Energy drives logic

Newcastle University

Asynchronous Design for IoT, or 25+ happy years in Newcastle and 100 Technical Reports on async.org.uk

Alex Yakovlev

uSystems Research Group

Newcastle University

ALIOT Summer School July 2018

Outline

- **There is no Outline today!!!**

People involved

- Patrick Degenaar (CANDO) – Optoneuro, Building implantable uSystems
- Andrey Mokhov (A4A, POETS) – asynchronous, concurrency, graph algebras, high performance computing
- Rishad Shafik (PRiME) – energy efficient many core, approximate computing, Machine learning
- Senior RAs and CoIs: Fei Xia (PRiME), Delong Shang (SAVVIE), Danil Sokolov (A4A)

OUR RESEARCH VISION:

- “REAL POWER” (ENERGY-MODULATED) COMPUTING

- LOW ENERGY: MIN ENERGY POINT

- RESILIENCE TO VARIABILITY

- POWER-COMPUTE CO-DESIGN

- “APPROXIMATE” COMPUTING

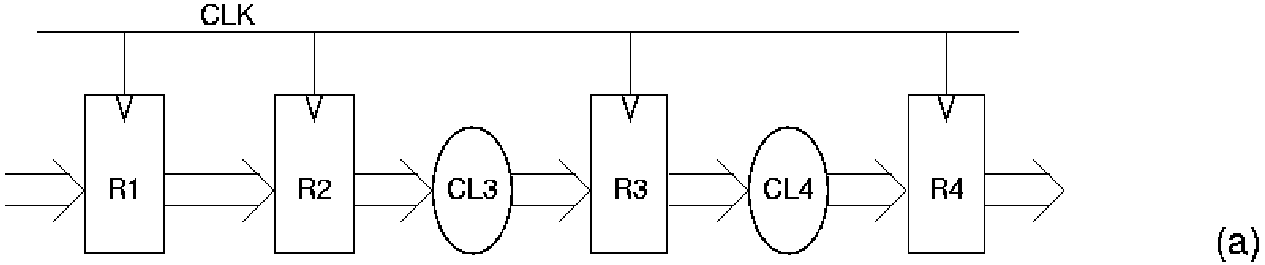
- MIXED SIGNAL DESIGN AUTOMATION

How it all began …

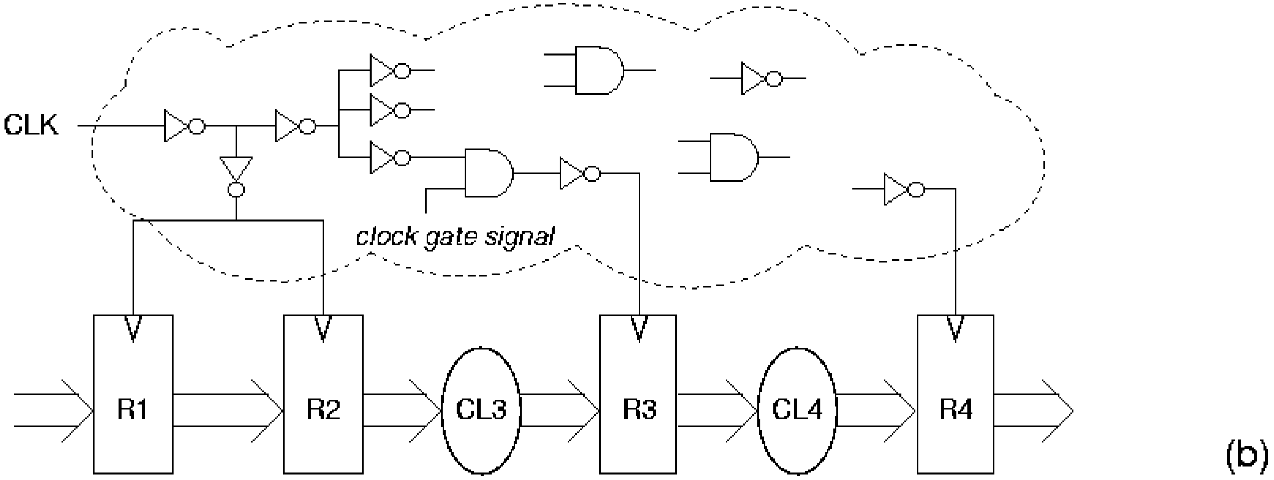
- Prior to that:
  - David Kinniment’s research on async processor designs and metastability and synchronsers in the 60s & 70s (Manchester)
  - My research on async design, models, interfaces in the 80s (St. Petersburg)
  - 1984-85 joint work (with Harry Whitfield, Albert Koelmans and Martin McLaughlin) on CAD for async (Newcastle) plus interactions with Brian Randell and Maciej Koutny on concurrency and dependability

Synchronous clocking

How we think



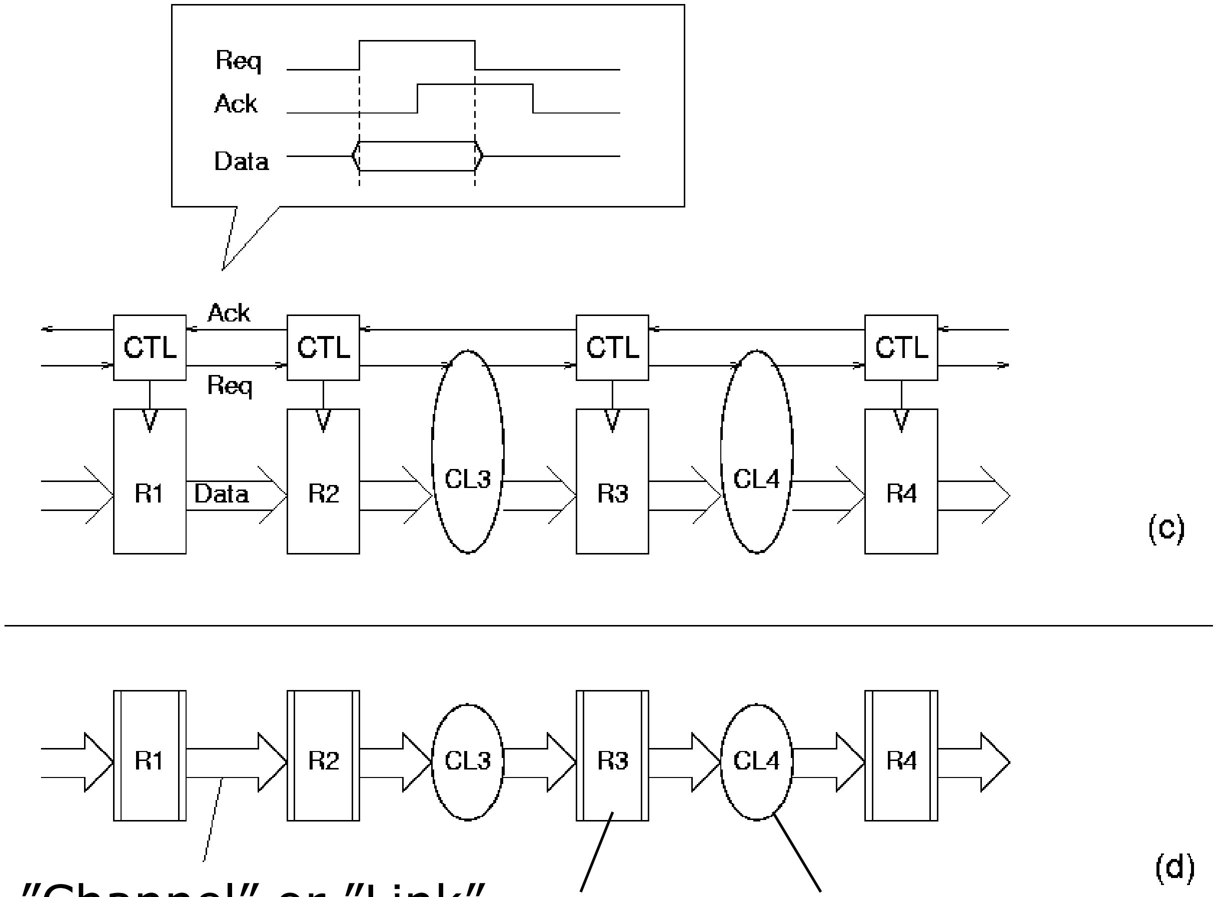
What we design



Asynchronous handshaking

What we design

How we think



”Channel” or ”Link”

Handshake CL

Handshake latch

Brief History of Asynchronous clocking

- In the 1950s there were no clocks in computers – they were ‘asynchronous’
- Theory of asynchronous switching circuits was created – David Muller et al from University of Illinois
- In the 1960s-70s circuits become more complex and designing them was easier with the clock
- In the 1980s VLSI circuits appeared and there began a battle between synchronous (circuits were complex to design and test; CAD tools started to appear based on clocked FSM paradigm) and asynchronous (interacting with the real world, metastability problems, working on average rather than worst case delays)

Brief History of Asynchronous clocking

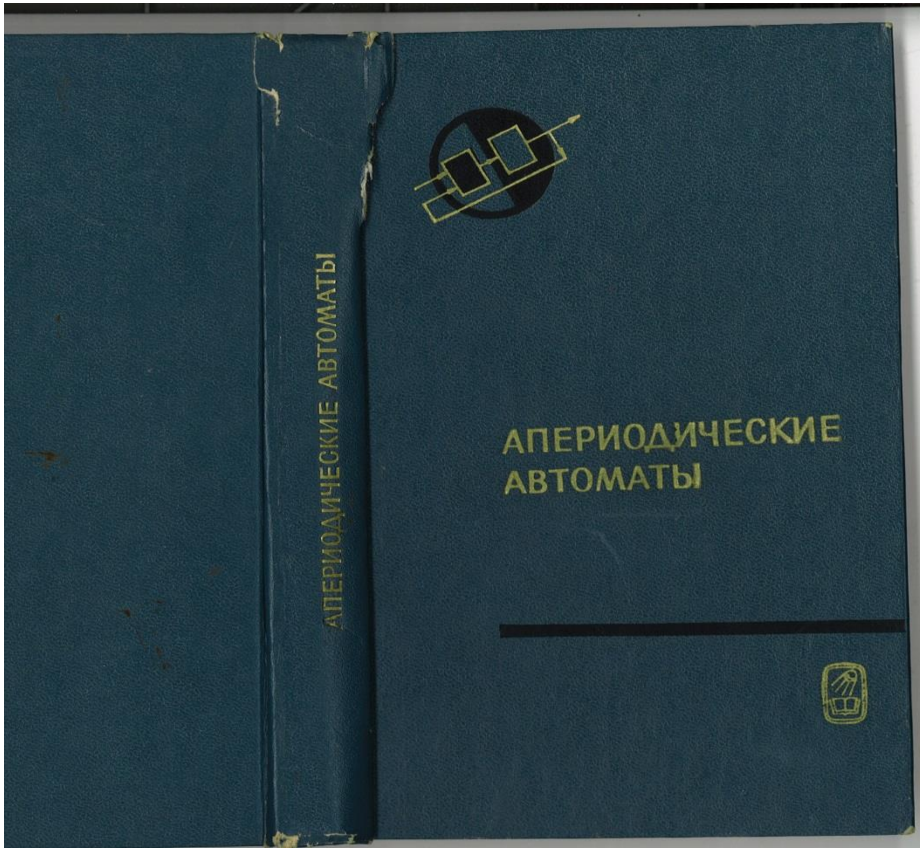
- In the 1990s drive for low power began and async gave some advantages; later the problem of design reuse came about and GALS approach was propose
- In the 2000s drive for robustness against PVT variations, security, noise and EMI reductions gave more drive to async
- In the 2010s multi Billion transistor chips design – many clock/Vdd domains, NOCS and GALS is a reality
- Push for ULTRA low power causes to work in near-threshold and subthreshold
- Analog-Mixed Signal comes on chip – needs little digital control – again asynchronous is important

LETI people in async world: back then

- Victor Varshavsky and his group
  - Early work of Varshavsky on digital systems and automata
  - 1970s: work in the USSR Academy of Sciences Institute of Economics and Mathematics (LOCEMI) and Institute of Social-Economic Problems (ISEP)
  - 1980: the group moves to LETI (MO EVM)
  - 1980s: the group develops methods for synthesis, analysis, new tools, first VLSI chips (FIFO in PLMs), multiprocessor systems (fault tolerant token ring interface for airborne onboard computers) -> first GALS system – 1986
  - Early 1990s: Trassa co-operative: libraries of modules, Trassa synthesis-analysis tools under MS DOS

Victor Varshavsky: pioneer of asynchronous design

The first book on asynchronous design



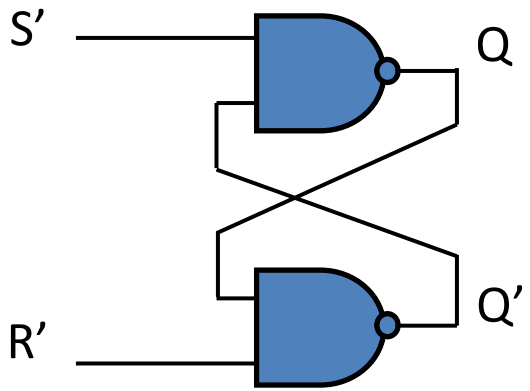
Aperiodic Automata, edited by V. Varshavsky, Nauka, Moscow, 1976

It contained:

Basic blocks with completion (flip-flops, registers, combinational)

Implementation of control (direct translation)

Asynchronous latches

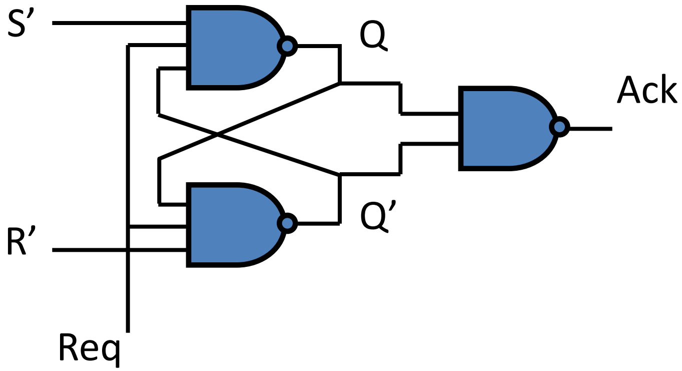


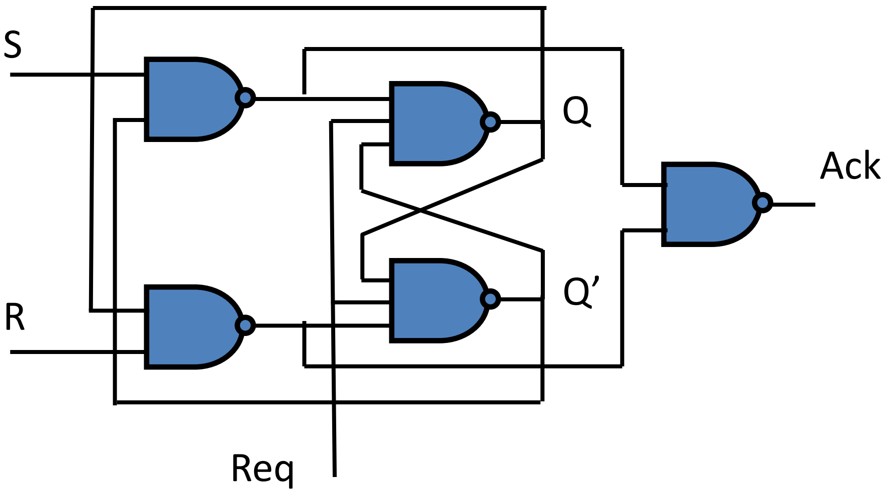
Classic RS-latch:
observing outputs cannot tell if
operation
is completed

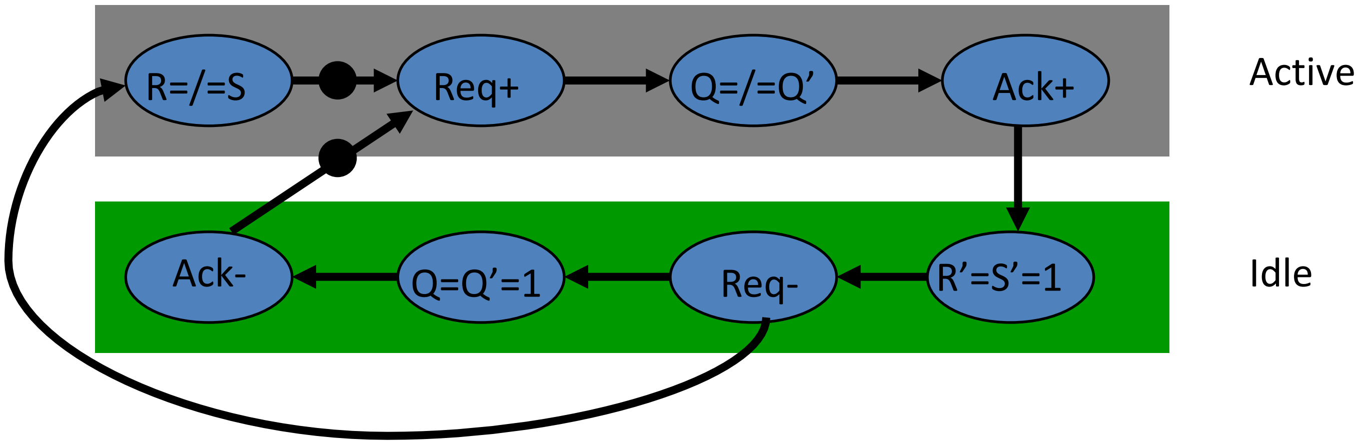
Completion detection

Partition computation into Idle and Active phases

Asynchronous latches







R=/=S

Req+

Q=/=Q’

Ack+

Active

Ack-

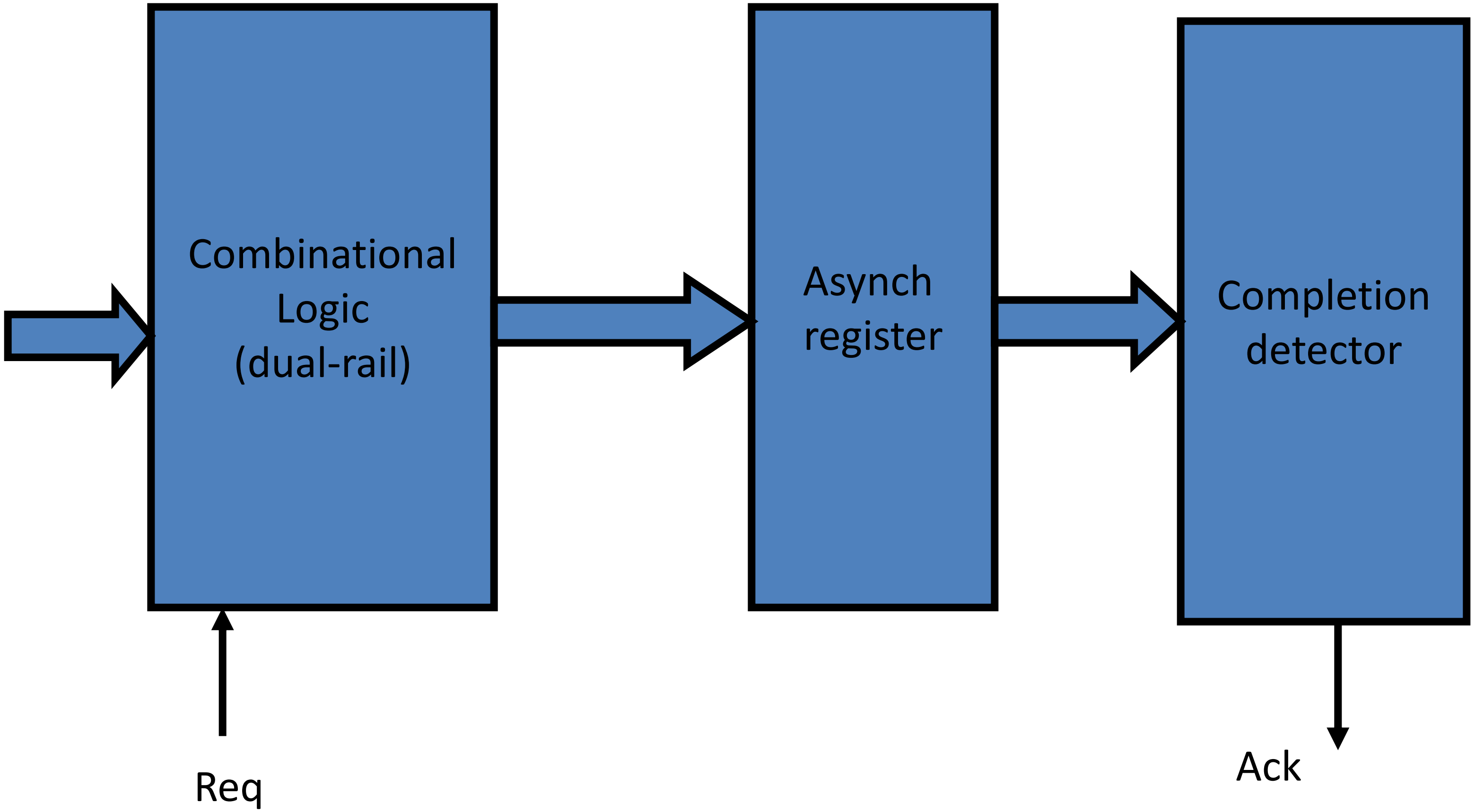
Q=Q’=1

Req-

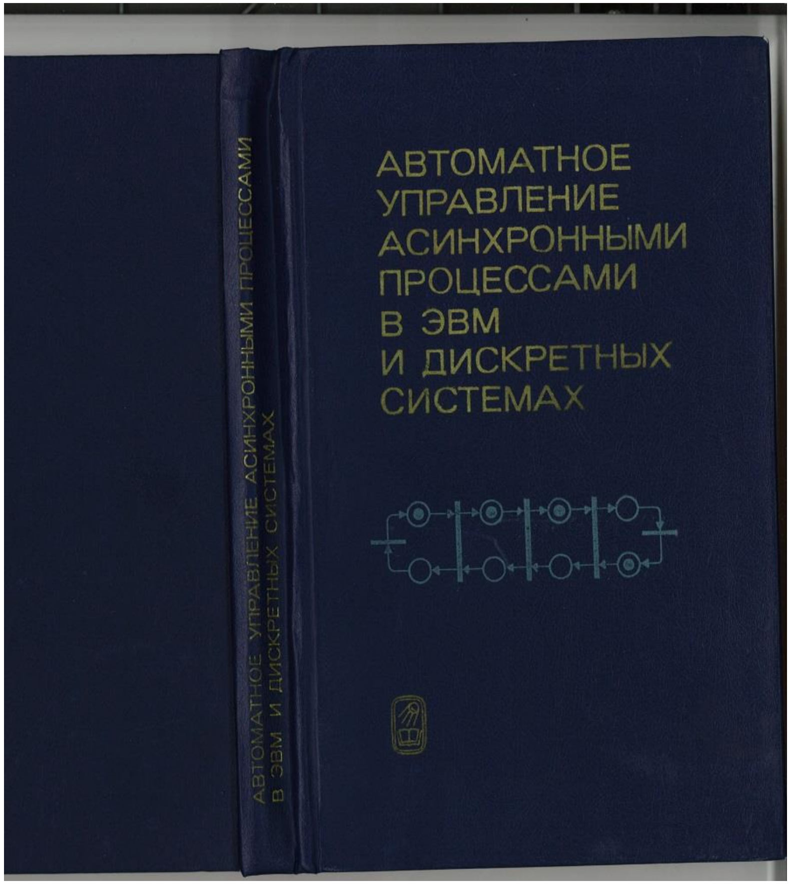
R’=S’=1

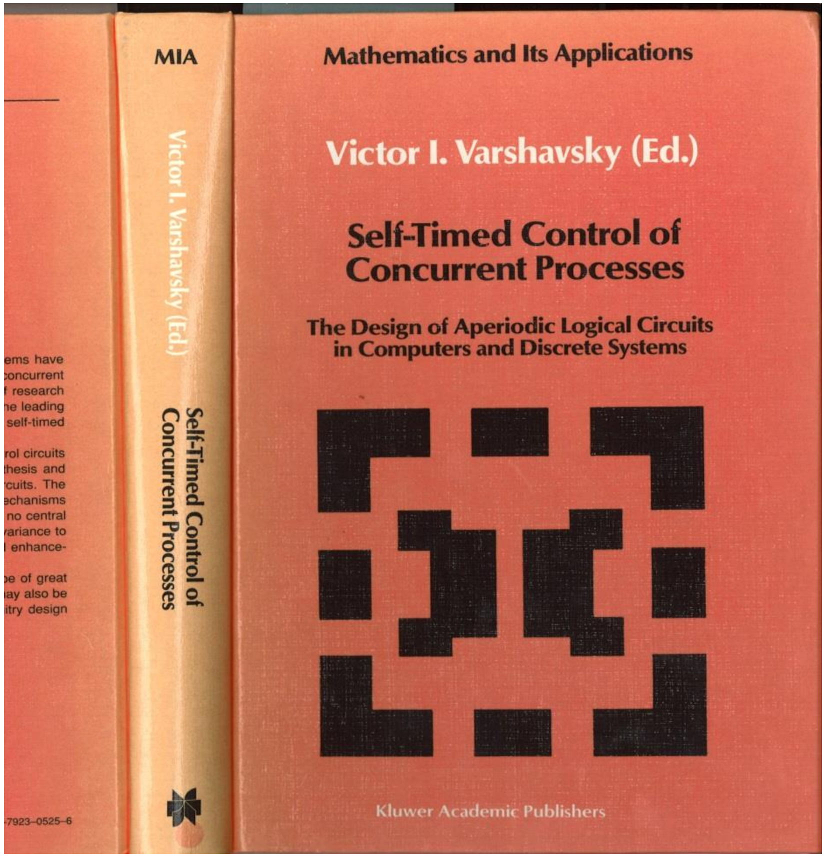
Idle

Two-phase asynchronous FSMs



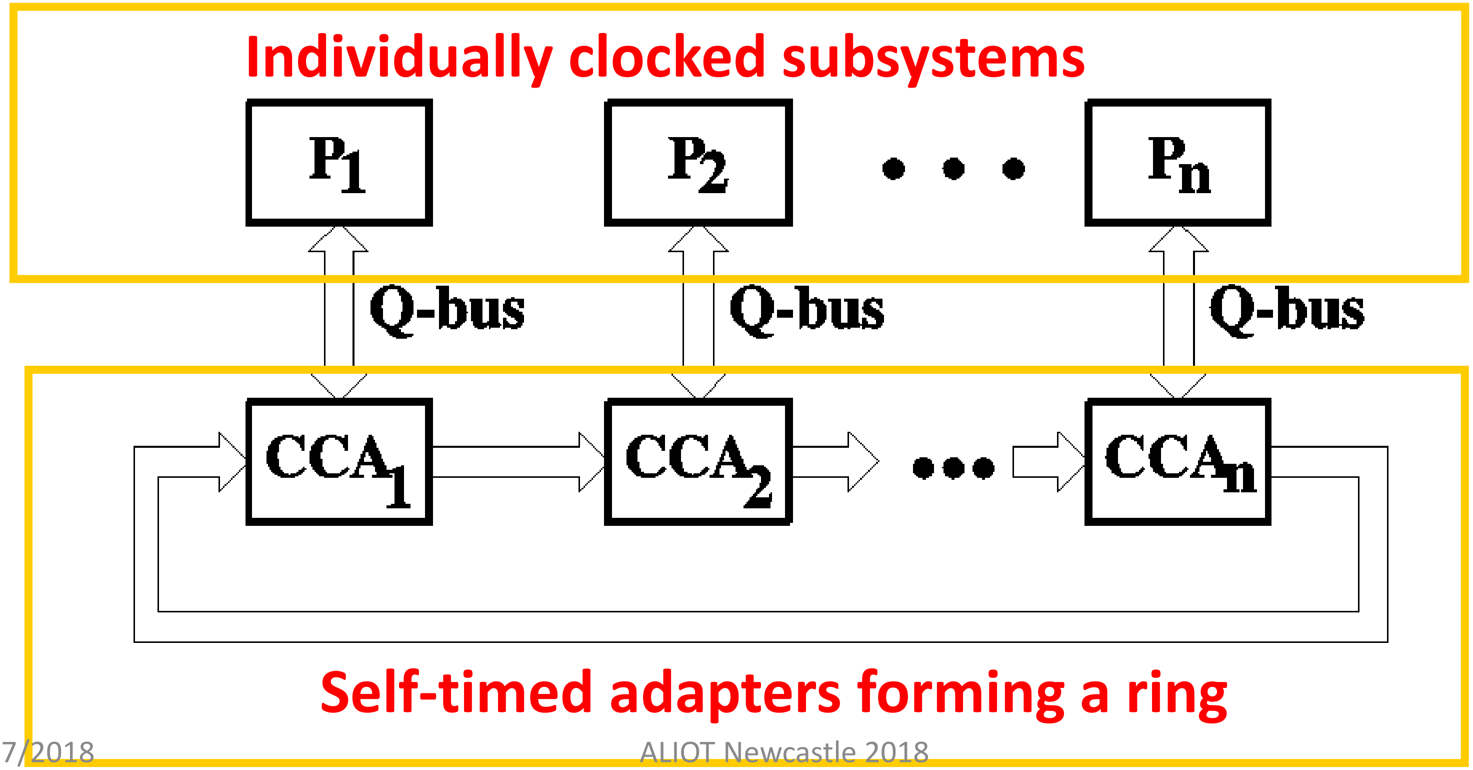
Second book (1986, 1990)


