Spike Processing on an Embedded Multi-task Computer: Image Reconstruction

C. Luján-Martínez, A. Linares-Barranco, M. Rivas-Pérez,
Á. Jiménez-Fernández, G. Jiménez-Moreno and A. Civit-Balcells

Dept. de Arquitectura y Tecnología de Computadores,
Universidad de Sevilla, Sevilla, Spain.
{cdlujan,alinares,mrivas,ajimenez,gaji,civit}@atc.us.es

Fifth Workshop on Intelligent Solutions in Embedded Systems.

WISES 07, U. Carlos III, Madrid, Spain.
June 2.007.

Neuro-inspired systems

are electronic circuits that mimic the processing of the biological nervous system.

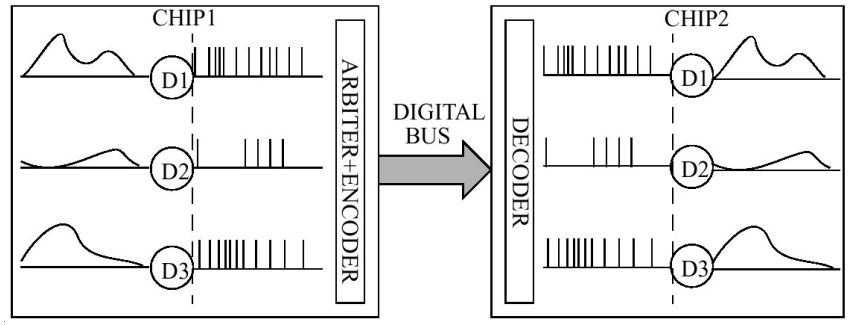
In biology,

the neural system is divided in layers of neurons. These layers are processing the information from the sensors.

- Each neuron of a layer is connected to a projective field of neurons in other layers, following a weighted pattern.
- Neurons communicate each others using ionic pulses, called *spikes*, with time constants of milliseconds.
- Information resides in the pulses frequency.

Problem:

The connectivity between layers implies a huge number of point to point connections and chips have limited the number of pins.

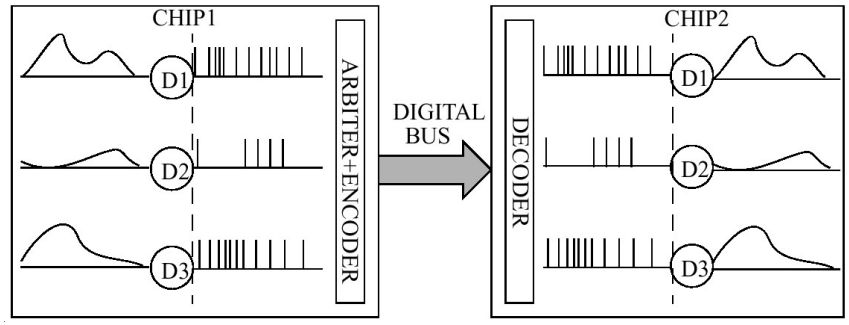


Solution:

Temporal multiplexation of neuron's pulses in a high speed digital bus, Address-Event-Representation.

Problem:

The connectivity between layers implies a huge number of point to point connections and chips have limited the number of pins.



Solution:

Temporal multiplexation of neuron's pulses in a high speed digital bus, Address-Event-Representation.

Each time a cell on a sender device generates a spike,

a digital word representing a code or address for that cell is placed on the external inter-chip digital bus, the AER bus. This word is called *event*.

In the receiver chip,

the spikes or events are guided to the cells whose code or address appeared on the bus.

In this way,

cells with the same address in the emitter and receiver chips are virtually connected by streams of spikes.

Additional handshaking lines,

Acknowledge and Request, are used for completing the asynchronous communication.

In addition,

transmitting the cell addresses allows performing extra operations on the events while they travel from one chip to another. The output of a silicon retina

- can be easily translated,
- scaled, or
- rotated by simple mapping operations on the emitted addresses,

making AER not only a communication channel.

