ZOMBIE
SURVIVAL PLAN

STEP 1
GATHER WEAPONS

STEP 2
SECURE TRANSPORTATION

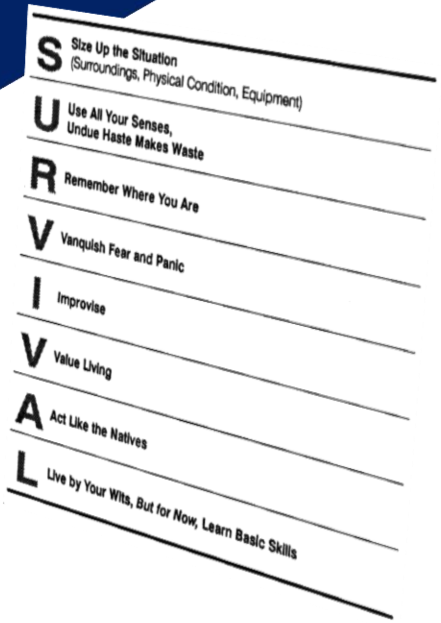
STEP 3
LOCATE FRIENDS AND FAMILY

STEP 4
CURE THE INFECTION
ONE BULLET AT A TIME

If you build a plan that requires a thirty step sequence to implement, it is guaranteed to fail. Keep it simple!

Developing survival instincts in computing systems

Alex Yakovlev
Newcastle University
UK



S Size Up the Situation (Surroundings, Physical Condition, Equipment)

U Use All Your Senses, Undue Haste Makes Waste

R Remember Where You Are

V Vanquish Fear and Panic

I Improvise

V Value Living

A Act Like the Natives

L Live by Your Wits, But for Now, Learn Basic Skills

Acknowledgement to members of the MSD group: Fei Xia, Delong Shang, Danil Sokolov, Andrey Mokhov, Xuefu Zhang, Abdullah Baz, Reza Ramezani, Ra’ed Aldujaily, Nizar Dahir, Ammar Karkar, Ghaith Tarawneh, Ioannis Syranidis and to our colleague Terrence Mak

Outline

- Survival instincts in real life
- “Survival instincts” in computing systems
- Energy-Power modulation
- Instincts and system layers of functionality
- Mechanisms in energy and data processing (reference-free sensors are the key!)
- Mechanisms in communications
- Future developments

Wisdom

- *"The very essence of an instinct is that it is followed* ***independently of reason****."*

  1871 C. Darwin *Descent of Man* I. iii. 100

- *"The operation of instinct is* ***more sure and simple than that of reason****."*

  1781 E. Gibbon *Decline & Fall* (1869) II. xxvi. 10

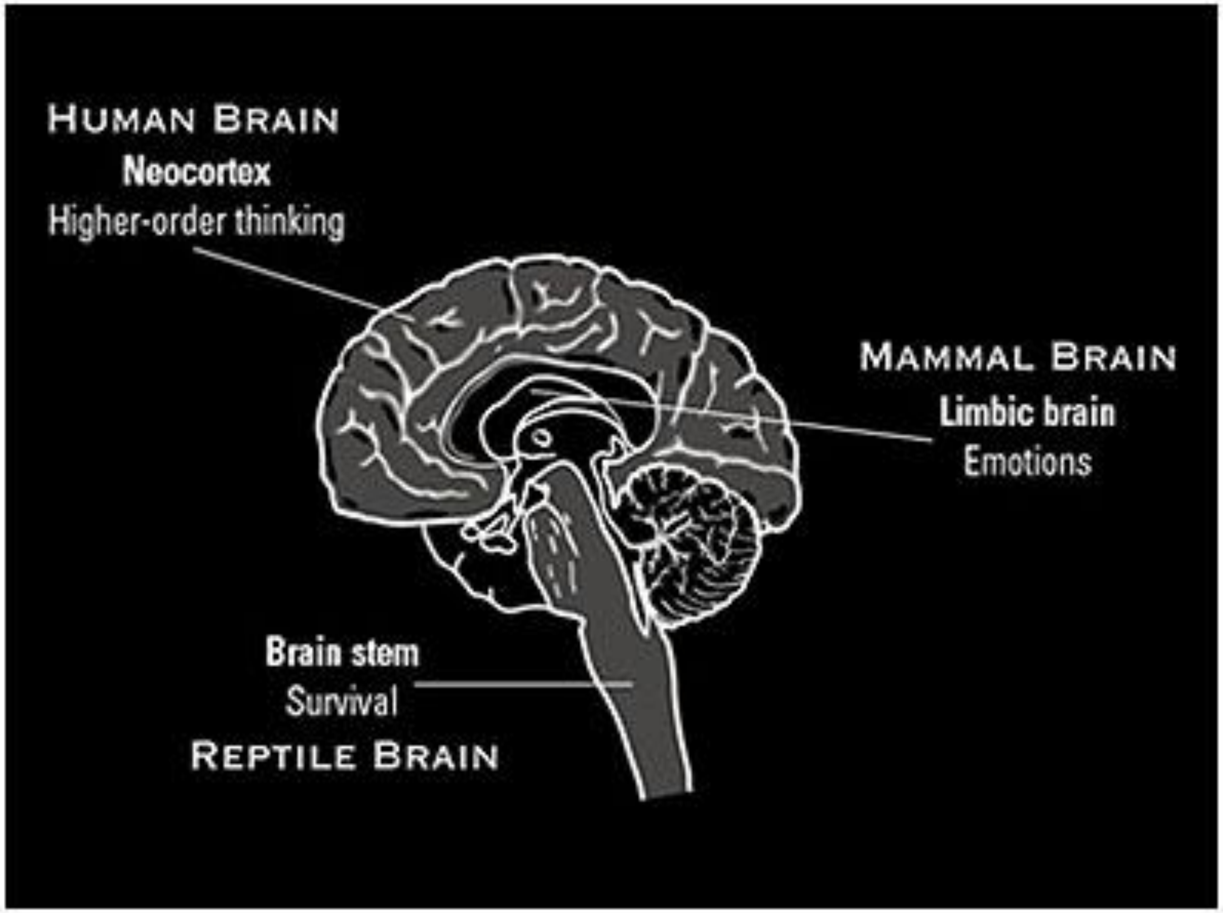
What is survival in general terms?

- Quotes from OED:
  - “**Survival**: The continuing to live after some event; remaining alive, living on”
  - “**Instinct**: (a) An innate propensity in organized beings (esp. in the lower animals), varying with the species, and manifesting itself in acts which appear to be rational, but are performed without conscious design or intentional adaptation of means to ends. Also, the faculty supposed to be involved in this operation (formerly often regarded as a kind of intuitive knowledge). (b) Any faculty acting like animal instinct; intuition; unconscious dexterity or skill”

Survival in general terms

- Video about Jean-Luc Josuat, who got caught in a cave for 5 weeks without food and water:
  - http://videos.howstuffworks.com/discovery/6835-human-body-built-for-survival-video.htm
  - First his reaction was to actively search for food - due to orexin, a hormone produced in the hypothalamus, that is generated to trigger alertness and all parts of his body to work faster;
  - But at a later stage, some ‘more hardwired’ instincts (inherited by humans from primitive organisms through evolution) started to prevail in the brain and everything slowed down to ensure survival when energy sources became short
- Surviving from different upsets, disasters and general causes of disruption

Where are survival instincts in brain?



Survival in computing systems

Survival *from* what:

- Faults in the system
  - Defects
  - Aging
  - Transients (inside gates, crosstalk on signal lines, IR drops
  - …
- Upsets outside the system
  - Radiation
  - Power supply
  - Signal distortions
  - …
- Physical effects (mixed internal and external)
  - Temperature fluctuations
  - EMI
  - …

Survival in computing systems

Survival *of* what:

- Structure
- Behaviour
- Specific functionality

Relation between survival and tolerance, resilience, recoverability, longevity, re-production, …?

There are specific aspects of survival when power is variable, intermittent, …

Scale and range of power and energy disruptions

Characterisation of the power profile for the system in space and time

Difference between Survivability and …

- Dependability (Fault-tolerance …)
  - Dependable systems typically want to restore their full functionalities, hence large costs for redundancy; survivability is supposed to be less resource-demanding
- Graceful degradation
  - GD systems typically have a smooth (often quantitative) reduction in their performance, rather than “qualitative” transitions to a more restricted (more critical) set of functionalities as needed for survival
- Other factors: Performability, Quality of Service etc.

“Deep, or Instinct-based, Survival” as opposed to conventional survivability

- Conventional survivability in ICT is more about software systems (cf. Knight and Strunk, *Achieving Critical System Survivability through Software Architectures*, 2004) that make transitions between different services depending on the operating environment
- They do not consider deep, embedded layers of hardware/software that work in proportion to the level of available energy/power resources

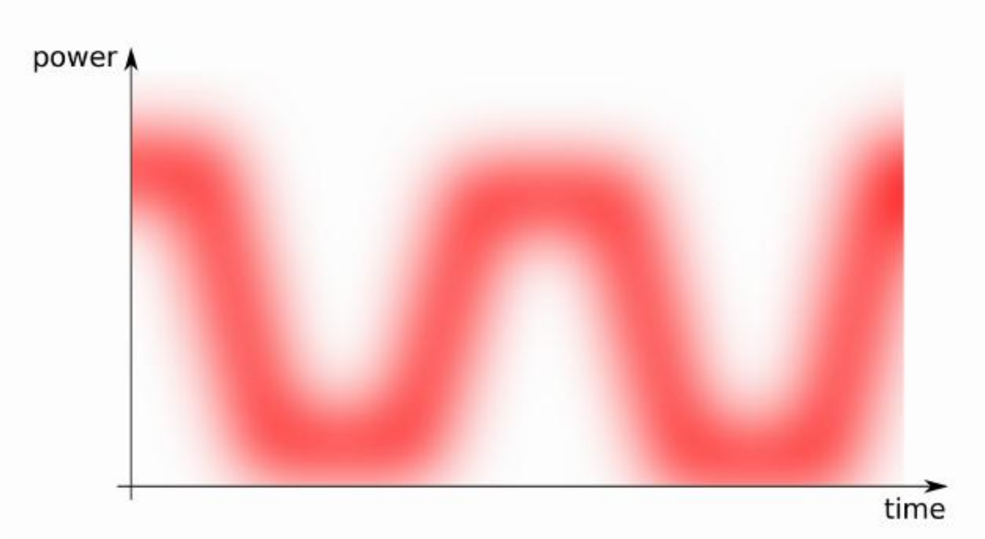
- **Deep survival** is a new concept, inspired by nature, which maintains operation in many structural and behavioural layers, with mechanisms (“instincts”) developed and accumulated in bodies due to biological evolution

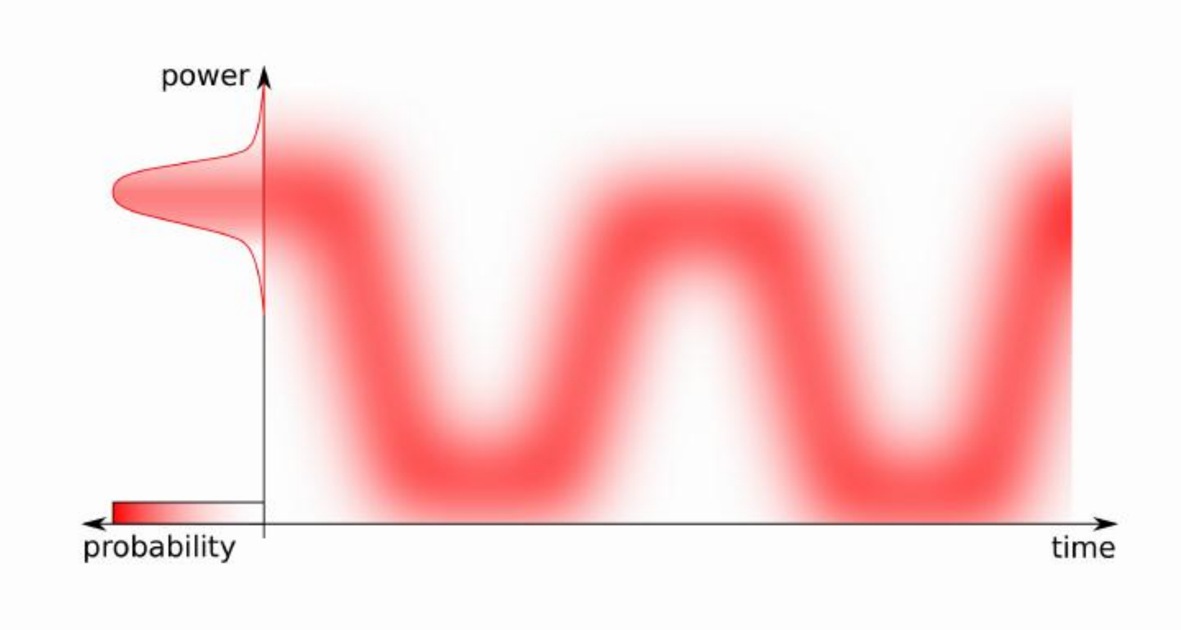
Power/Energy modulation

- The principle of **power/energy-modulated computing** is fundamental for deep survival
- Any piece of electronics becomes active and performs to a certain level of its delivered quality in response to some level of energy and power
- A quantum of energy when applied to a computational device can be converted into a corresponding amount of computation activity
- Depending on their design and implementation systems can produce meaningful activity at different power levels
- As power levels become uncertain we cannot always guarantee completely certain computational activity

Power profile

Global prediction for a part of the system

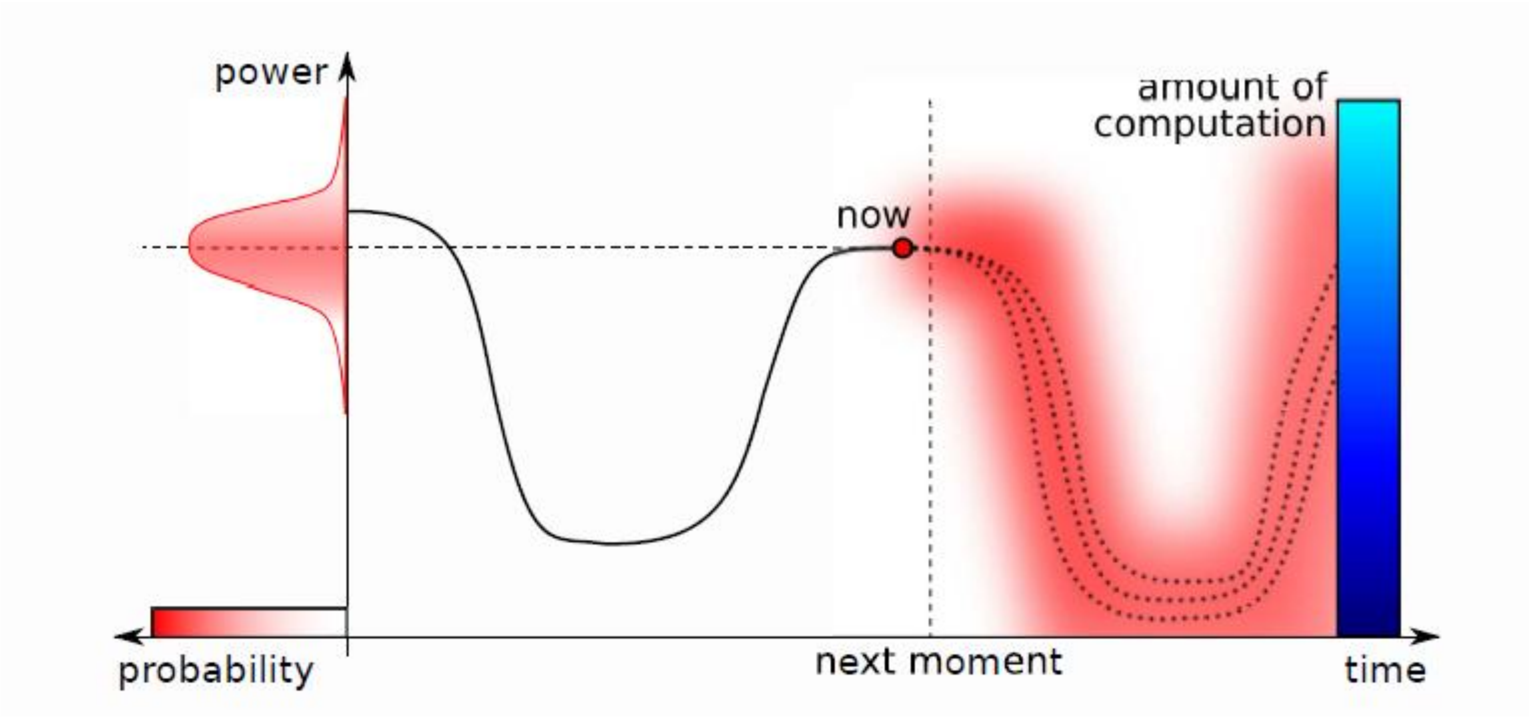




Probability distribution at each time instant

Power-modulation in time

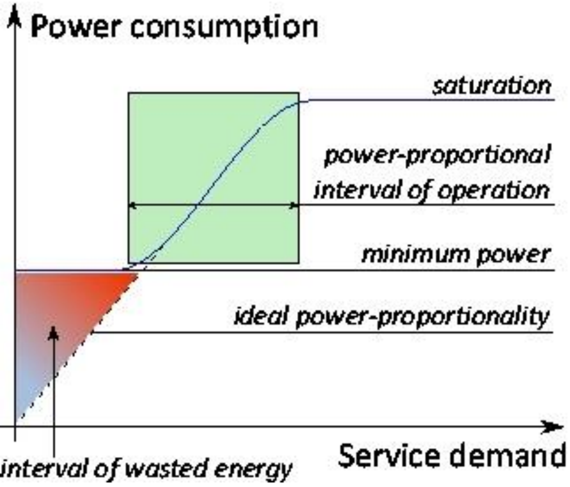
- Localised prediction, from every moment at present
- Power has a certain profile (time trajectory) in the past and uncertain future
- Power-proportional computing …

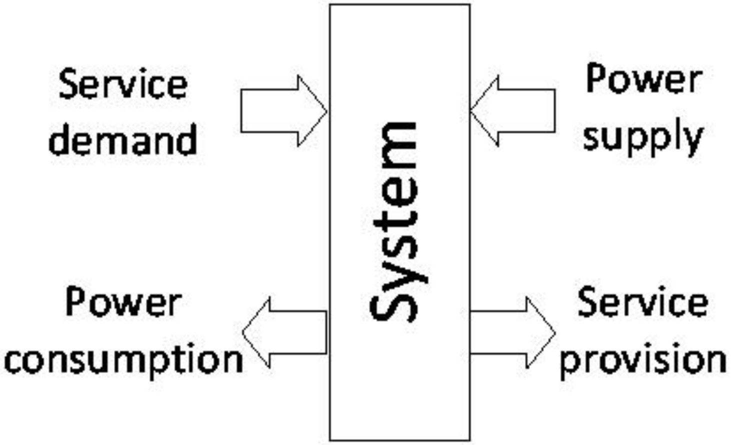


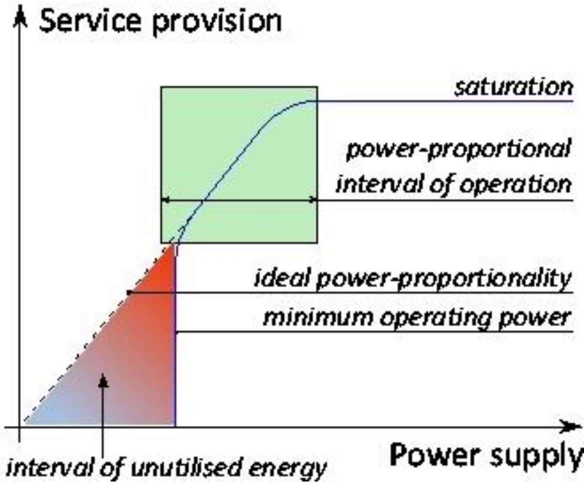
Power proportionality: two views

Energy optimisation for required service demand

Service provision optimisation for constrained power supply







Service-modulated processing

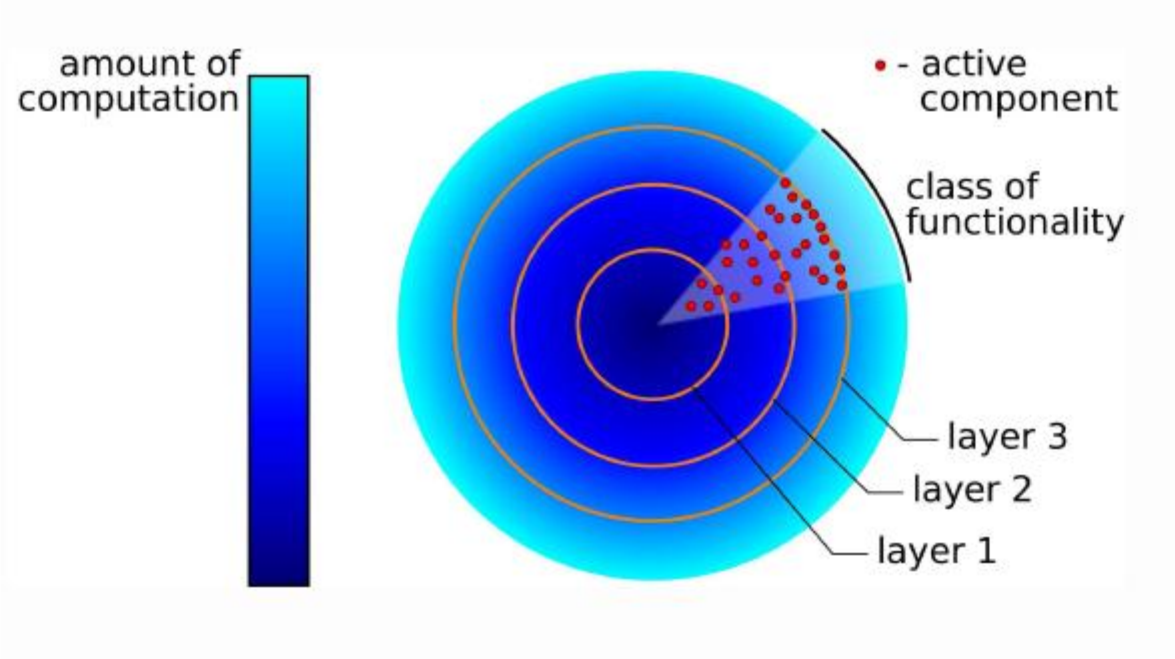
Energy-modulated processing

Power-Energy Modes versus Layers

- When **systems are driven by the service demand** requirements they tend to follow the principle of multi-modality, where the system "consciously" switches between a full functionality mode to a hibernating mode primarily depending on the data processing requirements. Survival aspects here **are limited** to the ability of mode management
- But **what if the power level drops** (externally) .... ?
- To extend the frontier of survivability, system design should also follow the **energy-modulation approach**, and this leads to structuring the system design along **partially or fully independent layers** (cf. Darwin's "*The very essence of an instinct is that it is followed* ***independently of reason****.*")

Power-modulated multi-layer system

- Multiple layers of the system design **can turn on at different power levels** (analogies with living organisms' nervous systems or underwater life, layers of expensive/cheap labour in most of the resilient economies)
- As power goes higher new layers turn on, while **the lower layers ("back up") remain active** – this is where instincts become more in charge!
- The more active layers the system has the more power resourceful and capable of surviving it is



Categories of “instincts”

- The most important is probably energy/power -awareness, i.e. sensing, detection and prediction of power failures
- Storing energy “for the rainy day”
- Retaining key data
- Reactive and optimising mechanisms
- Layers of power-driven functionality
- …

Basic Actions behind Instincts

- ability to accumulate SOME energy, initially and at any time after long interruption, say by charging a passive element
- ability to switch, e.g. generate SOME events
- ability to make a decision, e.g. is there an event or not?