АВТОМАТНОЕ УПРАВЛЕНИЕ АСИНХРОННЫМИ ПРОЦЕССАМИ В ЭВМ И ДИСКРЕТНЫХ СИСТЕМАХ
АВТОМАТНОЕ УПРАВЛЕНИЕ АСИНХРОННЫМИ ПРОЦЕССАМИ В ЭВМ И ДИСКРЕТНЫХ СИСТЕМАХ


ems have
concurrent
f research
he leading
self-timed
rol circuits
thesis and
cuits. The
echanisms
no central
variance to
enhance-
e of great
ay also be
itry design
-7923-0525-6
MIA
Victor I. Varshavsky (Ed.)
Self-Timed Control of Concurrent Processes
Mathematics and Its Applications
Victor I. Varshavsky (Ed.)
Self-Timed Control of Concurrent Processes
The Design of Aperiodic Logical Circuits in Computers and Discrete Systems
Kluwer Academic Publishers

Fault-Tolerant Self-Timed Ring(1981-1986)

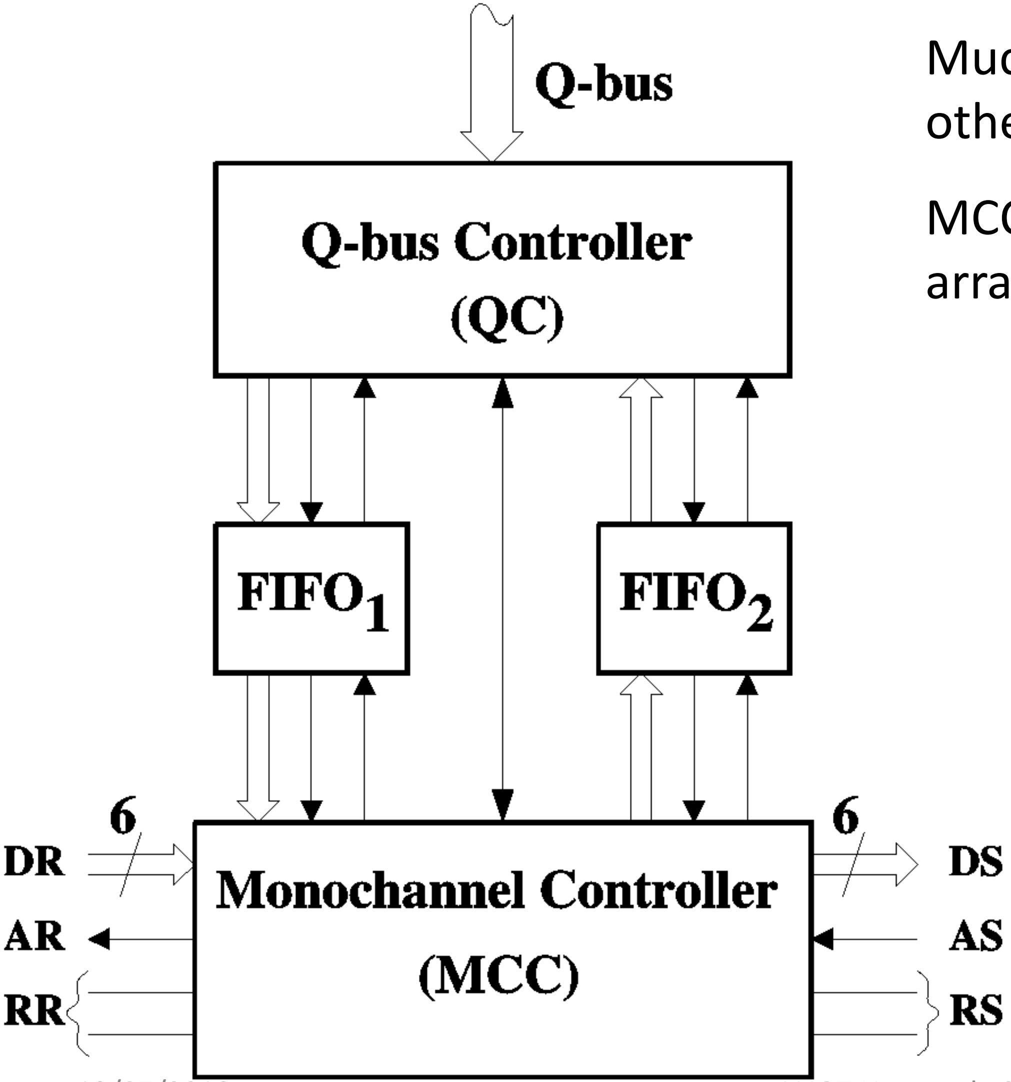
*For an on-board airborne computer-control system which tolerates up to two faults. Self-timed ring was a **GALS system** with **self-checking and self-repair at the hardware level***



Individually clocked subsystems

Self-timed adapters forming a ring

Communication Channel Adapter



Much higher reliability than a bus and other forms of redundancy

MCC was developed TTL-Schottky gate arrays, approx 2K gates.

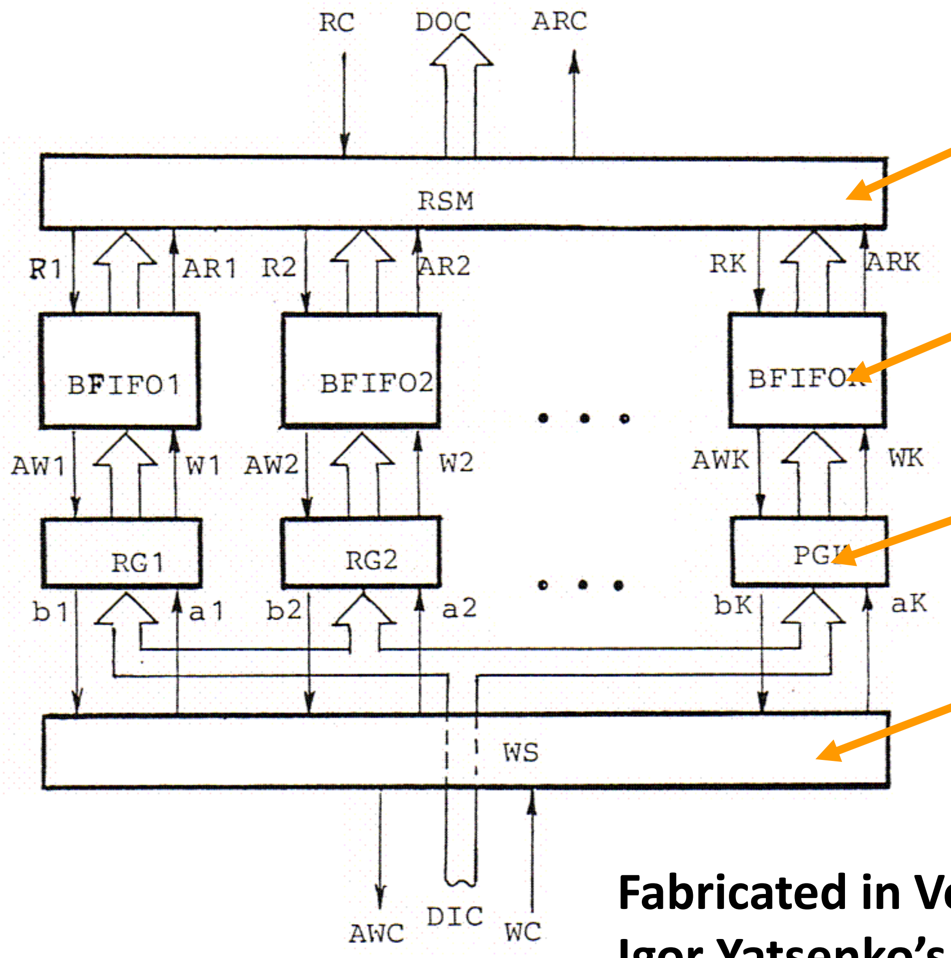
Data (DR,DS) is encoded using 3-of-6 Sperner code (16 data values for half-byte, plus 4 tokens for ring acquisition protocol)
AR, AS – acknowledgements
RR, RS – spare (for self-repair) lines

Self-Timed Ring published

- V.I. Varshavsky, V.B. Marakhovsky, L.Ya. Rosenblum, Yu.S. Tatarinov, V.Ya. Volodarskii, A.V. Yakovlev, **THREE papers** in Avtomatika & Vychislitelnaya Tekhnika, Control and Computer Science (Translated from Russian, Alerton Press), (1988-1989)
- A. Yakovlev, V. Varshavsky, V. Marakhovsky and A. Semenov, "Designing and asynchronous pipeline token ring interface", Proc. 2nd Working Conf. on Asynch. Design Methodologies, London, May 1995, IEEE CS Press, N.Y., pp. 32-41, 1995.

Self-Timed FIFO (1985-1988)

Basic FIFO are connected using a wagging-tail buffer method



Read mux

Basic FIFO

Buffer registers

Write demux

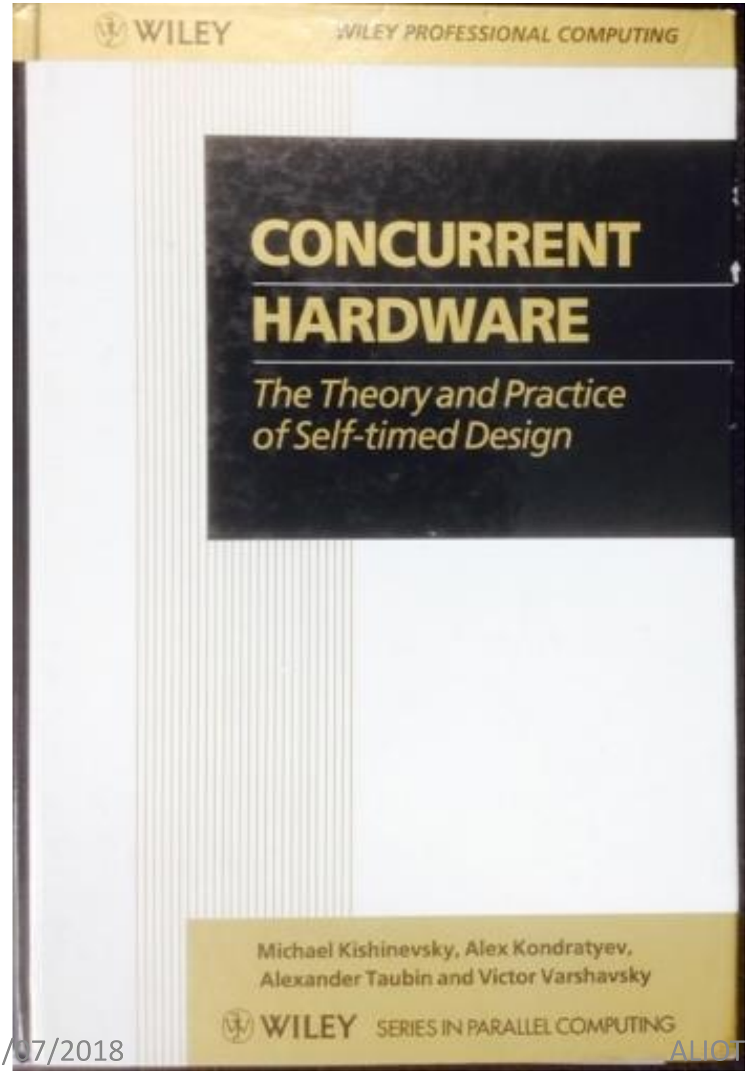
Throughput doesn't depend on the delay of Basic FIFO.

Fabricated in Voronezh, and fully described in Igor Yatsenko's PhD thesis, 1988

LETI people in async world: recent past

- From 1993: Varshavsky is in Univ of Aizu in Japan with part of his group (Marakhovsky, Kishinevsky, Kondratyev, Taubin)
- From 1990: Rosenblum is in the USA
- From 1991: Yakovlev is in the UK
- Starodoubtsev is in Russia (NTO of Academy of Sciences)
- Active work on synthesis and verification methods for STGs
- Collaboration between Yakovlev, Kishinevsky, Kondratyev, Taubin and Cortadella (Barcelona) and Lavagno (Turin)
  - Many papers in IEEE Transactions and leading confs: ASYNC, DAC
  - Software tool Petrify
  - Springer book published in 2002

Group’s Books on Async Design



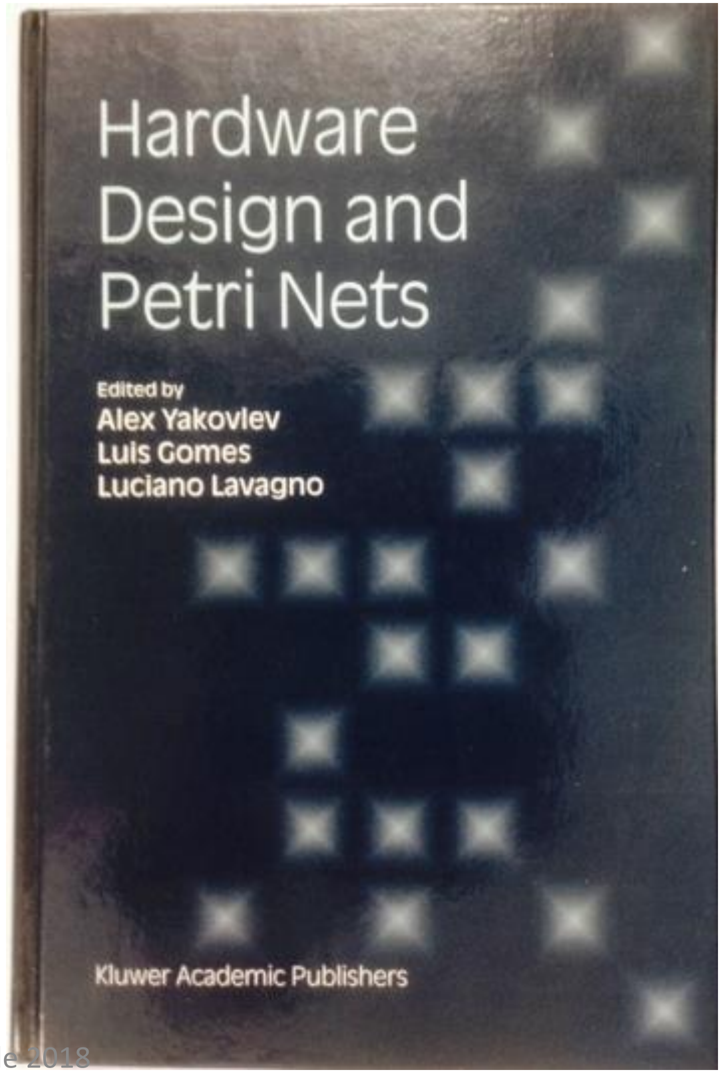
WILEY WILEY PROFESSIONAL COMPUTING

CONCURRENT HARDWARE

The Theory and Practice of Self-timed Design

Michael Kishinevsky, Alex Kondratyev, Alexander Taubin and Victor Varshavsky

WILEY SERIES IN PARALLEL COMPUTING

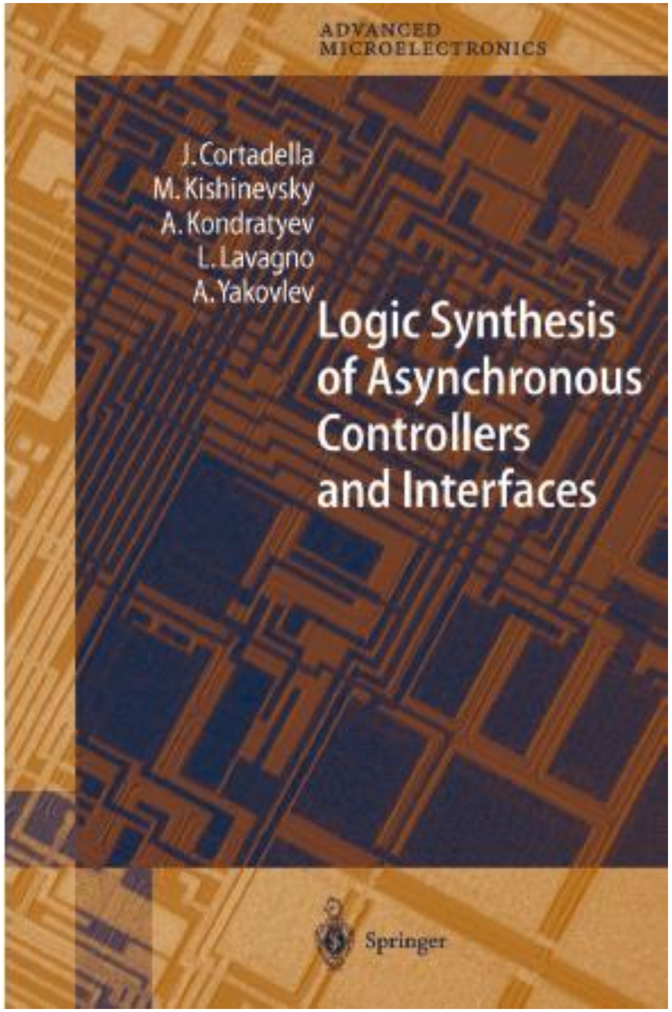


Hardware Design and Petri Nets

Edited by
Alex Yakovlev
Luis Gomes
Luciano Lavagno

Kluwer Academic Publishers

Book on Petrify Method and Tools



ADVANCED MICROELECTRONICS

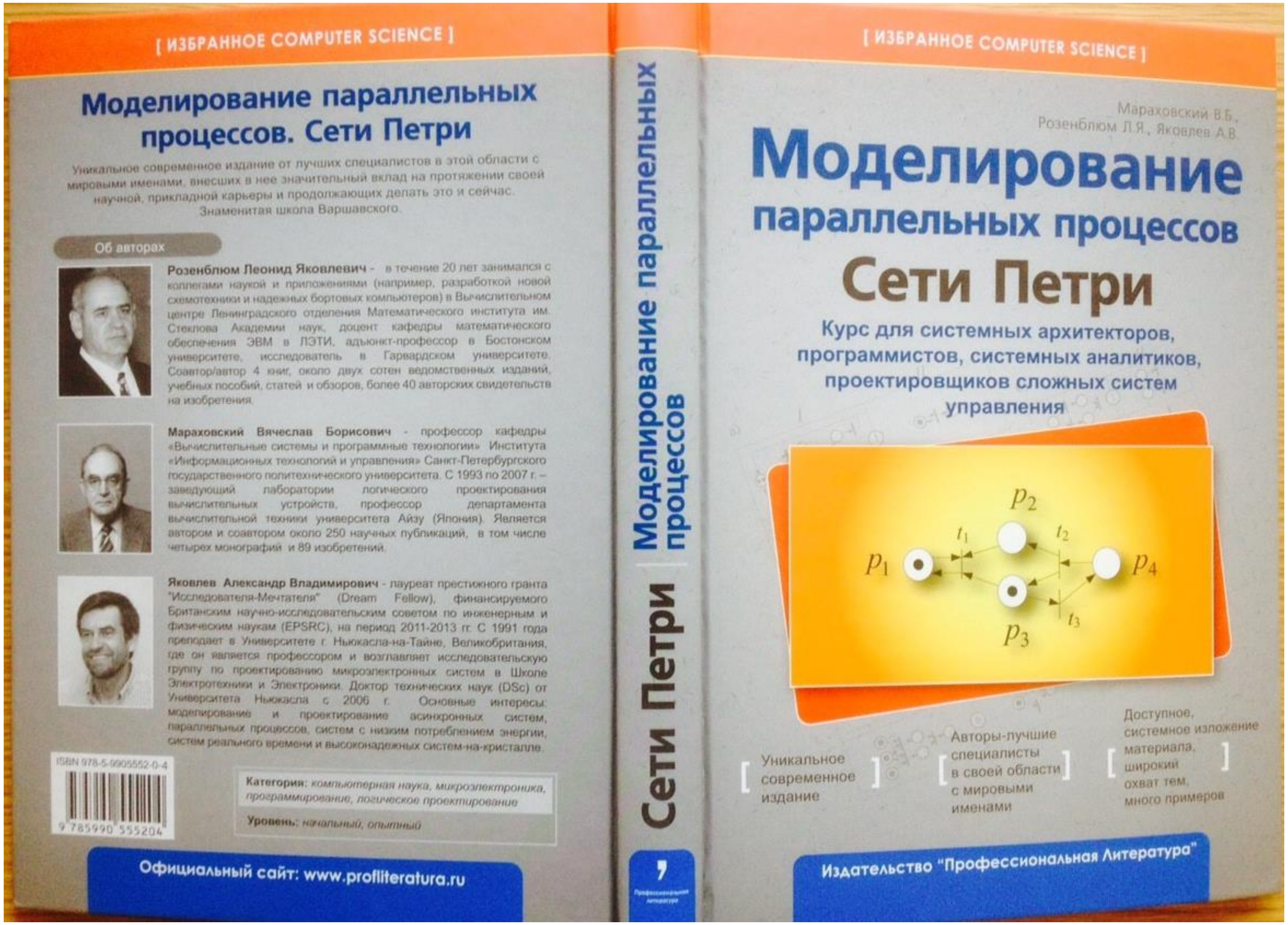
J. Cortadella
M. Kishinevsky
A. Kondratyev
L. Lavagno
A. Yakovlev

Logic Synthesis of Asynchronous Controllers and Interfaces

Springer

- The book on Signal Graphs Methods and tools.
- Collaboration with Cortadella (UPC Barcelona) and Lavagno (Turin, Cadence)
- Kishinevsky (Intel), Kondratyev (Theseus Logic, Cadence)

The most recent book from the group



[ИЗБРАННОЕ COMPUTER SCIENCE]

Моделирование параллельных процессов. Сети Петри

Уникальное современное издание от лучших специалистов в этой области с мировыми именами, внесших в нее значительный вклад на протяжении своей научной, прикладной карьеры и продолжающих делать это и сейчас. Знаменитая школа Варшавского.

Об авторах

Розенблюм Леонид Яковлевич - в течение 20 лет занимался с коллегами наукой и приложениями (например, разработкой новой схемотехники и надежных бортовых компьютеров) в Вычислительном центре Ленинградского отделения Математического института им. Стеклова Академии наук, доцент кафедры математического обеспечения ЭВМ в ЛЭТИ, адъюнкт-профессор в Бостонском университете, исследователь в Гарвардском университете. Соавтор/автор 4 книг, около двух сотен ведомственных изданий, учебных пособий, статей и обзоров, более 40 авторских свидетельств на изобретения.

Мараховский Вячеслав Борисович - профессор кафедры «Вычислительные системы и программные технологии» Института «Информационных технологий и управления» Санкт-Петербургского государственного политехнического университета. С 1993 по 2007 г. – заведующий лаборатории логического проектирования вычислительных устройств, профессор департамента вычислительной техники университета Айзу (Япония). Является автором и соавтором около 250 научных публикаций, в том числе четырех монографий и 89 изобретений.

Яковлев Александр Владимирович - лауреат престижного гранта "Исследователя-Мечтателя" (Dream Fellow), финансируемого Британским научно-исследовательским советом по инженерным и физическим наукам (EPSRC), на период 2011-2013 гг. С 1991 года преподает в Университете г. Ньюкасла-на-Тайне, Великобритания, где он является профессором и возглавляет исследовательскую группу по проектированию микроэлектронных систем в Школе Электротехники и Электроники. Доктор технических наук (DSc) от Университета Ньюкасла с 2006 г. Основные интересы: моделирование и проектирование асинхронных систем, параллельных процессов, систем с низким потреблением энергии, систем реального времени и высоконадежных систем-на-кристалле.

