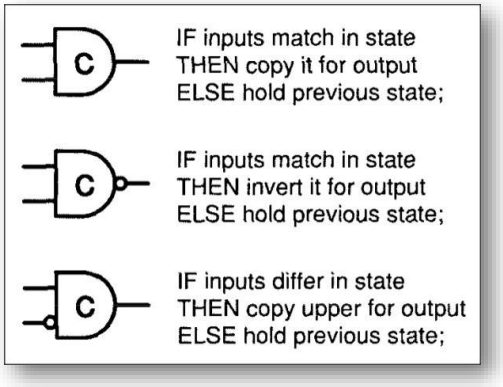
CE653 – Asynchronous Circuit Design

Instructor: C. Sotiriou

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

Contents

The C Element

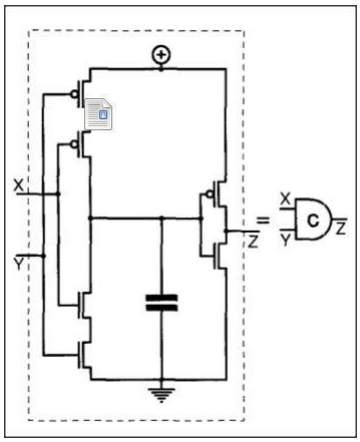


- AND gate for events

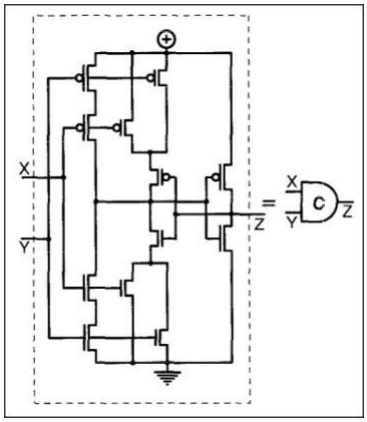
a	b	c
0	0	0
0	1	c′
1	0	c′
1	1	1

- What is the OR gate equivalent for events?

C-Element Implementations



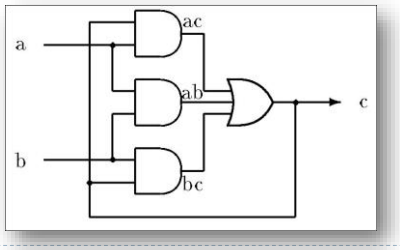
Dynamic C-Element

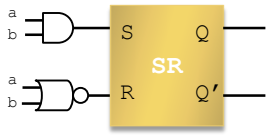


Static C-Element

C-Element Implementations

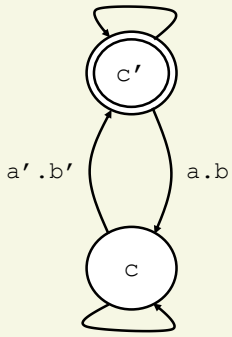
- Using Standard Cells
  - 2-input C-Element's Logic Function is:
    $c = ab + c(a + b) = ab + bc + ac$
  - n-input C-Element is:
    $c = a_1a_2...a_n + c(a_1 + a_2 + ... + a_n)$
- Thus may be implemented using two level logic or using an SR latch