For example, let's take Sensing and examine where these actions are used…

Instincts in Computer Systems

- Mechanisms in energy and data processing domains
  - Reference-free self-sensing and monitoring
  - Elastic memory for survival
  - Elastic power-management for survival
- Mechanisms in communication fabric
  - Monitoring progress in transactions (link level failures, deadlock detection)
  - Power noise and thermal monitoring
  - Non-blocking communications

(SELF) SENSING and CONDITION MONITORING

Reference-free sensing

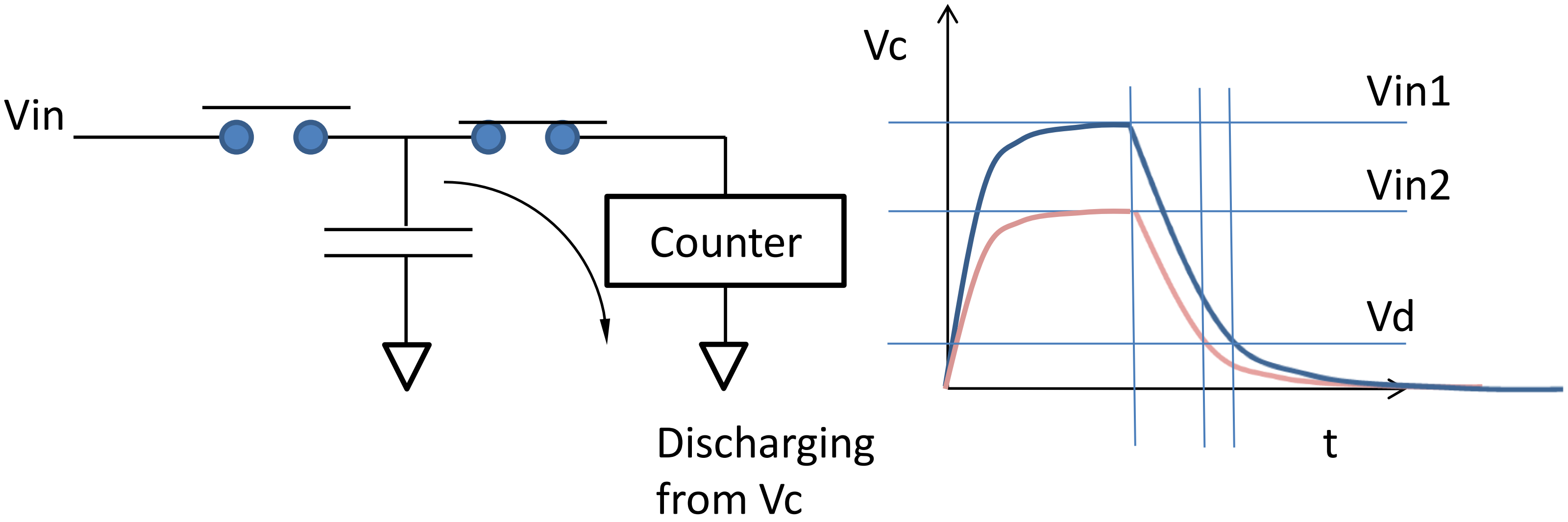
Sensors must work in a changing environment with uncertainty, where constant and reliable references are not available

Possible options:

- Sensing by charge-to-code conversion
- Sensing by differentiators in delays
- Sensing by crossing characteristic mode boundaries
- Sensing by measuring metastability rates

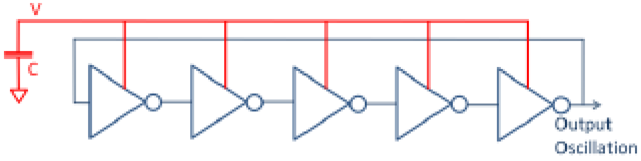
Sensing by charge-to-code conversion

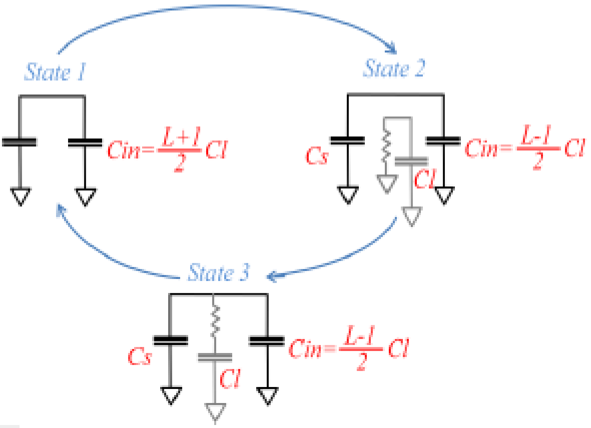
- Some energy is first sampled into a capacitor
- Then discharged through some load registering the quantity of energy (just like in a waterwheel!)



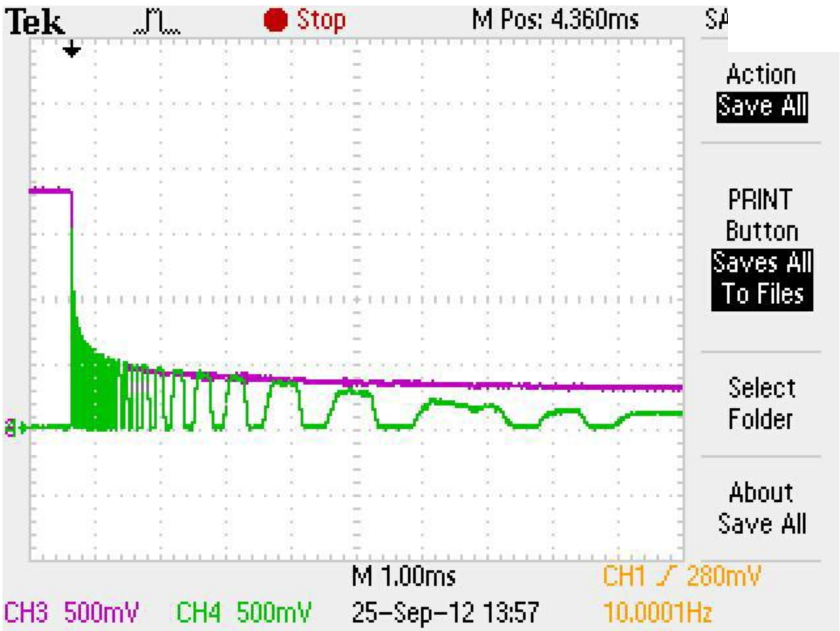
Asynchronous counter works until voltage drops to some low value where it dies. The number it got to encodes Vin.

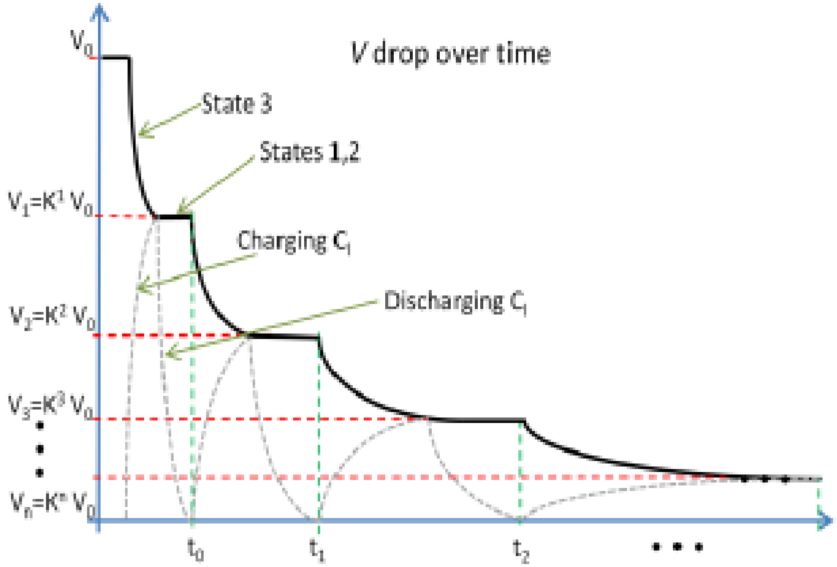
BTW: what is the law with which capacitor is discharged through a switching circuit?





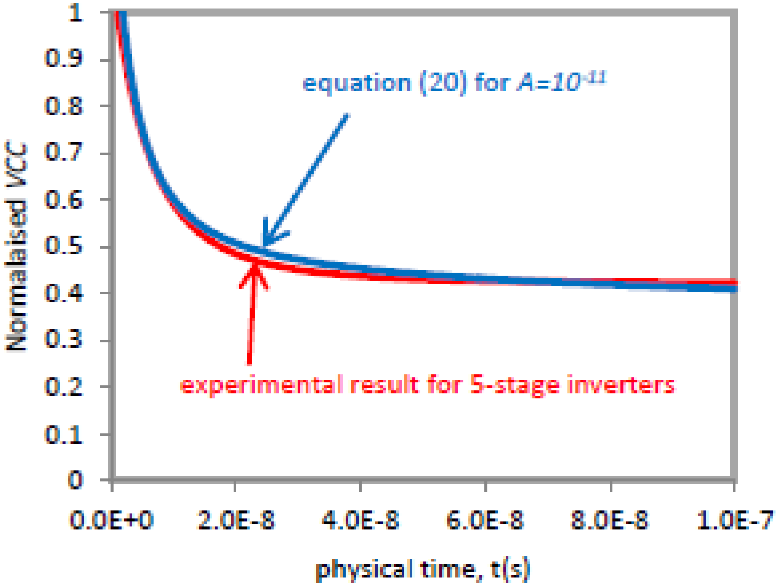
$$V_N = \frac{A}{t(1-K)+AK}$$

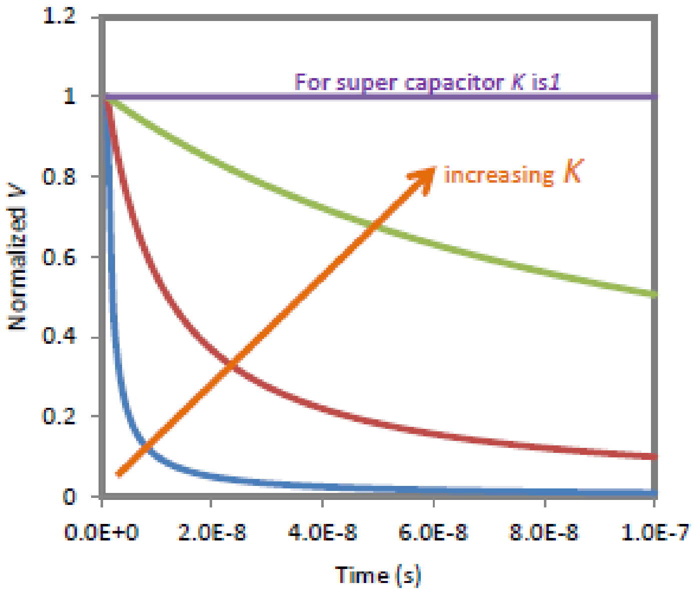




For super-threshold region the discharge is a hyperbola!

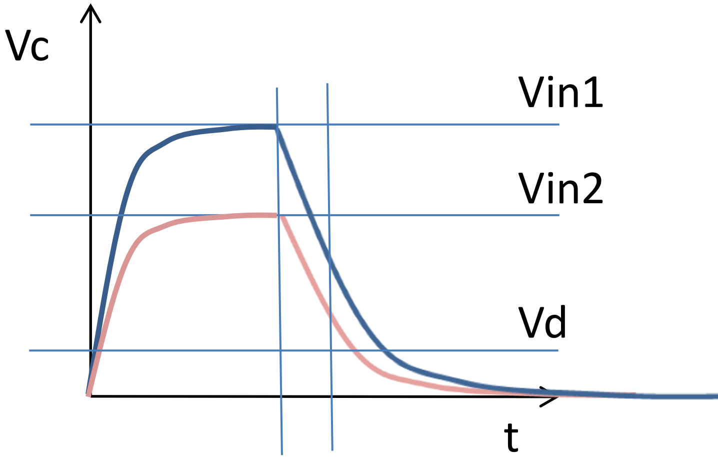
$$V_N = \frac{A}{t(1-K)+AK}$$





The reference-free issue

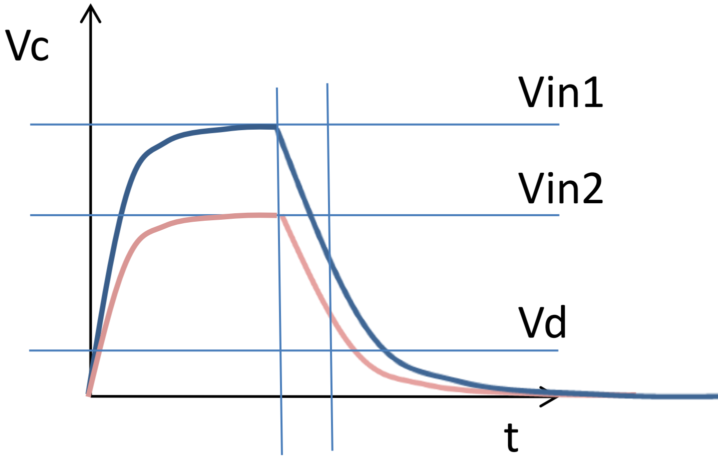
- How to control the time?
- Completely dead computation unit (e.g. counter) does not provide any information (e.g. the last number the counter counted to, which encodes Vin, is lost on death).
- So counter must be stopped before dying completely.
- You can stop counting at the same time, irrespective of Vin – constant sensing/conversion delay.
- However, this "same time" implies timing reference or some clock.



The reference-free issue

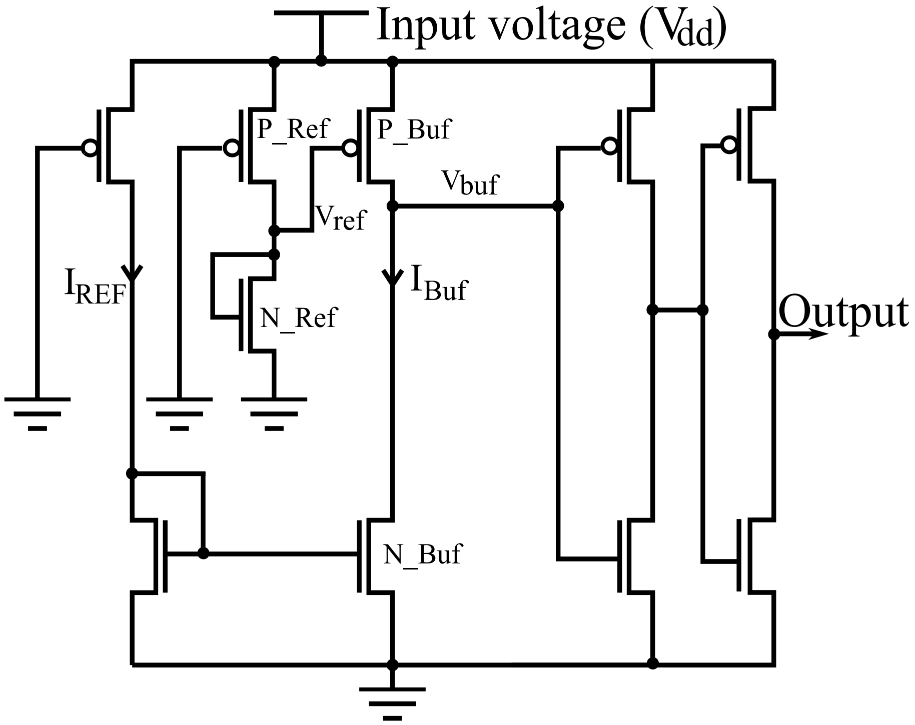
Vd is still a constant reference!

