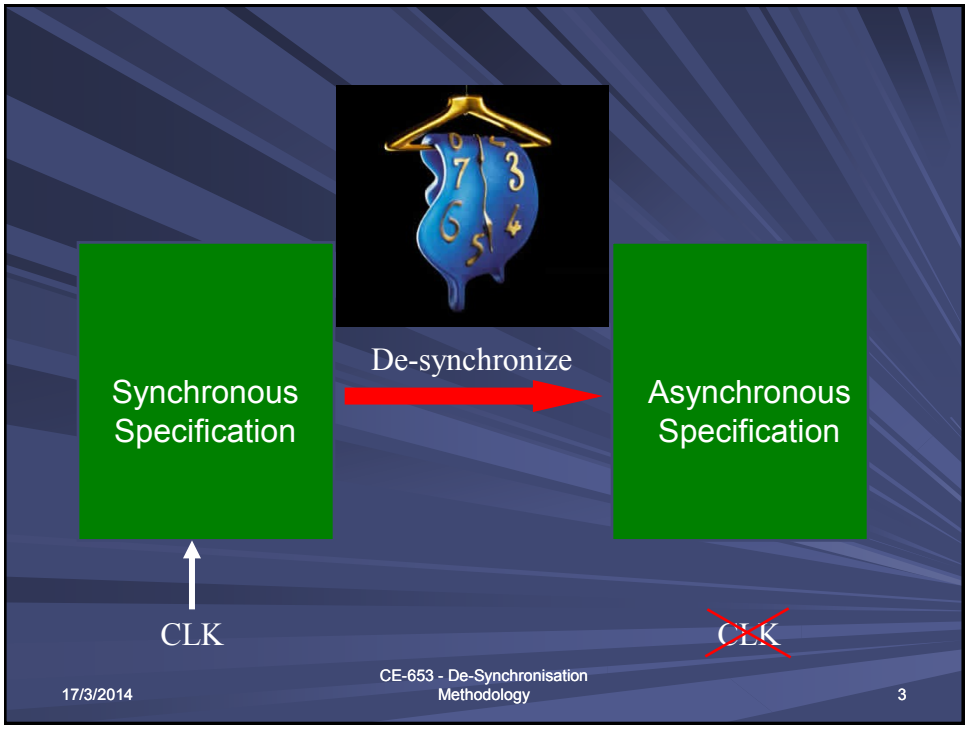
CE653 – Asynchronous Circuit Design

Instructor: C. Sotiriou

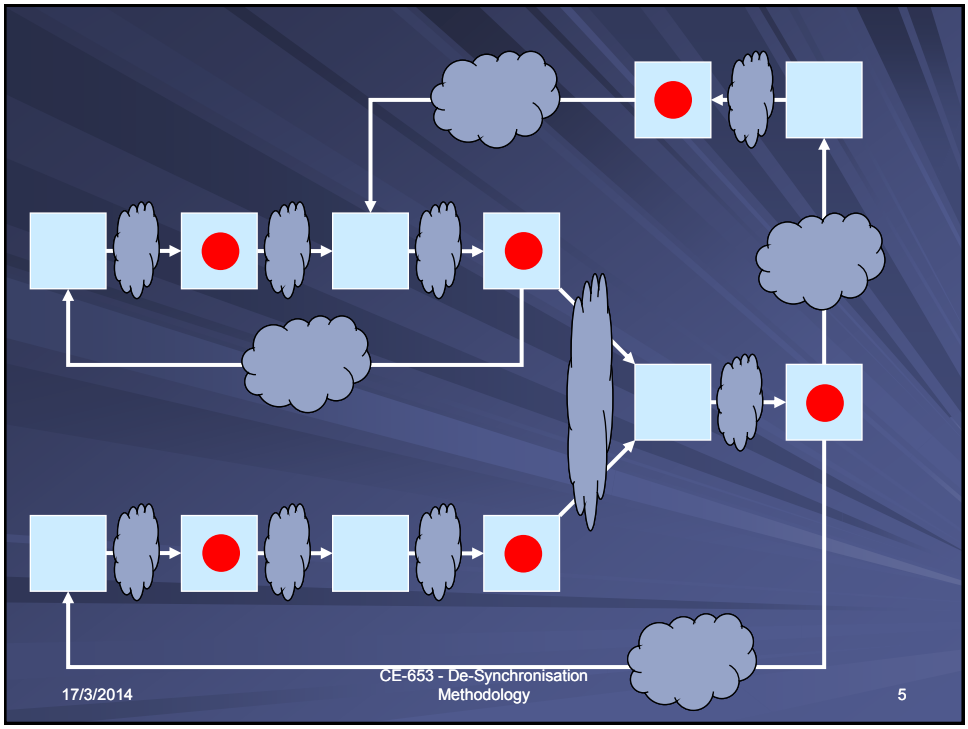
http://inf-server.inf.uth.gr/courses/CE653/

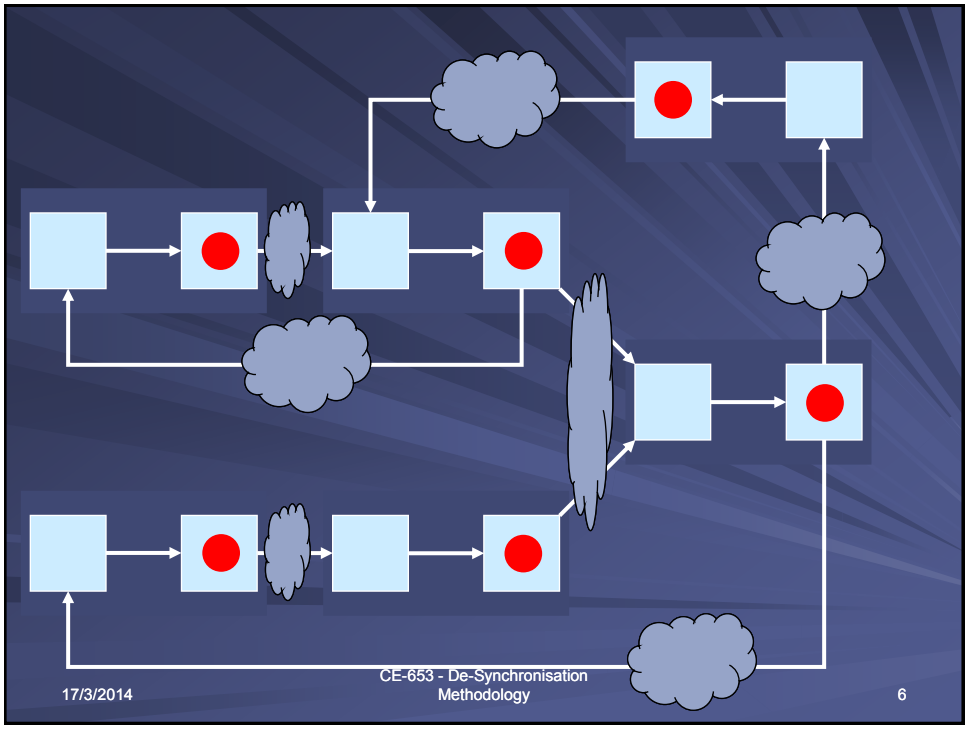
De-synchronization Theory and Fundamentals



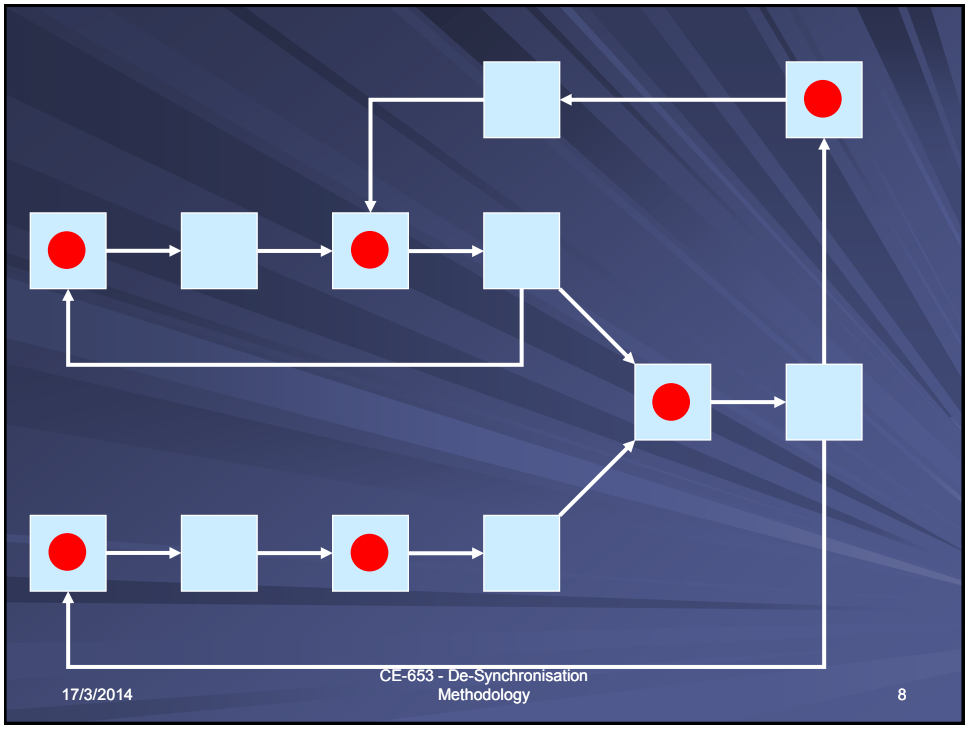
Prior work

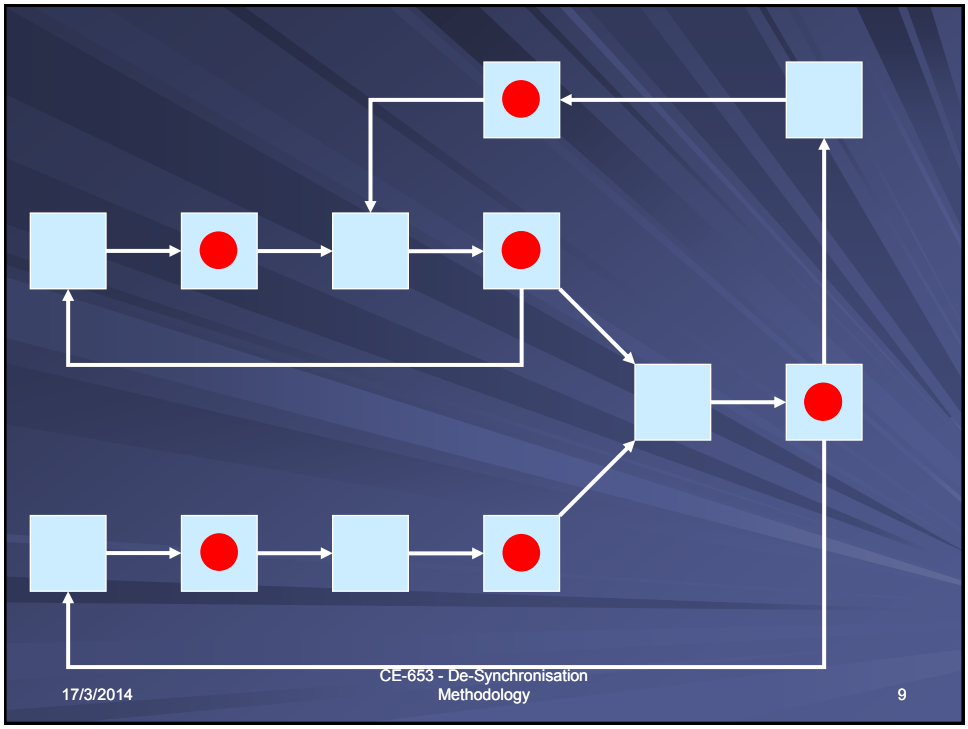
- Micropipelines (Sutherland, 1989)
- Local generation of clocks
  - Varshavsky et al., 1995
  - Kol and Ginosar, 1996
- Theseus Logic (Ligthart et al., 2000)
  - Commercial HDL synthesis tools
  - Direct translation and special registers
- Phased logic (Linder and Harden, 1996)
  (Reese, Thornton, Traver,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