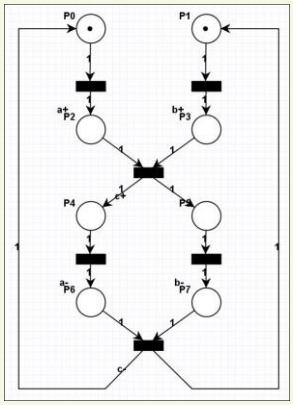




C Gate Formal Models

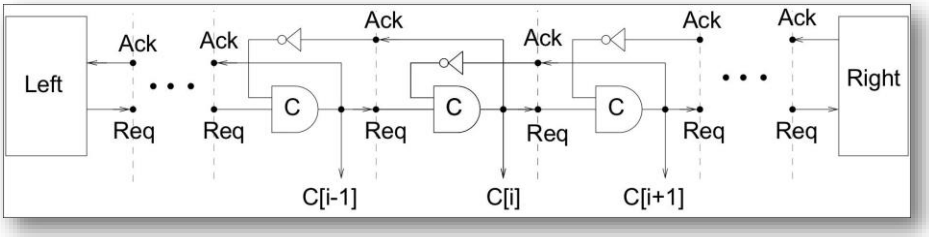
FSM	PTnet





- *Empty transitions* ➔ else stay in same state

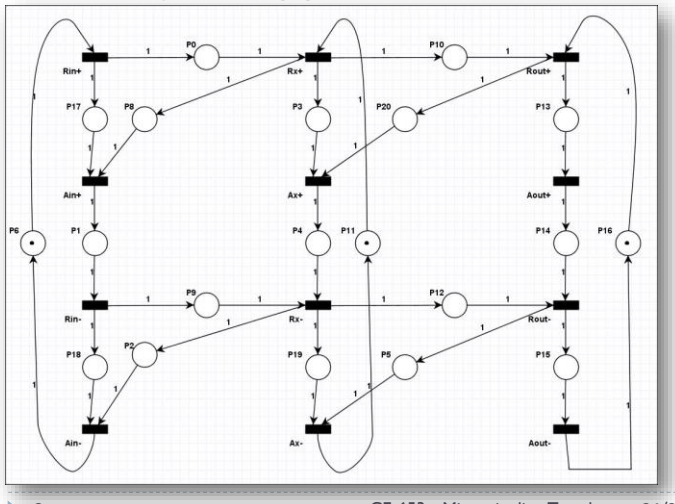
Micropipeline Control Structure

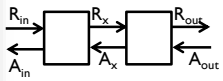


- Req/Ack are handshaking pairs
- C[i] accepts 1/0 from C[i-1] only if C[i+1]=0/1
- Think of 1010101.. as waves: $l_0\ l_0\ l_0\ l_0$..
- The C-elements propagate waves precisely
- Timing depends on local delays, may vary along the pipe
- If RIGHT is quiet, pipe will fill and stall
- Same for 4-phase, 2-phase

Micropipeline Analysis - PTnets

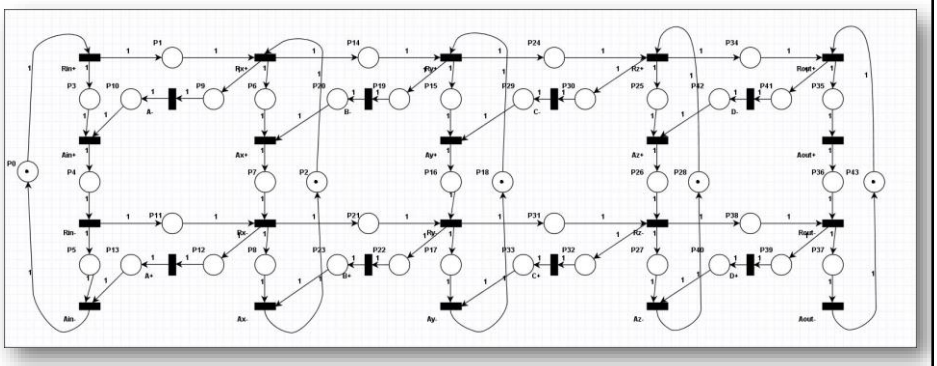
- Two-stage Micropipeline PTnet:





Micropipeline Analysis - PTnets

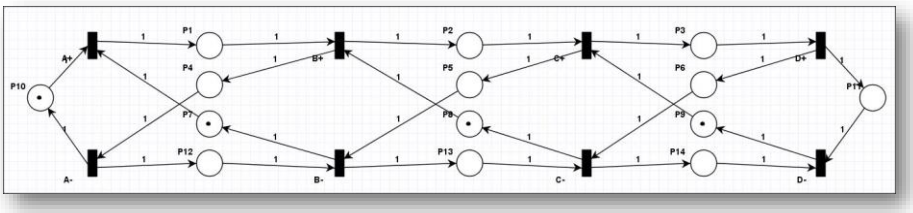
- Four-stage Micropipeline PTnet:



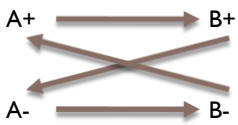
- Represents 4 C Elements and their environments
- ***Correct operation may be verified by possible signal orders***

