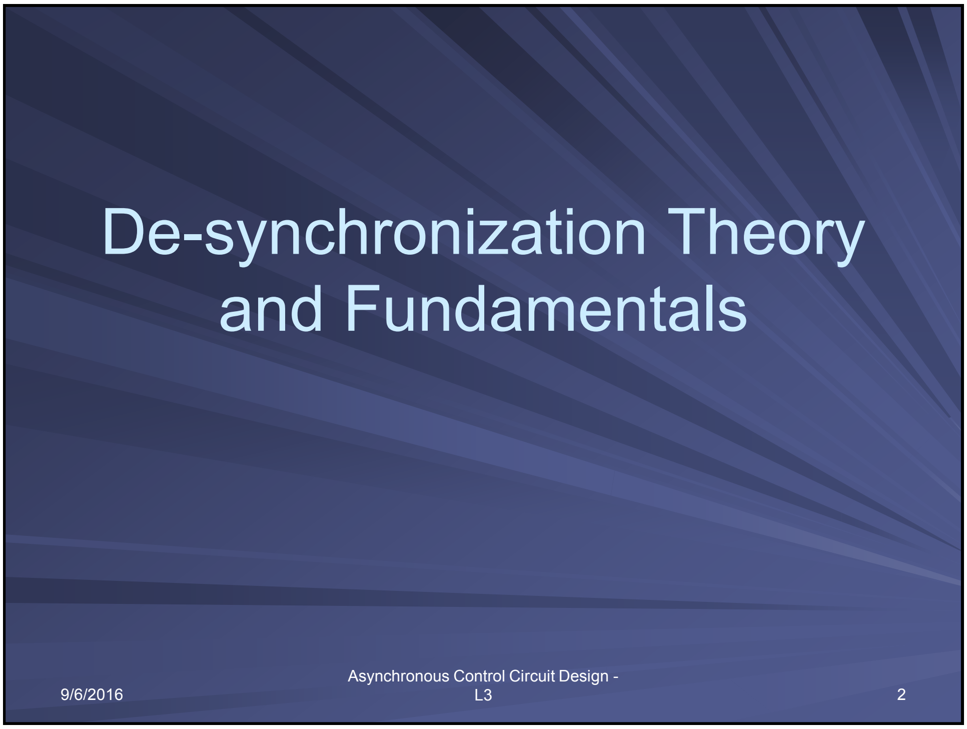
UNIVERSITY OF THESSALY

TECHNION Israel Institute of Technology

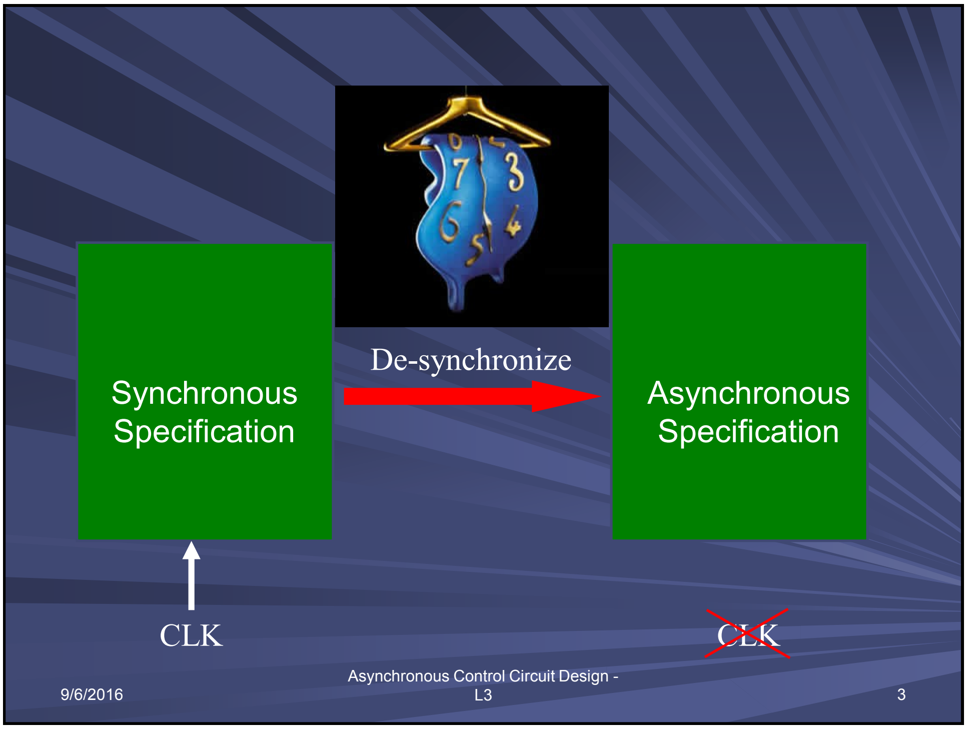
USC Viterbi School of Engineering University of Southern California

Asynchronous Design Seminar at University of Verona – Lecture Notes 3

De-Synchronization



De-synchronization Theory and Fundamentals



Synchronous Specification

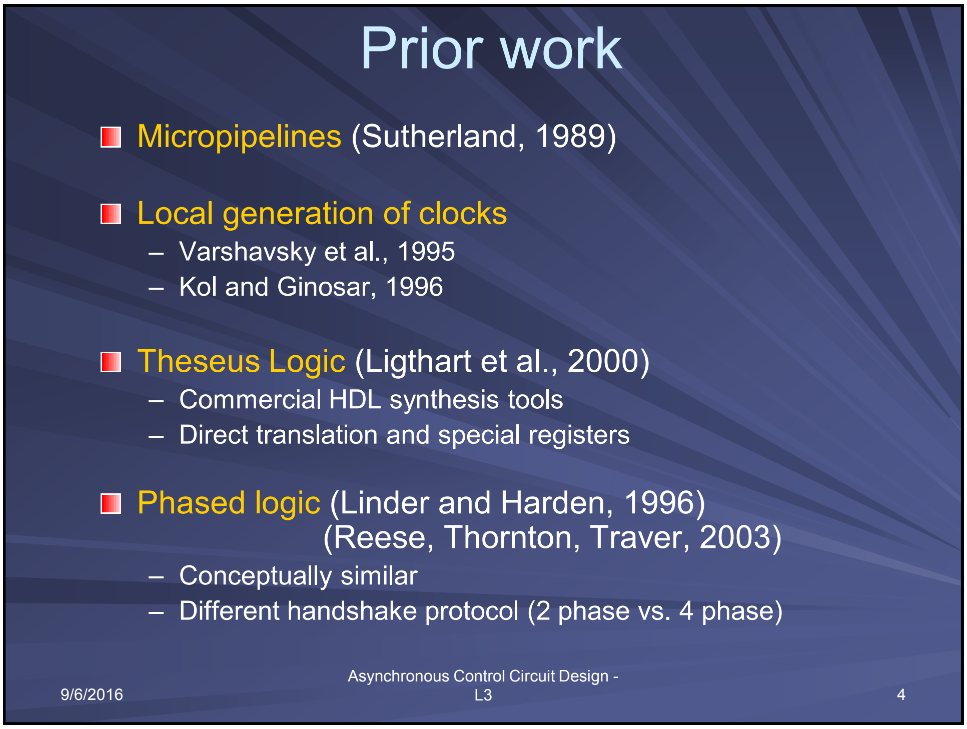
De-synchronize

Asynchronous Specification

CLK

~~CLK~~

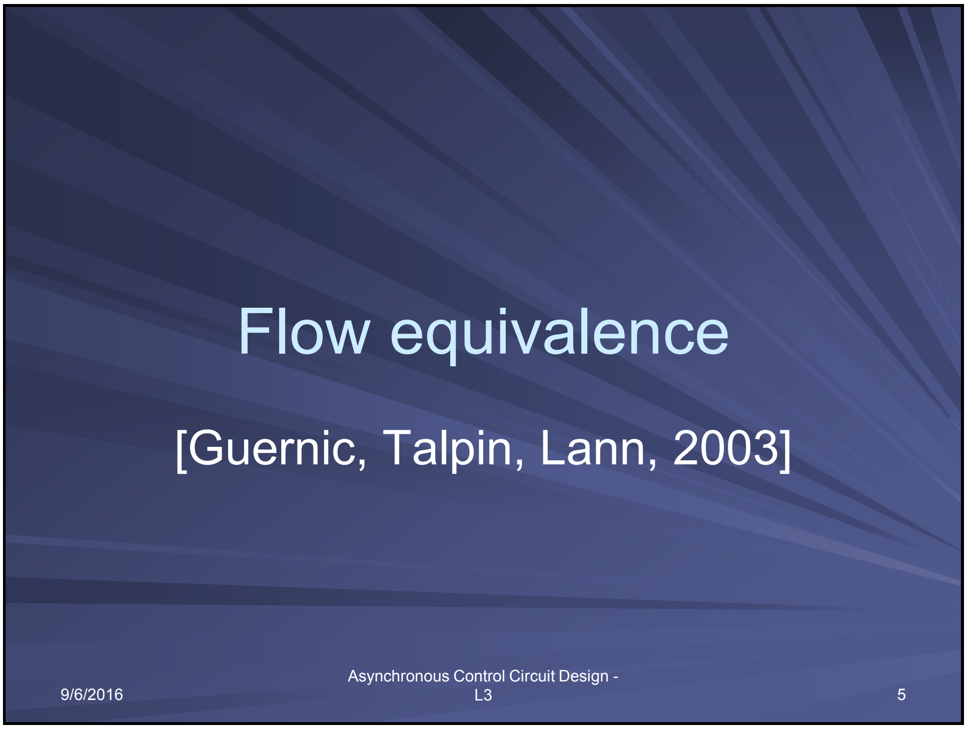
9/6/2016 Asynchronous Control Circuit Design - L3 3



Prior work

- Micropipelines (Sutherland, 1989)
- Local generation of clocks
  - Varshavsky et al., 1995
  - Kol and Ginosar, 1996
- Theseus Logic (Ligthart et al., 2000)
  - Commercial HDL synthesis tools
  - Direct translation and special registers
- Phased logic (Linder and Harden, 1996)
  (Reese, Thornton, Traver, 2003)
  - Conceptually similar
  - Different handshake protocol (2 phase vs. 4 phase)

9/6/2016 Asynchronous Control Circuit Design - L3 4



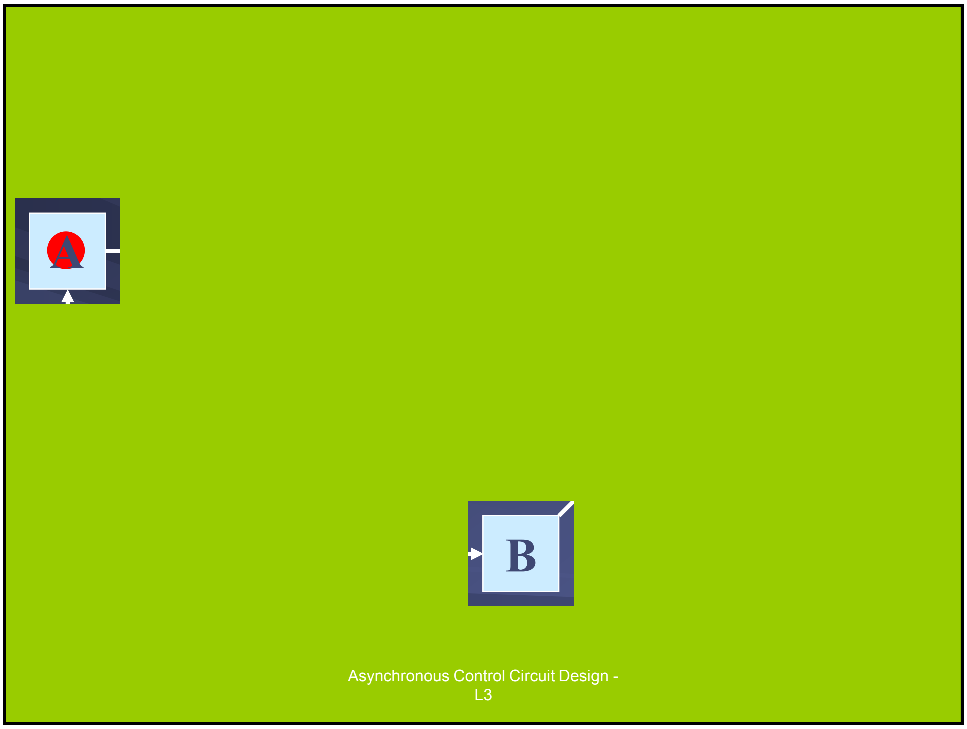
Flow equivalence

[Guernic, Talpin, Lann, 2003]

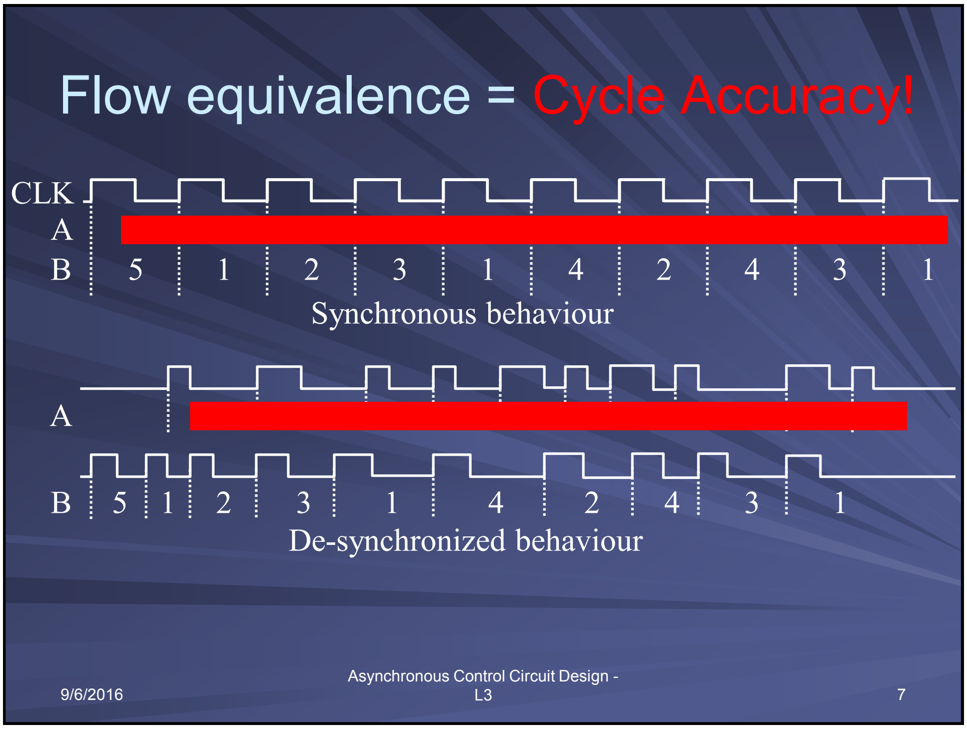
9/6/2016

Asynchronous Control Circuit Design - L3

5



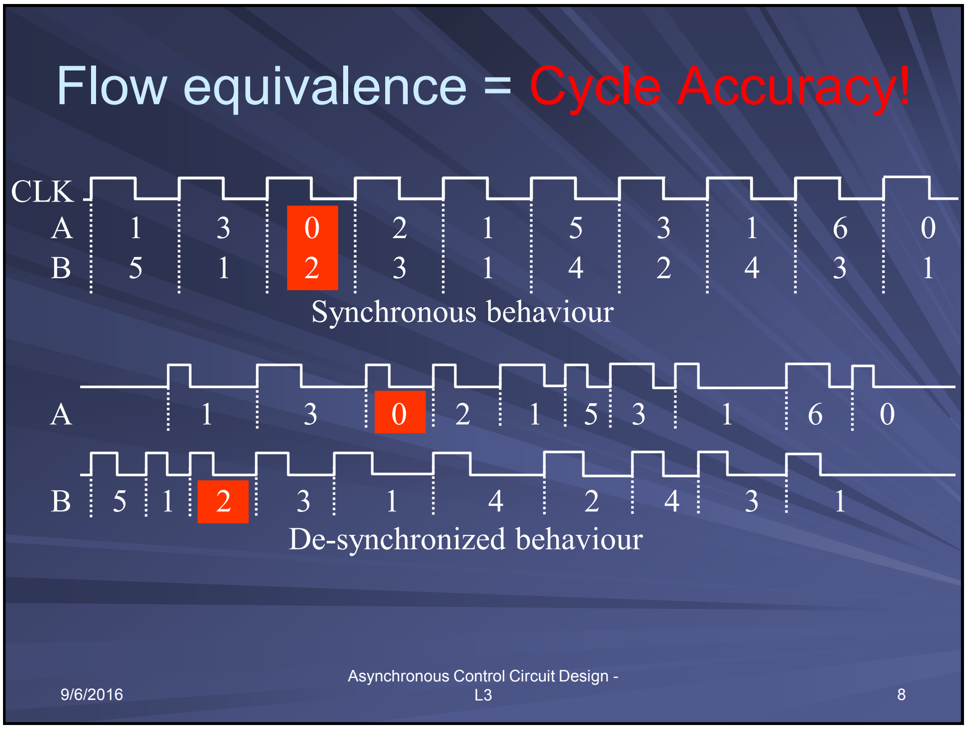
Asynchronous Control Circuit Design - L3



Flow equivalence = Cycle Accuracy!