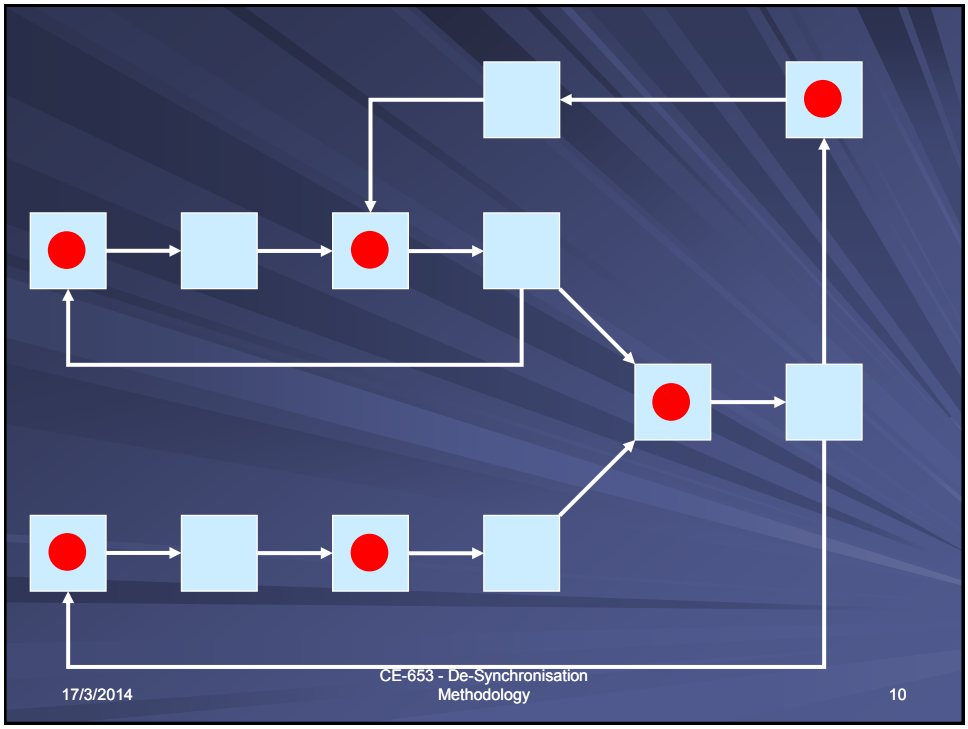

17/3/2014
CE-653 - De-Synchronisation Methodology
5

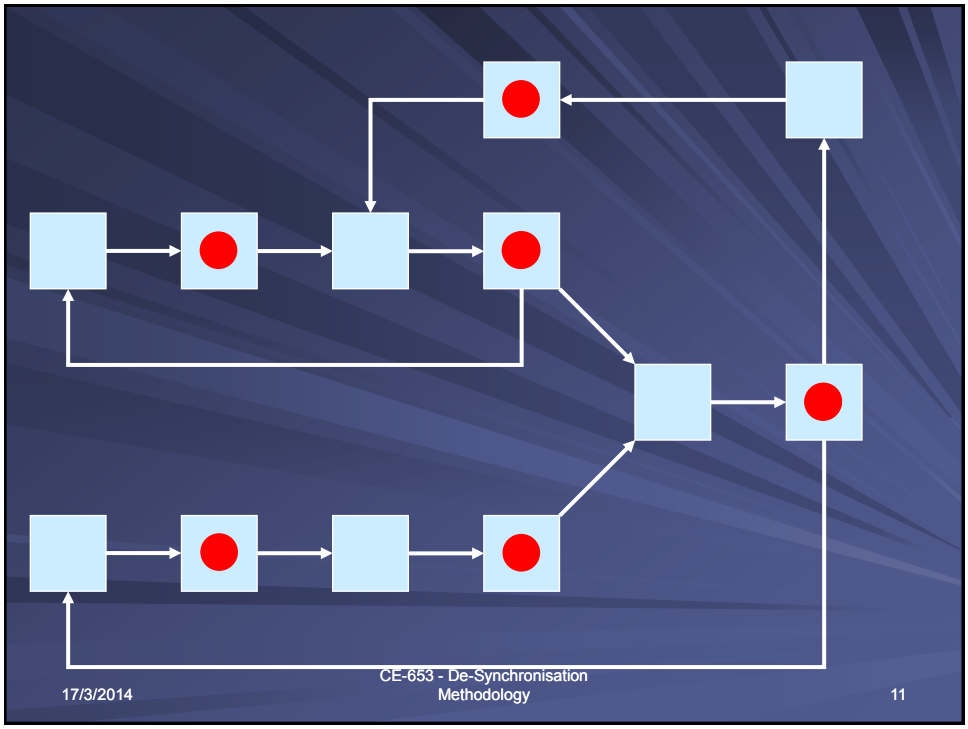

17/3/2014
CE-653 - De-Synchronisation Methodology
6

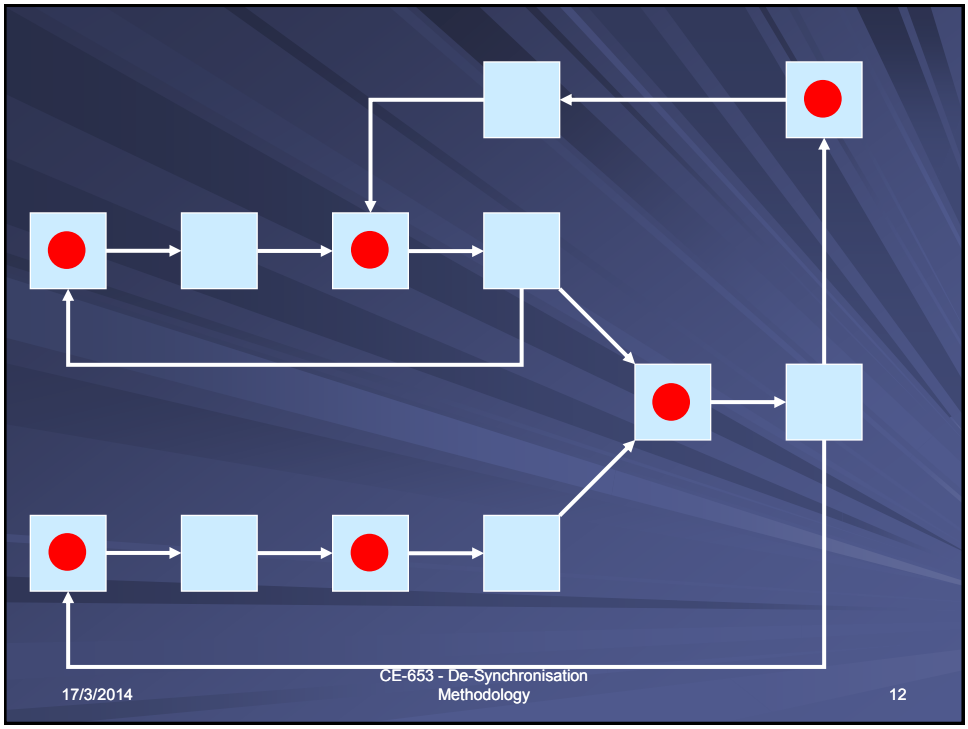
Synchronous flow

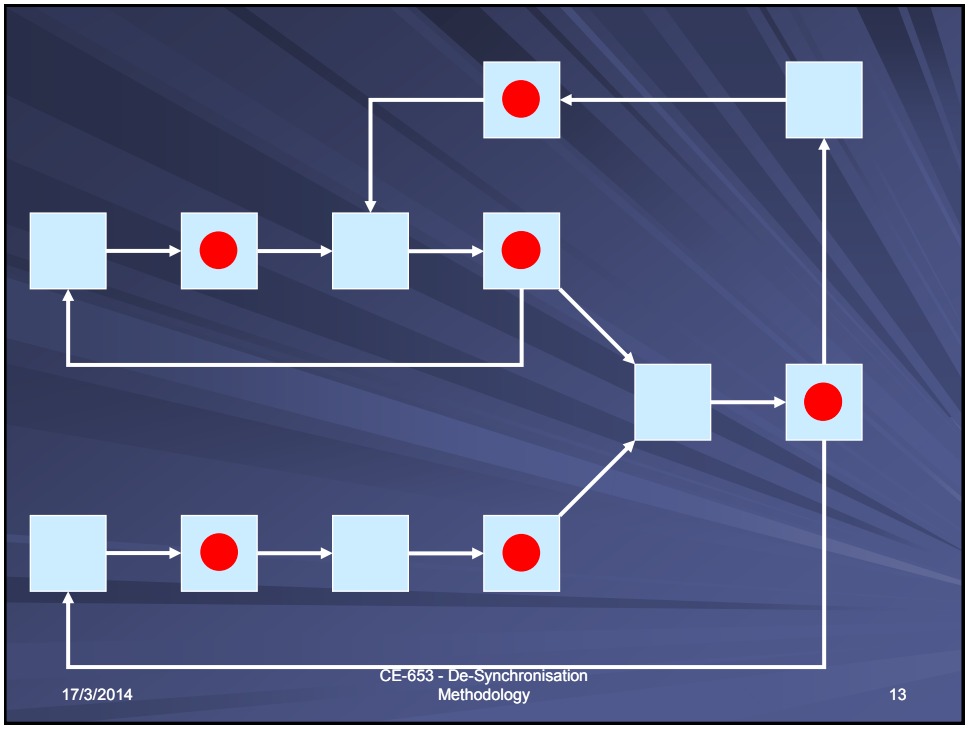


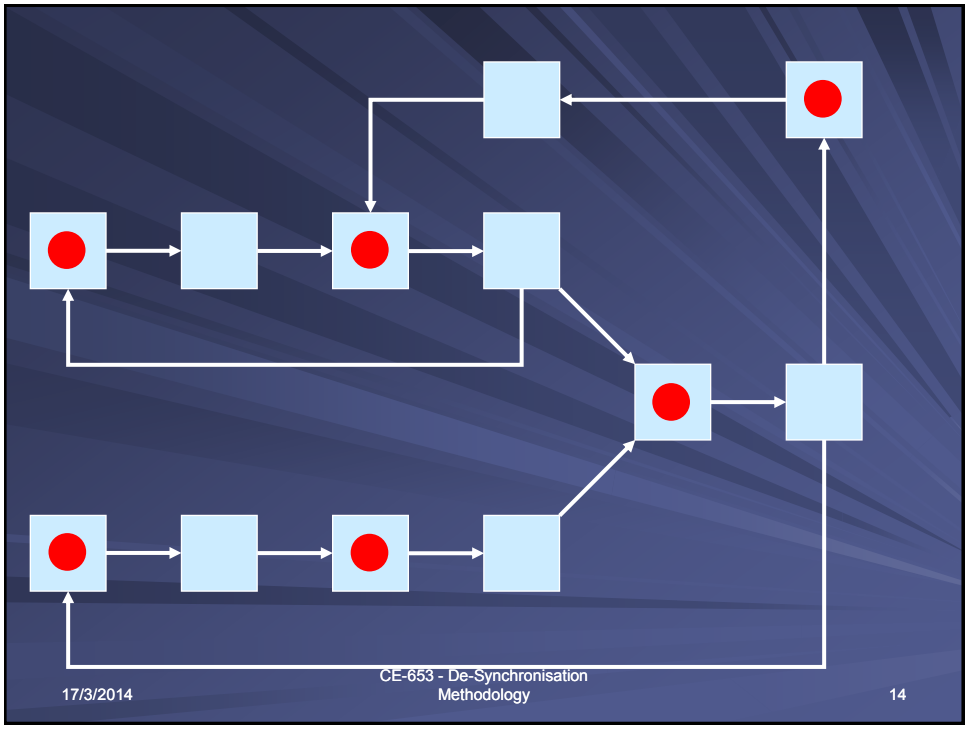

17/3/2014
CE-653 - De-Synchronisation Methodology
9

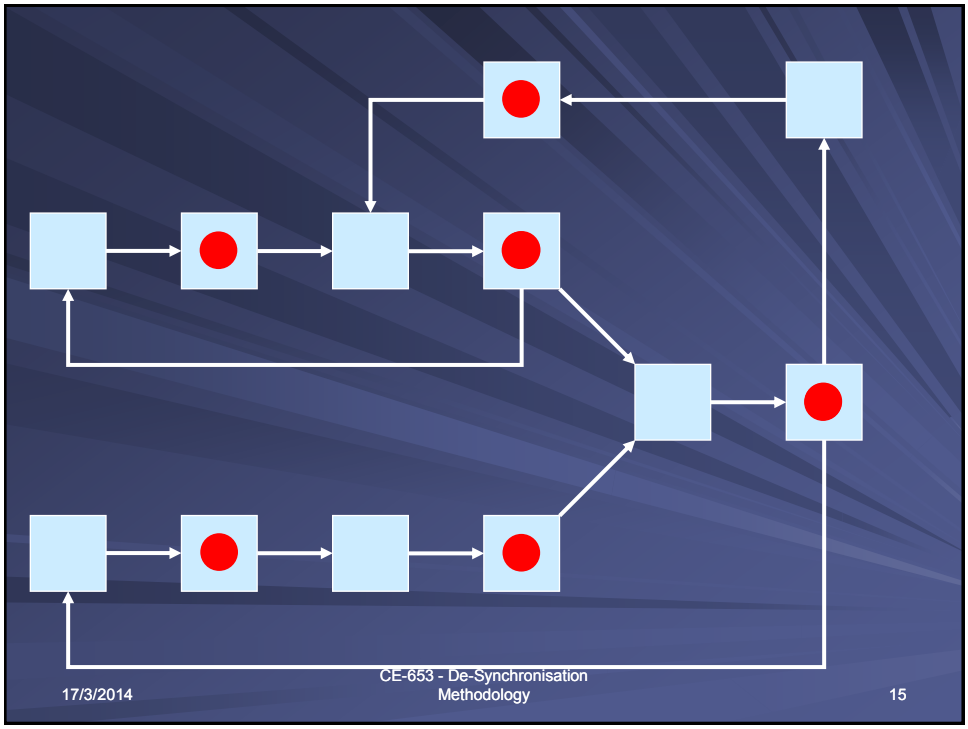

17/3/2014
CE-653 - De-Synchronisation Methodology
10

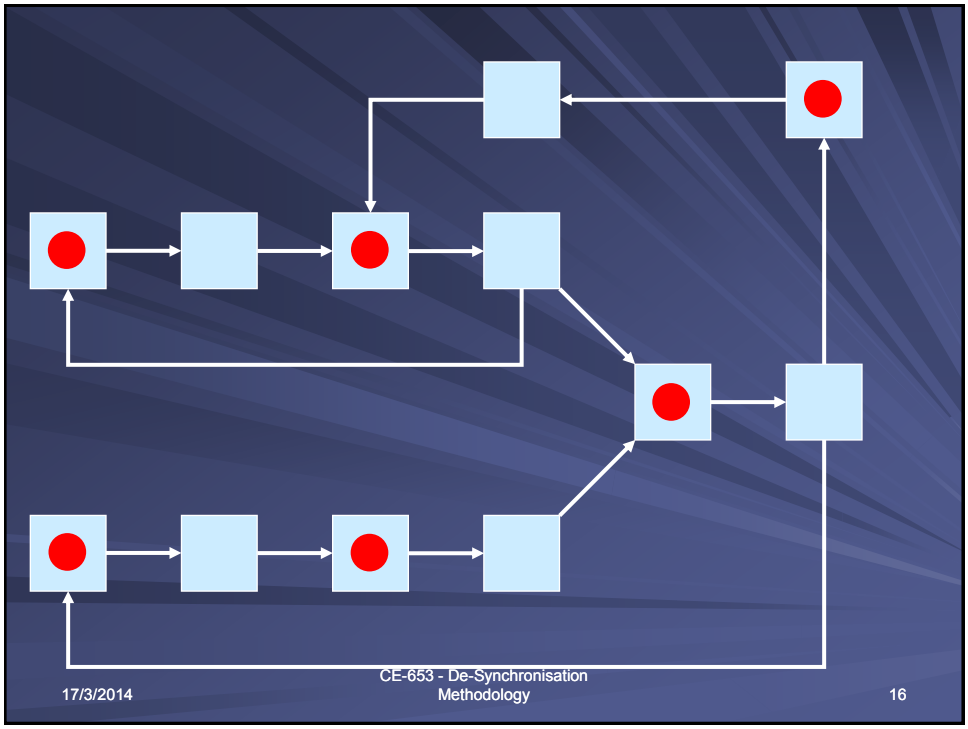


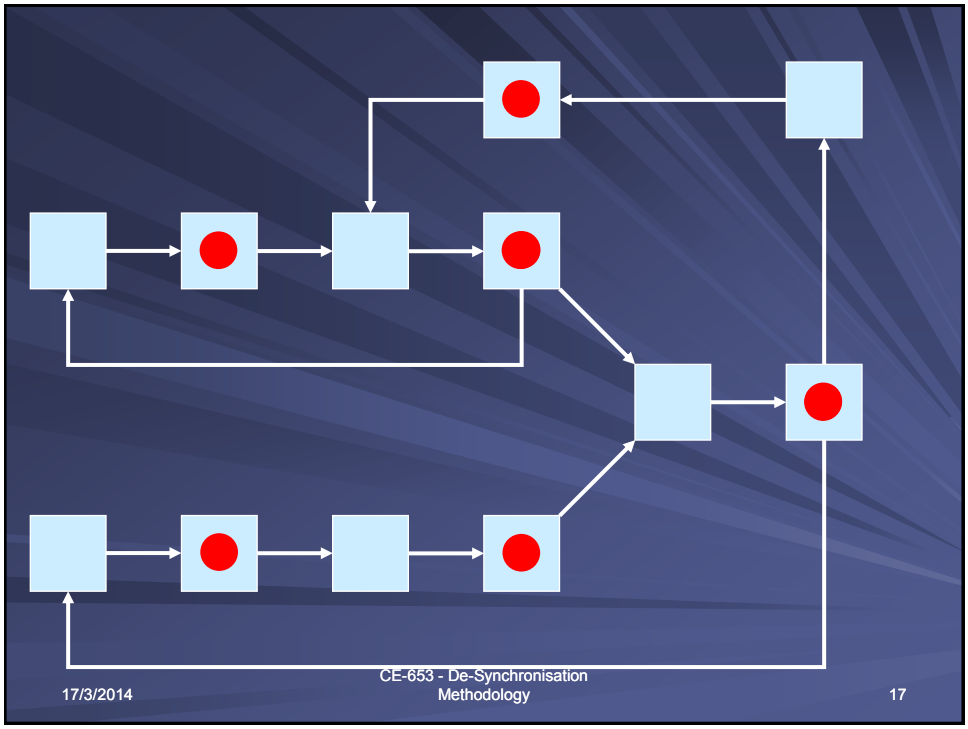



17/3/2014
CE-653 - De-Synchronisation Methodology
13

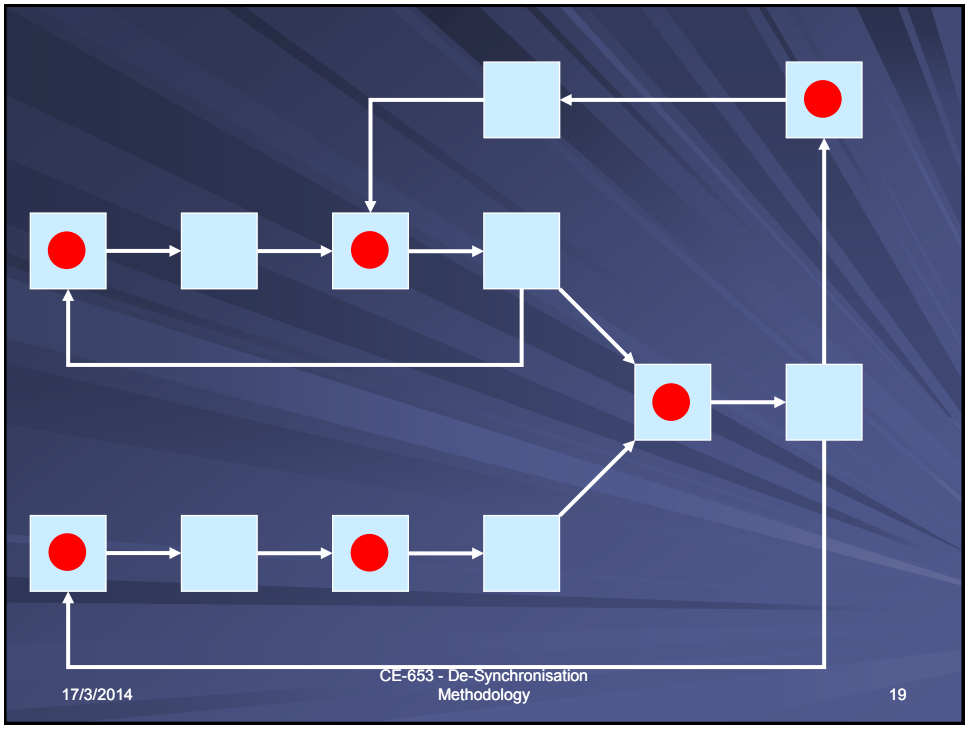

17/3/2014
CE-653 - De-Synchronisation Methodology
14

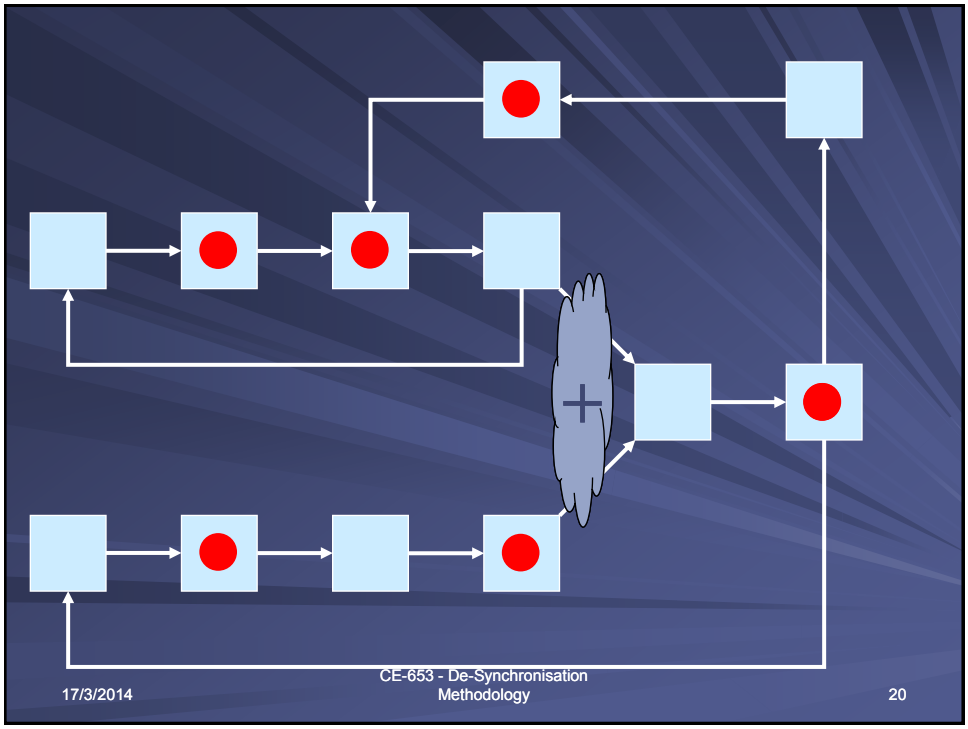

17/3/2014
CE-653 - De-Synchronisation Methodology
15

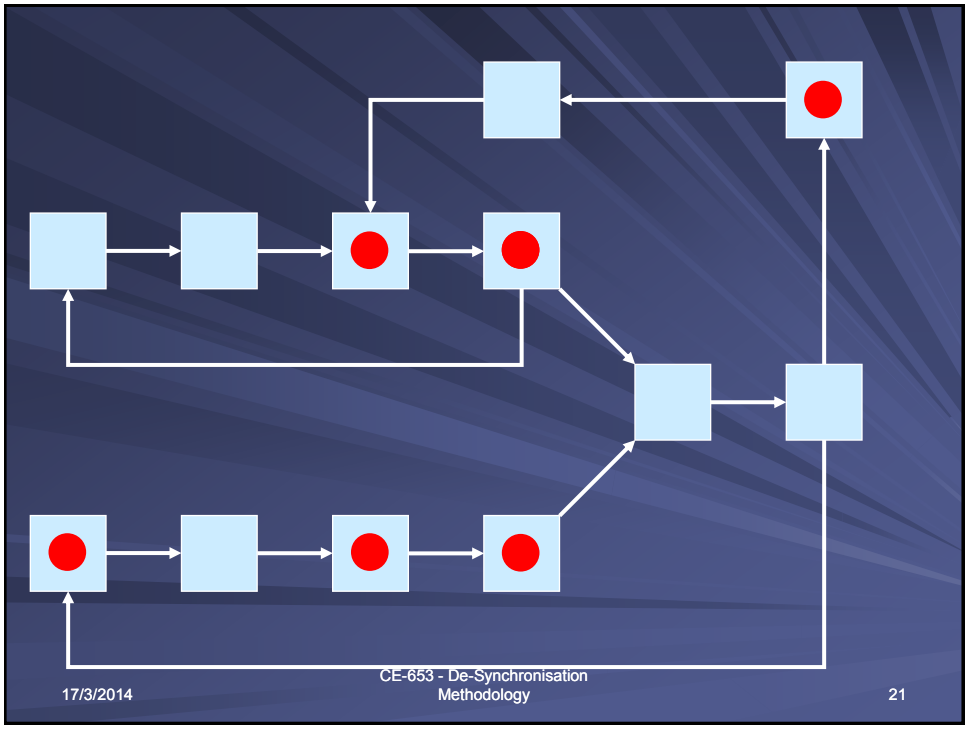

17/3/2014
