Multiple Input Digital Arbiter with Timestamp Assignment for Asynchronous Sensor Arrays

M. Hofstätter, A. N. Belbachir, *Member*, *IEEE*, E. Bodenstorfer and P. Schön
Information Technologies
ARC Seibersdorf research GmbH
Tech Gate Vienna, Donau-City-Straße 1, 1220 Vienna, Austria
{michael.hofstaetter, ernst.bodenstorfer, ahmed.belbachir, peter.schoen}@arcs.ac.at

Abstract—**This paper introduces a novel concept for arbitrating the access of multiple asynchronous data sources to a shared communication bus, while adding a timestamp, which represents high precision temporal information, to sensor information. This principle is based upon an arbiter serving uncorrelated inputs and generating a stream of data packets. The information contained in the data packets consists of the address information identifying the data source, data from the source and a timestamp value with respect to the occurrence of the bus request (event) generated by the corresponding data source. To enhance the adaptability to particular applications, the time resolution can be varied. The proposed concept has the advantage of delivering a sorted output data stream of nearly concurrent events (labeled with the same timestamp) which is very advantageous for consecutive data processing. Furthermore, this arbitration method is very efficient as it enables the utilization of the maximum output transfer rate for a given clock frequency. A potential usage in asynchronous vision chips is intended. This concept is demonstrated using an asynchronous vision chip containing 512 autonomous optical sensor elements.**

I. INTRODUCTION

This work is concerned with the development of a new concept for a fast and accurate digital circuitry that may support the handling of multiple asynchronous data sources. The intended application concerns asynchronous vision chips for vision systems. A typical vision system [1] contains two parts: an imager (synchronous or asynchronous) and a processing subsystem. The asynchronous imager consists of a set of sensors elements supported by an analog and/or digital circuitry [2] called “arbiter”. The processing subsystem typically includes a Digital Signal Processor (DSP) for the analysis and the application-specific usage of the sensor data.

Most vision applications require accurate real time processing capability. Fault inspection [3] and scene surveillance/analysis [4] are examples where those systems exploit the changes in the visual scene to analyze and follow-up moving objects [5]. They are usually faced with fast moving objects in a scene, such that traditional arbiter structures [2] are not adequate to maintain the accurate handling without losing information, especially if those systems are equipped with a large number of autonomous optical sensor elements (typically > 500) and the temporal information is important.

This paper presents a dedicated concept of a digital arbiter able to accurately handle multiple asynchronous sensor elements and adding a timestamp to the sensor data, which represents high precision temporal information. This digital circuitry has been successfully verified within a simulation environment. Furthermore, an asynchronous vision chip, based on this developed digital arbiter, was manufactured and the first test results validate the simulation performance. The paper is structured as follows. In Section II, a dedicated asynchronous vision chip is described. Section III presents the concept of the synchronous digital arbiter and the functionality. The performance analysis of this digital circuitry is presented in Section IV. A summary together with potential improvements are given in Section V.

II. THE ASYNCHRONOUS VISION CHIP

The asynchronous vision chip consists of asynchronous and autonomous optical sensor elements [6] that are sensitive to illumination intensity changes such that bus requests (events) are generated in case of exceeding a threshold. The terminals of the sensor elements are connected to separate inputs of the arbiter. To distinguish the respective inputs, a set of labels is used. These labels are address values which are uniquely assigned to the corresponding input of the arbiter. The combination of an event and the respective address yields the targeted spatial information of the sensor elements, which is represented by the Address Event (AE). Timestamps are representing the temporal information with respect to occurrences of the events. Thus, the chip provides accurate temporal and spatial information of illumination intensity changes at the output.

The authors thank the Austrian Council for Research and Technology Development for the financial support.

III. The Synchronous Digital Arbiter

This section describes the detailed architecture and the functionality of the digital arbiter. For illustration purpose, this concept is described on two rows of autonomous sensor elements.

