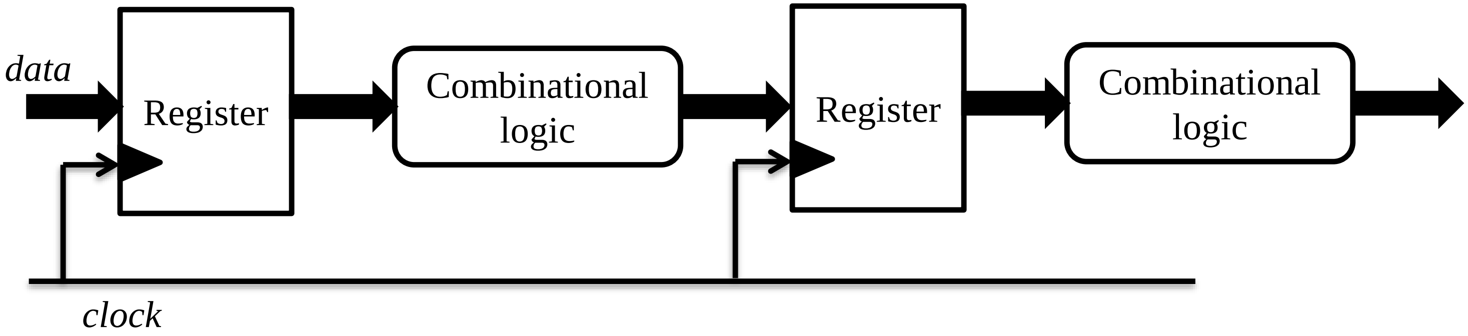
For the Asynchronologists

L. FESQUET, Y. DECOUDU, R. IGA, R. ACQUINO GUAZZELLI

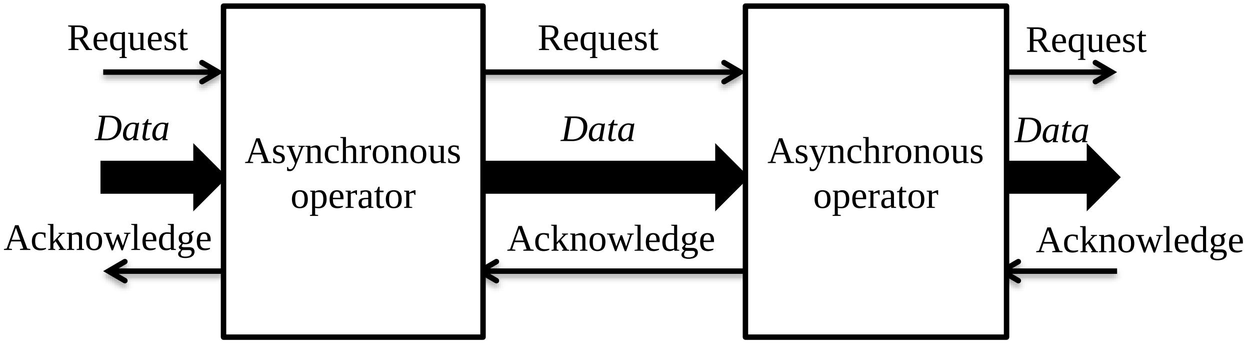
Circuit , Device and System Integration Group
TIMA Laboratory