CE-653 - De-Synchronisation Methodology
16

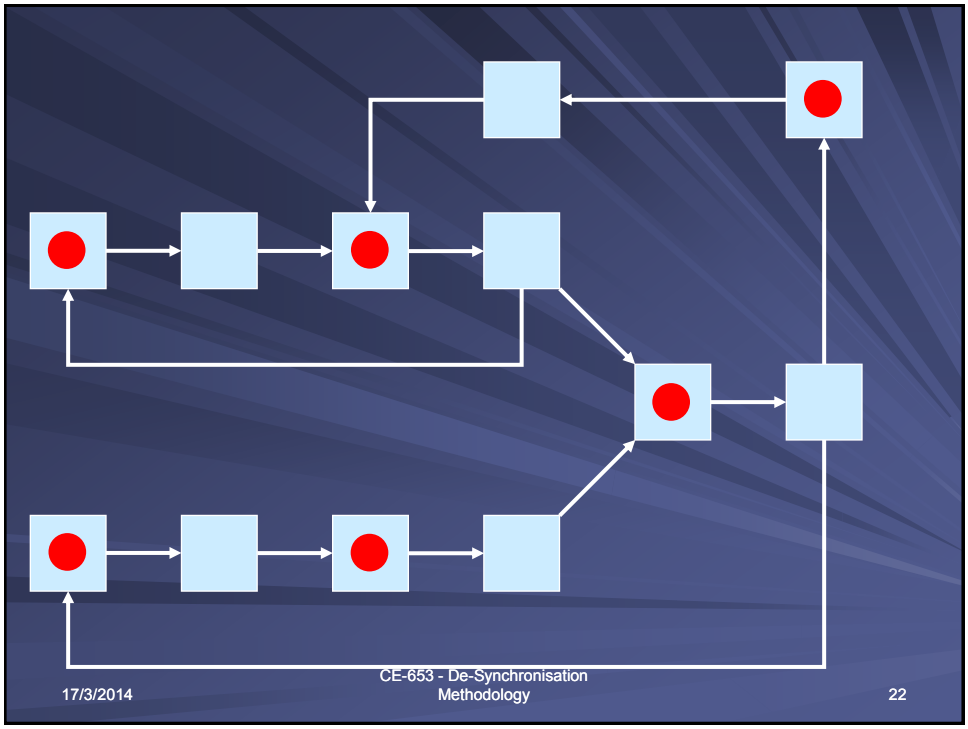


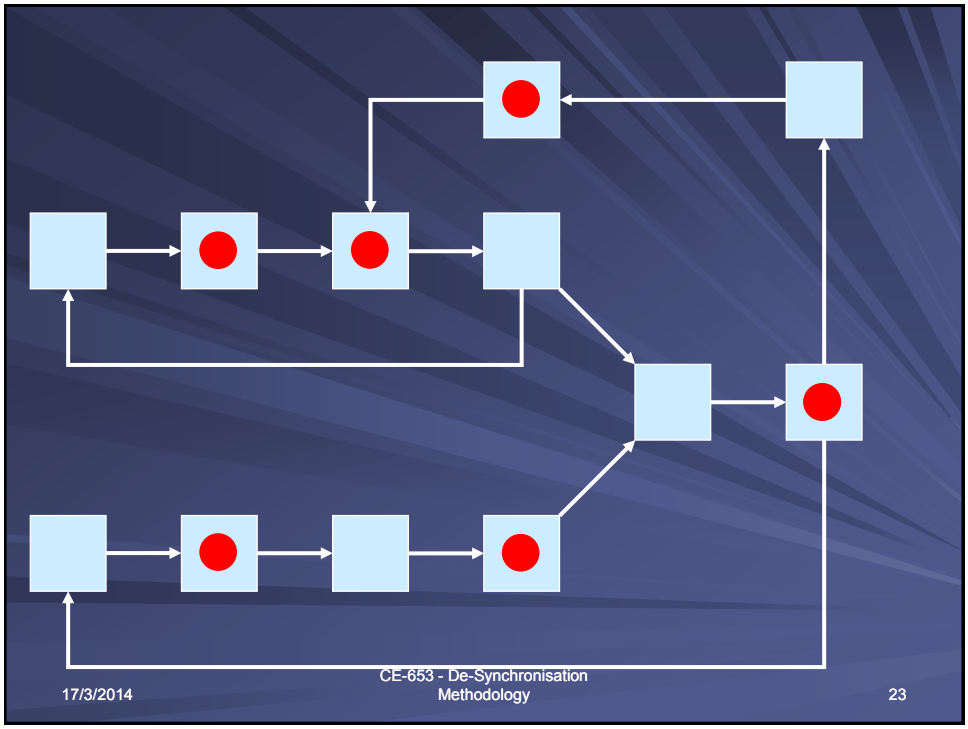
De-synchronized flow

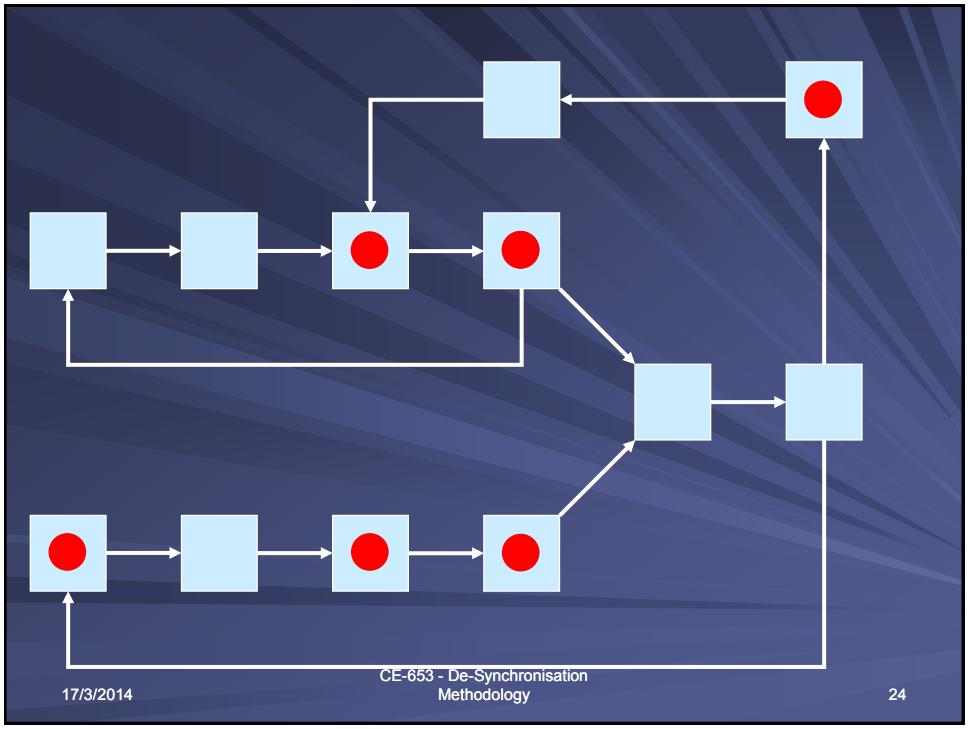

17/3/2014
CE-653 - De-Synchronisation Methodology
19

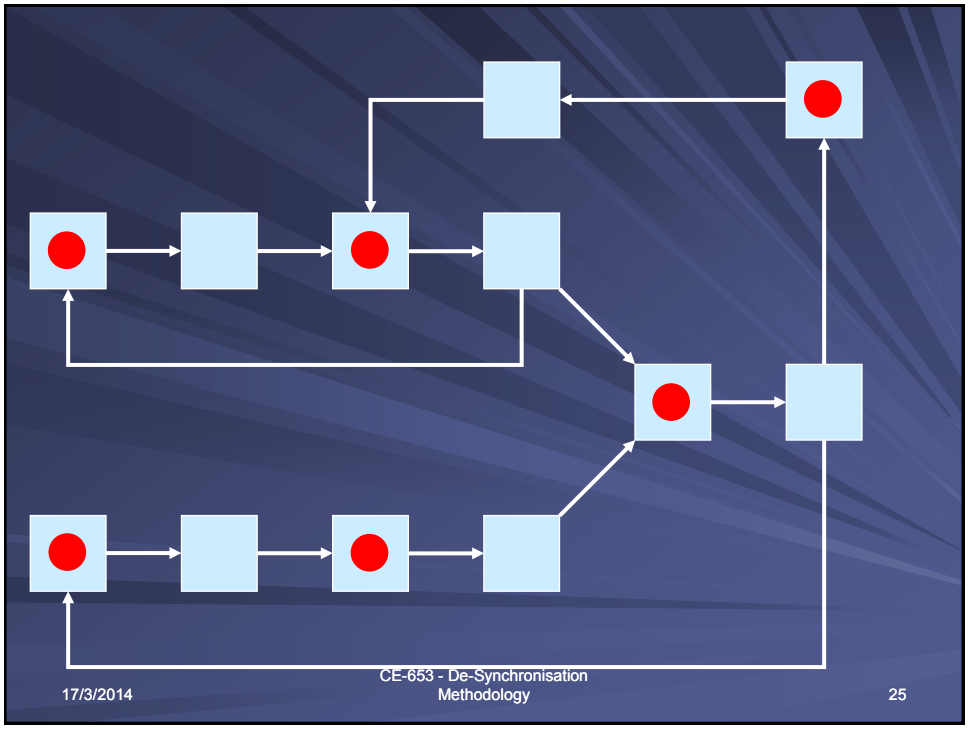

17/3/2014
CE-653 - De-Synchronisation Methodology
20

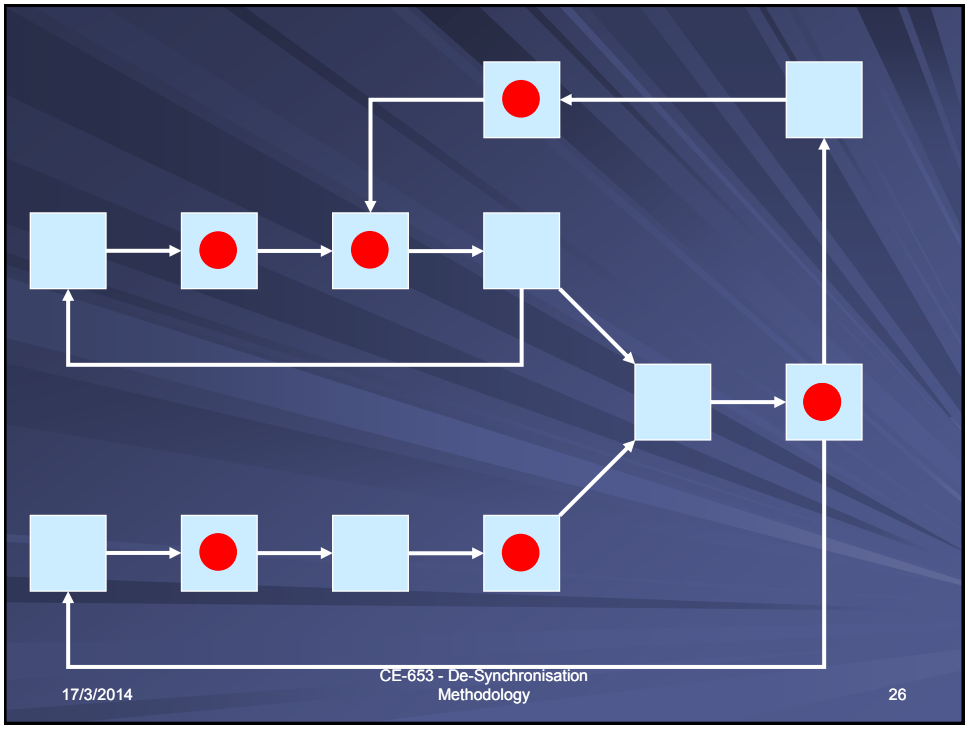


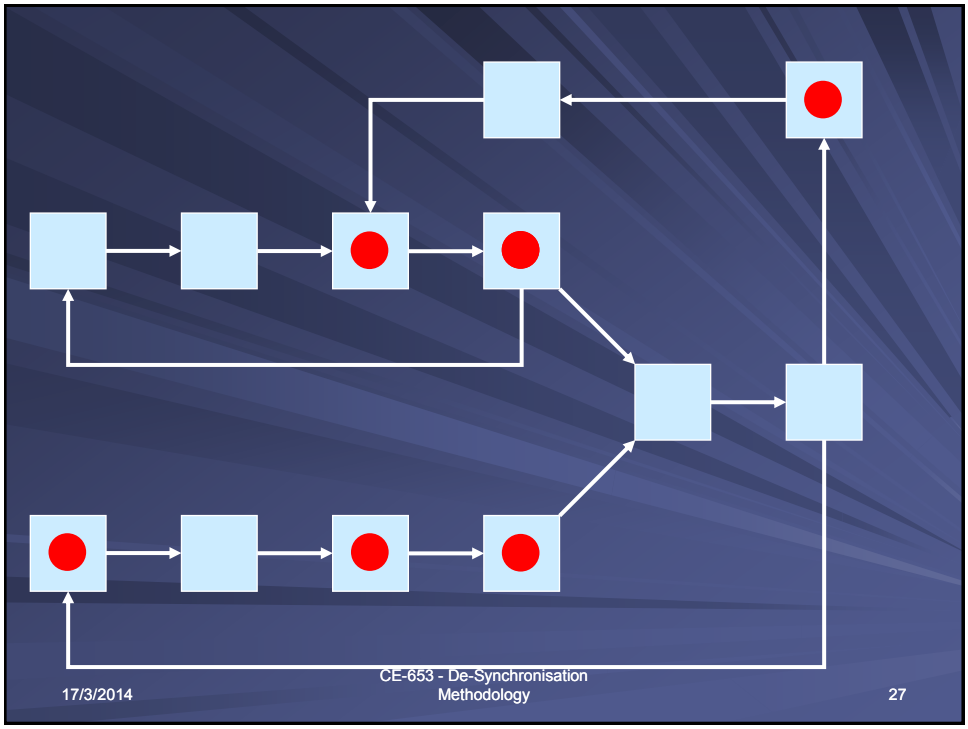


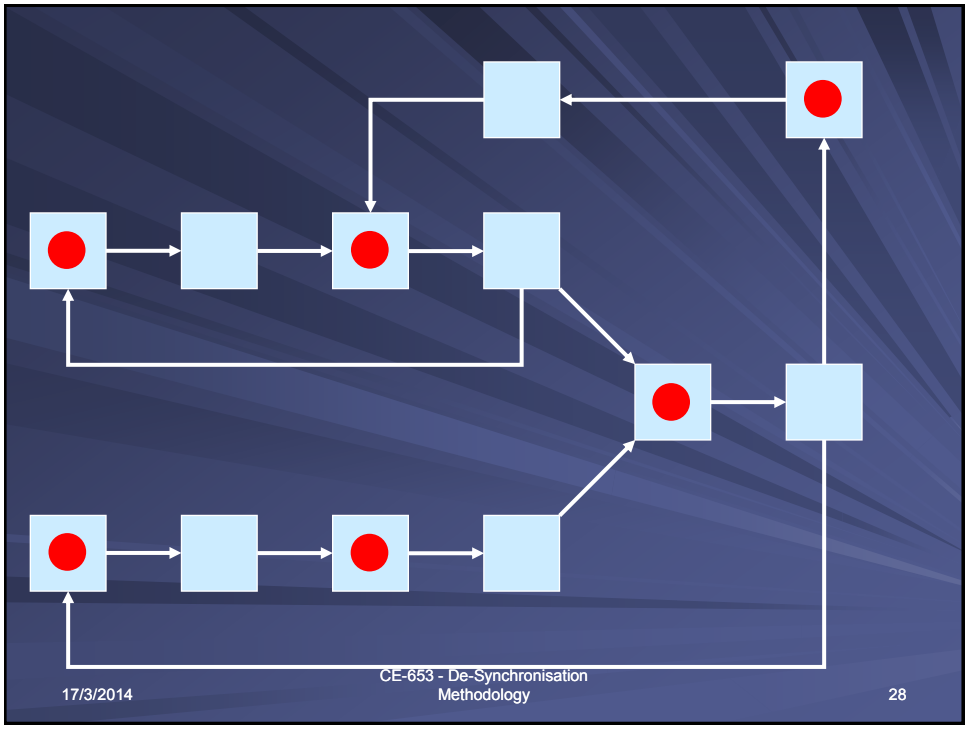


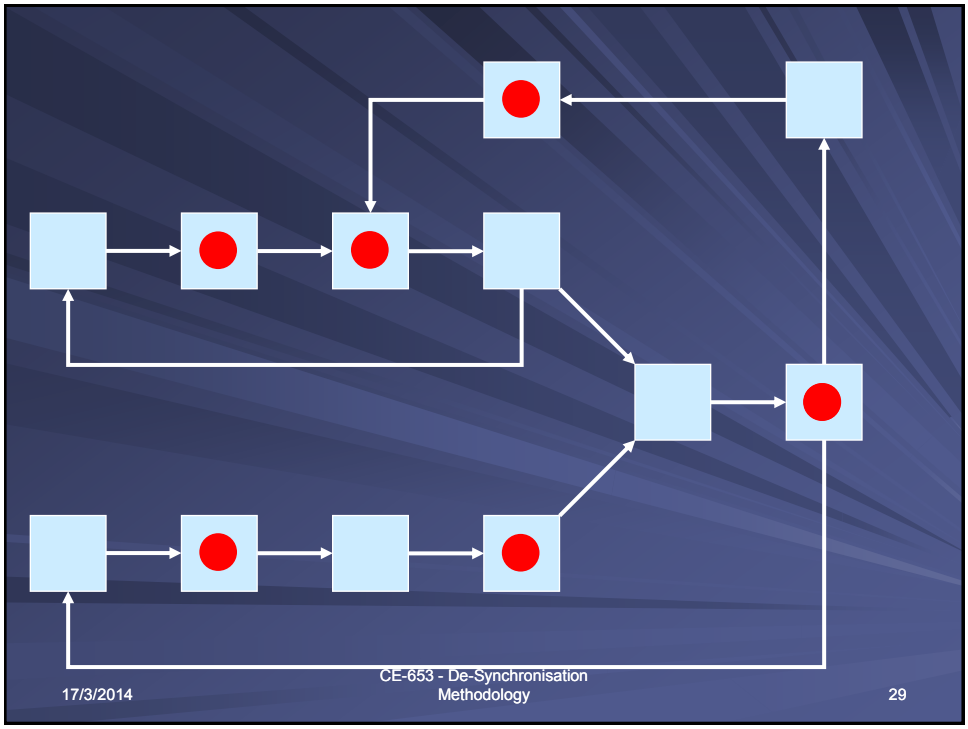


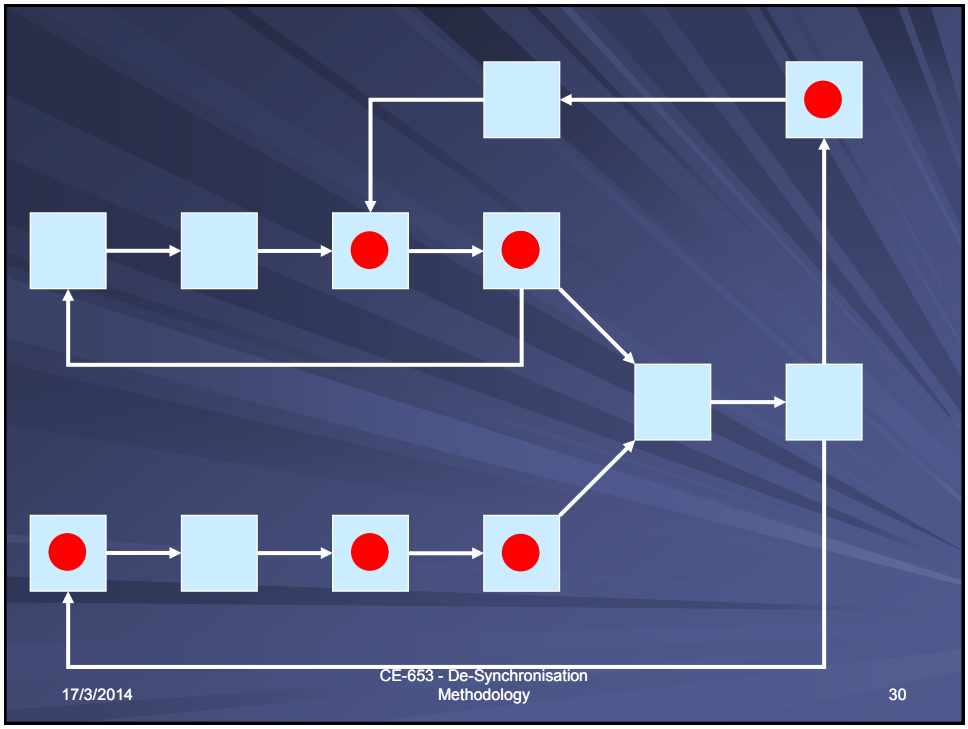






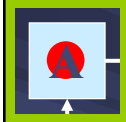




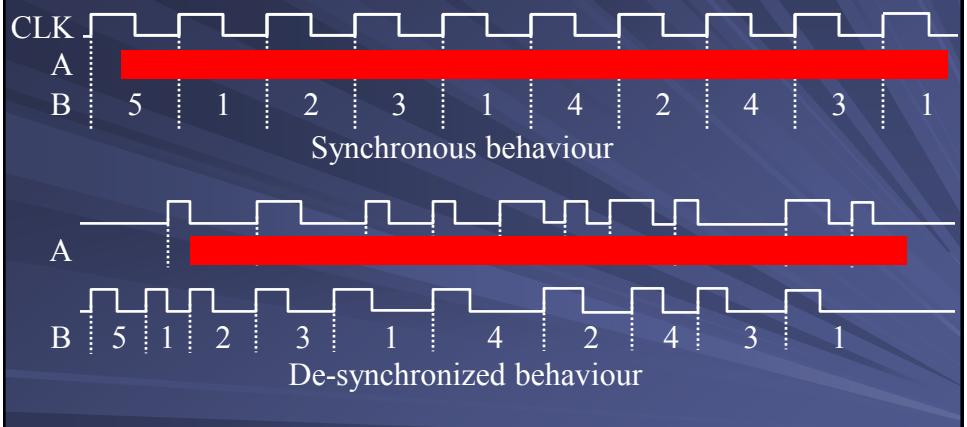


Flow equivalence

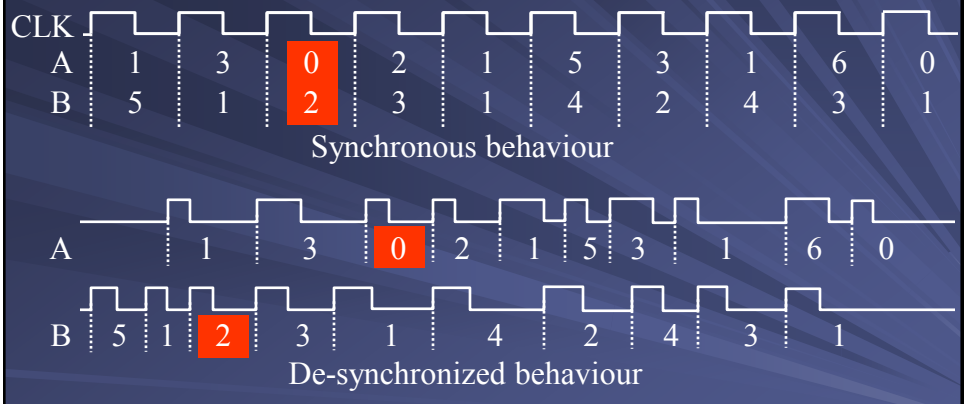
[Guernic, Talpin, Lann, 2003]



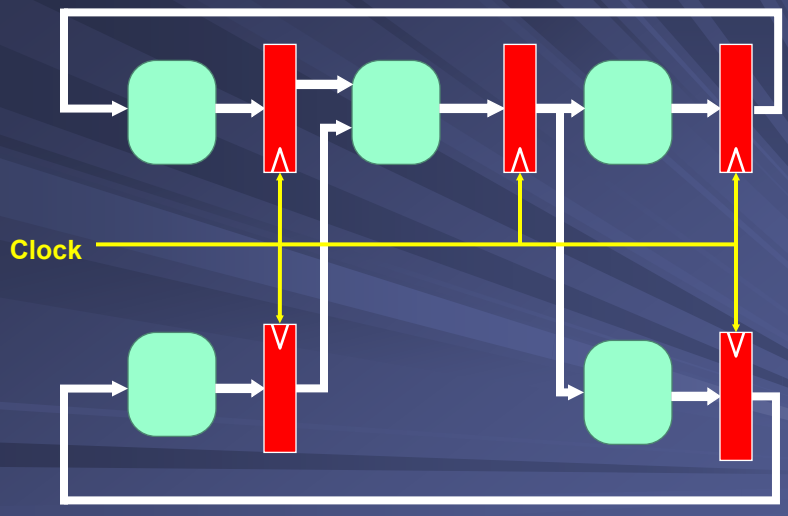
Flow equivalence = Cycle Accuracy!



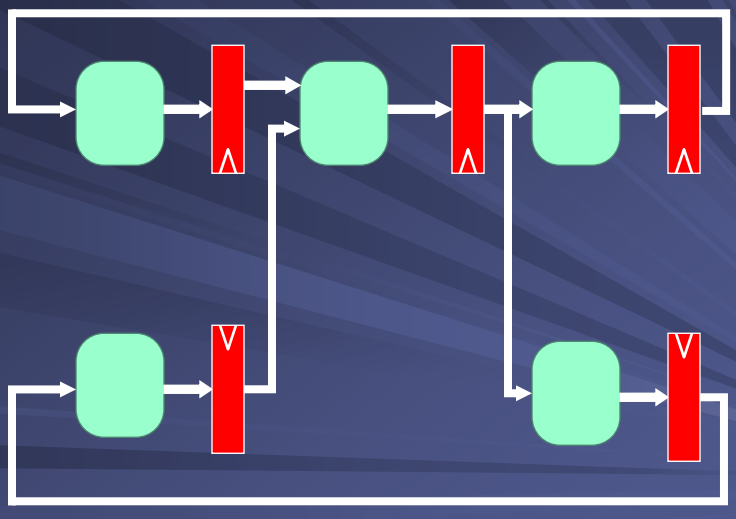
Flow equivalence = Cycle Accuracy!