In artificial vision systems based in AER,

it is widely used the *rate-coded* AER

In this scheme,

each cell corresponds to a pixel and its activity is transformed into pixel event frequency.

Although,

it is inefficient for conventional image transmission:

Monochrome VGA resolution yields a peak rate of

$$(480 \times 640 \text{ pixels/frame}) \times (256 \text{ spikes/pixel}) \times (25 \text{ frames/s}) \times (19 \text{ bit/spike}) = 37 \text{ Gbit/s}.$$

On the other hand,

the lost of some events does not mean a degradation in the application.

Transmitting preprocessed images instead of raw images,

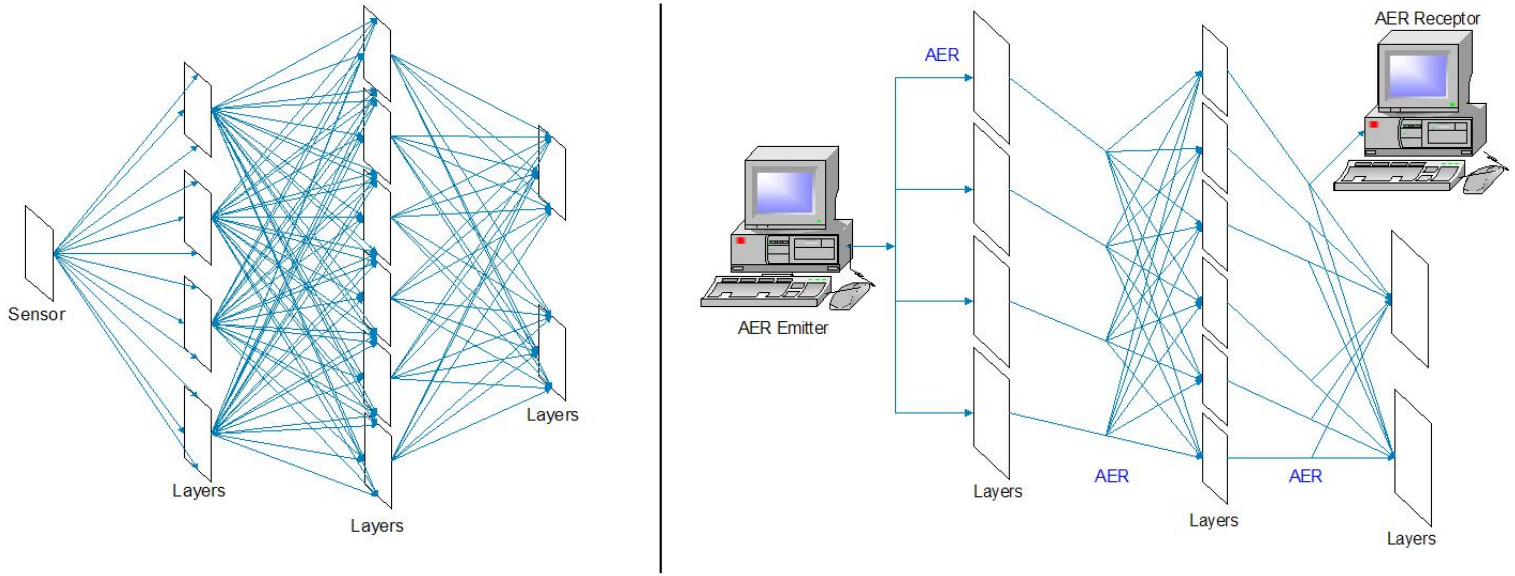
such as edges or contrast, in which 20 gray levels are satisfactory and only (1-10 %) of pixels will present appreciable contrast.

$2 \sim 3$ orders of magnitude

In addition,

present day AER hardware uses image resolutions between 64×64 and 128×128 pixels at the most.