Micropipeline Analysis - PTnets

- Four-stage Micropipeline Ptnet
  - Reduction to Latch Control Signals



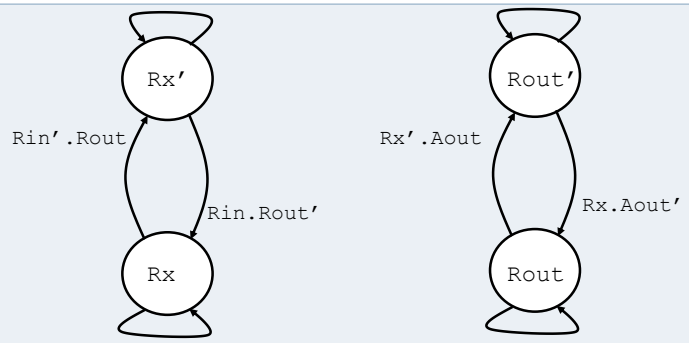
- Characteristic Pattern:



Micropipeline Analysis - MSFSMs

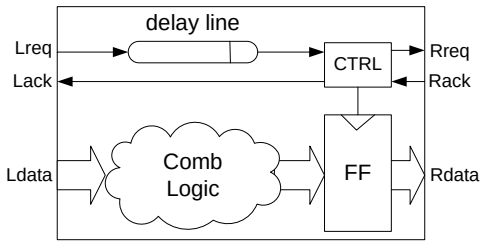
- Two-stage Micropipeline FSMs:

First stage FSM **Second stage FSM**



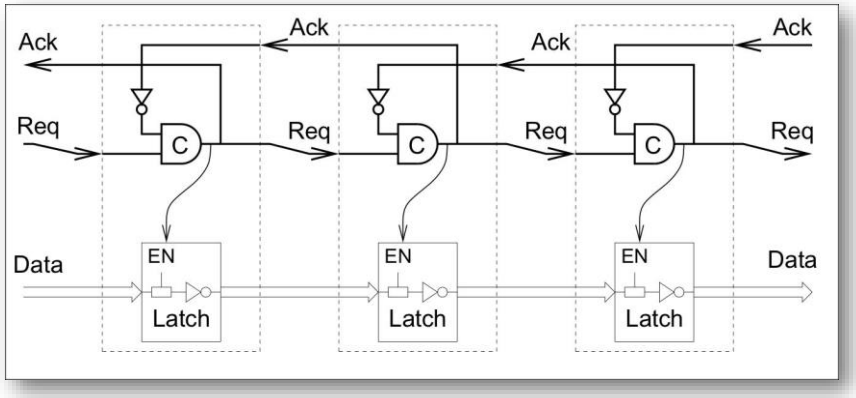
- *What about LHS/RHS FSM Interfaces?*
  - What are the two environment FSMs Rin/Ain, Rout/Aout?

Bundled-Data Concept



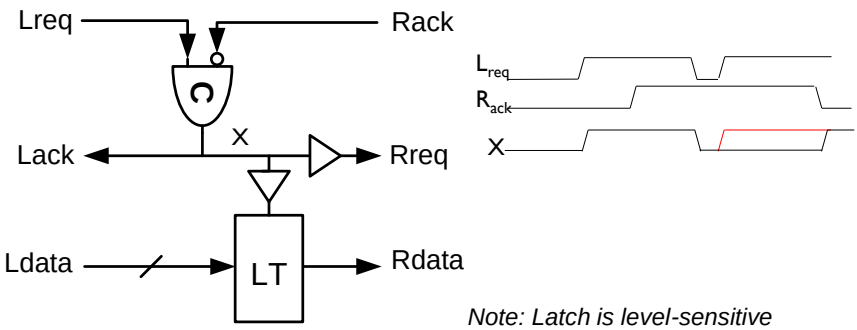
- Static combinational logic (typically) and standard FFs or latches
- Channels use bundled-data protocol
- Delay line matches worst-case delay of combinational logic
  - Margin limits performance, particularly because adds to forward latency
- Controller CTRL drives local clock to bank of FFs (or latches)
  - Designed using known templates or burst-mode controllers, signal-transition graphs, or syntax directed translation

Four-Phase Micropipeline



- Latches are active high
- Four-phase, so latches open on first ½ of h/s and close on second ½