Digital Microelectronic Project, 03/27/2020

Thinking and designing differently



Synchronous – Global synchronization

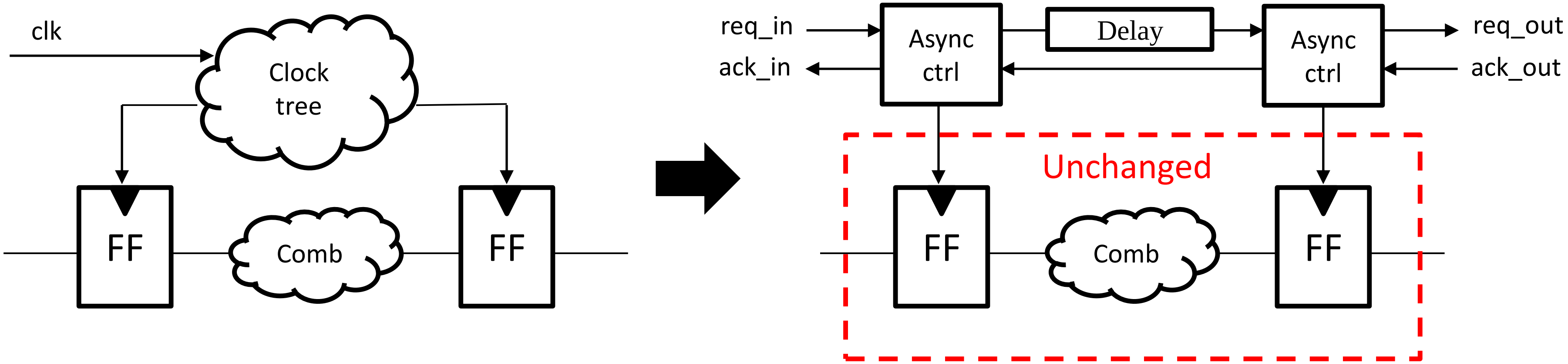


Asynchronous – Local synchronization

- Modularity: “plug-and-play”
- Local worst case approach for the delays
- Low power
- Electro-magnetic compatibility
- Robustness
- Security

Bundle-Data Circuits

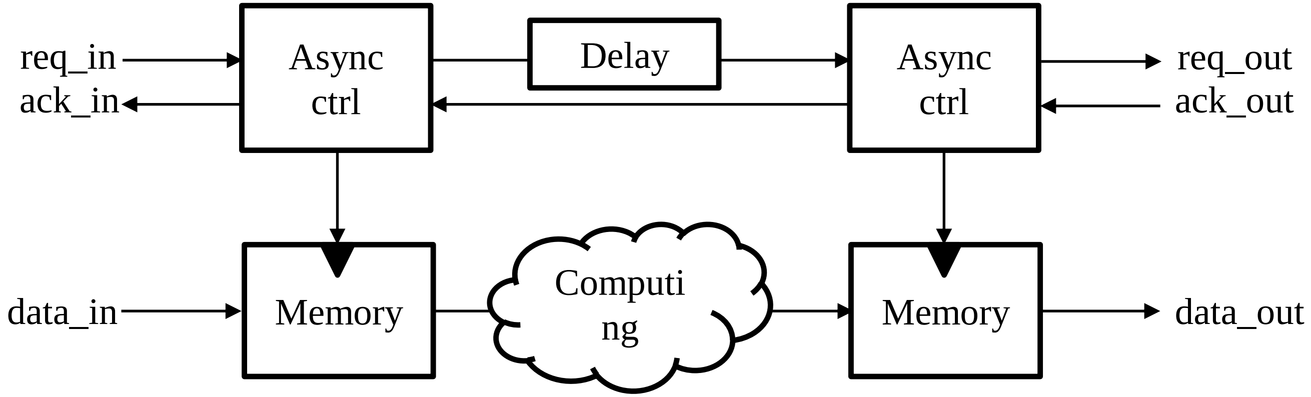
- **Structural difference between synchronous circuits and Bundle-Data Circuits**



Extraction of the clock tree and replacement with an asynchronous control circuit

Targetted circuit model

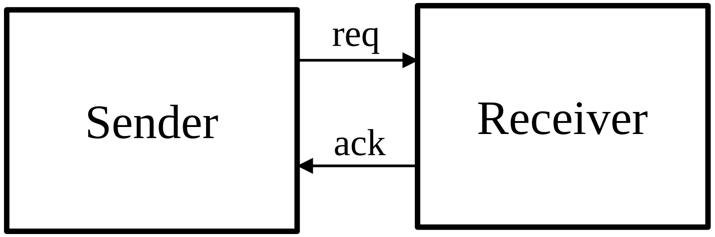
- **Micropipeline = asynchronous circuit class with bounded delays in gates and negligible delays in wires**
  - Self-timed, event-driven, elastic pipelines proposed first by I. Sutherland in [Sut89]
  - Linear / non linear



