he Third International Conference on Advances in Signal, Image and Video Processing
SIGNAL 2018
May 20, 2018 to May 24, 2018 - Nice, France

Sensing and Sampling for Low-Power Applications

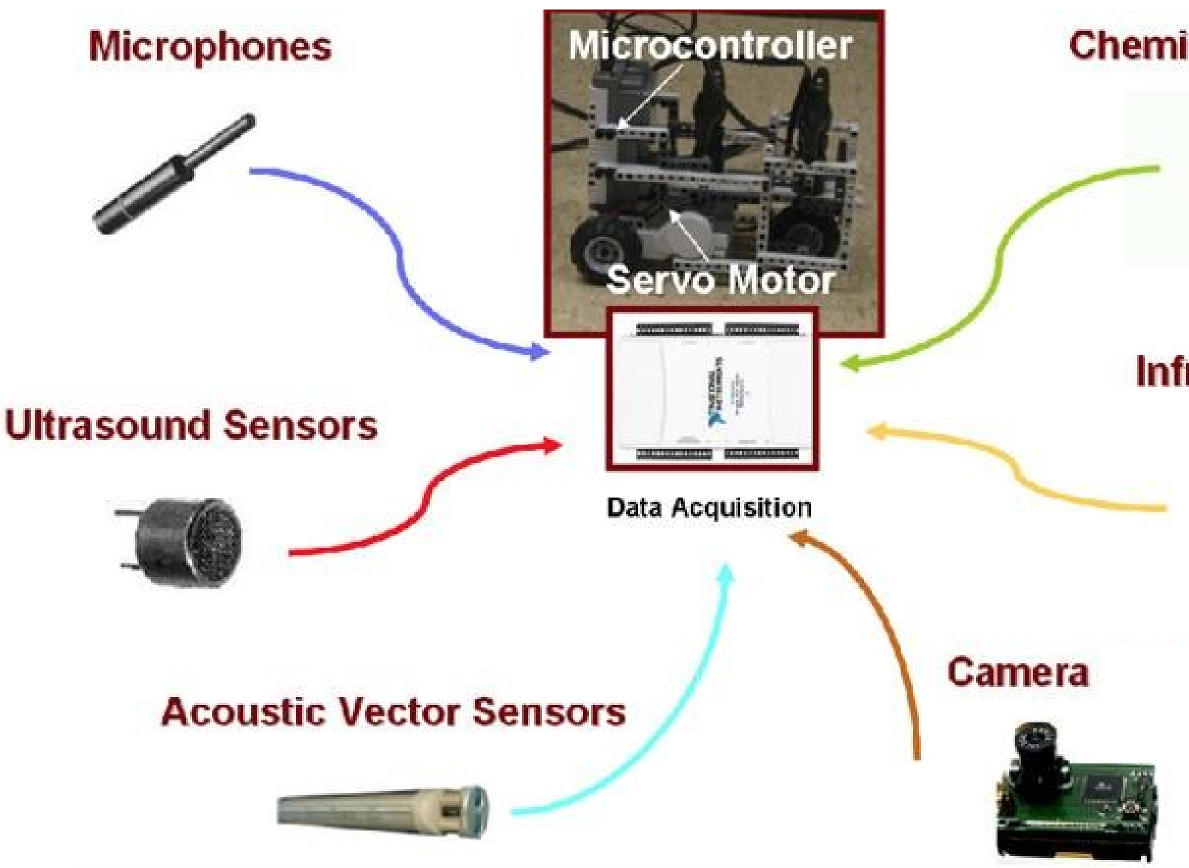
Laurent Fesquet

University Grenoble Alpes / CNRS – TIMA *– Grenoble, France*

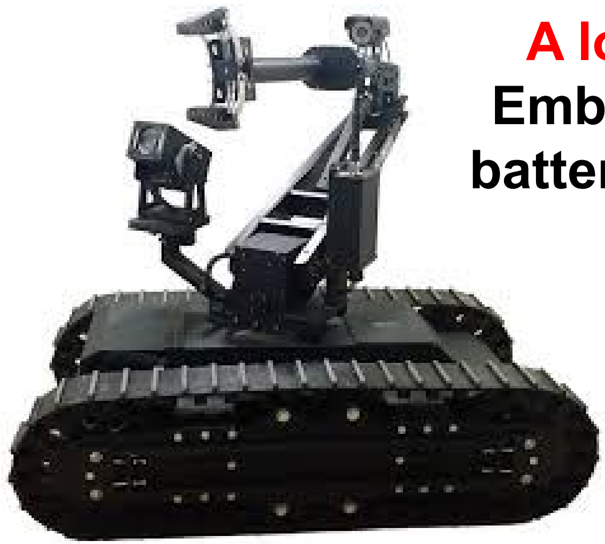
EPFL – ICLab *– Neuchâtel, Switzerland*

Laurent.Fesquet@univ-grenoble-alpes.fr

Sensors: the eyes of your application!

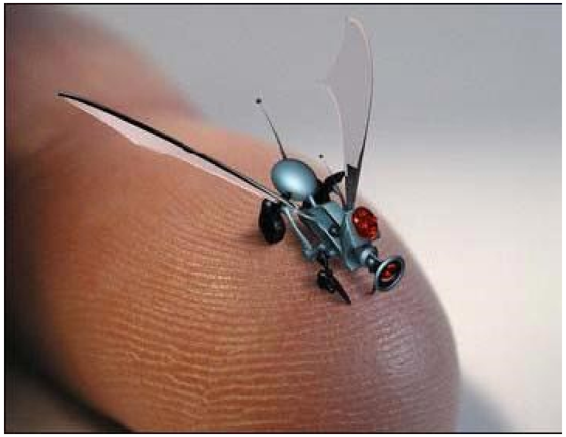


Enough energy for your robot?



A lot of sensors!
Embed a very large battery for autonomy

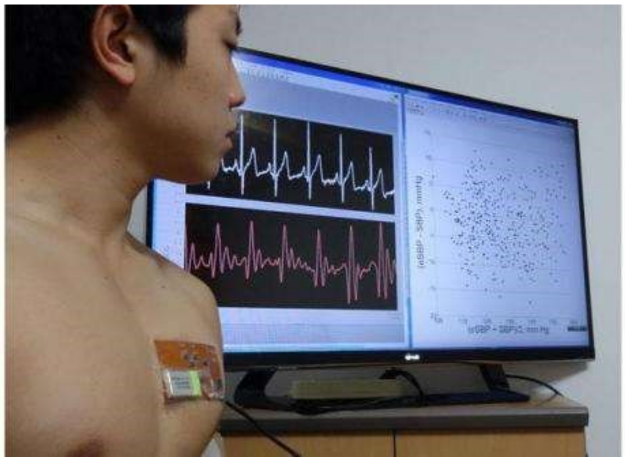
A smaller one here!

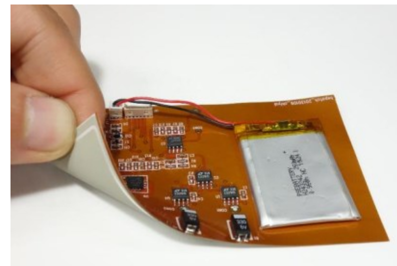


But here?

Trade-off between:
- Computation capabilities
- Sensing capabilities
- Actuation capabilities
- Autonomy

Enough memory for your application?





Patch sensor for continuous blood pressure long-term monitoring

S. Noh, et al. 2012

Memory accordingly chosen with:
- Sampling frequency
- Monitoring time
- Memory power consumption
- RF transmission system

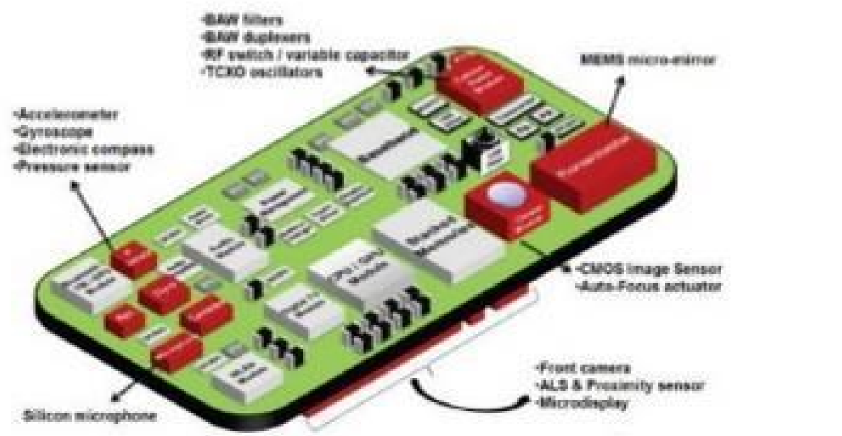
Patch sensor for fall detection

C. Arslan, et al. 2015

Smart Sensors and Actuators

MEMS in Smart Phones

- Accelerometers
- Gyroscopes
- Electronic Compass
- Pressure Sensors
- Microphones
- Micro speakers
- Auto focus
- (Pico) Projectors
- RF MEMS

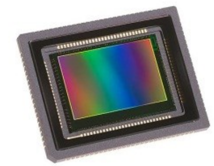


Source: Yole Development

Source: MEMS Technology Roadmapping, Michael Gaitan, NIST Chair, iNEMI and ITRS MEMS Technology Working Groups Nano-Tec Workshop 3, 31 May 2012