A. TS Generation

Typically timestamps are assigned by the processor subsystem after arbitration. This setup has two disadvantages, which cause an inaccuracy of the timestamp value in general. First of all, the order of events occurring during the arbitration process can not be maintained. Hence, the order at the output depends on the arbitration strategy. Secondly, the arbitration process causes additional latency. The architecture presented here performs the timestamp assignment with respect to the occurrence of an event at the arbiter input. They are generated using a continuous counting device while assigning the current counter value to the events. The timestamps are combined with the corresponding address events to compose a stream of data packets, which are called Timed Address Events (TAE) at the output of the arbiter. Events with the same timestamp value are interpreted as concurrent and they are arbitrated according to descending addresses.

B. Architecture

The top level architecture of the synchronous arbiter and the Sensor Front End (SFE) is shown in Fig. 1. A central unit “Arbiter control” is used to control the arbiter units “intra row arbiter” and “inter row arbiter”. Additionally, it combines the timestamps with the corresponding address events by controlling the unit “TAE generation” accordingly. The latter generates the TAE output. The timestamp period T_{TS} is derived from the clock frequency f_{clk} and a programmable pre-scale value (*psv*) denoted in (1). From the system point of view this timestamp period is also the minimal time resolution.

$$T_{TS} = \frac{1}{f_{clk}} \cdot psv \qquad (1)$$

The SFE consists of the sensor elements and synchronization units “sync”, which synchronize the sensor element outputs with the system clock “clk” and adjust the pulse-width to one clock period. Each sensor element signals positive (ON) and negative (OFF) illumination changes on separate terminals. Both request lines “sreq_on” and “sreq_off” are represented by the “R_Y” (Y=0…J-1) signal in Fig. 1 and are representing the sensor element data of one bit (ON or OFF). A request (event) is acknowledged by the corresponding “A_Y” signal. Each row of sensor elements is connected to a separate “Intra row arbiter” unit, which is depicted in Fig. 2.

The input stage “EIF” of the “Intra row arbiter” stores the occurring events during one timestamp period T_{TS}. Every change of the timestamp value causes the transfer of the “EIF” status into the Event FIFOs called “EFIFO”. All “EFIFO” units are simultaneously controlled with the same sequence.

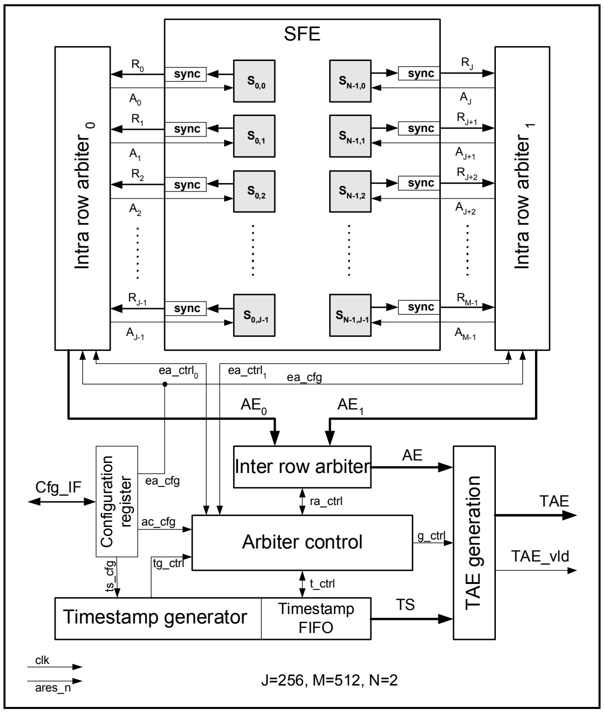

Figure 1. Top level architecture of the arbiter

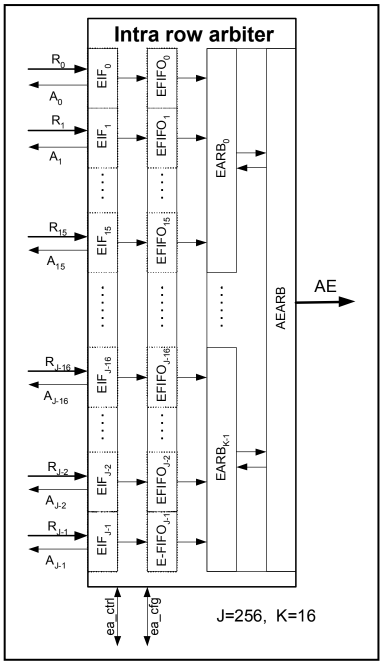

Figure 2. Intra row arbiter architecture

The EFIFOs are used to handle event rate peaks from the SFE while keeping the timestamp period, even if the event peak rate is higher than the maximum TAE rate for a given clock frequency f_{clk}. The control of the EFIFOs and the