Synchronous behaviour

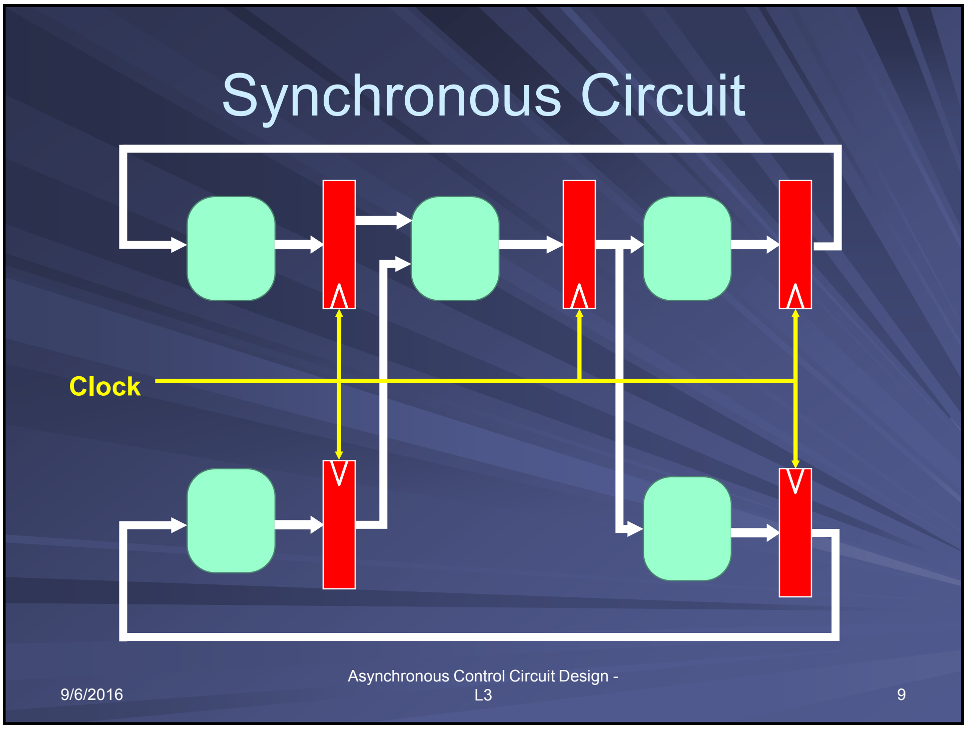
De-synchronized behaviour



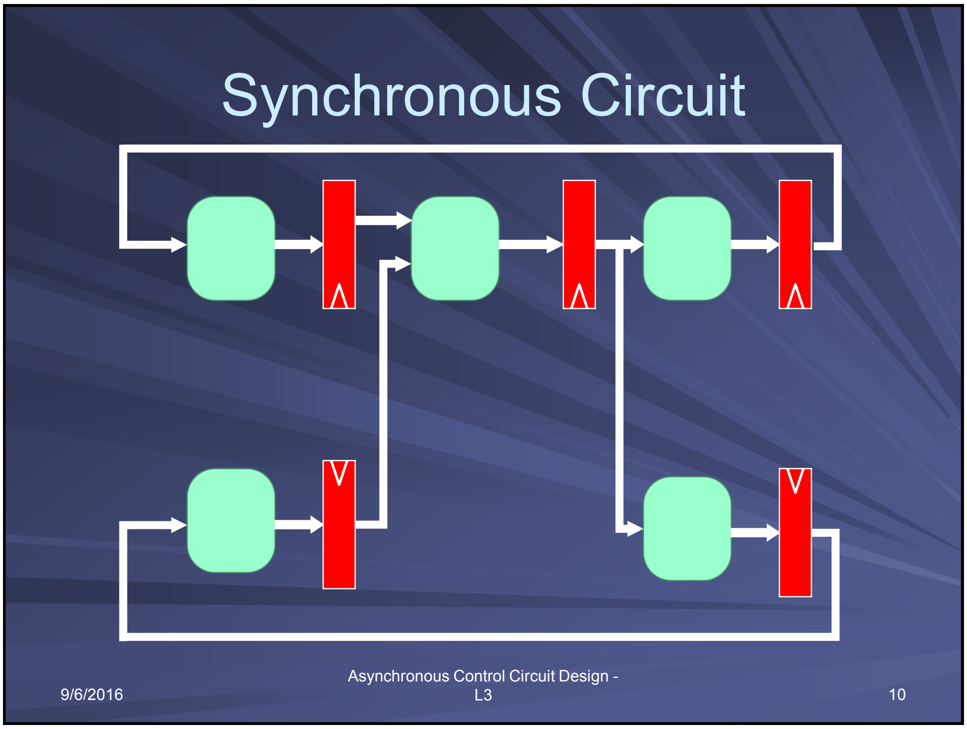
Flow equivalence = Cycle Accuracy!

Synchronous behaviour

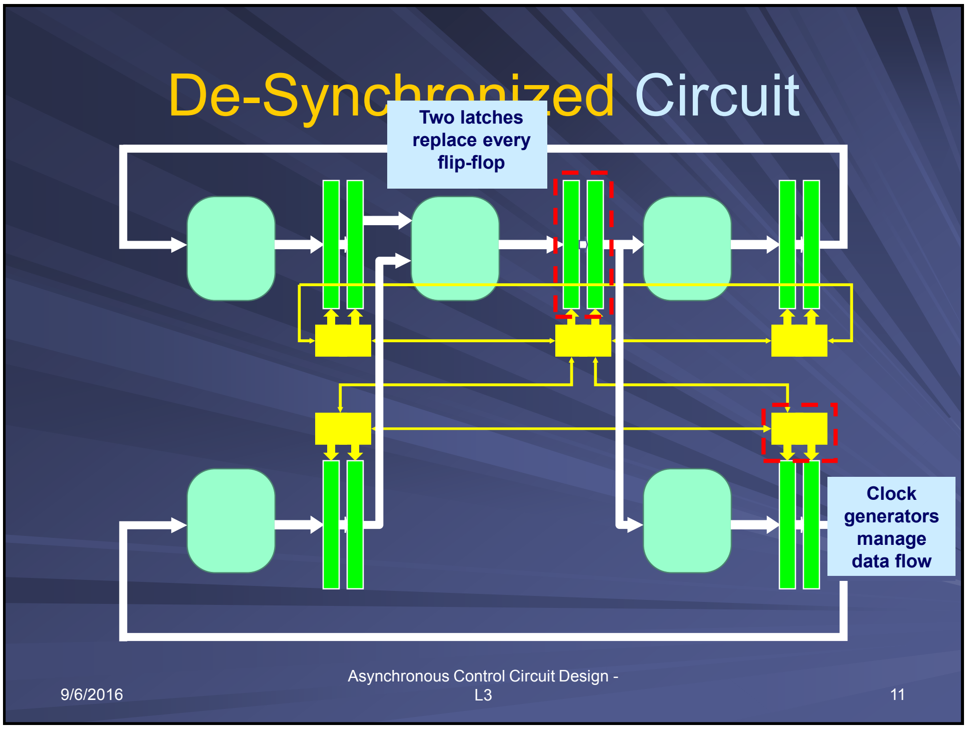
De-synchronized behaviour



Synchronous Circuit



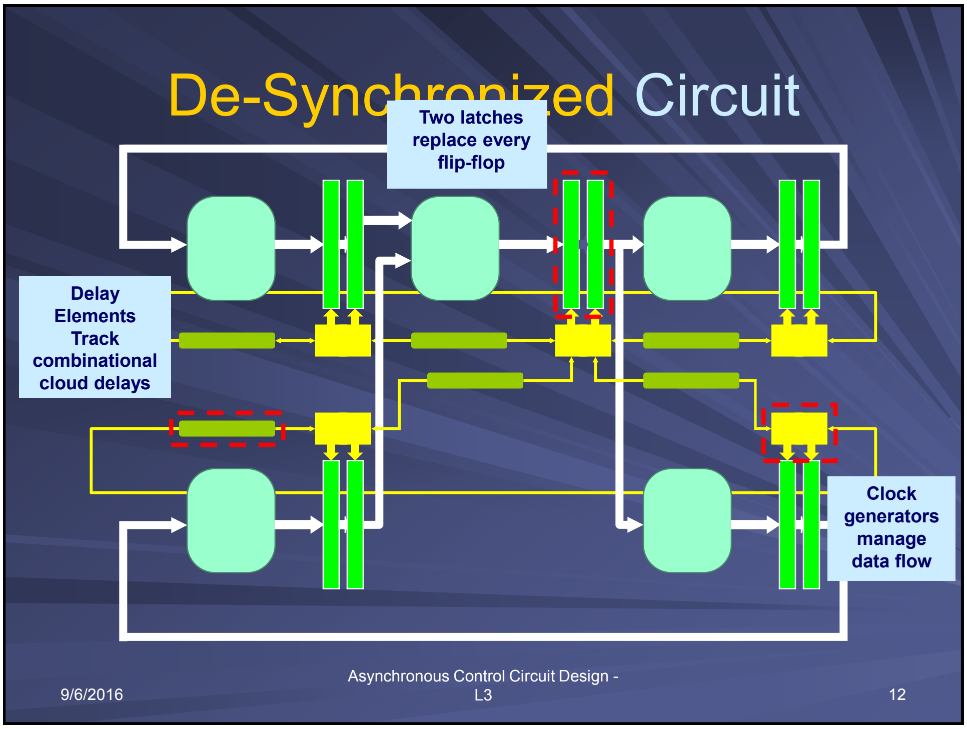
Synchronous Circuit



De-Synchronized Circuit

Two latches replace every flip-flop

Clock generators manage data flow

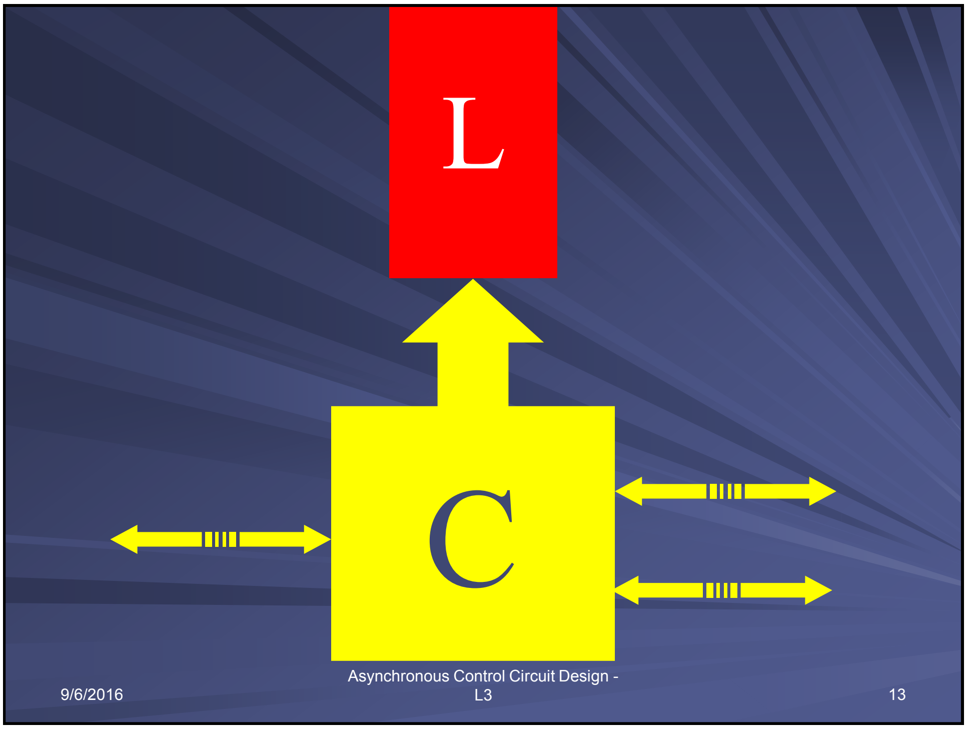


De-Synchronized Circuit

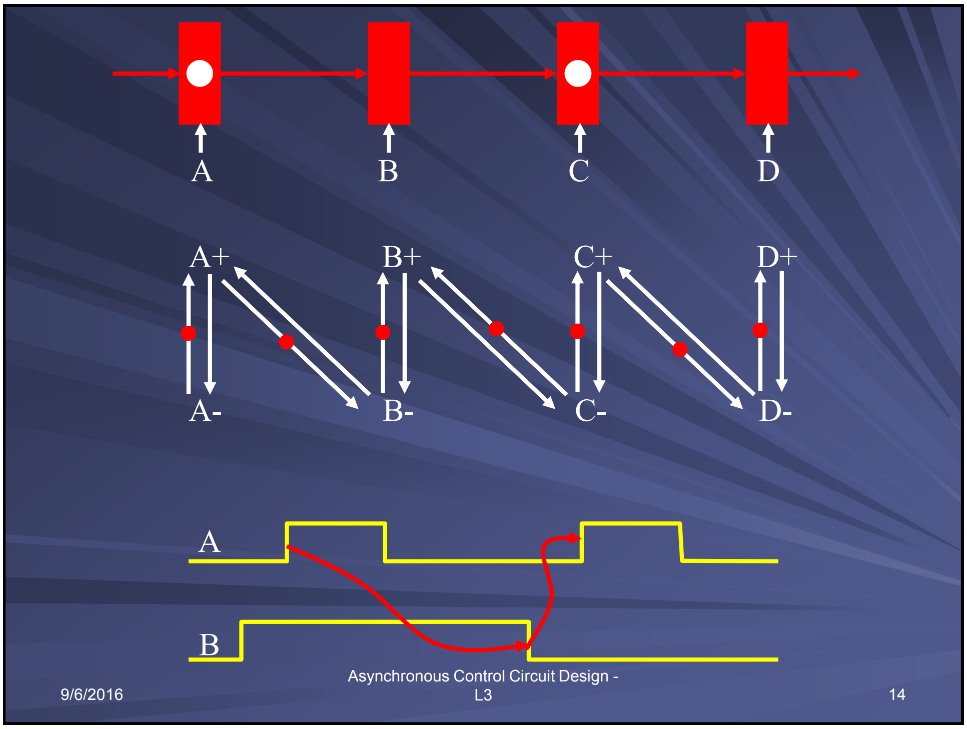
Two latches replace every flip-flop

Delay Elements Track combinational cloud delays

Clock generators manage data flow

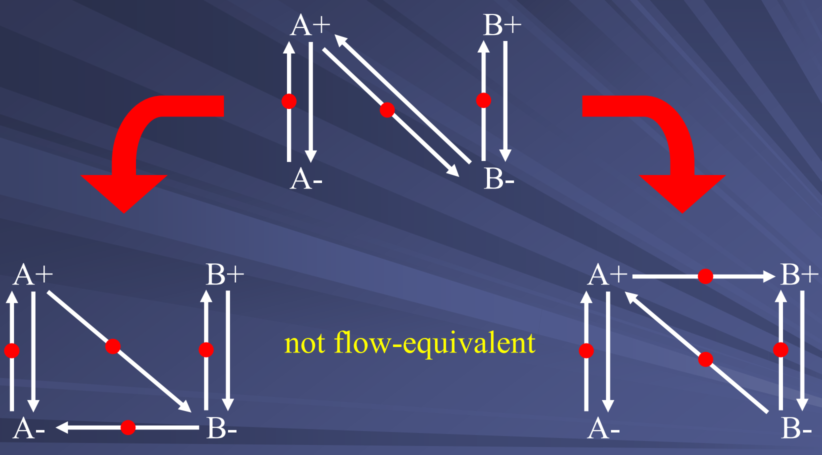


9/6/2016 Asynchronous Control Circuit Design - L3 13

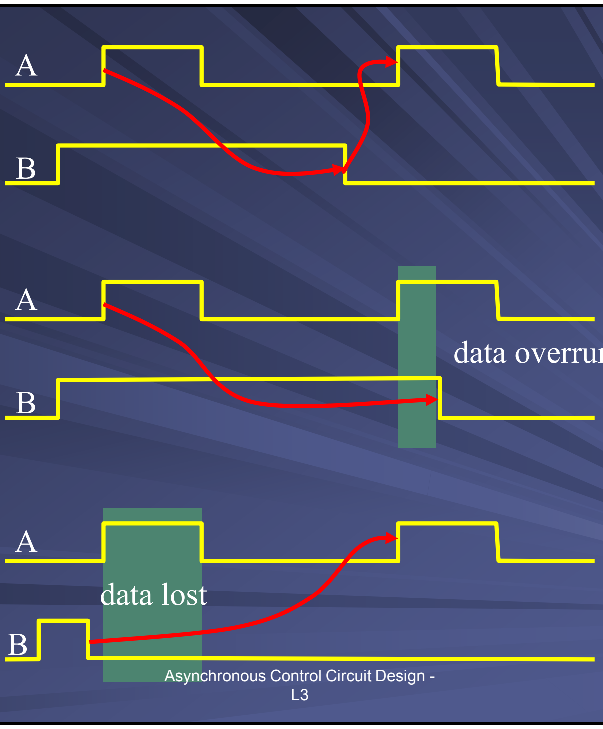


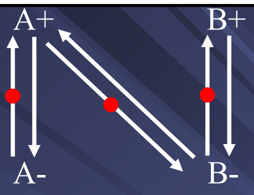
9/6/2016 Asynchronous Control Circuit Design - L3 14