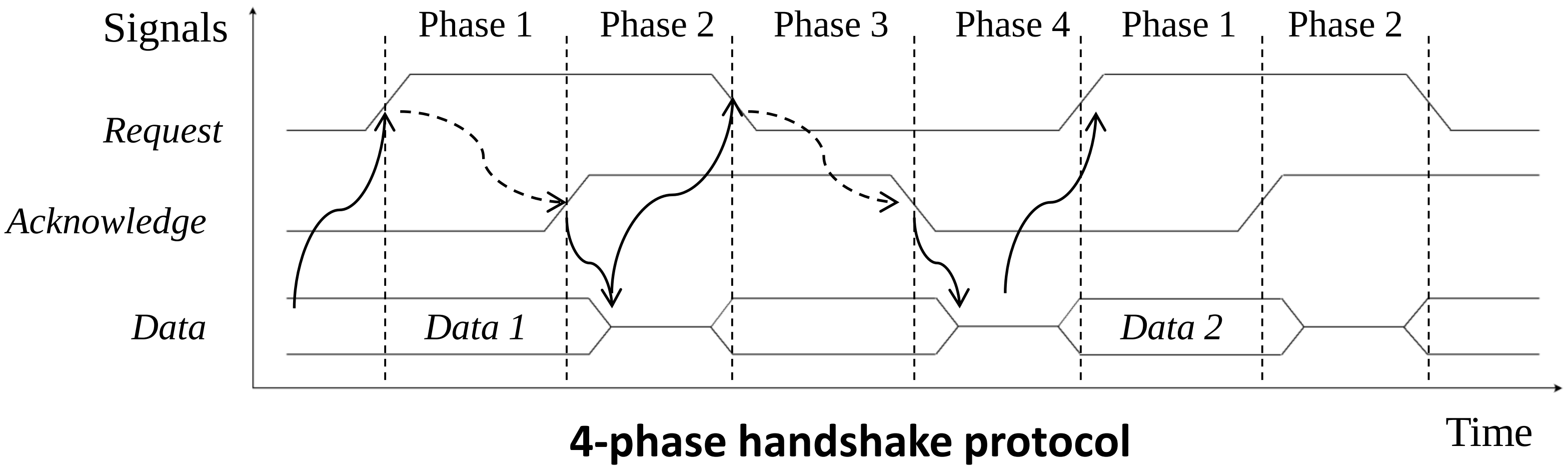
- **Bundled data and handshake communication protocol**
  - Data bus + 2 wires for requests and acknowledgments
  - Four-phase handshake

[Sut89] I. E. Sutherland, Micropipelines, in Communications of the ACM (Association of Computing Machinery), Vol/Issue:32/6, pages 720738, 1989.

Communication protocol



1) Sender activates *req* with valid data
2) Receiver detects *req=1* and activates *ack*
3) Sender detects *ack=1*, invalidates data and inactivates *req*
4) Receiver detects *req=0* and inactivates *ack*

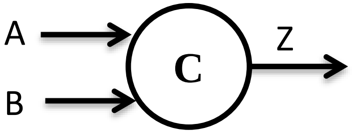


4-phase handshake protocol

Storage, control and synchronization

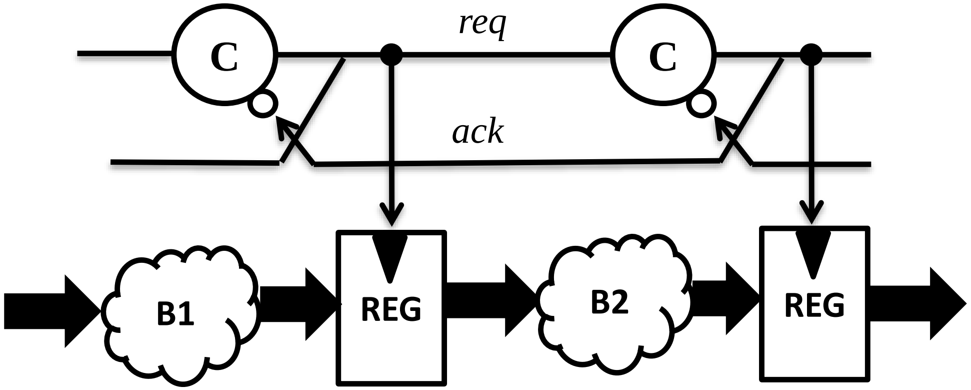
- **Storage:** Flip-flops or master/slave latches

- **Synchronization:** Muller gate ("Rendez-vous" element)