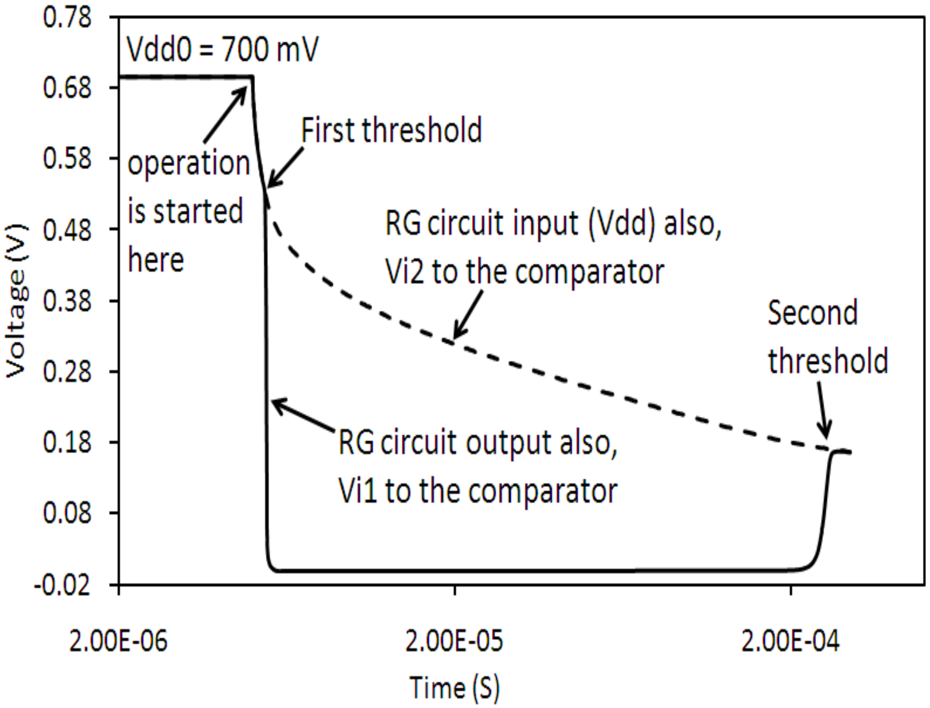
But it does not have to be externally sourced. It could be based on some internal constant such as the threshold of a semiconductor device



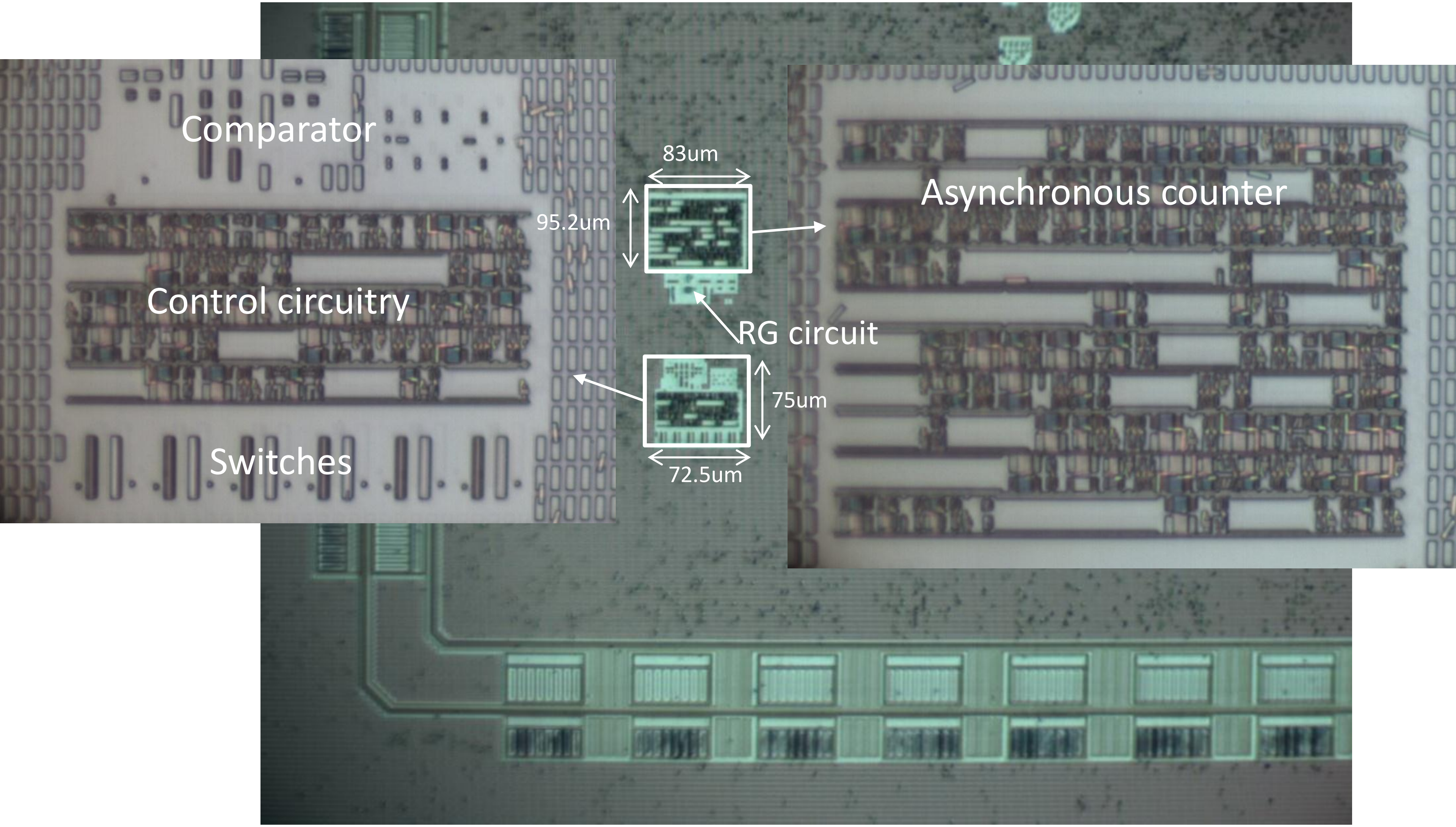
Internal reference generator

Using the transistor threshold voltage as a reference …

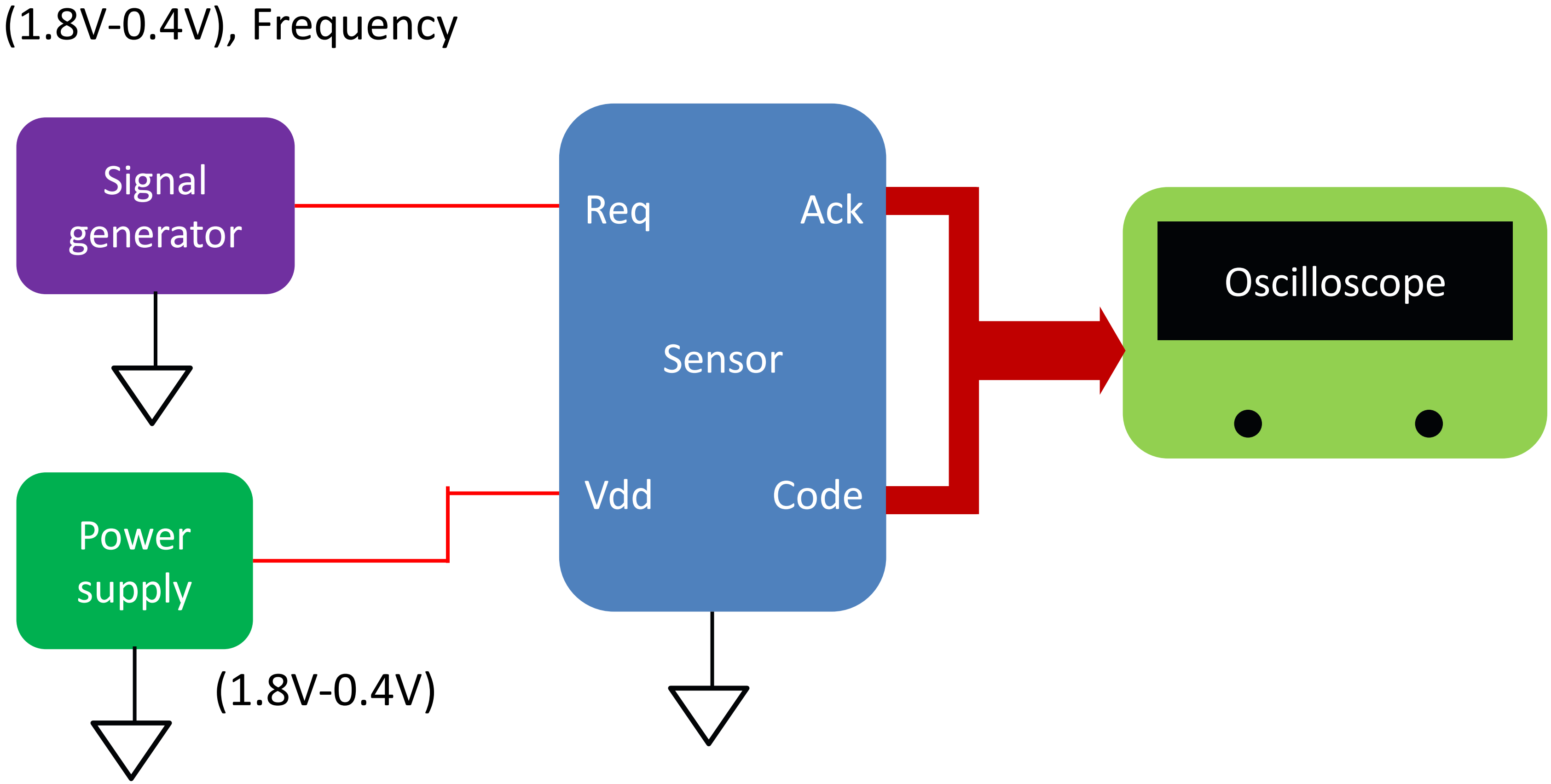




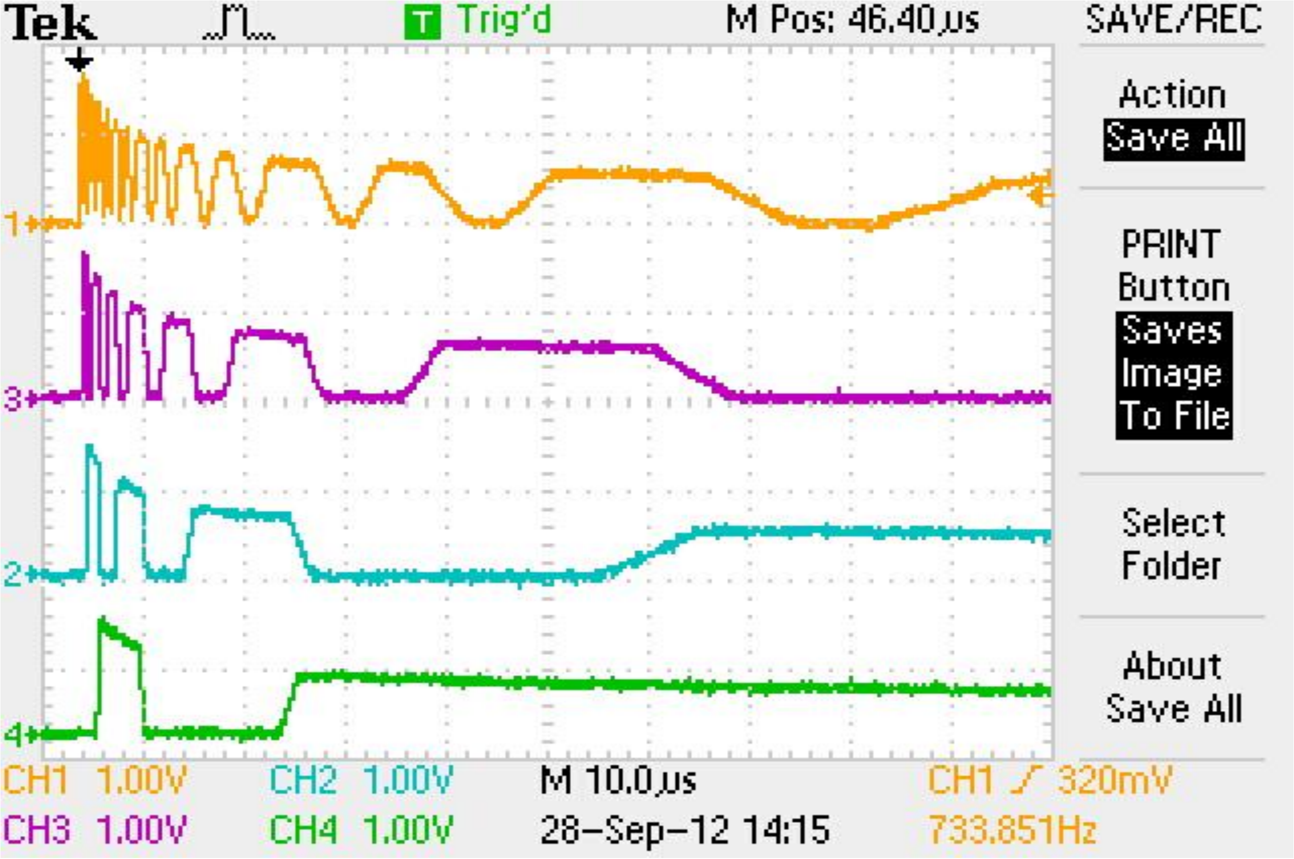
Sensor chip in 180nm CMOS



Test setup

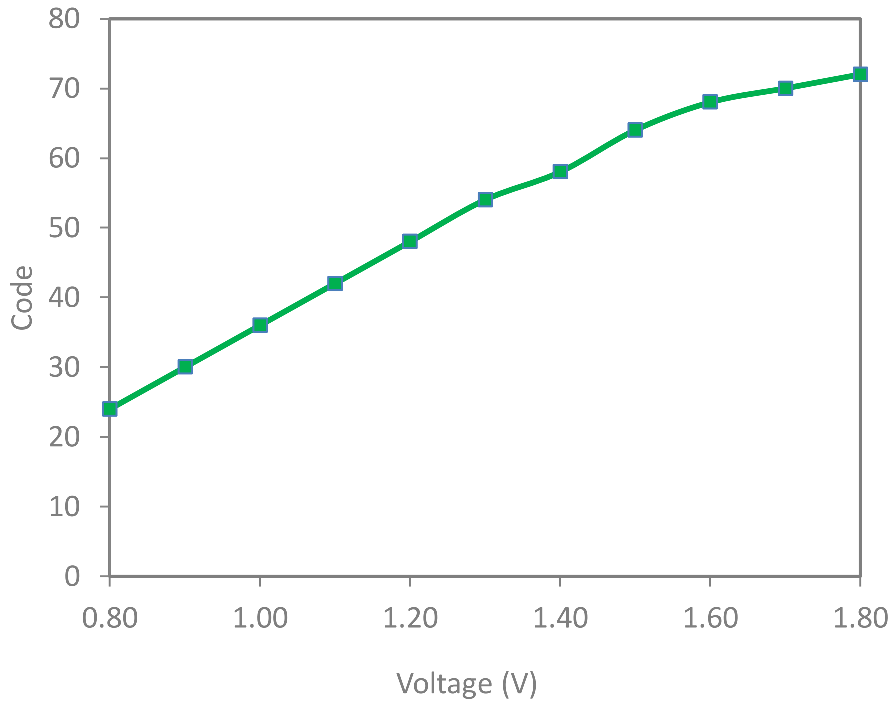

(1.8V-0.4V), Frequency
Signal generator
Req
Ack
Sensor
Vdd
Code
Oscilloscope
Power supply
(1.8V-0.4V)

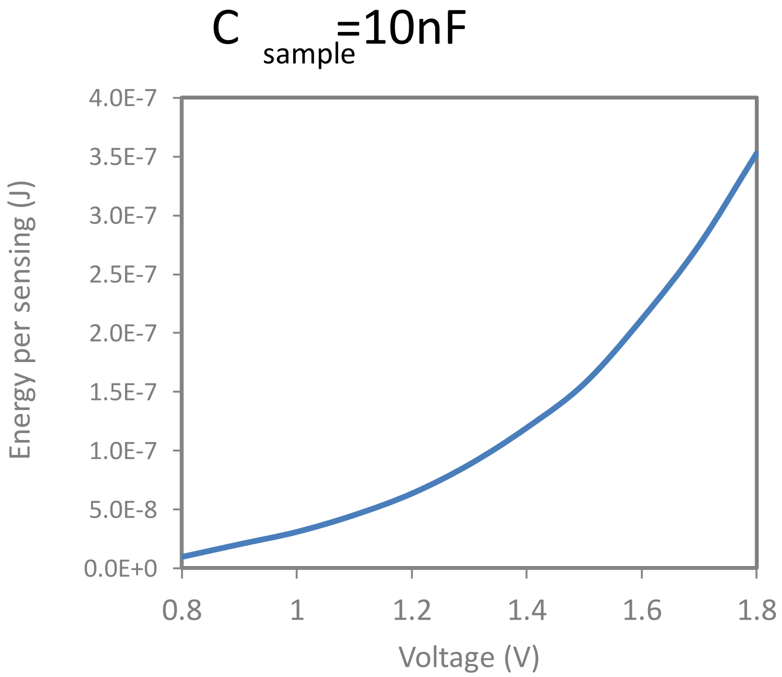
Experimental Results from the chip testing



Output of the counter while it is powered by the sampling capacitor

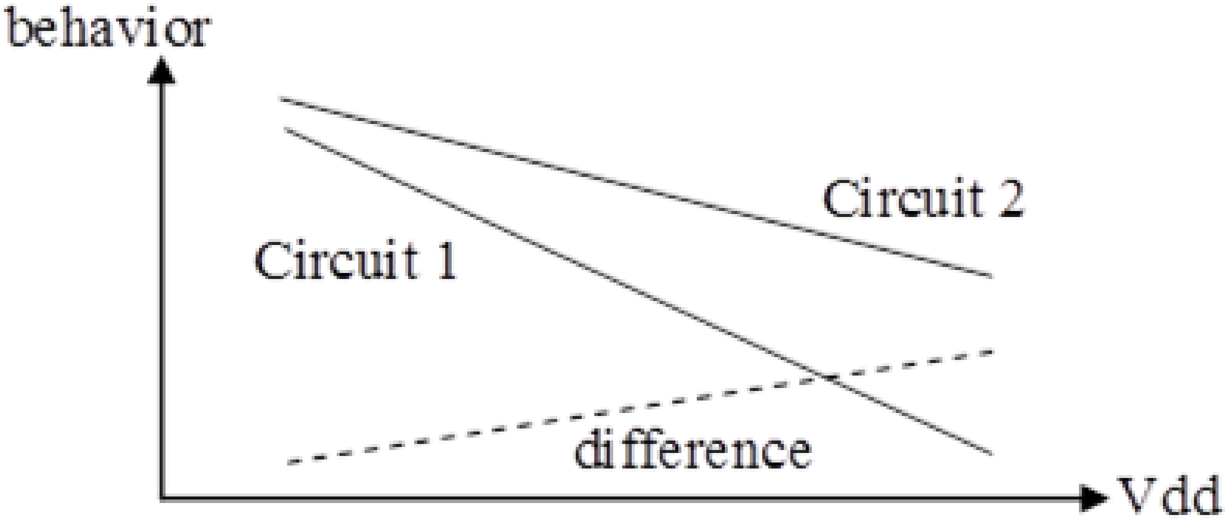
Output count and energy consumption





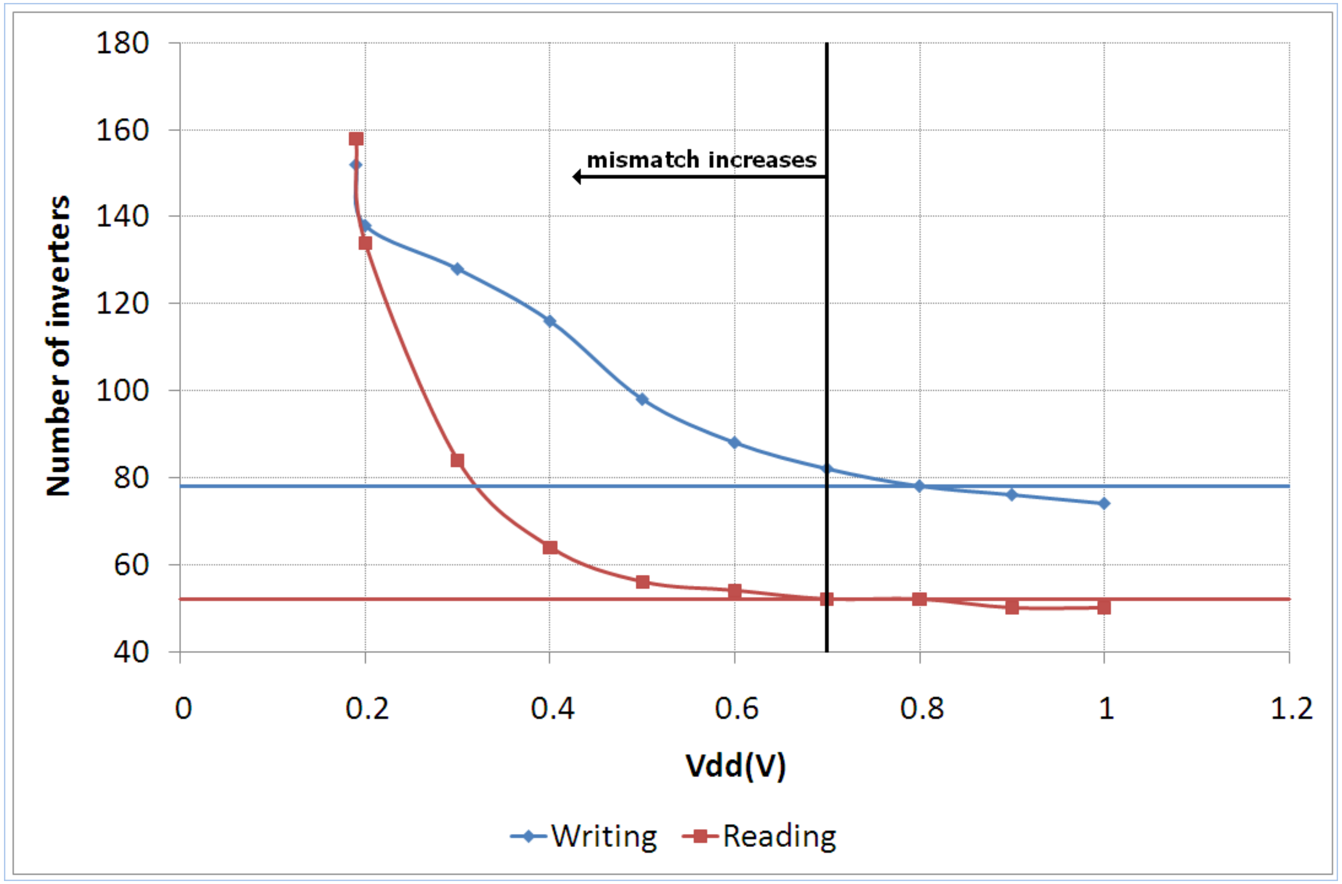
Reference-free sensing using difference in behaviour