Can we increase concurrency ?



not flow-equivalent



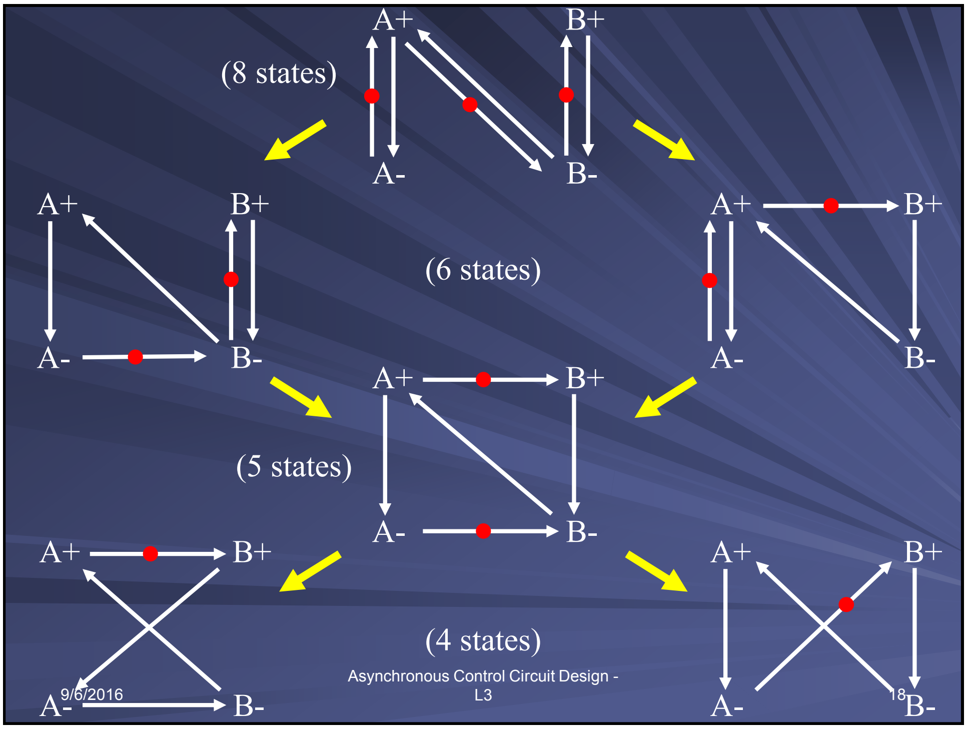


Can we reduce concurrency ? How much ?

9/6/2016

Asynchronous Control Circuit Design - L3

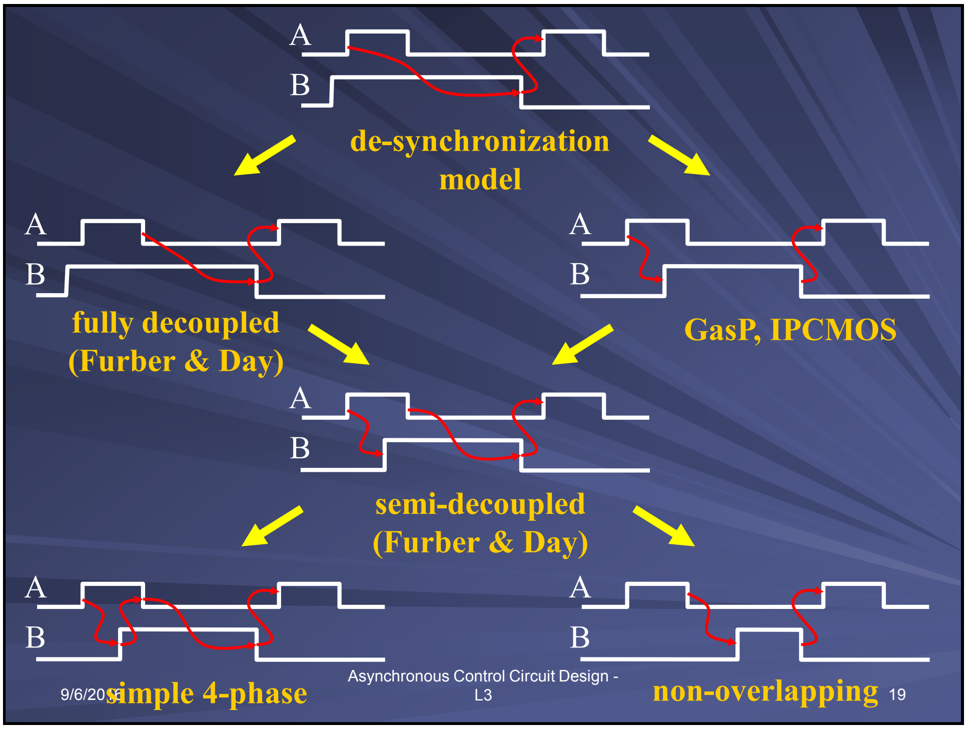
17



9/6/2016

Asynchronous Control Circuit Design - L3

18



de-synchronization model

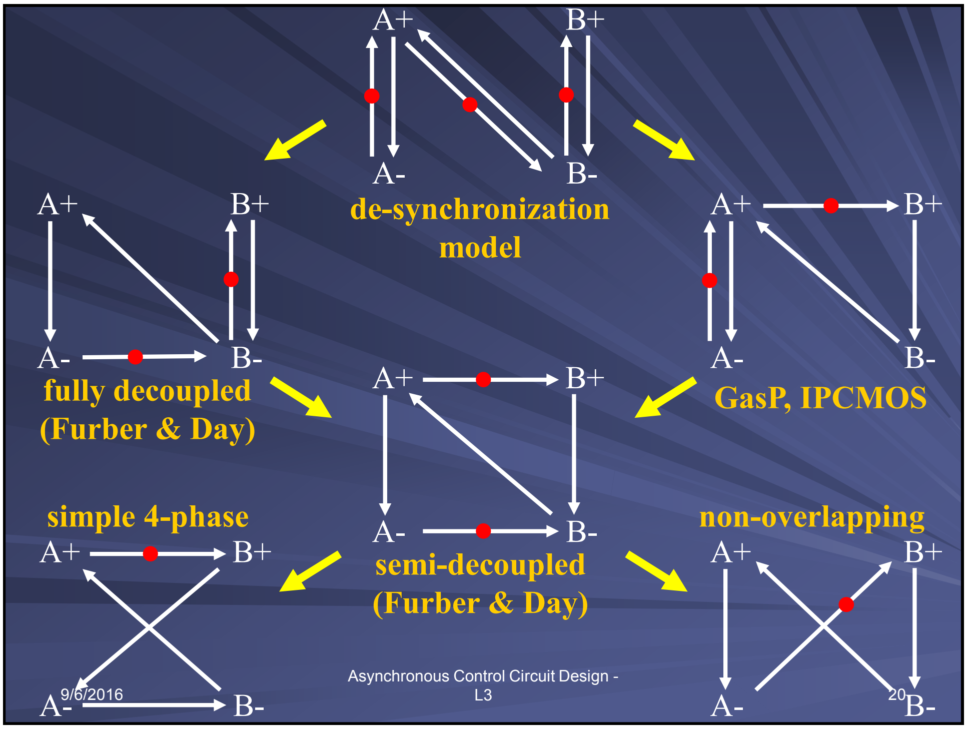
fully decoupled (Furber & Day)

GasP, IPCMOS

semi-decoupled (Furber & Day)

simple 4-phase

non-overlapping



de-synchronization model

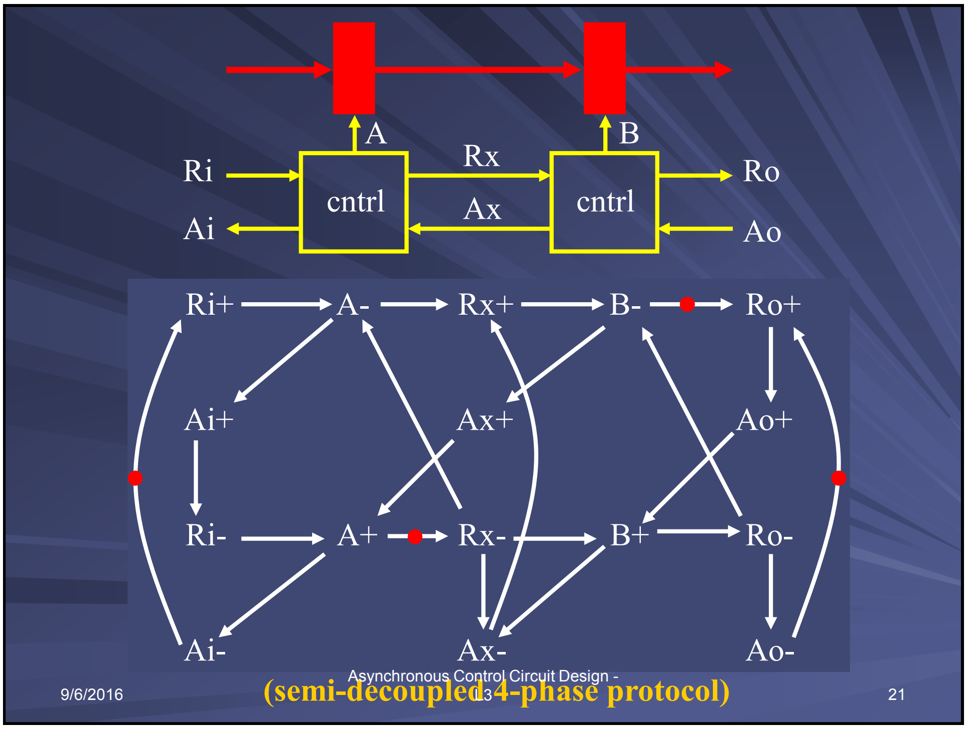
fully decoupled (Furber & Day)

GasP, IPCMOS

simple 4-phase

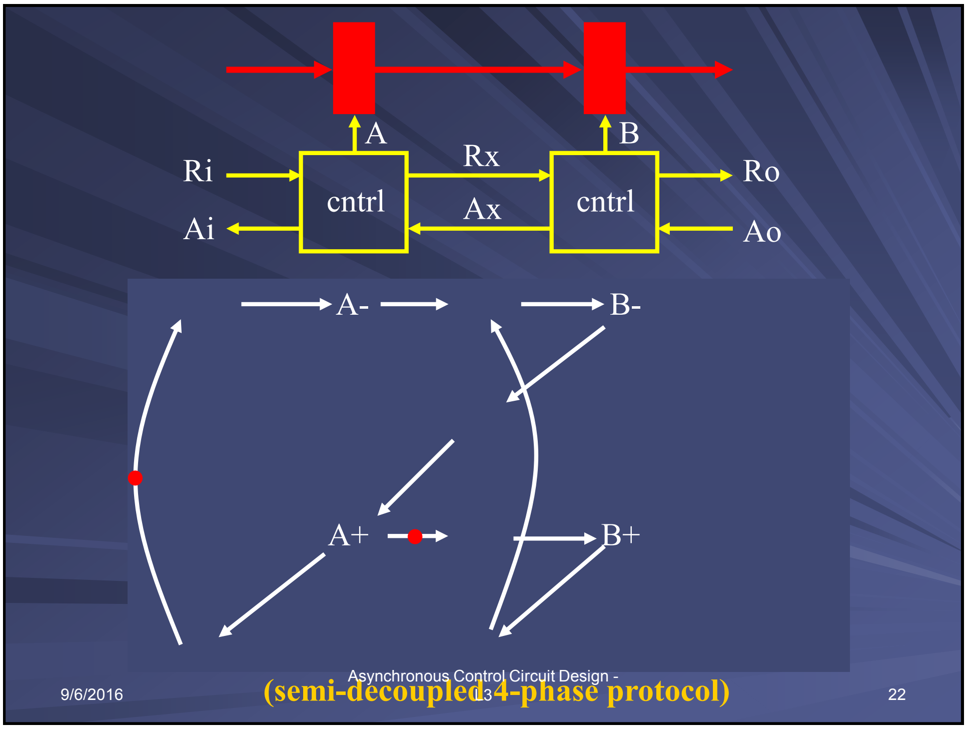
semi-decoupled (Furber & Day)

non-overlapping



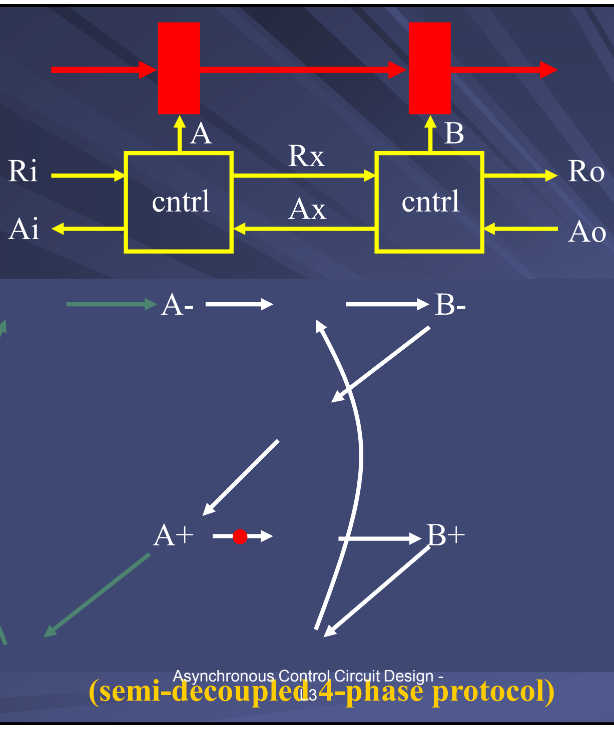
(semi-decoupled 4-phase protocol)

9/6/2016 Asynchronous Control Circuit Design - 13 21

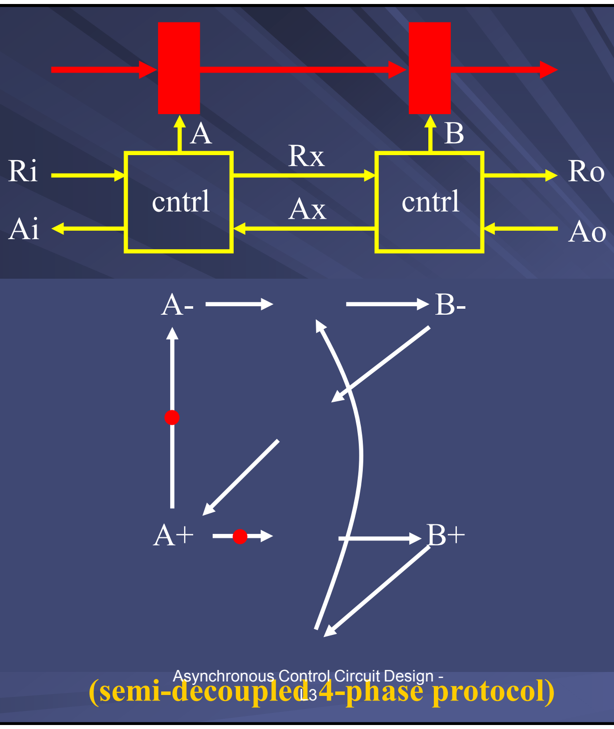


(semi-decoupled 4-phase protocol)

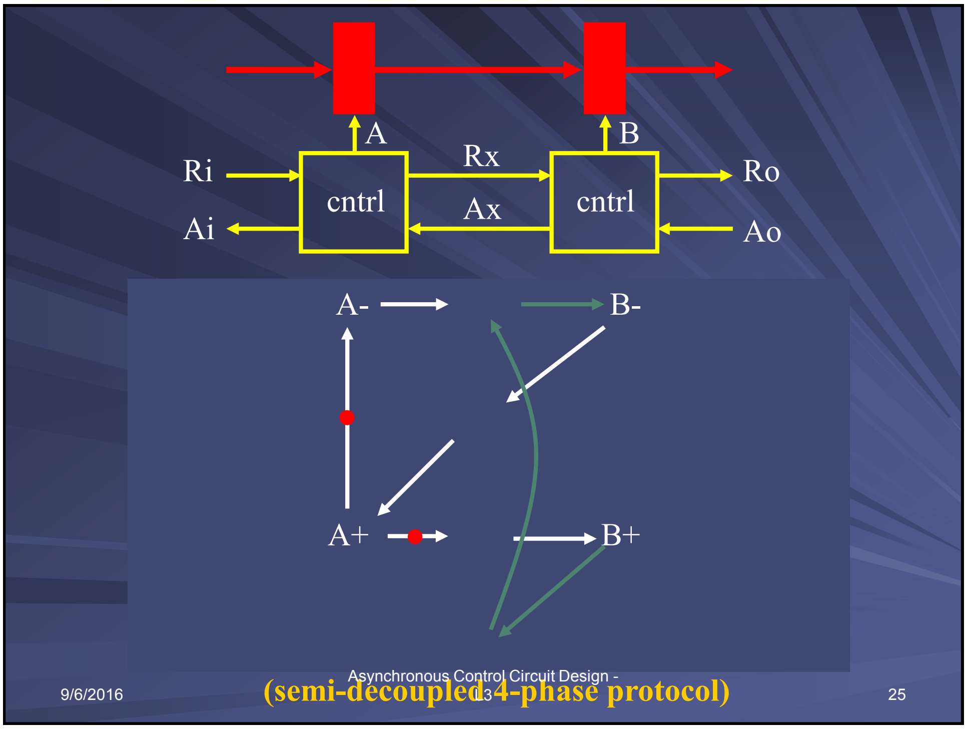
9/6/2016 Asynchronous Control Circuit Design - 13 22



