An Embedded High Performance Data Acquisition and Pre-Processing Interface for Asynchronous Event-Based Silicon Retina Data

Christoph Sulzbachner, Jürgen Kogler, Wilfried Kubinger
AIT Austrian Institute of Technology
Safety & Security Department
Safe and Autonomous Systems
Donau-City-Strasse 1, 1220 Vienna, Austria
Email: {christoph.sulzbachner, juergen.kogler, wilfried.kubinger}@ait.ac.at

***Abstract*—In this paper we present an embedded high performance Serial RapidIO™ data acquisition interface for Silicon Retina technology based computer vision applications. The Silicon Retina technology is a new kind of bio-inspired analogue sensor that provides only event-triggered information depending on variations of intensity in a scene. Unaltered parts of a scene without intensity variations need neither be transmitted nor processed. Due to the asynchronous behavior and the varying data-rates up to a peak of 6M events per second (Meps) per channel and a time resolution of 10ns of the imager, a distributed digital signal processing system using both a single-core and a multi-core fixed-point digital signal processor (DSP) is used. The single-core DSP is used for data pre-processing of the compressed data streams and forwarding it to the multi-core DSP, which processes the actual data. Pre-processing also includes disposing the data required for processing on the multi-core system using a data parallelism concept. We discuss both design considerations, and implementation details of the interface and the pre-processing algorithm.**

I. Introduction

Many sensor technologies are already in use for Reliable Advanced Driver Assistance Systems (ADAS). These systems should generally improve the driving process of a vehicle. A significant part of ADAS is based on vision systems e.g. road sign recognition. The EU-funded project ADOSE[1] (reliable *a*pplication-specific *d*etection of road users with vehicle *o*nboard *se*nsors) is focused on cost-effective sensor technologies for ADAS. Our aim at ADOSE is to develop an embedded Silicon Retina stereo system used for pre-crash warning/preparation side impact detection. This system uses a bio-inspired analogue optical sensor and thus, a novel approach for data acquisition and processing has to be realized.

These types of sensors detect intensity changes in an observed scene at pixel level, with each pixel delivering its address, and event data separately and independently, whenever it detects an intensity change. Unaltered parts of a scene that have no intensity variation need neither be transmitted nor processed. Thus, the amount of data varies over time depending on the scene. Due to this asynchonous behavior peak data-rates up to 6M events per second (Meps) per channel and a time resolution of 10ns are achieved.

The Silicon Retina stereo system consists of two bio-inspired optical sensors, an embedded system for data acquisition and pre-processing based on a TMS320C6455 single-core fixed-point digital signal processor (DSP), and an embedded system for processing the stereo matching algorithm based on a TMS320C6474 multi-core fixed-point DSP platform with three cores.

Data acquisition techniques for conventional cameras are prevalent and interfaces are available. Due to the special characteristics of this sensor, approaches for embedded real-time data acquisition do not exist. In this paper we give an overview of the optical sensor and the algorithm approach for data processing. The sensor principle used is quite unconventional. This requires new algorithmic approaches for solving the stereo correspondence problem, which has big influence on the data-handling and pre-processing. Pre-processing in this context means preparing the acquired data to afford parallel data processing on the multi-core system.

The remainder of this paper is outlined as follows: Section II gives an overview of related address event representation communication, debugging and acquisition interfaces and methodologies. Section III introduces the optical sensor and describes its core features. Section IV gives an overview of the embedded system and the DSPs used for acquisition, pre-processing and stereo matching. Section V shows the concept of the hardware and details of the data acquisition and the pre-processing. The next Section covers the communication from the acquisition and pre-processing embedded system to the stereo matching embedded system. Section VII shows a high performance approach for data acquisition of the processing embedded system using Serial RapidIO™. Finally, we give a conclusion about the work.

II. Related Work

Conventional optical sensors capture image data frame-by-frame at a fixed frame-rate. A series of frames contains a vast

[1] http://www.adose-eu-org

amount of redundant data. The Silicon Retina uses an event-triggered concept, where an Event $\mathbf{E}$ is a three-tuple consisting of the coordinates $\mathcal{X}$ and $\mathcal{Y}$, and a timestamp $\mathcal{T}$.