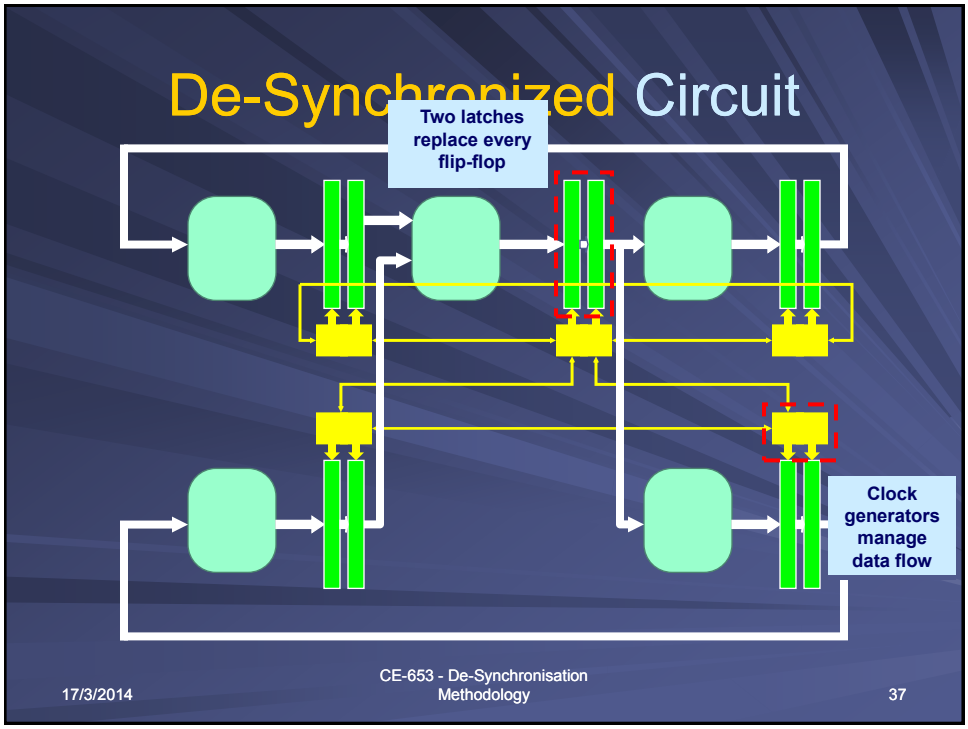


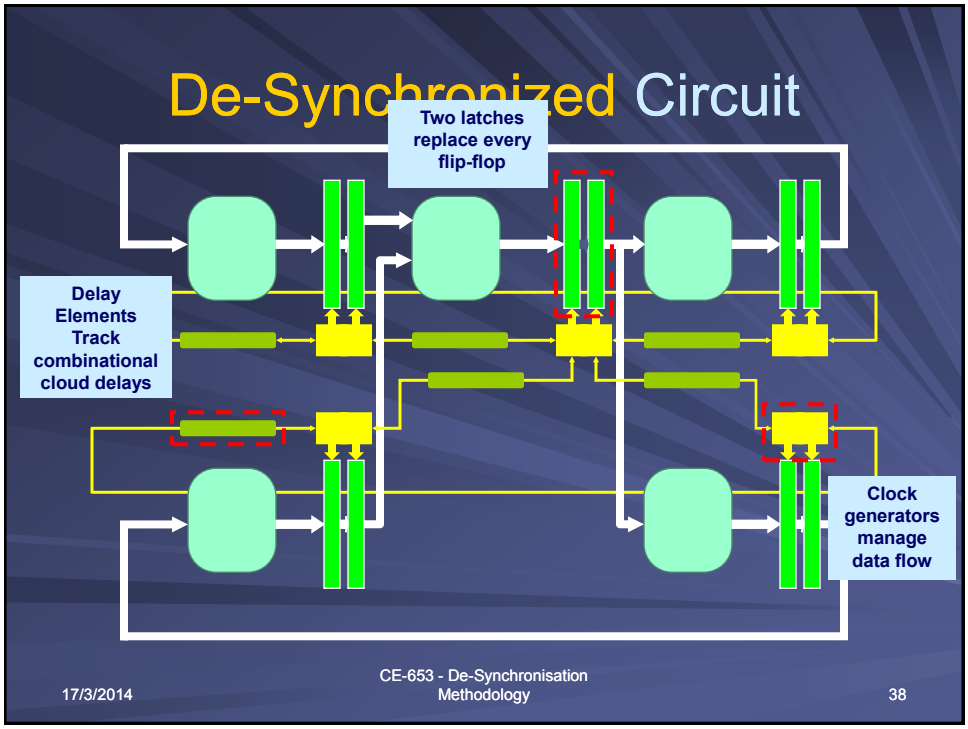
Synchronous Circuit

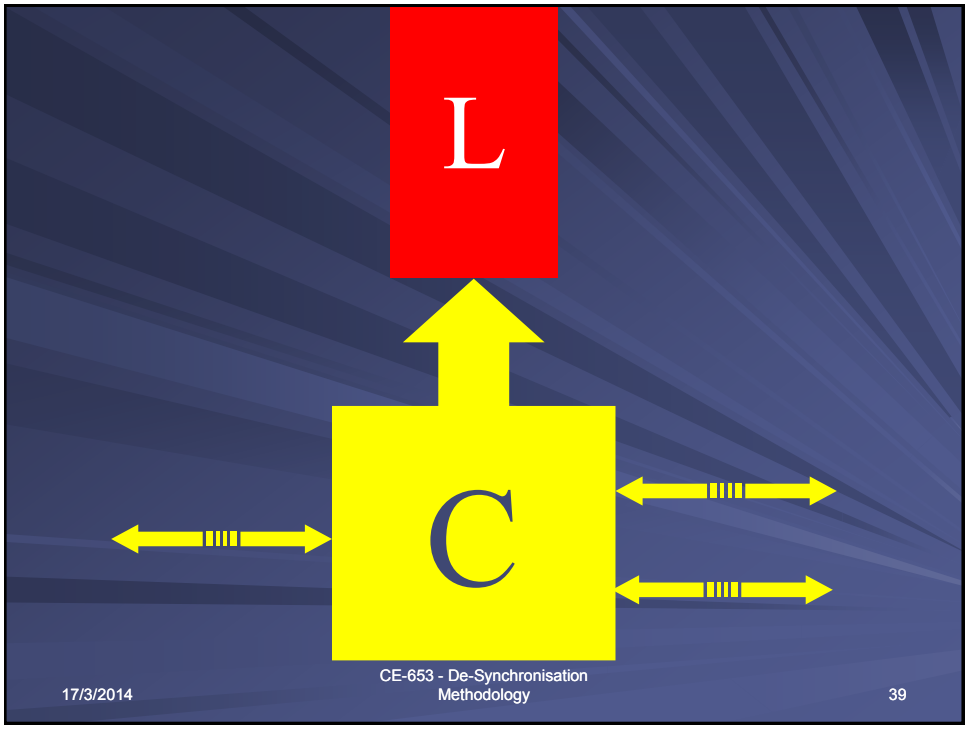


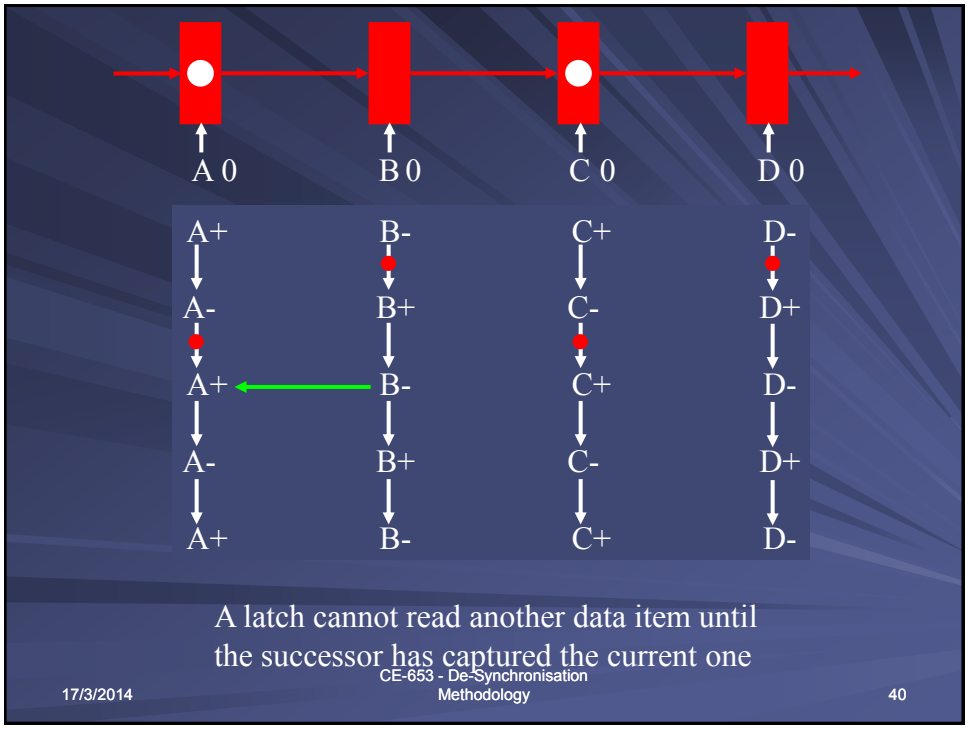
Synchronous Circuit



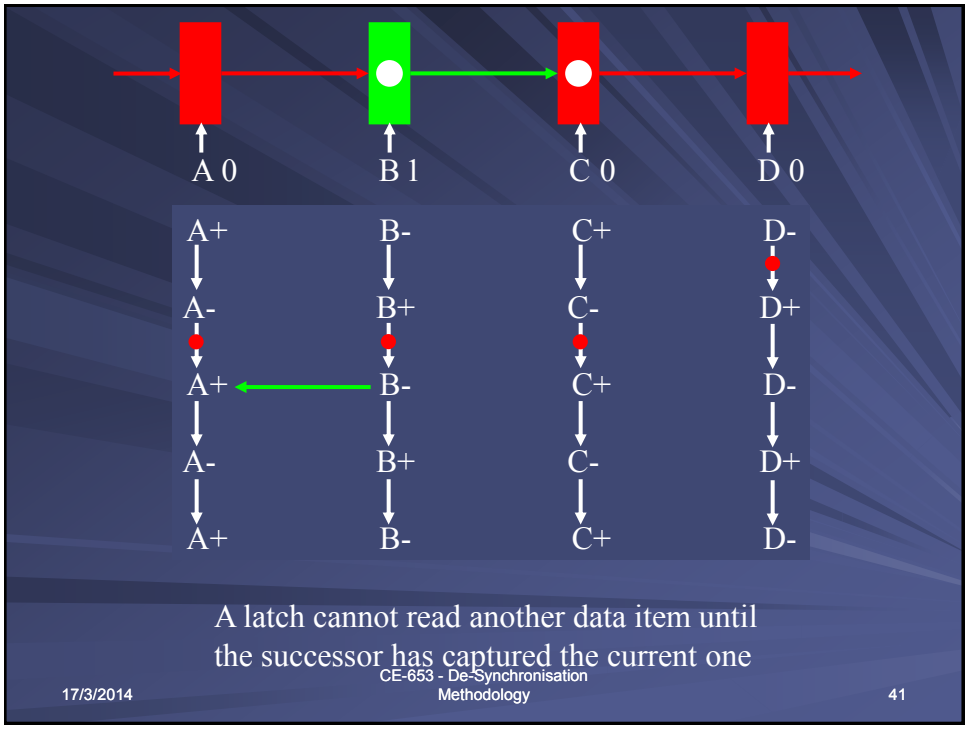

De-Synchronized Circuit
Two latches replace every flip-flop
Clock generators manage data flow
17/3/2014
CE-653 - De-Synchronisation Methodology
37


De-Synchronized Circuit
Two latches replace every flip-flop
Delay Elements Track combinational cloud delays
Clock generators manage data flow
17/3/2014
CE-653 - De-Synchronisation Methodology
38

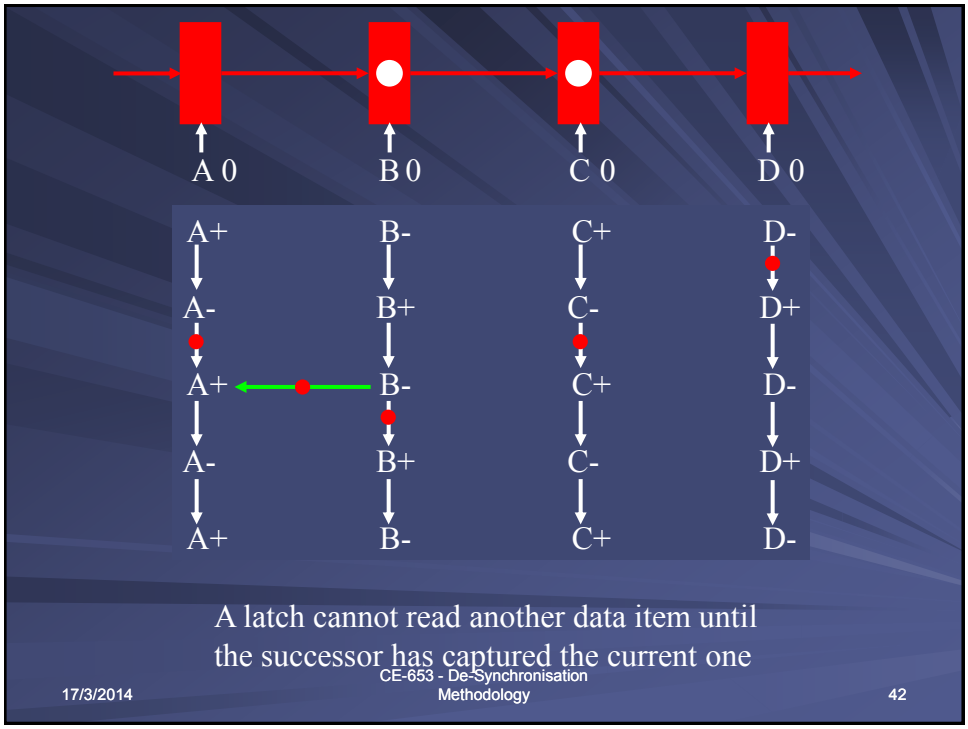




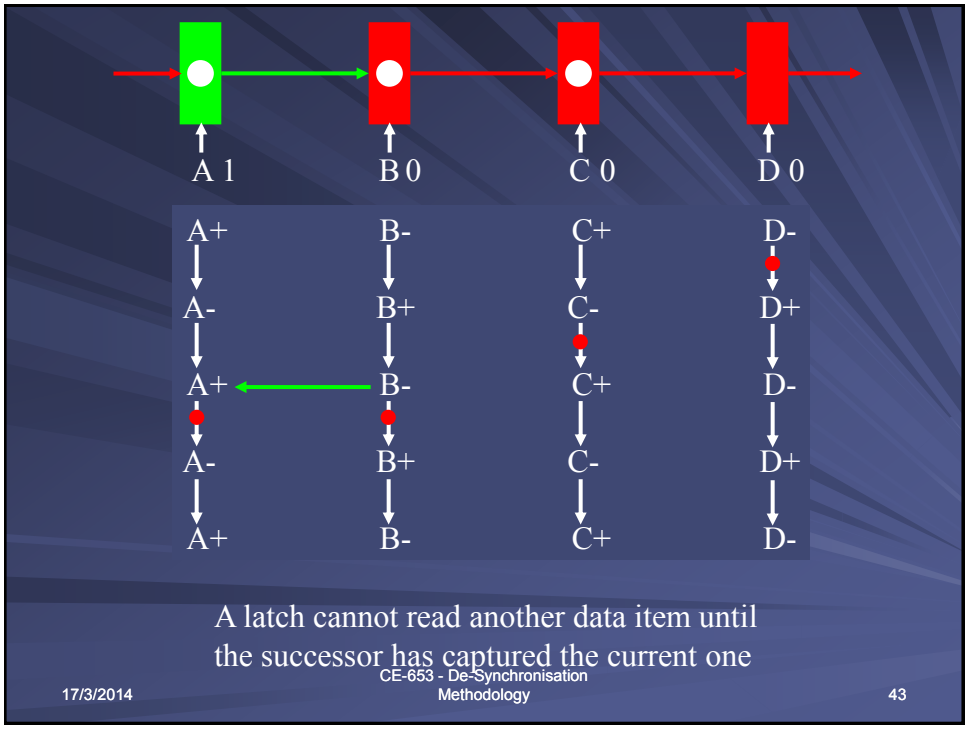
A latch cannot read another data item until the successor has captured the current one



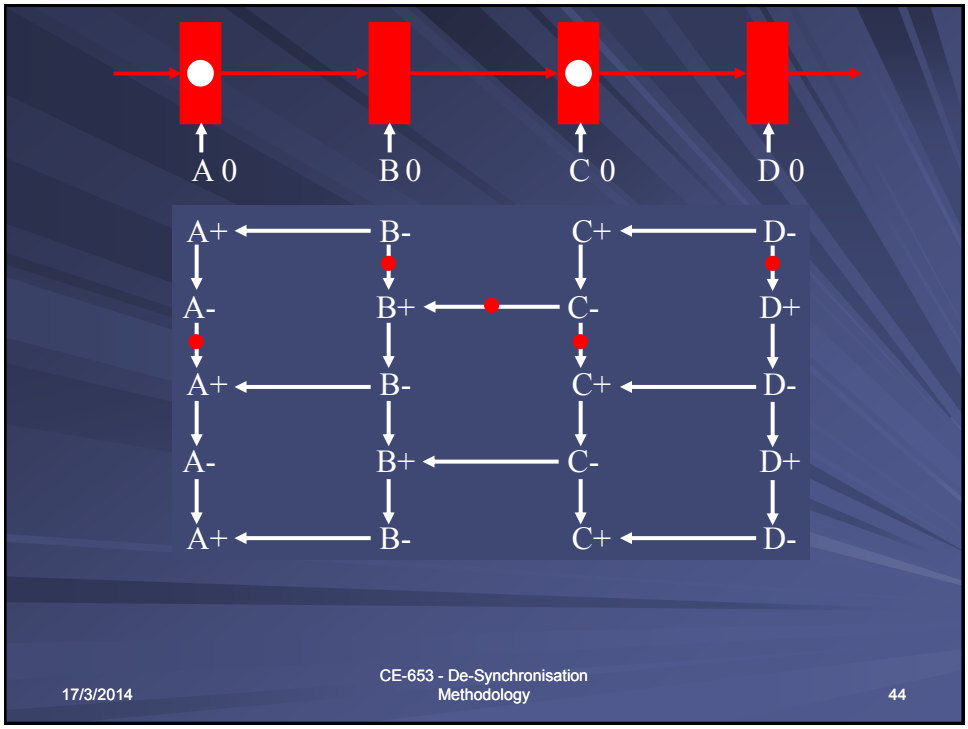
A latch cannot read another data item until the successor has captured the current one

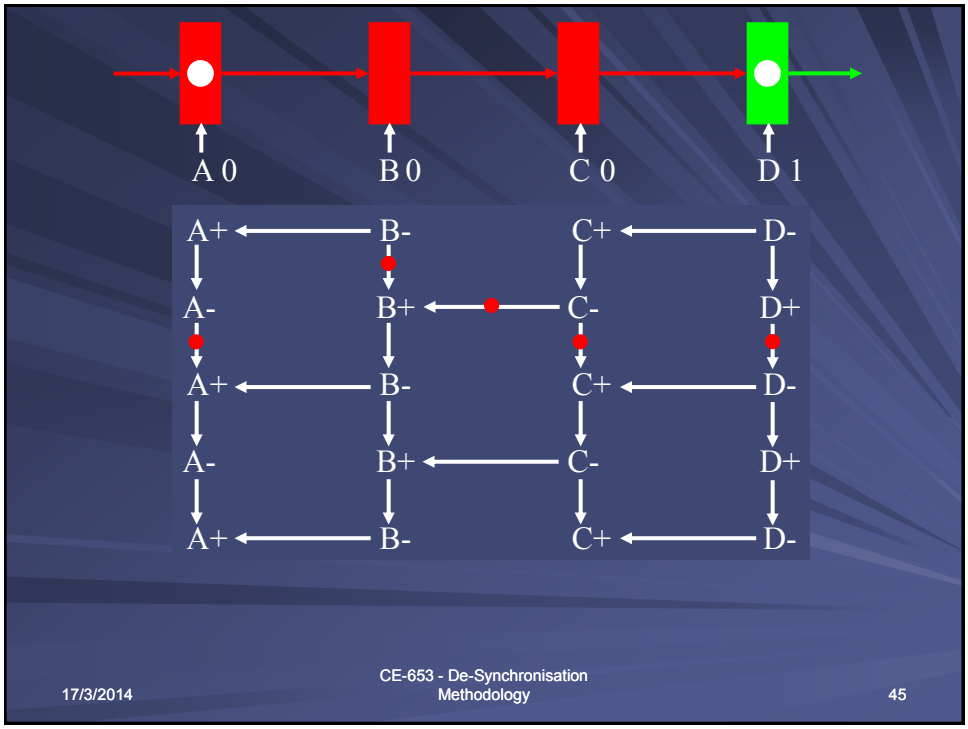


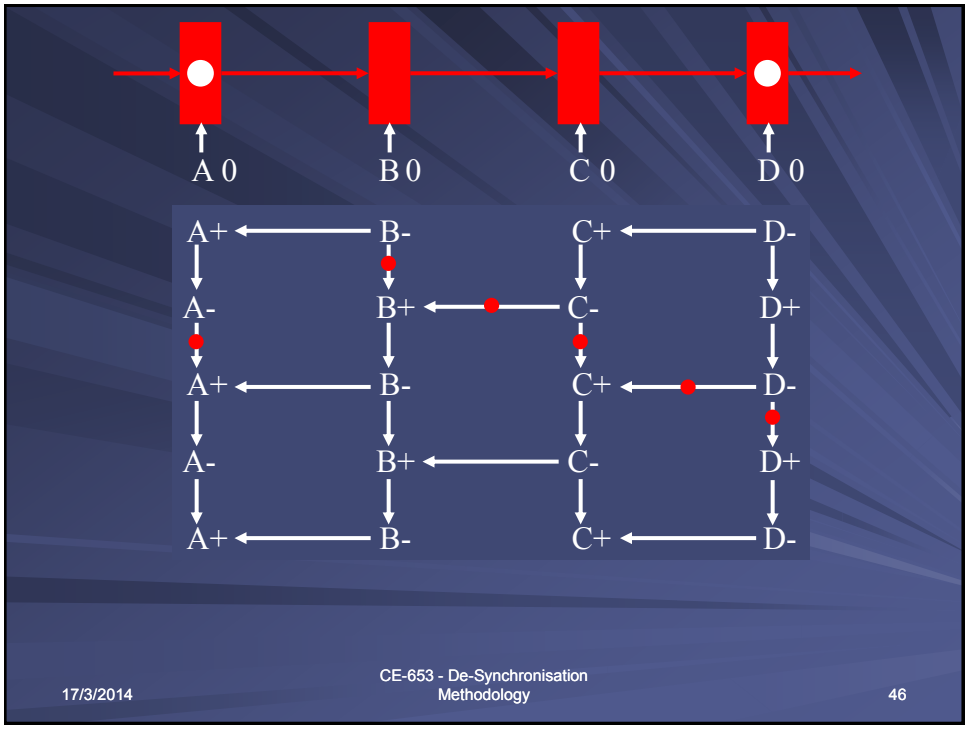
A latch cannot read another data item until the successor has captured the current one