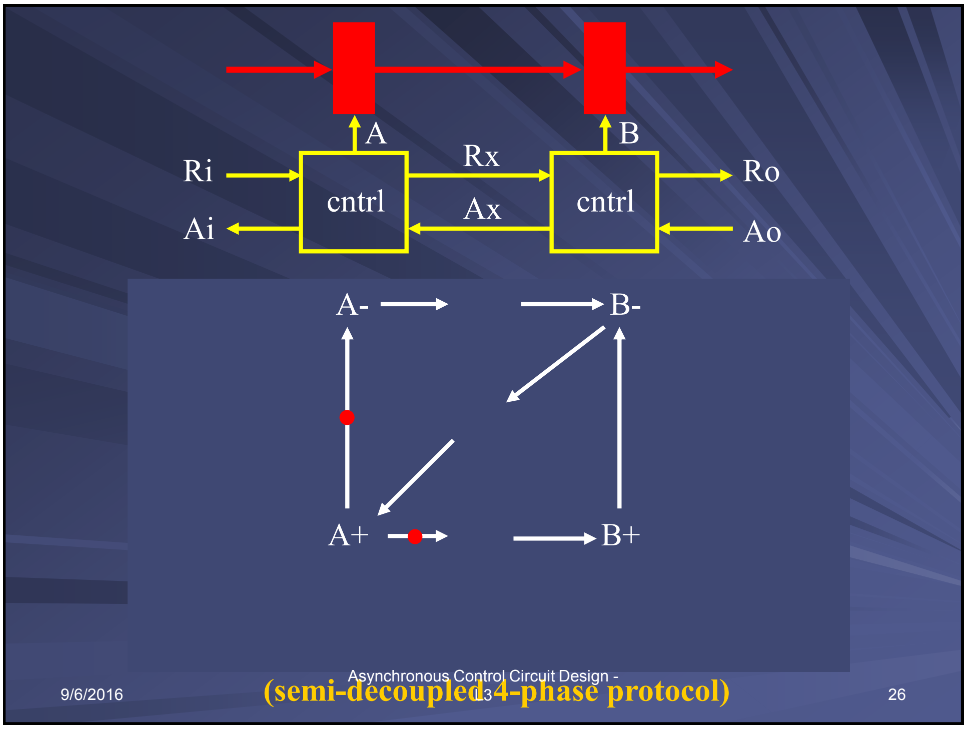
(semi-decoupled 4-phase protocol)



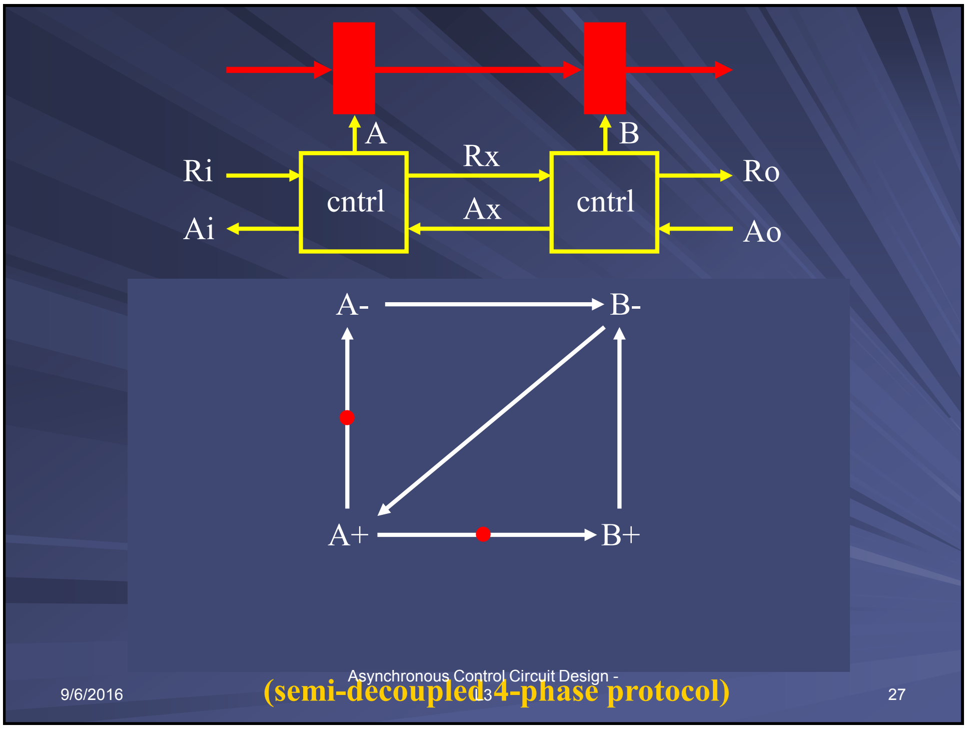
(semi-decoupled 4-phase protocol)



(semi-decoupled 4-phase protocol)



(semi-decoupled 4-phase protocol)

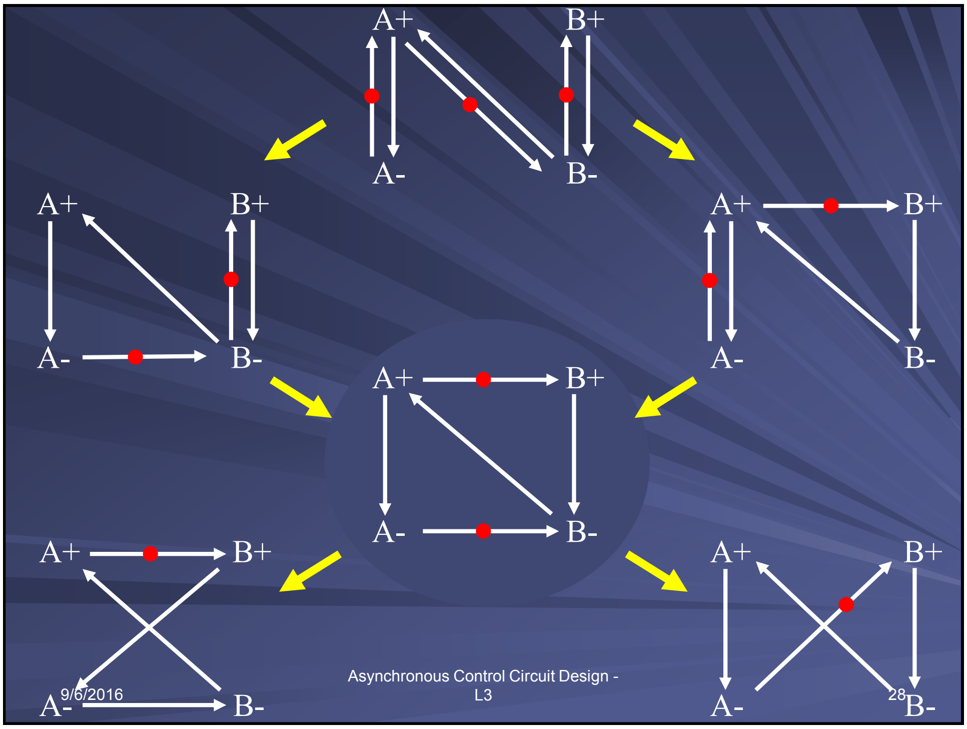


(semi-decoupled 4-phase protocol)

9/6/2016

Asynchronous Control Circuit Design - L3

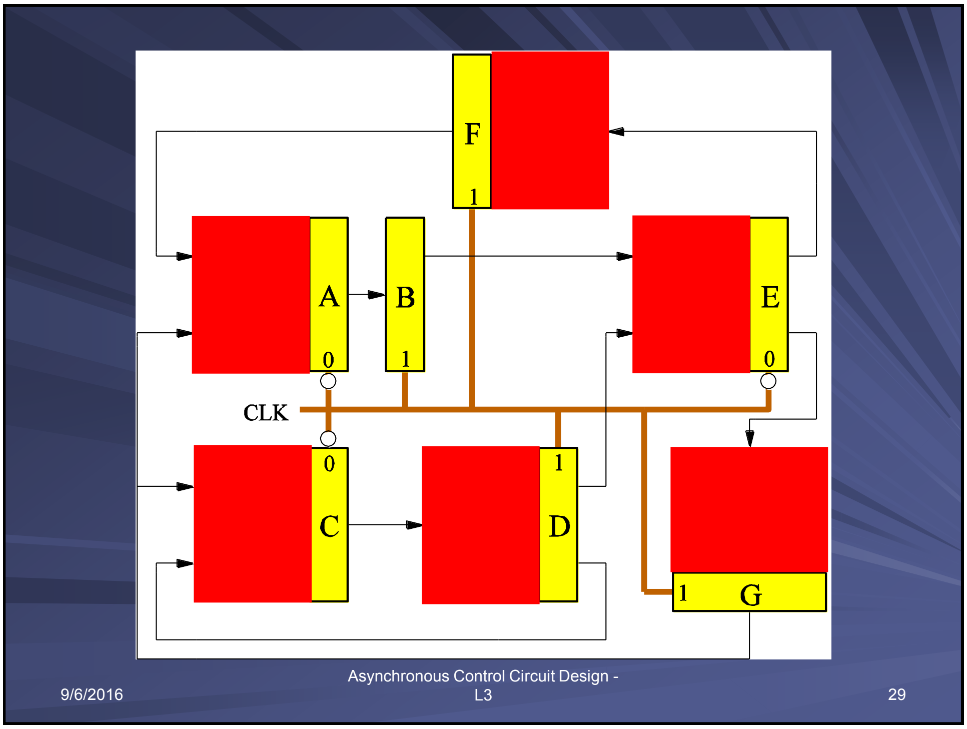
27

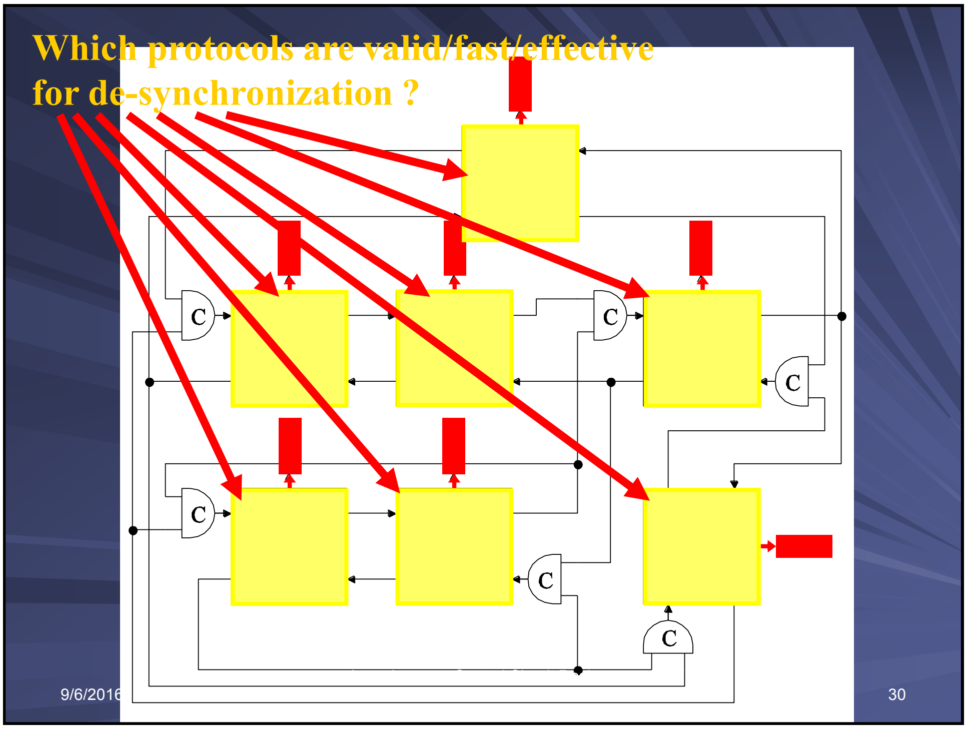


9/6/2016

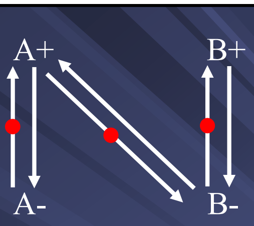
Asynchronous Control Circuit Design - L3

28





Which protocols are valid/fast/effective for de-synchronization ?



Theorem:

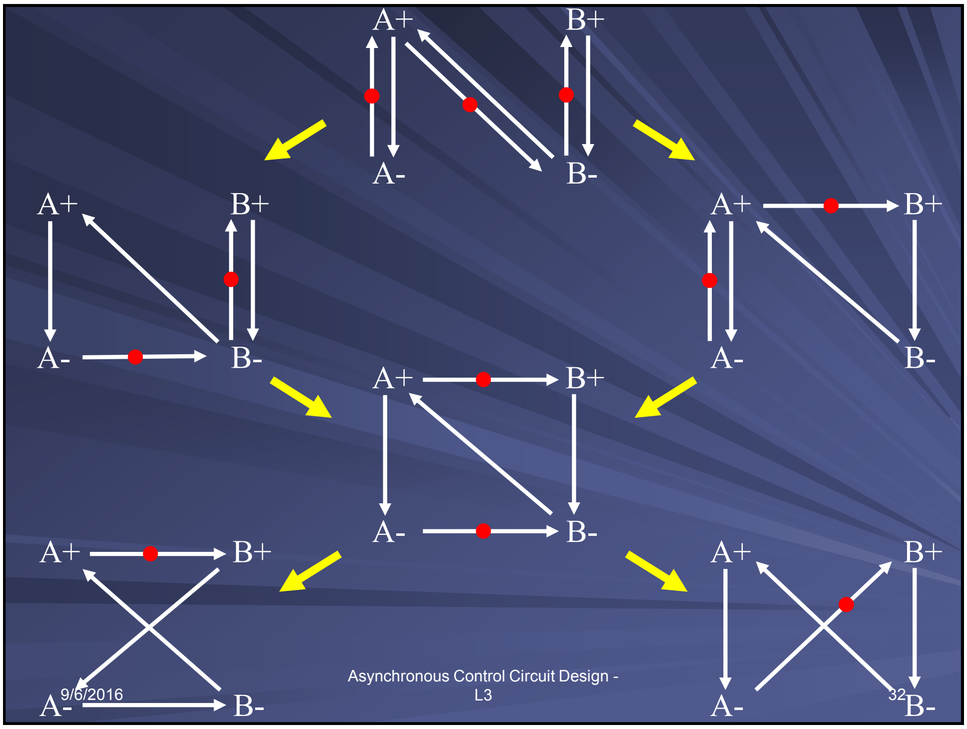
the de-synchronization protocol preserves flow-equivalence

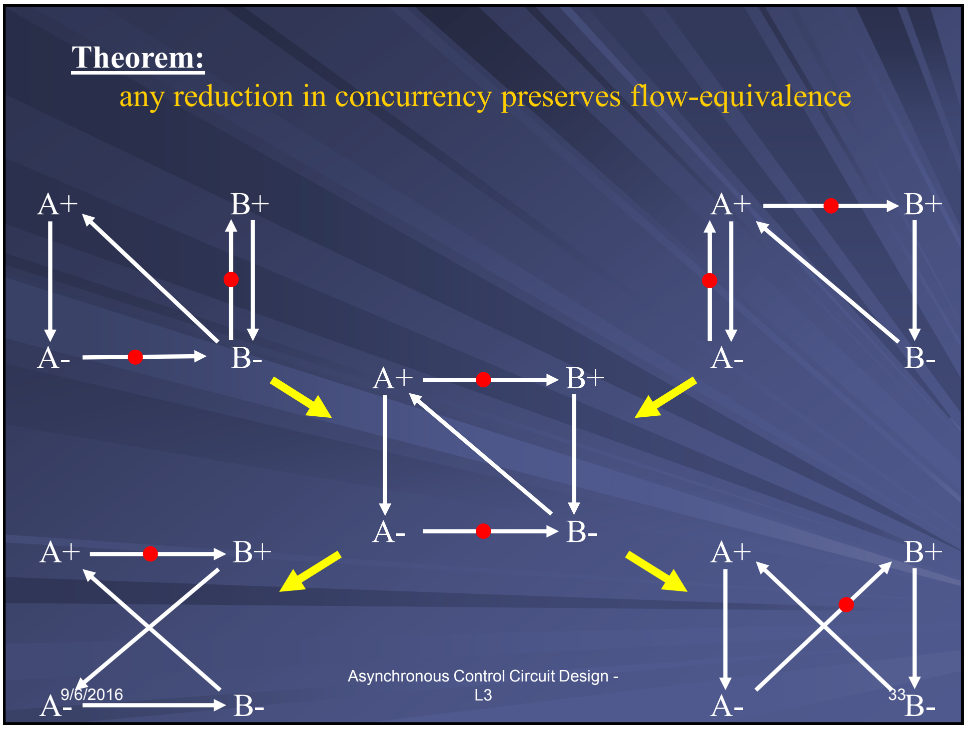
Proof: by induction on the length of the traces