Berner et. al. [1] describe in their paper a high-speed USB 2.0 address event representation interface that allows simultaneously monitoring and sequencing data. Their claim was to develop a simple, cheap and user-friendly device where the timestamps are generated with a 16-bit counter in a CPLD. The peak event-rate for monitoring is 6 million events per second (Meps) and for sequencing 3.75 Meps with four bytes per event including a separate timestamp for each event. Merolla et. al. [2] give an overview of using USB as a high-performance interface for neuromorphic systems and they cover both hardware and software considerations.

Dante et. al. [3] show in their work a device for interfacing address-event based neuromorphic systems that is implemented on a peripheral component interface (PCI) board. This system can handle four transmitters and four receivers and the three main functions are: monitoring, sequencing, and mapping. Monitoring contains reading and time-stamping events to make them available for further processing. Sequencing stands for generating traffic, and mapping transforms incoming events into outgoing events in different modes. The paper gives no detailed information on performance.

Litzenberger et. al. [4] present an embedded smart camera for high speed vision. They connect the optical sensor via first in first out (FIFO) buffer memory to a Blackfin fixed-point DSP from Analog Devices. Following a data available request from the optical sensor, the data is stored in the FIFO. The DSP samples the data from the FIFO while it is not empty and timestamps the events afterwards in the DSP. Here, significant performance is used for data acquisition that would be required for data processing. Due to the latency of the generation and the input to the DSP, the timestamps of the events are distorted.

III. Optical Sensor

The discussed type of sensor technologically goes back to Fukushima et. al. [5] in 1970, who implemented an electronic model of a retina. The first retina imager on silicon basis was developed by Mead and Mahowald [6]. Within ADOSE, we used two different versions of the optical sensor. The prior version has a resolution of 128×128 pixels and a time resolution of 1ms. Due to processor and hardware restrictions, the optical sensor is able to deliver up to sustained 300keps. This version uses an UDP socket for communication [7].

The new version of the optical sensor is a 304×240 pixels vision sensor with a time resolution of 10ns. These types of sensors exploit a very efficient asynchronous event-driven data protocol that only delivers variations of intensity. In this way data redundancy is mostly reduced. Unaltered parts of a scene that have no intensity variation need neither be transmitted nor processed. Figure 1 shows a visualized image with a pedestrian, where the events of 20ms are accumulated. The optical sensor has a 20-bit parallel asynchronous AER interface with hardware-accelerated pre-processing of the events, including 10ns time-stamping, and a region-of-interest (ROI) filter. The overall performance of the AER interface is sustained 5.125Meps [8].

To predict the occurring data-rate of the new optical sensor both theoretical and empirical data-rate estimations were done. Analysis of the optical sensor with a test pattern generator showed that the average stimuli frequency of a pixel is approximately 30Hz. Hence, the average data-rate of the new optical sensor is about 2.07Meps or 8.29MiB/s, and the maximum data-rate is about 8.29Meps of 33.18MiB/s. In the empirical estimation the optical sensor was stimulated by several traffic conditions in a real-world environment resulting in an average data-rate of 5.3Meps or 21.2MiB/s.

Due to the high data-rate requirements of the system consisting of two optical sensors, an interface is required that produces a minimum protocol overhead.

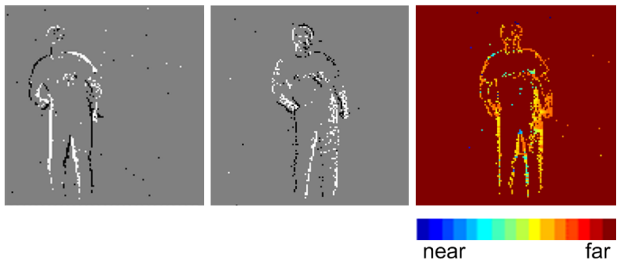

Fig. 1. Visualized image with pedestrian captured with the Silicon Retina; left: left imager; middle: right imager; right: depth image

A. Time-stamping and Synchronization

At the 128×128 pixels sensor time-stamping of the events is processed in an interrupt service routine on a DSP. Due to restrictions for predicting the latency of the events from the appearance to the software routine, $\mathbf{E}(\mathcal{T})$ is uncorrectable distorted. Furthermore, each event was attributed with an own timestamp resulting in high data redundancy [4].

The 304×240 pixels optical sensor with a time resolution of 10ns uses a parallel interface similar to an asynchronous memory interface and a compressed event protocol. This protocol separates between time and event data. Events with the same timestamp only have to transmit the timestamp once at the beginning. Time-stamping is now done in hardware resulting in much lower latencies [8].