ISBN 978-5-9905552-0-4
9 785990 555204

Категория: *компьютерная наука, микроэлектроника, программирование, логическое проектирование*

Уровень: *начальный, опытный*

Официальный сайт: www.profliteratura.ru

Моделирование параллельных процессов

Сети Петри

[ИЗБРАННОЕ COMPUTER SCIENCE]

Мараховский В.Б.,
Розенблюм Л.Я., Яковлев А.В.

Моделирование параллельных процессов

Сети Петри

Курс для системных архитекторов, программистов, системных аналитиков, проектировщиков сложных систем управления

[Уникальное современное издание]

[Авторы-лучшие специалисты в своей области с мировыми именами]

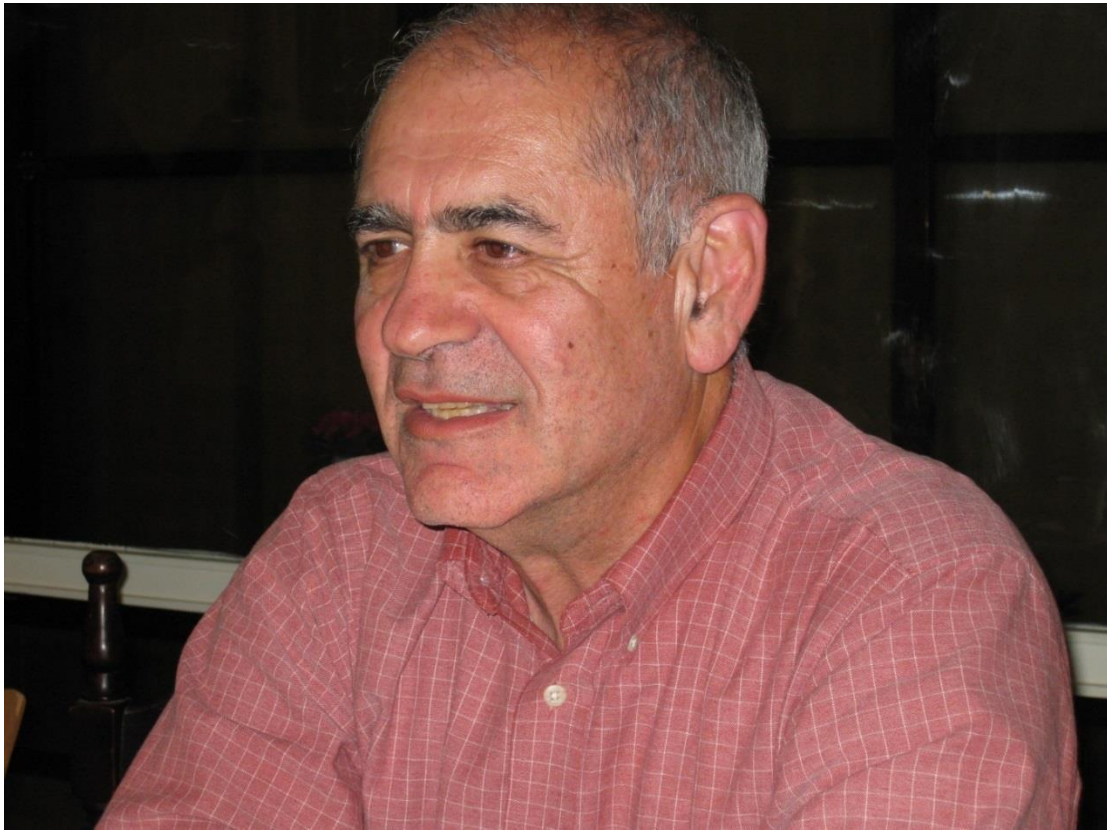
[Доступное, системное изложение материала, широкий охват тем, много примеров]

Издательство "Профессиональная Литература"

Some ‘remainders’ of Varshavsky’s group

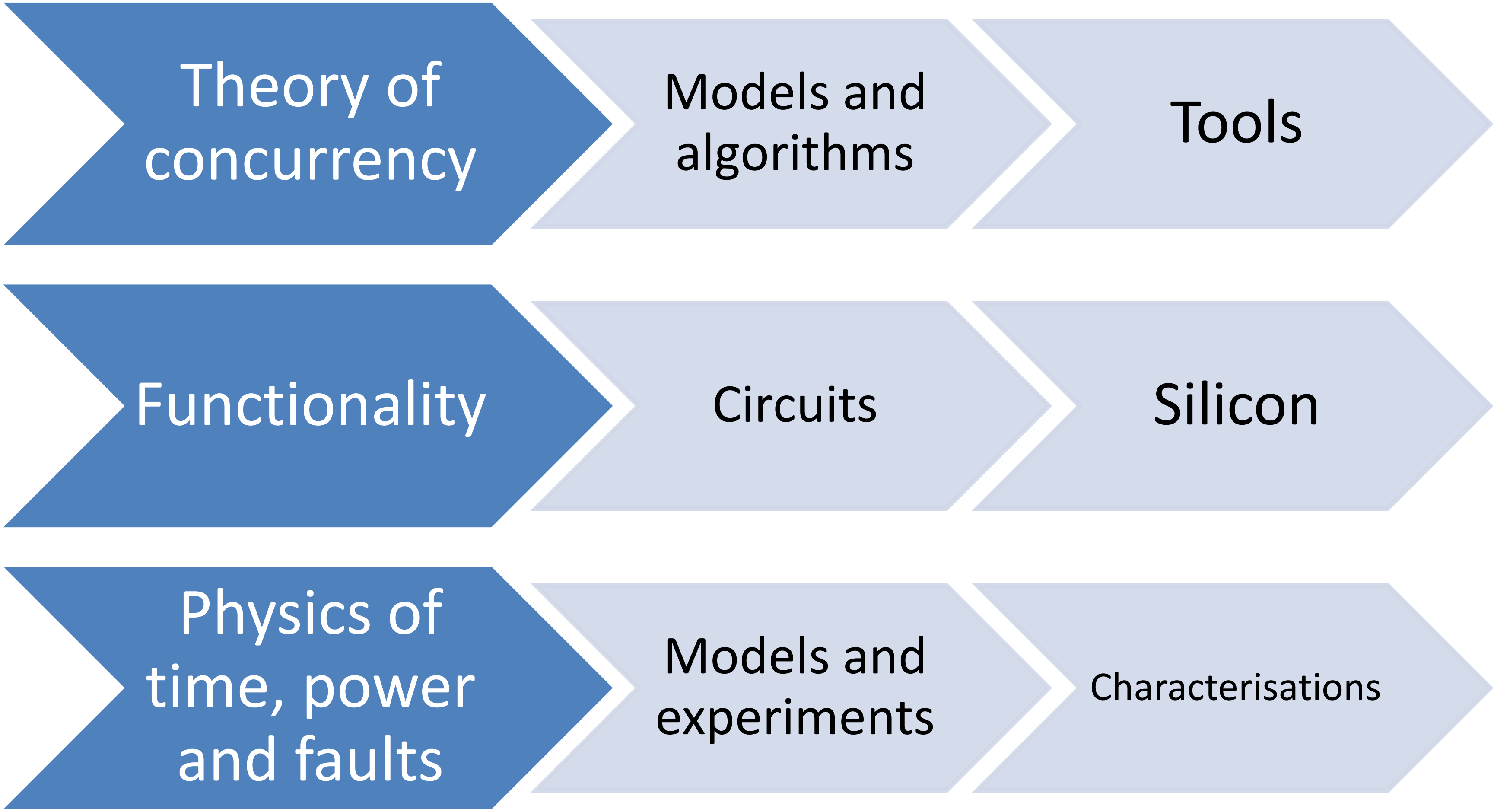
Victor Varshavsky with members of his group at our reunion in Eilat, Israel, in 2000; left to right: Alex Kondratyev, Alexander Taubin, Michael Kishinevsky, Victor Varshavsky, Alex Yakovlev (me) and Maria Yakovlev(my wife).

Leonid Rosenblum



Leonid Rosenblum, who introduced Petri nets into asynchronous circuit modelling and popularized Petri nets in USSR in the 70s and 80s.

Three main research threads



Theory of concurrency

Models and algorithms

Tools

Functionality

Circuits

Silicon

Physics of time, power and faults

Models and experiments

Characterisations

But how has it all unrolled in Newcastle … from 1990s

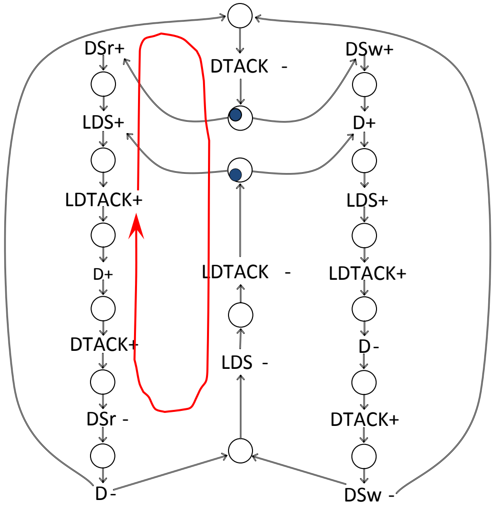
The key element for success

- Work of a team where all aspects of talent have a chance to develop and flourish
- The team – both locally (Newcastle) and globally (worldwide)
- Async research is a wonderful area which combines the three main threads and different people can exploit the three main categories of strengths:
  - Computer programming
  - Electronic (Circuits and Systems) design
  - Analytical

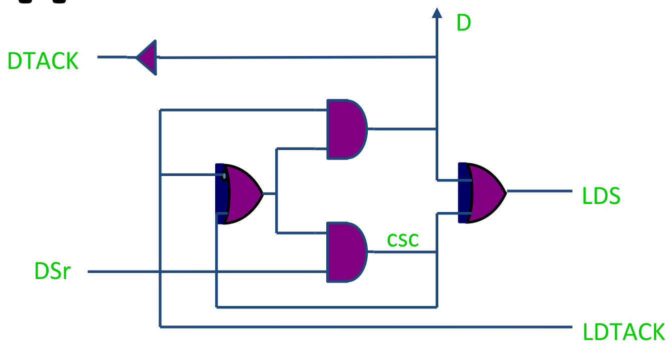
1990s: Logic Synthesis of Asynchronous Control

- Formalisation of STGs and relation to state-based models; theory, models and tools for async controller synthesis, async processor prototyping
- Projects: ASAP, ASTI, ACiD-WG
- People: Jordi Cortadella, Luciano Lavagno, Alex Kondratyev, Mike Kishinevsky, Alexander Taubin, Albert Koelmans, Alexei Petrov, Alex Semenov, Enric Pastor, Lee Lloyd, Marta Pietkiewicz-Koutny
- Some highlights: synthesis flow, monotonic cover, CSC solving, regions for async, Or-causality, relative-timing ('lazy' approach), unfolding-based verification (incl. nets with read-arcs), tool Petrify; application to Molnar's counterflow pipeline design

Petrify flow

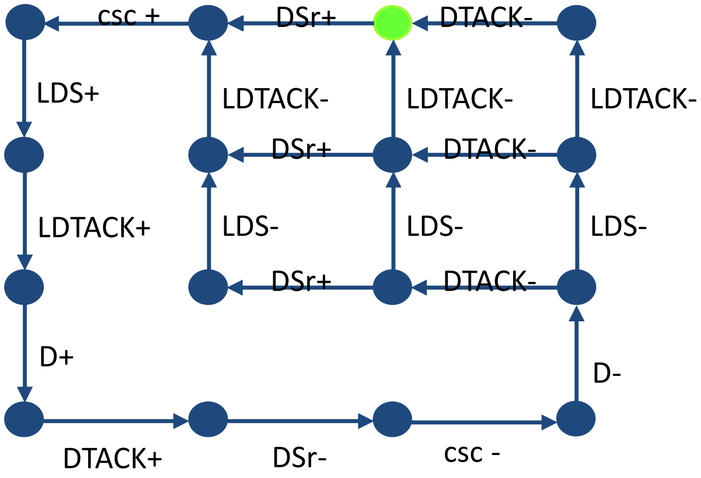


synthesis



Logic asynchronous circuit
(for Read Cycle)

Read Cycle State Graph (VME bus)



Boolean equations:

LDS = D ∨ csc

DTACK = D

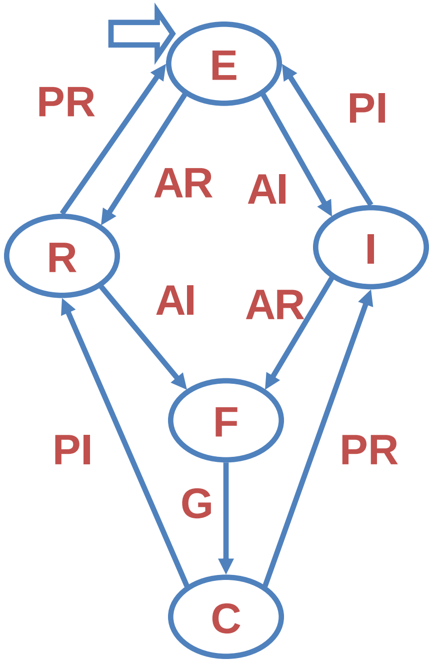
D = LDTACK

csc = DSr

Complete State Coding (CSC)

Example: counterflow pipeline

Molnar’s 5-state TS:

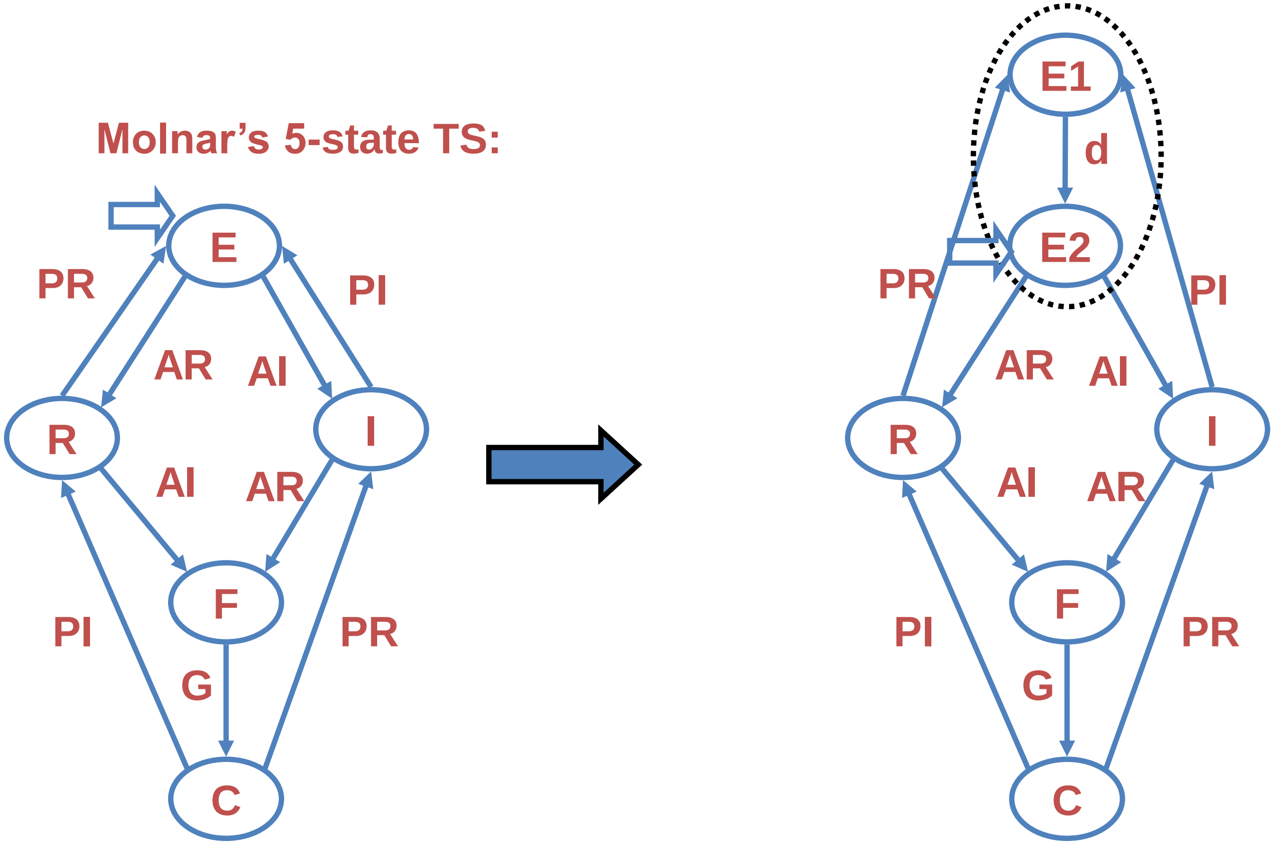


Solution:

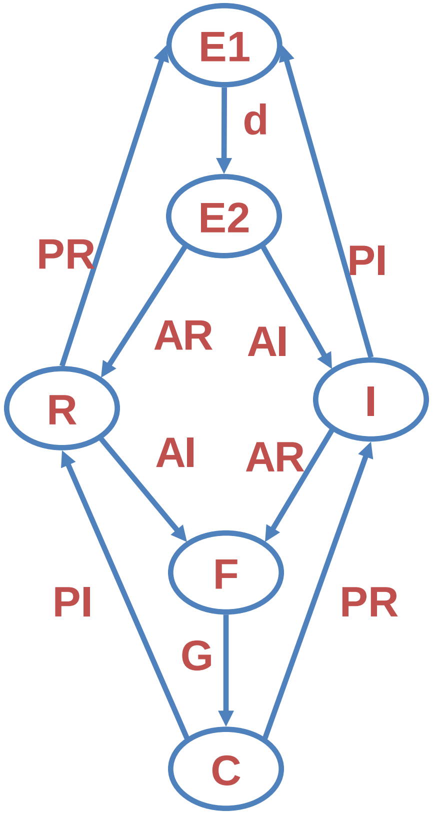
Split a state (E) and insert a silent action (d), preserving behavioural (observational) equivalence

Example: counterflow pipeline

Molnar's 5-state TS:



Example: counterflow pipeline



Semi-elementary TS

Minimal set of regions:

r1 = {E1,E2,I} <- pre(AR), post(PR)

r2 = {E1,E2,R} <- pre(AI), post (PI), co(d)

r3 = {R,F,C} <- pre(PR), post(AR), co(G)

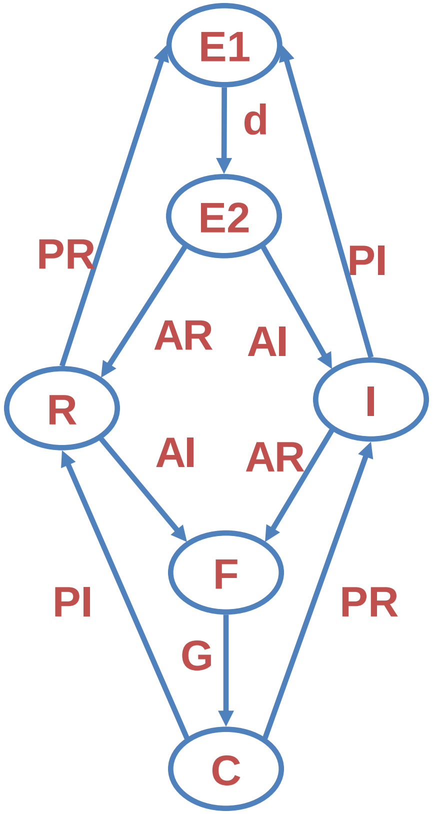
r4 = {I,F,C} <- pre(PI), post(AI)

r5 = {E2,I,C} <- pre(PI,AR), post(G,d)

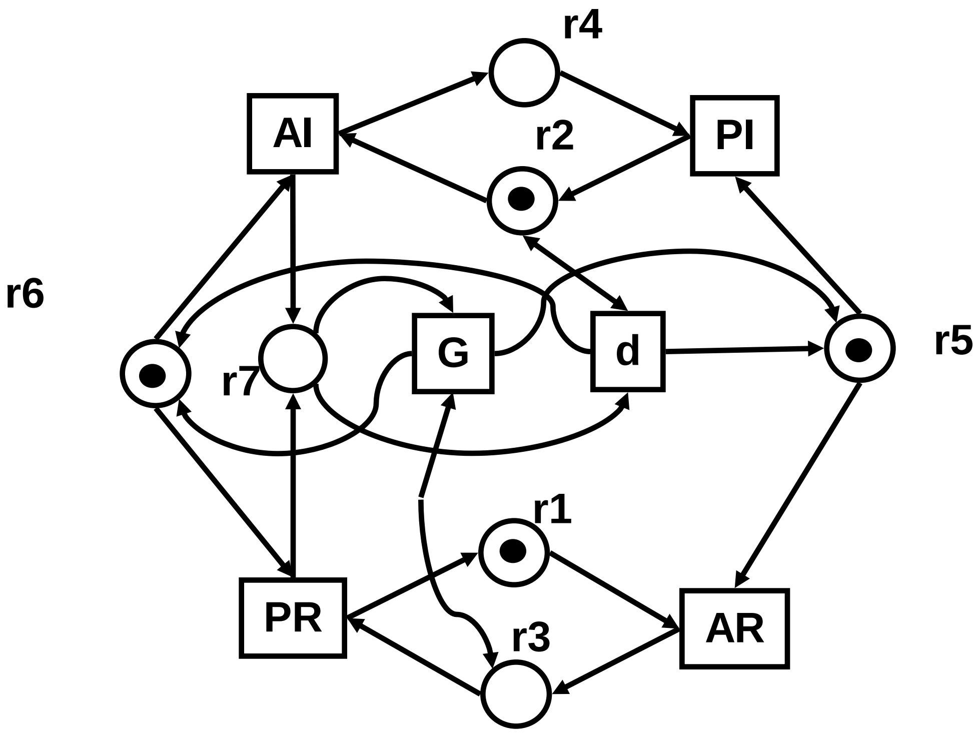
r6 = {E2,R,C} <- pre(PR,AI), post(G,d)

r7 = {E1,I,F} <- pre(G,d), post(PR,AI)

Example: counterflow pipeline

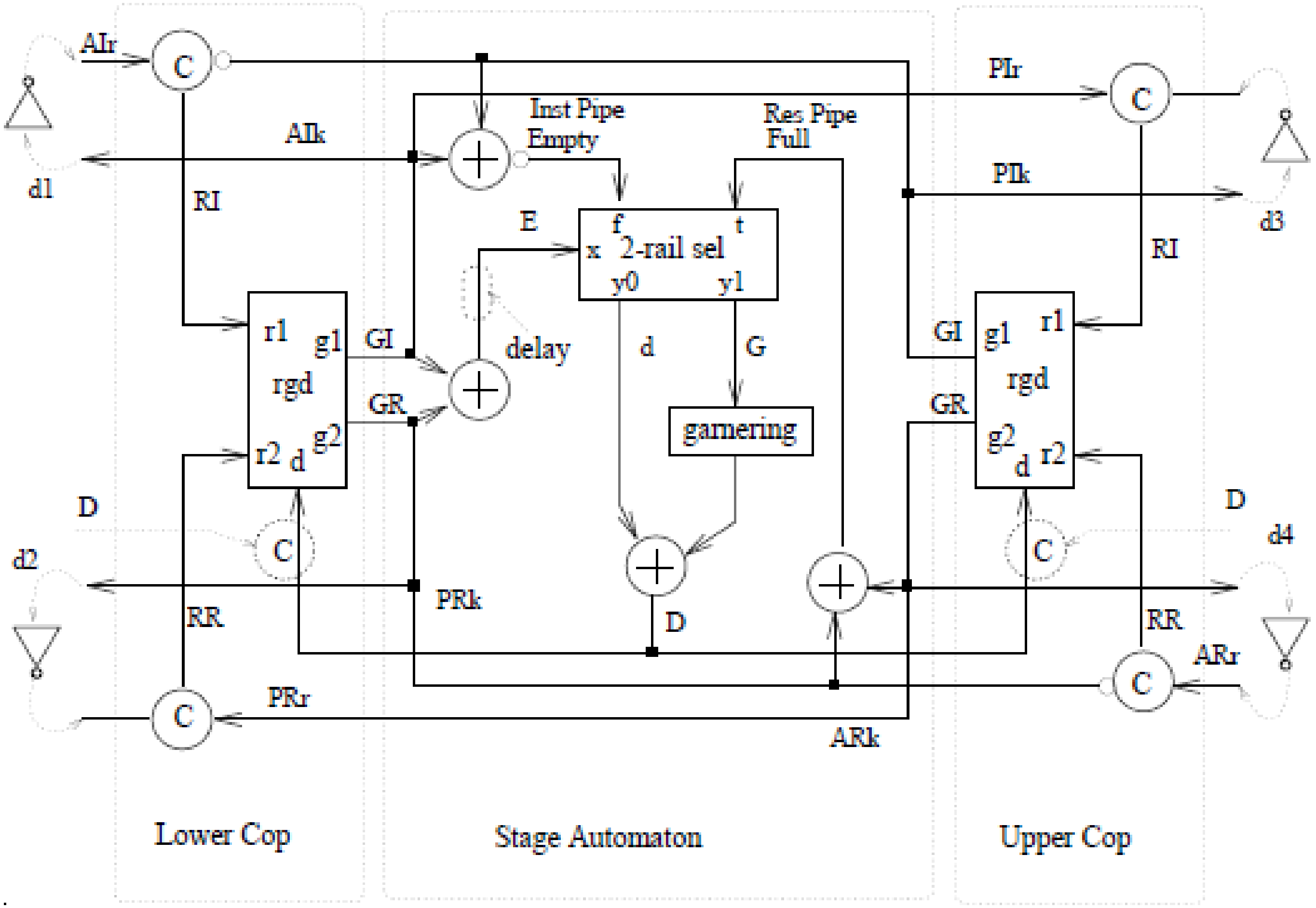


Semi-elementary TS



Semi-elementary Petri net

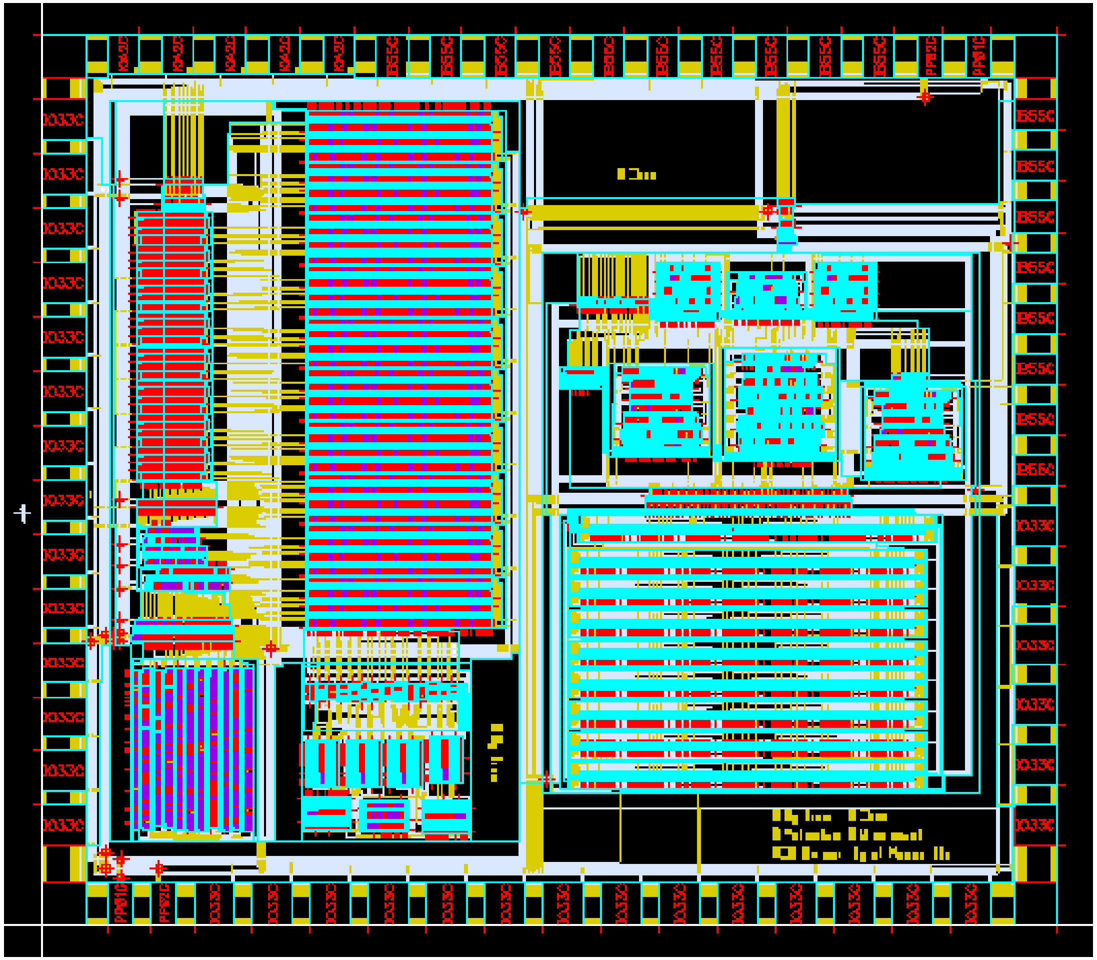
CFPP Implementation



1990s: Arbiter design

- Design of circuits with internal conflicts, all kinds of arbiters, modelling and analysis of arbiters asynchronous communication mechanisms
- Projects: HADES
- Highlights: method for factorising mutexes, new priority arbiters, 3 and 4 slot ACMs with mutexes
- People: David Kinniment, Fei Xia, Alex Bystrov, Delong Shang, Ian Clark, Bo Gao, Isi Mitrani, Tony Davies, Hugo Simpson and Eric Campbell (MBDA)
- First VLSI chip – HADIC (0.6um CMOS, Europractice)