- If two types of circuits have different behaviour (e.g. delay) when Vdd changes, the difference may encode the Vdd



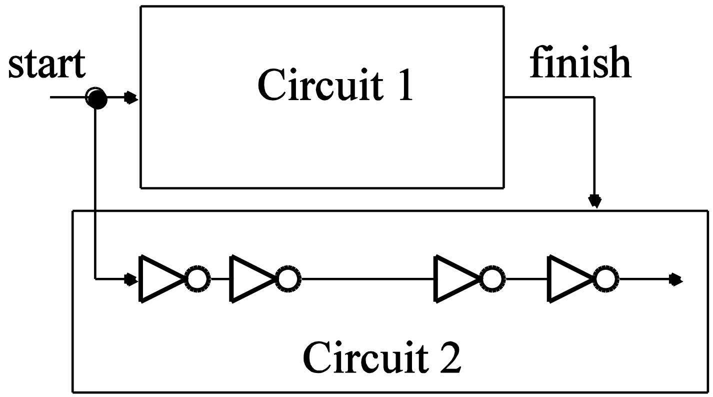
Delay differentiators

– The memory-logic delay mismatch when Vdd reduces

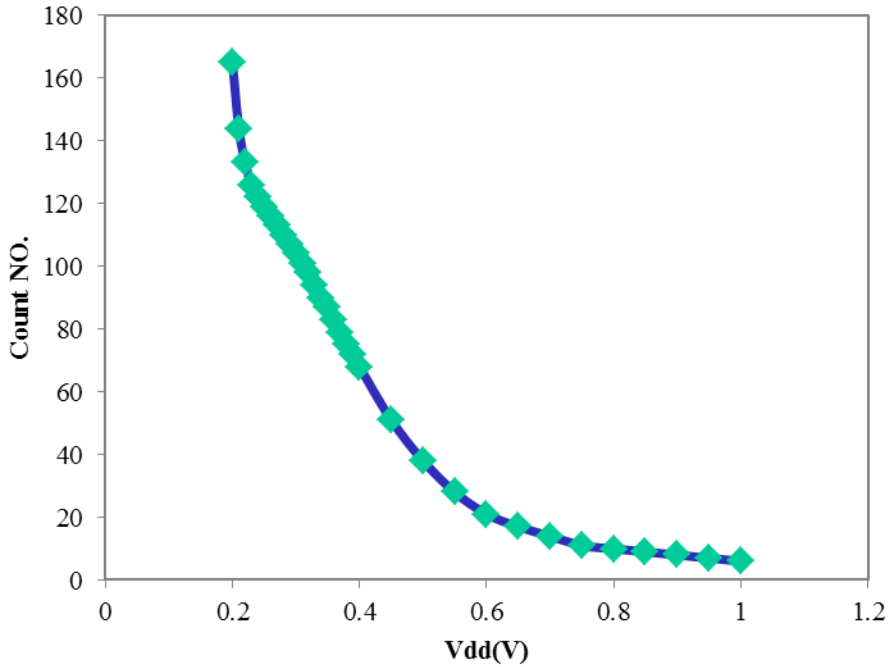


Using delay differentiators

Using memory as Circuit 1 and regular logic (chain of inverters) as Circuit 2:



1. Charge the sampling capacitor with Vin, after a while we have Vc=Vin tracking relation.



2. When a sensing/conversion command comes, break capacitor away from Vin and start circuits 1 and 2 together.

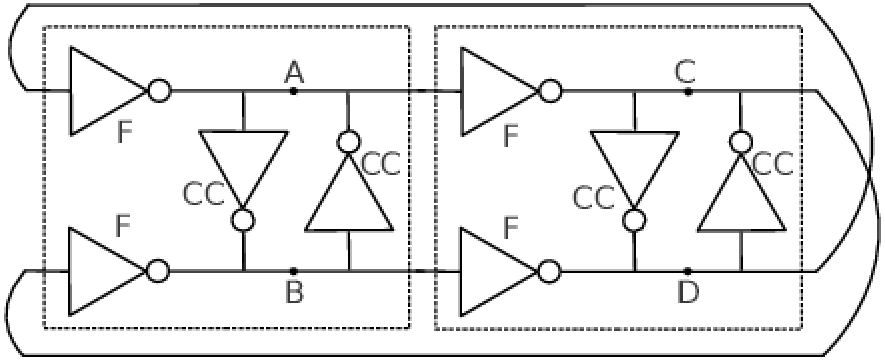
3. When circuit 1 activity ends, output code (count) from circuit 2.

Sensing by detecting oscillations

When you want to know if Vdd drops below some critical point

- Identification of voltage threshold crossing based on the change of circuit operating modes
- 4-phase clock generation, clock recovery, complex signal processing

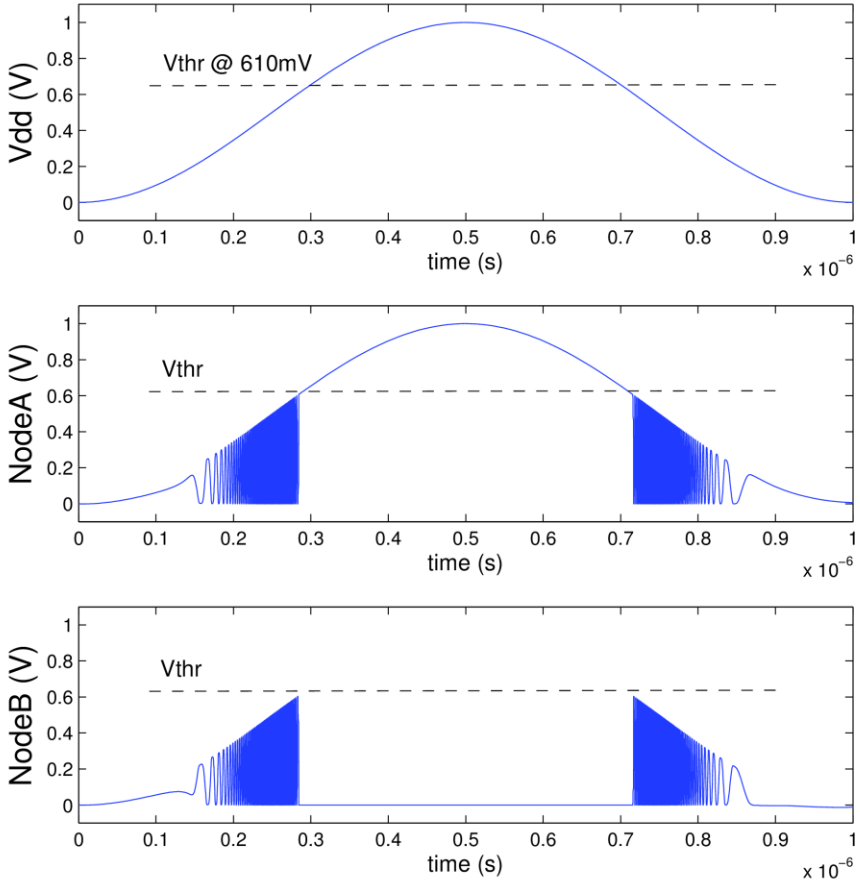
- Stage : 2 forward (F) inverters,
- 2 cross-coupled (CC) inverters
- Two operating modes
- Oscillation
- Latching/Locking

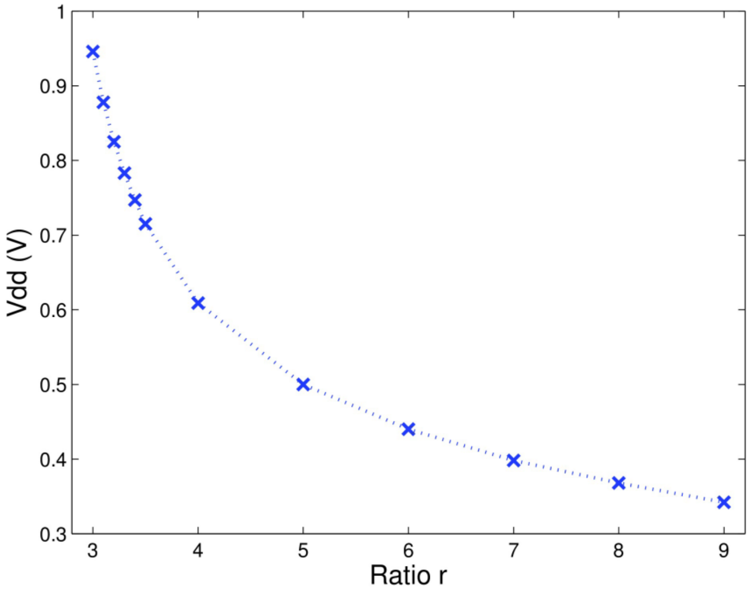


$$r = \frac{width\ of\ CC}{width\ of\ F}$$

Parameter settings

Oscillatory and non-oscillatory modes on two sides of threshold; thresholds set with inverter size ratios

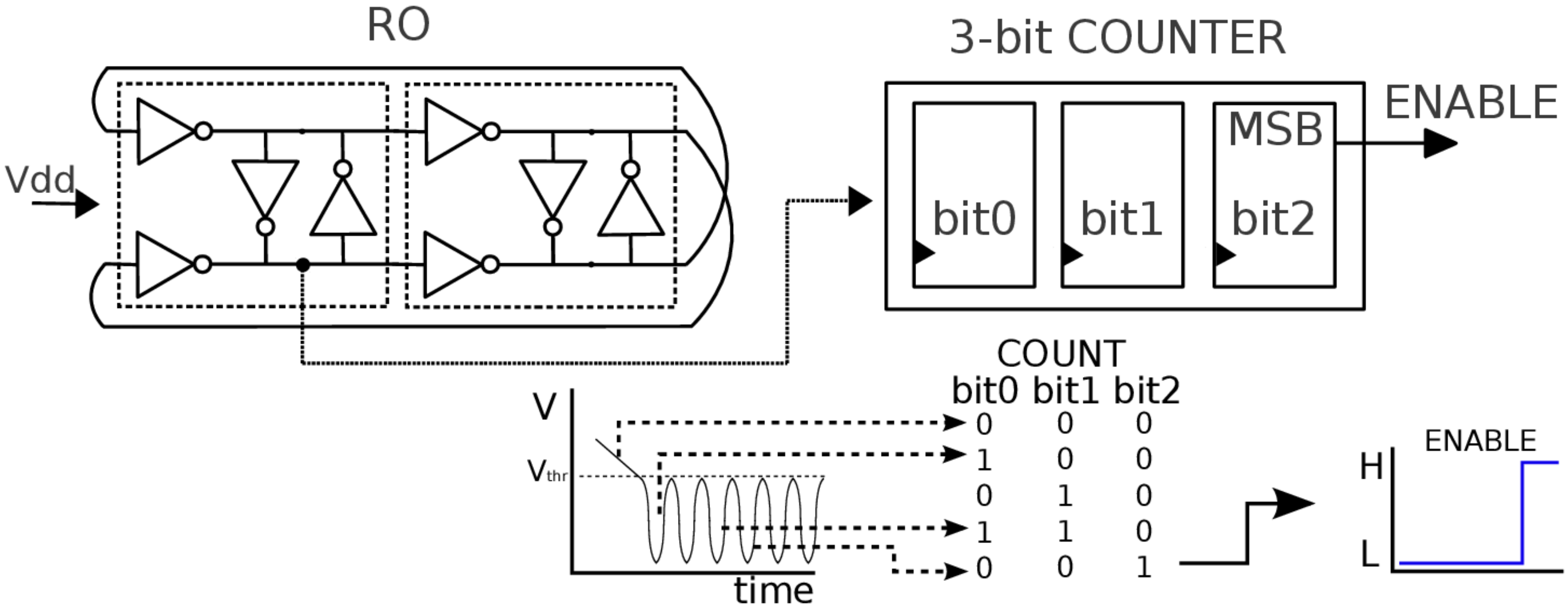




$$r = \frac{width\ of\ CC}{width\ of\ F}$$

Detecting oscillations

Configuration for the detection of the onset of oscillation



COUNT

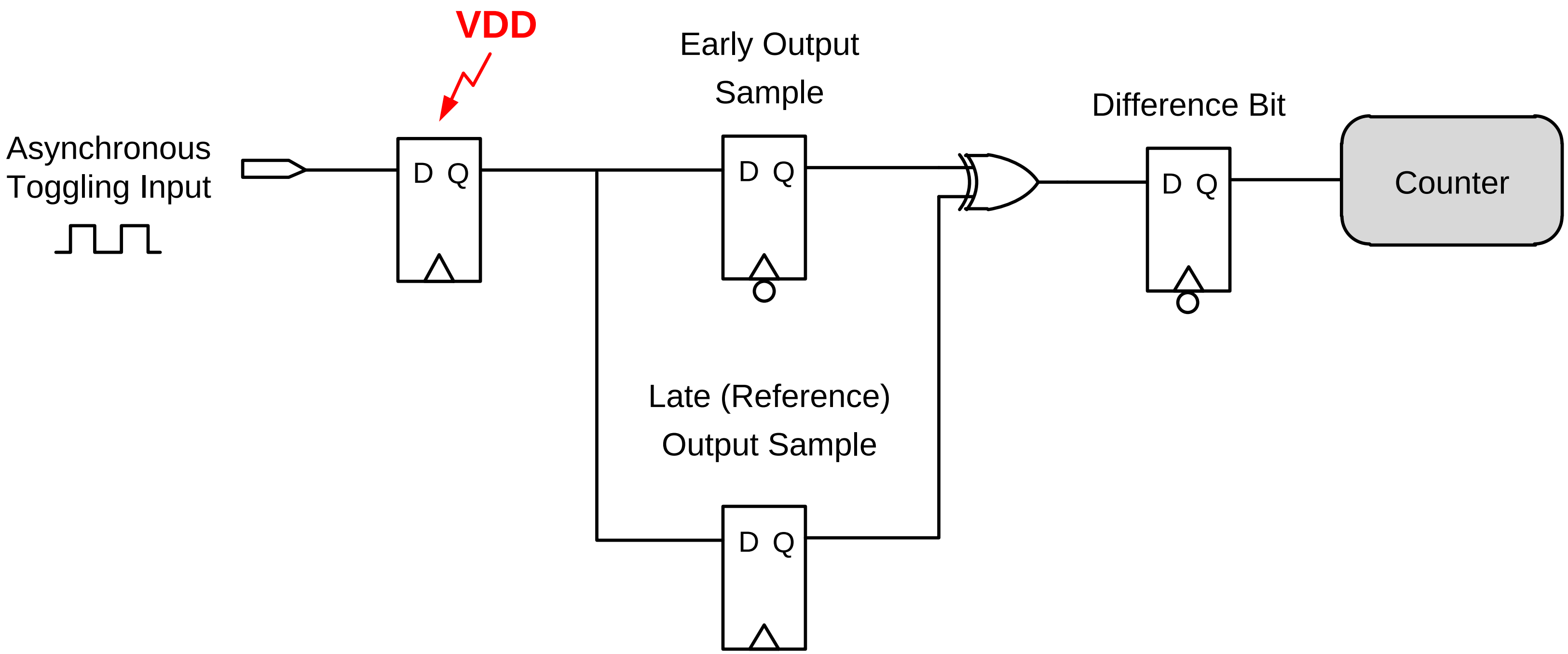
bit0	bit1	bit2
0	0	0
1	0	0
0	1	0
1	1	0
0	0	1

Making use of metastability

- Metastability offers a nice way of removing external references in Voltage and Temperature sensors
  - When the setup and hold time conditions of a flip-flop are not met, the flip-flop may become metastable
  - A metastable flip-flop will take extra time to decide whether to go logic high or low (decision time = clock-to-q delay)
  - The "decision making" time constant (τ) is a function of Vdd

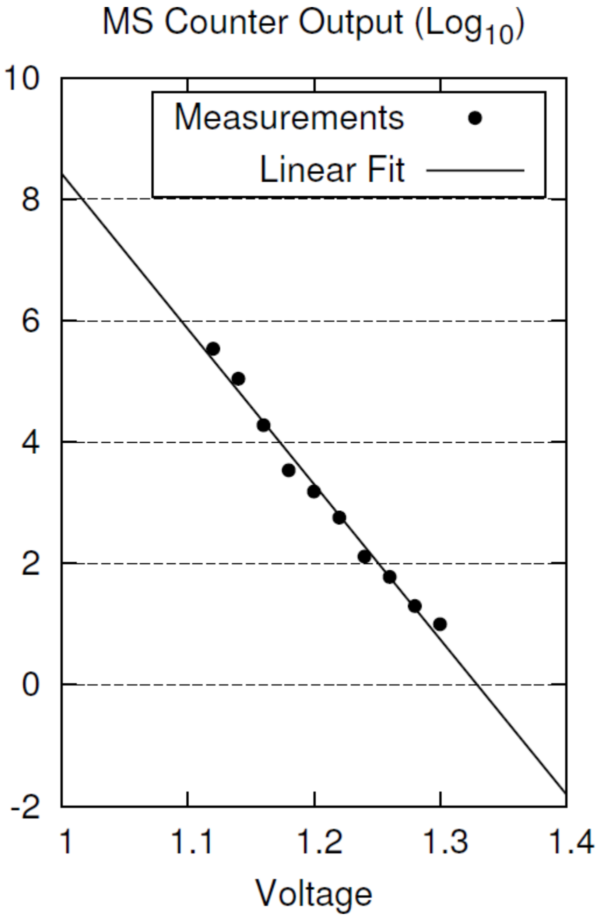
Making use of metastability

- **Idea:** Use the time constant (τ) to quantify Vdd
- **How:** Count the rate at which the flip-flop fails to decide!



Making use of metastability

- Sensors – Making use of metastability
  - **Response function:** $\text{Count} = n \times K \times e^{Sp}$
  - **Advantages:**
    - Purely digital
    - Very compact (4FF's plus one XOR gate)
    - High precision



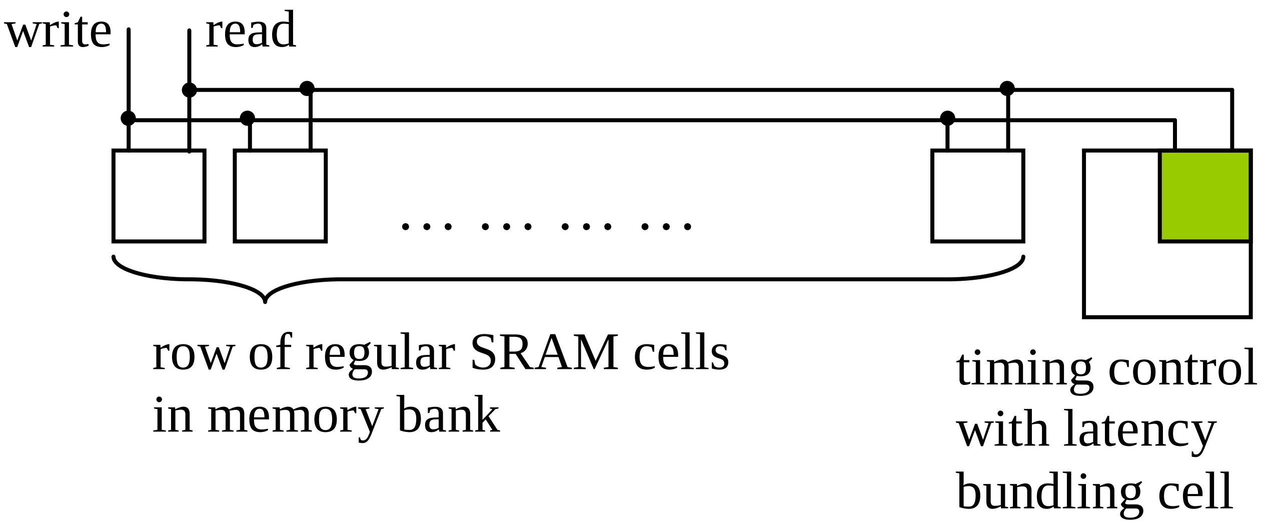
FPGA Measurements

(Altera Cyclone II)

RETAINING DATA: ELASTIC MEMORY

Elastic Data Storage

- Self-timed SRAM



write

read

row of regular SRAM cells
in memory bank

timing control
with latency
bundling cell

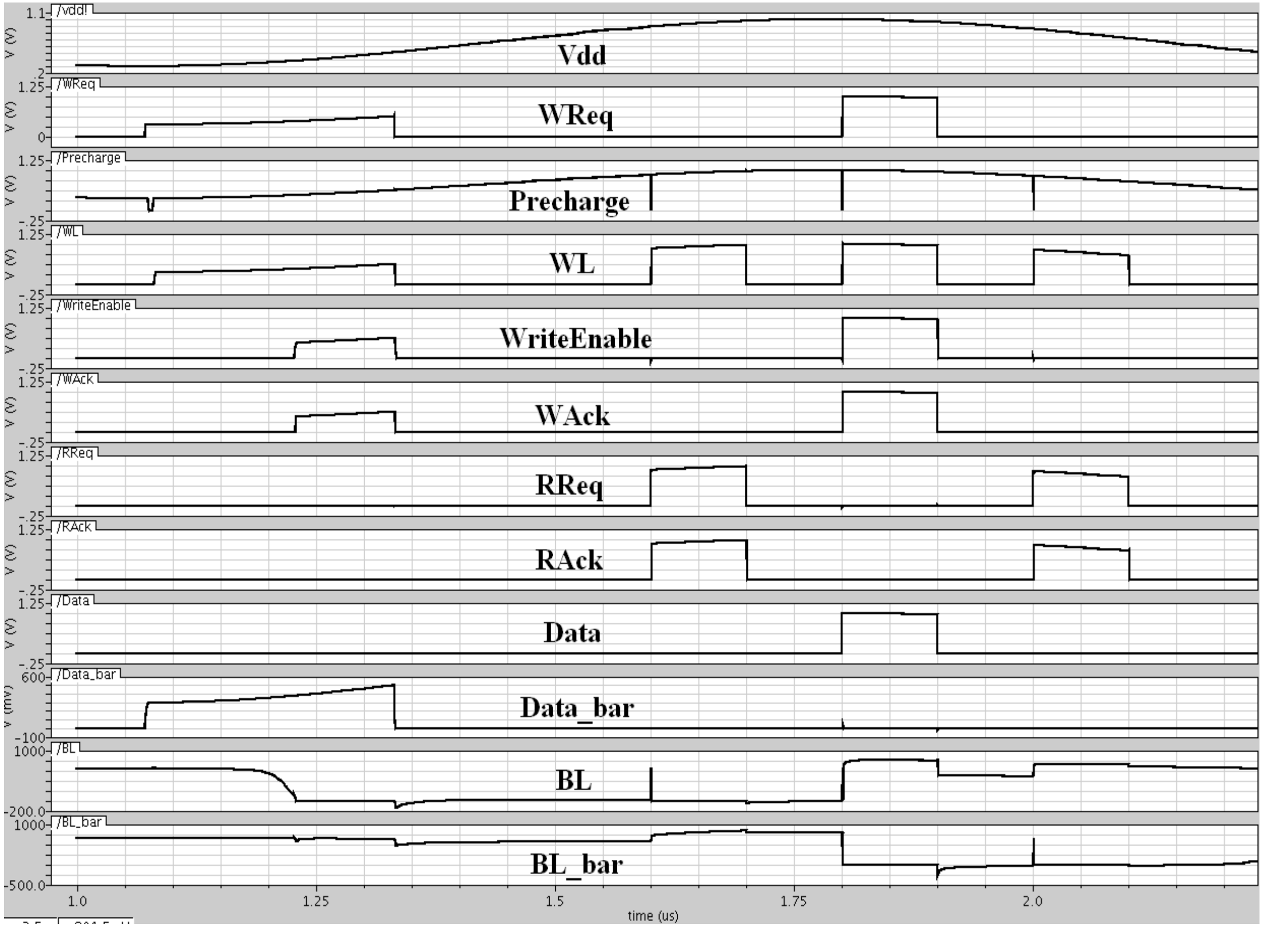
Time-
bundled
SRAM with
fully SI cell
as bundling
unit

6T solution for energy efficiency.