Sensors consume by themselves and produce computational activity

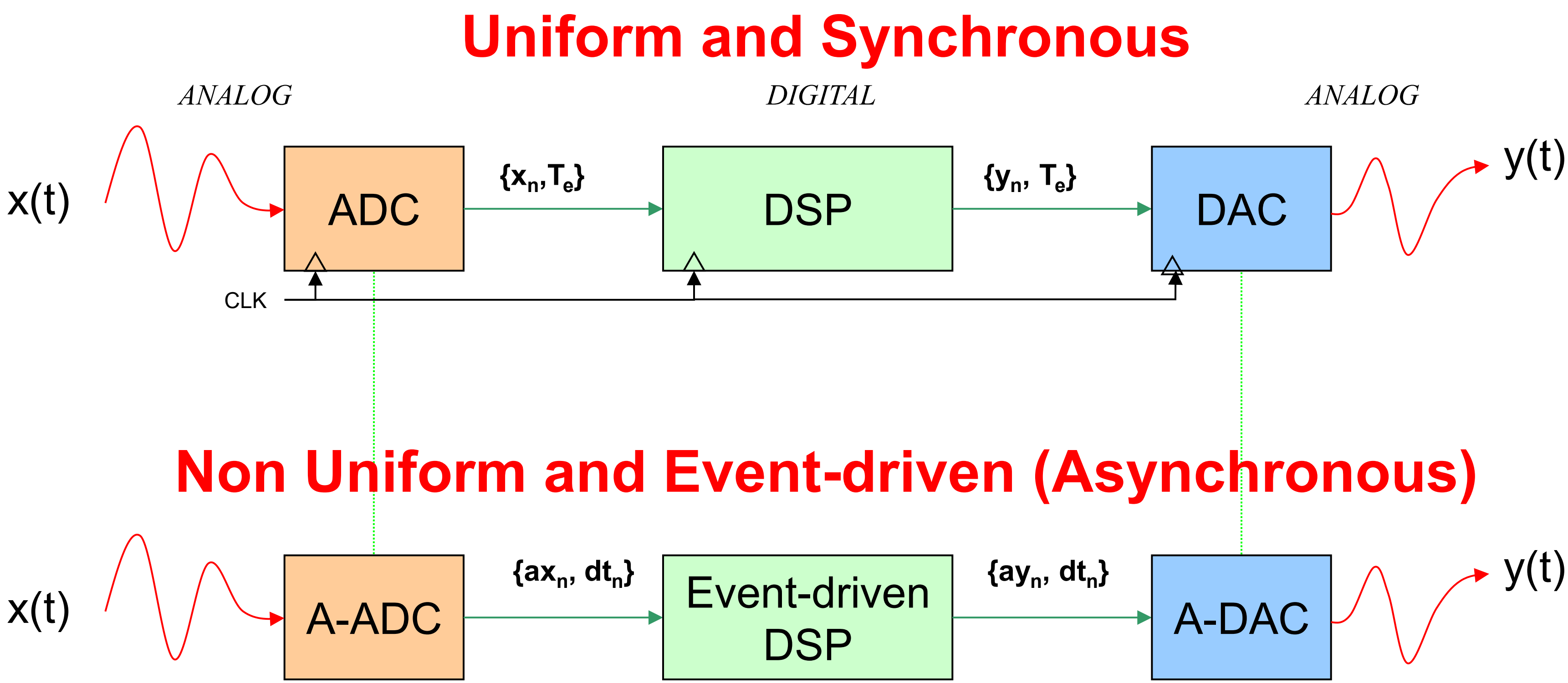
Internet of Things Challenges



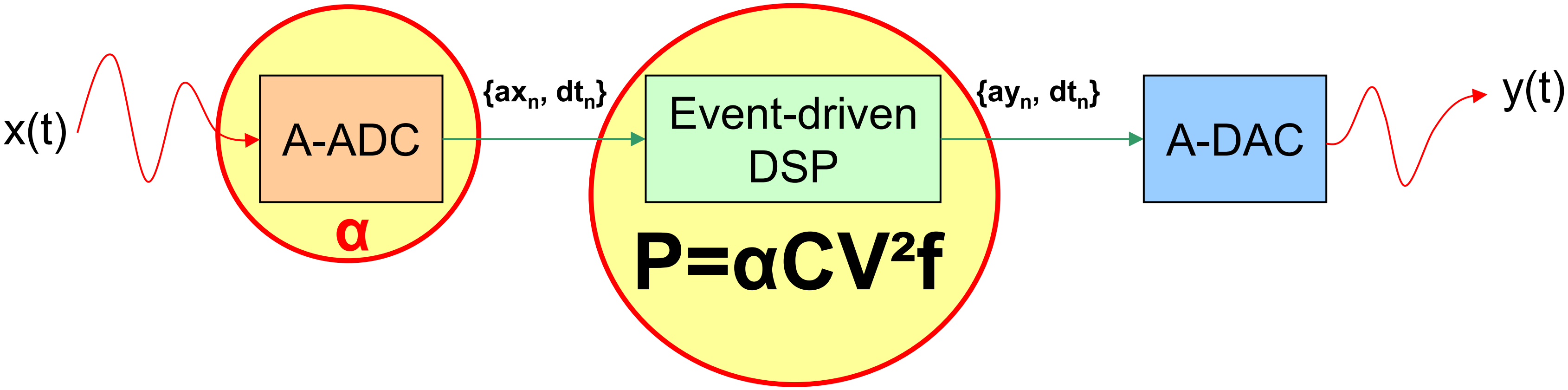
+ more data
+ more storage
+ more communications
+ more computations
+ more consumption

Nyquist-Shannon Theorem

What can we do?
Change sampling and processing!



How to mitigate energy in electronic systems?



- Power consumption is sensitive to V^2, f and C

➡ Reduce **V, f and C**

… **but you will loose performances**

Many, many, many references on V, f and C reduction!!!

- Other option:

➡ **Reduce the activity α**

Important questions 1/2

David E. Muller and Ivan Sutherland

Do we need a clock for synchronizing digital circuits?

- No! It exist plenty of circuit synchronization alternatives!!!

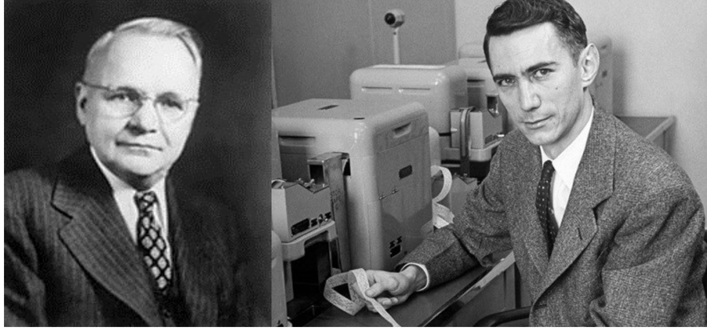
Are this “asynchronous” circuits realistic?

- Yes, indeed! Many have already been fabricated.
- Intel’s neuromorphic chips are asynchronous!

How to reduce power consumption ?

Remove the useless activity and suppress the clock.

Important questions 2/2



Harry Nyquist and Claude Shannon

Claude Shannon and Harry Nyquist are they responsible of the digital data deluge?

- We will not answer to this question but Big Data is today a reality!
- Big Data is power-hungry

Can we find a better sampling scheme to stem this digital data deluge and stop the energy waste?

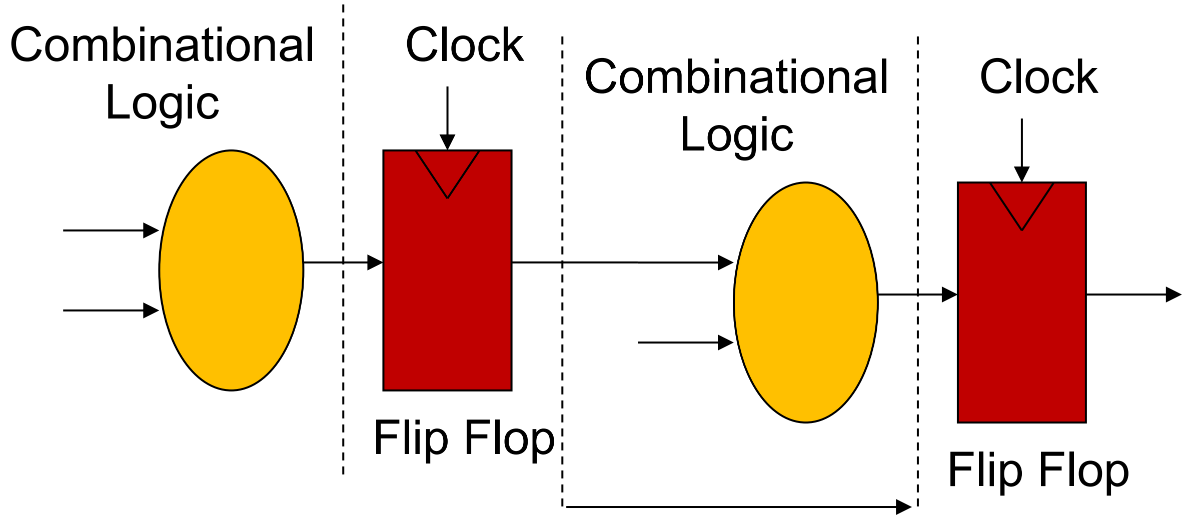
- We hope so !!!

Outline

- Event-driven circuits
- Non-uniform sampling
- Sampling in matrix sensors
- Conclusion

Designing synchronous circuits

- Synchronous circuit model
- Synchronous circuits use timing assumptions



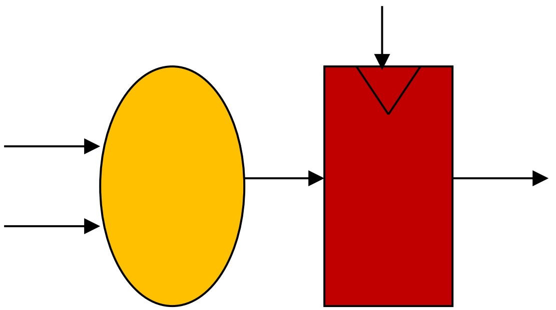
Critical path = Longuest path (worst case)

Correct behavior when Tcritical < Tclock

Asynchronous Circuits Principles

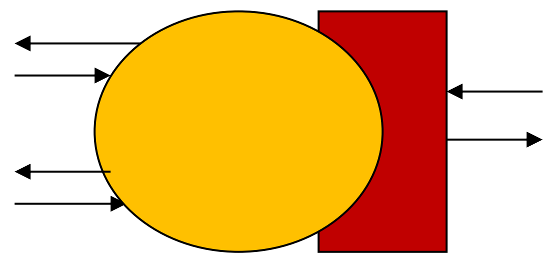
- At the hardware module abstraction

Every rising edge clock
triggers the computation



Synchronous

Data availability
triggers the computation



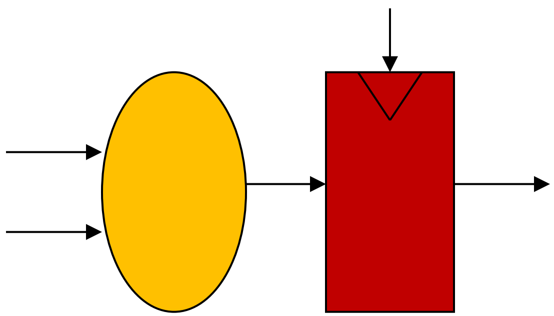
Asynchronous

→ **Global Clock** is *replaced*
→ by **local channels** (handshaking)

Asynchronous Circuits Principles

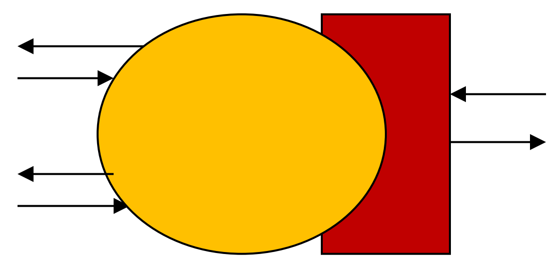
- Data-flow instead of control-flow

if rising_edge of clock **then**
    send output = f(inputs)
else
    output remains unchanged
end if



Synchronous

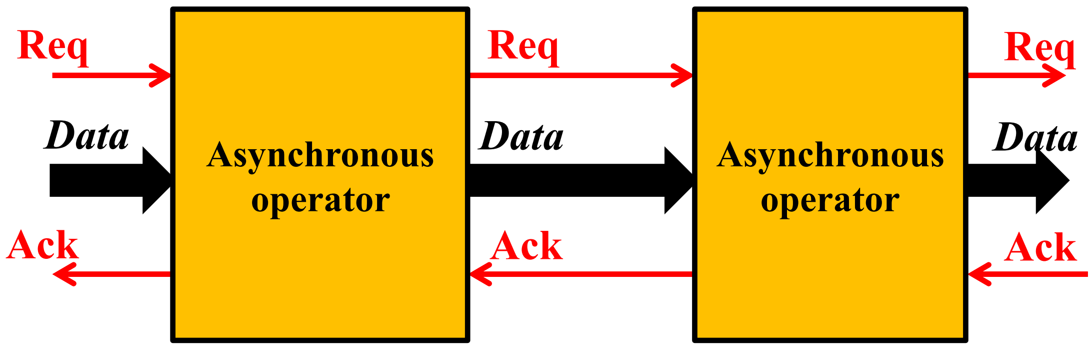
wait for valid inputs
    output = f(inputs)
complete input transactions
wait for output ready to receive
    send output
complete output transaction