HADIC: Our first Newcastle's asynchronous chip

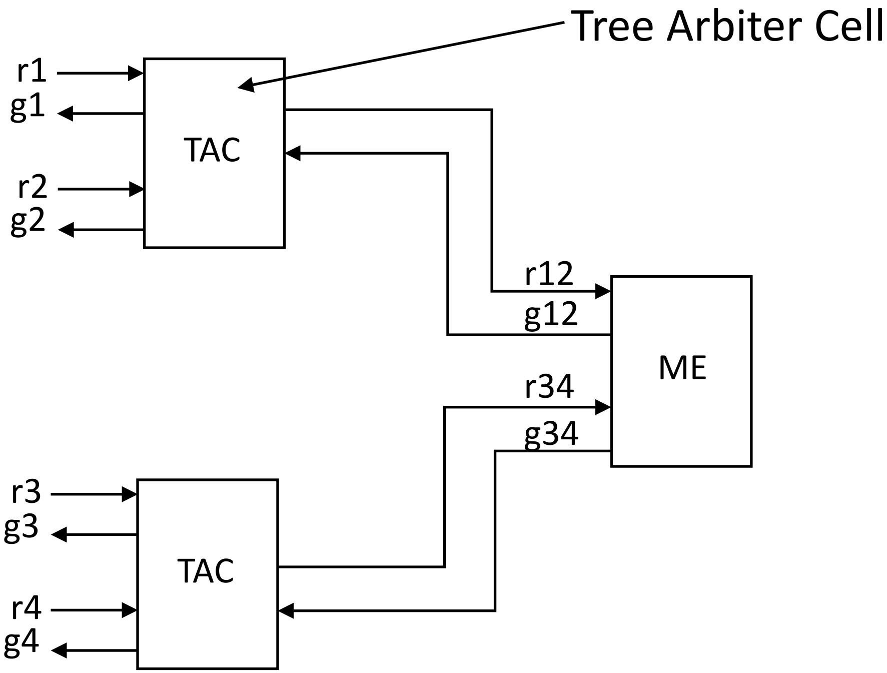


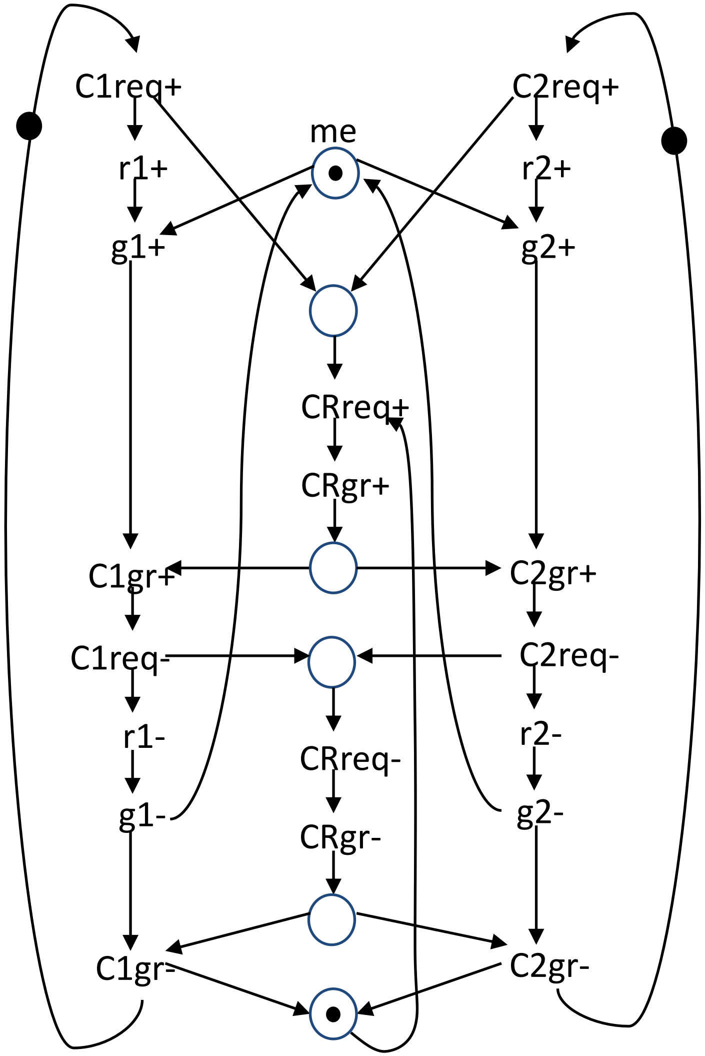
Designed using
Europractice
tools
On a SUN
w/station called
Sadko

What HADIC had

- a **three-way ordered arbiter** comprising a three-way mutex element, based on a three-stable FF
- a **one-hot pipeline FIFO buffer**, and a request mask circuit, to demonstrate the effect of request-ordering in avoiding secondary metastability effects discussed,
- an **eight-way ordered arbiter with a low-latency tree arbiter** and a low-latency cyclic FIFO buffer, to demonstrate the role of low latency techniques in ordered arbitration with large numbers of channels,
- an **eight-way static priority arbiter** with a non-conventional (non-topological) priority allocation mapping to demonstrate the effect of static prioritisation defined as an arbitrary function,
- an **eight-way dynamic priority arbiter** with four priority levels (the priority data encoded in dual rail code) to study the variations of arbiter latency against different conditions of issuing prioritised requests and effects of early control signal propagation in the priority logic with a tree structure,
- an **eight-way token ring arbiter** with a one-level cascaded buffer, to study the effects of improved done-to-grant and practical fairness via buffering,
- a **comparator with a latch for an A/D converter**, to study the metastability effects in value/time domains,
- an **on-chip tester, for arbiters and asynchronous communciation mechanisms**, containing a number of original circuits to experiment with system-on-chip testing.

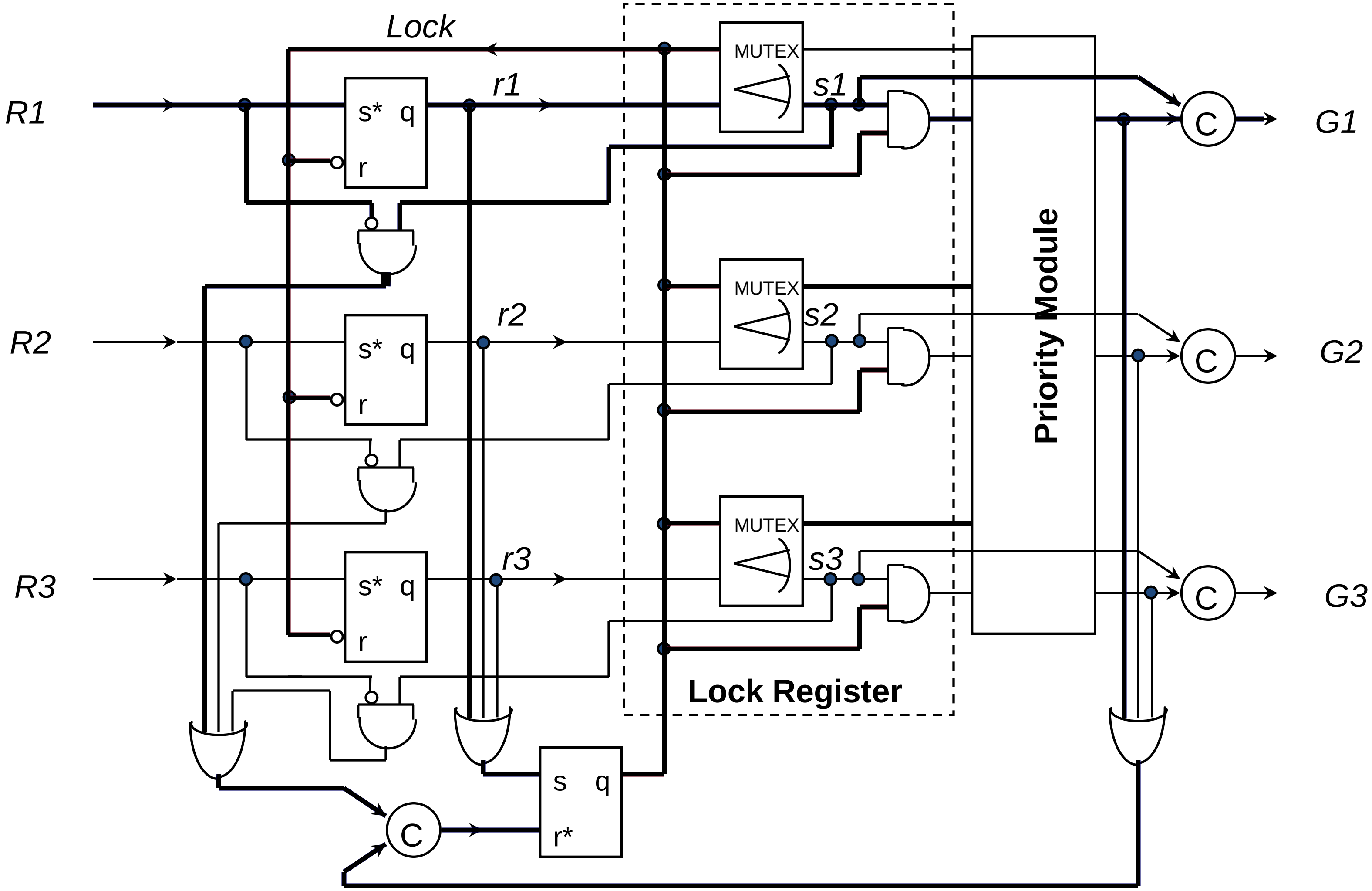
Eaxmple: Multiway Arbiters (tree)

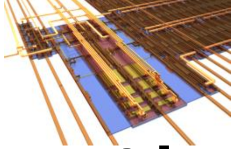




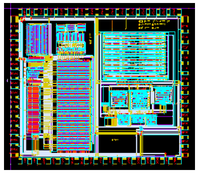
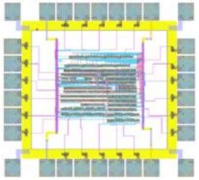
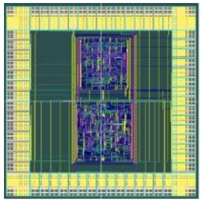
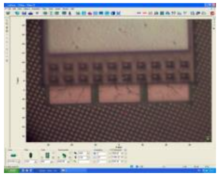
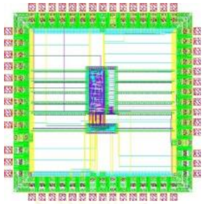
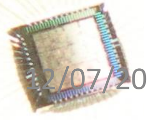
4-way arbiter built as a cascade arbiter with OR-causal request propagation

Example: static priority arbiter

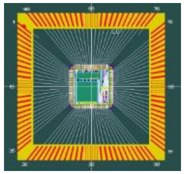
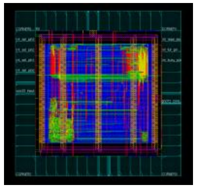
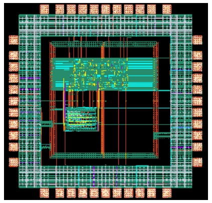
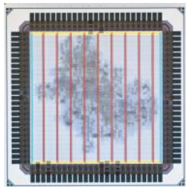
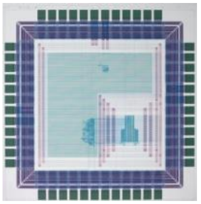




Newcastle Async Chip Designs

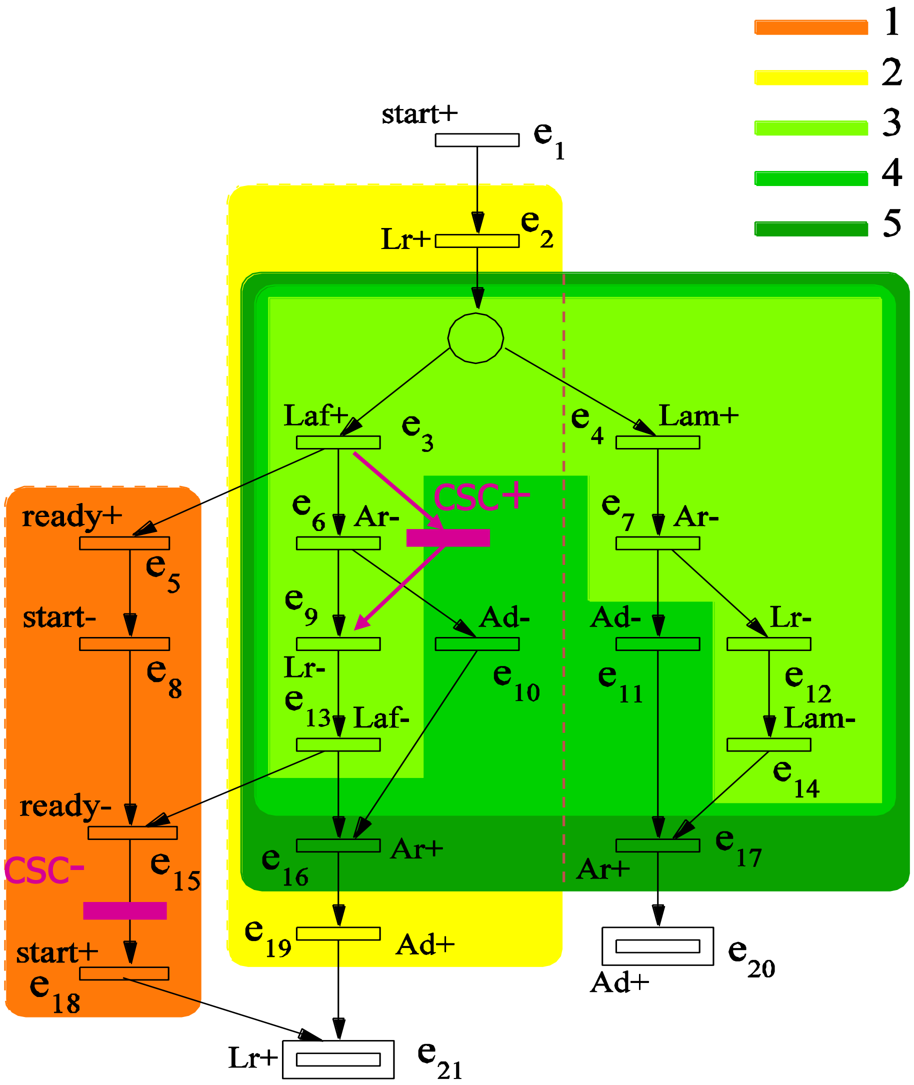
- Arbitration: HADIC1 (2000, CMOS 0.6um) – asynchronous arbiters, ADC, Async communication mechanism
- Time measurement: TMC (2003, CMOS 0.2um, collab. with Sun) time amplifier and time-to-digital converter
- Synchronization: SYRINGE1 (2006, UMC 0.18um), SYRINGE2 (2008, UMC, 0.13um) metastability measurement, robust synchronizer
- Security: SCREEN1 (2005, AMS 0.35um), SCREEN2 (2006, AMS 0.18um) AES cores with power balancing; SURE (2008, UMC 0.13um), Galois Encoded Logic, Script1 &2 (2010, TSMC 90nm)
- Networks on Chip: NEGUS1 (2007, UMC 0.13um), phase-encoding signalling, NeuroNOC (2009, TSMC 90nm) reconfigurable neural network, 3D-chip (2011, MIT Lincoln Labs, 150nm with TSVs)
- Energy harvesting: Holistic1 (2011, UMC 90nm), self-timed SRAM, Holistic 2 (2012, UMC 0.18um), reference free voltage sensor
- Power proportionality: Async 8051 (2013, STM CMP 0.13um)
- SAVVIE and A4A chips: to be reported

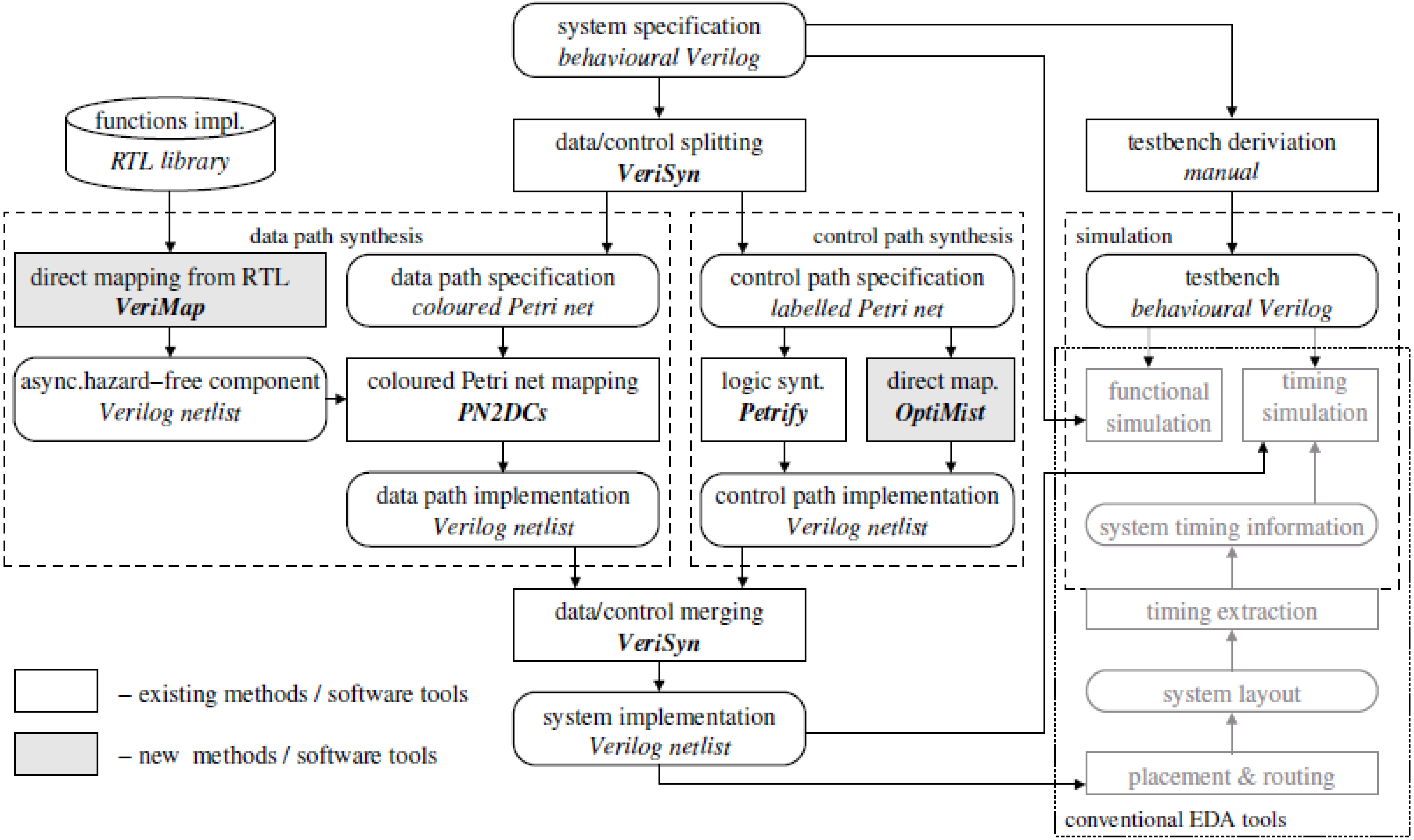
2000-2005: Synthesis and Testing

- Synthesis line: visualisation, unfolding-based synthesis, direct synthesis, negative gate synthesis, generalised OR-causality; regions, ACM synthesis
- Projects: BREACH, MOVIE, BESST, STELLA
- People: Maciej Koutny, Agnes Madalinski, Victor Khomenko, Danil Sokolov, Frank Burns, Deepali Koppad, Nikolai Starodoubtsev, Josep Carmona, Fei Hao, Kyller Gorgonyo
- Highlights: Core maps for visualization, Tracker-Bouncer for direct mapping, Lots of Tools: ConfRes, PUNF/MPSAT, Verisyn, Optimist, PN2DCs; application of direct mapping to S. Furber’s duplex interface design; online testing for async protocols; generalised regions synthesis

