

[11] Xilinx. Xilinx ug534 ml605 hardware user guide ug534 (v1.4).

[12] Xilinx. Xilinx virtex-6 fpga configuration user guide ug360 (v3.2).