Asynchronous

Asynchronous Circuits Principles

Clockless circuits

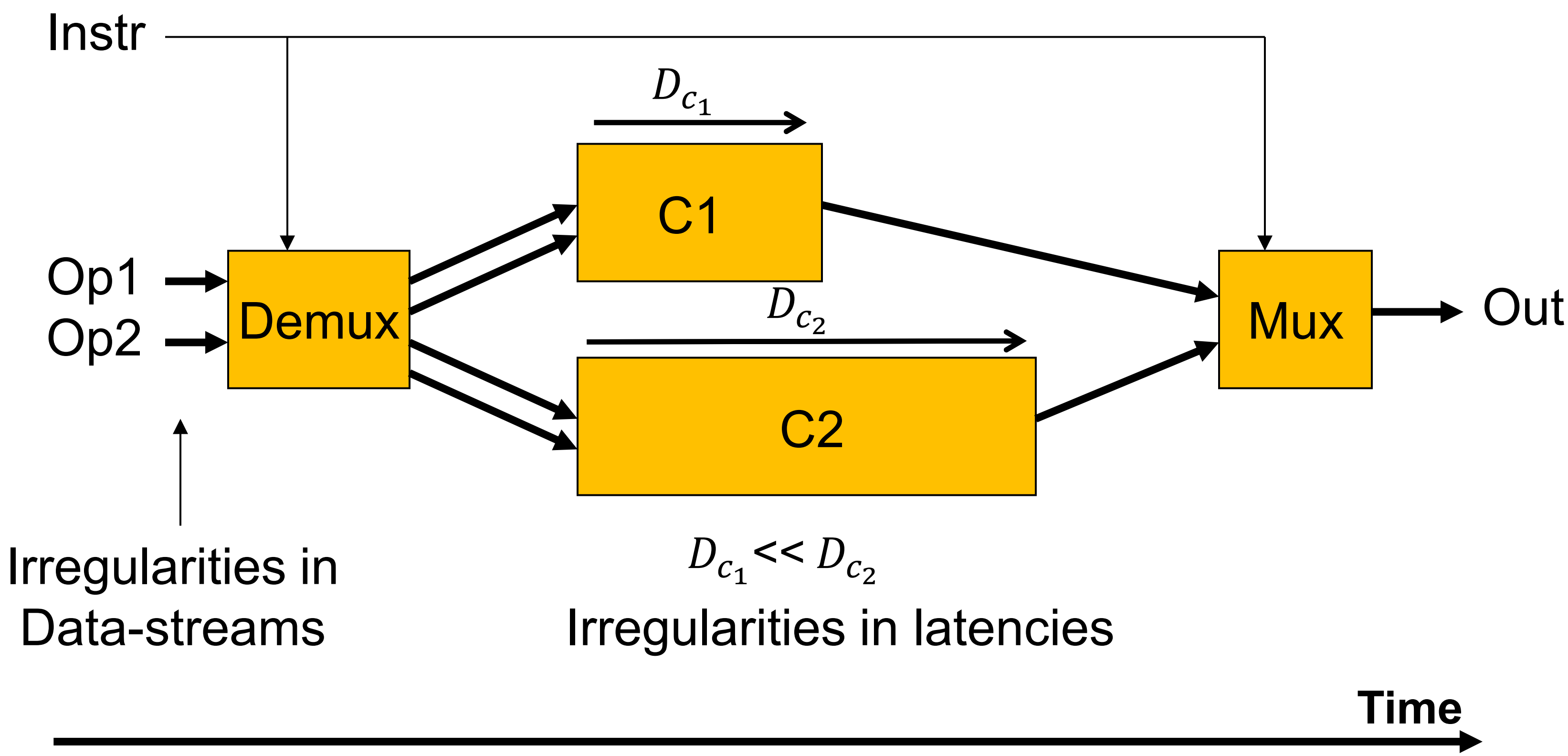


Motivations / expected advantages

- Modularity
- Speed
- Low power
- Electro-Magnetic Compatibility
- Robustness
- Security

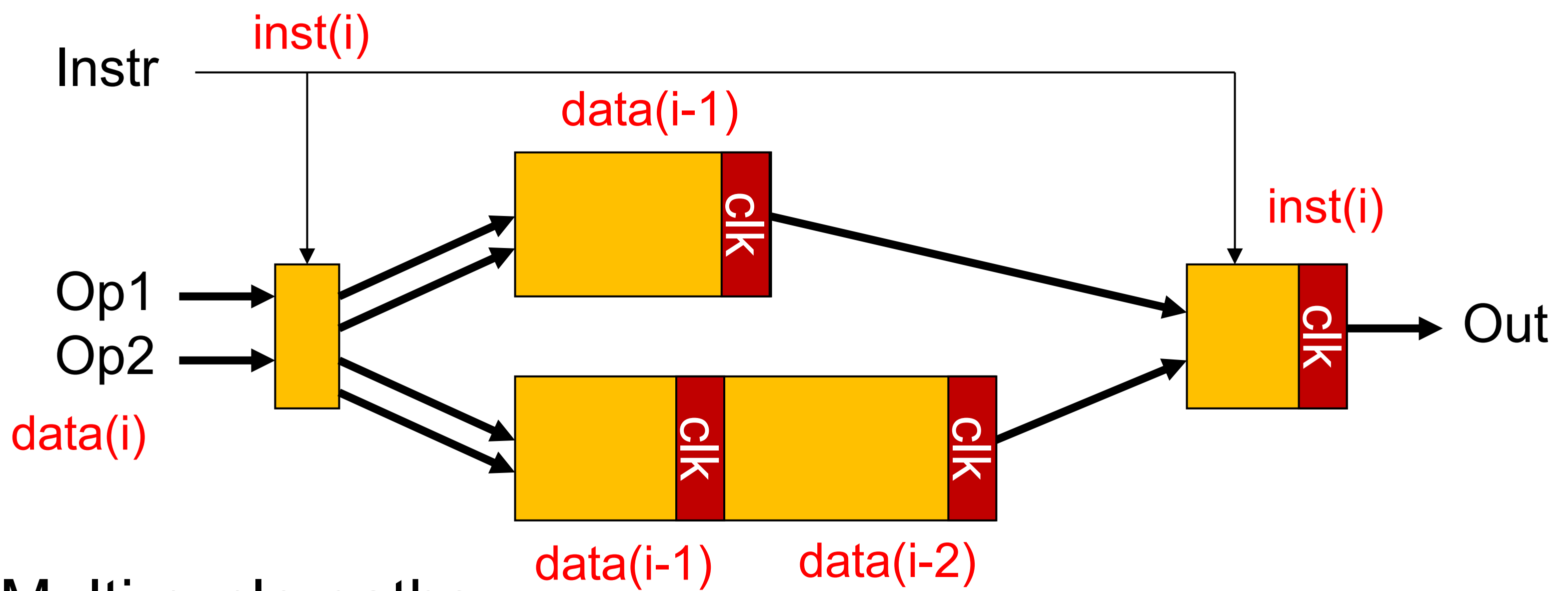
Asynchronous Circuits Principles

- A Non-linear pipeline



Asynchronous Circuits Principles

- Synchronous circuits: pipelines taking into account data flow irregularities

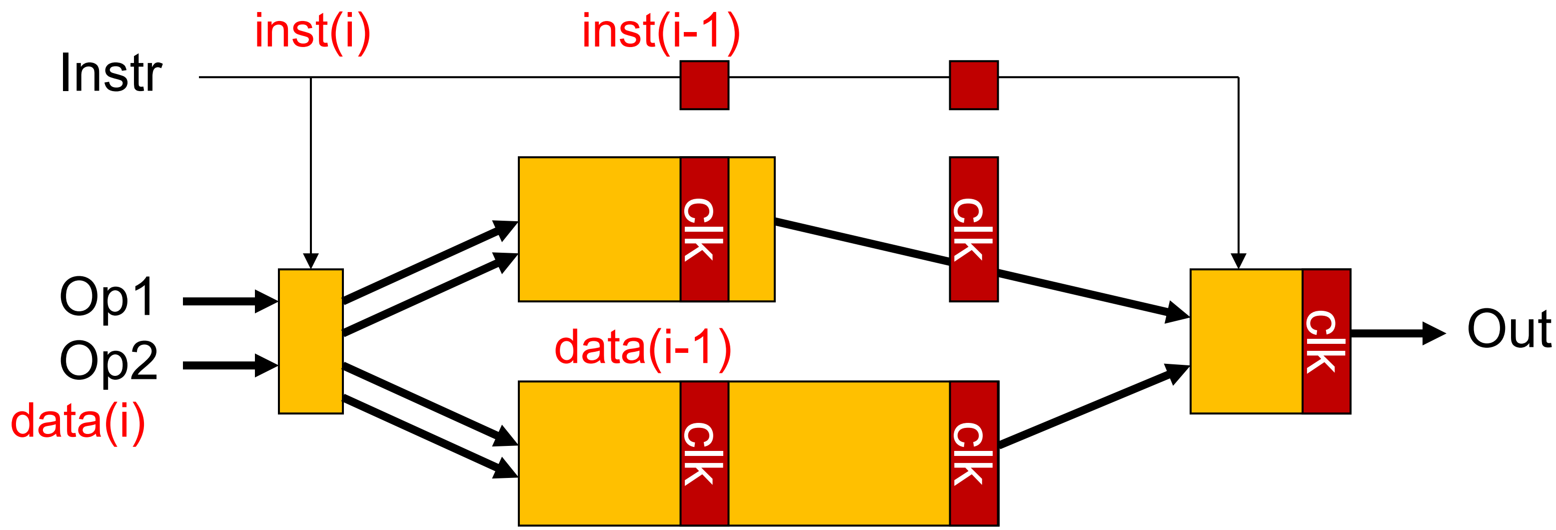


- Multi-cycle paths
- Need to know the state at each cycle (FSM)

Time

Asynchronous Circuits Principles

- Synchronous circuits: balance the pipelines (Worst Case)

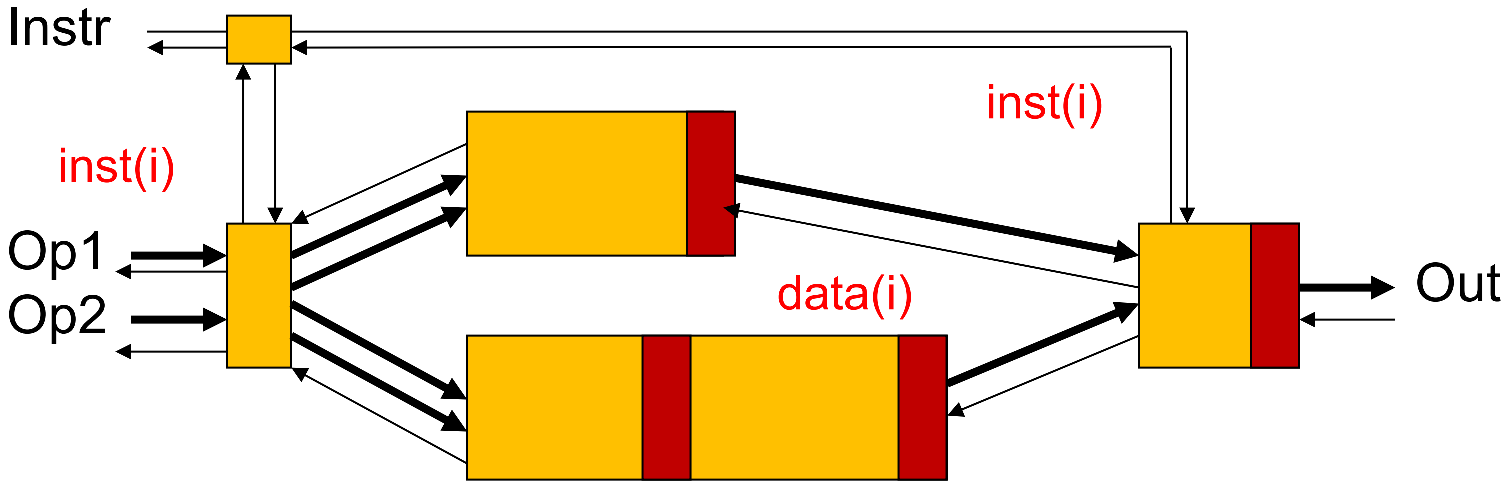


- Difficult to exploit input data stream irregularities
- Increased latency
- increased power consumption