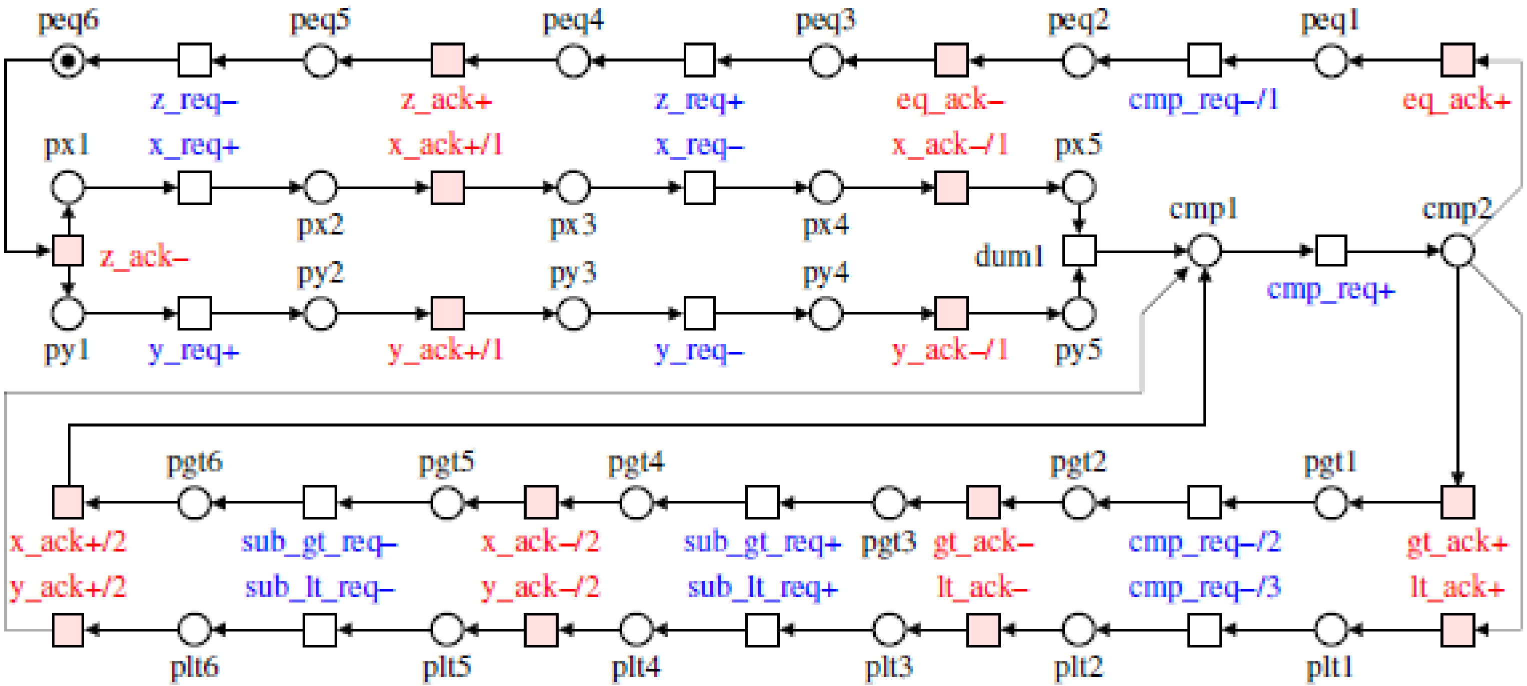
CSC Core visualisation using unfoldings



BESST Design Flow

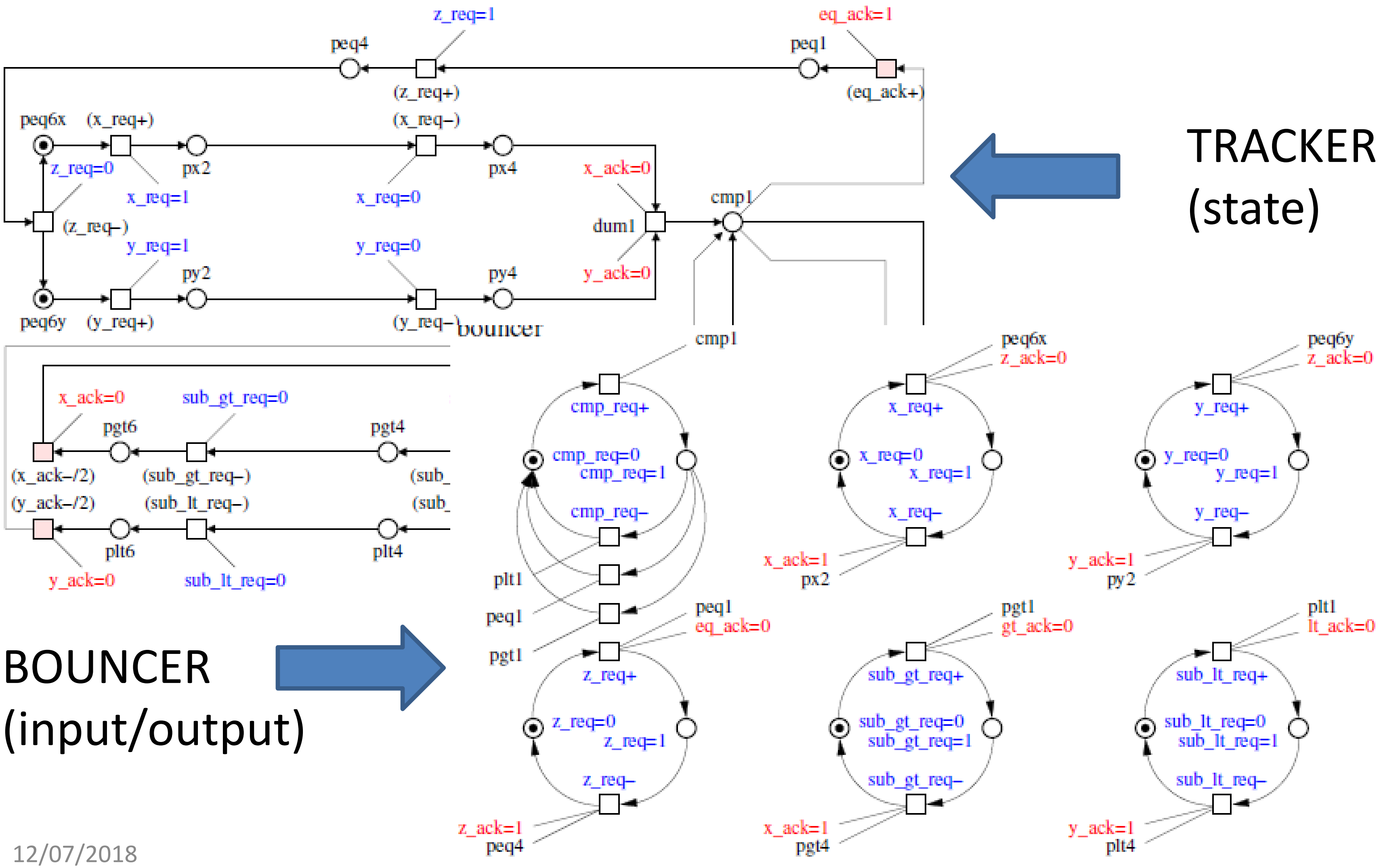


Example: GCD using Optimist flow



Range of Optimisations for latency and size could be applied

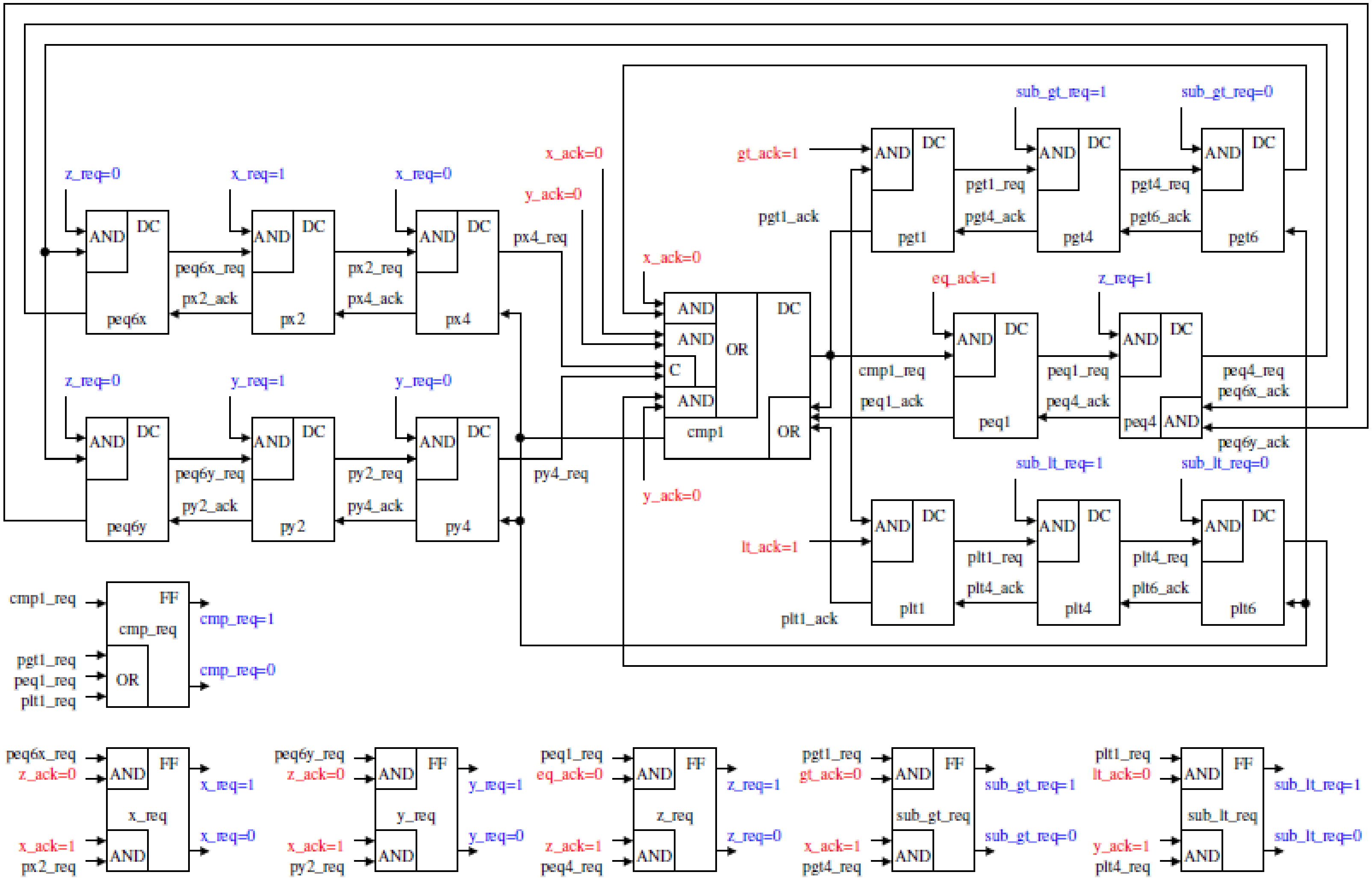
Example: GCD using Optimist



TRACKER
(state)

BOUNCER
(input/output)

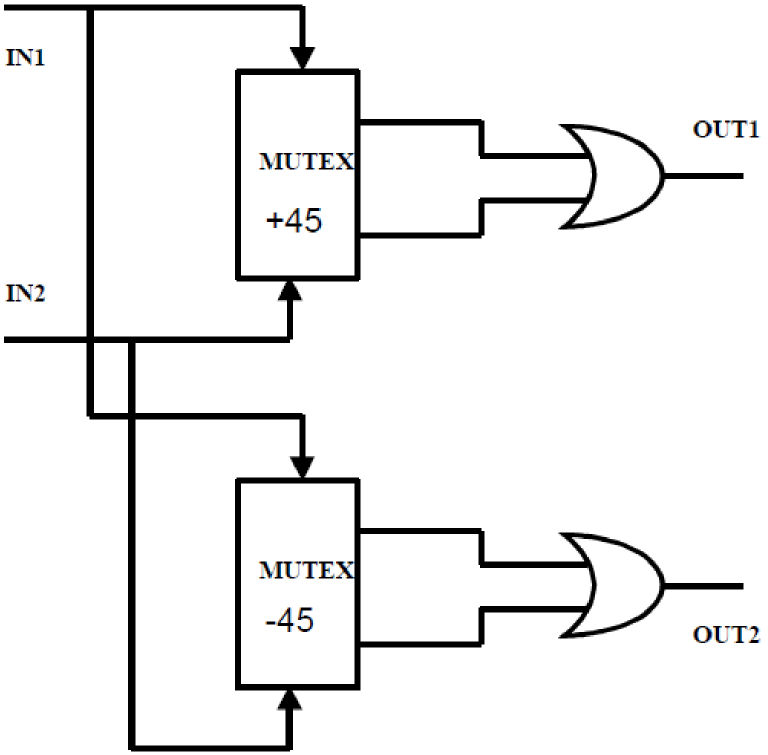
Direct mapping: David cell circuit

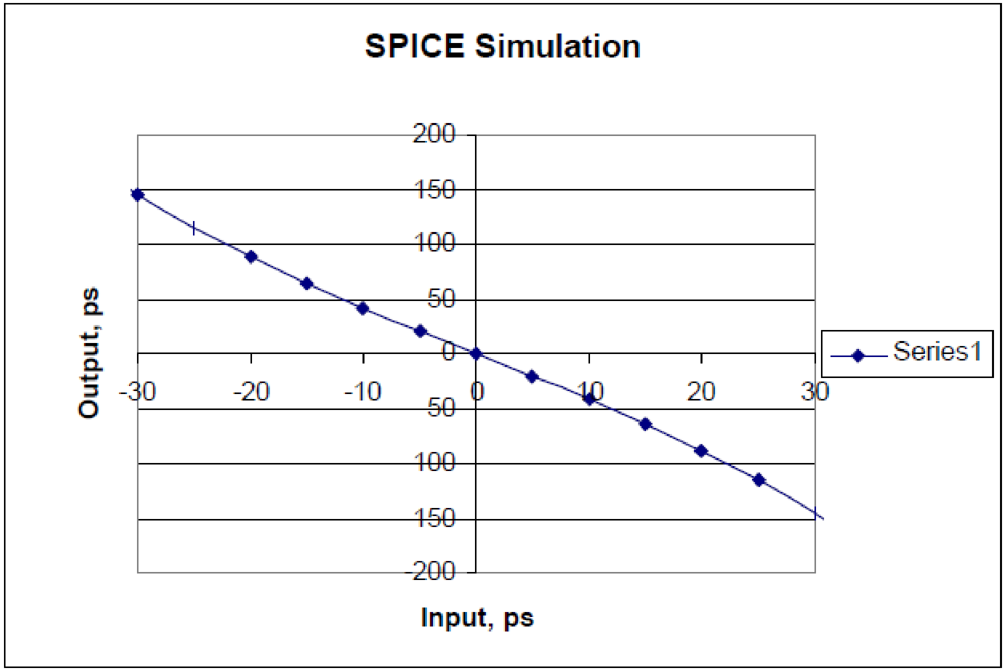


2000-2010: Metastability and Event Processing

- Synchronisation, time to digital conversion, time amplification, robust synchronization, speculative synchronization, dynamic systems models of analogue circuits
- Projects: COHERENT, STEP, SYRINGE, RelCel
- People: Gordon Russell, Graeme Chester, Oleg Maevsky, Amir Abbas, Nikos Minas, Keith Heron, Jun Zhou, Fei Hao, Yuan Chen, Stan Golubcovs, Ghaith Tarawneh, Ioannis Syranidis, Ahmed Alahmadi, Mohammed Alshaikh, James Guido; collaboration with Charles Dike (Intel), Ian Jones (Sun/Oracle) Marc Renaudin (Grenoble)
- Highlights: Metastability physical measurements, RNG, TD converters, time amplifier, FPGA synchronsers, speculative synchronizers; reconfigurable synchronizers, soft arbiters; Several Chips (via Europractice and Sun)

Time difference amplifier



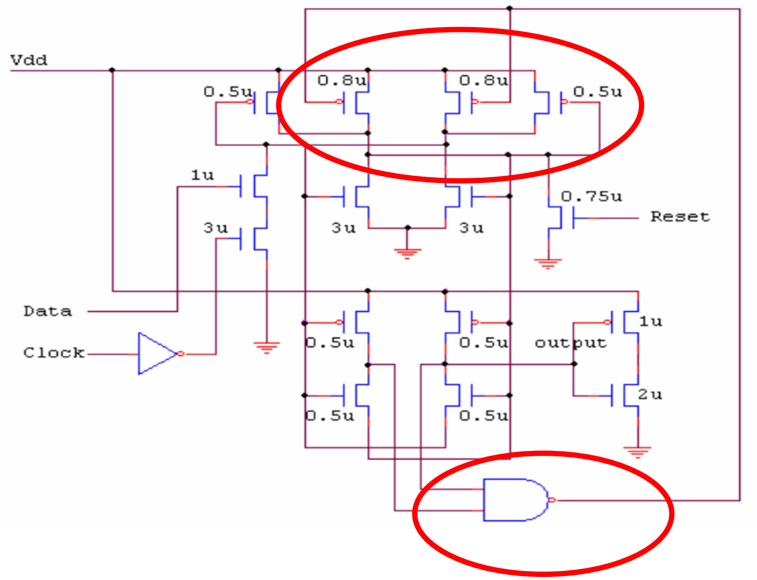


$$\Delta_{out} = \tau.\ln(45+\Delta_{in}) - \tau.\ln(45-\Delta_{in})$$

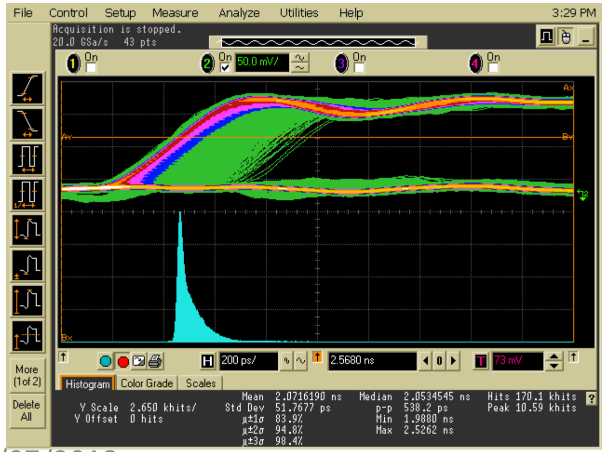
Was used as the first cascade of time2digital converter

Robust synchronizer

Dynamic feedback control



	Measurement Results (ps)			
Vdd(v)	Jamb Latch B		Robust Synchronizer	
	$>10^{-14}$ s	$<10^{-14}$ s	$>10^{-14}$ s	$<10^{-14}$ s
1.8	19.44	35.55	15.27	34.92
1.7	21.75	37.29	16.53	35.76
1.6	25.64	40.93	19.38	38.25
1.5	28.77	52.36	20.29	43.07
1.4	36.22	66.17	23.75	50.36
1.35	45.43	75.35	28.51	58.19

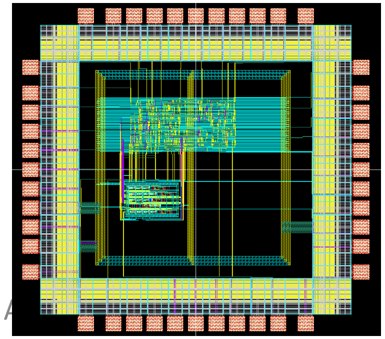


```
Acquisition is stopped.
20.0 GSa/s   43 pts
```

2 On ☑ 50.0 mV/

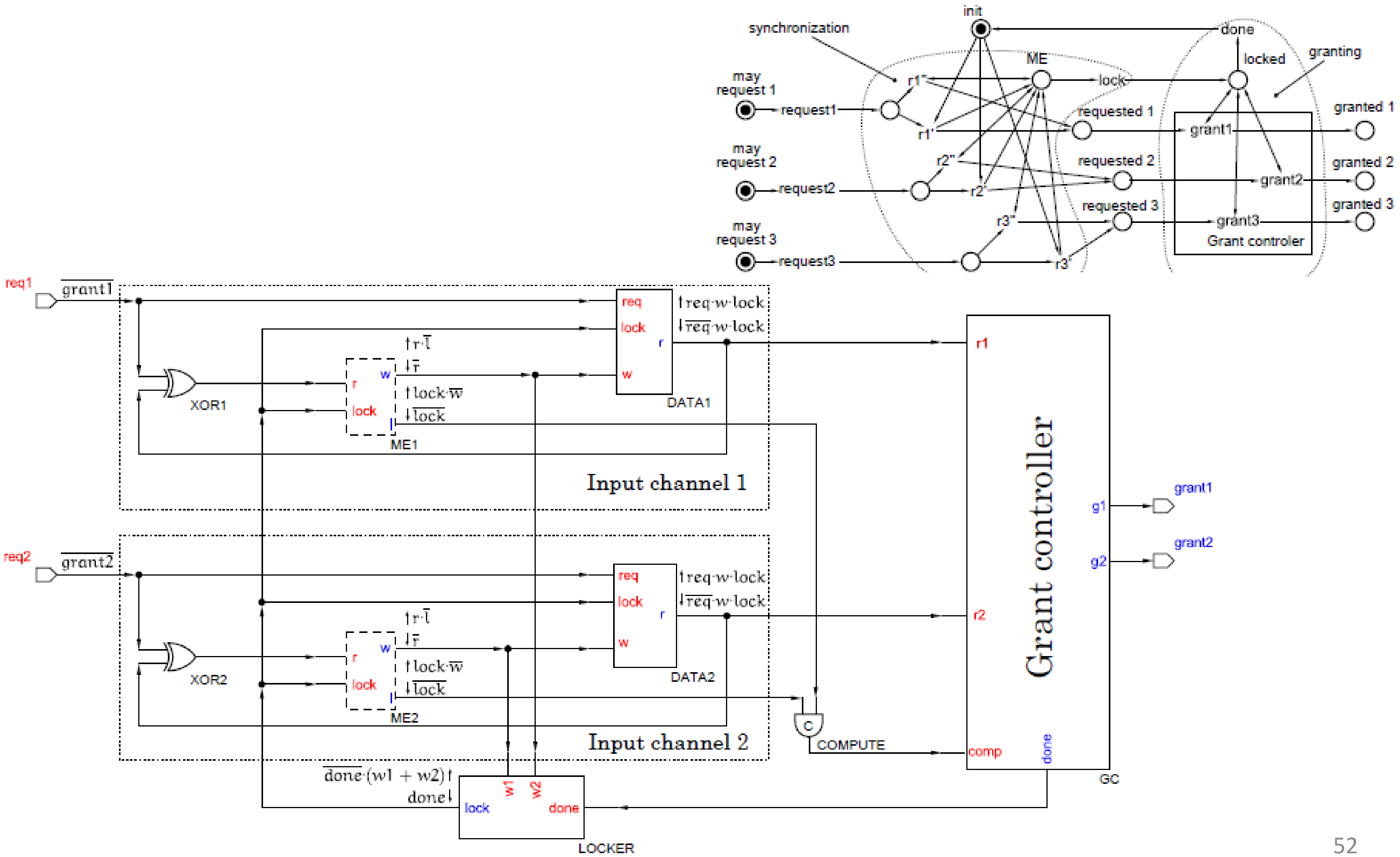
200 ps/ 2.5680 ns 0 73 mV

```
                          Mean   2.0716190 ns   Median  2.0534545 ns   Hits 170.1 khits
  Y Scale  2.650 khits/   Std Dev 51.7677 ps       p-p  538.2 ps       Peak 10.59 khits
 Y Offset  0 hits          µ±1σ  83.9%             Min  1.9880 ns
                           µ±2σ  94.8%             Max  2.5262 ns
                           µ±3σ  98.4%
```



τ (metastability time constant) vs Vdd

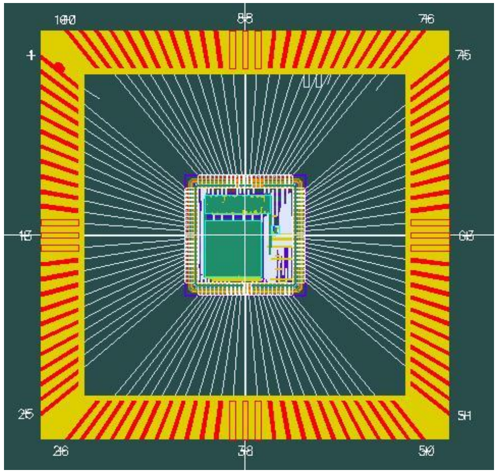
Multiway arbiters: general approach



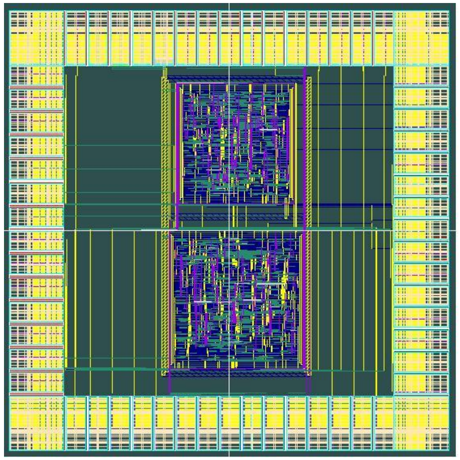
2004-2009: Security

- Asynchronous circuits for power-attack resistance, mixed radix synthesis
- Projects: SCREEN, SURE, STRIP
- People: Julian Murphy, Ashur Rafiev, Atmel people (Ewart Grey, Russell Hobson, Steve Pickles)
- Highlights: Automatic translation from single to dual rail; Tools: Verimap; RMMixed; Many Chips: SCREEN1/2, SURE, STRIP1/2

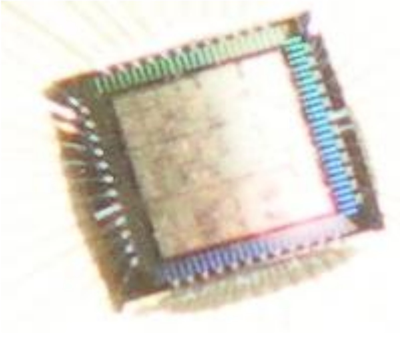
Some Security Chips



SCREEN1: AES encryption core design using alternating spacer dual-rail signalling for power balancing (generated from synchronous RTL using in-house tool VeriMap)

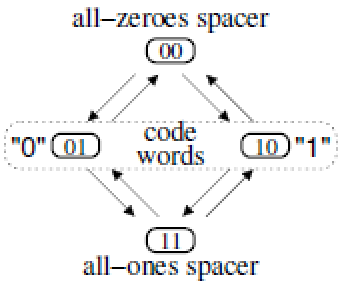


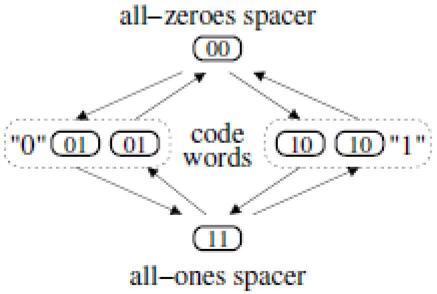
SCREEN 2: different (synchronous single-rail and power-balanced) versions of AES S-box.

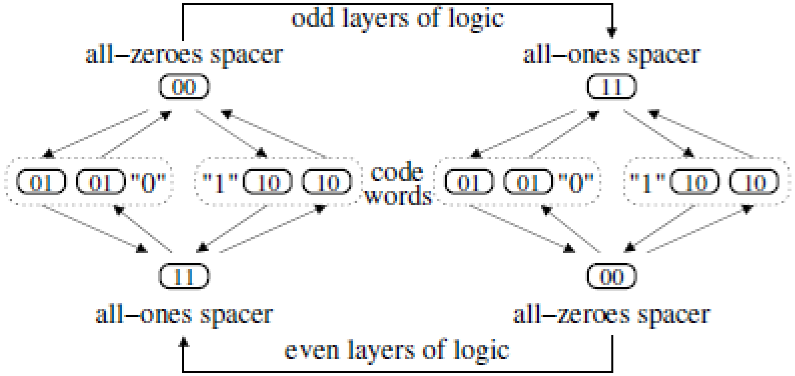


SURE:
Galois Encoded Logic AES-128 IP core. Implements side-channel secure "high radix" AES-128 encryption in a fully parallel architecture to investigate the security and study the market feasibility of the technology.