$1 \sim 2$ orders of magnitude



In multistage systems,

- events are generated at the front end, they travel and are processed down the whole chain (without waiting to finish processing each frame). Also,
- information is reduced after each stage, thus reducing the event traffic.

AER-tools and Event Rates:

- Rome PCI-AER $\longrightarrow$ 1 Mevents/s
- CAVIAR PCI-AER $\longrightarrow$ 8 Mevents/s
- USB-AER $\longrightarrow$ 25 Mevents/s
- USB2AER $\longrightarrow$ 5 Mevents/s
- mini USB-AER $\longrightarrow$ 300 Kevents/s

They

- are mainly based on reconfigurable hardware.
- achieve a high bandwith.
- useful for mapping, monitoring and sequencing.
- But a PC is needed for the high level spike processing.

AER-tools and Event Rates:

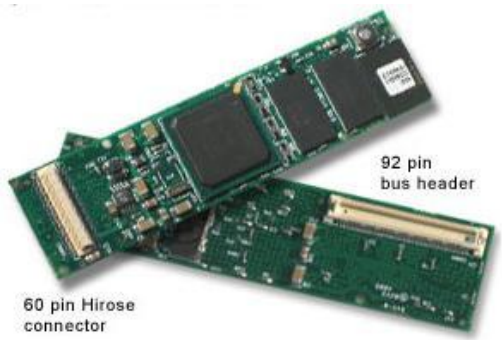
- Rome PCI-AER $\longrightarrow$ 1 Mevents/s
- CAVIAR PCI-AER $\longrightarrow$ 8 Mevents/s
- USB-AER $\longrightarrow$ 25 Mevents/s
- USB2AER $\longrightarrow$ 5 Mevents/s
- mini USB-AER $\longrightarrow$ 300 Kevents/s

They

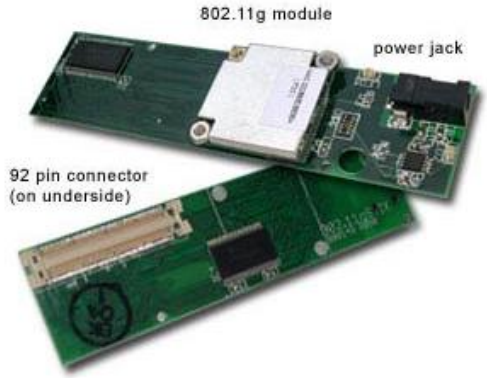
- are mainly based on reconfigurable hardware.
- achieve a high bandwith.
- useful for mapping, monitoring and sequencing.
- But a PC is needed for the high level spike processing.

The Embedded Computer

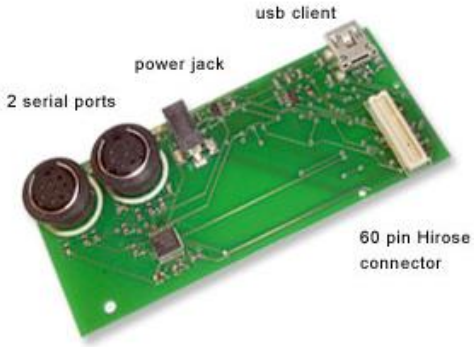
Connex board:



Wifistix board:



Tweener board:



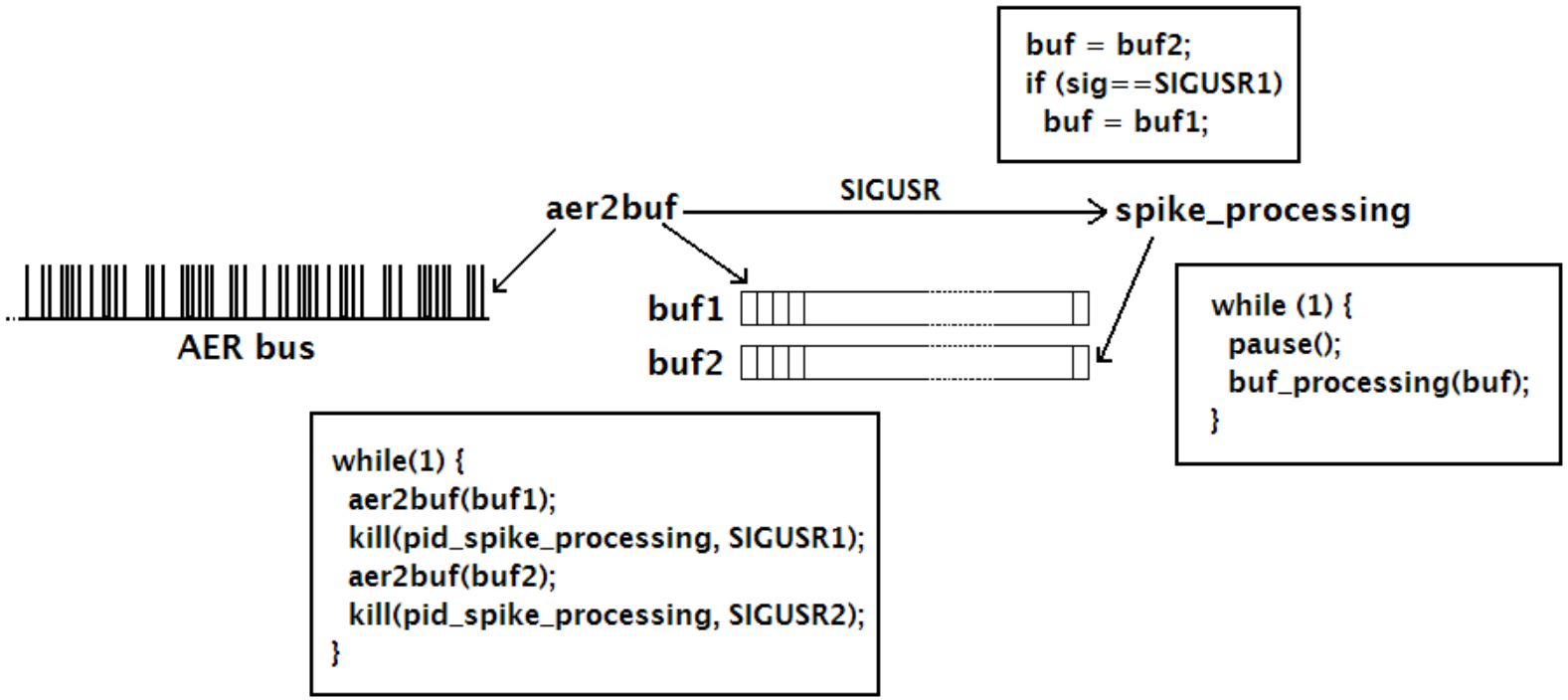
Hardware:

- Intel XScale PXA255 400 MHz (32 bit processor).
- 32 KB data cache and 32 KB instruction cache
- 64 MB RAM (with MMU)
- 84 GPIOs (serial ports, I2C, PWM, LCD, USB client 1.1,...)
- 16 MB of Flash mem.
- wireless connectivity (Bluetooth or IEEE 802.11b).

Software:

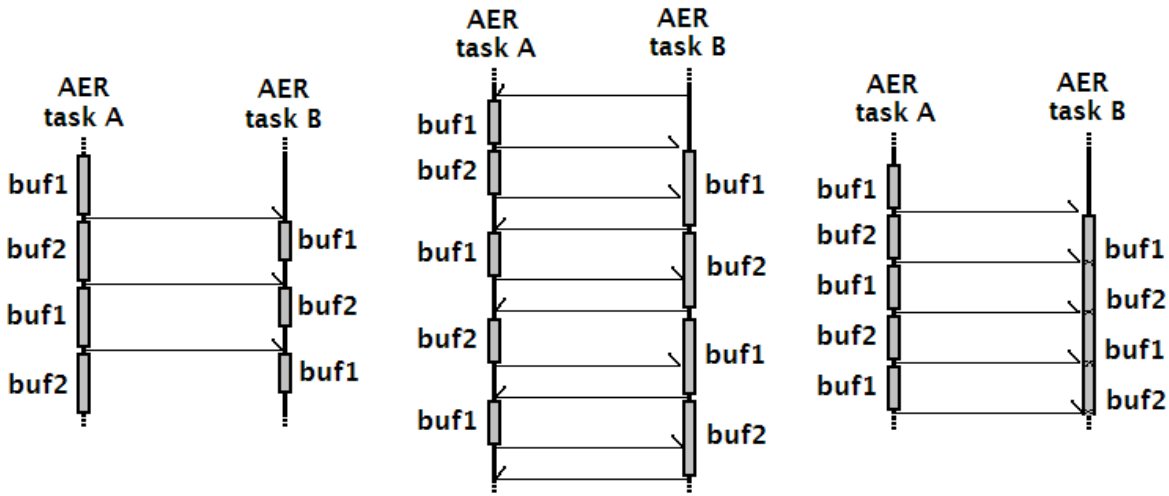
- GNU/Linux 2.6 kernel.
- Buildroot.
- uClibc.

- for high level spiking processing
- over a multi-task system
- when receiving events from the AER bus.



Spike Processing over Multi-task

II. Execution Flowchart Possibilities



Filling the buffer lasts more than consuming it, AER task B.

An acknowledge signal is used when AER task B has finished using the buffer.

AER task B aborts and starts using an updated buffer.

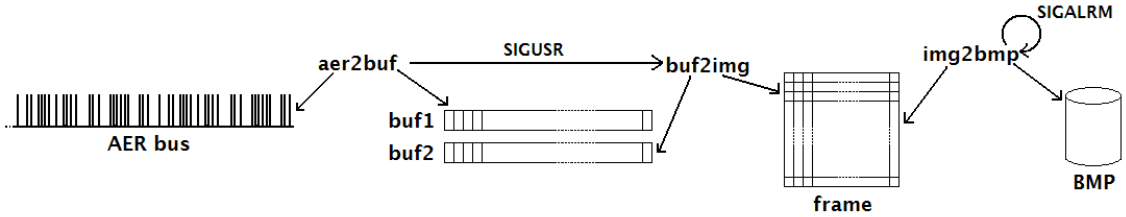
- If T_{frame} is the duration of a single frame, $\Rightarrow$ $t = n \times T_{frame} (n \in [0, \infty))$, called the integration time.