Induction hypothesis: same latch values at reset

Induction step:

same values at cycle i ➔ same values at cycle i+1





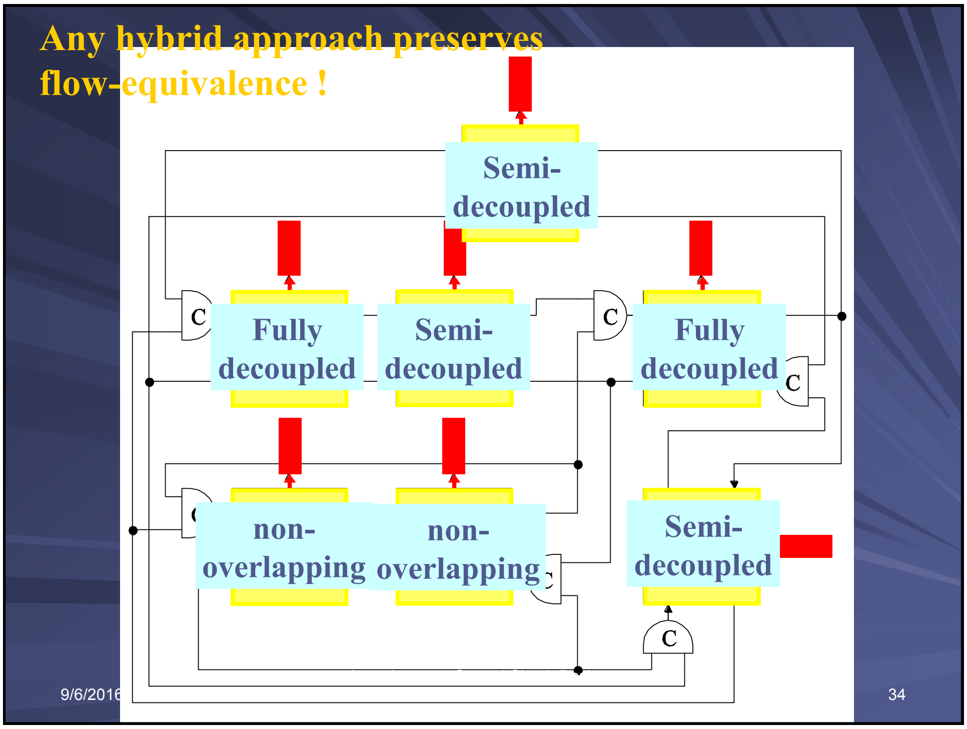
Theorem:

any reduction in concurrency preserves flow-equivalence

9/6/2016

Asynchronous Control Circuit Design - L3

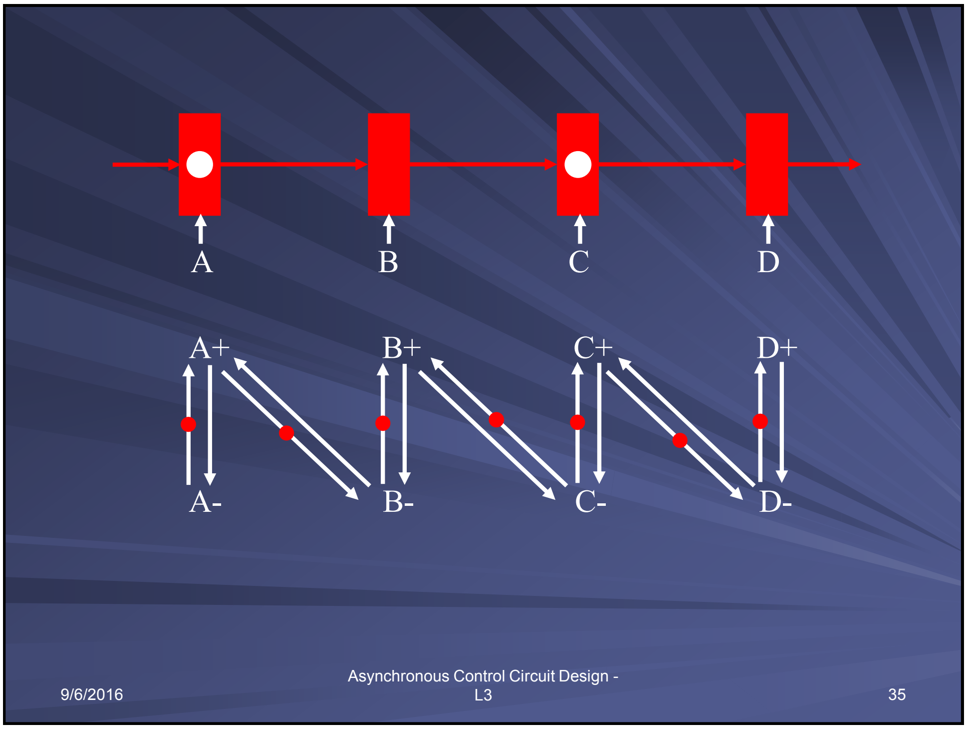
33

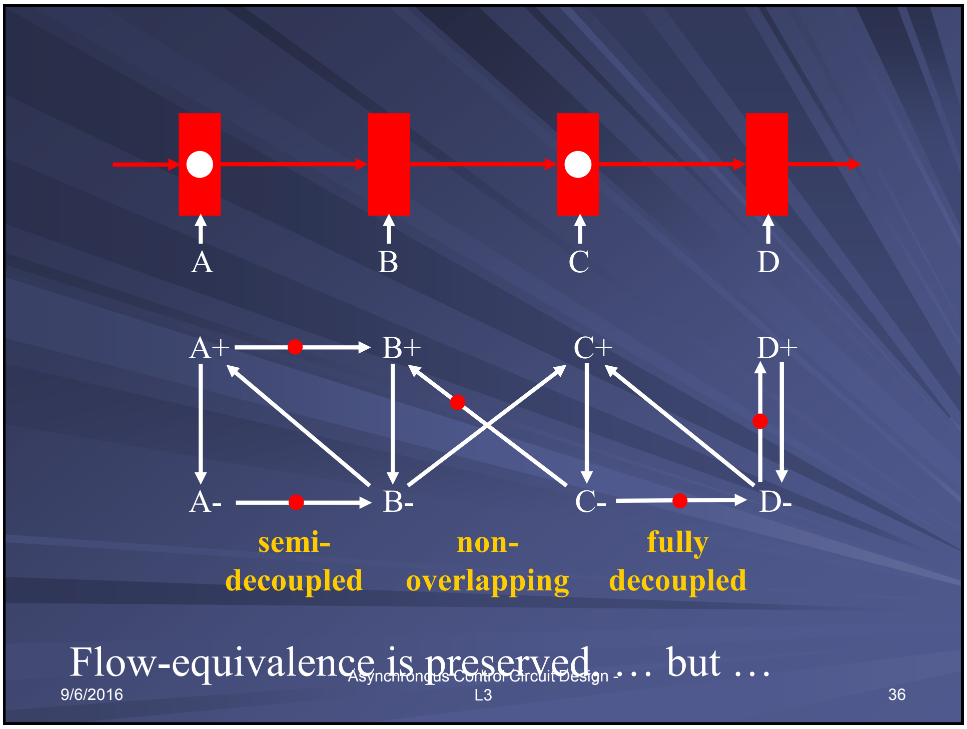


Any hybrid approach preserves flow-equivalence !

9/6/2016

34





semi-decoupled

non-overlapping

fully decoupled

Flow-equivalence is preserved … but …

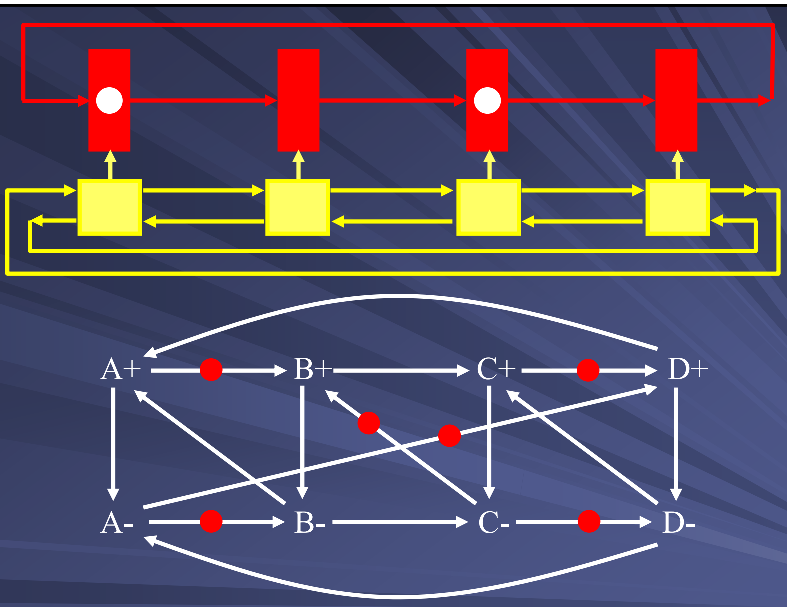
Live-ness?

- Preservation of flow-equivalence:

  all the generated traces are equivalent

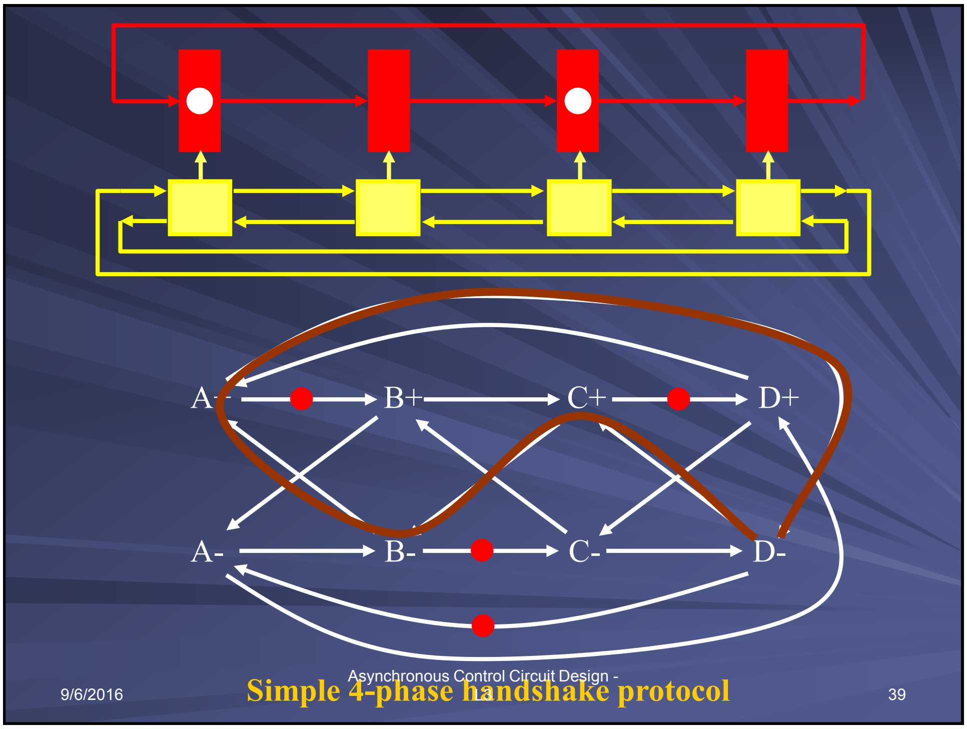
- Are all traces generated ?
  (Is the marked graph live ?)

  Not always !

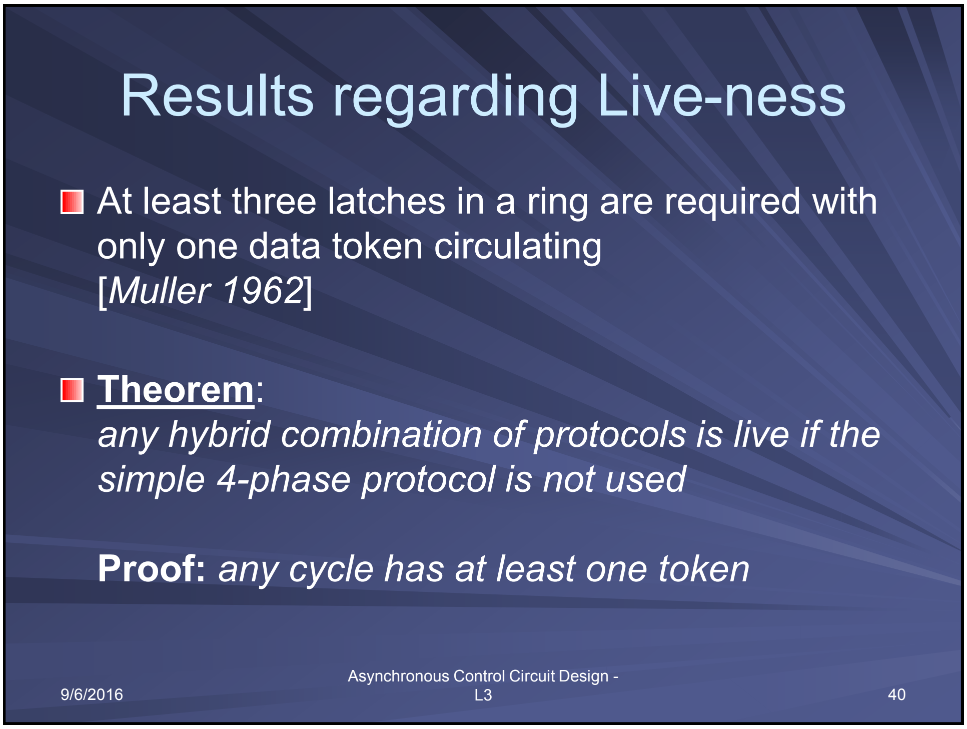


Semi-decoupled 4-phase handshake protocol

Liveness: all cycles have at least one token [Commoner 1971]



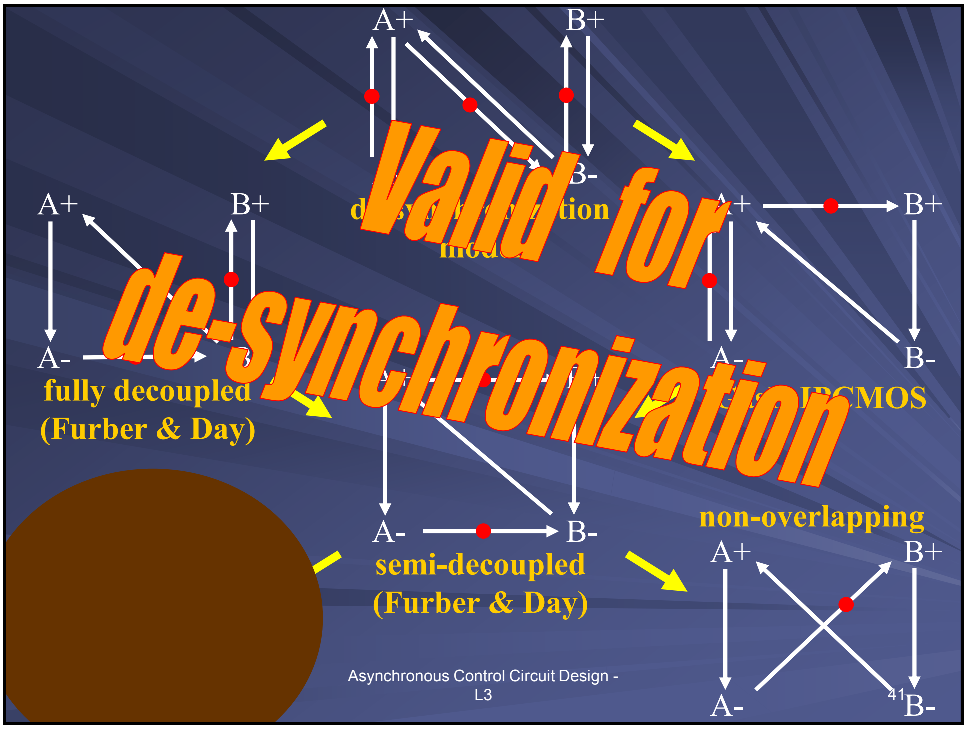
Simple 4-phase handshake protocol



Results regarding Live-ness

- At least three latches in a ring are required with only one data token circulating [*Muller 1962*]
- **Theorem**: *any hybrid combination of protocols is live if the simple 4-phase protocol is not used*