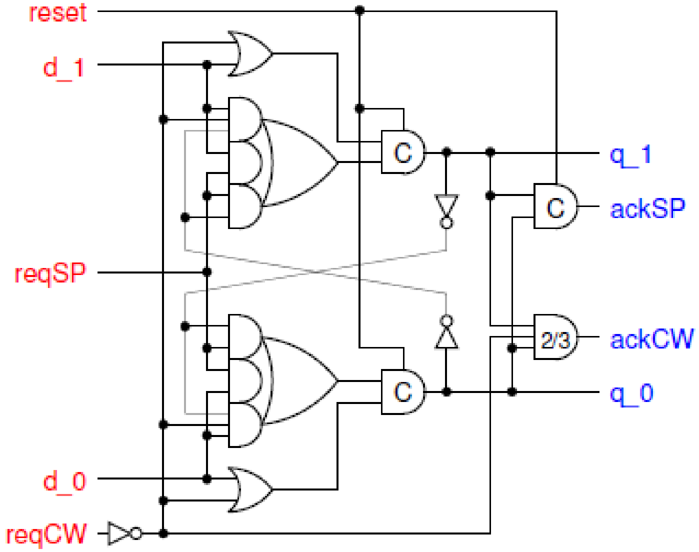
Verimap: Dual spacer and negative gate optimization

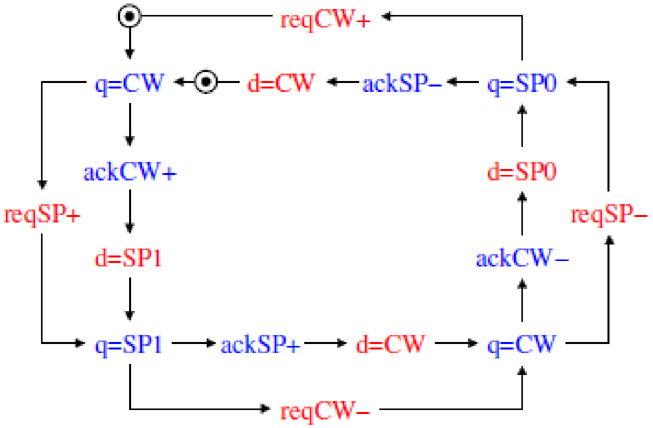






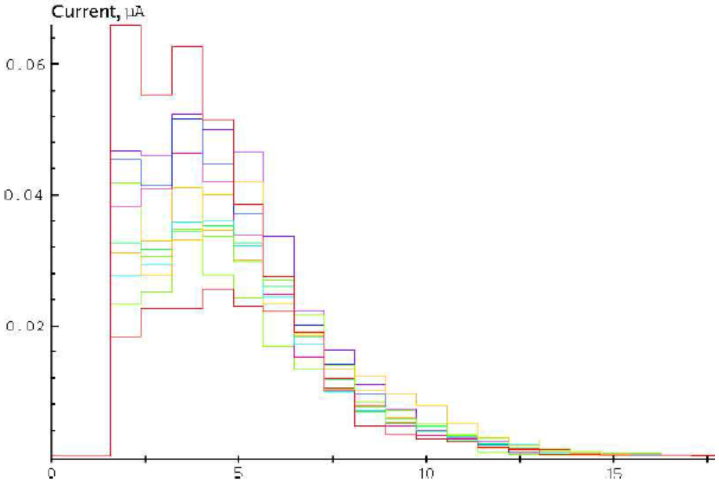
Alternating spacer dual-rail flip-flop



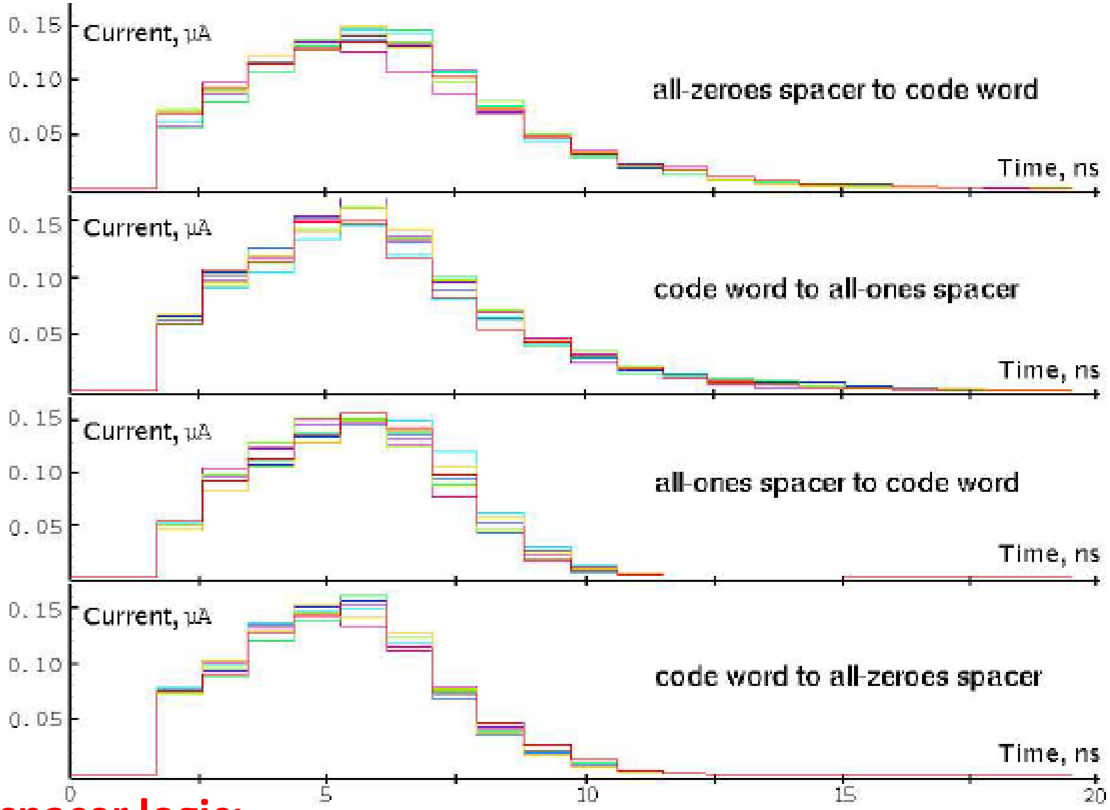


Example: Verimap AES design

Single Rail current trace



Dual Rail alternating spacer traces



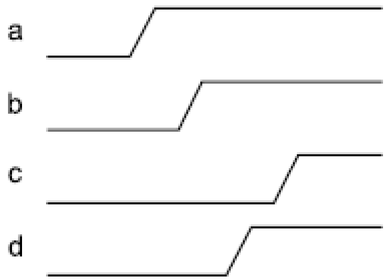
Key property of dual-rail alternating spacer logic:

For every operational cycle of the circuit all logic nodes fire => maximum power balancing

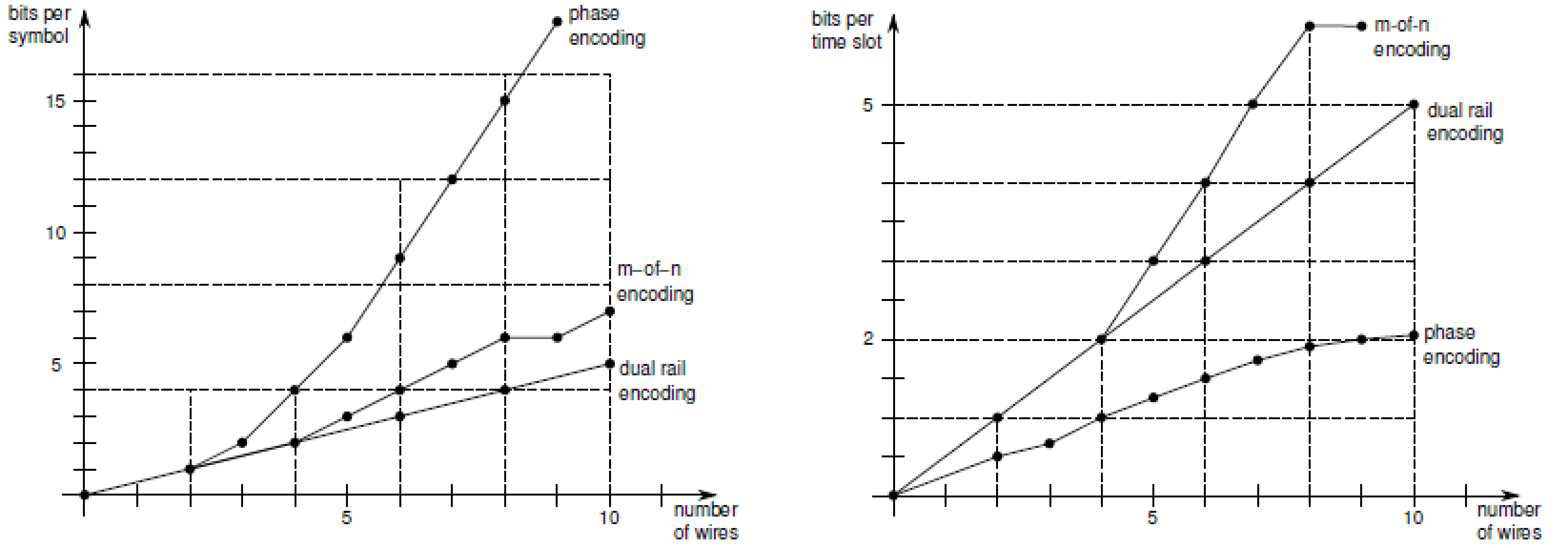
2005-2015: NoCs, FPGAs and New models

- New types of interconnects, communications, cross-talk, routing, neural network NoCs, surface wave comms, async FPGAs
- Projects: NEGUS
- People: Enzo D’Alessandro, Andrey Mokhov, Basel Halak, Sohini Dasgupta, Robin Emery, Terrence Mak, Ra’ed Aldujaili, Nizar Dahir, Ammar Karkar, Hock-Soon Low; Southampton (Bashir Al-Hashimi and Simon Ogg)
- Highlights: Phase-encoding, CPOGs, cross-talk resilient interconnects, models of GALS and wrappers, on-line deadlock detection; power/thermal model; Async FPGA flow; Chips: Negus1 and NeuralNoC

Phase encoding



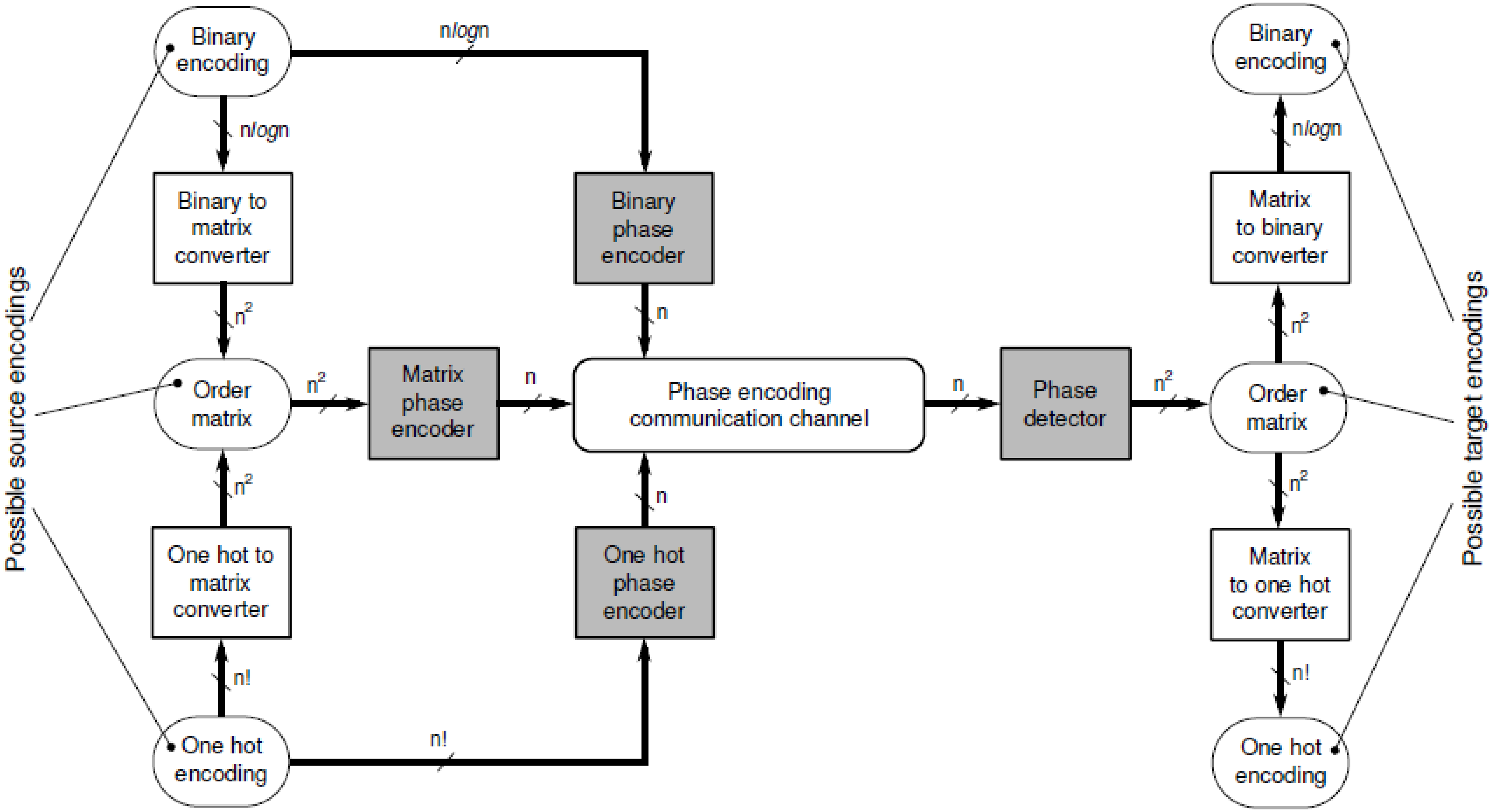
An example of phase encoded symbol *"abdc"*
(total no of symbols is 4!=24)



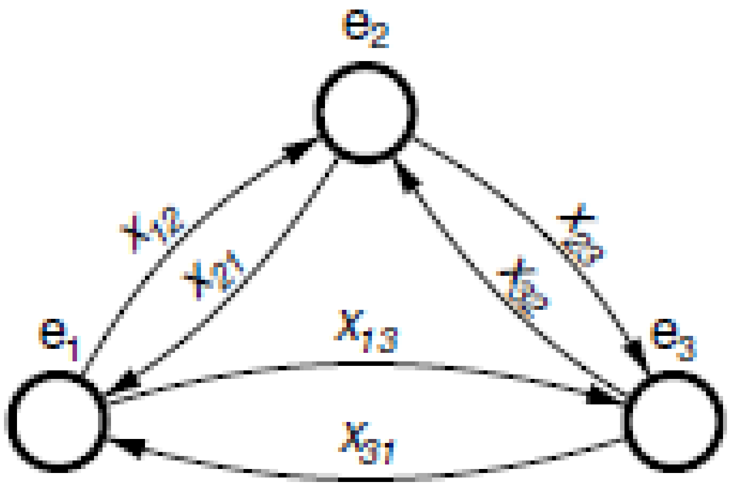
(a) number of bits per symbol

(b) number of bits per time slot

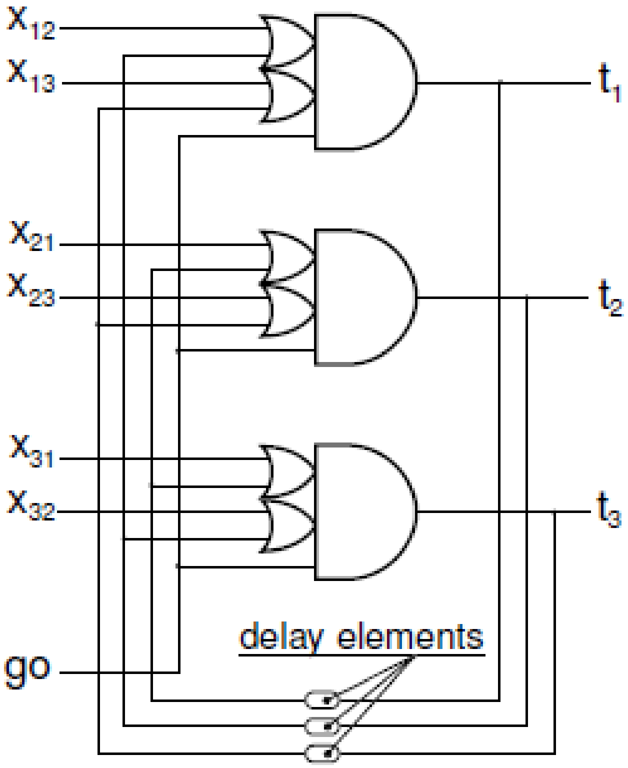
Phase encoding comms



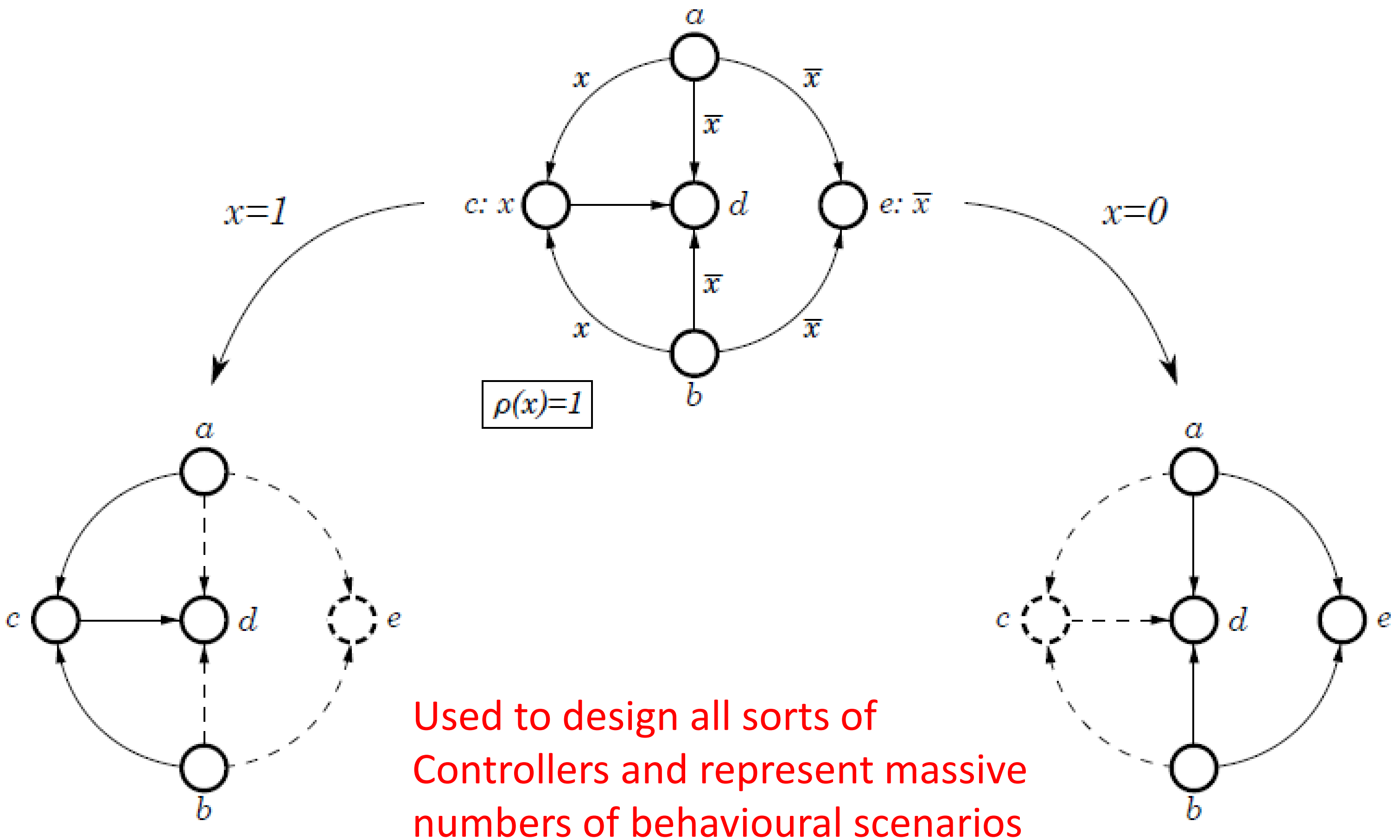
Synthesis of logic for phase encoding



(a) CPOG specification



Conditional Partial Order Graphs

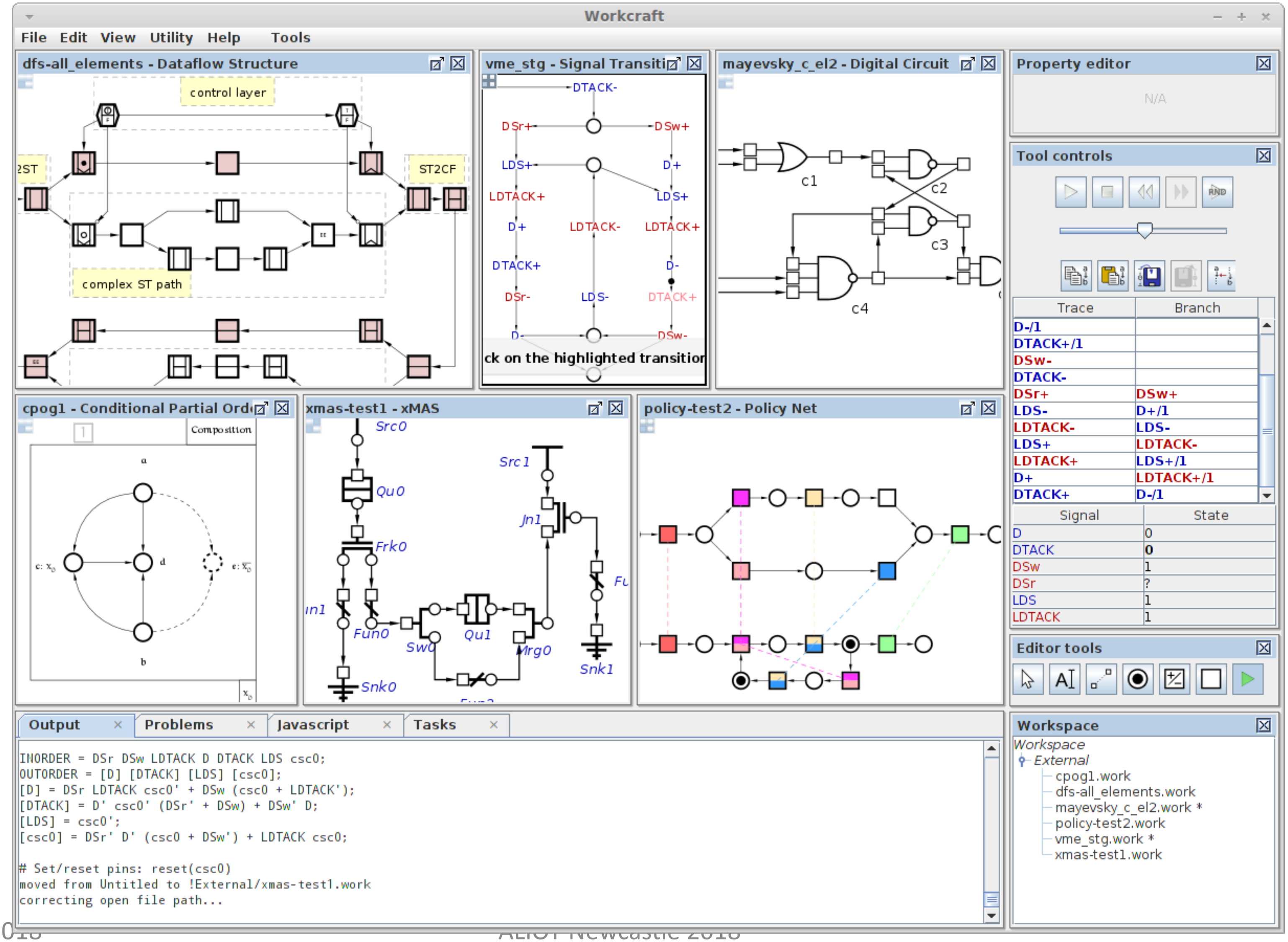


Used to design all sorts of Controllers and represent massive numbers of behavioural scenarios …Also led to Algebra of Graph

2005-2010: Synthesis

- Interpreted graph models, Compositionality, Data-path synthesis, SDFS and anti-tokens, Strong and Weak Indicatability, Synthesis of circuits with isochronic forks; Online data path testing
- Projects: SEDATE, VERDAD, GAELS
- People: Ivan Poliakov, Arseniy Alekseyev, Yu Zhou (Joe), Kelvin Gardiner, Yu Li; Manchester (Doug Edwards, Will Toms, Charlie Brej), Augusburg (Walter Vogler, Dominik Wist)
- Highlights: Theory for compositions, IF relaxation; Workcraft toolkit and many plugins

Workcraft toolkit



Trace	Branch
D-/1	
DTACK+/1	
DSw-	
DTACK-	
DSr+	DSw+
LDS-	D+/1
LDTACK-	LDS-
LDS+	LDTACK-
LDTACK+	LDS+/1
D+	LDTACK+/1
DTACK+	D-/1

Signal	State
D	0
DTACK	0
DSw	1
DSr	?
LDS	1
LDTACK	1

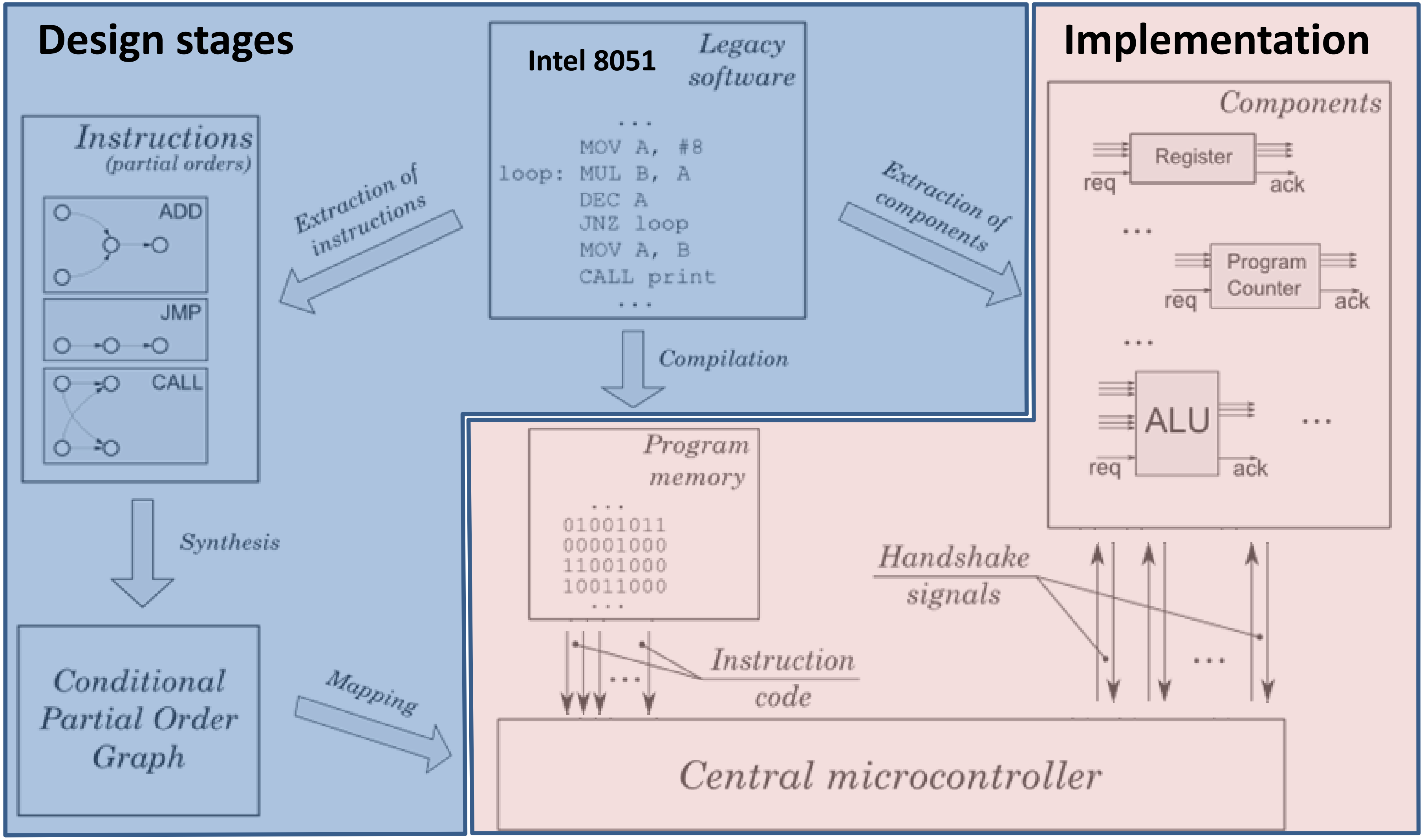
```
INORDER = DSr DSw LDTACK D DTACK LDS csc0;
OUTORDER = [D] [DTACK] [LDS] [csc0];
[D] = DSr LDTACK csc0' + DSw (csc0 + LDTACK');
[DTACK] = D' csc0' (DSr' + DSw) + DSw' D;
[LDS] = csc0';
[csc0] = DSr' D' (csc0 + DSw') + LDTACK csc0;

# Set/reset pins: reset(csc0)
moved from Untitled to !External/xmas-test1.work
correcting open file path...
```