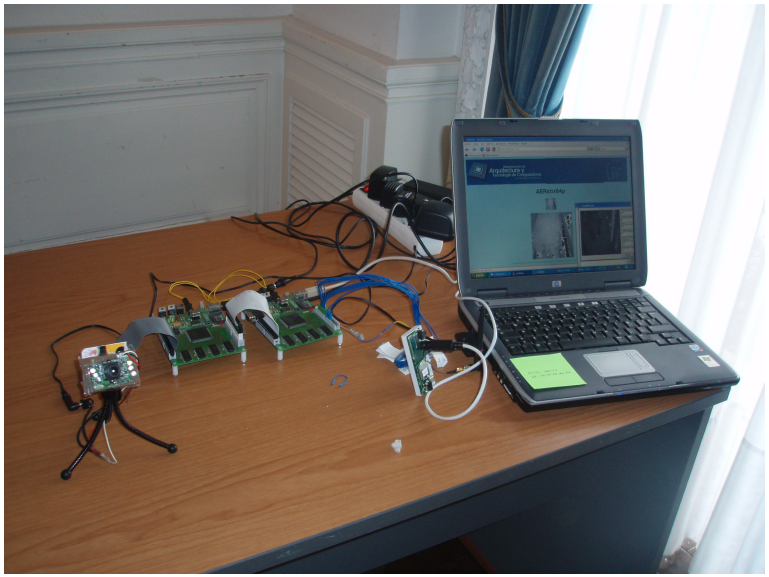
Then,

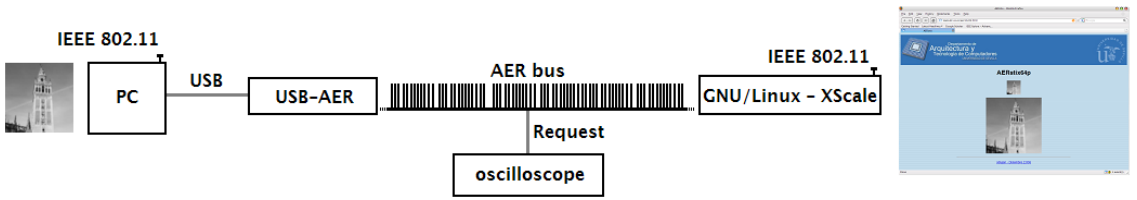
1. for each event address, $[(x, y)] \leftarrow [(x, y)] + 1$.
2. at $t = (n + 1) \times T_{frame}$, the content of the 2-D memory is reset.

Image Reconstruction from Event Stream

II. Software Architecture and Scenario







Event Rate, ER, and Worst Event Rate, WER, in Kevents/s:

Test	Timer	No other process		Other processes	
		ER	WER	ER	WER
Processes	100 Hz	540	450	530	200
Threads	100 Hz	770	620	770	259
Processes	1000 Hz	500	430	500	430
Threads	1000 Hz	775	660	770	660

By setting HZ to 1000

the ER is enclosed to a “controlled” interval.

Event Rate, ER, and Worst Event Rate, WER, in Kevents/s:

Test	Timer	No other process		Other processes	
		ER	WER	ER	WER
Processes	100 Hz	540	450	530	200
Threads	100 Hz	770	620	770	259
Processes	1000 Hz	500	430	500	430
Threads	1000 Hz	775	660	770	660

By setting HZ to 1000

the ER is enclosed to a “controlled” interval.

GNU/Linux 2.6:

SCHED_RR:	SCHED_FIFO:	SCHED_OTHER:
each process is only allowed to run for a maximum time quantum.	a simple scheduling algorithm without time slicing.	dynamic priority time-sharing scheduler, based on the nice level and increased for each time quantum the process is ready to run, but denied to run by the scheduler.

- "scheduling" $\in O(1)+$ pre-emptive code $\Rightarrow$ better response time.

Ruled by SCHED_RR

"threads implementation" remains **840 KEvents/s**.

GNU/Linux 2.6:

SCHED_RR:	SCHED_FIFO:	SCHED_OTHER:
each process is only allowed to run for a maximum time quantum.	a simple scheduling algorithm without time slicing.	dynamic priority time-sharing scheduler, based on the nice level and increased for each time quantum the process is ready to run, but denied to run by the scheduler.

- "scheduling" $\in O(1)+$ pre-emptive code $\Rightarrow$ better response time.

Ruled by SCHED_RR

"threads implementation" remains **840 KEvents/s**.

The minimum time between events, the system is able to achieve, is

1.16 μs.

So, the mean Inter-Spike-Interval, ISI, for

SCHED_OTHER:	**SCHED_RR:**
ISI = **1.29 μs**	ISI = **1.19 μs**

Therefore, the threads implementation presents a deviation from the best of

SCHED_OTHER:	**SCHED_RR:**
11%	2.5%

The minimum time between events, the system is able to achieve, is

1.16 μs.