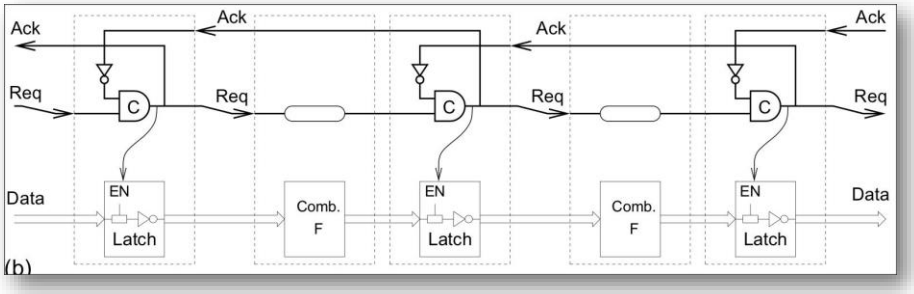
Four-Phase Micropipeline Stage



Note: Latch is level-sensitive

- Four-phase protocol on Lreq/Lack and Rreq/Rack
  - Latch is simple level sensitive traditional latch
  - This is a half-buffer!

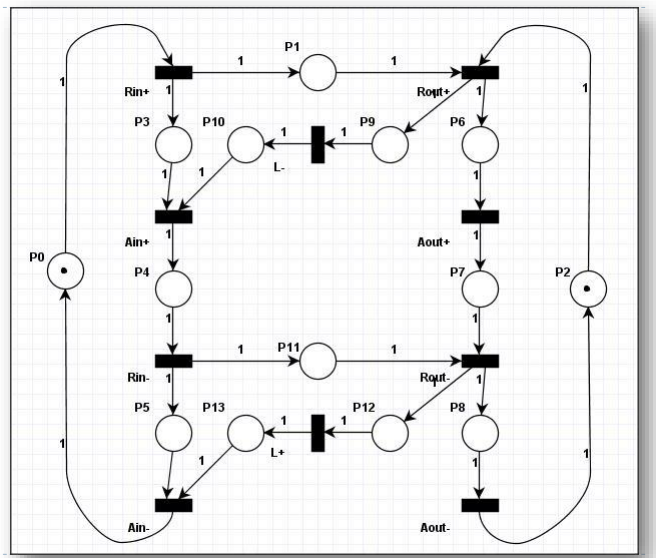
Four-Phase Micropipeline with C.L.



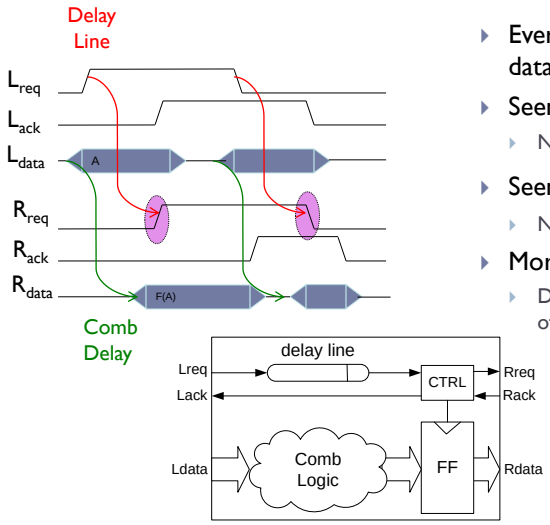
- Delay elements mimic worst-case critical path of Combinational Logic F of each pipe stage

Micropipeline 4-Phase PTnet including Latch Enable

- L
  - **Active-high** in PTnet
- L-
  - Closes Latch
- L+
  - Opens Latch

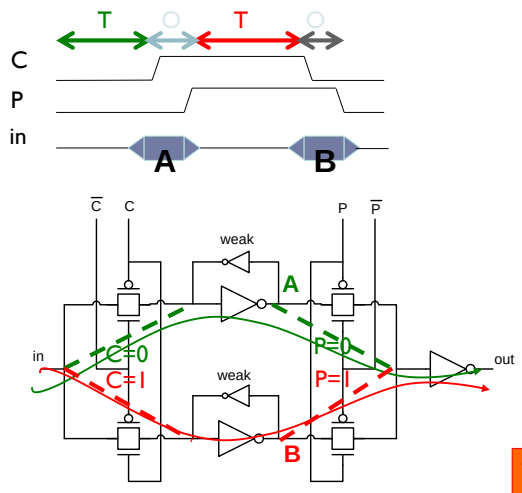


Two Phase Signaling



- Every Toggle on request => new data
- Seems to be faster (NRZ)
  - Not in general
- Seems to be low power
  - Not usually the case
- More complex Control
  - Data Validity coded in phase difference of Req & Ack