10T solution for core-function survivability.

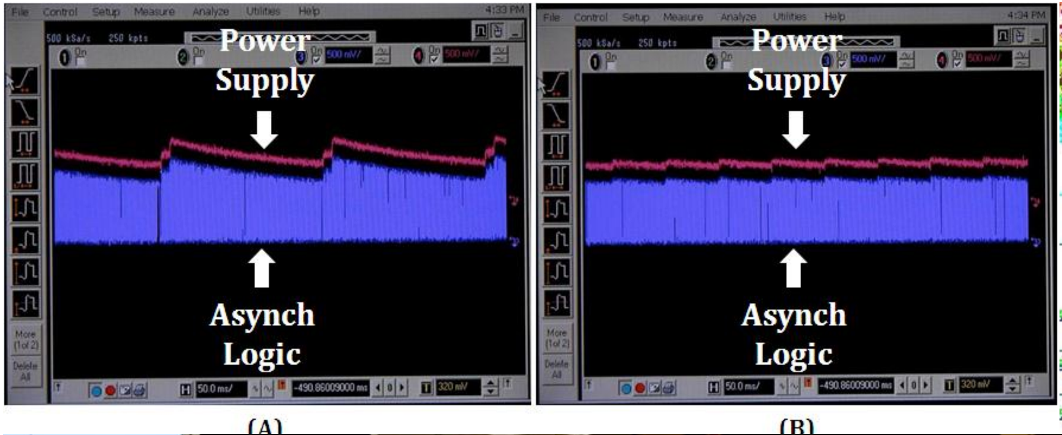
Self-timed SRAM

- Self-timed SRAM under variable Vdd



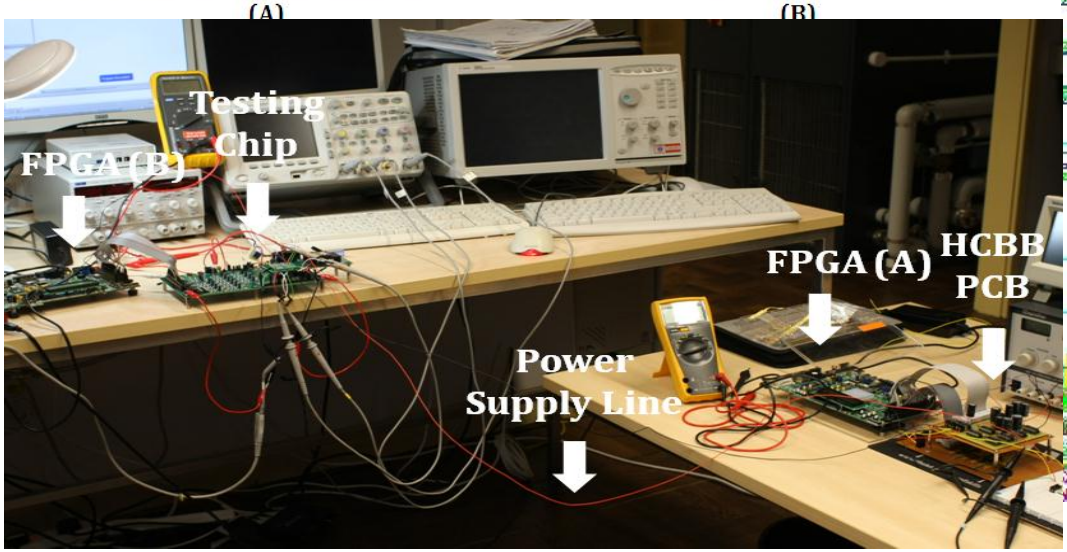
SRAM Chip in 90nm CMOS

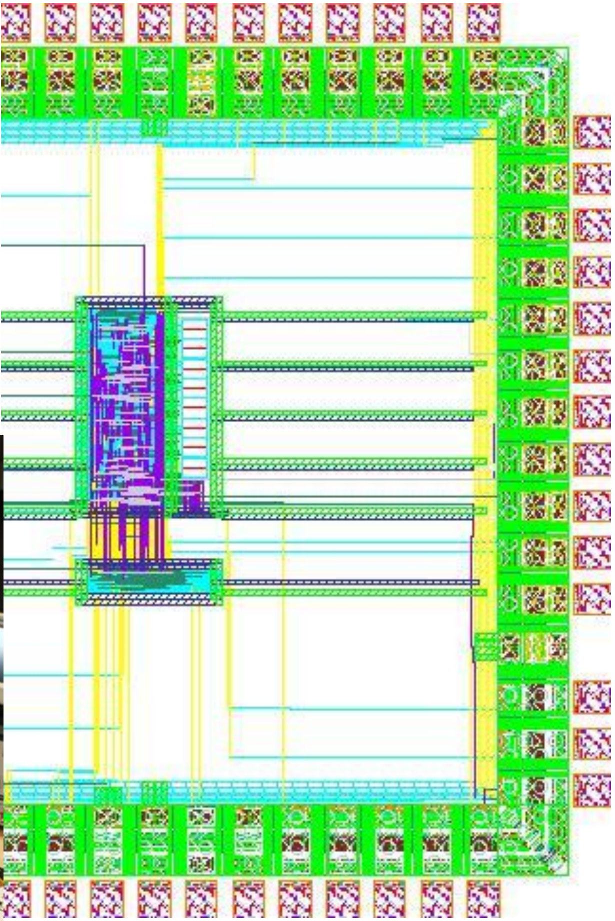
- Self-timed SRAM



(A)

(B)



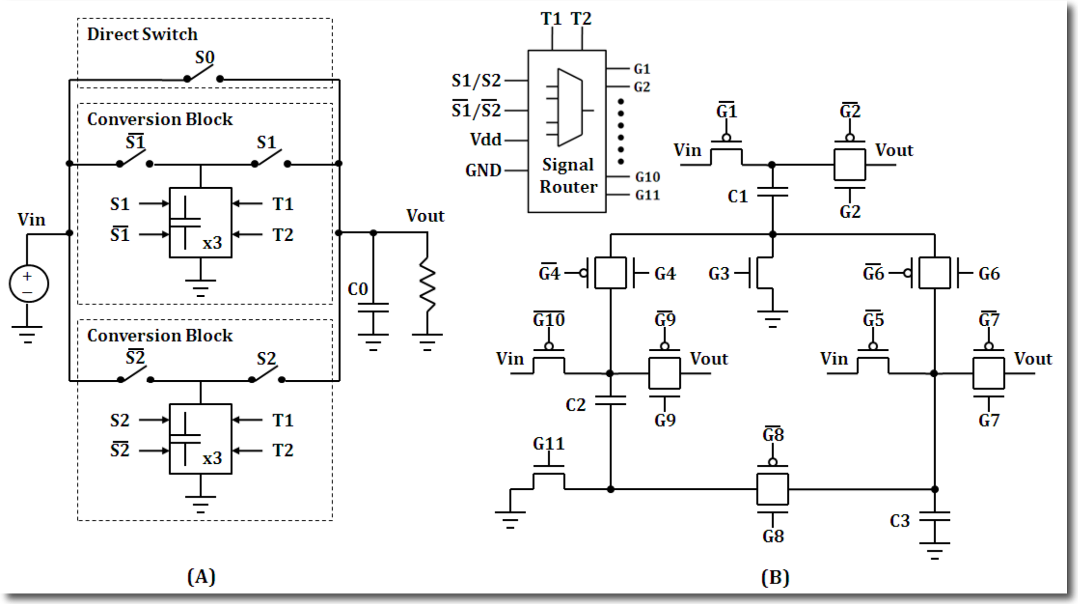


RETAINING ENERGY: ELASTIC POWER MANAGEMENT

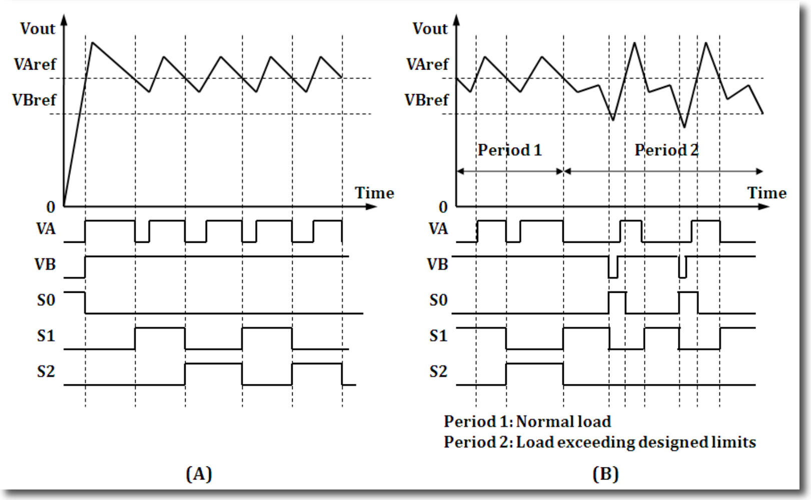
Power Management

– Conventionally there is switched capacitor DC/DC converter (SCC)

– Converts constant input Vdd to constant output Vdd according to a set of ratios



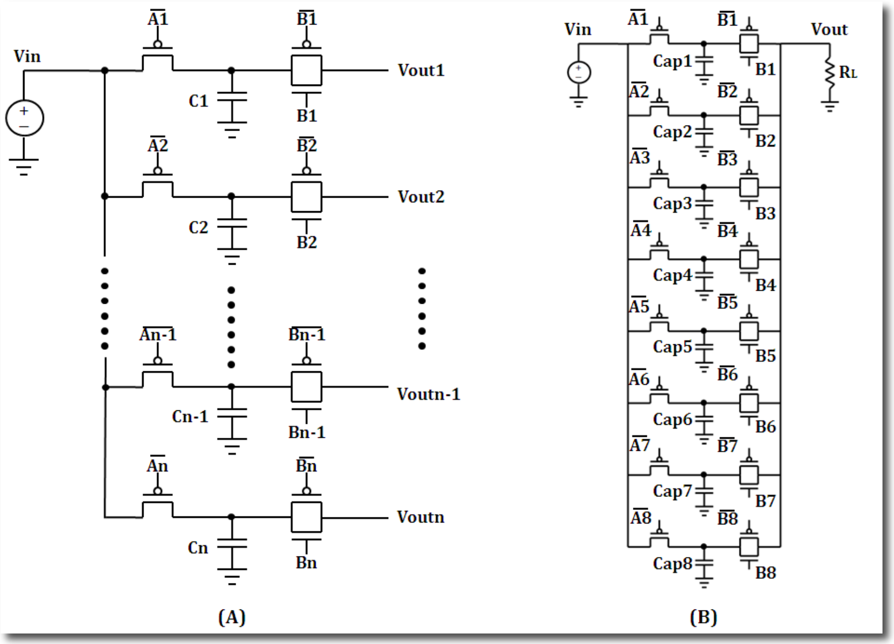
SCC Structure



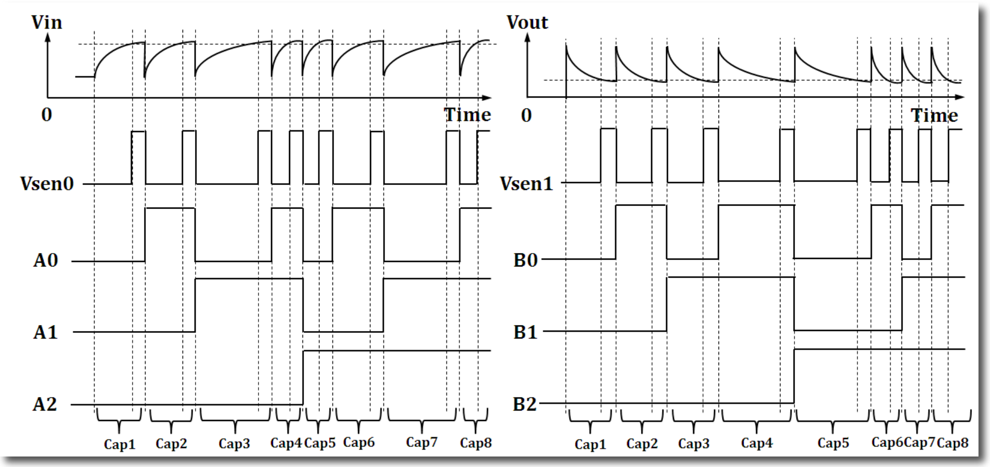
SCC Behaviour

Elastic Power Management

- What if the load does not demand constant Vdd?
- Can now use a capacitor bank block (CBB) with linear charging/discharging