Time

Asynchronous Circuits Principles

- Asynchronous circuits: manage irregular data flows



- No need to know the state
- Local synchronizations preserve the functional correctness
- Latency is minimum, as well as power consumption
- Simple to compose a complex system

Time

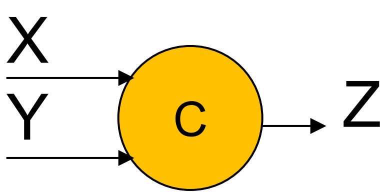
Asynchronous circuit design principles

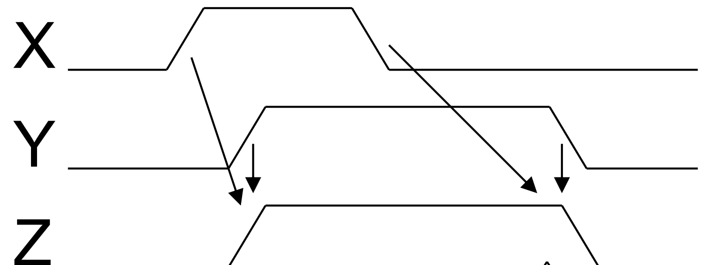
The **C-Element** or **Muller** gate

A, B, … and C!

D. E. Muller

Symbol

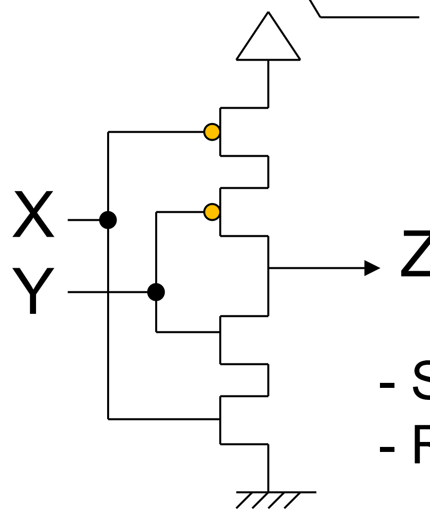




Truth table

X	Y	Z
0	0	0
0	1	Z^{-1}
1	0	Z^{-1}
1	1	1

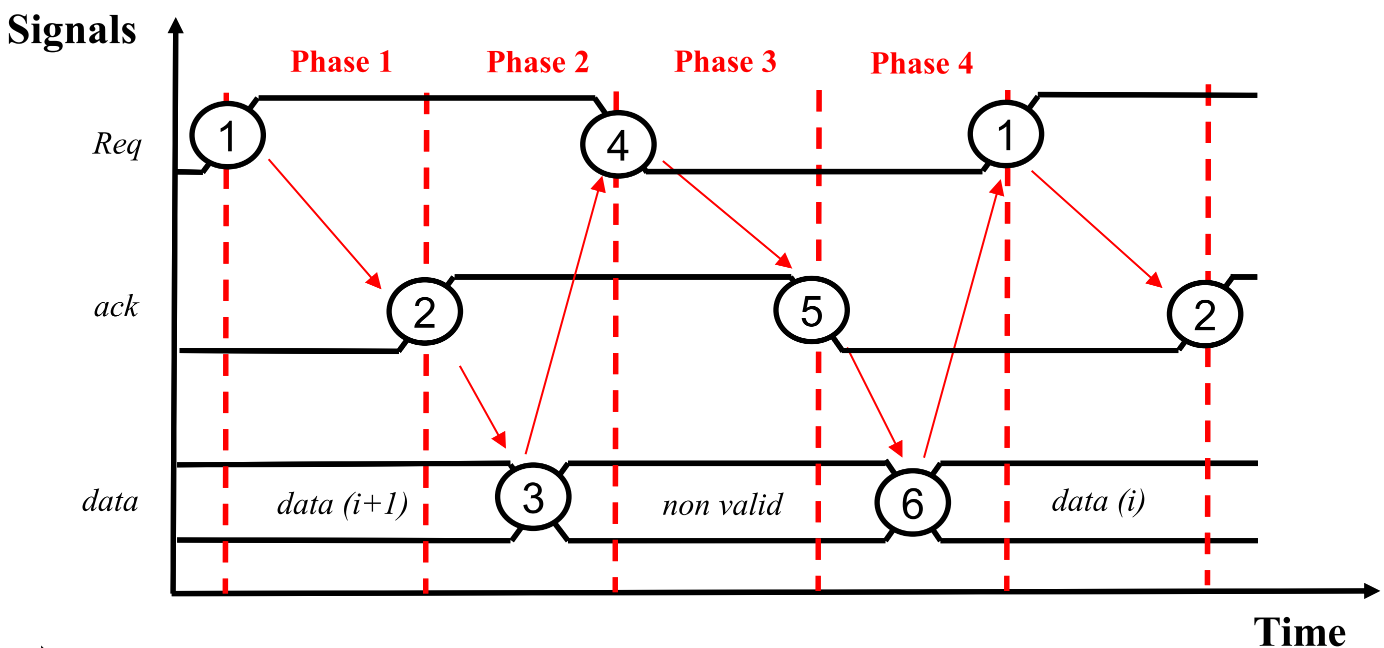
Z = XY + Z(X+Y)



- State holding
- Reset

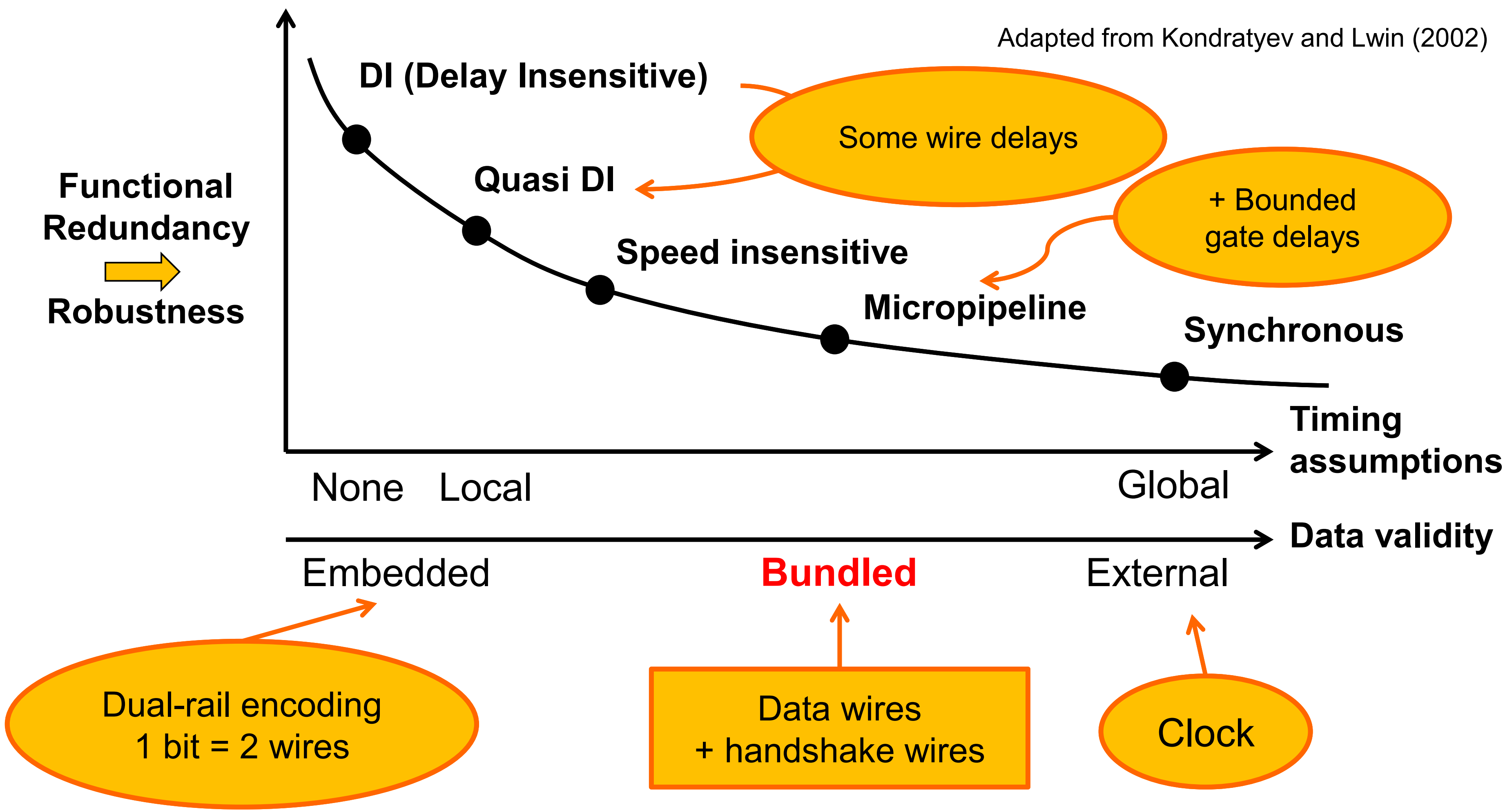
Asynchronous circuit design principles

- 2-phase protocol (NRZ)
- 4-phase protocol (RZ)