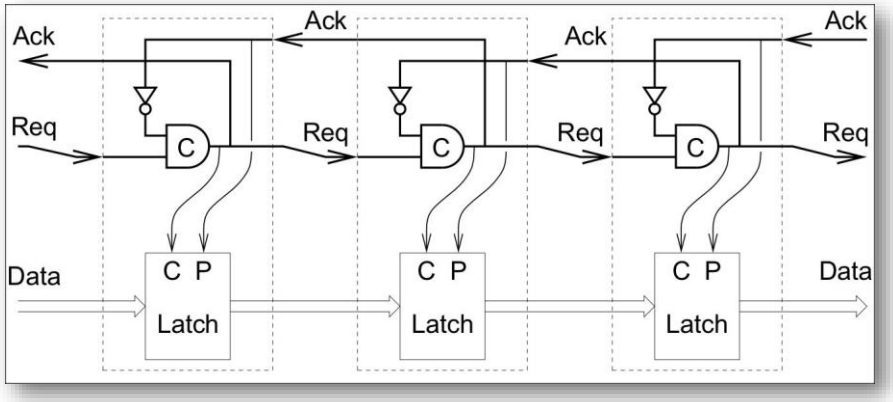
Capture-Pass Data Latch



- Two-phase data-latch
  - Become transparent when a Pass "P" event occurs
  - Become opaque when a Capture "C" event occurs
  - Store data alternatively in top and bottom inverter loops

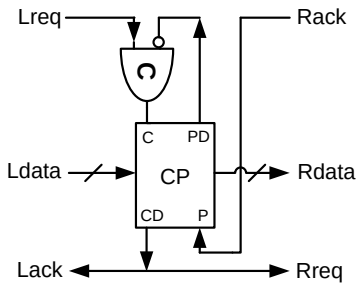
Too many Transistors

Two-Phase Micropipeline



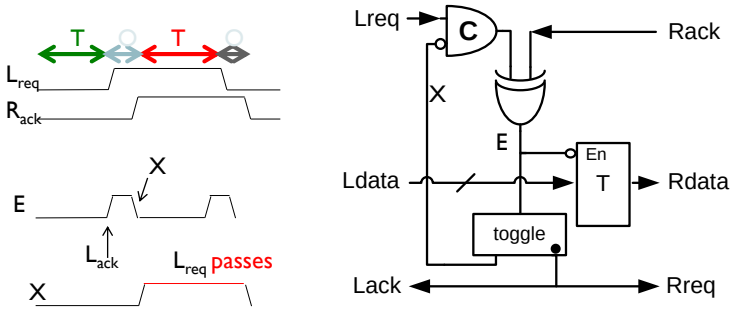
- Simple version without Cd (C done), Pd (P done)
  - Little difference anyway in generating fake done signals (using delay elements)

Two-phase Micropipeline Stage – with CP Latches



- Store data upon Lreq event when last Pass event is done
  - Ensure new store data only when old data is no longer needed
  - Identified by right hand side changing Rack indicating does not need data
    - Transition of Rack causes CP latch to go transparent
- Note two-phase protocol on Lreq/Lack and Rack/Req
  - Full-buffer handshaking!