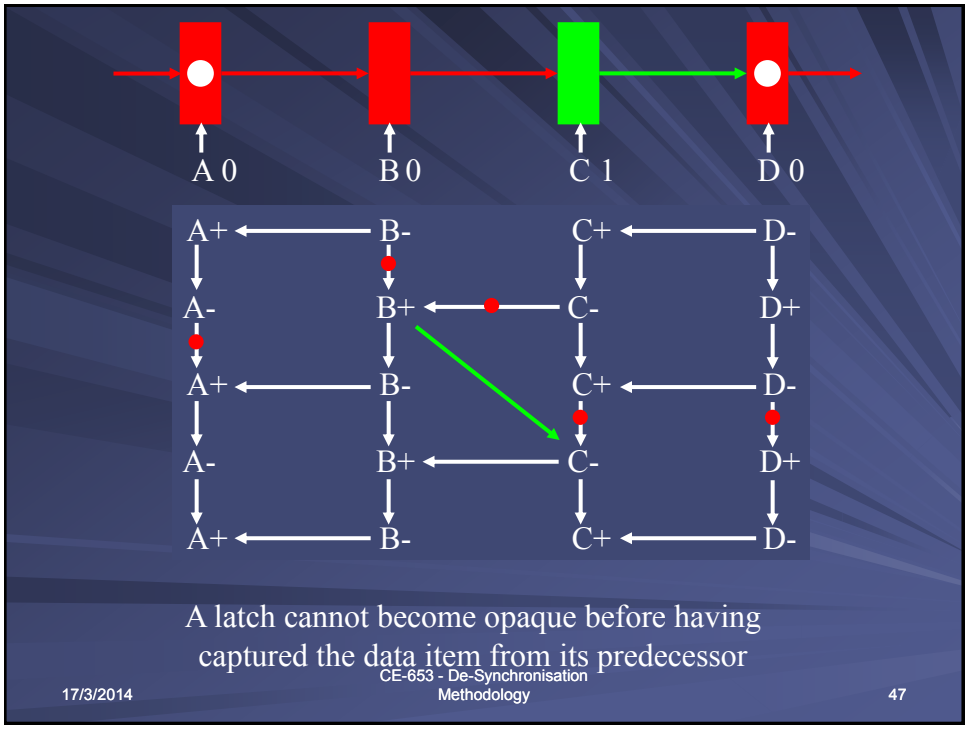


A latch cannot read another data item until the successor has captured the current one

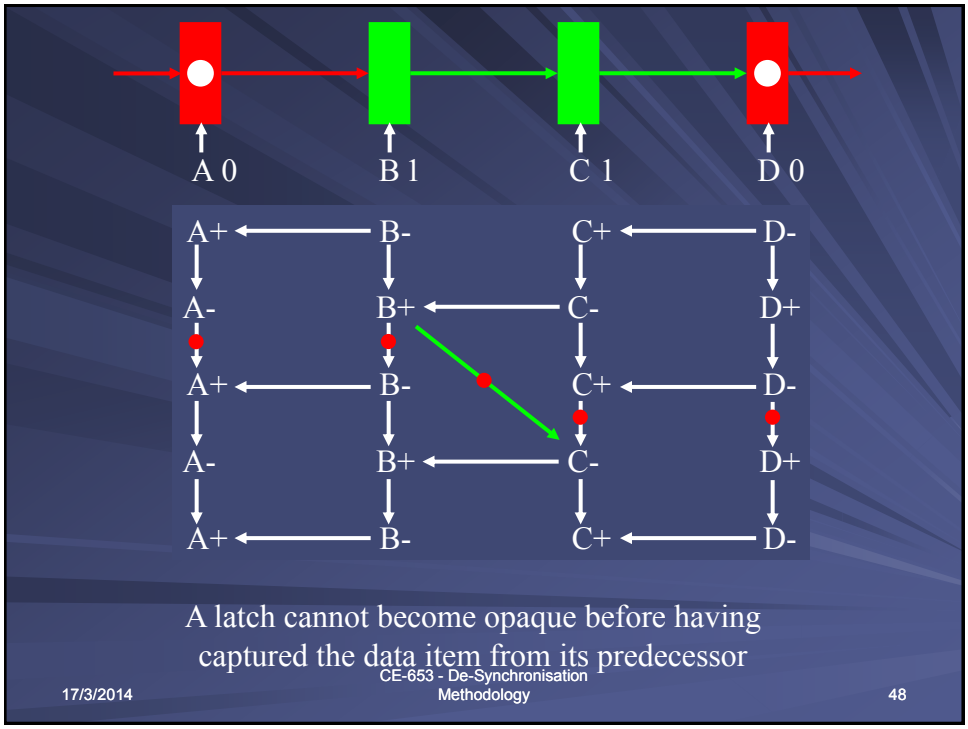



A 0
B 0
C 0
D 1
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-

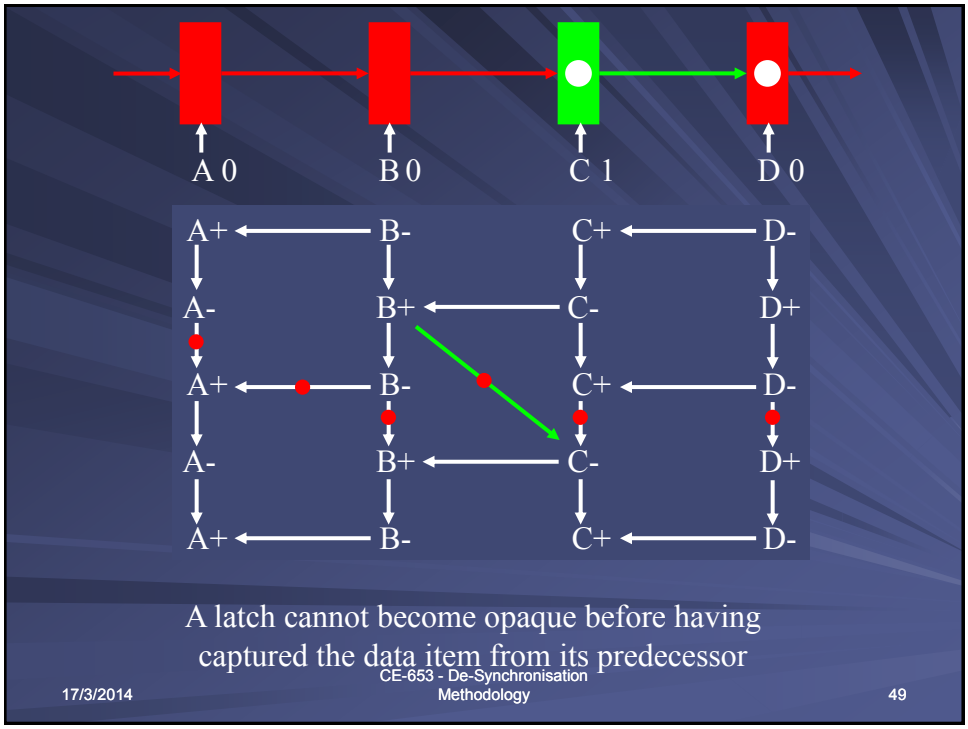

A 0
B 0
C 0
D 0
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-



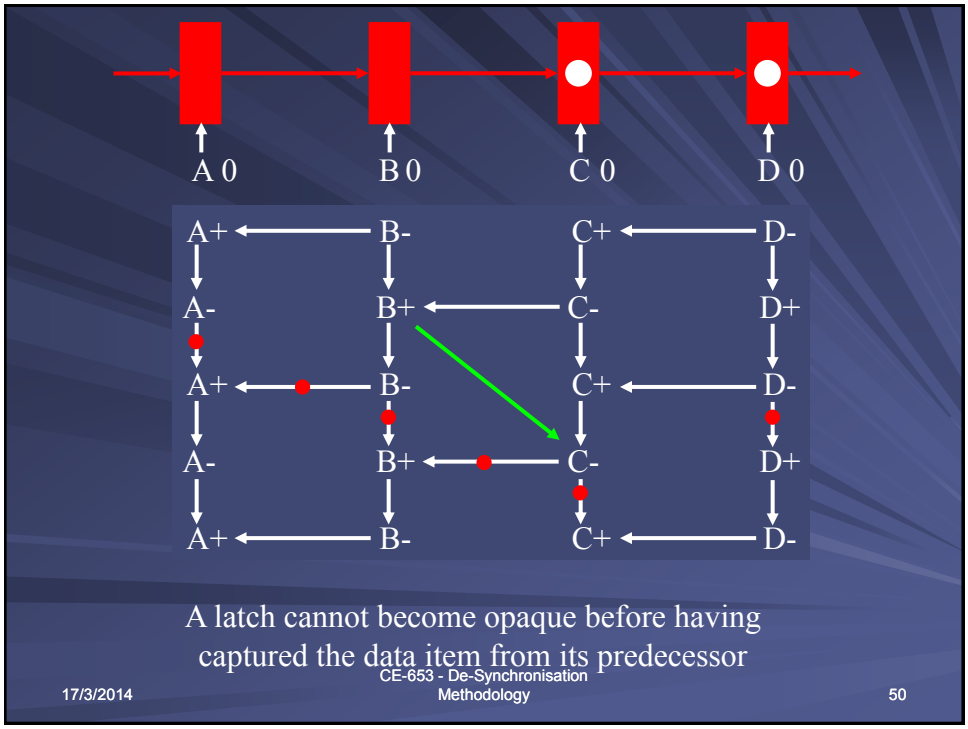
A latch cannot become opaque before having captured the data item from its predecessor



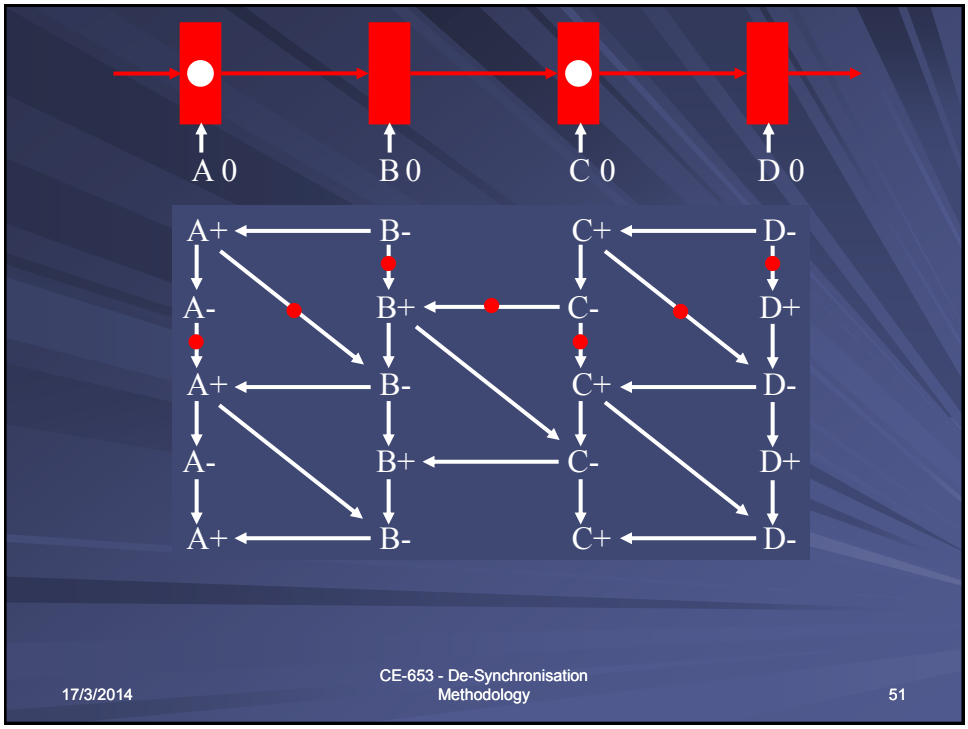
A latch cannot become opaque before having captured the data item from its predecessor

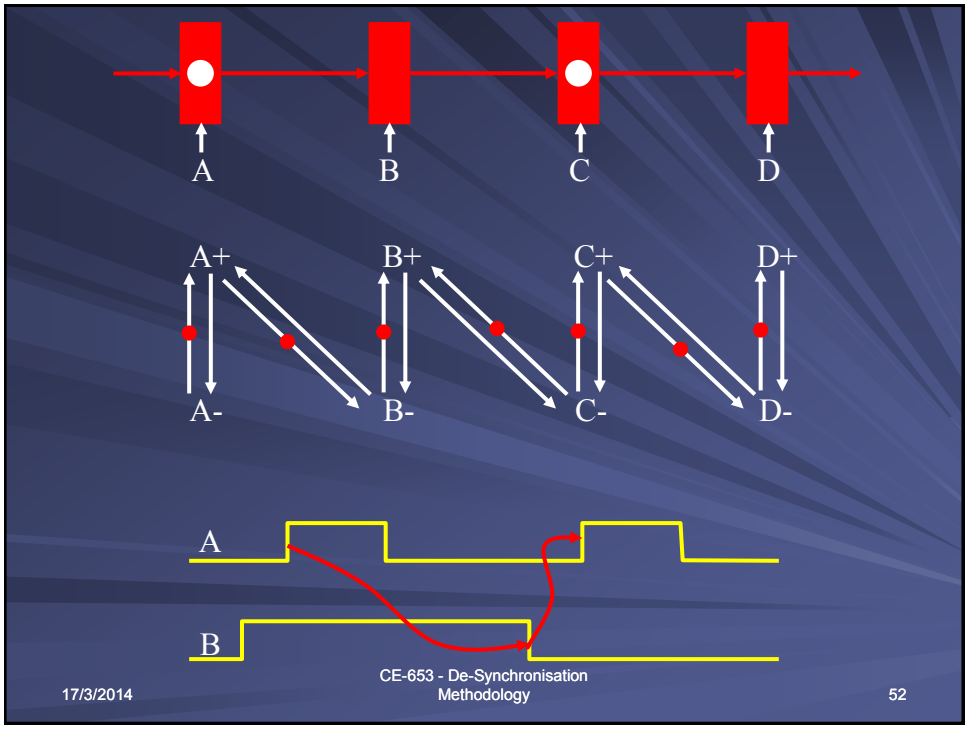


A latch cannot become opaque before having captured the data item from its predecessor

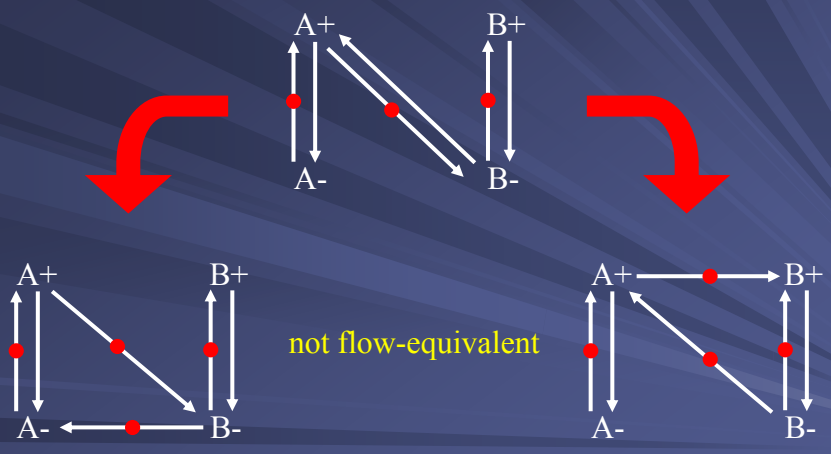


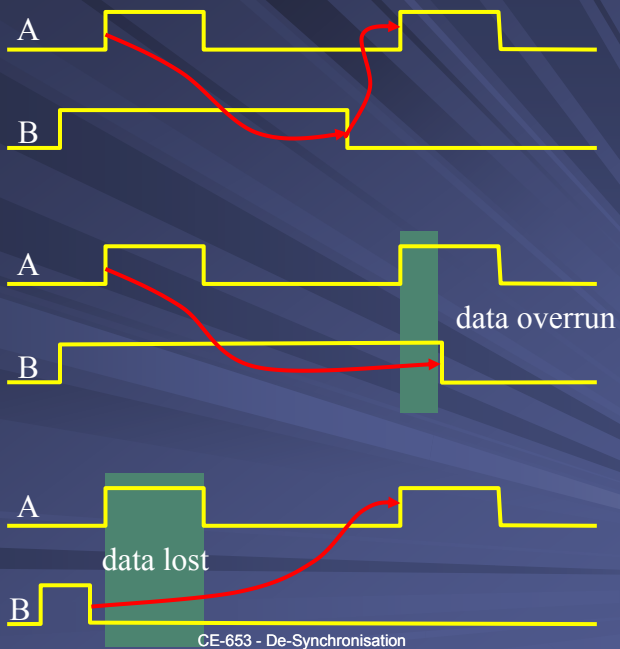
A latch cannot become opaque before having captured the data item from its predecessor

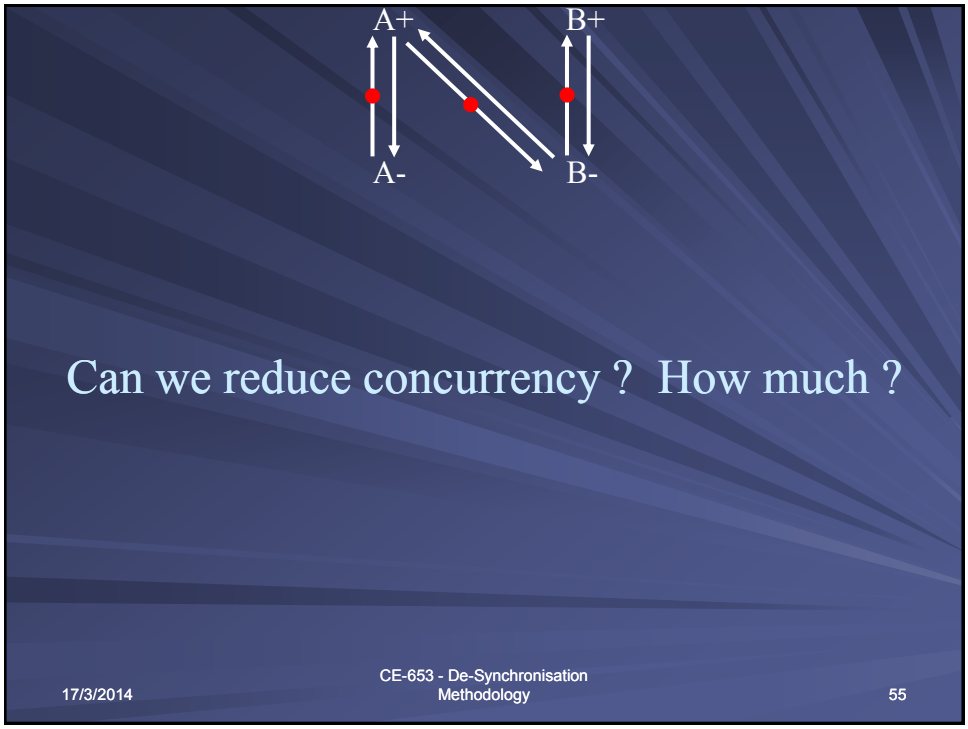

A 0
B 0
C 0
D 0
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-
A-
B+
C-
D+
A+
B-
C+
D-