A significant part of a stereo vision system is synchronization, that means that both sensor units have a common understanding of time. Synchronization of the 128×128 pixels optical sensor version was implemented in software by a master-slave concept using a serial interface for communicating the timestamp. The 304×240 pixels optical sensor is synchronized completetly in hardware using a common clock and synchronization signals.

B. Stereo Matching

Stereo Matching deals with the reconstruction of depth information of a scene captured from two different points of view. Scharstein and Szeliski [9] give a good overview of conventional stereo matching. Evaluations showed that area-based

techniques do not exploit the features of the Silicon Retina technology. New approaches are based on event-triggered stereo matching.

IV. Embedded System

The embedded system used to perform data acquisition and pre-processing is based on a TMS320C6455 single-core fixed-point DSP from Texas Instruments. Due to the high performance requirements of the stereo vision algorithms, a second DSP, which is a TMS320C6474, is dedicated for data processing. Both DSP models are based on the C64x+ DSP core from Texas Instruments. The C64x+ DSP core has a very long instruction word (VLIW) architecture with eight functional units. The TMS320C64x+ DSP Megamodule Reference Guide [10] gives more detailed information about this architecture.

Both the TMS320C6455 and the TMS320C6474 have many similarities, but in a more detailed view there are significant differences. The most significant difference between both DSPs is that the TMS320C6474 consists of three C64x+ DSP cores rather than one. This has a noticeable effect on the Peak million multiply accumulate cycles per second (MMACS) of each DSP. Another difference in terms of data acquisition is that the TMS320C6474 has no adequate parallel interface for connecting the parallel interface of the optical sensor.

Figure 2 depicts the schematics of the embedded system, which comprises two optical sensors that are connected to the adapter-board to buffer the data traffic. The adapter-board is connected to the external memory interface (EMIF) of the TMS320C6455 DSP starter kit (DSK) for data acquisition. After acquisition and pre-processing, the data is output via Serial RapidIO™ to the TMS320C6474 evaluation module (EVM), where it is further processed by the stereo algorithm.

A. Operating System

The implementation of both the acquisition and pre-processing embedded system, and the stereo matching embedded system uses DSP/BIOS, a pre-emptive scalable real-time kernel from Texas Instruments. There are various supported features supporting e.g. hardware abstraction or real-time analysis. Hardware Interrupts (HWI) are used to handle time critical asynchronous events. Software Interrupts (SWI) have a lower priority than HWIs, but a higher priority than conventional tasks and are software triggered. Tasks are threads that have lower priority than HWIs and SWIs, but higher priority than the idle task. The DSP/BIOS Manual [11] gives a detailed guide to the features of the kernel.

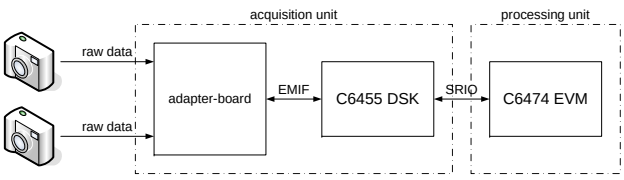

Fig. 2. Schematics of embedded system, consisting of the adapter-board and the TMS320C6455 DSK for data acquisition, and the TMS320C6474 EVM for data processing.

V. Data Acquisition and Pre-processing on the TMS320C6455

Various ways acquiring data from external devices exist. Using conventional imagers, images are captured at a constant frame-rate. This allows precisely calculating the amount of data and optimizing a technique therefore. Due to the variable amount of asynchronous data, different acquisition techniques were combined.

A. Overview of Hardware

The interconnection of the parallel interface of the optical sensor to the data acquisition interface of the DSP is realized with an adapter-board consisting of asynchronous FIFO buffers in the middle. The EMIF allows bi-directional interfacing of various amounts of FIFOs, both synchronous and asynchronous types with a minimum of glue logic. Setup, strobe and hold-times can be fully configured by the EMIF registers. Castille [12] gives a good overview of FIFO characteristics and types, and asynchronous interface signals of the TMS320C6000 family. Generally, arbiters are required to avoid collisions on a bus used by multiple bus members. The EMIF features four independent memory spaces that are controlled automatically by the EMIF unit [13]. Thus, logic is only required for resetting the FIFOs.