CBB Structure

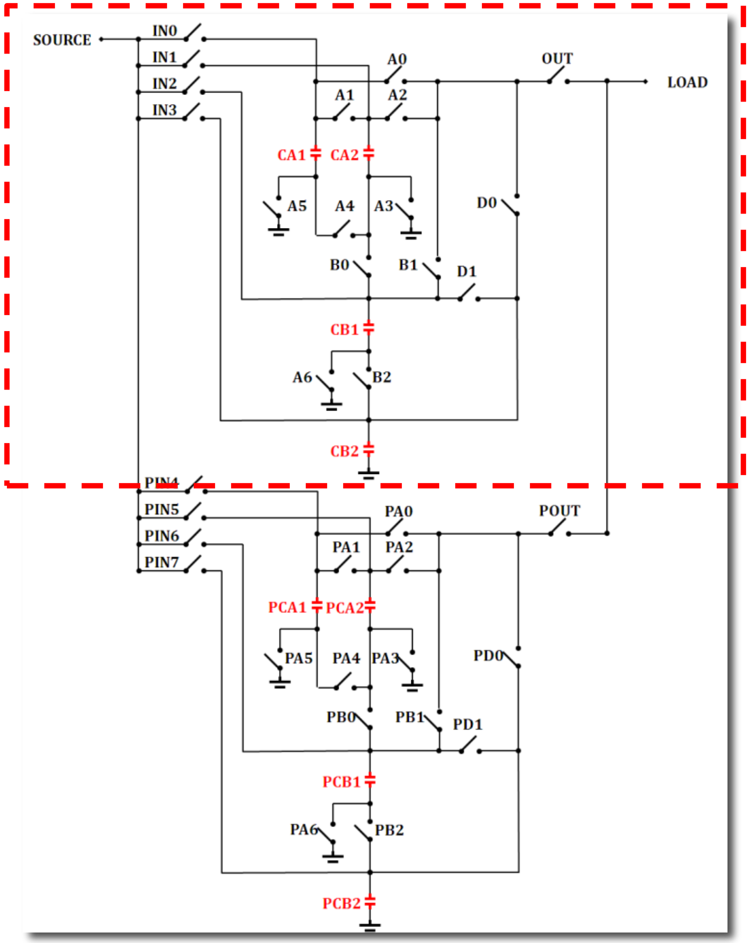


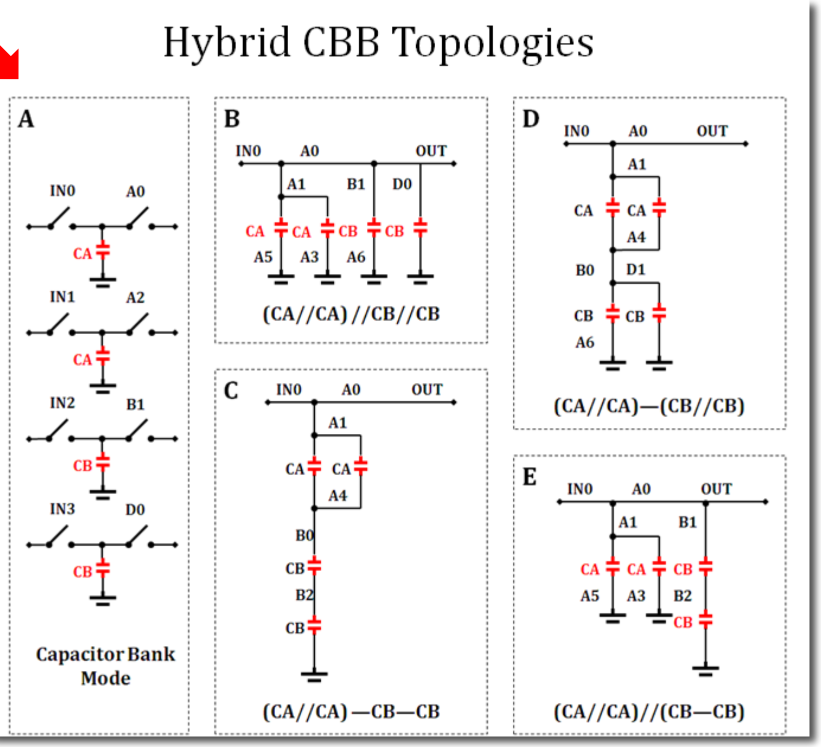
CBB Behaviour

Elastic Power Management

- Hybrid CBB for the best of both

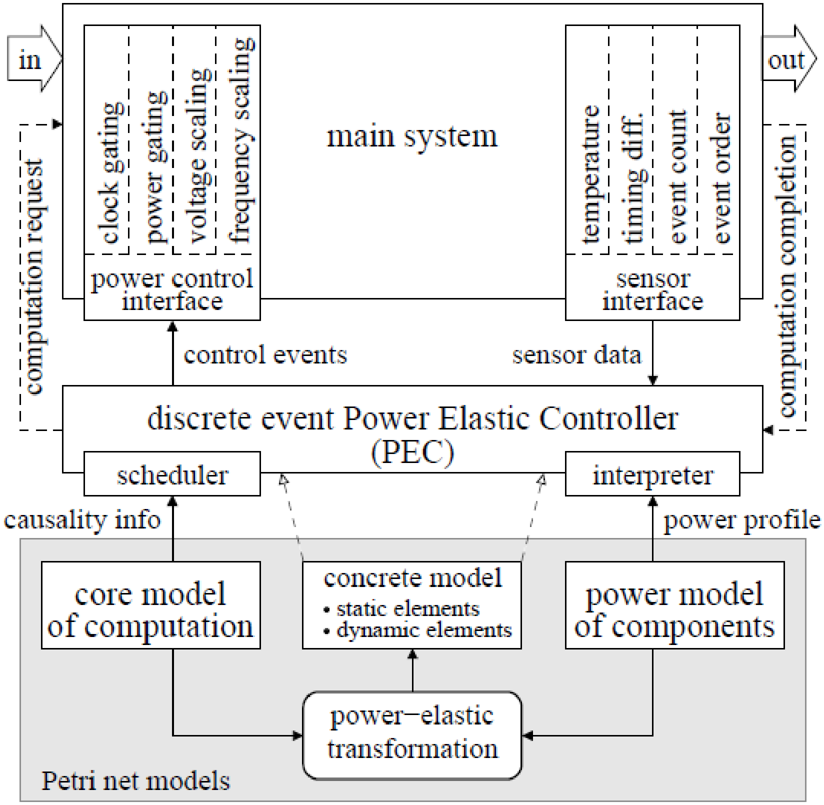
Hybrid CBB Structure



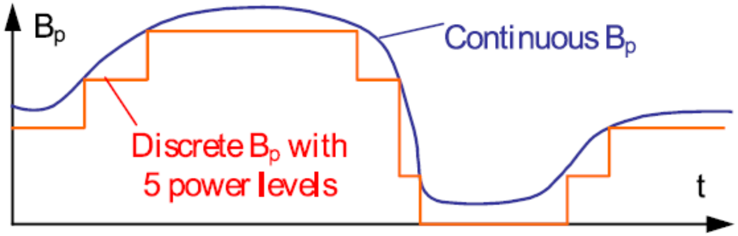


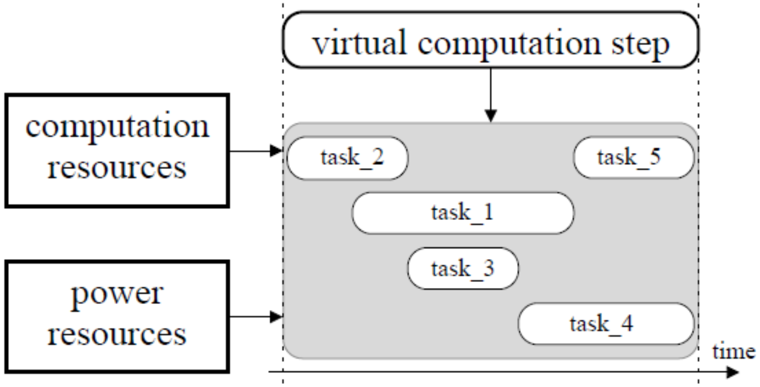
Energy-modulated task scheduling

- Task scheduling
  - Energy-modulated concurrency adjustments



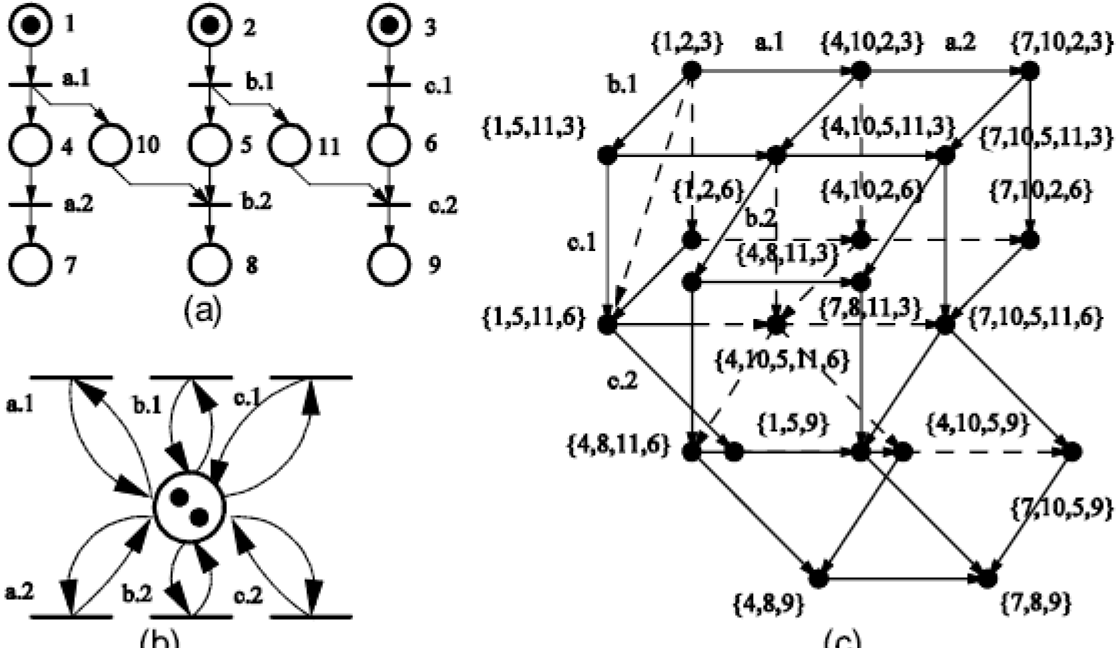
Input power profile





Energy-modulated task scheduling

- Task scheduling - Petri net modelling
  - Energy-modulated concurrency adjustments
  - Concurrency can be regulated with the number of tokens put into the control place in (b)



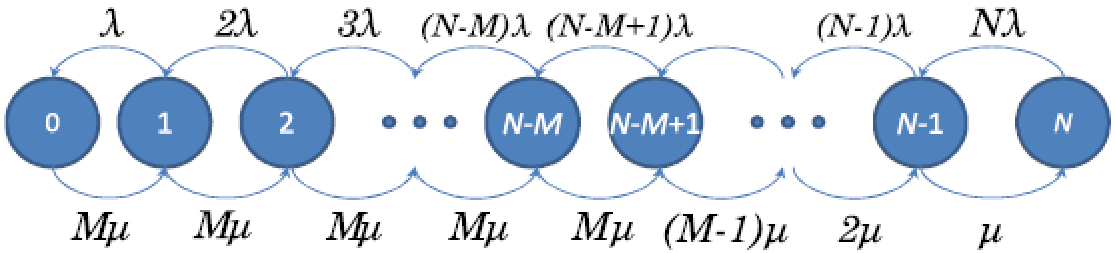
(a)

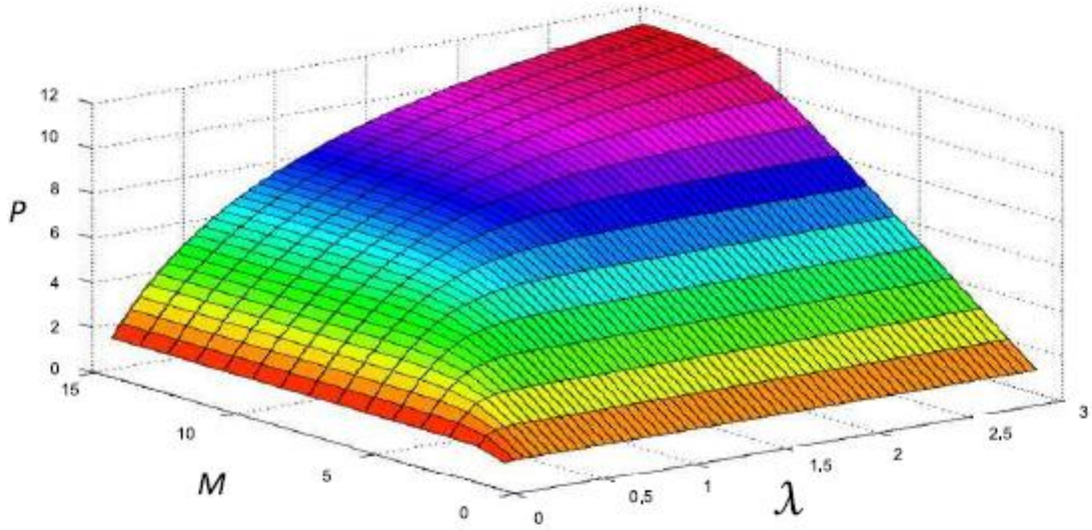
(b)

(c)

Concurrency and Power in Task Scheduling

- Task scheduling – Markov process modelling
  - Energy-modulated concurrency adjustments
  - The degree of concurrency (M) and its effect on power

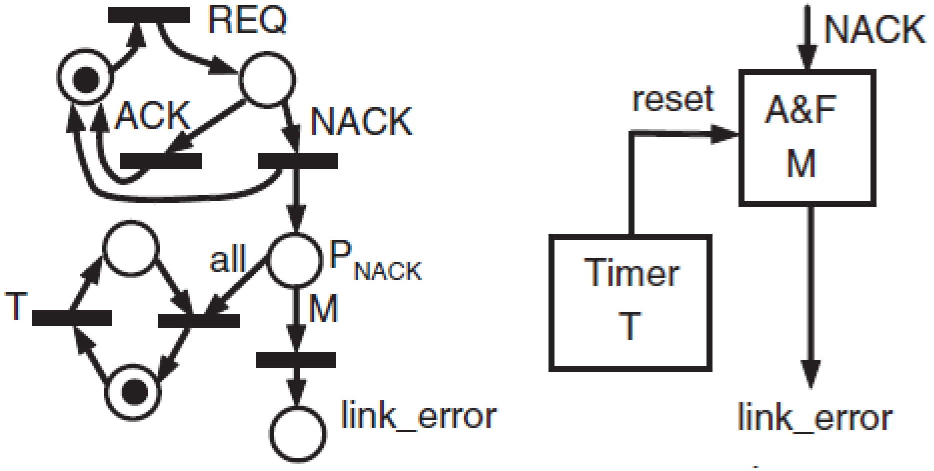




Mechanisms in COMMUNICATION FABRICS

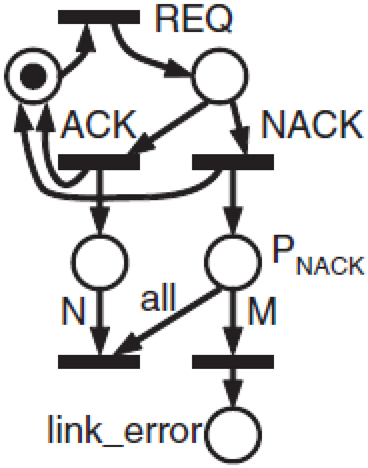
Self-Diagnosis and Monitoring

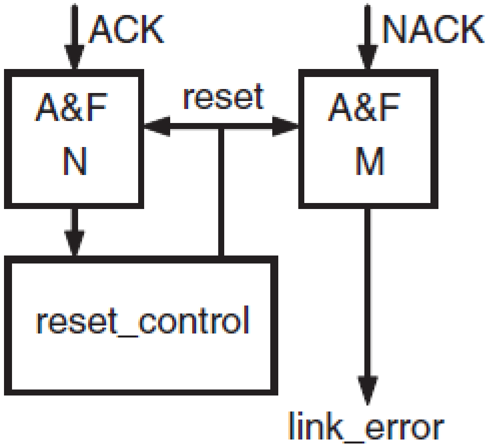
- Self-diagnosis and monitoring using thresholds and the accumulate and fire principle (here detecting non-transient faults in a network by analysing the number of faults during a constant time window)



Self-Diagnosis and Monitoring

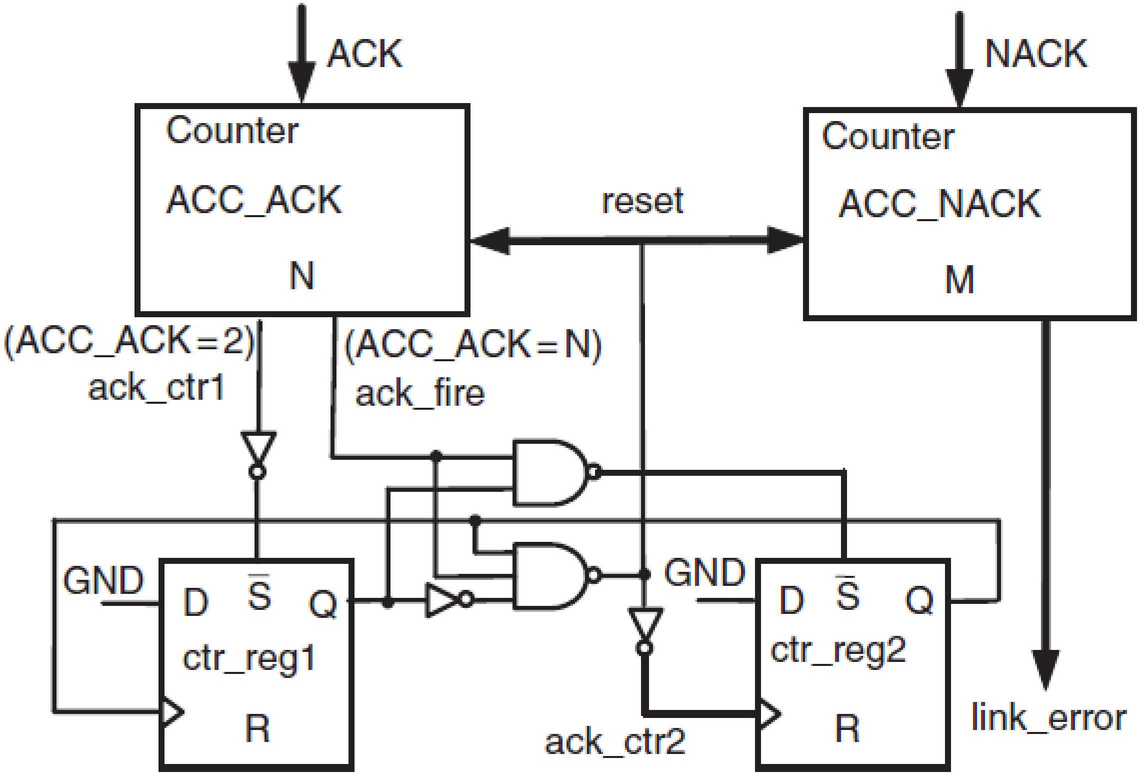
- Non-transient fault detection through monitoring fault density





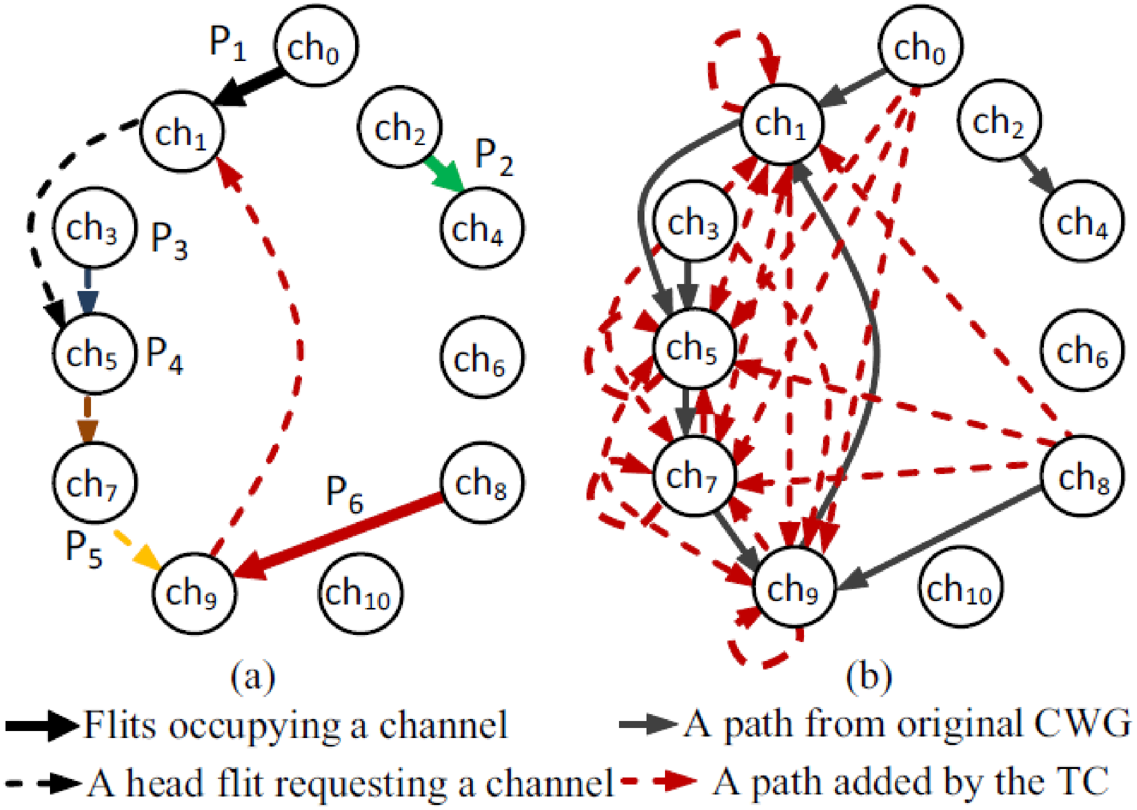
Self-Diagnosis and Monitoring

- Non-transient fault detection through monitoring fault density



Deadlock Detection

- Deadlock detection using distributed transitive closure
  - Channel Wait-for Graph to Transitive Closure computation

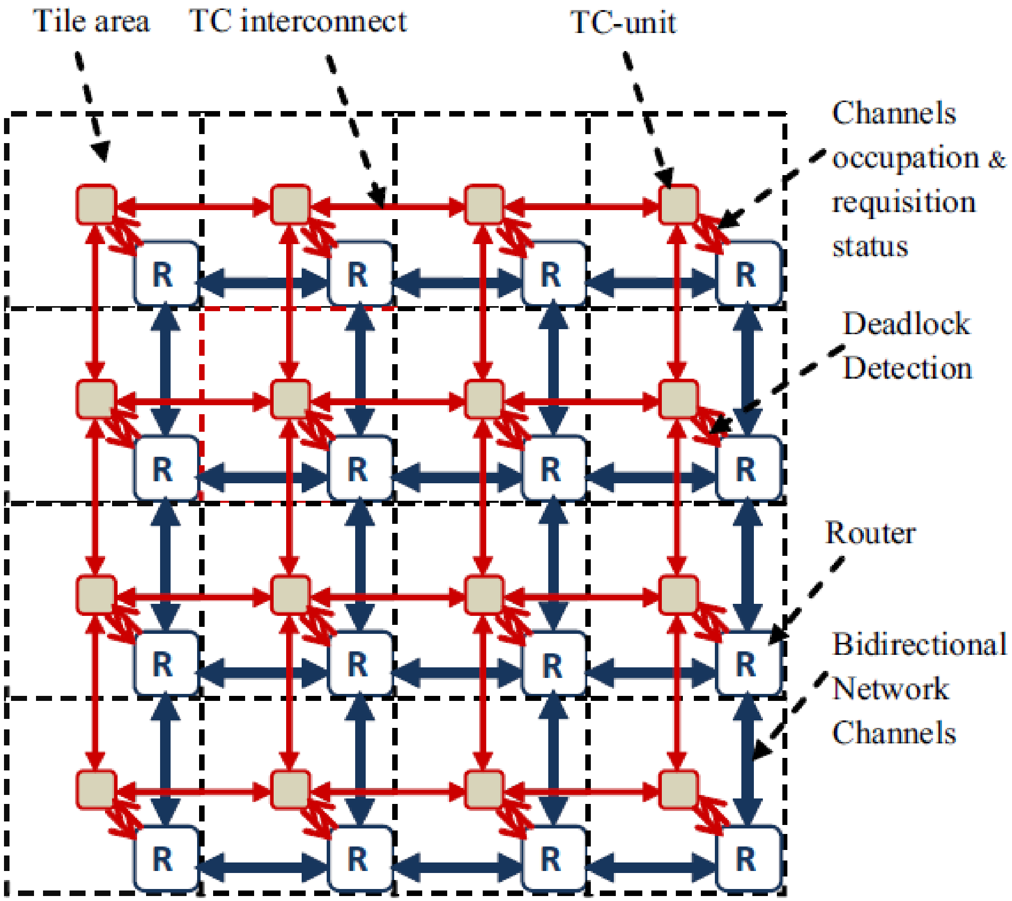


(a)

(b)