Workspace
External
cpog1.work
dfs-all_elements.work
mayevsky_c_el2.work *
policy-test2.work
vme_stg.work *
xmas-test1.work

2007-2015: Low Power and Energy-modulated computing

- Low power processors, energy harvesting and async, power-proportional design, power variation resilient circuits, new power supply methods, power gating and new
- People: Chihoon Shen (Korea), Reza Ramezani, Max Rykunov, Abdullah Baz, Xuefu Zhang, Haider Alrudainy, Alessandro De Gennaro
- Highlights: low power M430, Charge-to-code converter, Speed-independent SRAM, Power-proportional 8051, CBBs versus SCCs, NEMS/MEMS for power gating; Chips: Reference-free sensor, ASRAM, 8051, Async Pipeline

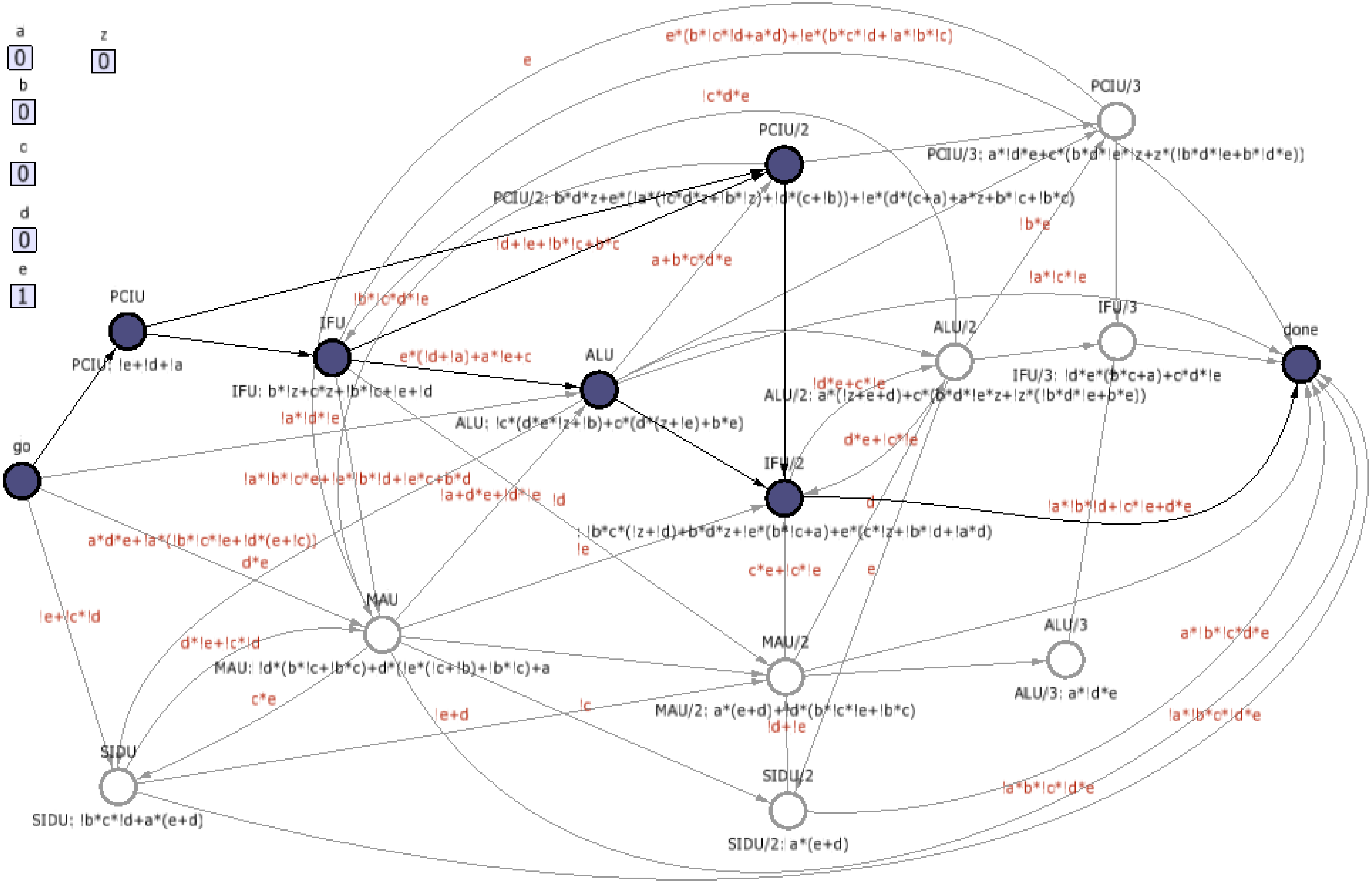
Conceptual view of the CPU design process



```
      ...
      MOV  A,  #8
loop: MUL  B,  A
      DEC  A
      JNZ  loop
      MOV  A,  B
      CALL print
      ...
```

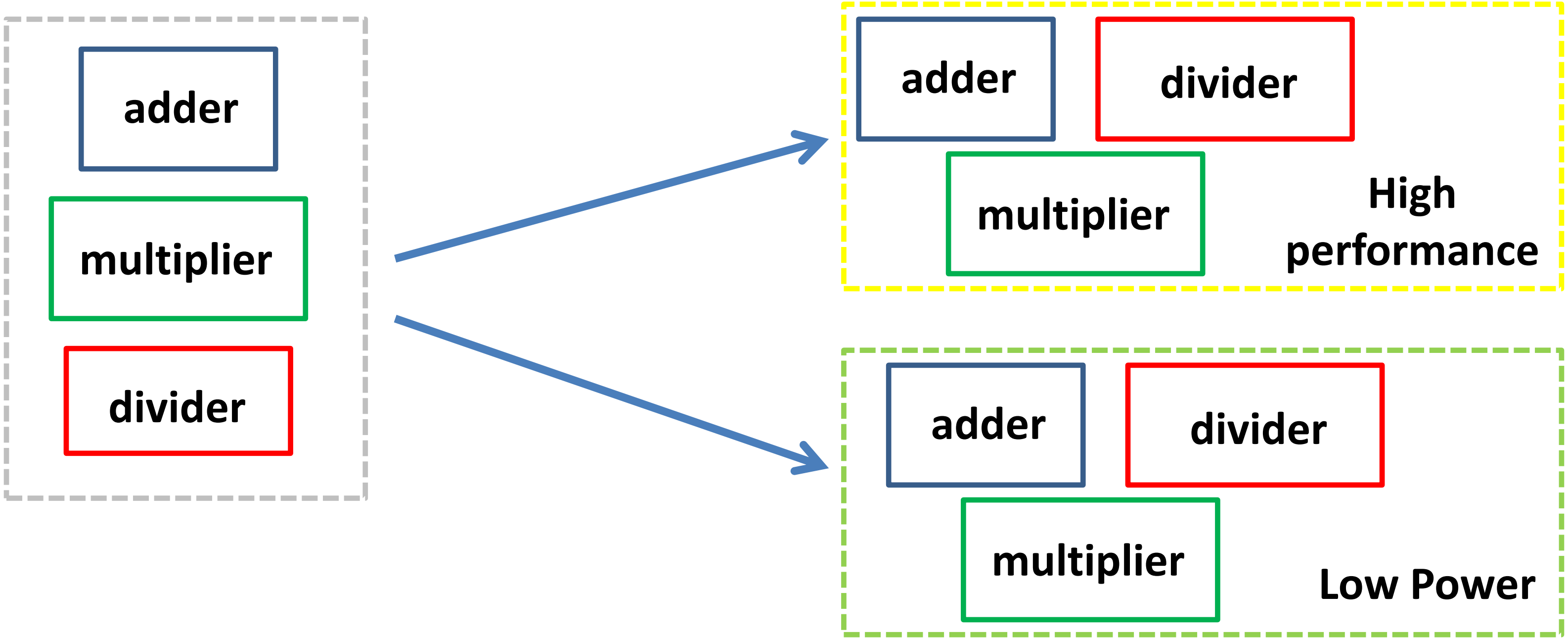
```
...
01001011
00001000
11001000
10011000
...
```

Intel 8051 ISA…



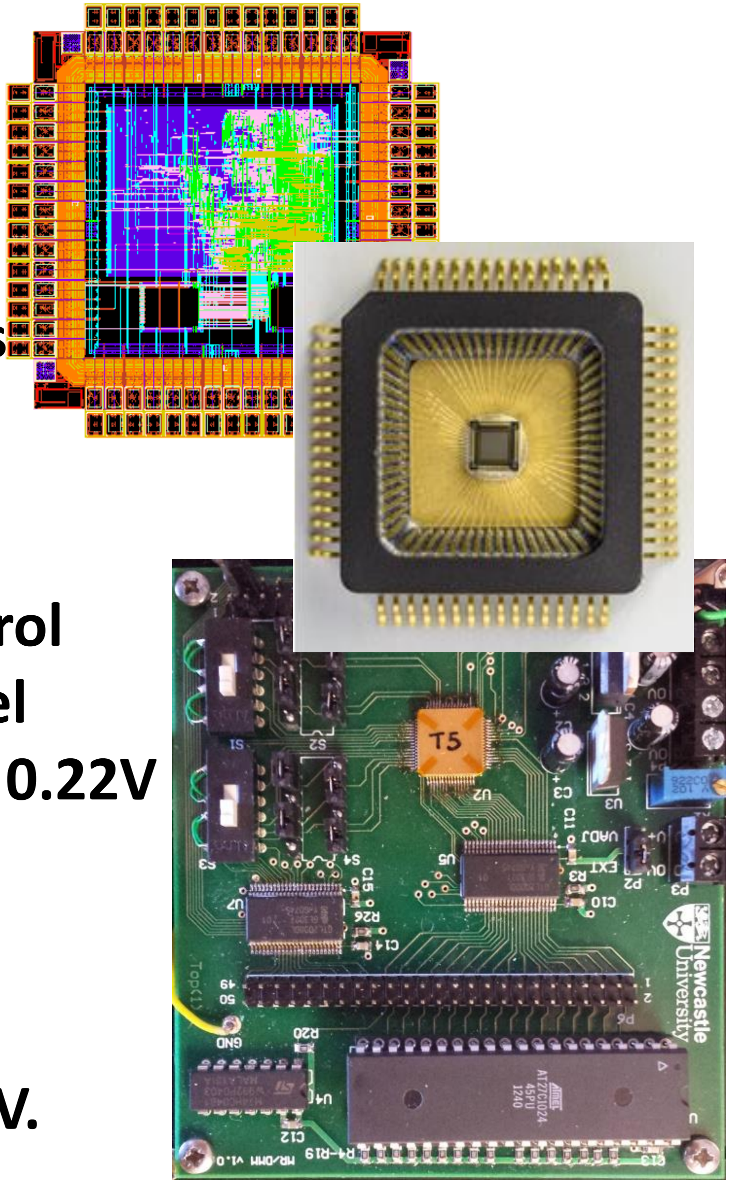
a
0
z
0
b
0
c
0
d
0
e
1
e*(b*!c*!d+a*d)+!e*(b*c*!d+!a*!b*!c)
e
!c*d*e
PCIU/3
PCIU/2
PCIU/3: a*!d*e+c*(b*d*!e*!z+z*(!b*d*!e+b*!d*e))
PCIU/2: b*d*z+e*(!a*(!c*d*z+!b*!z)+!d*(c+!b))+!e*(d*(c+a)+a*z+b*!c+!b*c)
!b*e
!d+!e+!b*!c+b*c
a+b*c*d*e
!a*!c*!e
PCIU
!b*!c*d*!e
IFU/3
IFU
ALU/2
done
e*(!d+!a)+a*!e+c
ALU
PCIU: !e+!d+!a
IFU/3: !d*e*(b*c+a)+c*d*!e
!d*e+c*!e
IFU: b*!z+c*z+!b*!c+!e+!d
ALU/2: a*(!z+e+d)+c*(b*d*!e*z+!z*(!b*d*!e+b*e))
!a*!d*!e
ALU: !c*(d*e*!z+!b)+c*(d*(z+!e)+b*e)
d*e+!c*!e
go
IFU/2
!a*!b*!c*e+!e*!b*!d+!e*c+b*d
!a+d*e+!d*!e
!d
d
!a*!b*!d+!c*!e+d*e
a*d*e+!a*(!b*!c*!e+!d*(e+!c))
: !b*c*(!z+!d)+b*d*z+!e*(b*!c+a)+e*(c*!z+!b*!d+!a*d)
!e
d*e
c*e+!c*!e
e
!e+!c*!d
MAU
ALU/3
a*!b*!c*d*e
d*!e+!c*!d
MAU/2
MAU: !d*(b*!c+!b*c)+d*(!e*(!c+!b)+!b*!c)+a
c*e
!c
ALU/3: a*!d*e
!e+d
MAU/2: a*(e+d)+!d*(b*!c*!e+!b*c)
!a*!b*c*!d*e
!d+!e
SIDU
SIDU/2
!a*b*!c*!d*e
SIDU: !b*c*!d+a*(e+d)
SIDU/2: a*(e+d)

Intel 8051 Datapath…



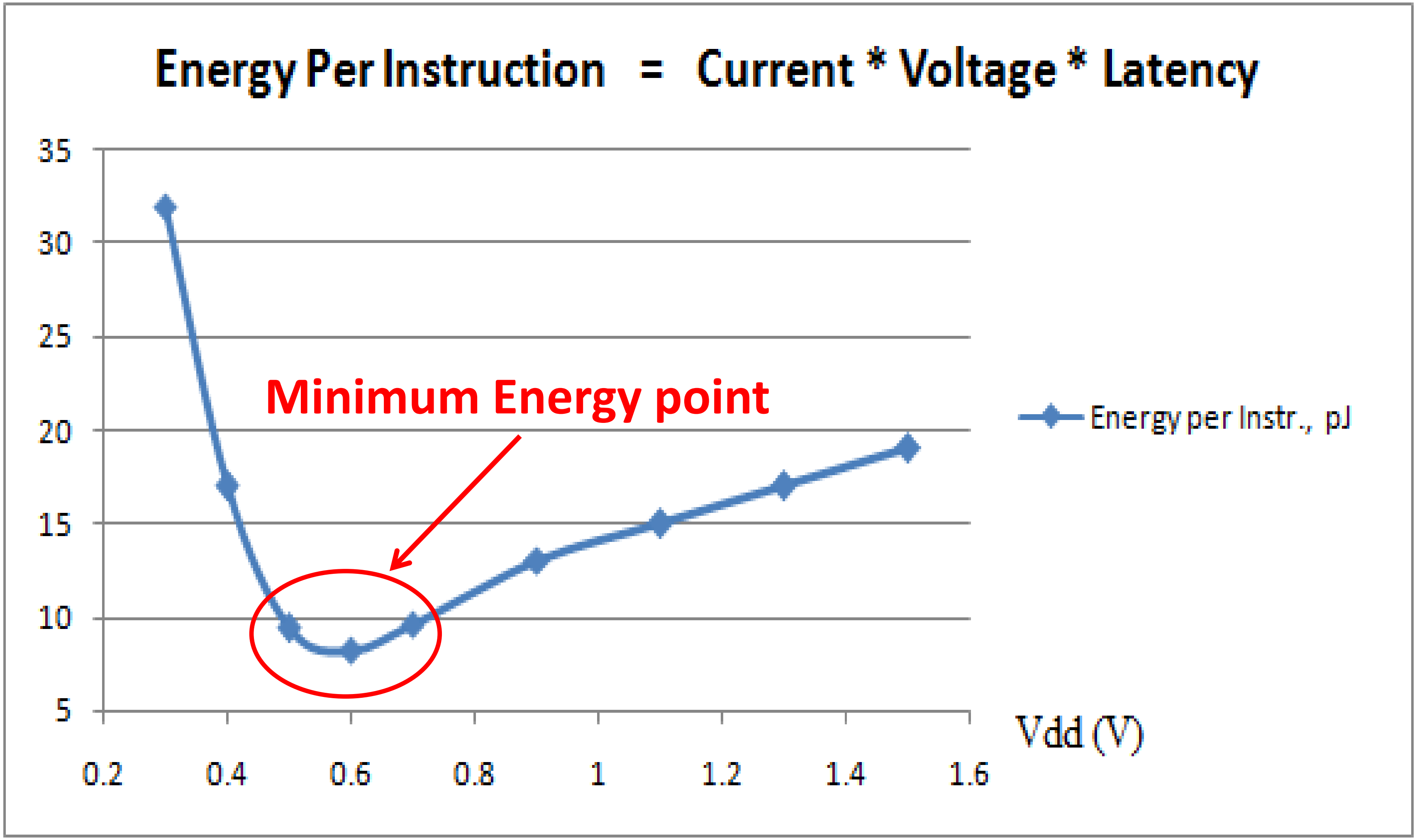
- ✓ Fully asynchronous implementation (bundled data protocol)
  - – adjustable delay lines
- ✓Fault tolerance operation
- ✓DFT integration

Some measurements…

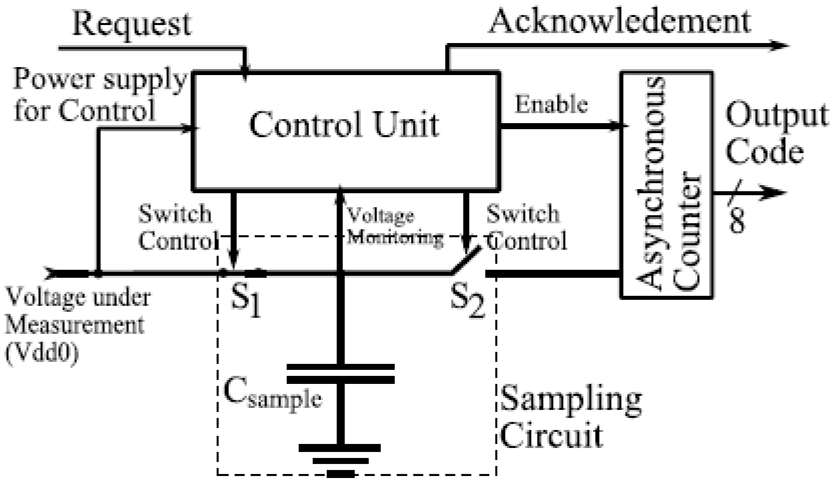


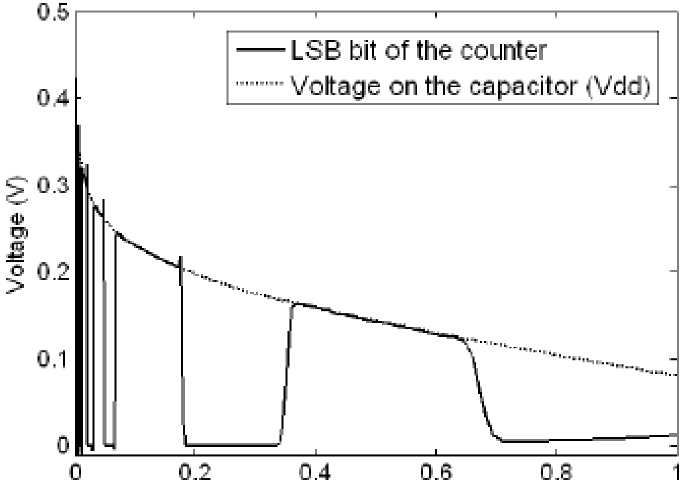
- 0.89V to 1.5V: full capability mode.
- 0.74V to 0.89V: at 0.89V the RAM starts to fail, so the chip operates using
- 0.22V to 0.74V: at 0.74V the program counter starts to fail, however the control logic synthesised using the CPOG model continues to operate correctly down to 0.22V
- 67 MIPS at 1.2 V.
- ~2700 instructions per second at 0.25V.

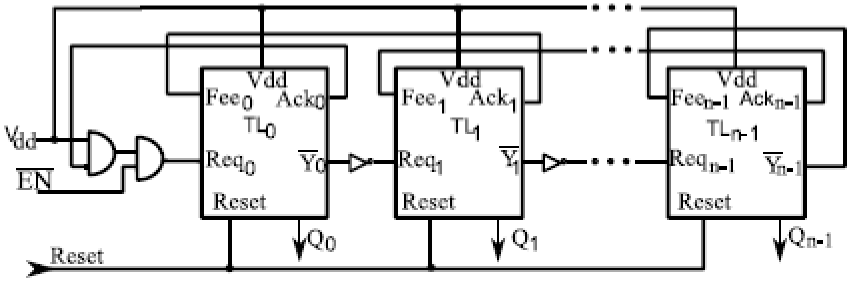
Energy Efficiency (measurements on real silicon - asynchronous 8051)