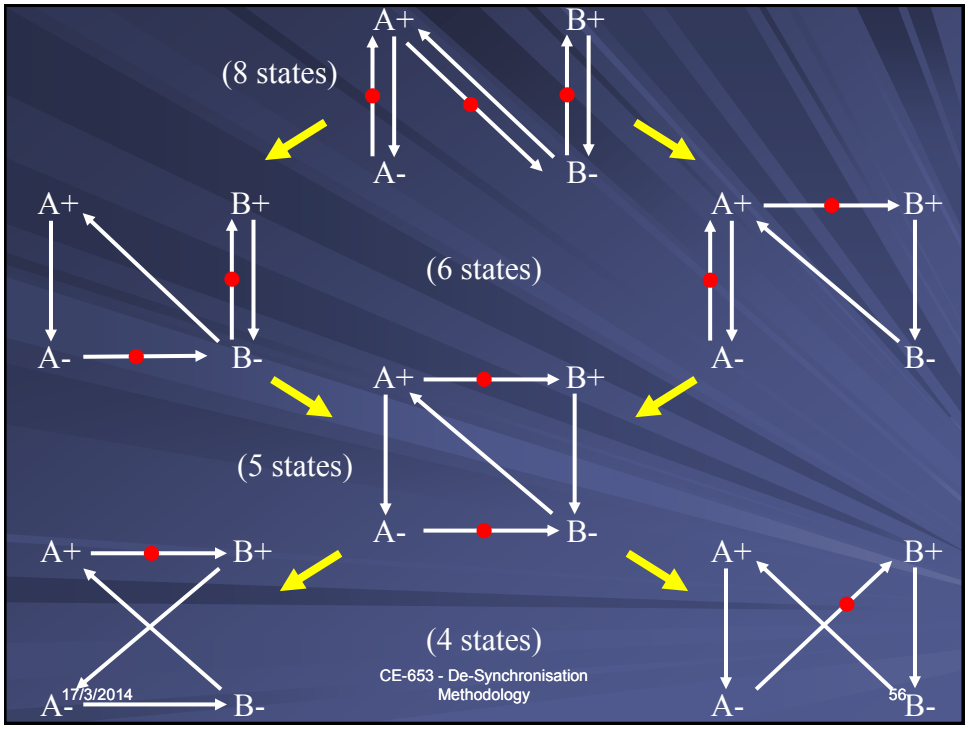

A
B
C
D
A+
B+
C+
D+
A-
B-
C-
D-
A
B

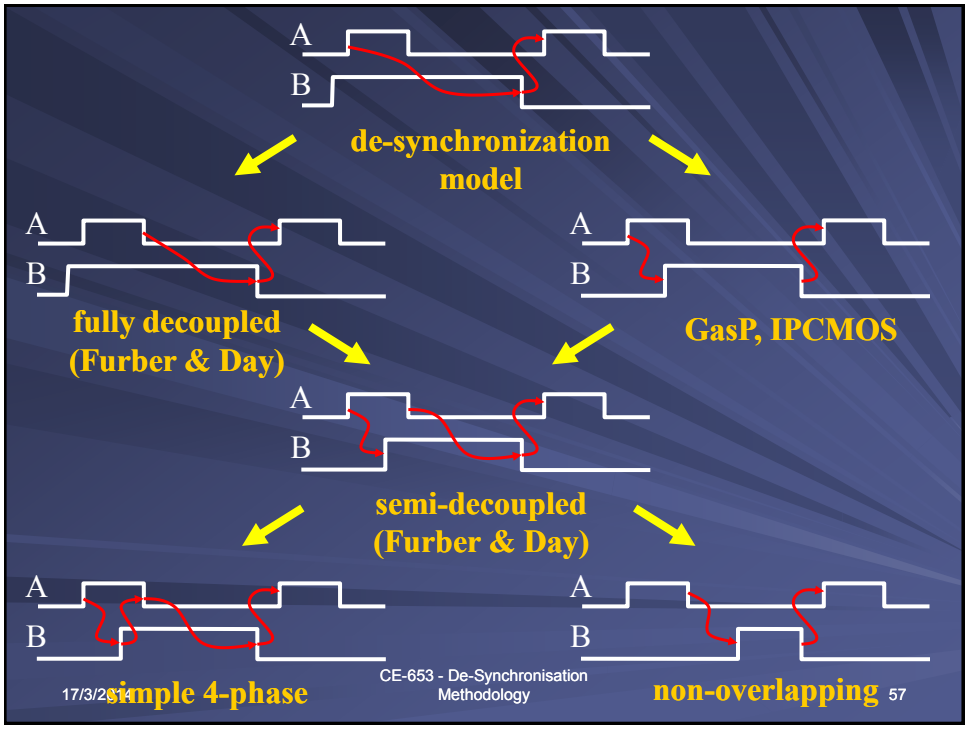
Can we increase concurrency ?

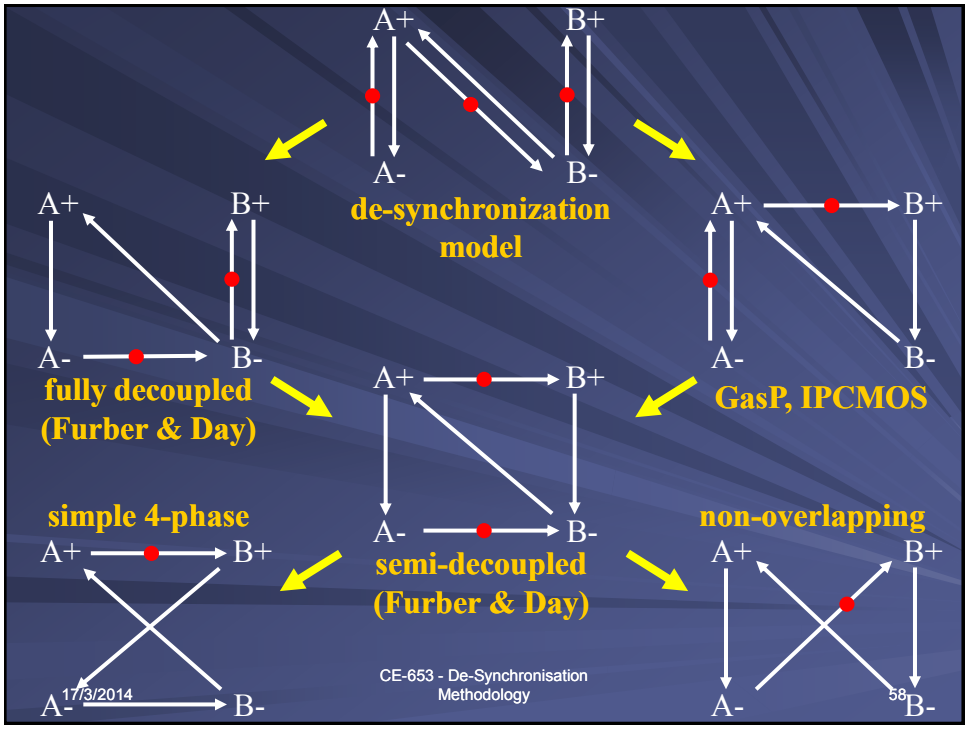


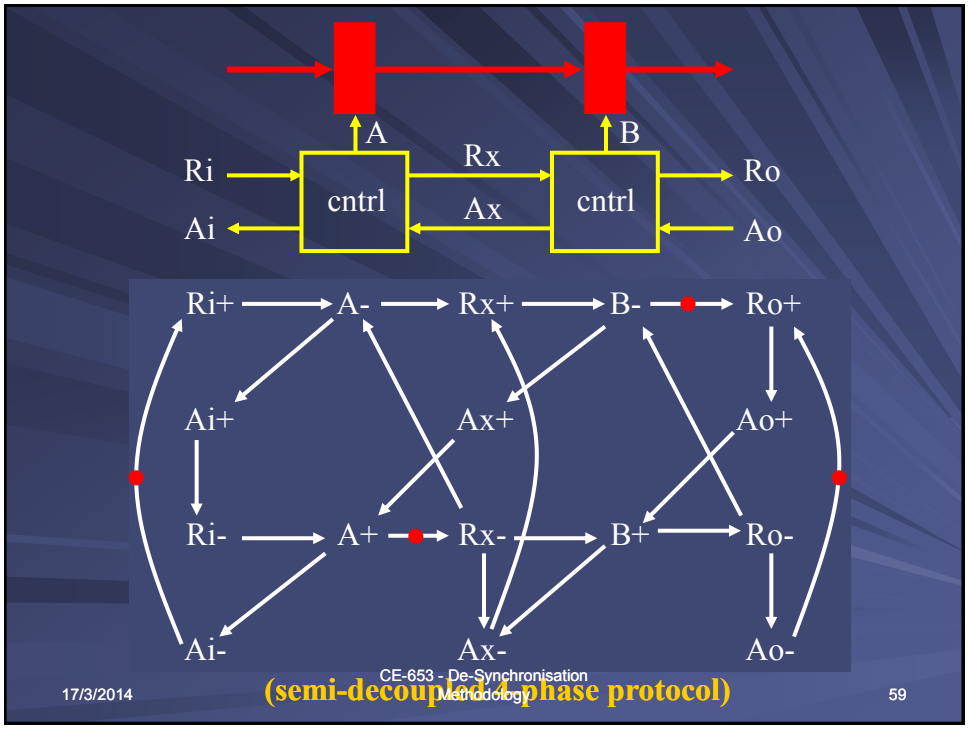



A+
B+
A-
B-
Can we reduce concurrency ? How much ?
17/3/2014
CE-653 - De-Synchronisation Methodology
55

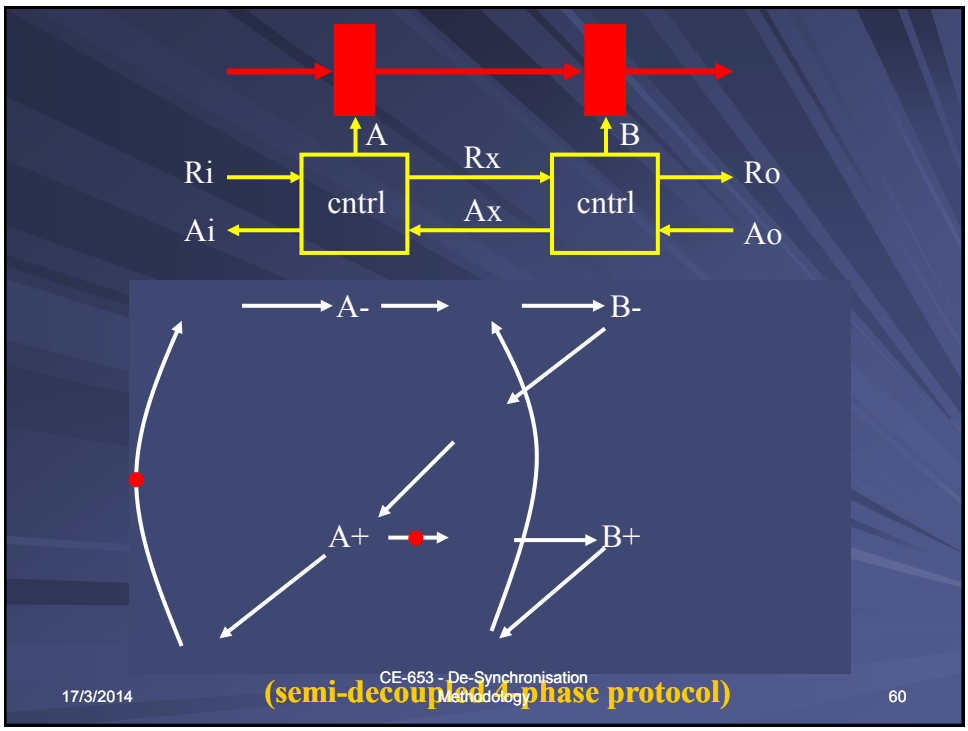

(8 states)
A+
B+
A-
B-
(6 states)
(5 states)
(4 states)
17/3/2014
CE-653 - De-Synchronisation Methodology
56


A
B
de-synchronization model
fully decoupled (Furber & Day)
GasP, IPCMOS
semi-decoupled (Furber & Day)
simple 4-phase
non-overlapping
17/3/2014
CE-653 - De-Synchronisation Methodology
57

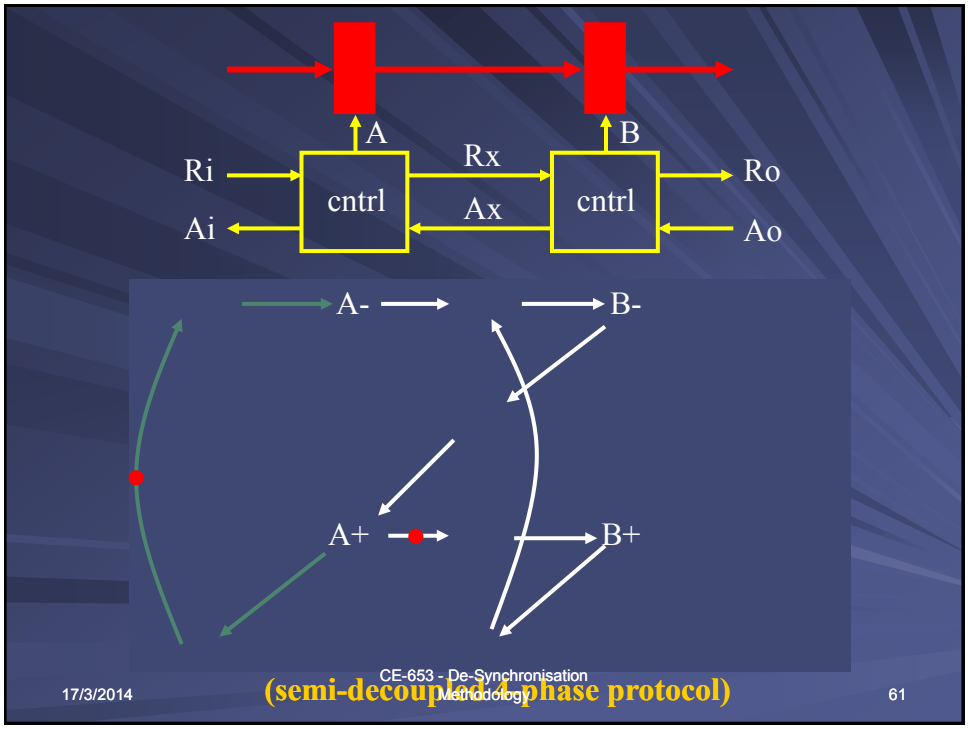

A+
B+
A-
B-
de-synchronization model
fully decoupled (Furber & Day)
GasP, IPCMOS
simple 4-phase
semi-decoupled (Furber & Day)
non-overlapping
17/3/2014
CE-653 - De-Synchronisation Methodology
58



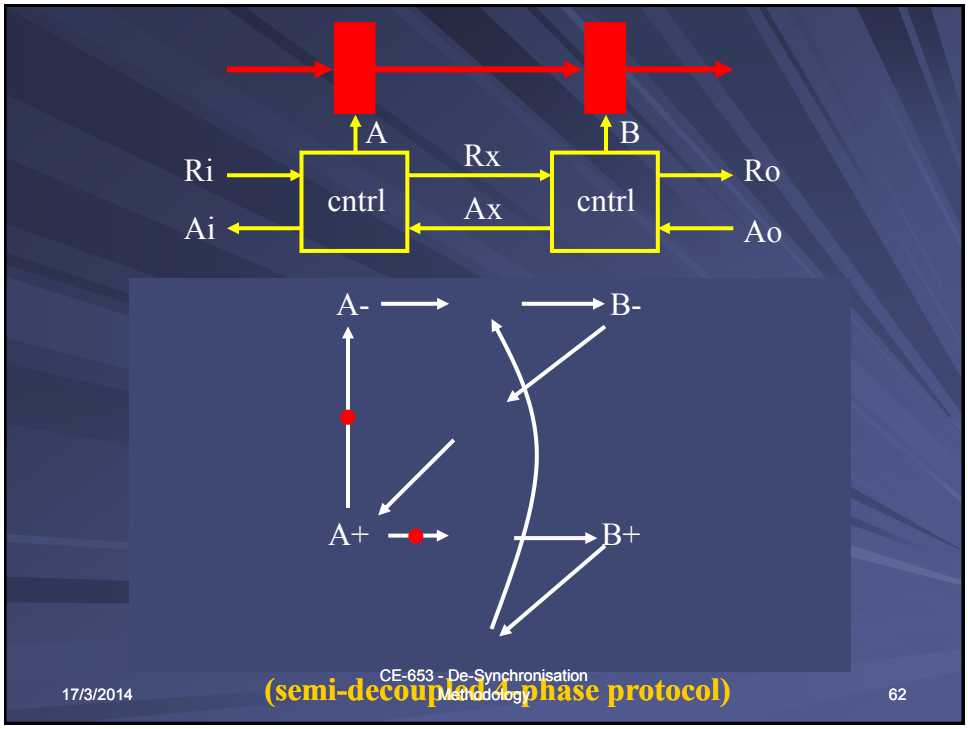
(semi-decoupled 4-phase protocol)



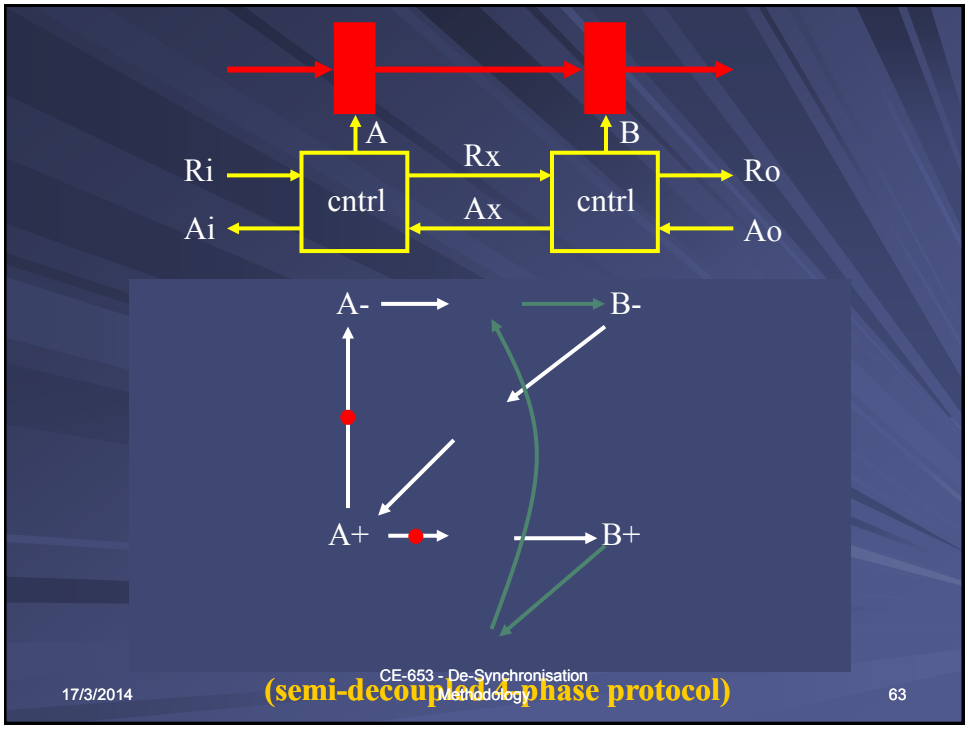
(semi-decoupled 4-phase protocol)



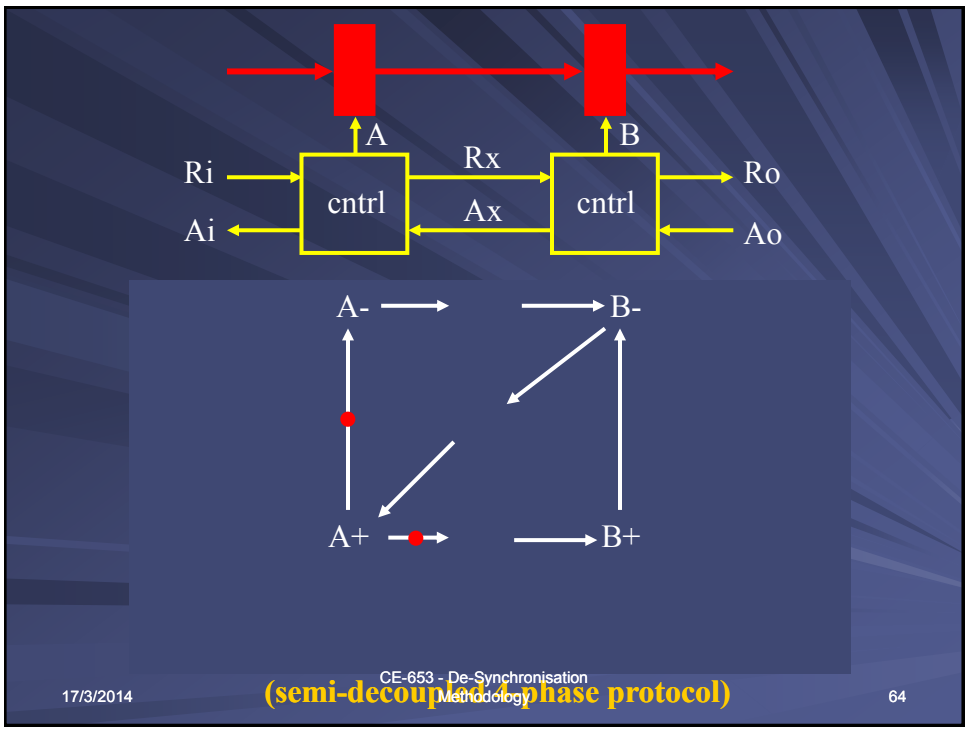
(semi-decoupled 4-phase protocol)