Figure 3 shows a schematic of interfacing the FIFOs by the EMIF and general purpose input output (GPIO) of the data acquisition DSP. The chip enable (CE) signals of the EMIF are used for setting the memory space. Both FIFOs are mapped to a different memory space in such a way that the internal arbiter of the EMIF can be used and no separate logic is required. The FIFO features an empty flag (EF), a half flag (HF), and a full flag (FF), which are directly connected to interruptible inputs I from the GPIO. Output O of the GPIO is used for setting the acquisition mode.

Depending on the empty, half, and full flag of the FIFOs the average data-rate on the memory interface can be calculated by incrementing the data amount of the corresponding flag per time unit. As long as this data-rate is below a defined threshold, a polling mode is used, otherwise data is transferred in burst mode.

B. Burst Mode

The EDMA3 controller consists of an EDMA3 channel controller (EDMA3CC) and an EDMA3 transfer controller (EDMA3TC). The EDMA3CC features a fully orthogonal transfer description, a flexible transfer definition, an interrupt generation for the transfer and error code, a debug visibility, 64 independent channels, four quick DMA (QDMA) channels, four programmable transfer queues, 16 event entries per queue, and a memory protection support. The EDMA3TC features four transfer controllers (TC), 128-bit wide read and write ports per TC, up to four in-flight transfer requests, programmable priority levels, support for two-dimensional

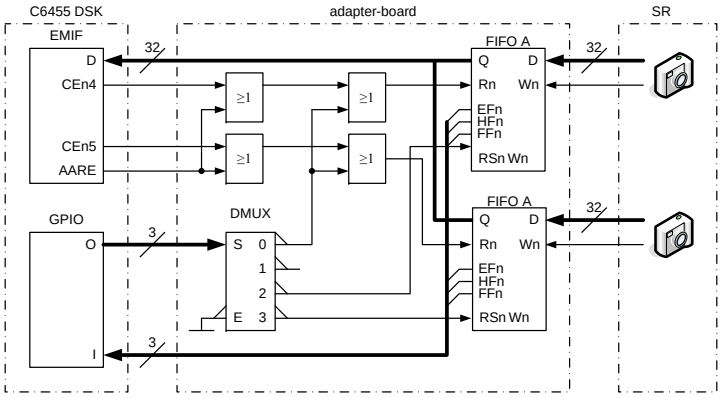

Fig. 3. Schematic of data acquisition, consisting of signals the EMIF interface for signalizing, and signals from the GPIO for mode selection and interrupt generation.

transfers, support for incremental or constant addressing mode transfers, interrupt and error support, little-endian and big-endian support, and memory mapped registers [14]. According the processor errata sheet, the FIFO addressing mode should not be used when reading from EMIF. There are two approaches presented to avoid this problem in [15].

The EDMA3 controller is based on a memory based architecture using a parameter RAM (PaRAM), which contains a whole transfer context. A special feature of this controller is linking and chaining. Linking is a mechanism for automatically reloading a new PaRAM context for a transfer when it is finished. Chaining is a mechanism that automatically triggers a new transfer after the current transfer has finished. With linking it is possible to build a multi-buffered input mechanism that uses a different memory destination for every transfer. This feature is exhaustively used to implement a double buffered input buffer for each optical sensor. The transfers are triggered by configured interrupt lines on the GPIO interface. Using this approach, no processing power of the DSP is required as soon as the data is fully available. In this context, chaining could be used for modification of the data format. Implementing a double buffered input requires three PaRAMs. The first PaRAM is registered for the transfer; the other PaRAMs are required for reloading and are linked to each other. For the initial startup, one PaRAM set has to be loaded twice and after the first transfer it will be overwritten by the other PaRAM context and so on. The PaRAM context also features a transfer complete code (TCC), which triggers an interrupt when an EDMA3 transfer has finished. All PaRAM sets have other TCCs to afford processing the correct memory.

When the TCC interrupt is triggered, a registered HWI routine is called and the TCC can be requested by software. To keep the runtime of the HWI as small as possible, only a SWI is posted based on the TCC. In the SWI, the posted value can be requested and the TCC can be inquired. This two level approach allows processing the correct memory in a SWI context with a minimum latency. The SWI routine pre-processed the acquired data for the multi-core system.

C. Pre-processing to Afford Parallelism

In this project, the stereo algorithm is processed on a multi-core DSP. Therefore the acquired data has to be pre-processed to afford parallel processing.