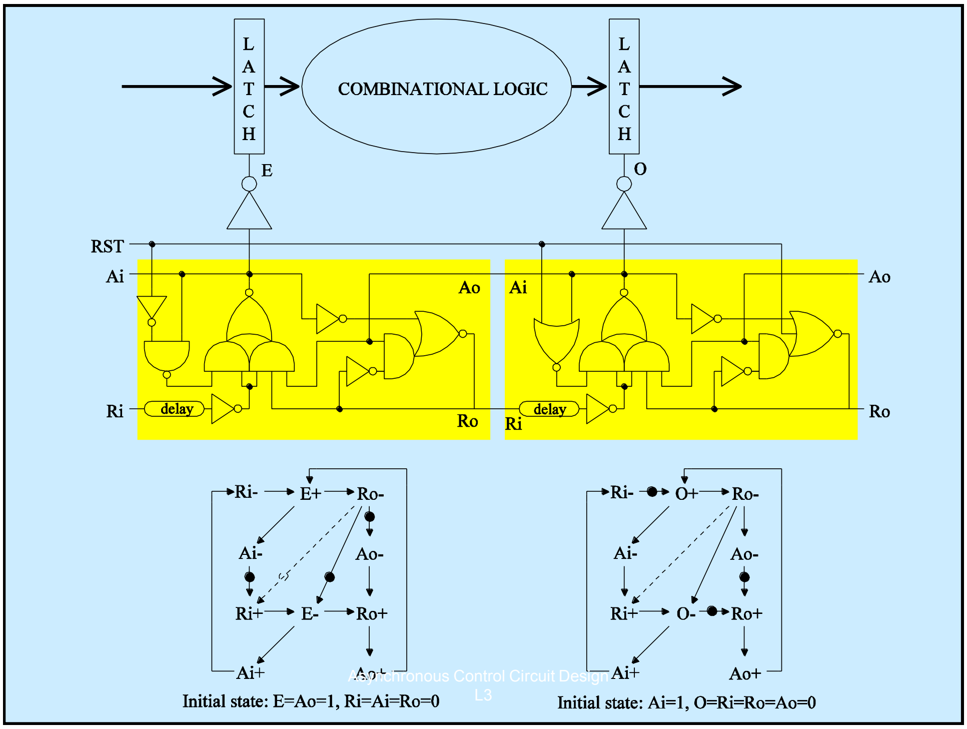
  Proof: *any cycle has at least one token*



Valid for de-synchronization

Asynchronous Control Circuit Design - L3

41

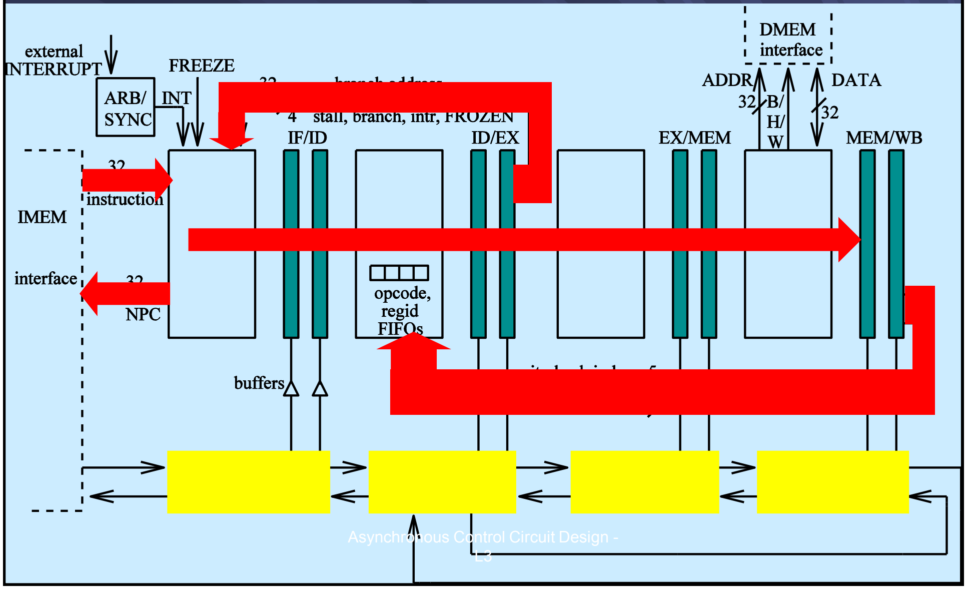


Initial state: E=Ao=1, Ri=Ai=Ro=0

Initial state: Ai=1, O=Ri=Ro=Ao=0

Asynchronous Control Circuit Design - L3

ASPIDA DLX block diagram



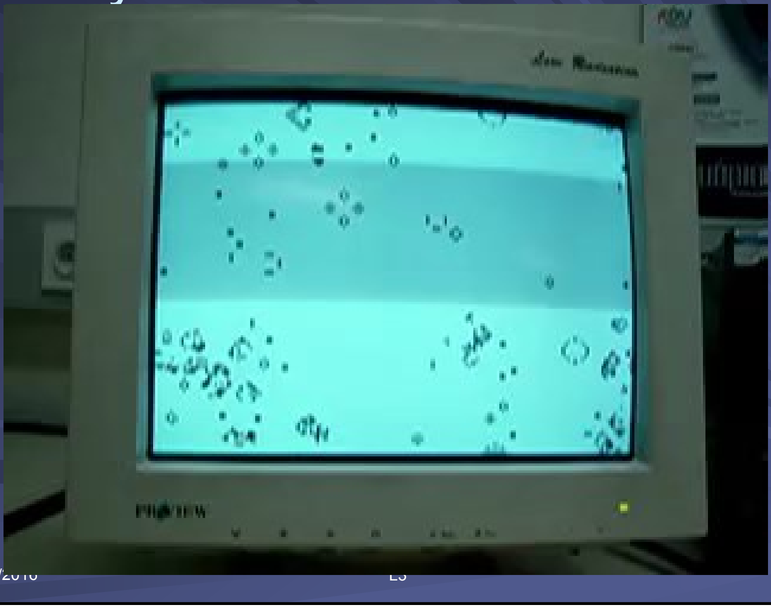
De-synchronization on FPGA

ASPIDA FPGA Implementation

- Xilinx Spartan IIE FPGA on a Diligent 2DE board
- FPGA contained:
  - De-synchronized DLX,
  - Processor memories
  - VGA driver
- Implemented Xilinx ISE
- Technology-portable Verilog design
- The full integer ISA and interrupt support is included
- DLX runs the "Game of Life" Algorithm
  - Fully-asynchronous
- VGA is fully synchronous

http://www.ics.forth.gr/carv/async/demo/

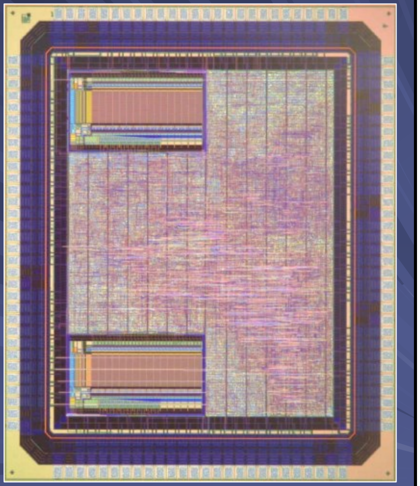
De-synchronized DLX on FPGA

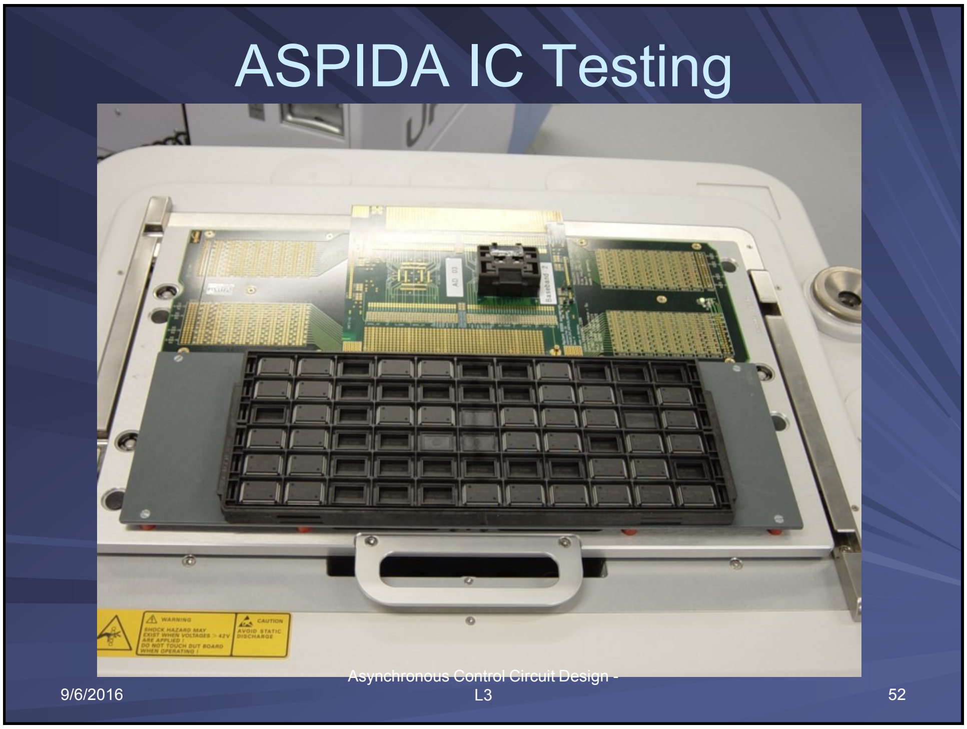


ASPIDA ASIC Design

ASPIDA IC

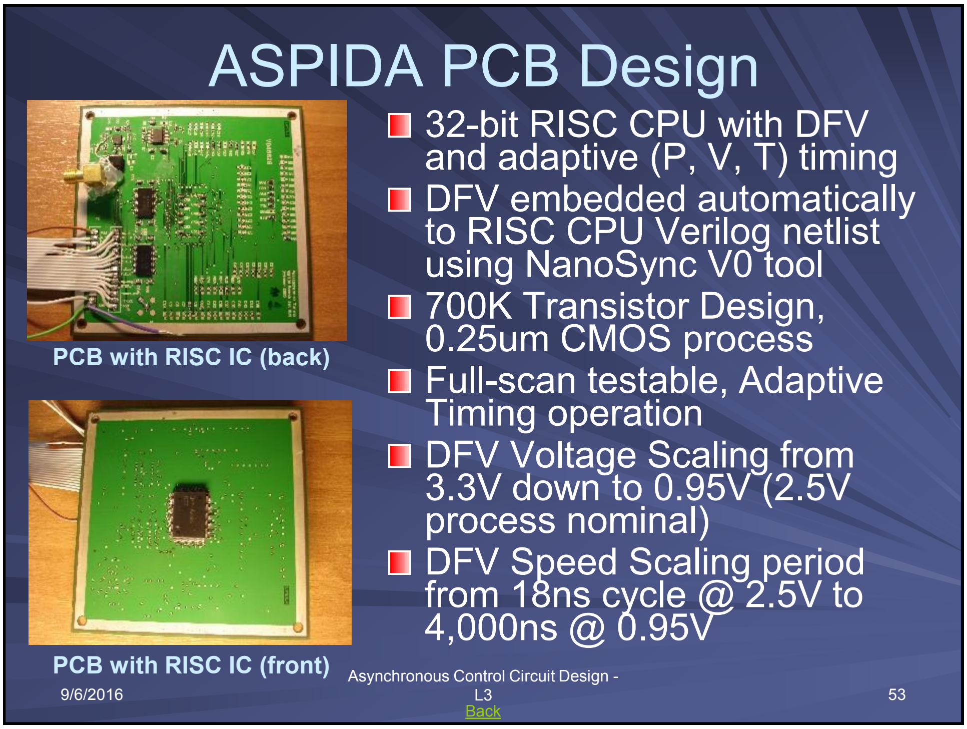
- 32-bit RISC CPU
- EU funded research project
- Two fully-functional ICs were manufactured with IHP 0.25um technology
- Runs in both *synchronous* and *de-synchronized* modes,
  - Direct comparison of results
- Measurements
  - Performance, Voltage scaling and EME measurements
  - On-tester functional tests
  - Lab analysis





ASPIDA IC Testing

9/6/2016 Asynchronous Control Circuit Design - L3 52



ASPIDA PCB Design

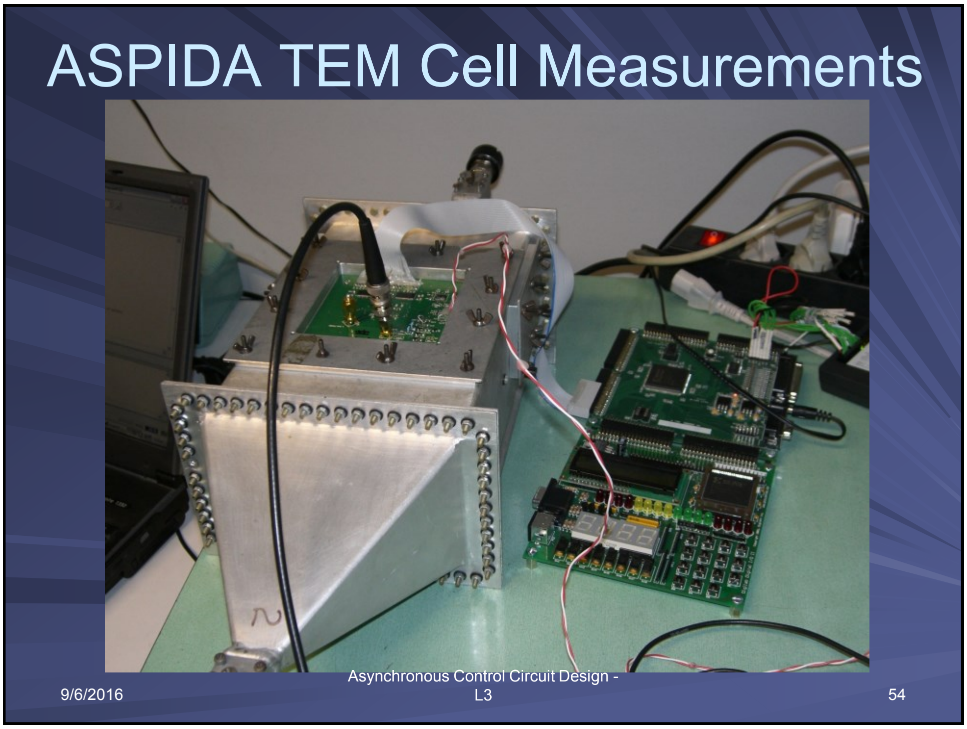
PCB with RISC IC (back)

PCB with RISC IC (front)

- 32-bit RISC CPU with DFV and adaptive (P, V, T) timing
- DFV embedded automatically to RISC CPU Verilog netlist using NanoSync V0 tool
- 700K Transistor Design, 0.25um CMOS process
- Full-scan testable, Adaptive Timing operation
- DFV Voltage Scaling from 3.3V down to 0.95V (2.5V process nominal)
- DFV Speed Scaling period from 18ns cycle @ 2.5V to 4,000ns @ 0.95V

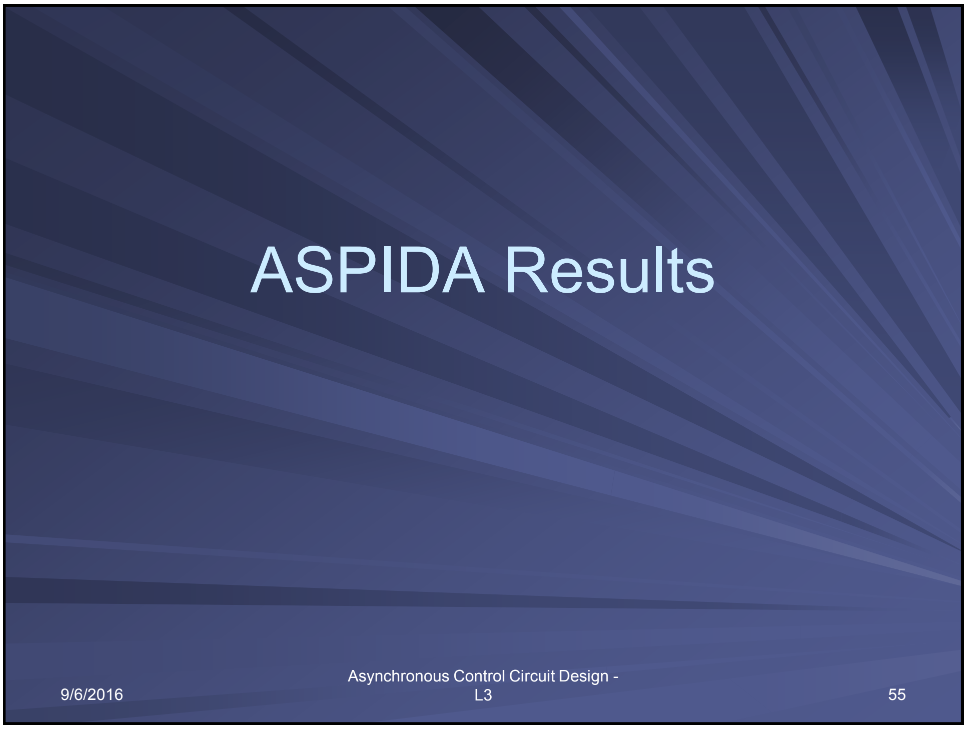
9/6/2016 Asynchronous Control Circuit Design - L3 53