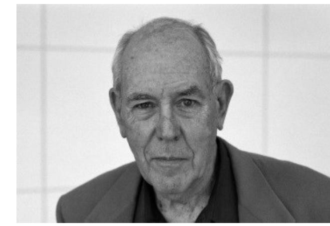


→ Several derivatives exist

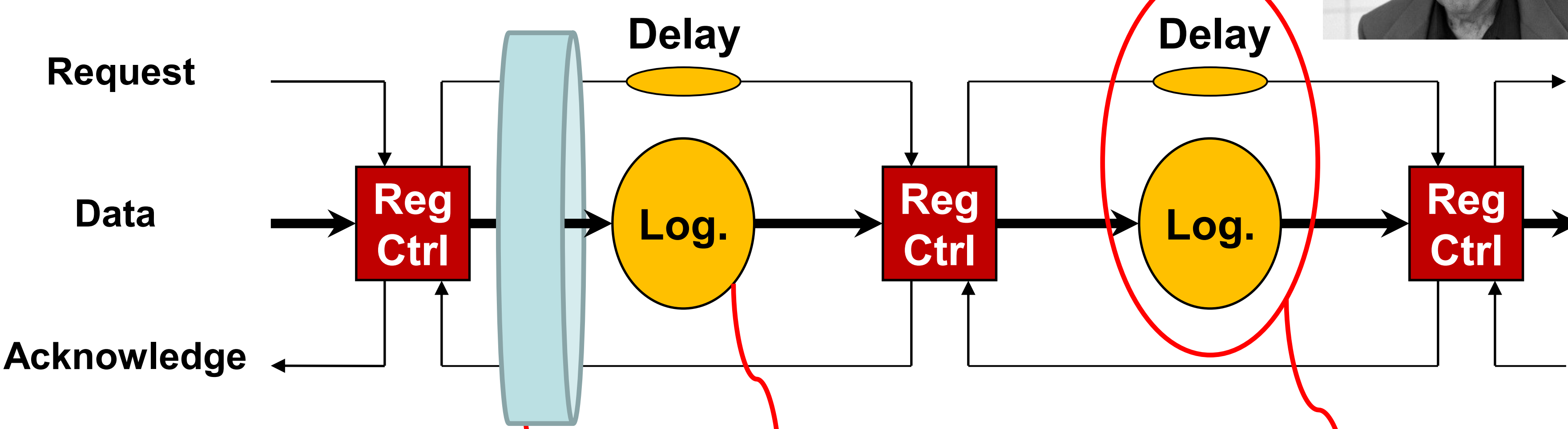
Asynchronous circuit classes



Micropipeline circuits



- Micropipeline (I. sutherland)

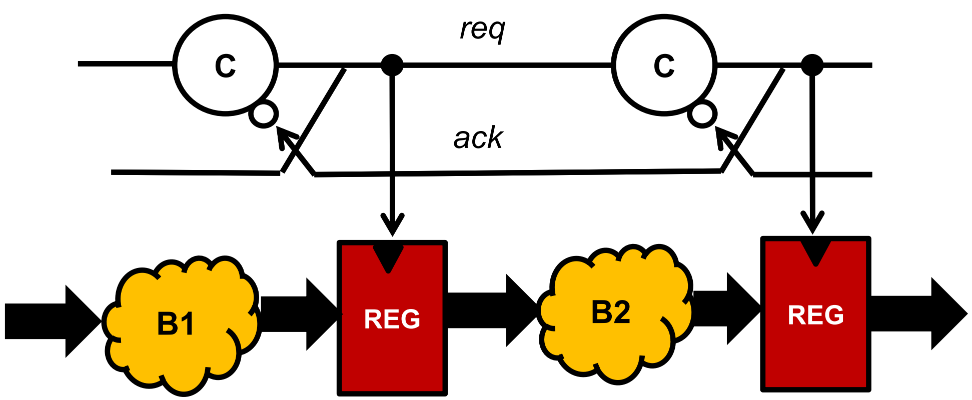


- Combinatorial logic is simple
- Communication channels (handshake based)
- Locally worst case approach

→ **Local timing assumptions**

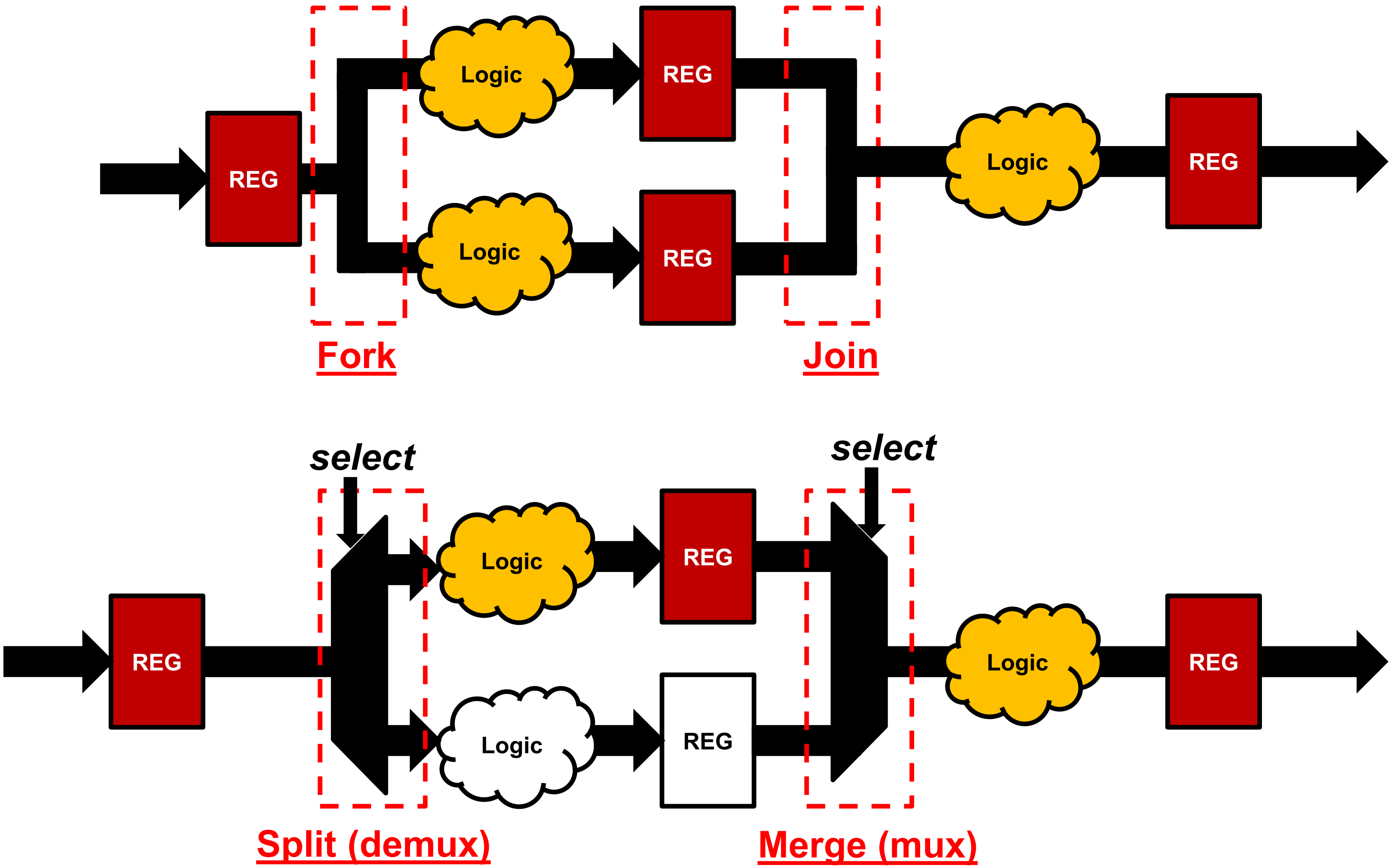
Micropipeline circuits

- **<u>Protocol:</u>** 2-phase or 4-phase
- **<u>Storage:</u>** Flip-flops, master/slave latches or event-driven registers
- **<u>Synchronization:</u>** DI controller



- **<u>Linear pipeline:</u>** Muller gate with an inverted input

Micropipeline circuits



Asynchronous circuit classes

- **Conclusion**
  - Micropipeline : Standard data-path + DI Controllers
  - More robust circuits
  - **Data-driven circuits**

The circuits only consume when data are processed
(no consumption without data)

- **Perspectives**
  - Reduce the data flow to mitigate power consumption
  - **Differently (smartly) sample the data!**

Outline

- **Event-driven circuits**
- **Non-uniform sampling**
- **Sampling in matrix sensors**
- **Conclusion**

Sampling is the success key

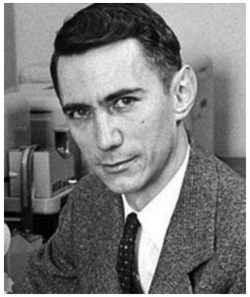
- Sampling based on the Shannon-Nyquist theorem
  - Efficient and general theory… **whatever the signals!**
- Smart sampling techniques
  - **More efficient** but less general approaches… for **specific signals**!

    → Power
    → Speed
    → EM
    → Safety, …

  - Need a more general mathematical framework
    - **F. Beutler**, “*Sampling Theorems and Bases in a Hilbert Space*”, Information and Control, vol.4, 97-117,1961
- Sampling should be **specific to signals and applications**

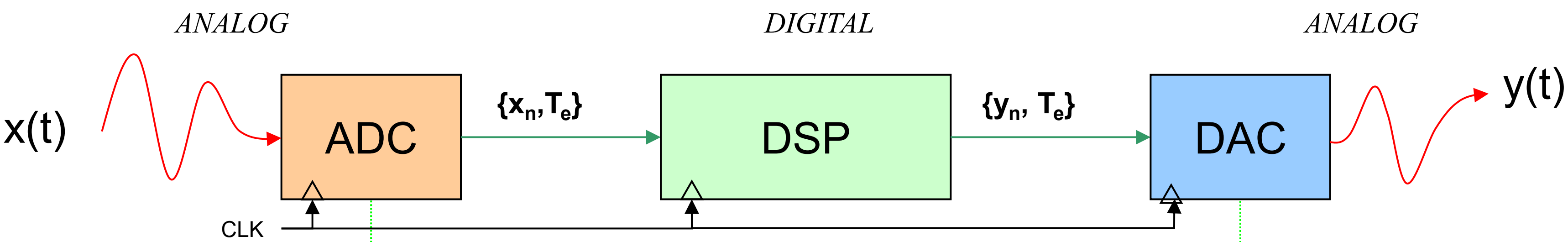
Frederick J. Beutler

What can we do?



Claude Shannon

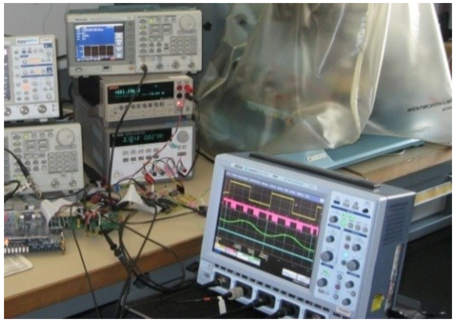
Uniform and Synchronous



Non Uniform and Event-driven (Asynchronous)

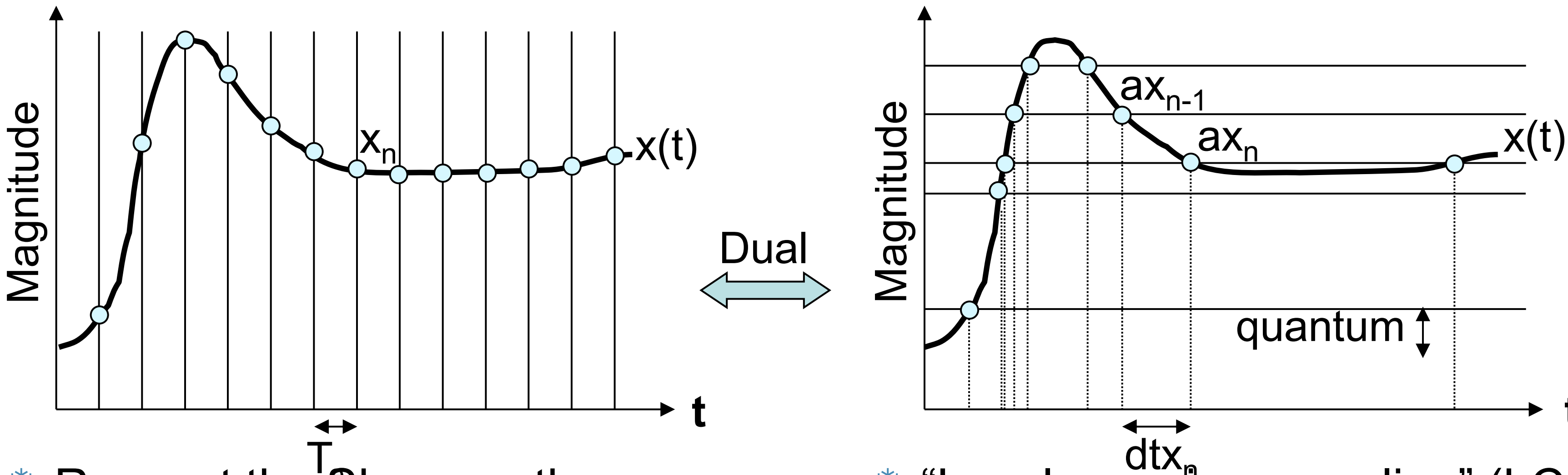
What to expect?