Culler and Singh [16] gives a detailed overview of parallel computer architectures. Regarding him, there are several types of parallelism:

- Parallelizing loops often lead to similar operation sequences or functions being processed on large amounts of data on parallel processing units. Splitting the amount of data to multiple units, each performing a lower amount of data is called data level parallelism.
- In addition to data level parallelism, performance intensive functions can exhibit function parallelism. Different processing sequences of the function or stages of a pipeline can be performed either concurrently on the same or different data or several of the independent sequences can be performed in parallel.

This architecture in this project mainly follows the data level parallelism approach, where the amount of data is scaled down to smaller amounts per processing unit, because an atomic operation of the stereo algorithm is a correlation per line. Advantages of this concept are a predictable processing load of the parallel units, scalability of the processing system, and a more simple development process of the code. In the case of Silicon Retina applications, predictable processing load assumes uniformly distributed data. Equation 1 shows the identification of the destination core n depending on $\mathbf{E}(\mathcal{Y})$, the number of parallel units N, and the height of the optical sensor H.

$$n = \frac{\mathbf{E}(\mathcal{Y})N}{H} \tag{1}$$

Figure 4 shows an example for partitioning events from two sources to three destinations, thus $N = 3$. An event $e_{n,m}$ is defined by $\mathbf{E}(\mathcal{X}, \mathcal{Y}, \mathcal{T})$ and a timestamp $t_{n,m}$ is described by $\mathbb{N}^+$, where n is the source identifier and m is a continuing unique number. Regarding the compressed data-format of the

source data stream, $T'_{S,n} = e_{n,m}(\mathcal{T})$, where $T'_{S,n}$ is the current timestamp of source n.

Due to n, the event is routed to the destination, whenever the destination time $T'_{D,n} = T'_{S,n}$, otherwise the current source timestamp is set before. Using this approach, the amount of data per processing unit can be minimized, because every destination only receives the according data that can directly be processed.

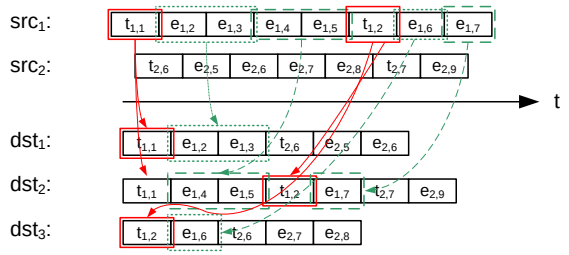

Fig. 4. Example of the Split Algorithm; The events e_{12}, e_{13}, e_{25}, and e_{26} are in the first third of the scene, e_{14}, e_{15}, e_{17}, and e_{29} are in the second third of the scene, and the remaining events are in the third third of the scene

D. Auxiliary Further Pre-processing

Depending on the stereo algorithm of the embedded system, further pre-processing is possible before outputting the data. In the case of a time-space correlating stereo algorithm, image rectification is important, and noise filtering is an optional task.

Image rectification is used to transfer the distorted coordinate system into a standard coordinate system with aligned epipolar lines. Rectification needs to be performed before separating the amount of data to the destination cores, otherwise during performing the stereo algorithm, one destination may require data from another core.

Noise filtering or noise reduction is an optional process that pre-processes the data from the optical sensors and removes events that are disqualified. This simply reduces the amount of data.

VI. Data Output on the TMS320C6455

After separating the amount of data dedicated to the destination, it needs to be transported from the TMS320C6455 to the TMS320C6474. According to the available interfaces of both DSPs, Ethernet and Serial RapidIO™ are possible interfaces.

Ethernet is the standard for wide scale interconnects networks and intended for box-to-box, board-to-board, chip-to-chip, and backplane interconnection. It has the ability to connect a multiple members, and has a very flexible and extensible architecture. Advantages of Ethernet to this project are variable packet sizes up to 9000 byte jumbo frames. Disadvantages are a high traffic overhead, a low Ethernet PDU size up to 1500 byte, a symbol-rate per pair of a 1000Base-T Gigabit of 128Mbaud [17], and a processing intensive stack. Both, the TMS320C6455 and the TMS320C6474 support a 10/100/1000Mbps operation mode. The EMAC peripheral module acts as a DMA master, which is directly connected to the switched central source (SCR) of the C64+ core. It either can access the internal or the external memory space.