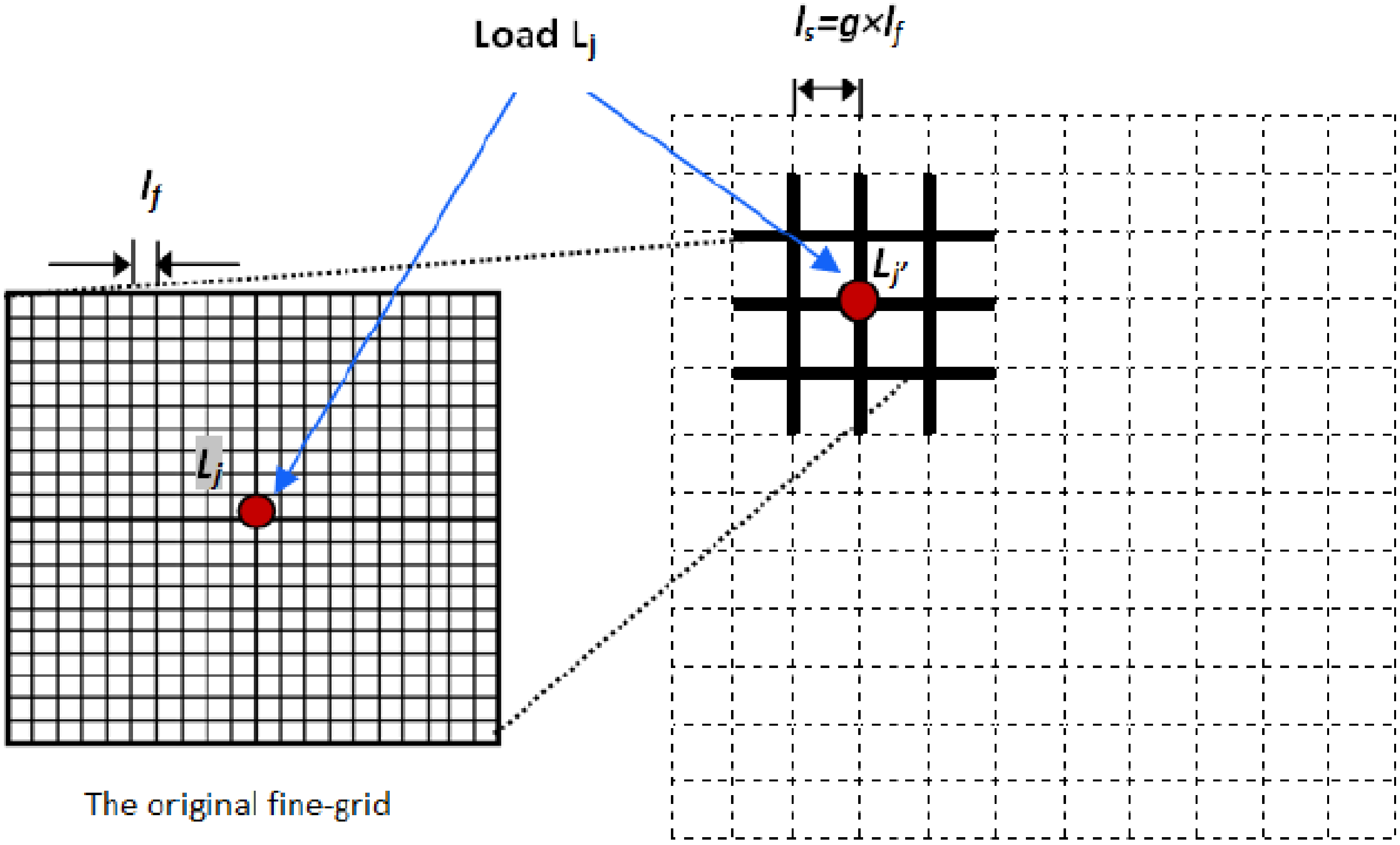
Deadlock Detection

- Deadlock detection using distributed transitive closure
  - TC computation network superimposed on regular network (different layers)


Tile area
TC interconnect
TC-unit
Channels occupation & requisition status
Deadlock Detection
Router
Bidirectional Network Channels
R

Power Noise Sensing and Monitoring

– Coarse-grid for power noise monitoring

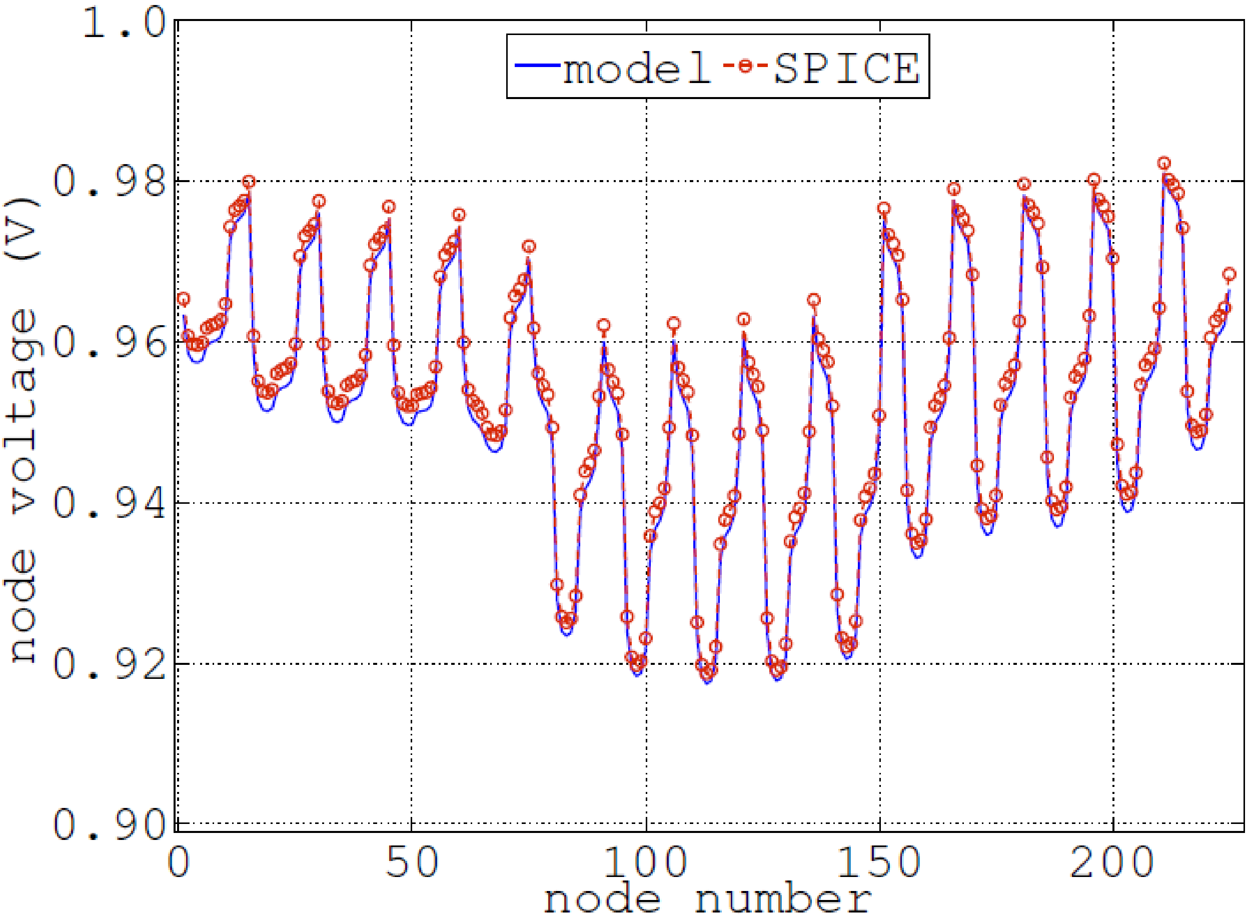


The original fine-grid

The reduced coarse-grid (g=10)