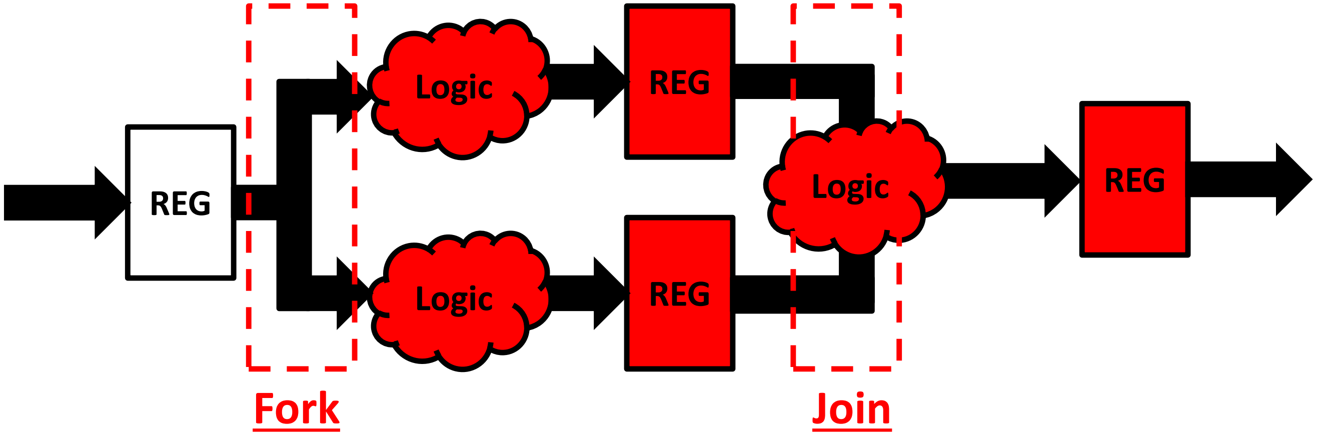
F	R	Z
0	0	*0*
0	1	*Memory*
1	0	*Memory*
1	1	*1*