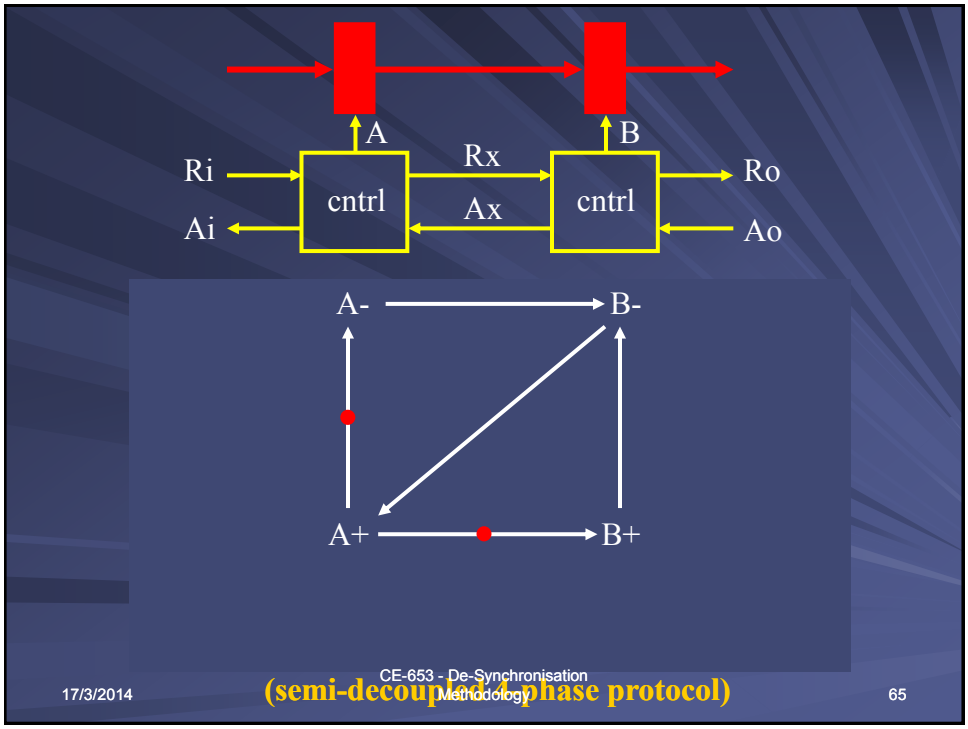
(semi-decoupled 4-phase protocol)



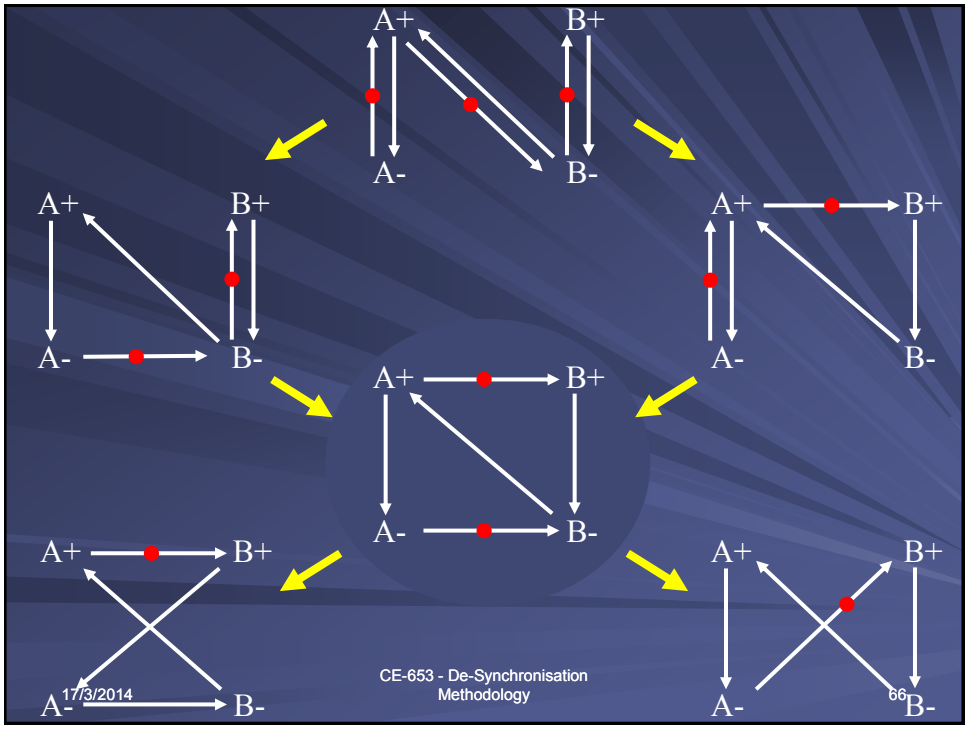
(semi-decoupled 4-phase protocol)

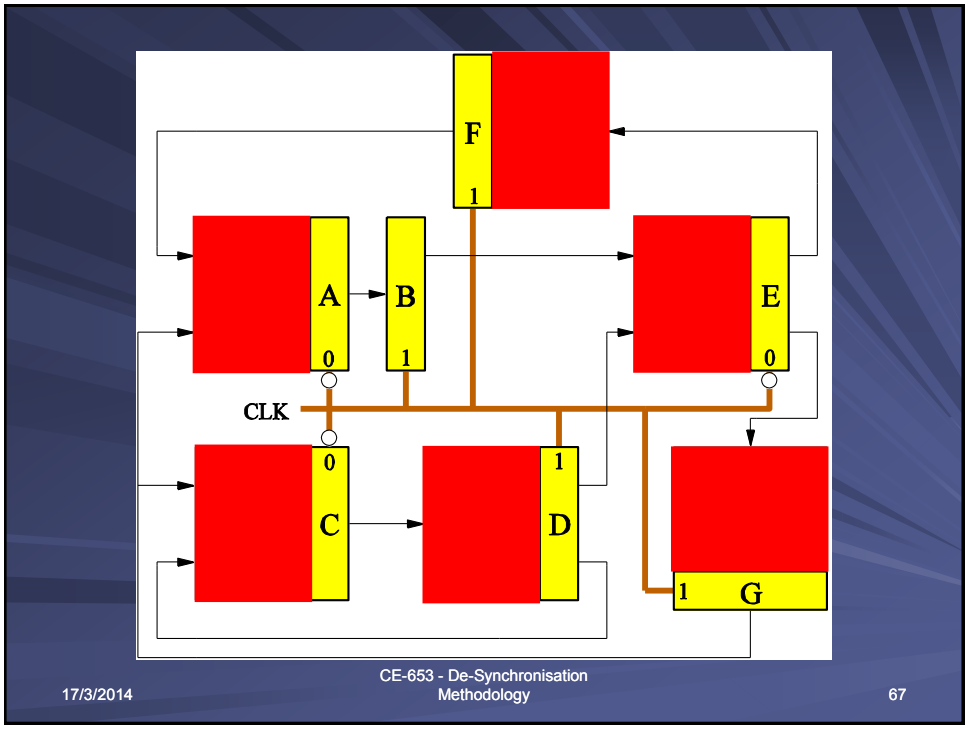


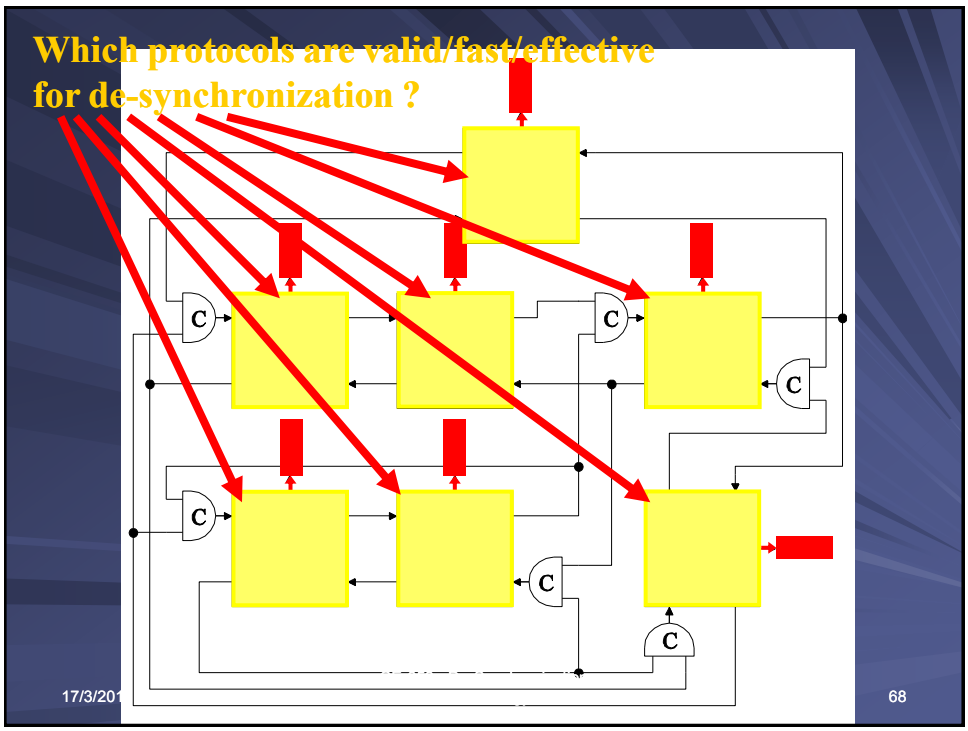
(semi-decoupled 4-phase protocol)

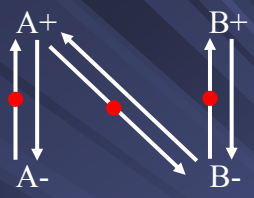


(semi-decoupled) 4-phase protocol)




F
1
A
0
B
1
E
0
CLK
0
C
1
D
1
G


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



Theorem:

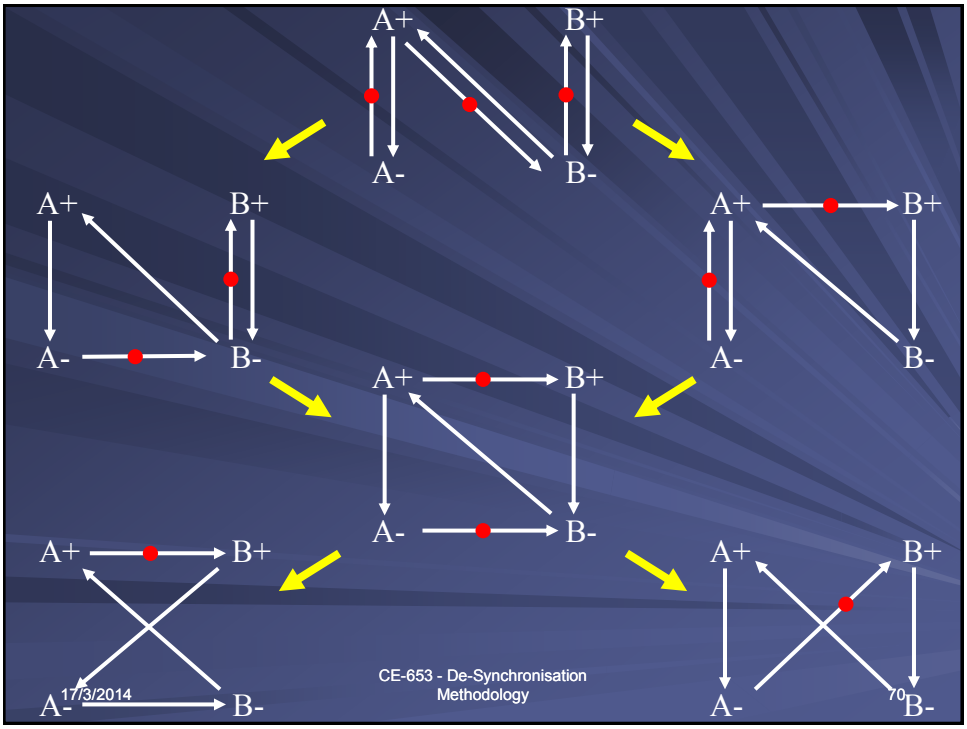
the de-synchronization protocol preserves flow-equivalence

Proof: by induction on the length of the traces

Induction hypothesis: same latch values at reset

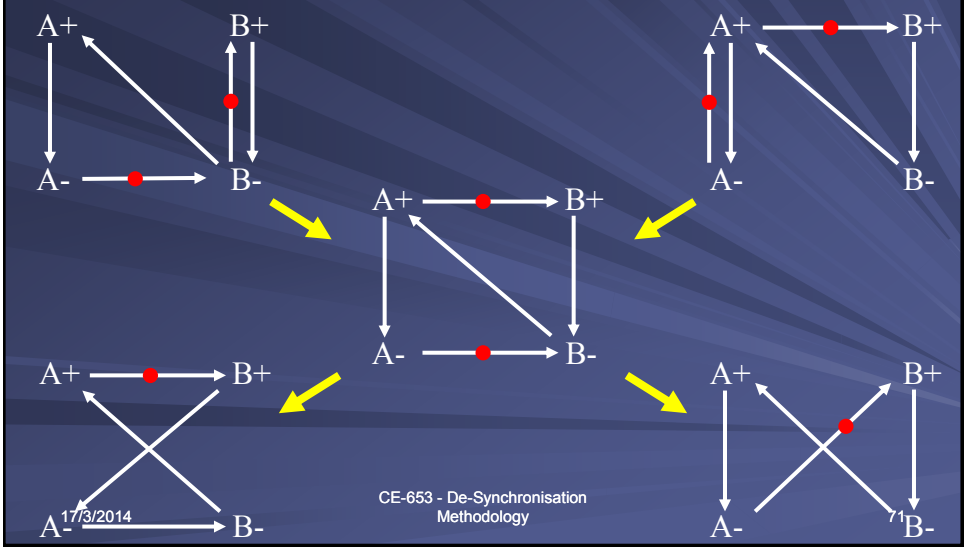
Induction step:

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

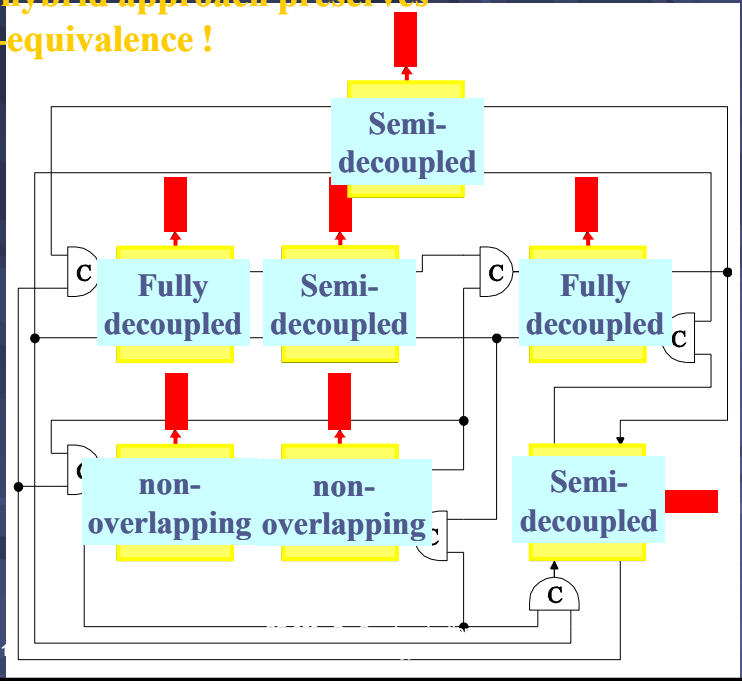


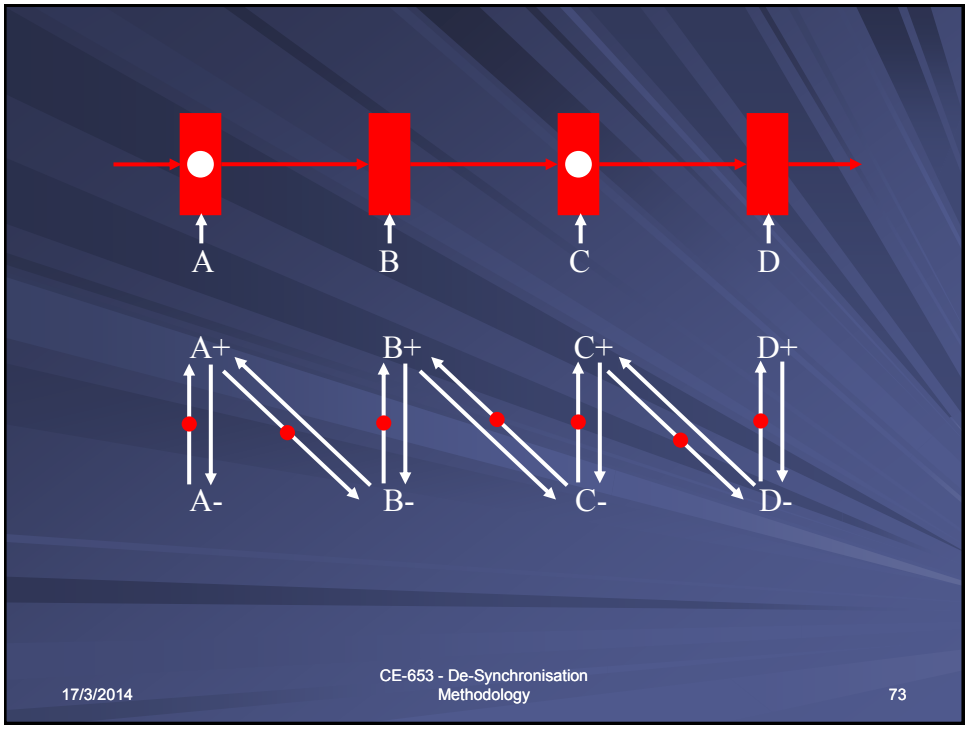
Theorem:

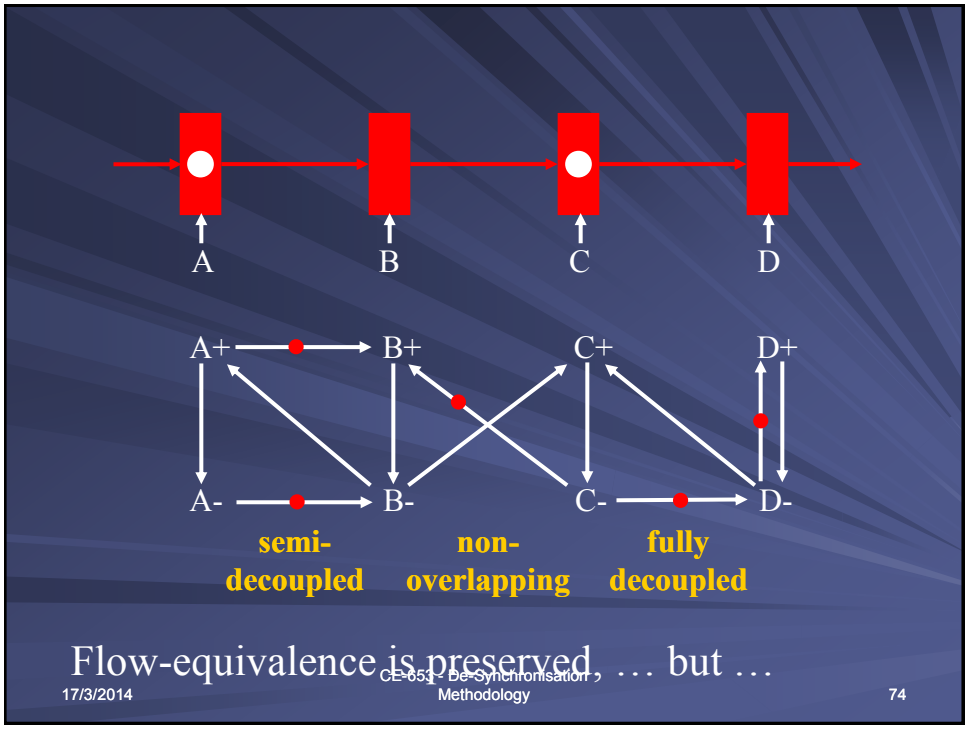
any reduction in concurrency preserves flow-equivalence



Any hybrid approach preserves flow-equivalence !







Flow-equivalence is preserved, … but …

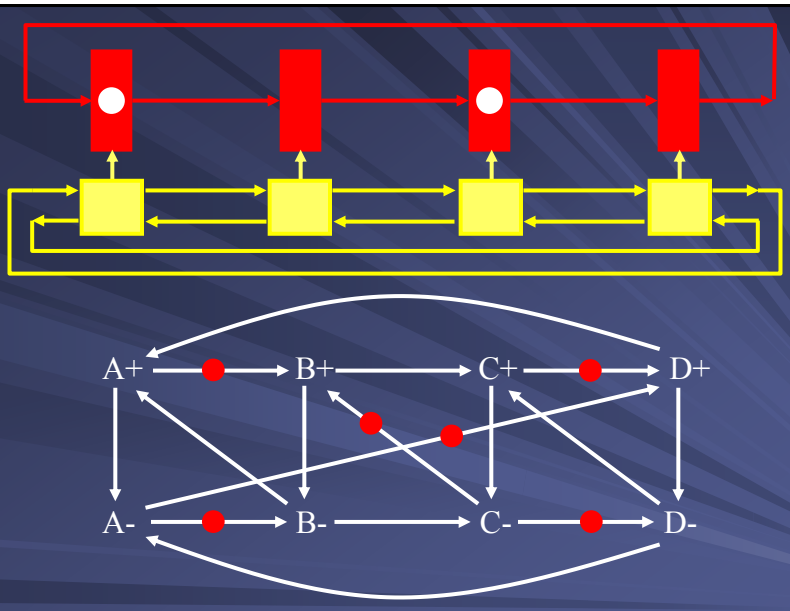
Live-ness?