Back



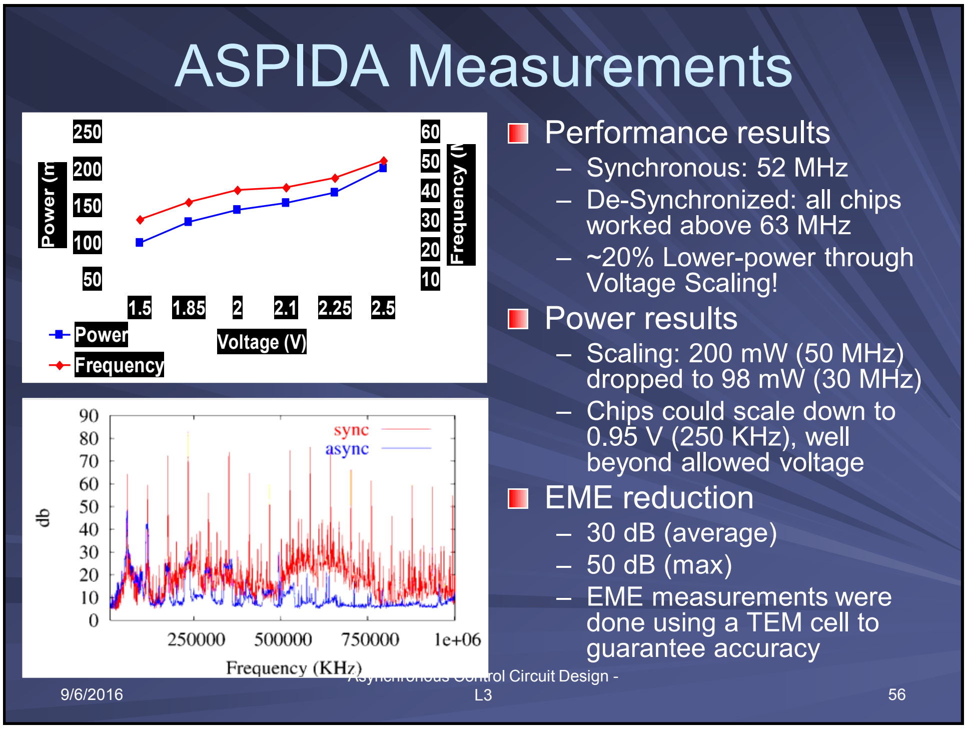
ASPIDA TEM Cell Measurements

9/6/2016 Asynchronous Control Circuit Design - L3 54



ASPIDA Results

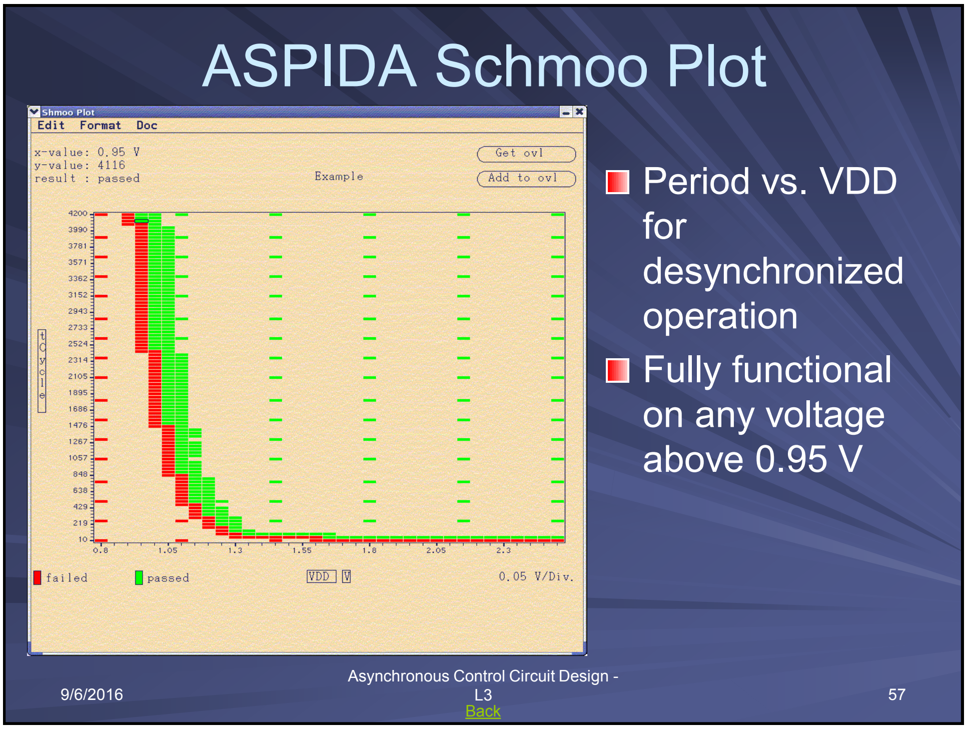
9/6/2016 Asynchronous Control Circuit Design - L3 55



ASPIDA Measurements

- Performance results
  - Synchronous: 52 MHz
  - De-Synchronized: all chips worked above 63 MHz
  - ~20% Lower-power through Voltage Scaling!
- Power results
  - Scaling: 200 mW (50 MHz) dropped to 98 mW (30 MHz)
  - Chips could scale down to 0.95 V (250 KHz), well beyond allowed voltage
- EME reduction
  - 30 dB (average)
  - 50 dB (max)
  - EME measurements were done using a TEM cell to guarantee accuracy

9/6/2016 Asynchronous Control Circuit Design - L3 56



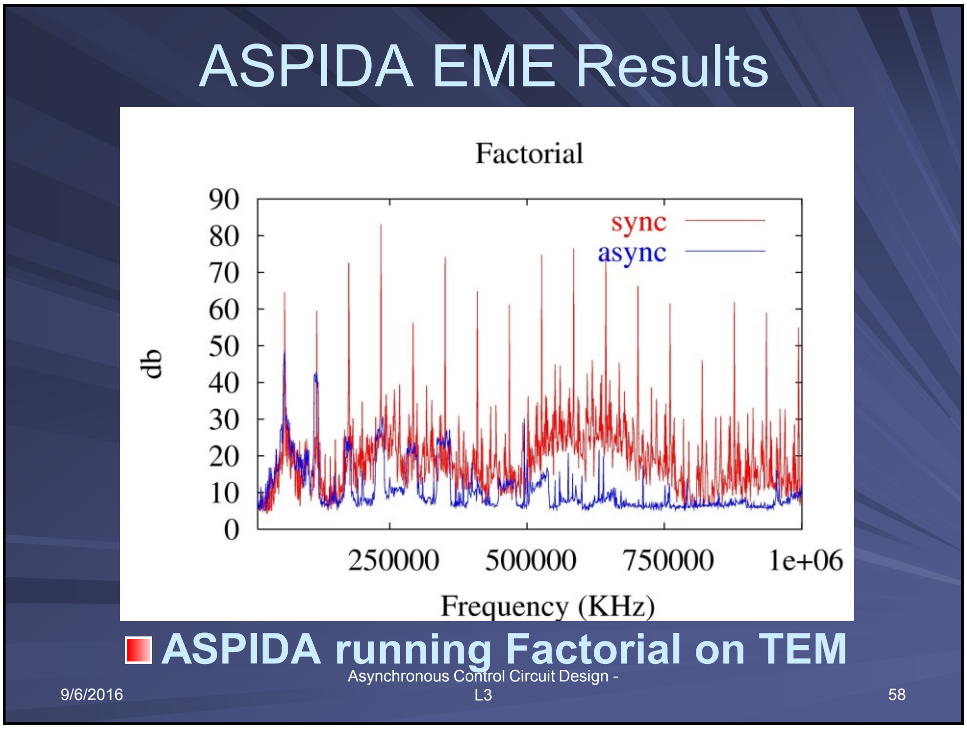
ASPIDA Schmoo Plot

```
x-value: 0.95 V
y-value: 4116
result : passed
```

- Period vs. VDD for desynchronized operation
- Fully functional on any voltage above 0.95 V

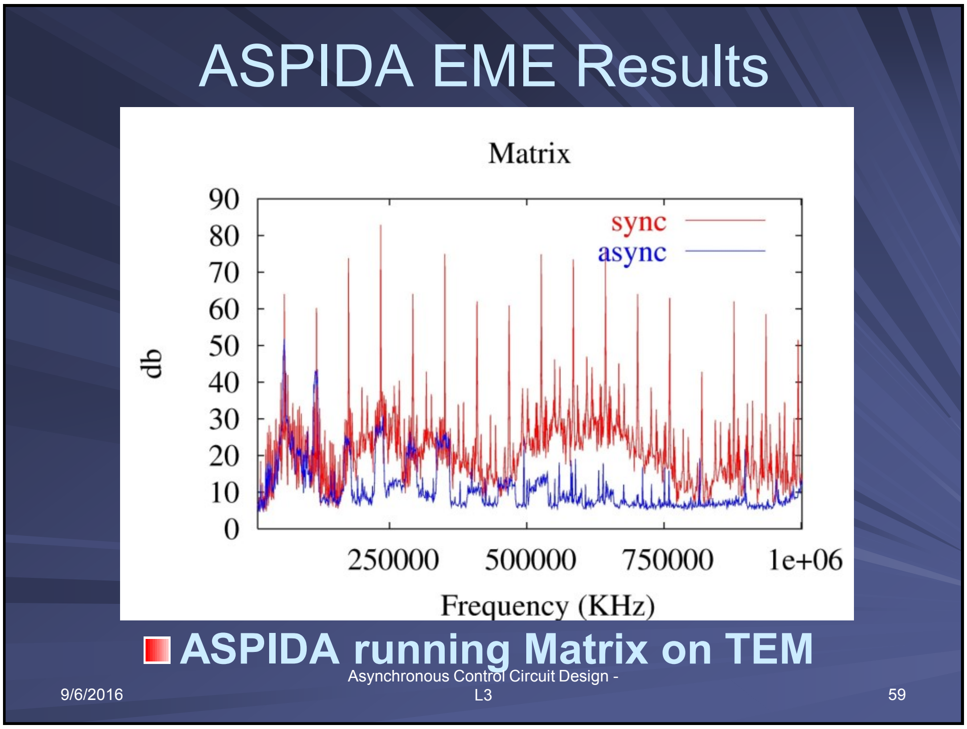
9/6/2016 Asynchronous Control Circuit Design - L3 57

Back



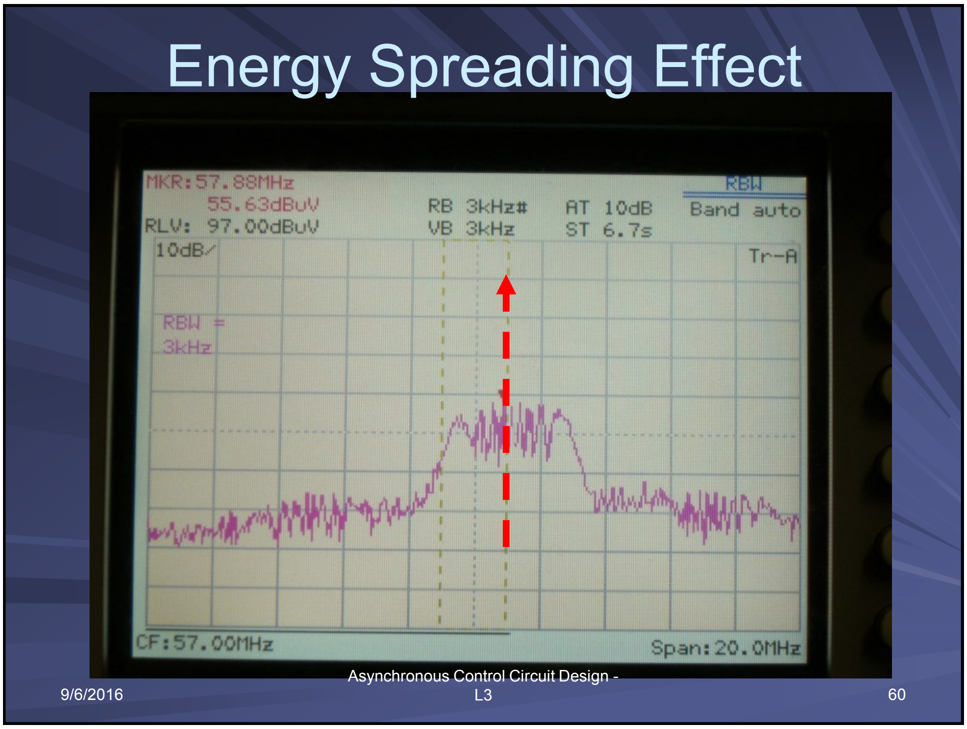
ASPIDA EME Results

- ASPIDA running Factorial on TEM



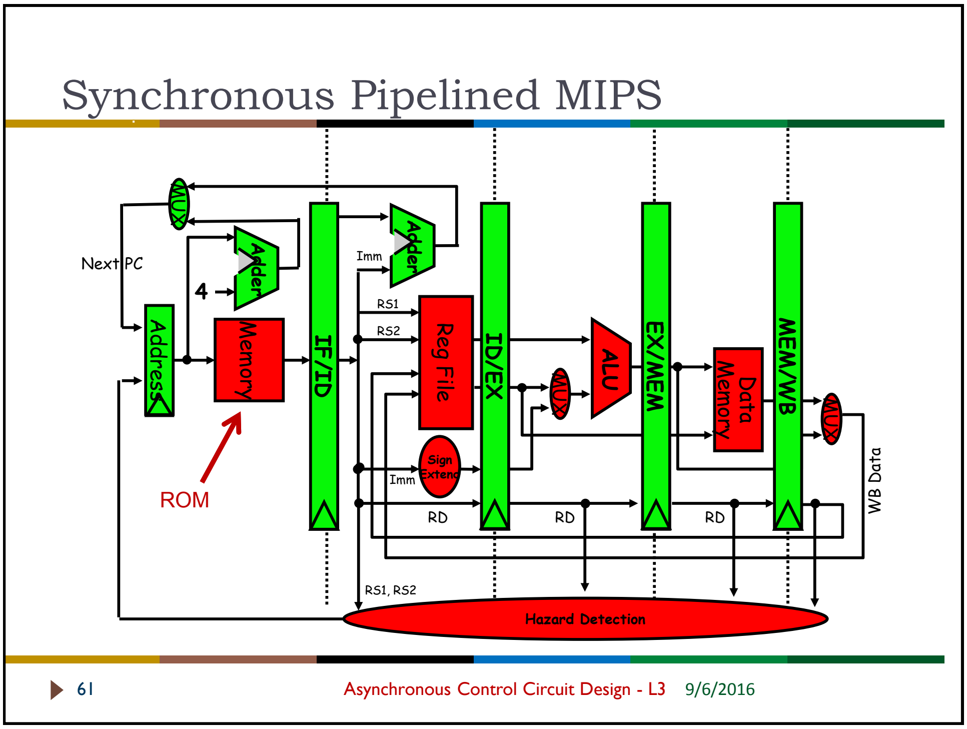
ASPIDA EME Results

- ASPIDA running Matrix on TEM



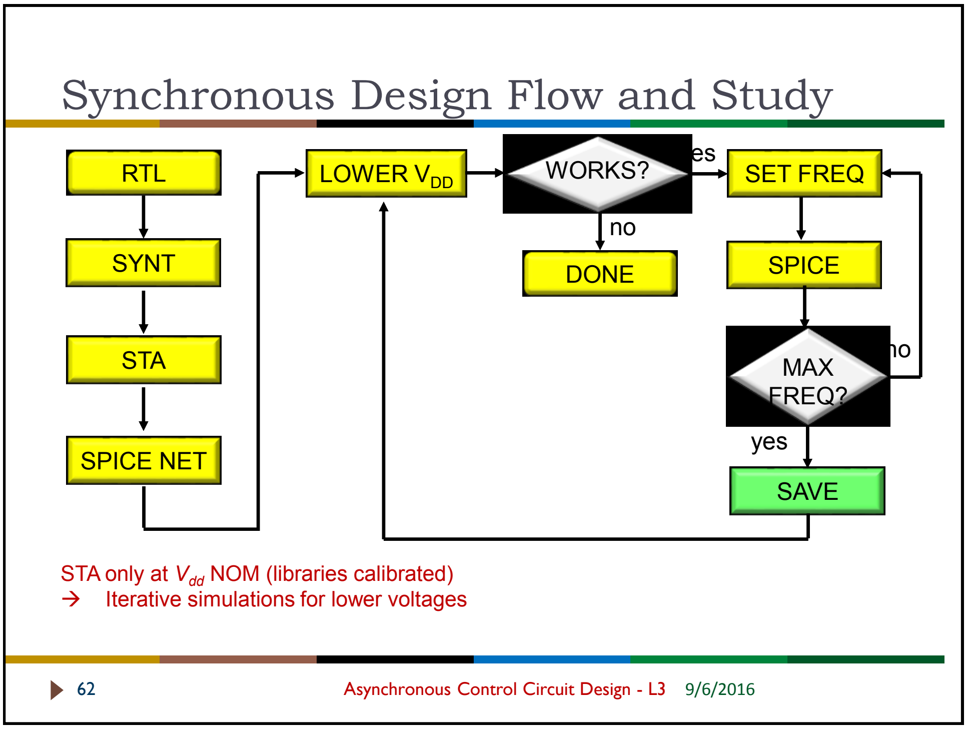
Energy Spreading Effect

9/6/2016 Asynchronous Control Circuit Design - L3 60



Synchronous Pipelined MIPS

61 Asynchronous Control Circuit Design - L3 9/6/2016



Synchronous Design Flow and Study

RTL

SYNT

STA

SPICE NET

LOWER V_{DD}

WORKS?

es

no

DONE

SET FREQ

SPICE

MAX FREQ?