Serial RapidIO™ is an electronic communication standard, which affords a reliable, high-performance packet switched interconnection technology for chip-to-chip and board-to-board. The architectural hierarchy consists of three layers. The logical layer specifies the protocols, the transport layer defines addressing schemes, and the physical layer contains the device interface. It is optimized for embedded systems usage and the advantages are a low overhead, variable packet sizes and data transfer sizes up to 4kiB. The symbol-rate per pair of a LP-Serial RapidIO™ is up to 3.125Gbaud [17], [18]. The Serial RapidIO™ peripheral device has its own DMA engine that is also directly connected to the SCR of the C64+ core. Thus, it can initiate memory transfers and has access to all memory locations.

The destination buffers shown in the previous section are managed by double buffered memories. One buffer can be accessed by the processing task, and the other buffer can be accessed by the Serial RapidIO™ peripheral for data transmission purposes. The Serial RapidIO™ peripheral consists of four load store units (LSU) for transmission of direct I/O packages and maintenance packages. The memory access unit (MAU) controls the perception of direct I/O packages. Using the four LSUs alternatively, resource locks can be avoided. Once a buffer is full, the transmission is initiated and automatically switched to the other memory bank. Depending on the output buffer and the destination, a doorbell is transferred afterwards. A doorbell is a short message intended to trigger and interrupt at the destination system. This interrupt is handled by a HWI.

VII. Data Acquisition on the TMS320C6474

The destination multi-core embedded system based on a TMS320C6474 consists of three cores. The TMS320C6455 is connected to the TMS320C6474 over Serial RapidIO™ channel 0. Using direct I/O transfers the source has to know the destination's memory mapping. Otherwise unpredictable conditions could occur, when the source overwrites memory during runtime. To be aware of these conditions, both a static and a dynamic approach exist. Using the static approach, the source needs to know the memory addresses and sizes of the destination. The disadvantage is that a complete system rebuild is required, when modifying the memory mapping. The dynamic approach exchanges the memory addresses and sizes during runtime in an initialization phase before exchanging data.

In ADOSE, the static approach is used. The destination application consists of a memory segment especially for Serial RapidIO™ data. This segment includes six memory spaces (double buffers for three cores). The source has to know the base address and the size of a buffer. This way, all buffer addresses can be calculated for a known buffer localization.

To avoid resource conflicts, the Serial RapidIO™ peripheral is initialized and processed only by one core. Initialization includes setting up, starting and registering required interrupts

required by the peripheral. Processing means executing these interrupt routines that include a LSU and Doorbell Handler, and reconnect when link errors occur.

A. Inter-core Communication

The multi-core architecture of the TMS320C6474 allows defining a full custom inter-core communication model. Global and local mapped memory are methods that allow both each core processing its own memory, and each core accessing the memory segments of the others [19]. The hardware semaphore module allows resource sharing of any resources, because they have no defined assignment. For an application it is essential that all cores have a common understanding of the semaphores [20]. The inter-processor communication (IPC) peripheral module provides inter-core interrupts for exchanging events [21], [22].

The core that initializes the Serial RapidIO™ peripheral is also dedicated for all doorbells. Affording forwarding of the doorbells to other cores, a routing table mapping doorbells to IPCs was realized. All cores have to register the doorbell code, they are interested in to afford forwarding. Whenever a doorbell is received by the doorbell interrupt of the primary core, the look-up-table is traversed to identify the cores that have registered the received doorbell code. Hence, for all these cores, the primary core triggers an IPC interrupt. The forwarding is implemented in the HWI context of the doorbell HWI. The HWI posts a semaphore that initiates the final processing of the stereo algorithm in a task context.

Within the physical layer the data is 8b/10b coded and the actual configuration of the interface using a streaming write operation with 32-bit addressed and 8-bit IDs, the theoretical maximal data-rate at a signaling-rate of 3.125GHz is 289.86MiB.

Every core processes different data, but each core is responsible for separate parts of the result. The result buffer is located in a global memory segment, where all cores have access to. Thus, no resource conflicts can occur and the cores do not have to communicate with each other.

VIII. Results and Conclusion

This paper presented a Serial RapidIO™ data acquisition interface for Silicon Retina based computer vision applications optimized for high performance. The performance is achieved by exploiting the processor peripherals at a minimum processor workload. The embedded system used was a distributed digital signal processing system using both a single-core and a multi-core DSP. Also, the pre-processing required for parallel data processing on a multi-core system using a data parallelism concept was discussed. Other kinds of schedulers to balance the work were not implemented. This approach gives a good balance between the cores with a very little overhead.