- **Control:** Muller gate with an inverted input

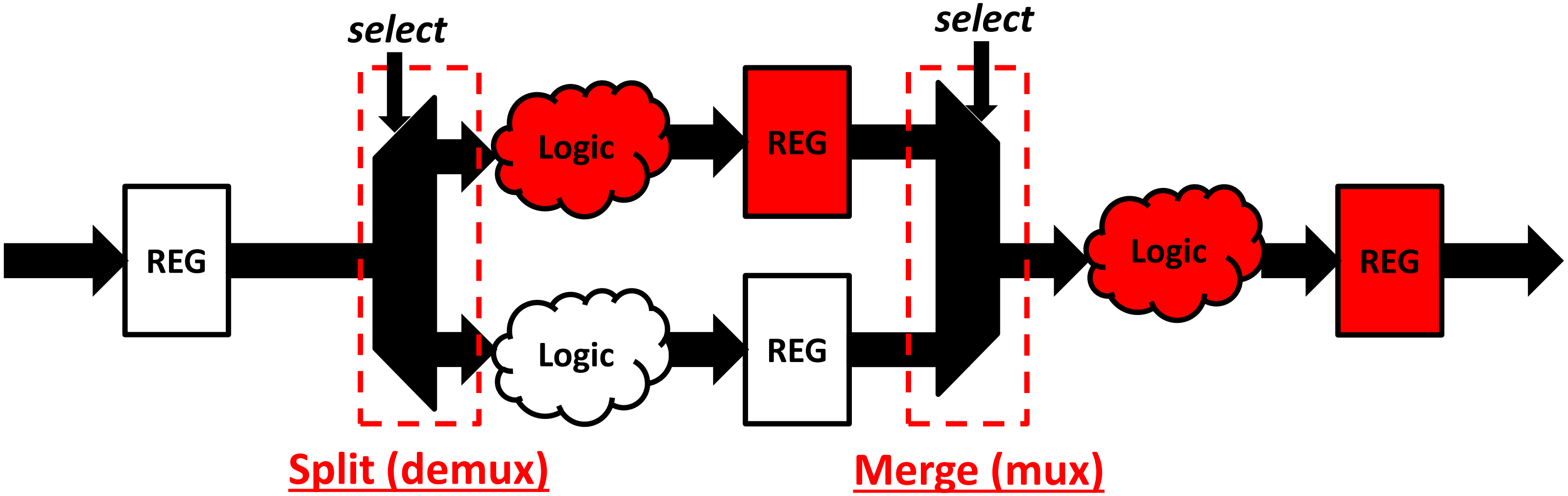


Non linear micropipelines

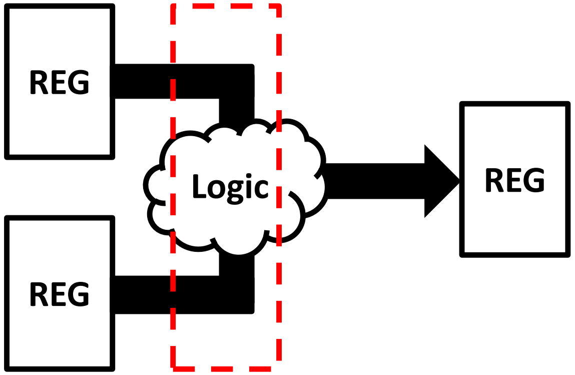
- “Full” synchronization components



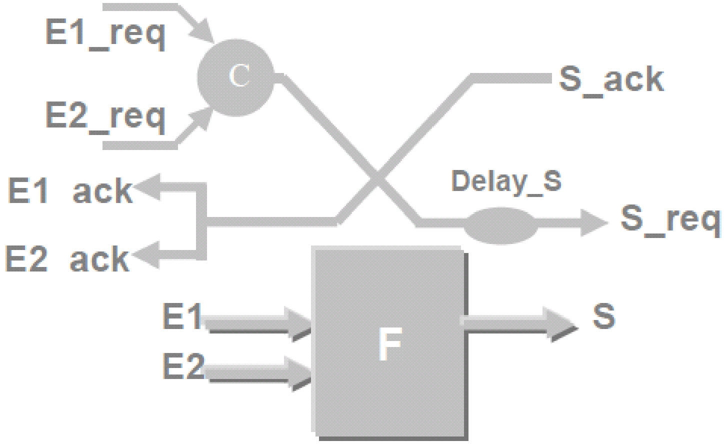
- “Partial” synchronization components



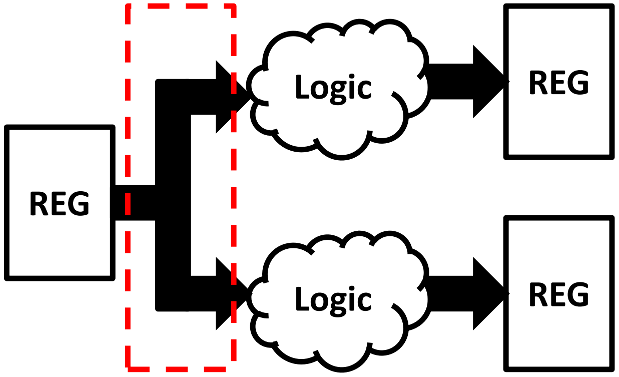
Join and Fork components