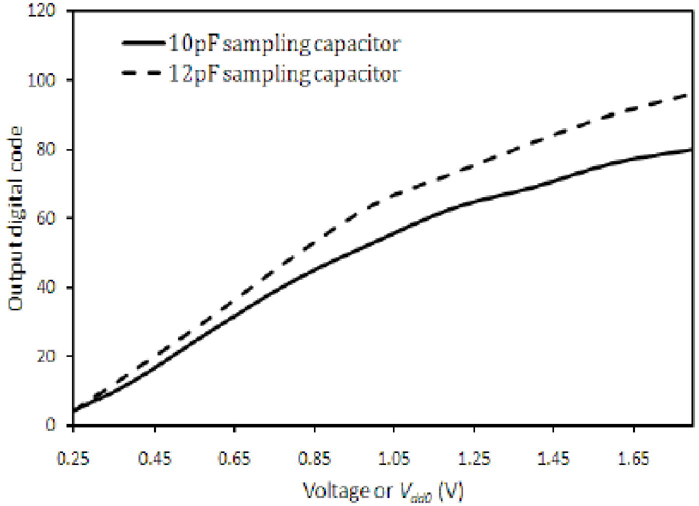


Reference-free Voltage Sensor

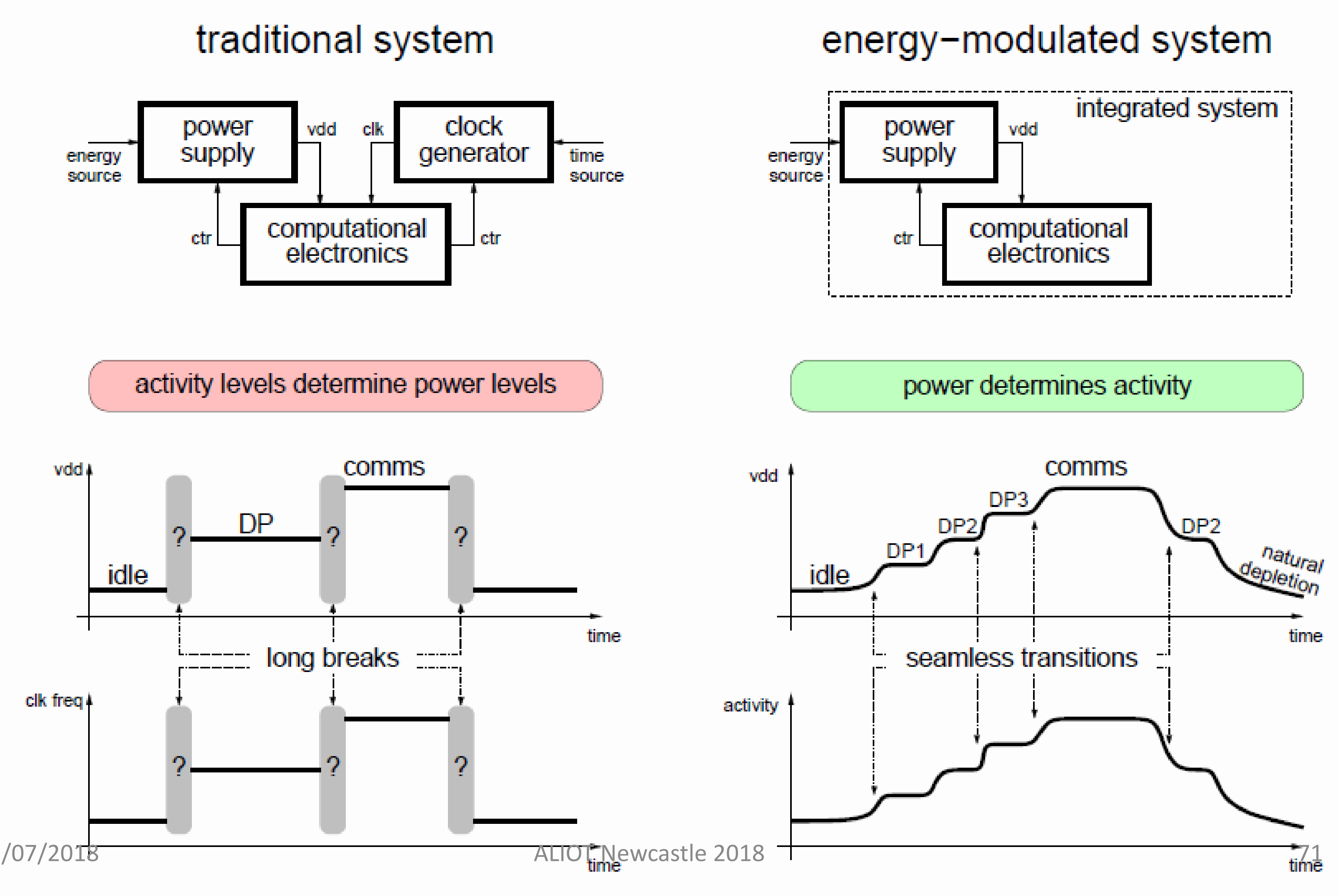




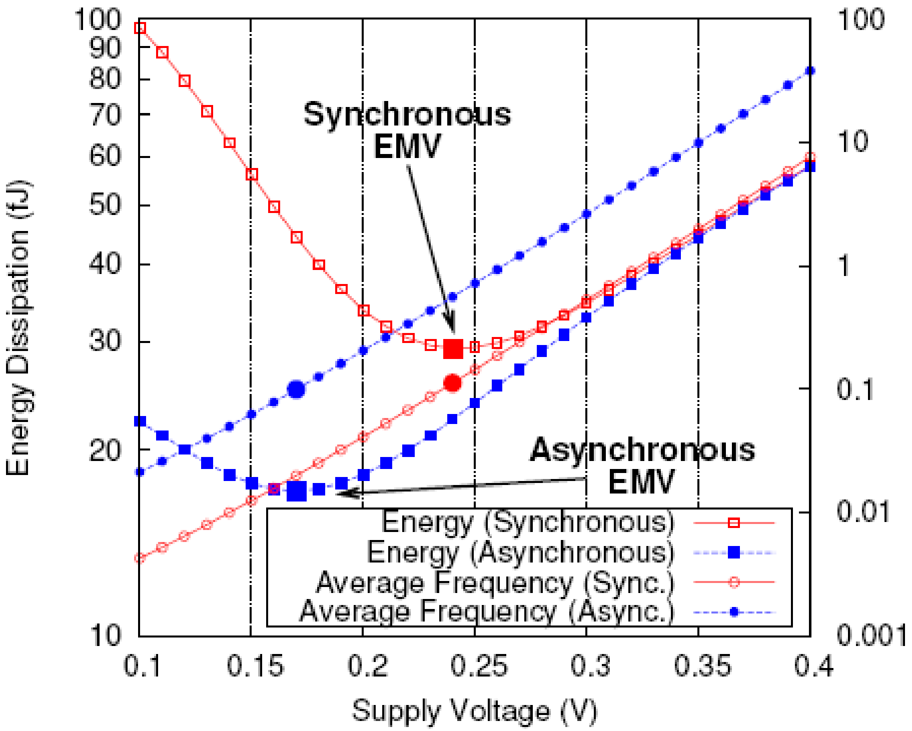




Traditional vs energy-modulated view



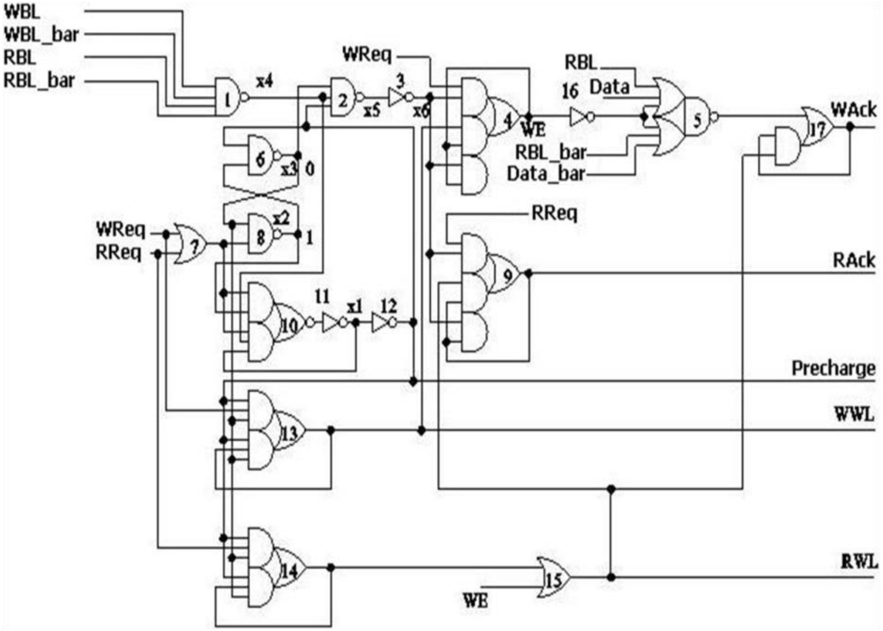
Synchronous vs Self-Timed Design (in terms of energy efficiency)



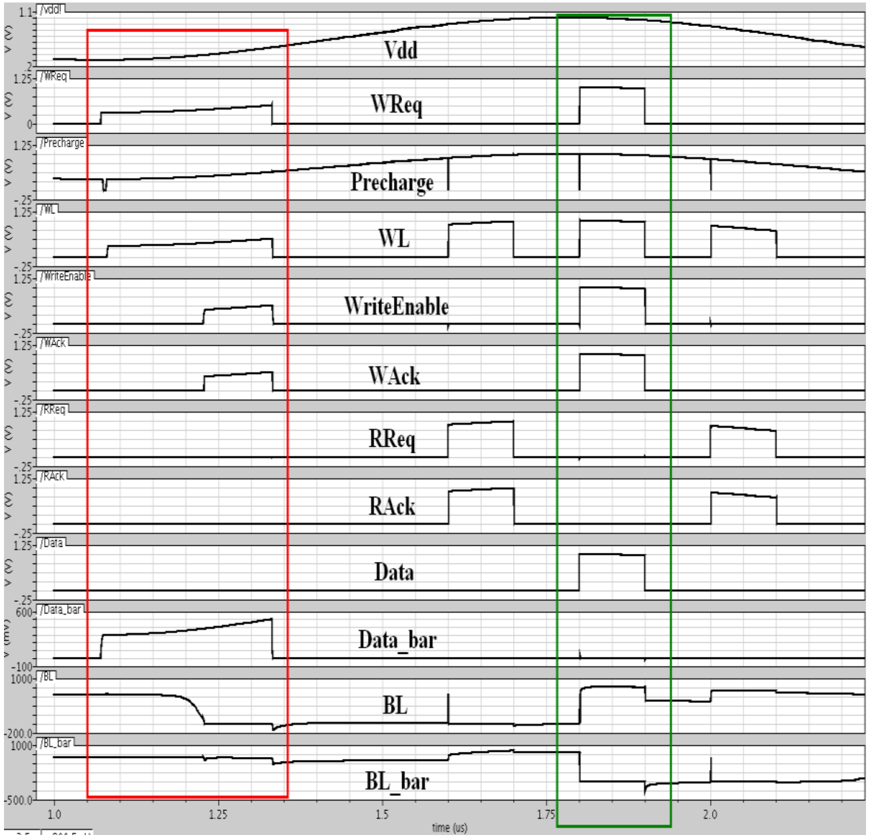
Asynchronous (self-timed) logic can provide completion detection and thus reduce the interval of leakage to minimum, thereby doing nothing well!

Source: Akgun et al, ASYNC'10

Self-timed SRAM



SRAM Controller

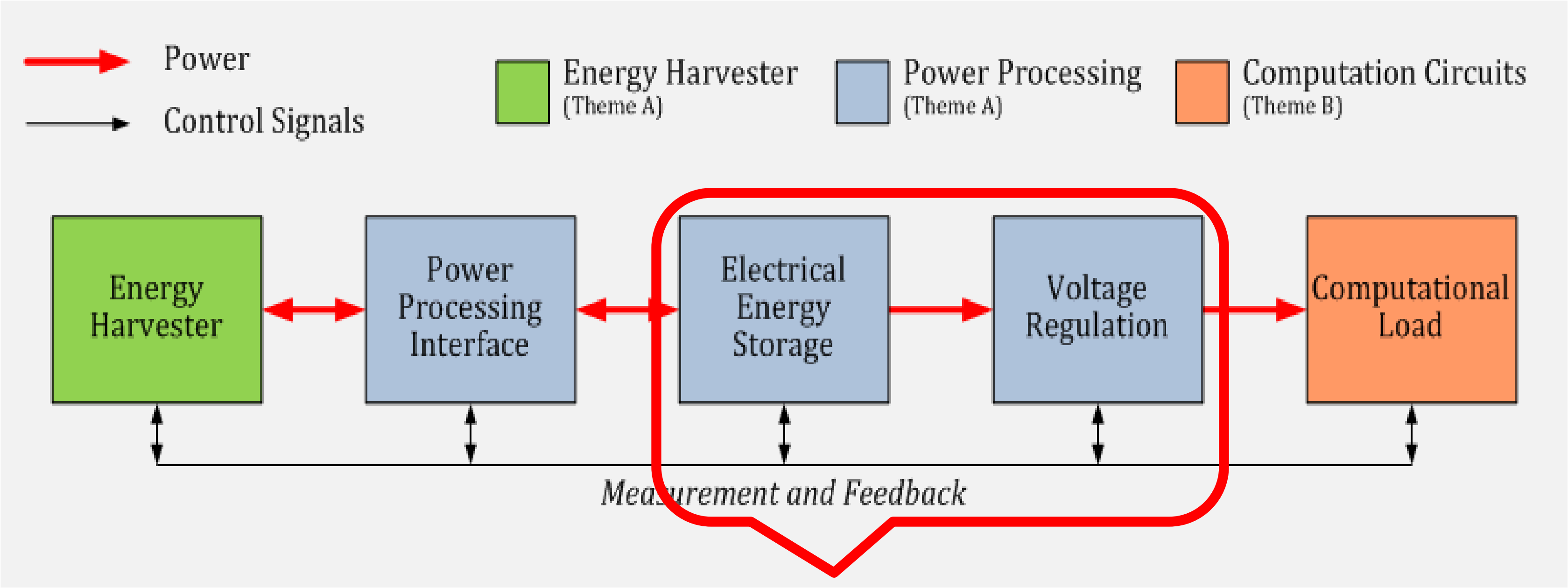


Today: living with Analog, new model mining, …

- Asynchronous for “little digital”, self-powered sensors, ADCs; synthesis of STG specifications
- Projects: SAVVIE, A4A
- People: Kaiyuan Gao, Yuqing Xu, Vladimir Dubikhin, Jonathan Beaumont, Austin Ogweno, Benafa Oyinkuro, Sergey Mileiko; Bernard Stark (Bristol)
- Industrial collaboration: Dialog Semiconductor

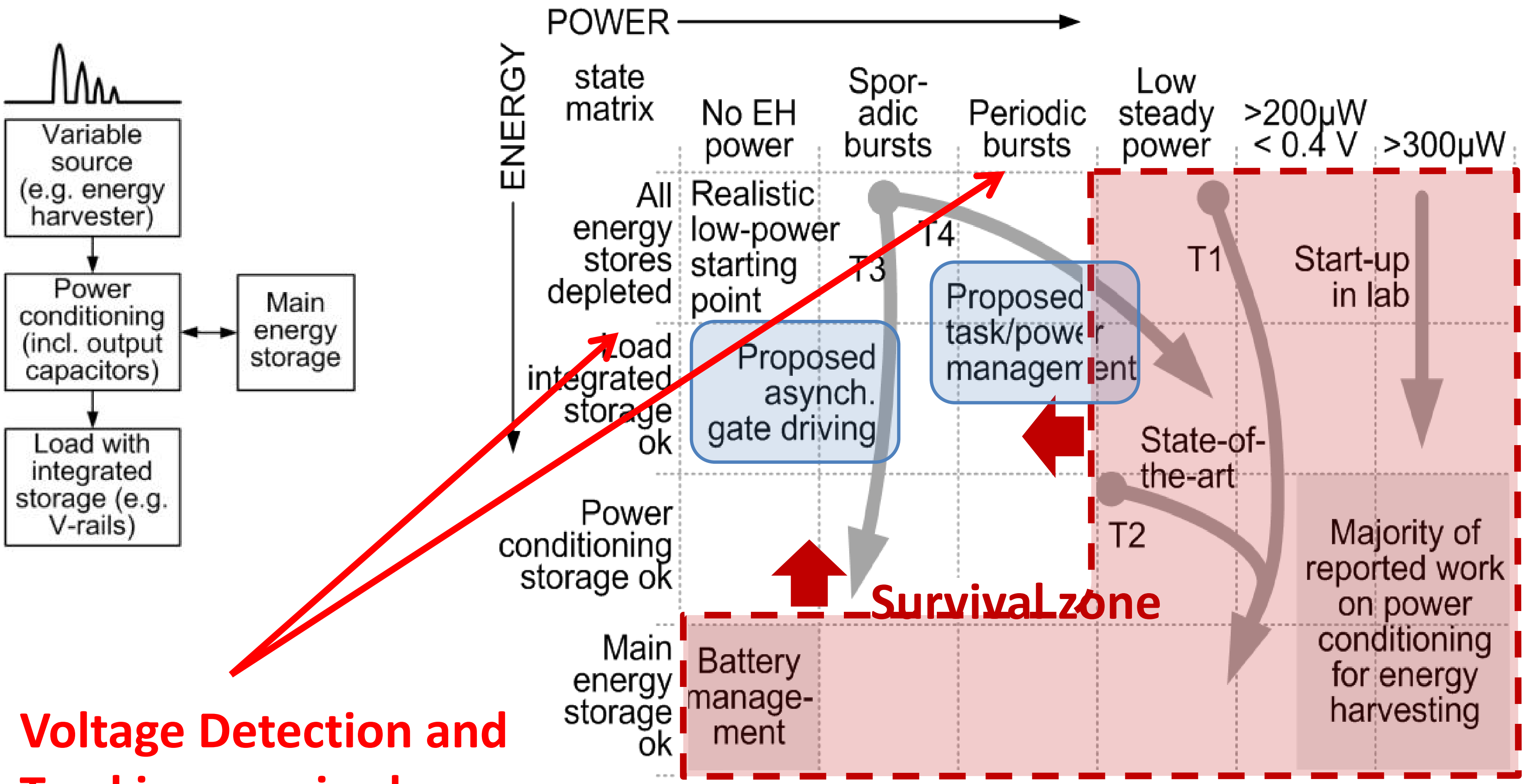
Energy harvesting systems

Using Holistic project vision:


Power
Control Signals
Energy Harvester (Theme A)
Power Processing (Theme A)
Computation Circuits (Theme B)
Energy Harvester
Power Processing Interface
Electrical Energy Storage
Voltage Regulation
Computational Load
Measurement and Feedback

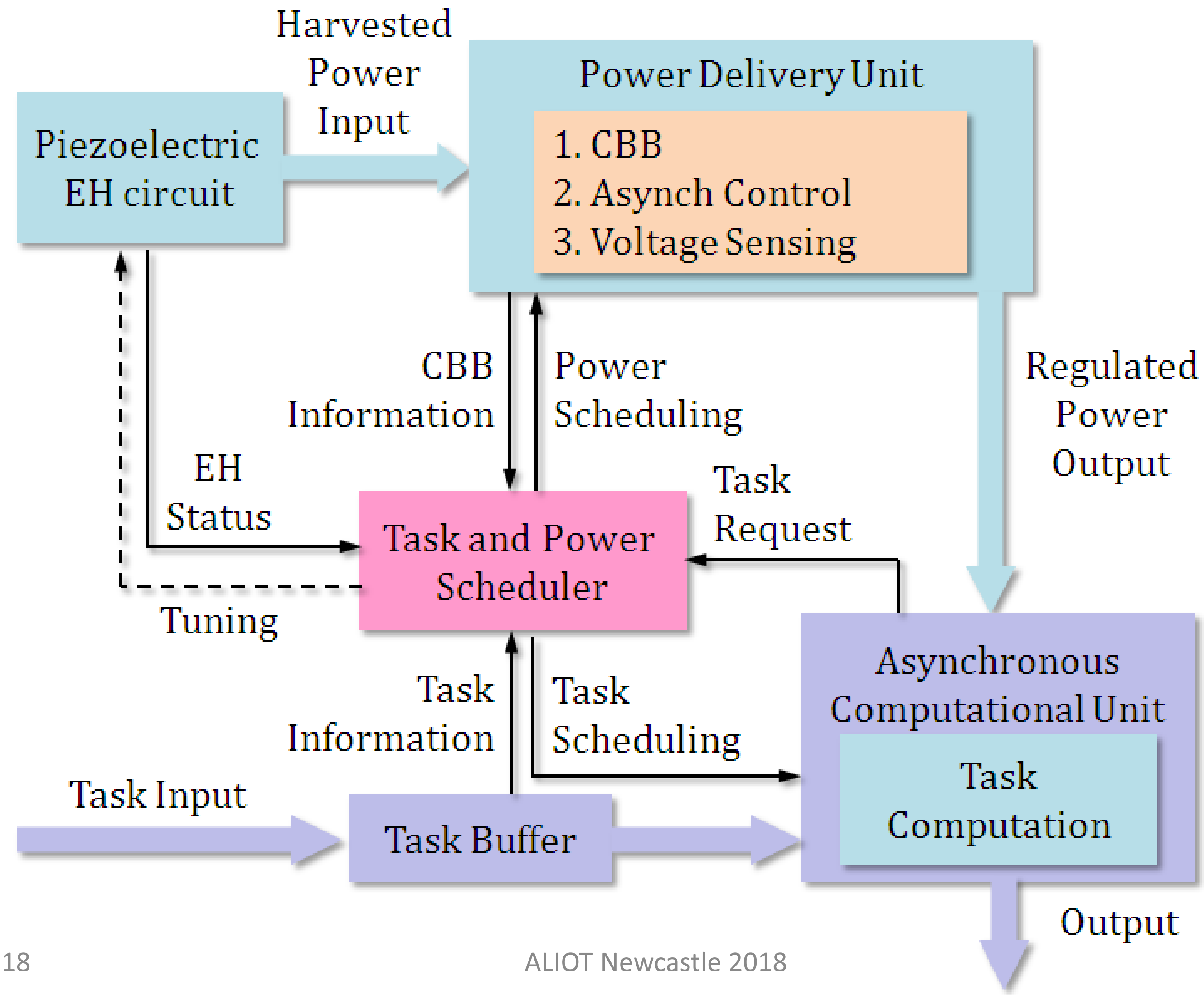
Sporadic source of energy does not allow for fancy power processing and therefore large storage

Staying alive in variable, intermittent, low-power environments (Savvie Project)

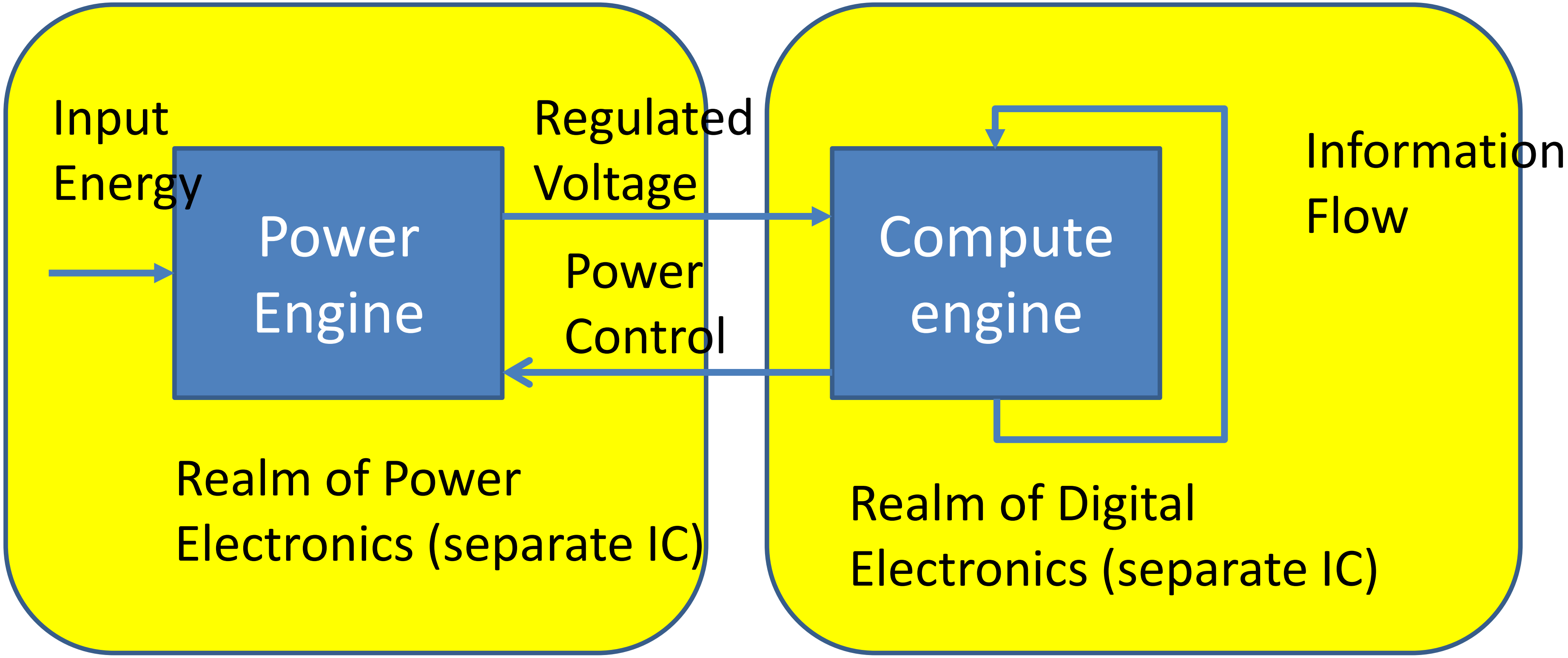

Variable source (e.g. energy harvester)
Power conditioning (incl. output capacitors)
Main energy storage
Load with integrated storage (e.g. V-rails)
POWER
ENERGY
state matrix
No EH power
Spor-adic bursts
Periodic bursts
Low steady power
>200μW < 0.4 V
>300μW
All energy stores depleted
Load integrated storage ok
Power conditioning storage ok
Main energy storage ok
Realistic low-power starting point
T1
T2
T3
T4
Proposed asynch. gate driving
Proposed task/power management
Start-up in lab
State-of-the-art
Majority of reported work on power conditioning for energy harvesting
Battery manage-ment
Survival zone

Voltage Detection and Tracking required

From Holistic systems to Power-Compute Co-design

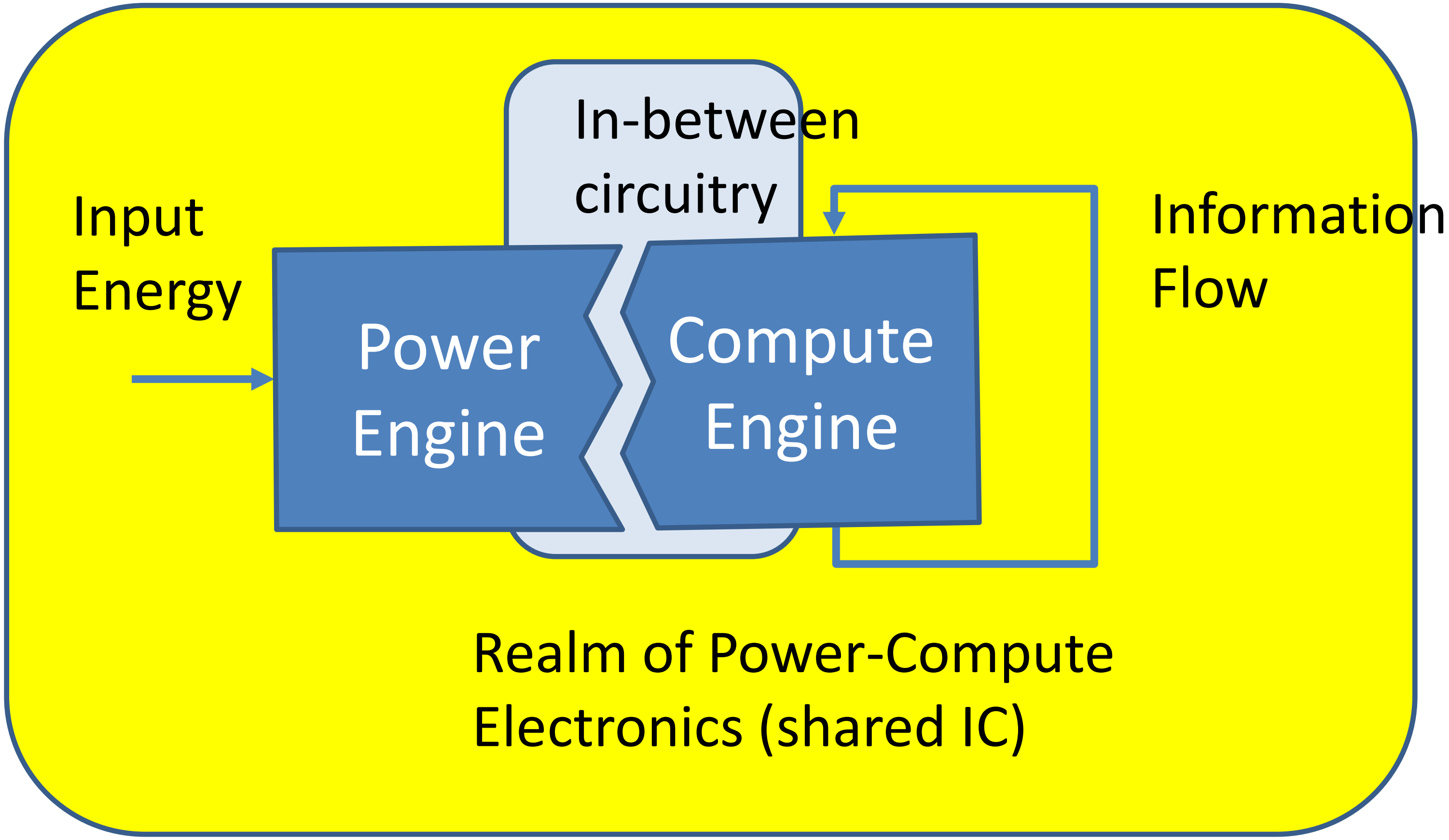


Traditional Power-Compute Divide



Separate optimization cycles

Power-Compute Co-design



Co-optimization cycle

Thank you!