So, the mean Inter-Spike-Interval, ISI, for

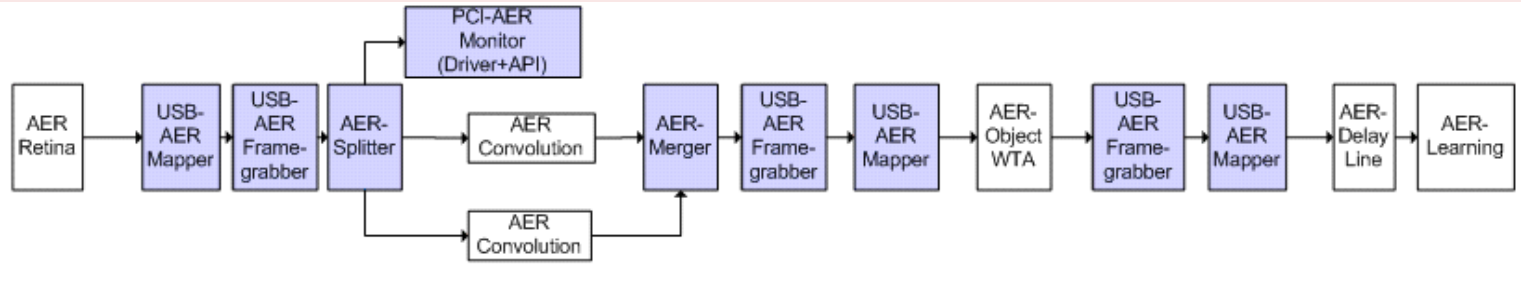
SCHED_OTHER:	**SCHED_RR:**
ISI = **1.29 μs**	ISI = **1.19 μs**

Therefore, the threads implementation presents a deviation from the best of

SCHED_OTHER:	**SCHED_RR:**
11%	**2.5%**

CAVIAR: a neuromorphic vision system totally based on AER where

the maximum throughput ER takes place at the output of the silicon retina for real applications.



ER $\in$ **[8, 150 Kevents/s]** $\Rightarrow$ **ISI** $\in$ **[6.66** μ**s, 123.8** μ**s]**

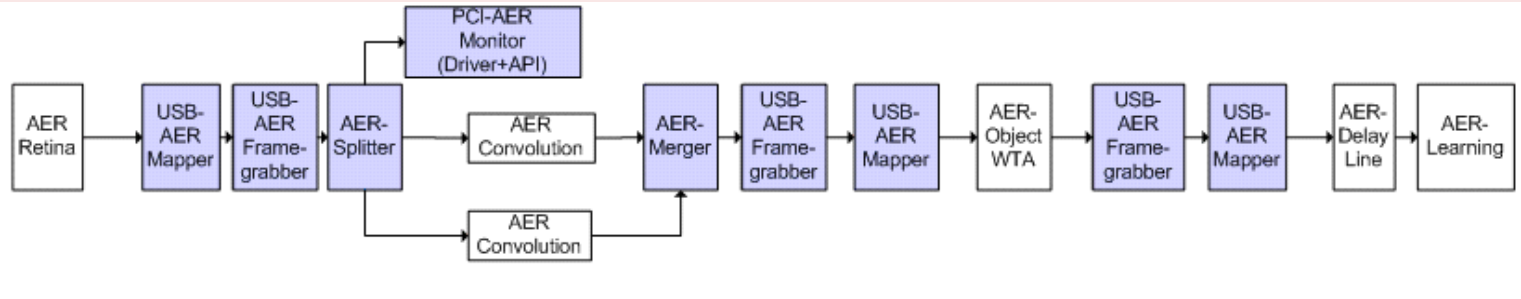
[P. Lichtsteiner and T. Delbruck. 64$\times$64 Event-Driven Logarithmic Temporal Derivative Silicon Retina. *Proceedings of IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, pages 157-160, 2005.]

Wich means

from **5.5** μ**s** to **122.6** μ**s**, or from **2200** to **50000 instructions**,
on a 32-bit μp at 400 MHz for any kind of high level spike processing.

CAVIAR: a neuromorphic vision system totally based on AER where

the maximum throughput ER takes place at the output of the silicon retina for real applications.