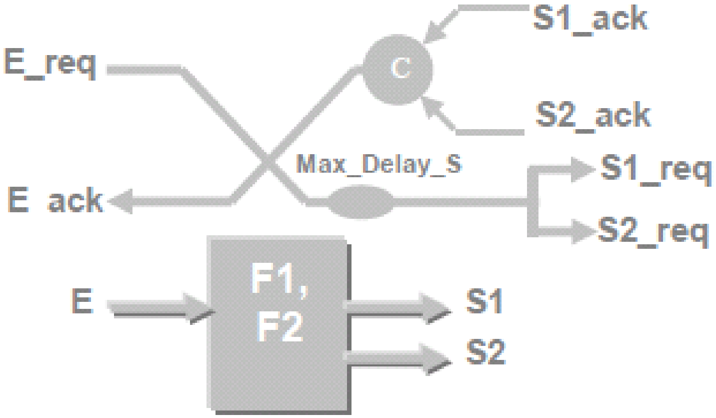
Join



Join control implementation

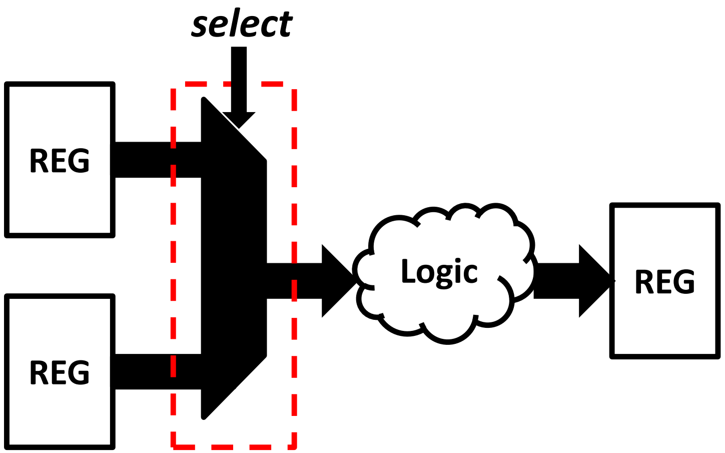


Fork

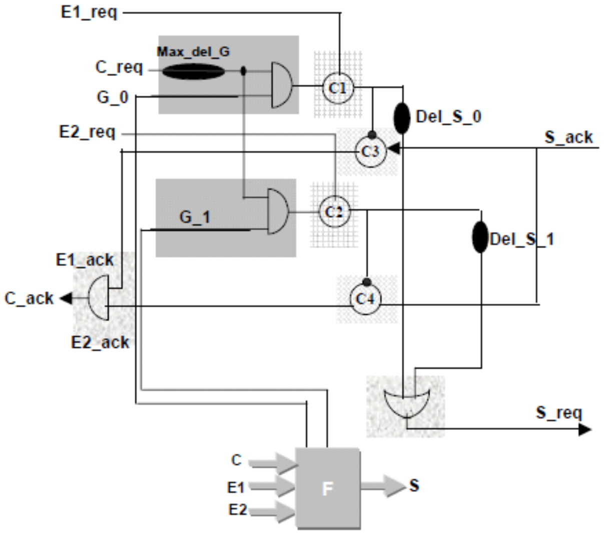


Fork control implementation

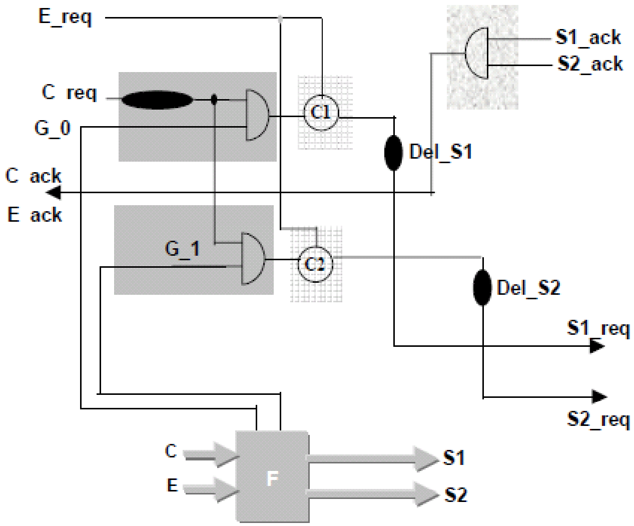
Merge and Split components



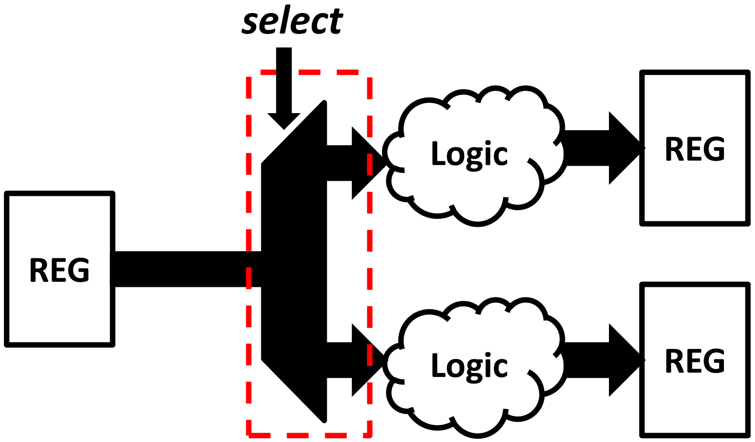
Merge (multiplexer)