"Timestamp FIFO" are carried out in the same way to keep the correlation consistent. In case of full EFIFOs (arbiter overflow), the timestamp period will be enlarged temporarily to avoid the loss of events. This occurrence is signaled on the TAE output. Upon storage of data in the EFIFOs, the arbitration process starts immediately. The hierarchically organized arbiter consists of three arbiter stages the Event ARBiter "EARB", Address Event ARBiter "AEARB" and the "Inter row arbiter". The arbiter stages are working in a pipelined manner. Additionally, each arbiter stage carries out two tasks simultaneously. They are generating address event values by assigning the corresponding address-parts according to the selected input and their stage. Concurrently, the next active input is determined. This method offers the utilization of the transfer bandwidth between the arbiter stages and the output bandwidth of the unit "TAE generation".

C. *TAE protocol*

The output of the TAE generation unit is used to transfer the timestamps and address events. Due to the limitation of the TAE bus width to 16 bit, the timestamp value and the address events are transferred sequentially, hereby offering the advantage that the timestamp value is transmitted once for corresponding address events. Bit 15 of the TAE data is used to distinguish between timestamp data and address events. Timestamp data and address events are always transferred together. If no events are pending no data are put out, with one exception. The timestamp counter wrap-around, indicated by bit 14 of the timestamp data is signaled in any case. This enables the processor to perform a timestamp value expansion. This sequence protocol is depicted in Fig. 3. An example of the "TAE" sequence exception is depicted in the lower half of Fig. 3. The occurrence of an arbiter overflow is signaled via bit 13 of the timestamp data. The timestamp value is represented by bits 13…0 of the timestamp data.

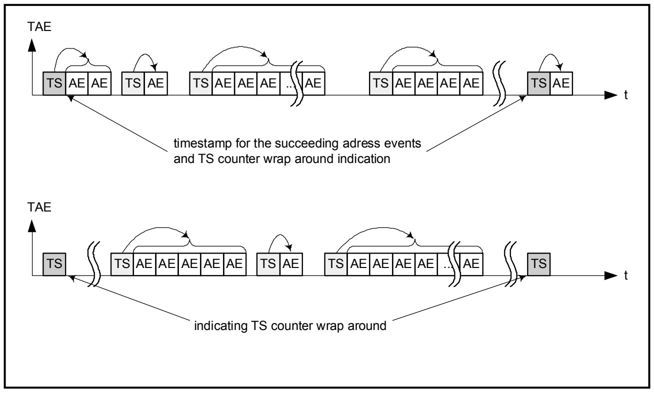
TAE
TS
AE
AE
TS
AE
TS
AE
AE
AE
..
AE
TS
AE
AE
AE
AE
TS
AE
t
timestamp for the succeeding adress events
and TS counter wrap around indication
TAE
TS
TS
AE
AE
AE
AE
AE
TS
AE
TS
AE
AE
AE
AE
..
AE
TS
t
indicating TS counter wrap around

Figure 3. Sequence protocol of TAE data types

IV. Performance Analysis

The performance considerations are structured in three parts. The first part describes the relationship between the application dependent maximum event rate of the SFE R_{SFEmax} and the time resolution. In the second part, the influence of the TAE protocol on the maximum address event transfer rate R_{AEmax} at the TAE output is discussed. An upper limit of the transfer performance is derived. The third part describes implementation details and simulation results.

T_{TS} represents the timestamp period for a given maximum event rate of one sensor element R_{SEmax}. To avoid the loss of events, T_{TS} has to be less than or equal to the reciprocal value of R_{SEmax} ($T_{TS} \leq 1/R_{SEmax}$). Introducing "EFIFO" offers the possibility of decreasing the time resolution T_{TS}, because the SFE event peak rate $R_{SFEpeak}$, which is defined by the number of active sensor elements per T_{TS}, can be up to $D_{FIFO} \cdot R_{AEmax}/16bit$, where D_{FIFO} describes the depth of the "EFIFO". In particular the time between events from different sensor elements can be measured more accurately.