ER $\in$ [**8, 150 Kevents/s**] $\Rightarrow$ **ISI** $\in$ [**6.66** μ**s, 123.8** μ**s**]

[P. Lichtsteiner and T. Delbruck. 64$\times$64 Event-Driven Logarithmic Temporal Derivative Silicon Retina. *Proceedings of IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, pages 157-160, 2005.]

Wich means

from **5.5** μ**s** to **122.6** μ**s**, or from **2200** to **50000 instructions**,
on a 32-bit μp at 400 MHz for any kind of high level spike processing.

Conclusion

- New philosophy of spike processing using a standalone multi-task environment directly connected to the AER bus,
- achieving up to **840 KEvents/s** while constructing a frame and
- letting the execution of [**2200, 50000**] 32 bit μp instructions,
- allowing high level spike processing while performing AER communication for high demanding applications.

Acknowledgements

We want to special thank our colleague at Computer Architecture and Technology Department from the University of Seville, Francisco Gómez-Rodríguez, for his great interest and his useful comments when carrying out this work.
We would also like to thank the NSF sponsored Telluride Neuromorphic Engineering Workshop, where this idea was born in a discussion group participated by Daniel Fasnacht, Giacomo Indiveri, Alejandro Linares-Barranco and Francisco Gómez-Rodríguez.

This work was supported by Spanish grant TEC2006-11730-C03-02 (SAMANTA 2).

Conclusion

- New philosophy of spike processing using a standalone multi-task environment directly connected to the AER bus,
- achieving up to **840 KEvents/s** while constructing a frame and
- letting the execution of [**2200, 50000**] 32 bit μp instructions,
- allowing high level spike processing while performing AER communication for high demanding applications.

Acknowledgements

We want to special thank our colleague at Computer Architecture and Technology Department from the University of Seville, Francisco Gómez-Rodríguez, for his great interest and his useful comments when carrying out this work.
We would also like to thank the NSF sponsored Telluride Neuromorphic Engineering Workshop, where this idea was born in a discussion group participated by Daniel Fasnacht, Giacomo Indiveri, Alejandro Linares-Barranco and Francisco Gómez-Rodríguez.

This work was supported by Spanish grant TEC2006-11730-C03-02 (SAMANTA 2).

Conclusion

- New philosophy of spike processing using a standalone multi-task environment directly connected to the AER bus,
- achieving up to **840 KEvents/s** while constructing a frame and
- letting the execution of [**2200, 50000**] 32 bit μp instructions,
- allowing high level spike processing while performing AER communication for high demanding applications.

Acknowledgements

We want to special thank our colleague at Computer Architecture and Technology Department from the University of Seville, Francisco Gómez-Rodríguez, for his great interest and his useful comments when carrying out this work.
We would also like to thank the NSF sponsored Telluride Neuromorphic Engineering Workshop, where this idea was born in a discussion group participated by Daniel Fasnacht, Giacomo Indiveri, Alejandro Linares-Barranco and Francisco Gómez-Rodríguez.

This work was supported by Spanish grant TEC2006-11730-C03-02 (SAMANTA 2).

Conclusion

- New philosophy of spike processing using a standalone multi-task environment directly connected to the AER bus,
- achieving up to **840 KEvents/s** while constructing a frame and
- letting the execution of [**2200, 50000**] 32 bit μp instructions,
- allowing high level spike processing while performing AER communication for high demanding applications.

Acknowledgements

We want to special thank our colleague at Computer Architecture and Technology Department from the University of Seville, Francisco Gómez-Rodríguez, for his great interest and his useful comments when carrying out this work.
We would also like to thank the NSF sponsored Telluride Neuromorphic Engineering Workshop, where this idea was born in a discussion group participated by Daniel Fasnacht, Giacomo Indiveri, Alejandro Linares-Barranco and Francisco Gómez-Rodríguez.

This work was supported by Spanish grant TEC2006-11730-C03-02 (SAMANTA 2).