- **Activity reduction** for many signals
  - 1 to 2 orders of magnitude)
- **Signal-dependent sampling** technique
- Dynamic activity selection (impossible if synchronous)
- **Direct processing of the non-uniform samples**



Differently sampling

Uniform sampling | **Non uniform sampling**



- Respect the Shannon theorem
- **Instants exactly known**
- Information: T_{sample}, { i_k }
- In an ADC: **Amplitude quantization**
- Many useless samples

- "Level-crossing sampling" (LCSS)
- **Amplitudes exactly known**
- Information: quanta, { dti_k }
- In an A-ADC: **Time quantization**
- Only useful samples

SNR with non-uniform sampling

- SNR for a sinusoid:

$$SNR_{dB} = 1{,}76 + 6{,}02.N$$ ⇨ **Number of bits**

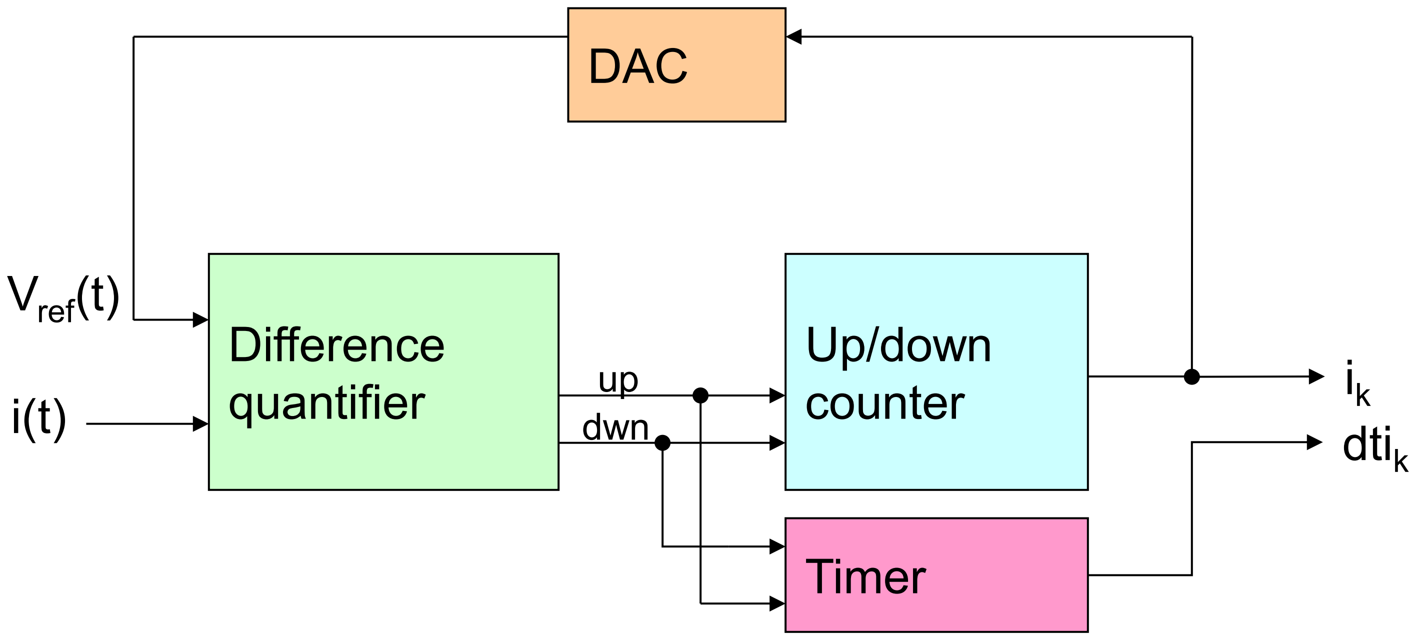
- Theoretically, noise is only due to amplitude quantization
- With non-uniform sampling, the time is quantized

$$SNR_{dB} = -11{,}2 - 20\log(fT_c)$$ ⇨ **Timer period**

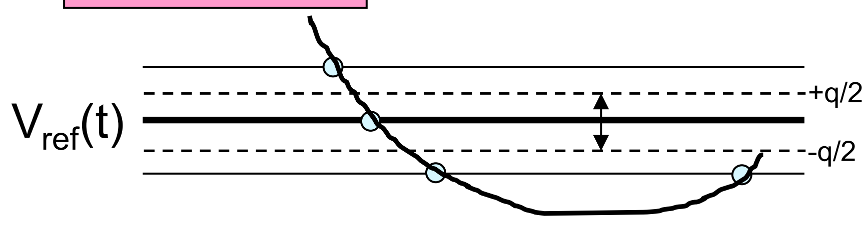
- Noise only depends on the **timer resolution** whatever the threshold distribution

A-ADC or ADC for LCSS

- A-ADC for Non Uniform Sampling



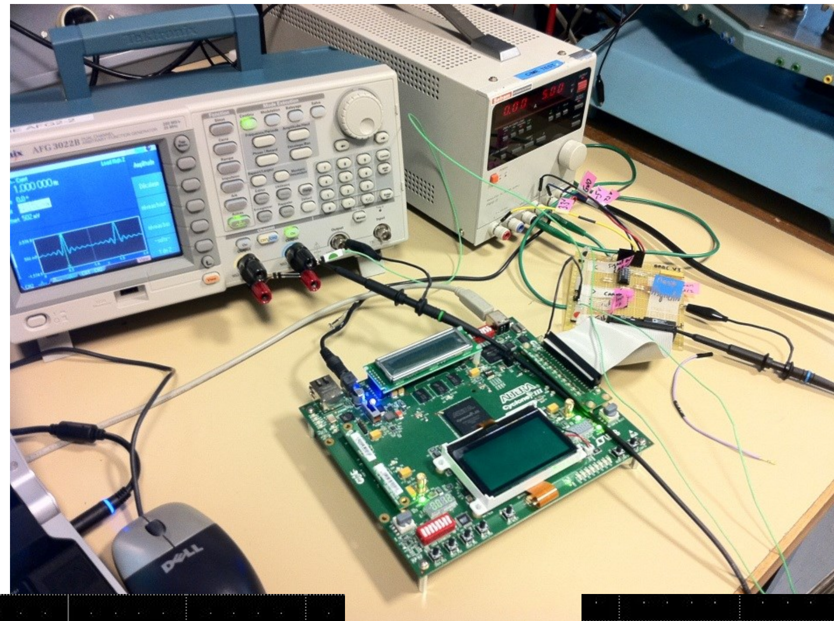
- If (i(t) – Vref(t)) > q/2 → up
- If (i(t) – Vref(t)) < -q/2 → dwn
- Else up = dwn = 0



A-ADC

Aeschlimann et al., 06

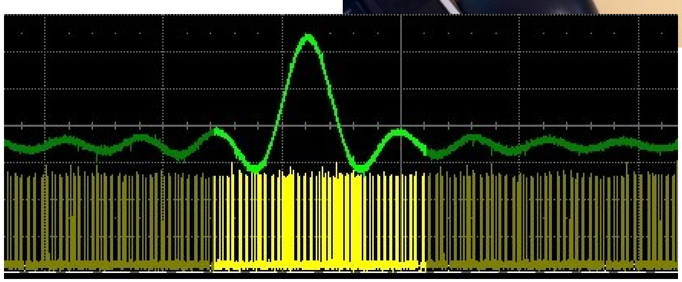
Beyrouthy et al., 11



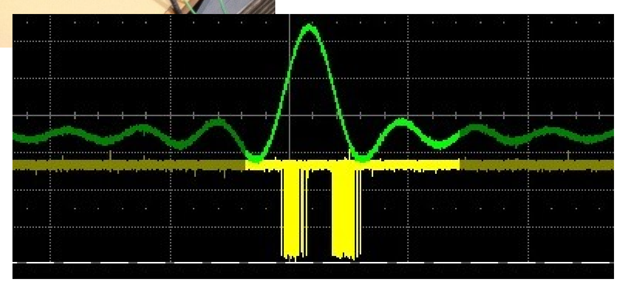
A-ADC

realized with circuits on-the-shelf.

(FPGA+DAC+Comparators)



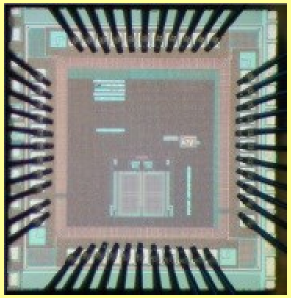
Synchronous ADC



Asynchronous ADC

A-ADC testchips

Microphotography of the A-ADC
In CMOS 130 nm technology from
STMicroelectronics



E. Allier et al., Async, 2003

A level-crossing flash asynchronous analog-to-digital converter

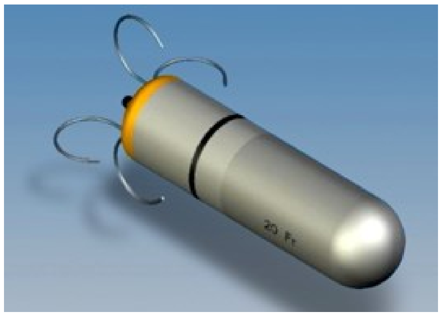
F. Akopyan et al., Async, 2006

A Clockless ADC/DSP/DAC System with Activity-Dependent Power Dissipation and No Aliasing

Y. Tsividis et al., ISSCC, 2008

Lowering in one step the **storage**, the **processing**, the **communications** and the **power consumption!**

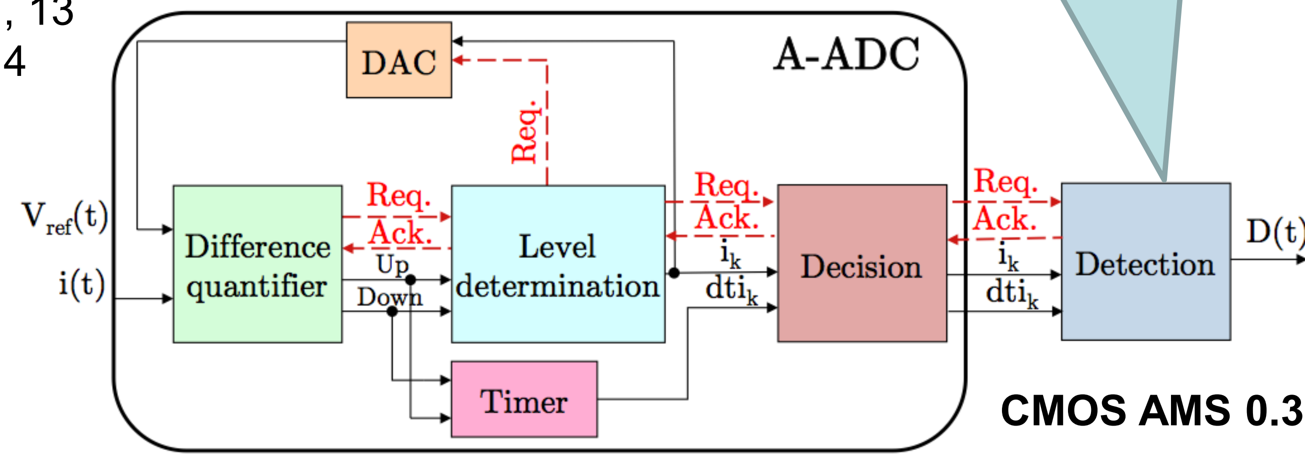
A successful experiment



- Context of the medical implants
- Activity patient measurements

T. Le Pelleter et al., 13
L. Fesquet et al., 14
E. Allier et al. 05

Event-driven block (Asynchronous logic)