The maximum output transfer rate R_{TAEmax} is mainly limited by the clock frequency. To simplify the interface timing with the processor subsystem, R_{TAEmax} was restricted to less than $16bit \cdot f_{clk}/2$(bit/s). The TAE data rate R_{TAE} is the sum of the timestamp date rate R_{TS} and the address event rate R_{AE} ($R_{TAE}=R_{TS}+R_{AE}$). Equation (2) defines the address event rate R_{AETTSi} during a certain timestamp period T_{TS}, including an arbitration-overhead of one clock period for the arbitration between address events labeled with different timestamps. N_i denotes the number of events during T_{TS}.

$$R_{AETTSi}=\frac{f_{clk}\cdot 16bit\cdot N_i}{2\cdot(N_i+1)+1}, \qquad \forall N_i \neq 0 \quad (2)$$

The maximum value R_{AEmax} is defined in (4), where N_{mean} is the mean value of the N_i as defined in (3)

$$N_{mean}=\frac{1}{D_{FIFO}}\sum_{i=1}^{D_{FIFO}} N_i, \qquad \forall N_i \neq 0. \quad (3)$$

$$R_{AE\max}=\frac{f_{clk}\cdot 16bit\cdot N_{mean}}{2\cdot(N_{mean}+1)+1} \quad (4)$$

To achieve a proper functionality the maximum event rate of the SFE R_{SFEmax} has to be less than or equal to $R_{AEmax}/16bit$ ($R_{SFEmax} \leq R_{AEmax}/16bit$) with $R_{SFEmax}=M_A \cdot R_{SEmax}$[1], where M_A is the number of active sensor elements. In the worst case, the number of events during $T_{TS} \cdot D_{FIFO}$ is assumed to be M_A as described in (5)

$$M_A=\sum_{i=1}^{D_{FIFO}} N_i, \qquad \forall N_i \neq 0. \quad (5)$$

Equation (6) describes the upper limit of the transfer performance, which is the typical case for selecting the value of T_{TS} for a given R_{SEmax} and M_A

$$R_{SFE\max} \leq \frac{M_A}{T_{TS}\cdot D_{FIFO}} \leq \frac{R_{AE\max}}{16bit}, \qquad \forall N_i \neq 0. \quad (6)$$

[1] Assuming that all sensor elements have the same R_{SEmax}.

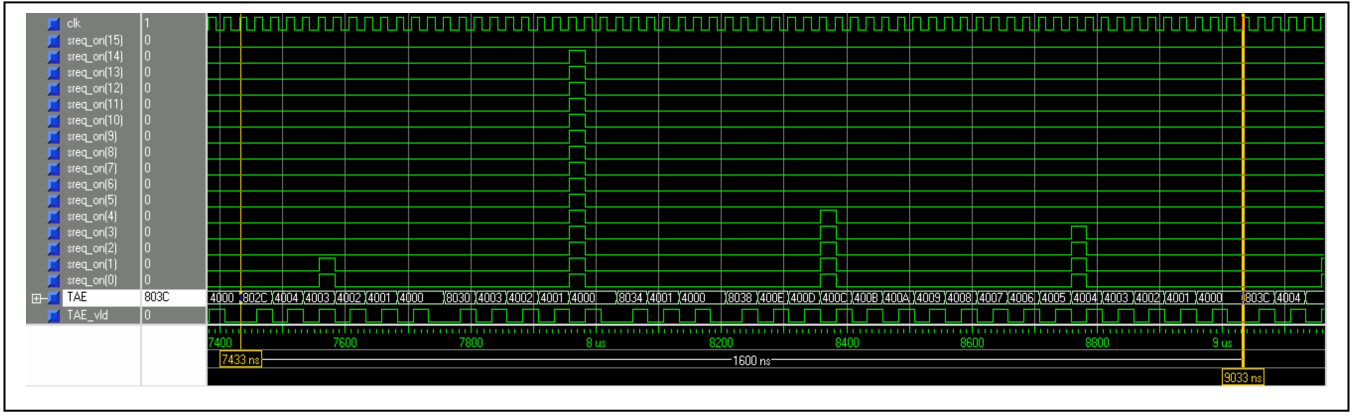

Figure 4. Simulation result for the parameter values: f_{clk} = 40 MHz, N_i = {2, 15, 5, 4, ...}, T_{TS} = 400 ns