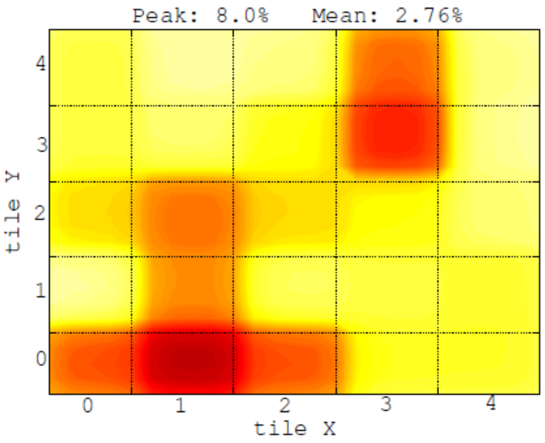
Power Noise Sensing and Monitoring

– Modelling compared with SPICE

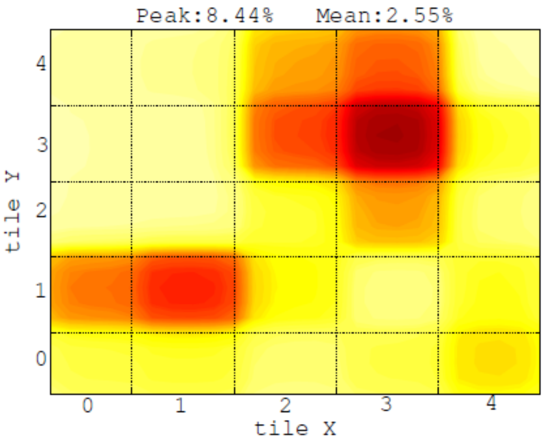


Power Noise Sensing and Monitoring

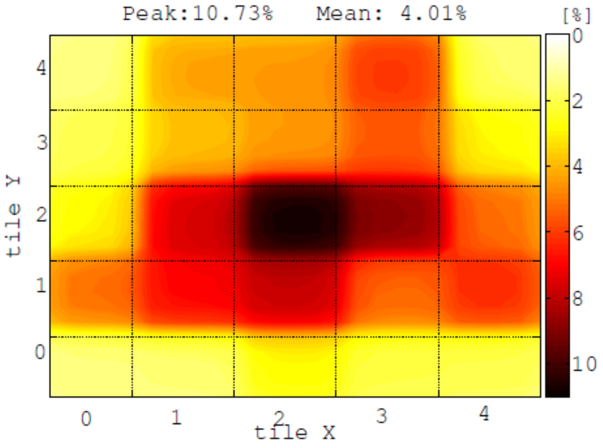
– Vdd drop for three mapping strategies



(a) Maximum performance mapping



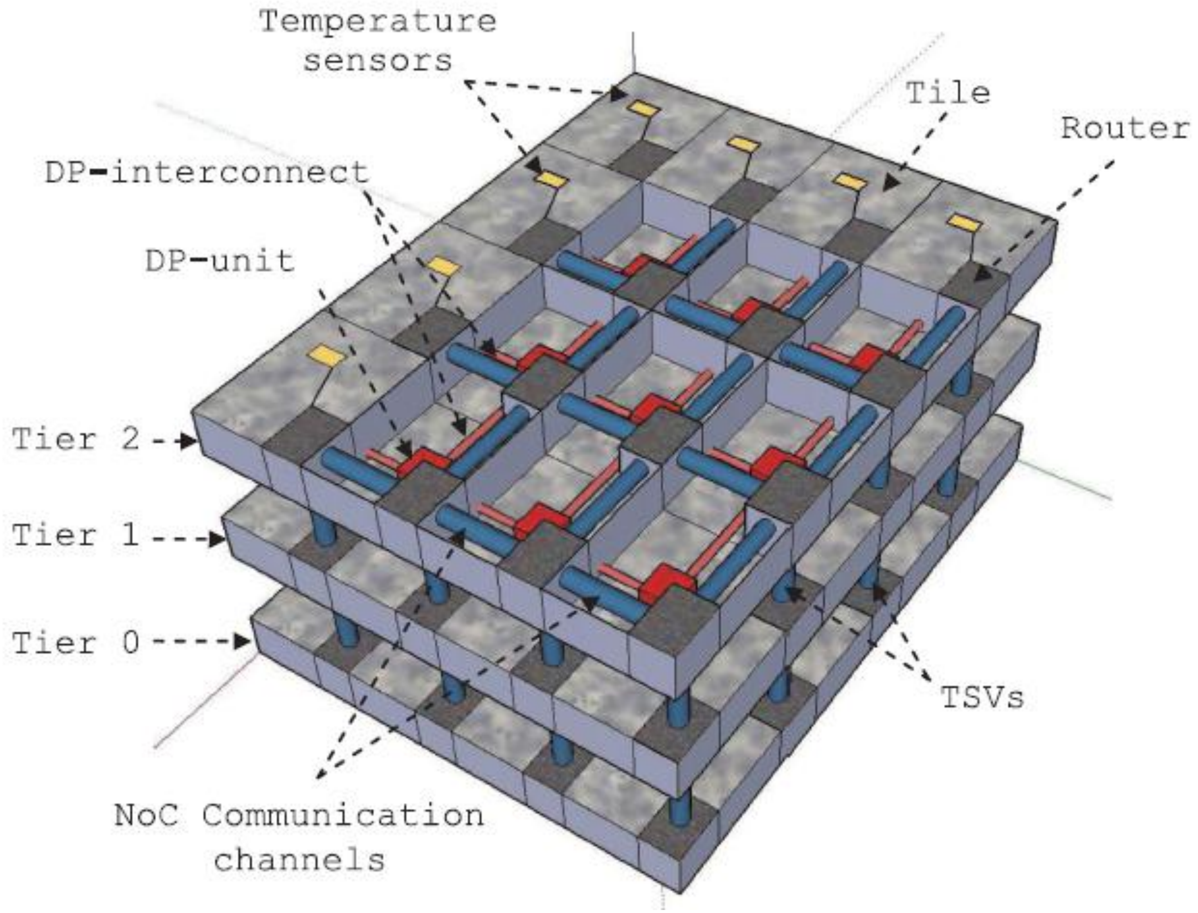
(b) Minimum energy mapping



(c) Random mapping

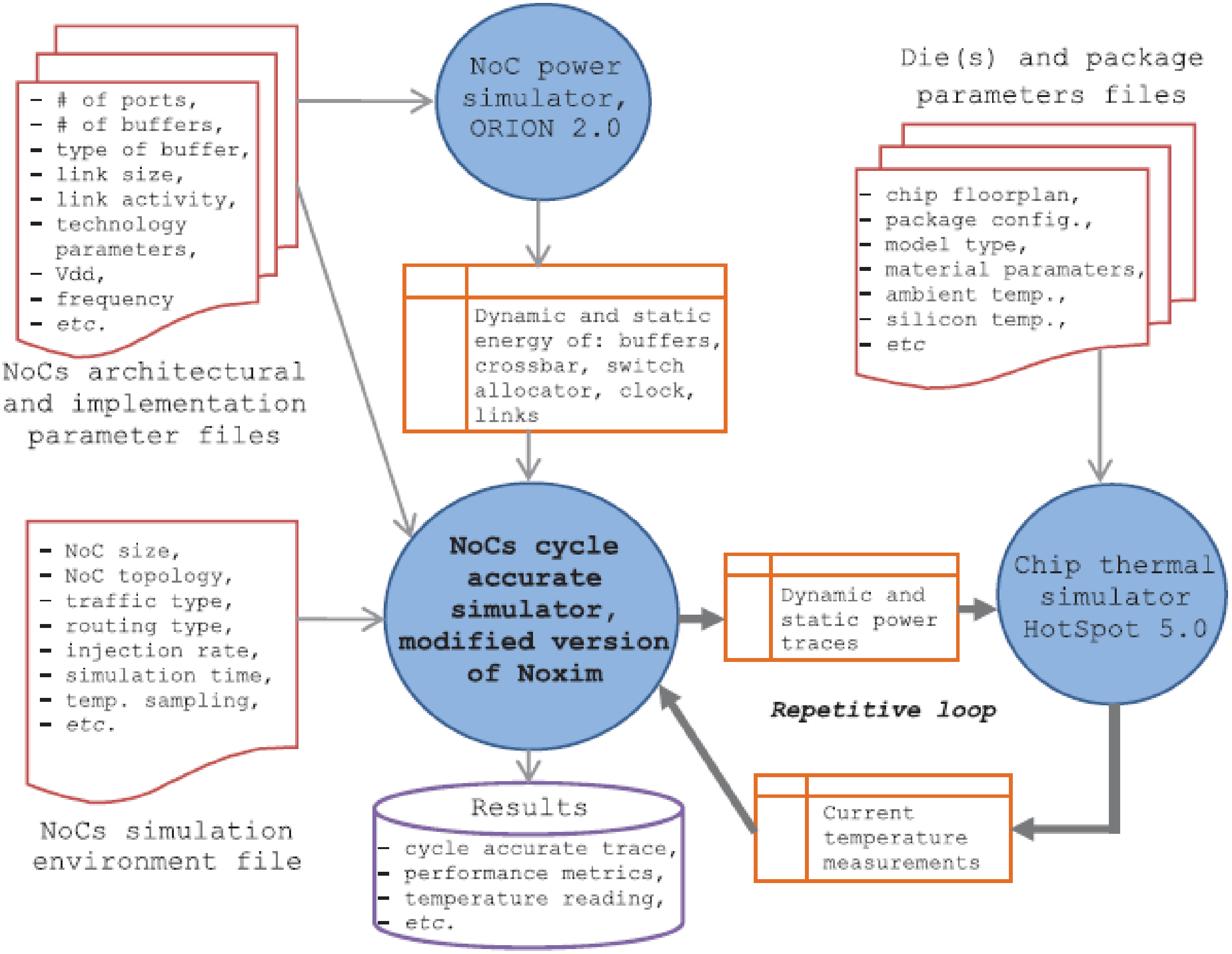
Thermal Sensing and Optimization

– On-chip dynamic programming network for thermal optimisation of 3D ship



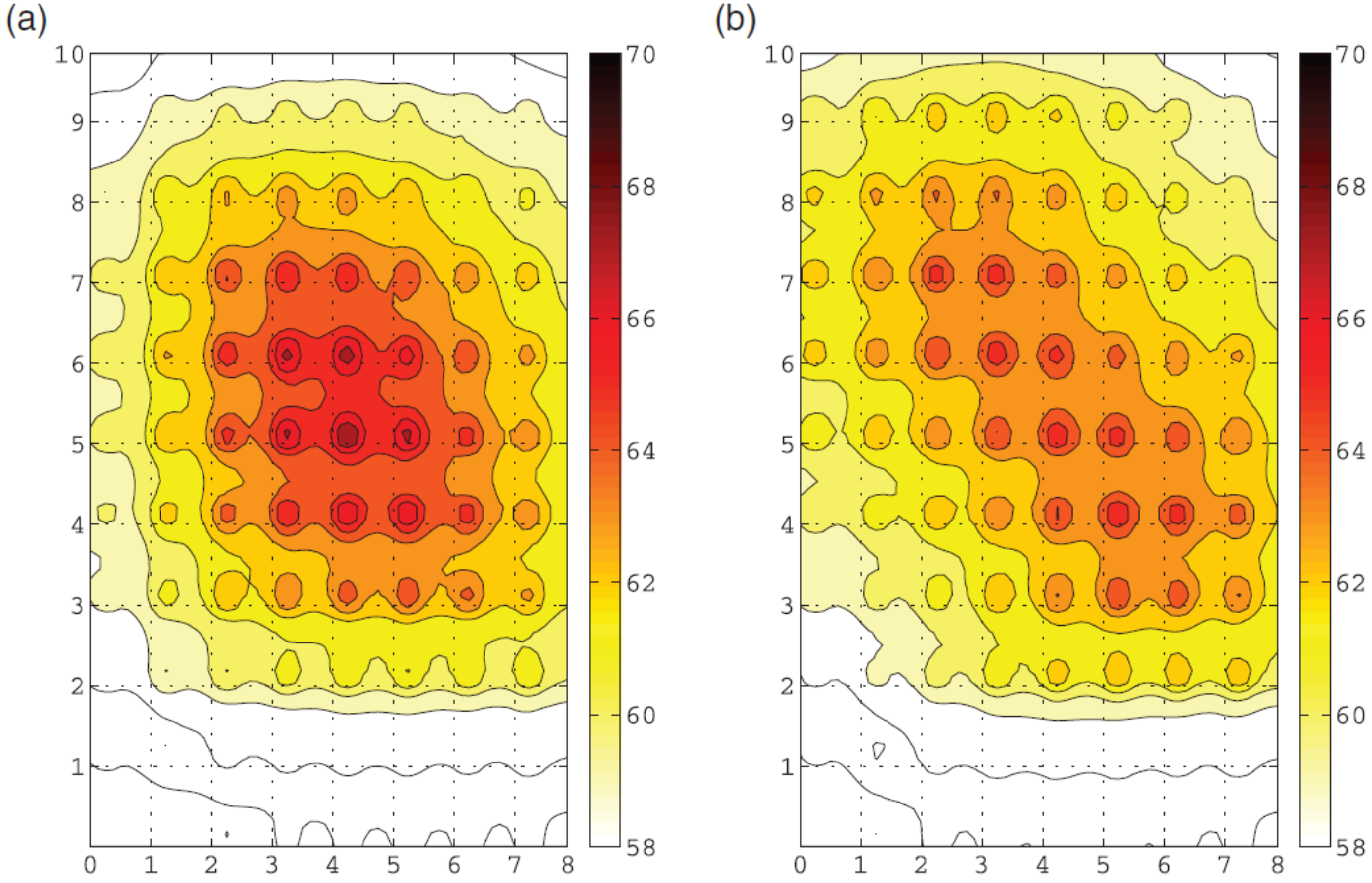
Thermal Sensing and Optimization

– Tool for thermal optimisation of 3D NoC – automated flow



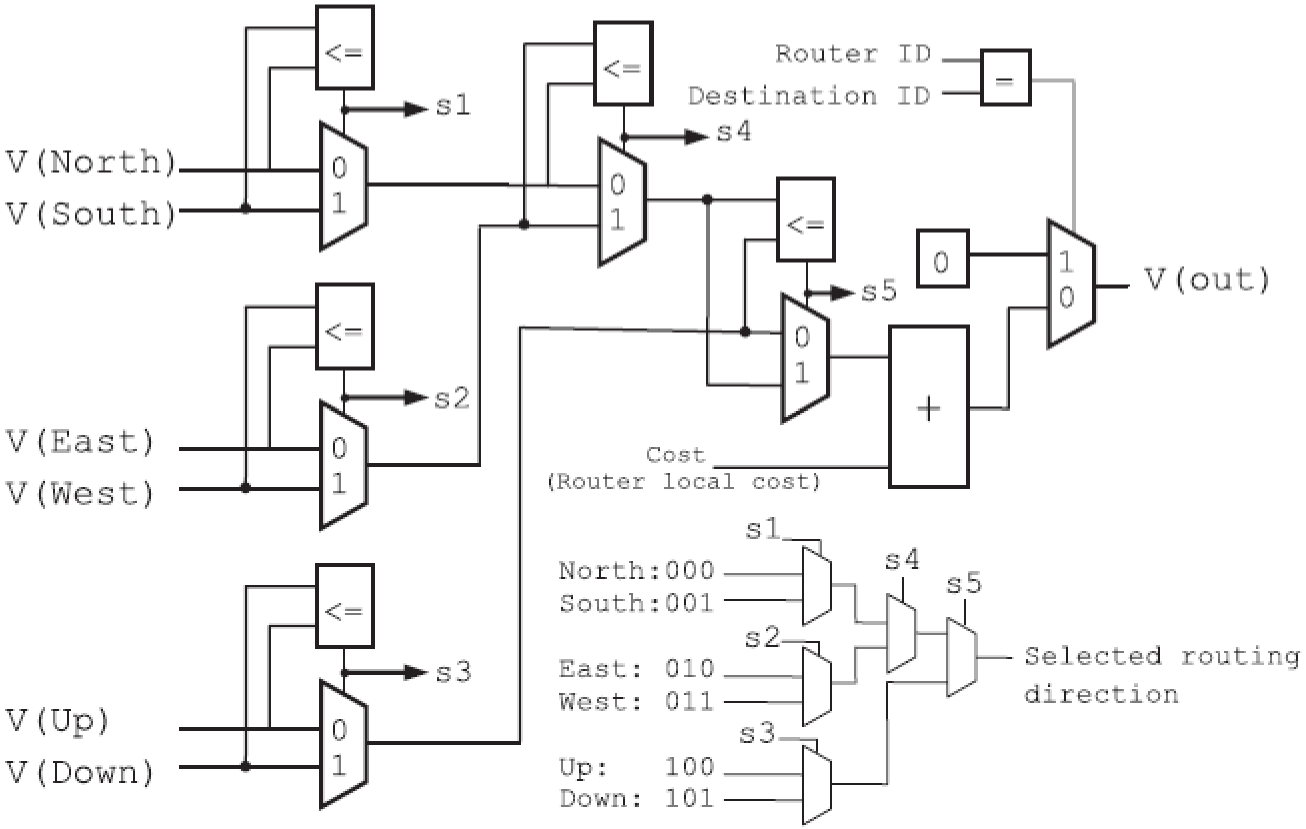
Thermal Sensing and Optimization

- Before and after for an 80-core model chip – hotspots reduced



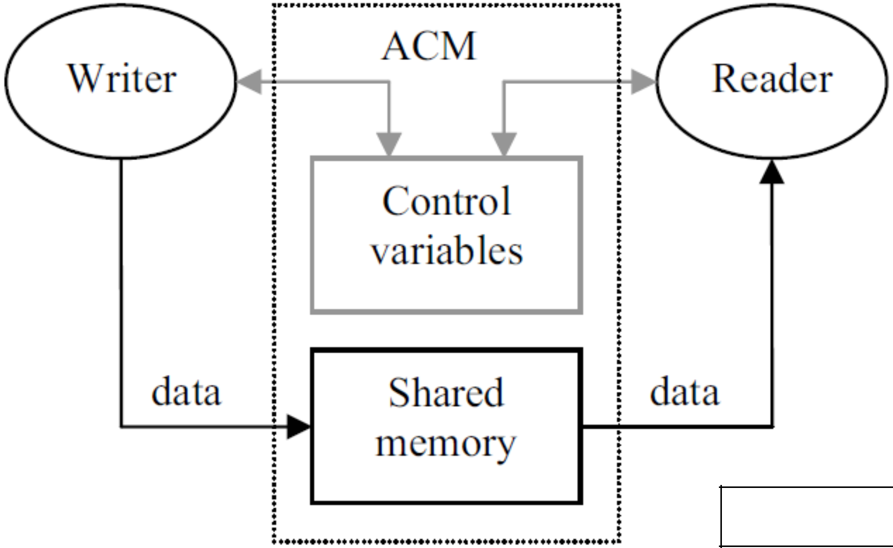
Thermal Sensing and Optimization

– DP unit to augment each router



Non-Blocking Communications

- Asynchronous communication mechanisms
  - A data-centric approach to data communication. Protocols determined by the type of data
  - Sensed and control data call for overwriting – newer data replace unused older data



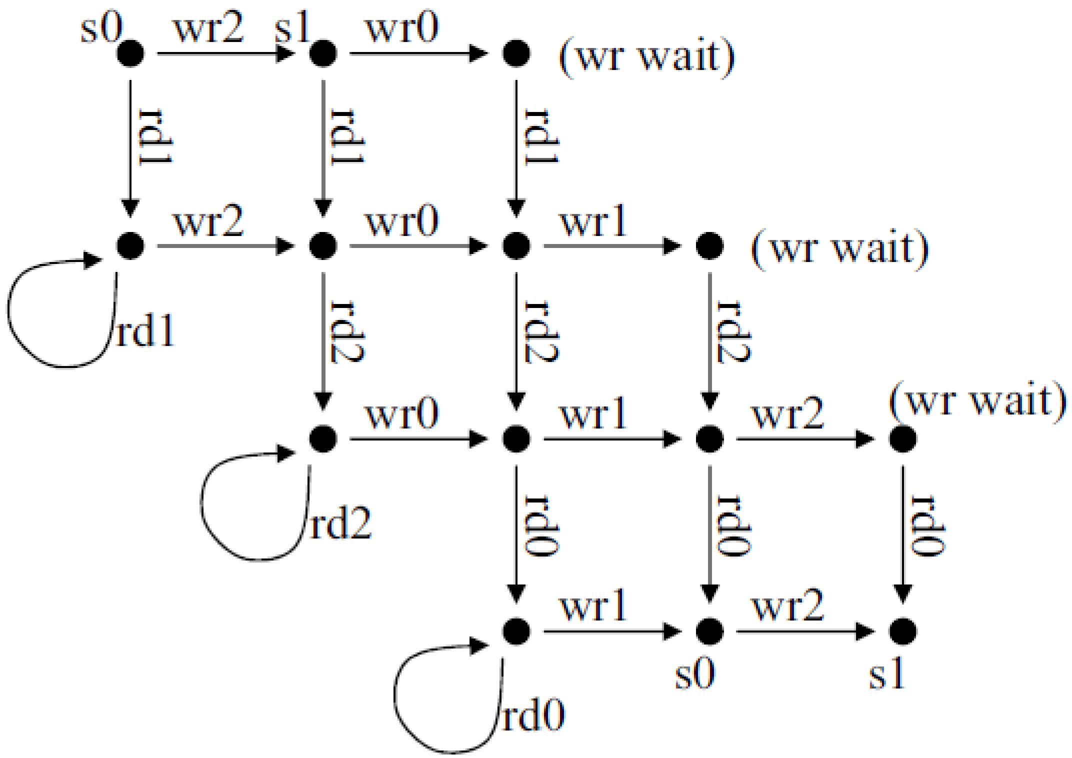
Writing and reading have their own timing and power conditions

Classification based on re-reading and overwriting

	NRR	RR
NOW	BB	RR-BB
OW	OW-BB	OW-RR-BB

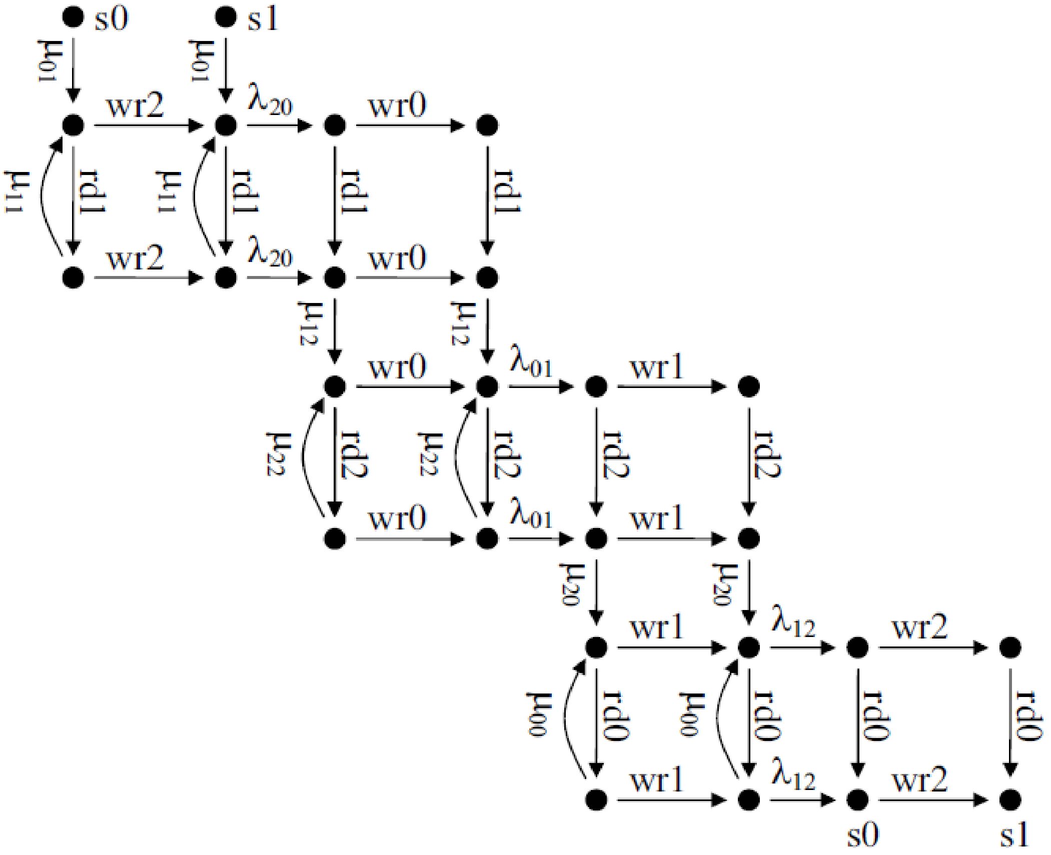
Non-Blocking Communications

- Asynchronous communication mechanisms
  - A 3-cell re-reading bounded buffer (RRBB)



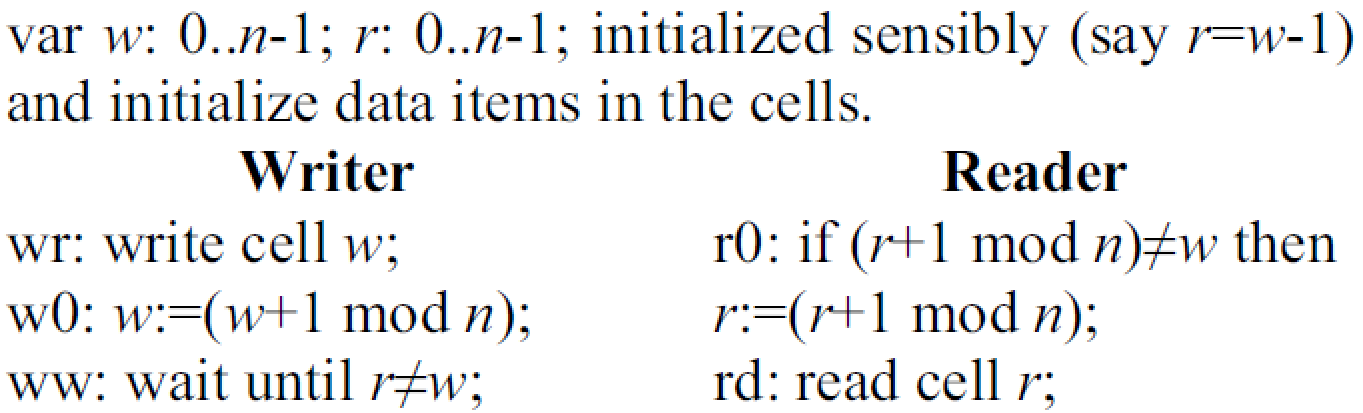
Non-Blocking Communications

- Asynchronous communication mechanisms
  - State graph with hidden actions



Non-Blocking Communications

- Asynchronous communication mechanisms
  - Synthesis from behaviour to state graph to Petri net models to algorithms to circuit implementations (HDD language programs)
  - ACM regions developed in Petri net synthesis theory
  - Example is the synthesis of n-cell RRBB from state graph model

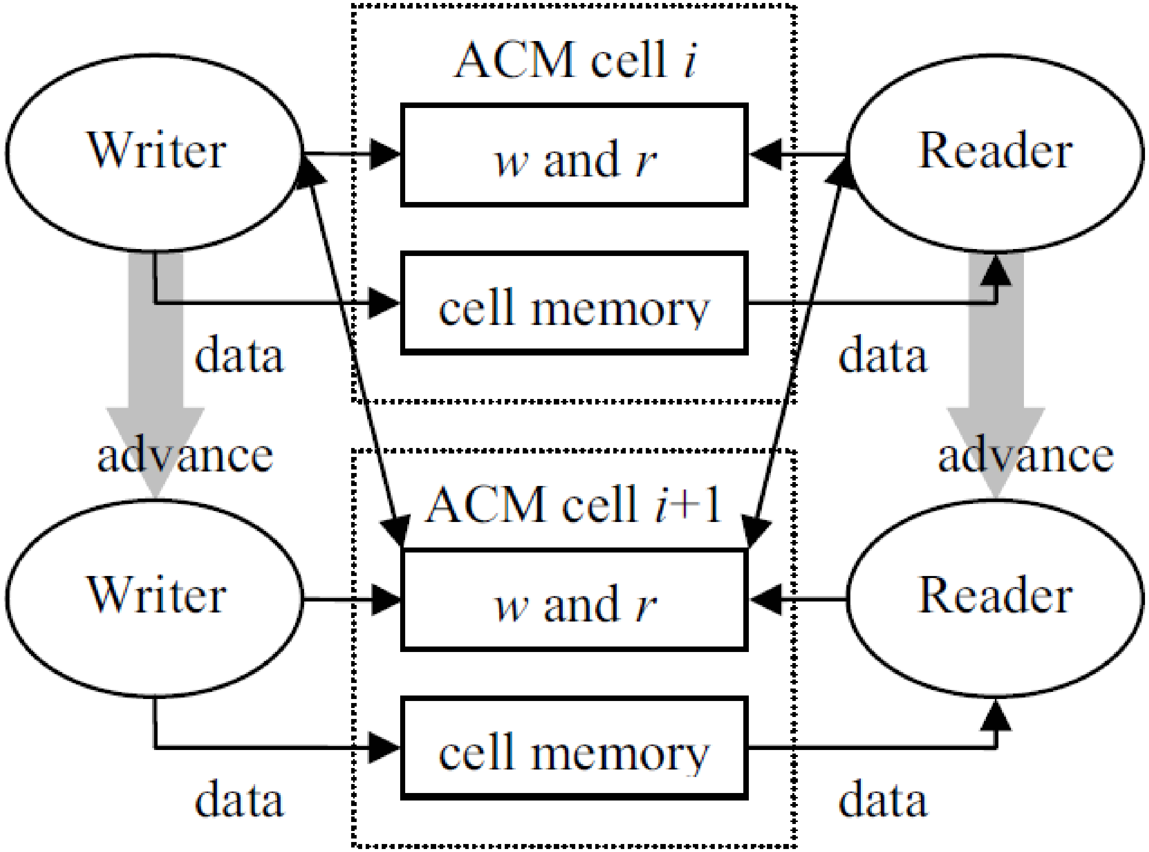


var w: 0..n-1; r: 0..n-1; initialized sensibly (say r=w-1) and initialize data items in the cells.

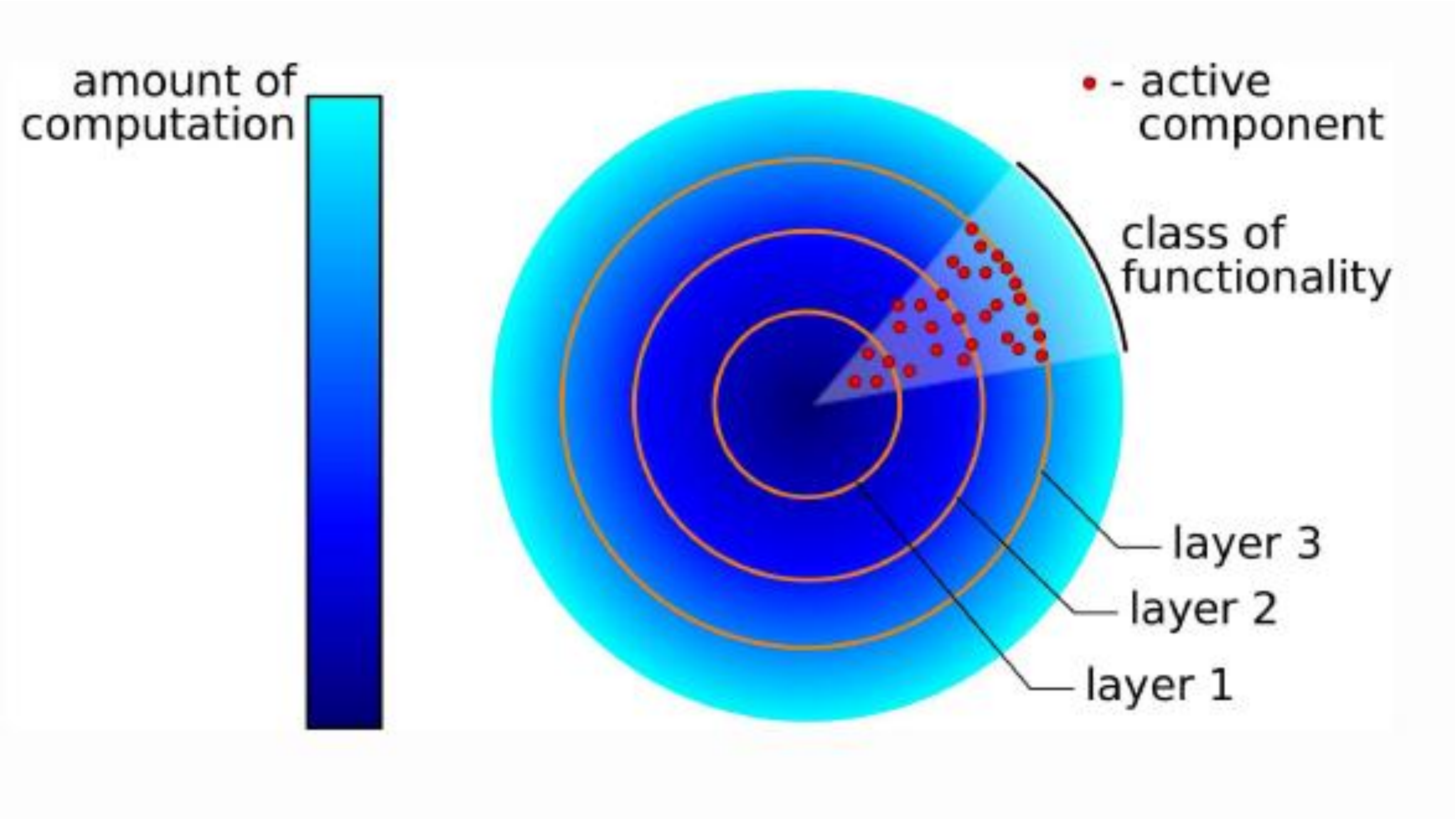
```
Writer
wr: write cell $w$;
w0: $w$:=($w$+1 mod $n$);
ww: wait until $r \neq w$;

Reader
r0: if ($r$+1 mod $n$)$\neq w$ then
$r$:=($r$+1 mod $n$);
rd: read cell $r$;
```

Non-Blocking Communications

- Asynchronous communication mechanisms
  - Modular design is possible: design a single cell ACM and expand to n cells through a process of linear expansion



Future developments: instincts and layers -> fabrics



Future developments

More diversified layers and inherent heterogeneity

- Power and data processing paths intertwined
- Digital and analogue fabrics
- Synchronous and asynchronous fabrics
- Multiple technology fabrics
- New design approaches – models that capture multi-modality and multi-layers
    - Combining structure and behaviour
    - Capturing overlay in functionality