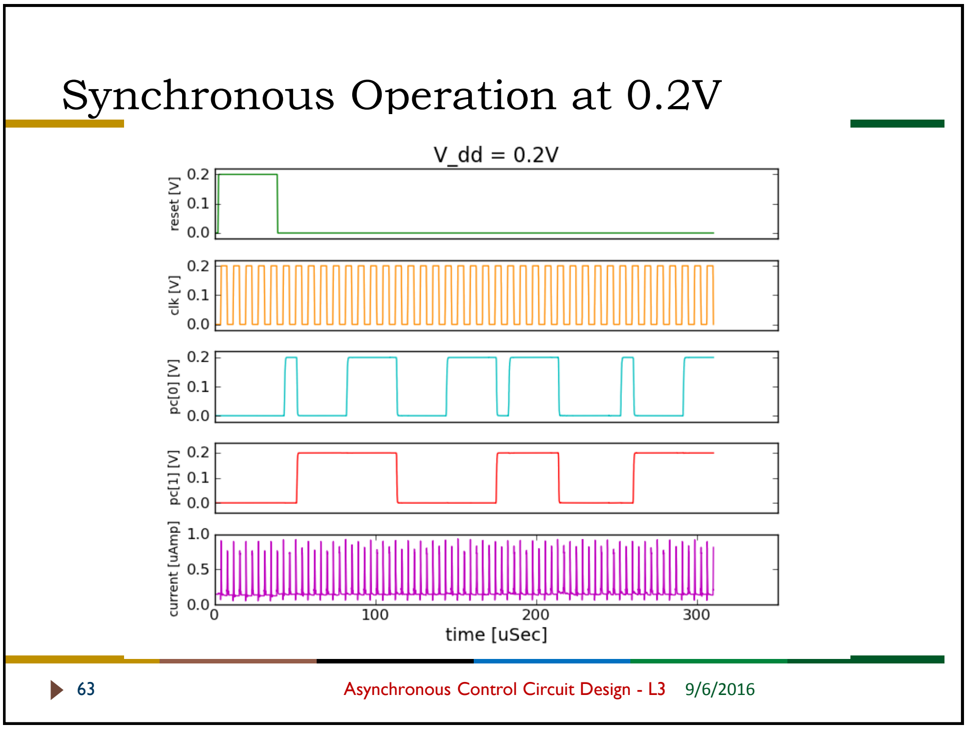
no

yes

SAVE

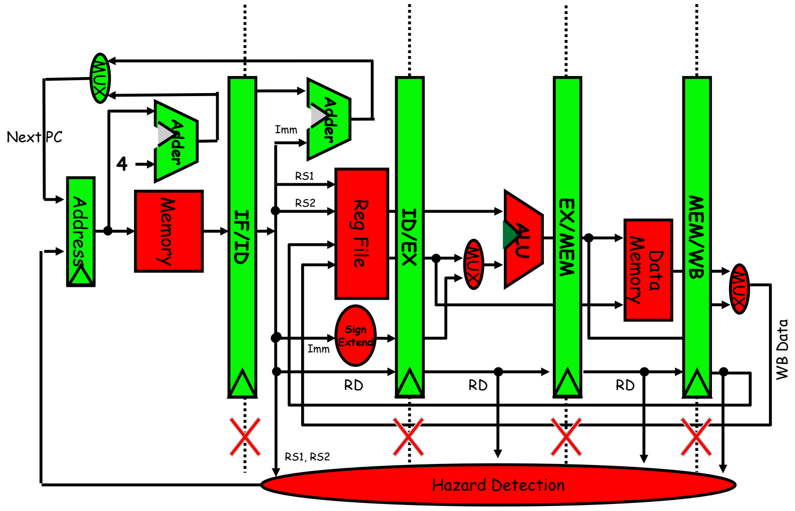
STA only at V_{dd} NOM (libraries calibrated)

→ Iterative simulations for lower voltages

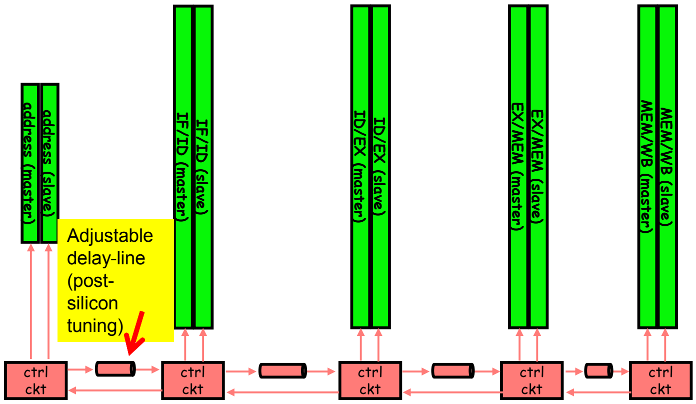


Synchronous Operation at 0.2V

De-Synchronizing the MIPS



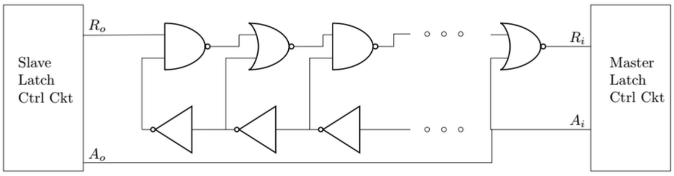
De-Synchronized Bundled-Data MIPS for sub-VT (uaMIPS)



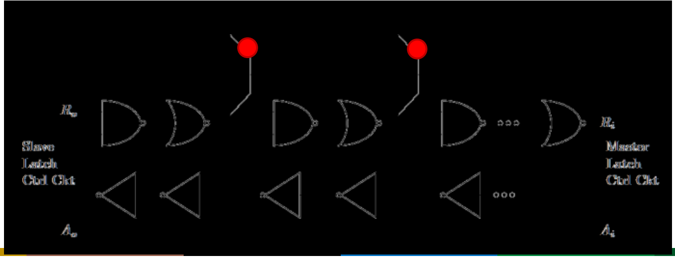
Adjustable delay-line (post-silicon tuning)

Post Silicon Tunable Asymmetric Delay Line

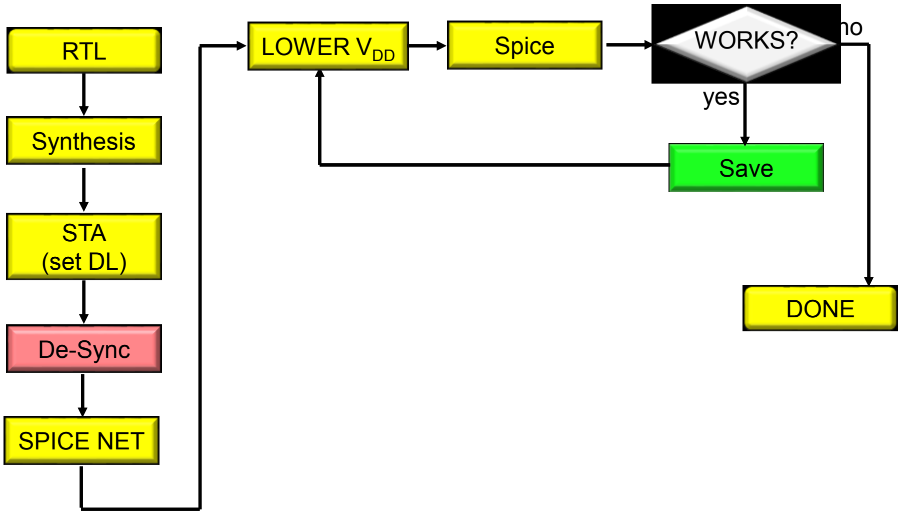
- Asymmetric Delay Lines, for fast Return to Zero



- Post-Si delay line tuning: Mitigate process variation



Asynchronous Design Flow and Study



RTL

Synthesis

STA
(set DL)

De-Sync

SPICE NET

LOWER V_{DD}

Spice

WORKS?

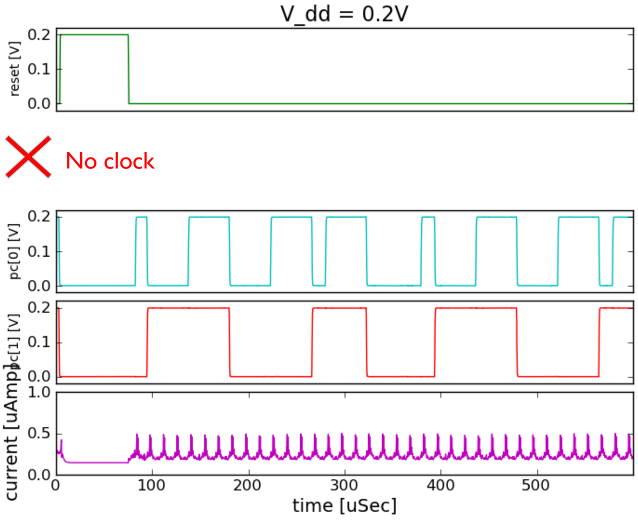
no

yes

Save

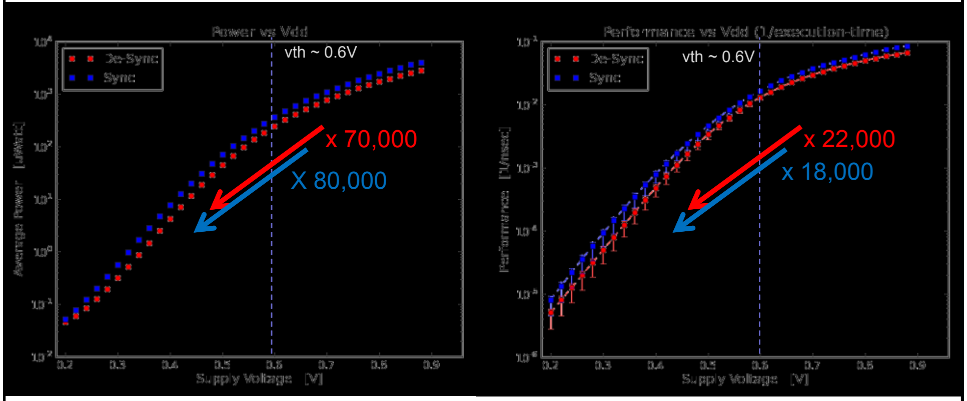
DONE

Asynchronous Operation at 0.2V



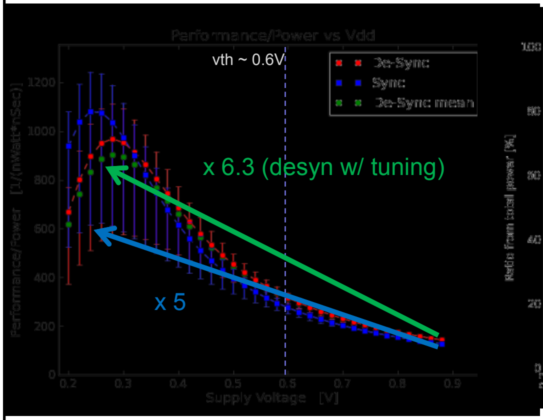
No clock

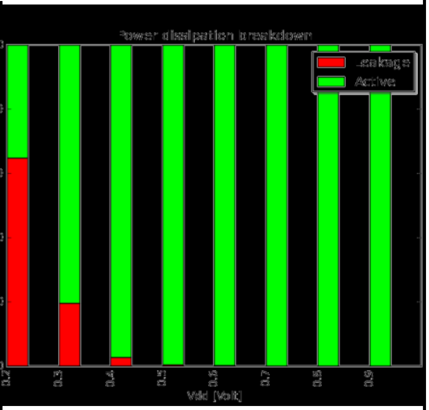
Asynchronous vs. Synchronous
uMIPS vs. uaMIPS Performance



Diamant, R, Ginosar, R, Sotiriou, C, "Asynchronous Sub-Threshold Ultra-Low Power Processor", Proceedings of PATMOS 2015, September 2015.

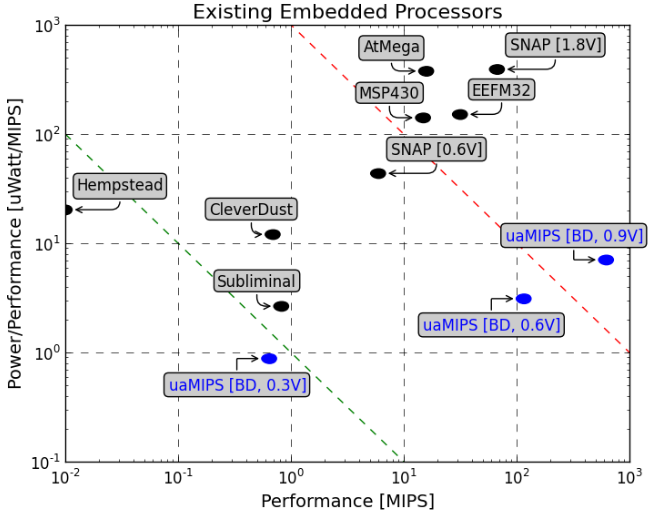
uaMIPS Results

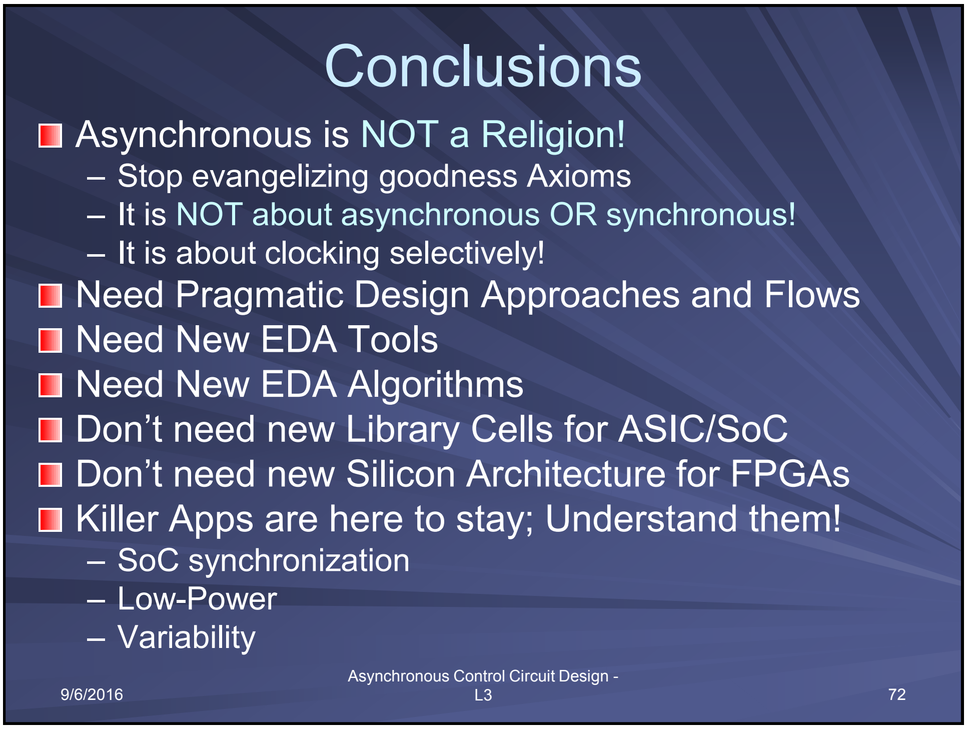




- Post-silicon tuning of delay lines:
  - No need to accommodate for process variations

ULP Embedded Processor Landscape





Conclusions

- Asynchronous is NOT a Religion!
  - Stop evangelizing goodness Axioms
  - It is NOT about asynchronous OR synchronous!
  - It is about clocking selectively!
- Need Pragmatic Design Approaches and Flows
- Need New EDA Tools
- Need New EDA Algorithms
- Don’t need new Library Cells for ASIC/SoC
- Don’t need new Silicon Architecture for FPGAs
- Killer Apps are here to stay; Understand them!
  - SoC synchronization
  - Low-Power
  - Variability