- Preservation of flow-equivalence:

  all the generated traces are equivalent

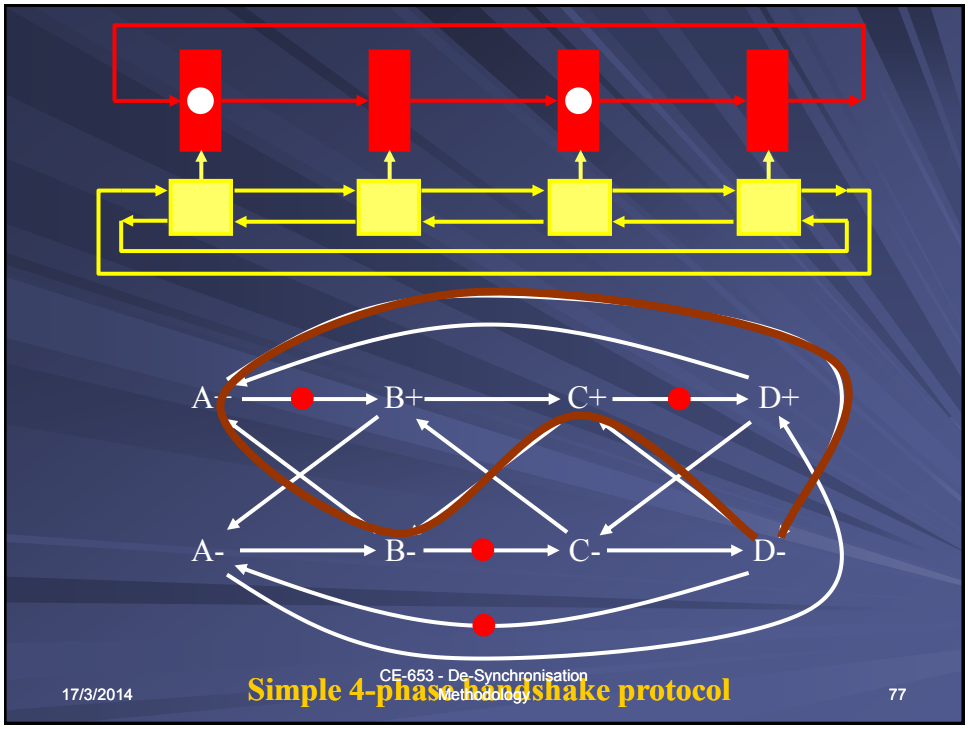
- Are all traces generated ?
  (Is the marked graph live ?)

  Not always !



Semi-decoupled 4-phase handshake protocol

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

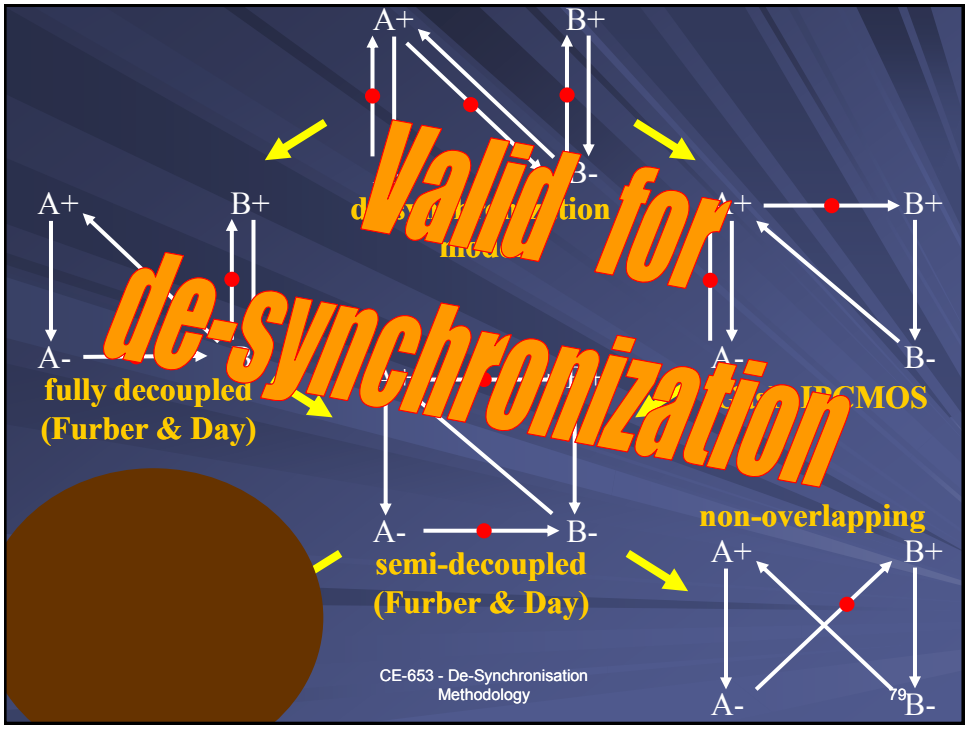


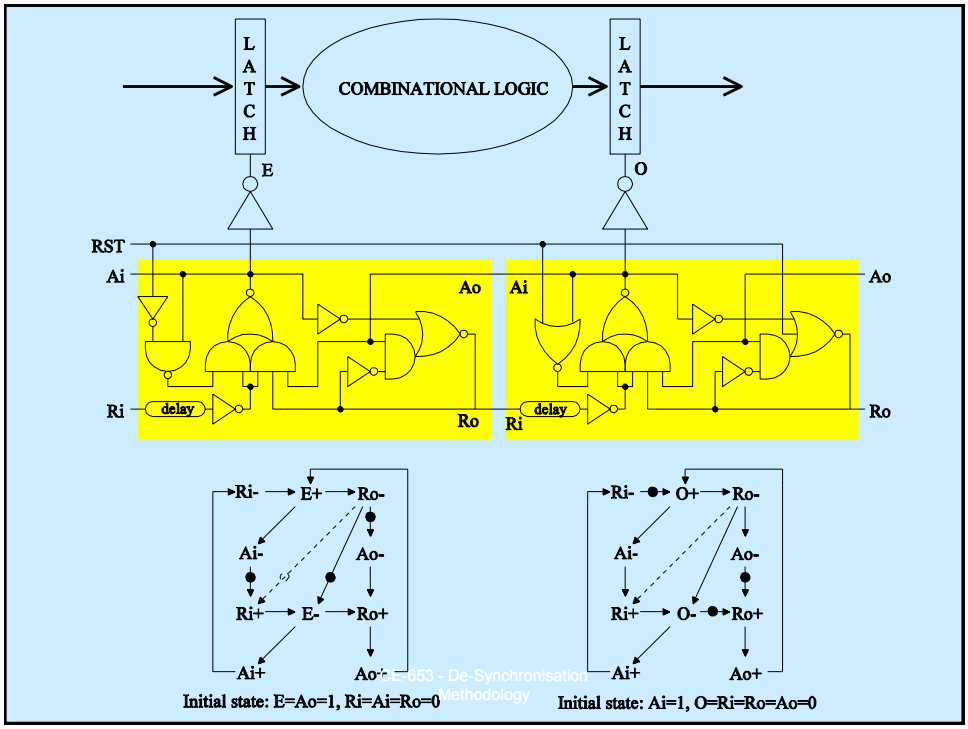
Simple 4-phase handshake protocol

Results regarding Live-ness

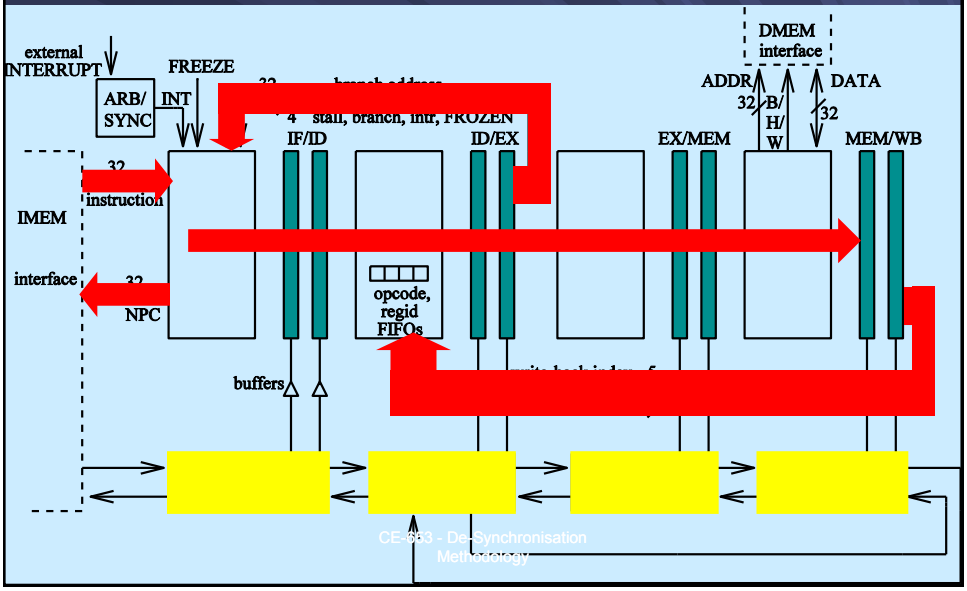
- At least three latches in a ring are required with only one data token circulating [*Muller 1962*]
- **Theorem**:
  any hybrid combination of protocols is live if the simple 4-phase protocol is not used

  Proof: *any cycle has at least one token*


A+
B+
A-
B-
Valid for
de-synchronization
fully decoupled
(Furber & Day)
semi-decoupled
(Furber & Day)
non-overlapping
CE-653 - De-Synchronisation Methodology
79


LATCH
COMBINATIONAL LOGIC
LATCH
E
O
RST
Ai
Ao
Ri
Ro
delay
Ri-
E+
Ro-
Ai-
Ao-
Ri+
E-
Ro+
Ai+
Ao+
Initial state: E=Ao=1, Ri=Ai=Ro=0
O+
O-
Initial state: Ai=1, O=Ri=Ro=Ao=0
CE-653 - De-Synchronisation Methodology

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

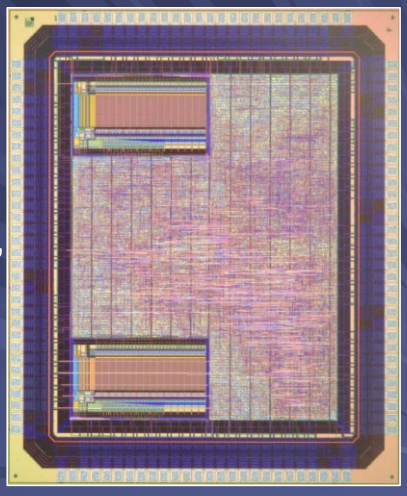
http://www.ics.forth.gr/carv/async/demo/

De-synchronized DLX on FPGA

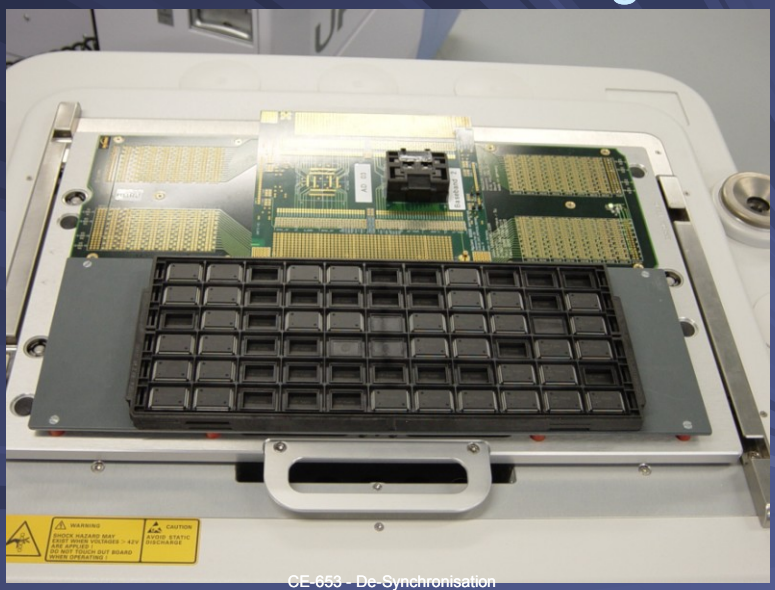
ASPIDA ASIC Design

ASPIDA IC

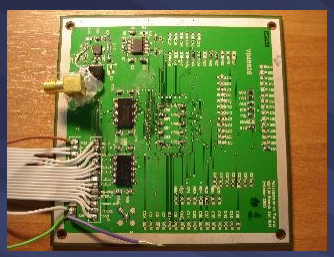
- 32-bit RISC CPU
- EU funded research project
- Two fully-functional ICs were manufactured with IHP 0.25um technology
- Runs in both *synchronous* and *de-synchronized* modes,
  - Direct comparison of results
- Measurements
  - Performance, Voltage scaling and EME measurements
  - On-tester functional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



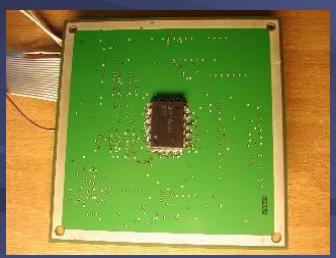
ASPIDA IC Testing



ASPIDA PCB Design



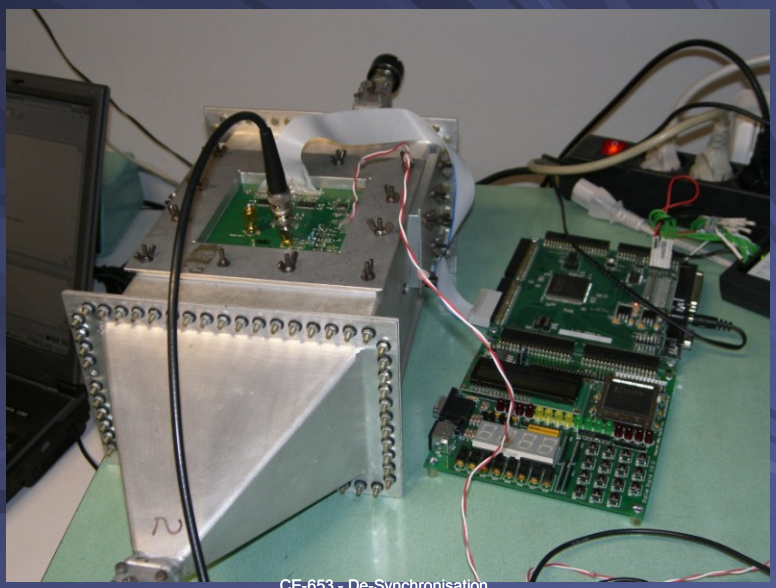
PCB with RISC IC (back)



PCB with RISC IC (front)

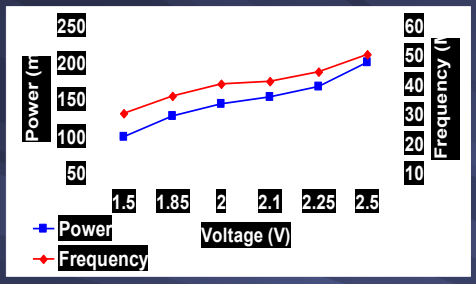
- 32-bit RISC CPU with DFV and adaptive (P, V, T) timing
- DFV embedded automatically to RISC CPU Verilog netlist using NanoSync V0 tool
- 700K Transistor Design, 0.25um CMOS process
- Full-scan testable, Adaptive Timing operation
- DFV Voltage Scaling from 3.3V down to 0.95V (2.5V process nominal)
-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

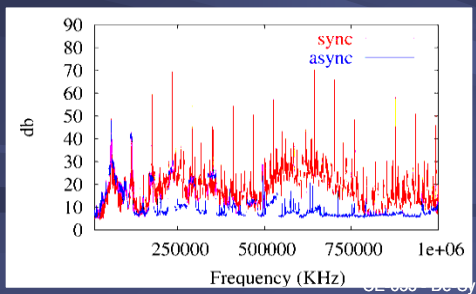
ASPIDA TEM Cell Measurements



ASPIDA Results

ASPIDA Measurements



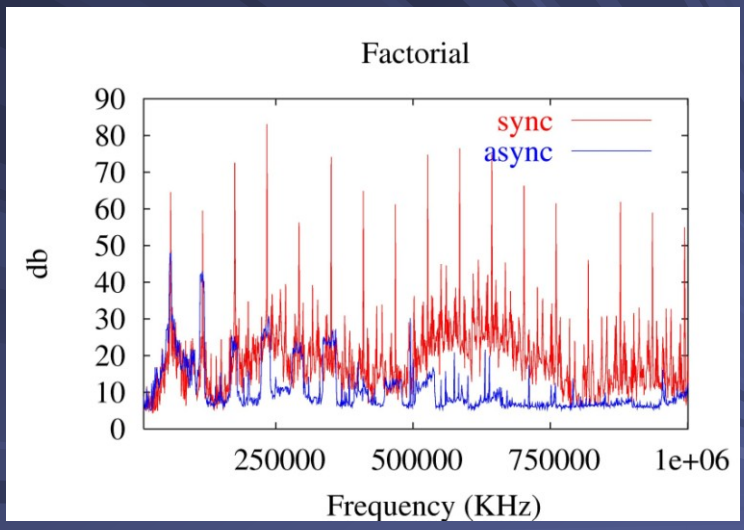


- Performance results
  - Synchronous: 52 MHz
  - De-Synchronized: all chips worked above 63 MHz
  - ~20% Lower-power through Voltage Scaling!
- Power results
  - Scaling: 200 mW (50 MHz) dropped to 98 mW (30 MHz)
  - Chips could scale down to 0.95 V (250 KHz), well beyond allowed voltage
- EME reduction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

ASPIDA Schmoo Plot

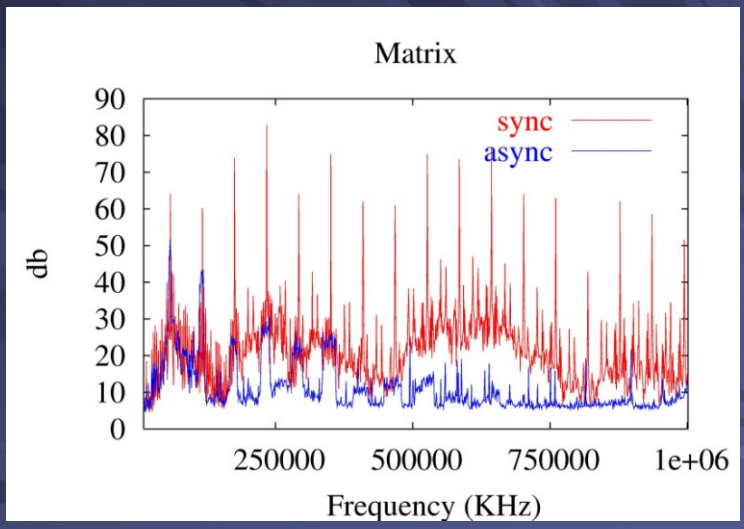
- Period vs. VDD for desynchronized operation
- Fully functional on any voltage above 0.95 V

ASPIDA EME Results



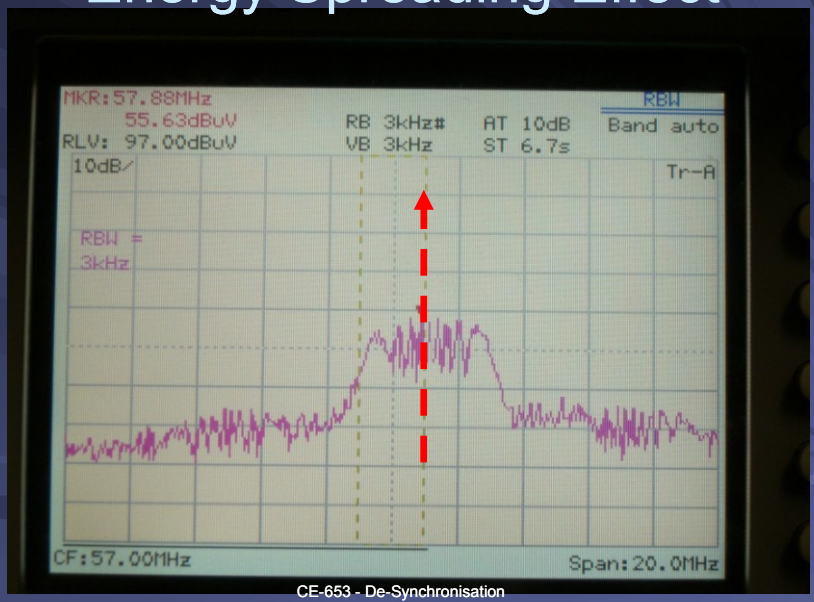
- ASPIDA running Factorial on TEM

ASPIDA EME Results



- ASPIDA running Matrix on TEM

Energy Spreading Effect



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