We showed that a total sustained data-rate of 10Meps for data acquisition, pre-processing and forwarding can be achieved. For stereo, this results in 5Meps per channel. This system has a performance increase of 1.67 compared to an available USB 2.0 interface. Further improvement can be achieved by using synchronous FIFOs and applying the burst mode also for lower data-rates by using variable FIFO interrupts with configurable data amounts.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement n° ICT-216049 (ADOSE).

References

[1] R. Berner and T. Delbruck and A. Civit-Balcells and A. Linares-Barranco, "A 5Meps $100 USB2.0 Address-Event-Monitor-Sequencer Interface," *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2007.

[2] P. Merolla and J. Arthur and J. Wittig, "The USB Revolution," *The Neuromorphic Engineer*, 2005.

[3] V. Dante and P. D. Giudice and A. M. Whatley, "PCI-AER – Hardware and software for interfacing to address-event based neuromorphic systems," *The Neuromorphic Engineer*, 2005.

[4] M. Litzenberger and A. N. Belbachir and P. Schön and C. Posch, "Embedded Smart Camera for High Speed Vision," *IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2007.

[5] K. Fukushima and Y. Yamaguchi and M. Yasuda and S. Nagata, "An Electronic Model of Retina," *Proceesings of IEEE*, vol. 48, no. 12, 1970.

[6] C. Mead and M. Mahowald, "A silicon model of early visual processing," *Neural Networks Journal*, vol. 1, 1988.

[7] P. Lichtsteiner and C. Posch and T. Delbruck, "A 128×128 120 dB 15 μs Latency Asynchronous Temporal Constrast Vision Sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, 2008.

[8] M. Hofstätter and P. Schön and C. Posch, "An integrated 20-bit 33/5M events/s AER sensor interface with 10n time-stamping and hardware-accelerated event pre-processing," *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2009.

[9] Daniel Scharstein and Richard Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2001.

[10] Texas Instruments, "TMS320C64x+ DSP Megamodule Reference Guide," Aug. 2008, Literature number: spru871j. [Online]. Available: http://www.ti.com

[11] ——, "TMS320C6000 DSP/BIOS 5.32 Application Programming Interface (API) Reference Guide," Dec. 2009, Literature number: spru403. [Online]. Available: http://www.ti.com

[12] ——, "TMS320C6000 EMIF to External FIFO Interface," Mar. 1999, Literature number: spra543. [Online]. Available: http://www.ti.com

[13] ——, "TMS320C645x DSP External Memory Interface (EMIF)," Mar. 2008, Literature number: spru971c. [Online]. Available: http://www.ti.com

[14] ——, "TMS320C645x DSP Enhanced DMA (EDMA3) Controller User's Guide," Jan. 2007, Literature number: spru966b. [Online]. Available: http://www.ti.com

[15] ——, "TMS320C6354/55 Digital Signal Processor Silicon Revisions 2.1, 2.0, 1.1 – Silicon Errata," Mar. 2009, Literature number: sprz234k. [Online]. Available: http://www.ti.com

[16] D. E. Culler and J. P. Singh, *Parallel Computer Architecture – A Hardware/Software Approach*. Morgan Kaufmann, 1999.

[17] G. Shippen, "Comparing Ethernet and RapidIO," *IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 2007. [Online]. Available: http://www.rapidio.org/education/technology_comparisons/Comparing_Ethernet_and_RapidIO_18_Jan_2007.pdf

[18] RapidIO Trade Association, "RapidIO Interconnect Specification," 2002.

[19] Texas Instruments, "TMS320C6474 Multicore Digital Signal Processor," Nov. 2009, Literature number: sprs552d. [Online]. Available: http://www.ti.com

[20] ——, "TMS320C6474 DSP Demaphore User's Guide," Oct. 2008, Literature number: sprug14. [Online]. Available: http://www.ti.com

[21] ——, "Inter-Core Communication on TMS320C6474," Jan. 2009, Literature number: sprab20. [Online]. Available: http://www.ti.com

[22] ——, "How to Approach Inter-Core Communication on TMS320C6474," Jan. 2009, Literature number: sprab25. [Online]. Available: http://www.ti.com