CMOS AMS 0.35 µm

Experiments based on **real physiological signals** (recorded on the patients)

What we learned
with the experiment

- **Be more specific** to signals and applications
- **Non-general approach**, but **reproducible**

With the medical

- No pre-processing
- **Less than 1% of data** compared to the uniform sampling
- **3 orders of magnitude reduction** on power

- **Non-uniform sampling well adapted to sporadic signals**

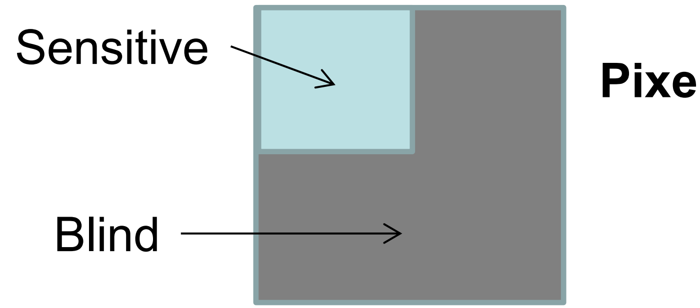
Outline

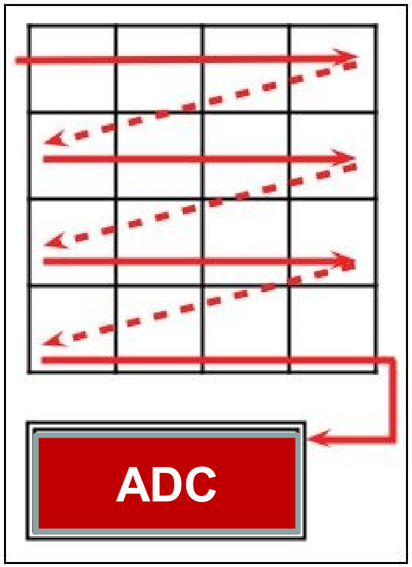
- Event-driven circuits
- Non-uniform sampling
- Sampling in matrix sensors
- Conclusion

Image sensor principles

- Based on **photodiodes**
- Pixel fill factor = optical quality
- **All pixels are read in sequence**
- Larger the sensor, higher the throughput (fixed frame rate)
- Higher the throughput, higher the ADC consumption

- The **ADC is the first contributor of power consumption**





Changing the paradigm in a realistic manner

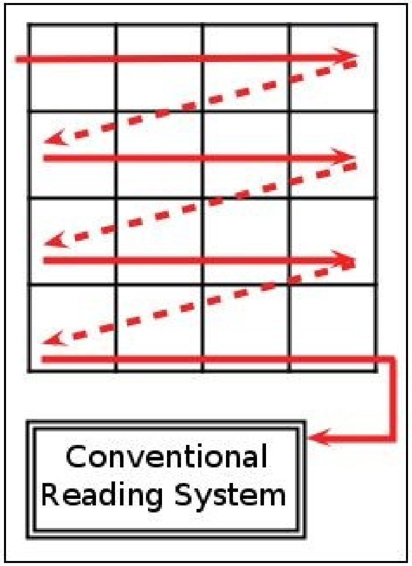
- Keep the fill factor reasonable
- **Reduce the throughput** without changing the frame rate
- **Remove the ADC to limit power** consumption
- Replace it by a digital circuit (more easy to implement)

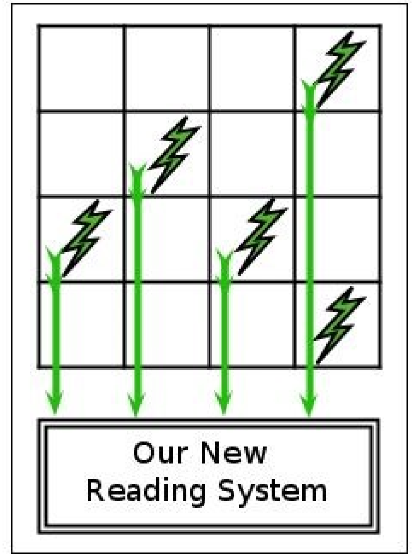
⇒ **Suppress redundancies**

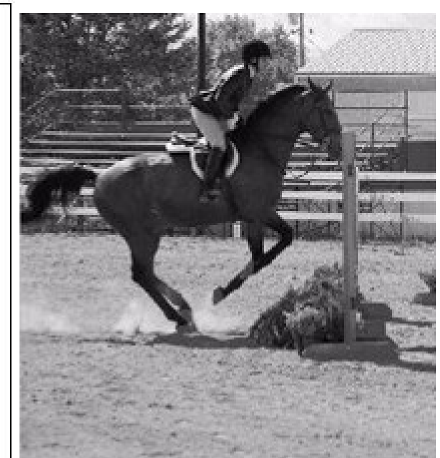
⇒ **Use Event-Driven logic (asynchronous)**

⇒ **Prefer Time-to-Digital conversion**

Towards an event-driven ADC in 2D







C. Posch

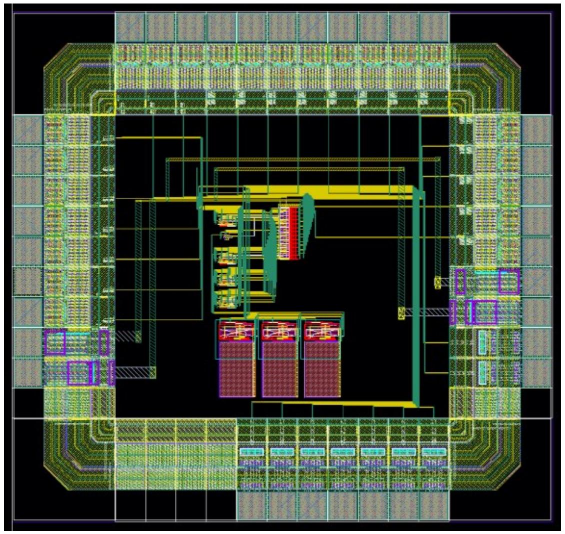
T. Delbrück

- Fully sequential reading
- High Throughput (worst case)
- Need of data compression

- Event-based reading
- Low Throughput
- Management of spatio-temporal redundancies