Merge control implementation

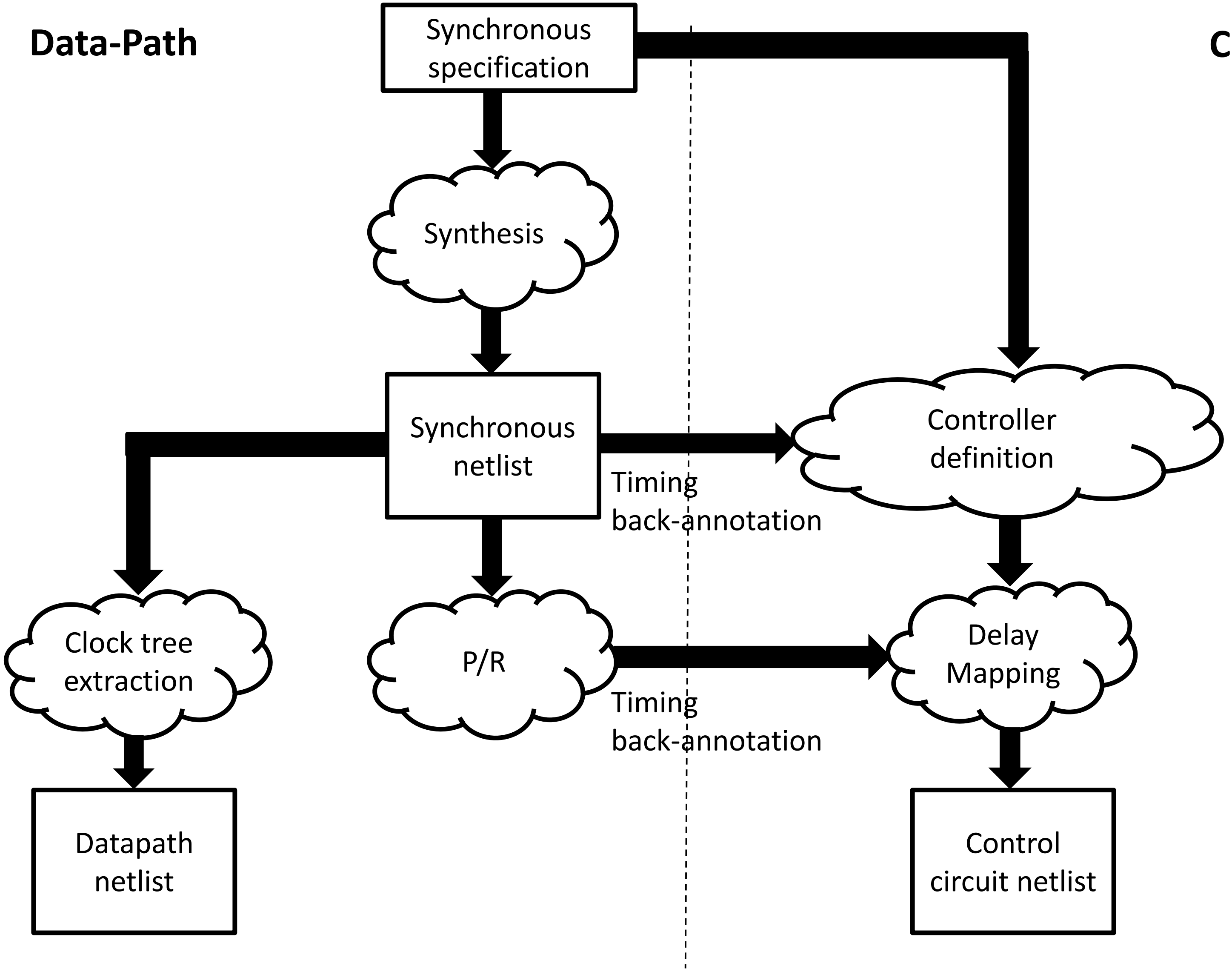


Split control implementation



Split (demultiplexer)

Design flow



Data-Path

Control-Path

Synchronous specification

Synthesis

Synchronous netlist

Controller definition

Timing back-annotation

Clock tree extraction

P/R

Delay Mapping

Timing back-annotation

Datapath netlist

Control circuit netlist