Considering timestamp periods with $N_i = 0$ results in a higher transferable event peak rate $R_{SFEpeak}$ (max. events per T_{TS}) for a given T_{TS}, because the available time for transferring address events is increased by $T_{TS}\cdot L$, where L denotes the number of consecutive timestamp periods with $N_i = 0$. Equation (7) provides a rough estimation

$$\frac{M_A}{T_{TS}\cdot(D_{FIFO}+L)} < \frac{f_{clk}}{2}. \qquad (7)$$

The current implementation was designed for a maximum clock frequency f_{clk} of 40 MHz. For the EFIFO depth a value of four was chosen to achieve a good tradeoff between production costs in terms of chip area and $R_{SFEpeak}$. The timestamp period T_{TS} is programmable in the range of $4/f_{clk}$ to $65532/f_{clk}$ in steps of 4.

Figure 4 depicts the waveform of a simulation where the x-axis represents the time. The y-axis contains the clock port "clk" and the fifteen synchronized request inputs "sreq_on(15…0)" of the active sensor elements (15…0) generating ON events. The port "TAE" represents the 16 bit TAE output (in hexadecimal), where bit 14 denotes either an ON event (bit 14 = 1) or an OFF event (bit 14 = 0), of the arbiter. The rising edge of the port "TAE_vld" indicates the validity of a TAE data value. For the simulation, a T_{TS} of 400 ns and a sequence of $N_I = \{2, 15, 5, 4, \ldots\}$ were chosen which results in the AE rate R_{AE} of $16.25\cdot10^6$ AE per second on the TAE output where the SFE peak rate $R_{SFEpeak}$ is $37.5\cdot10^6$ events per second.

The chip was fabricated in a 0.35µm process and first test results, using the build in test generators, confirm the simulation results.

V. Conclusion

This paper presented a dedicated digital arbiter concept proposed for vision applications including the description of the architecture and performance. The principle can be applied to a wide range of different applications, because the arbiter can easily be adapted to other sensor types or other sensor element arrays due to the modular architecture. The clock frequency can be adjusted to the application requirements due to the high utilization of the maximum output transfer rate determined by the clock frequency.

Preliminary test results of the fabricated asynchronous vision chip using this digital arbiter are promising. By using a clock frequency of 40 MHz, a minimal time resolution of 100ns for resolving the events and a maximum transfer rate of nearly 320 Mbit/s on a 16 bit wide bus is supported.

An important issue regarding most applications is the power consumption. To further improve the implementation of the arbiter it is intended to decrease the power consumption by implementing clock gating. This possibility was already addressed by considering clock gating during the definition of the structure, because all parts processing with different rates reside in different modules.

A patent application covering the arbiter architecture and functionality has been filed under Austrian patent application number A1649/2005.

References

[1] A. Hornberg, "Handbook of Machine Vision," *Wiley*, 2006.

[2] W.J. Dally and JW Poulton, "Digital Systems Engineering," *Cambridge University Press*, Cambridge UK, 1998.

[3] J. Wang and A.K. Asundi, "A computer vision system for wineglass defect inspection via Gabor-filter-based texture features," *Journal of Information Sciences*, vol. 127, pp. 157-171, August 2000.

[4] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade, "Introduction to the Special Section on Video Surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.22(8), , pp.745-746, August, 2000.

[5] A. Torralba, K. Murphy, W. Freeman, and M. Rubin, "Context-Based Vision System for Place and Object Recognition," *In Proc. 9th IEEE Int. Conf. on Computer Vision (ICCV),* vol. 1, pp. 273-280 , 2003

[6] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128x128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change," in *IEEE International Solid-State Circuits Conference (ISSCC2006)*, San Francisco, USA, Februay 2006.