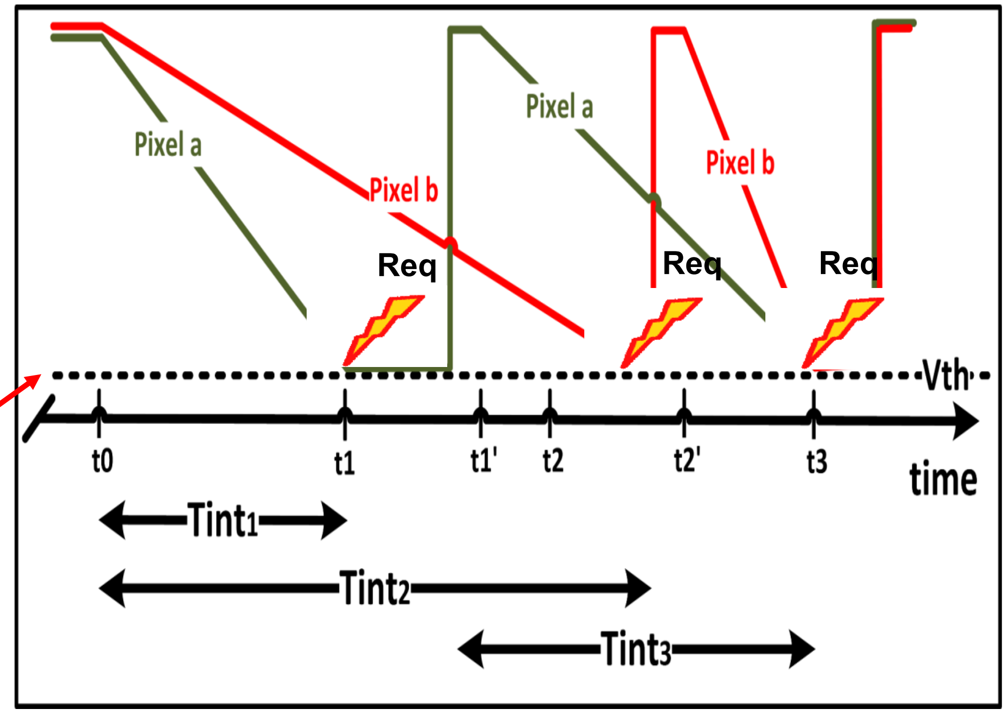
A. Darwish

Event-based Pixel



Pixel Testchip

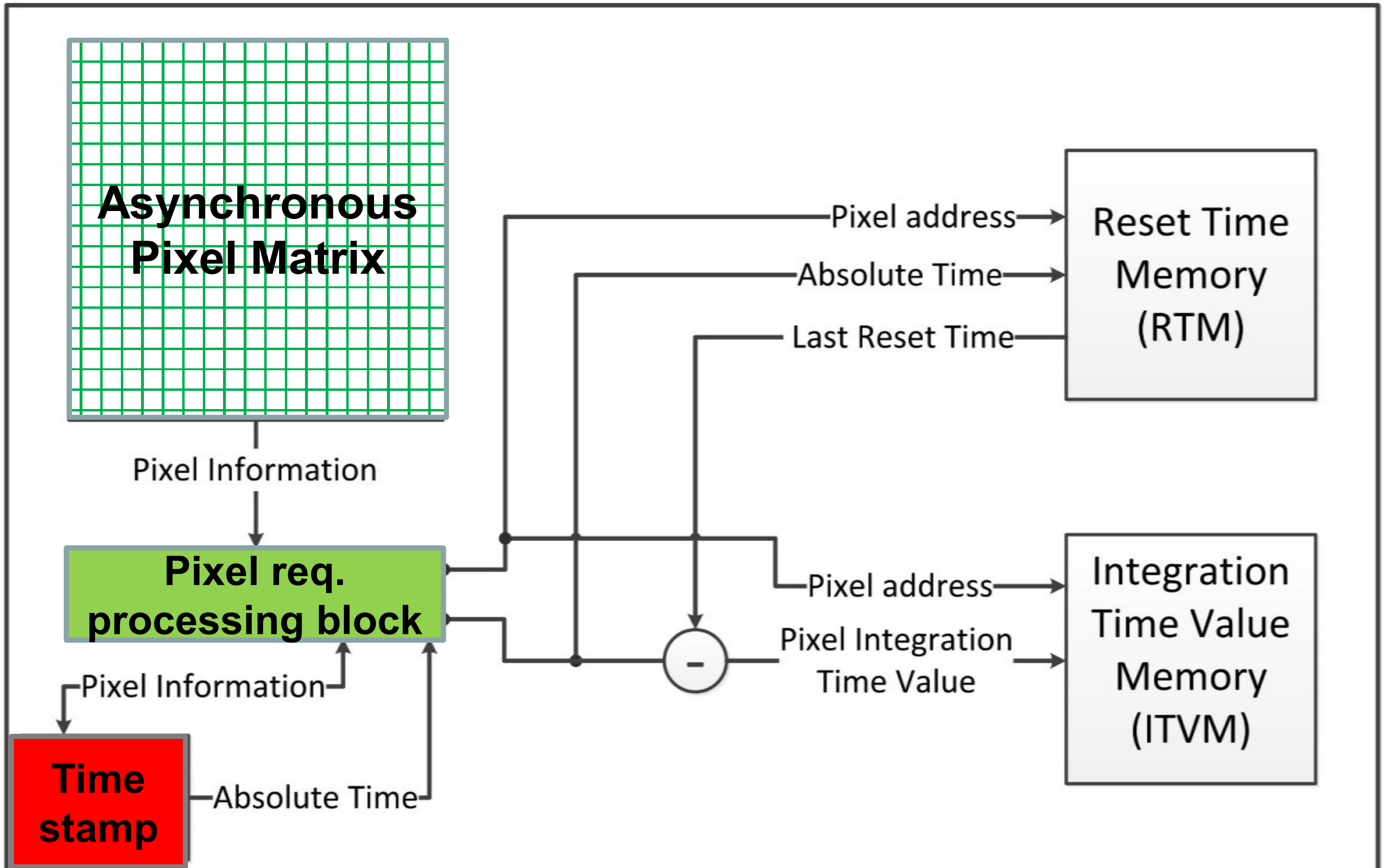
Adjust the image sensor sensitivity



- Based on **event-detection**
- **1-level crossing sampling** scheme
- **Unique integration time per pixel**
- **Time to first spike** encoding

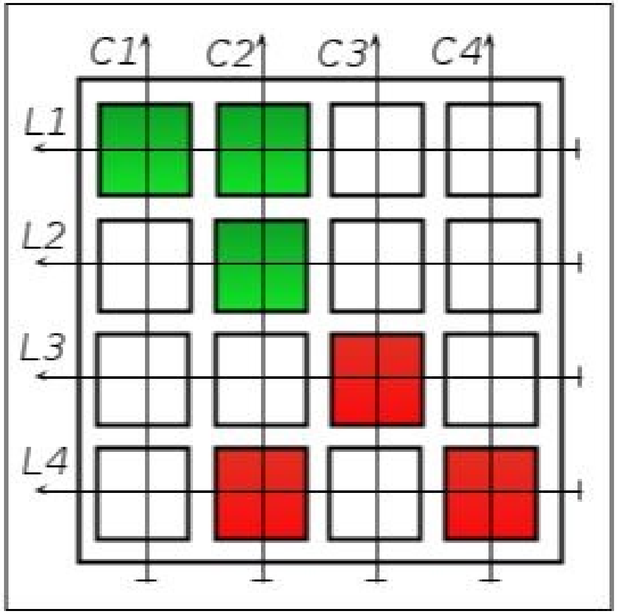
A. Darwish et al., EBCCSP, 2015
A. Darwish et al., NewCAS, 2014

Asynchronous readout architecture

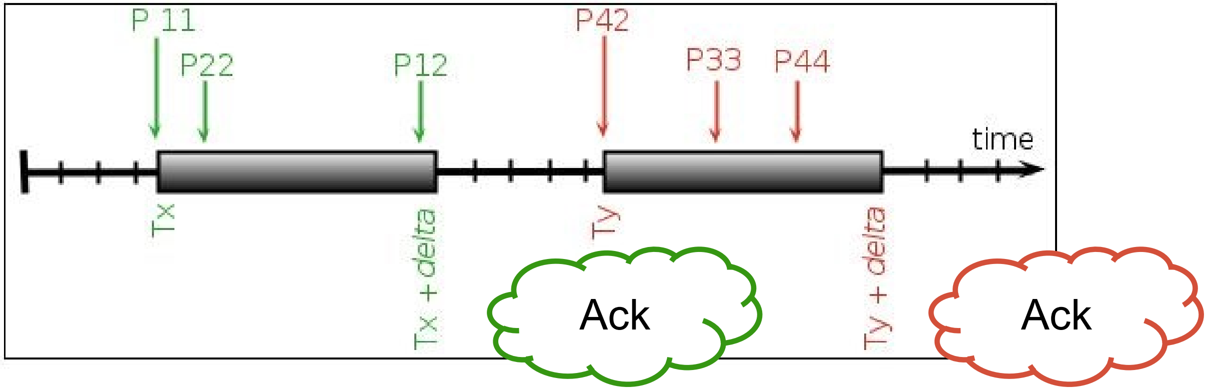


How do we suppress Spatial Redundancy ?

4 x 4 image sensor



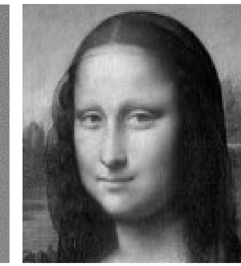
Darwish et al., 14



Simulation results

SSIM: Structural Similarity
PSNR: Peak-Signal-to-Noise Ratio



- **High PSNR and SSIM Values**
- **Low data flow rate**

Picture Sample	1	2	3	4
SSIM	0.869	0.943	0.925	0.978
PSNR	43.23 dB	41.97 dB	42.98 dB	43.22 dB
% of the original data flow	**15.5 %**	**4.23 %**	**0.47 %**	**3.88 %**

What we learned

with image sensors

- **1-level crossing sampling** in 2D
- Low percentage of readings per column (< 6 %)
- **Drastic data flow reduction**
- **Event-driven** digital circuitry
- **Adjustable resolution** and **dynamic range**

- **Don't need an ADC** (power consuming)
- **Intrinsic A-to-D conversion**

Outline

- **Event-driven circuits**
- **Non-uniform sampling**
- **Sampling in matrix sensors**
- **Conclusion**

General Conclusion

- **Sensing and processing data are power consuming**
- **Sensing and processing must be thought as a whole**

Suggested approach:

- Lesson 1: **Determine** the most efficient sampling!
- Lesson 2: Fit well **Event-Driven Circuits** (asynchronous)
- Lesson 3: **Ultra-Low Power**

Don't forget

- Less samples means:
  - Less computation, less storage, less communications,
  - **less power**

An energy efficient approach of digital processing

General Conclusion

Sampling is **signal-** and **application-dependent**

→ Only keep the useful samples!

Processing is sampling-dependent

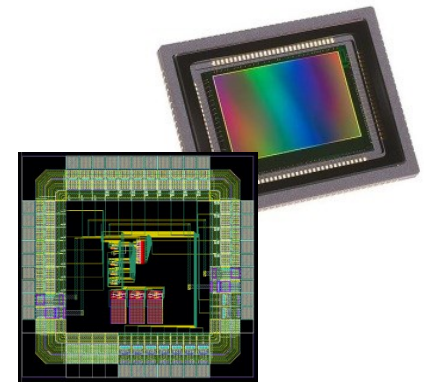


Image Sensors

Image sensors benefit from event-driven approach

Useful for **Smart Image Sensors** and **Matrixed MEMS**

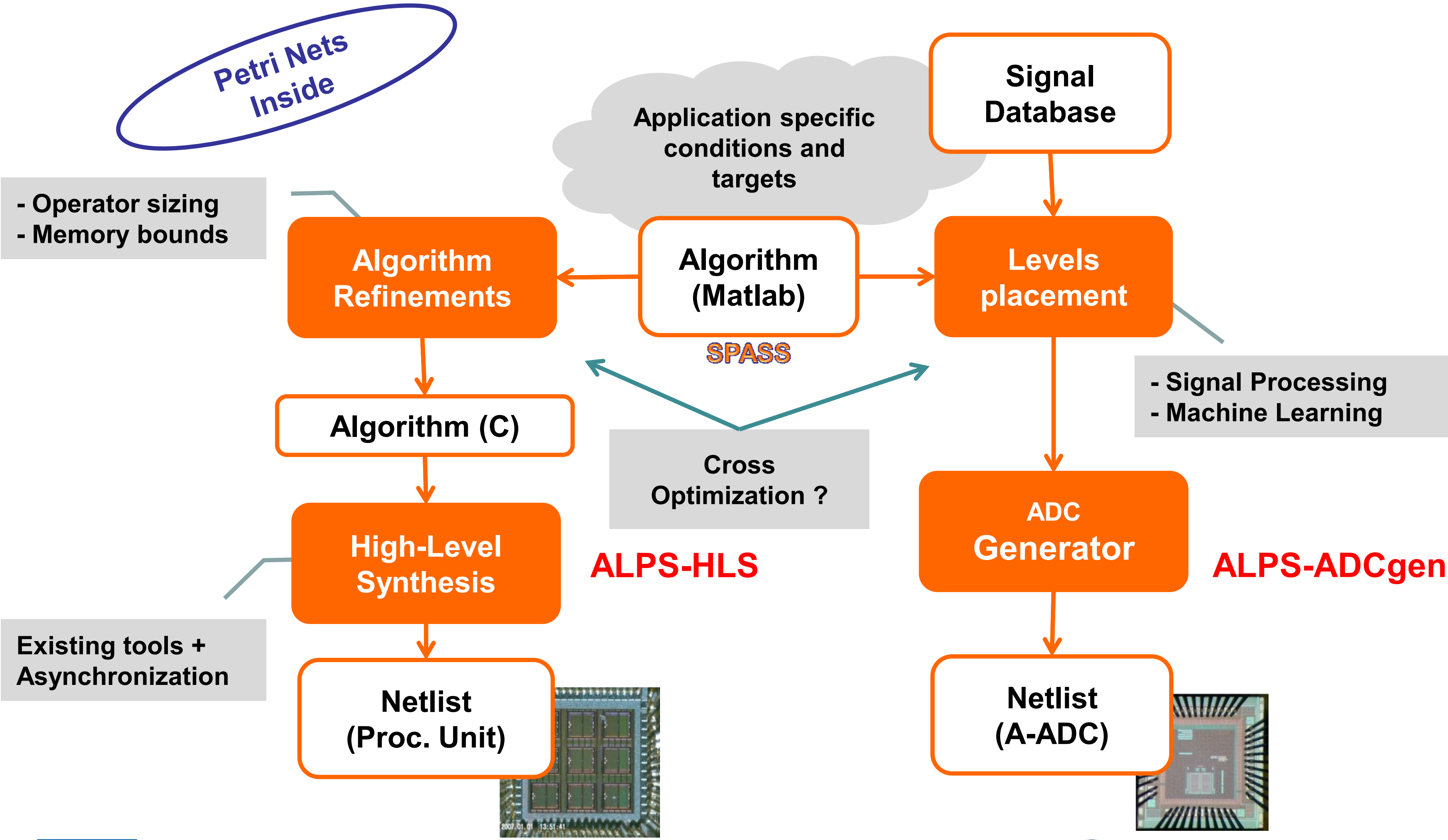
→ Vision and Robotics

- Need to rethink the image processing (non-conventional data flow)

Where we go …

ALPS Asynchronous Low-Power Synthesis

The Ultra-Low Power Design Flow



Non-uniform sampling is the future of digital universe!

Thanks for your attention