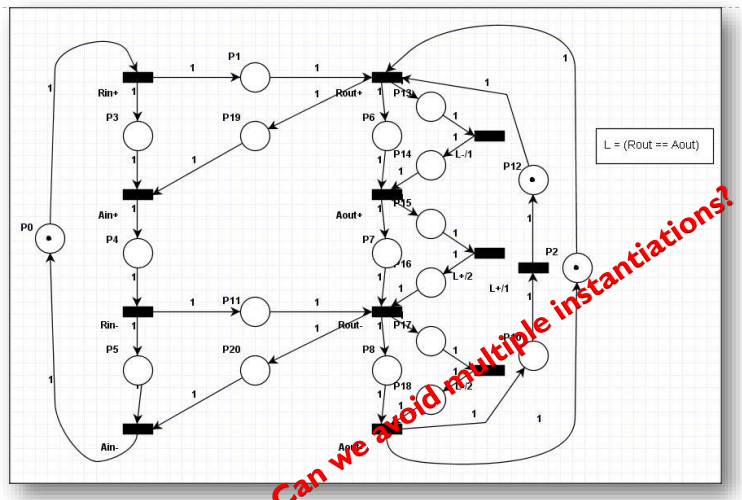
Two-phase Micropipeline Stage – with Transparent Latches



- Transparent Latches
  - Much smaller than CP latch
  - Level sensitive
  - Thus, we need to convert two-phase control to four-phase
- XOR acts as a two-phase to four-phase converter
- Toggle acts as a four-phase to two-phase converter

Overhead/bit?!

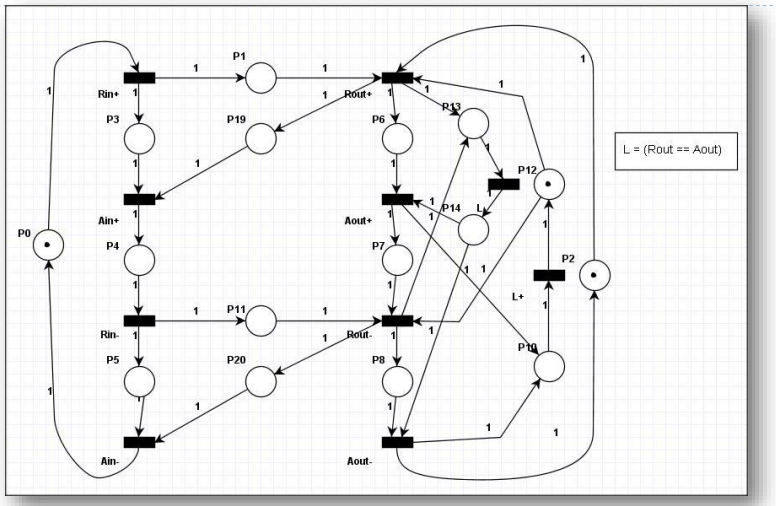
Micropipeline 2-Phase PTnet including Latch Enable



Can we avoid multiple instantiations?

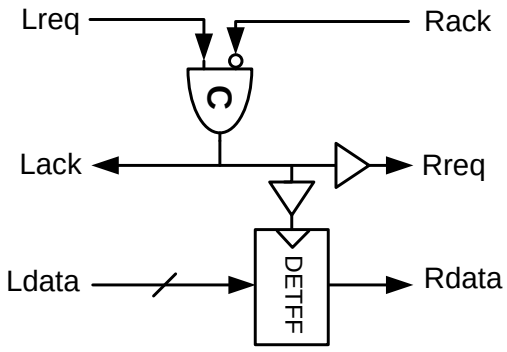
- Multiple instantiations of `L` signal (`L`/1, 2)

Micropipeline 2-Phase PTnet including Latch Enable and Single Latch Signal



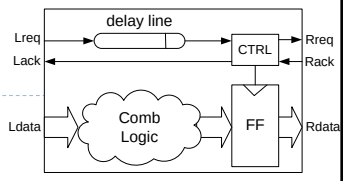
- Note the AC Return from Choice/Choice Needed

Two-phase Micropipeline Stage – with Double-Edge-Triggered FFs

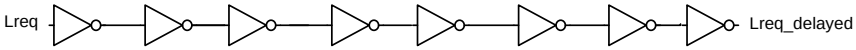


- Efficient Double-Edge-Triggered FFs can also be used
- Buffer to FFs can be removed avoiding delay overhead

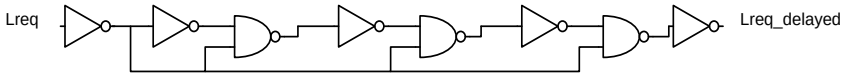
Delay Line Design



- Symmetric
  - Designed with an even # of inverters that model worst-case path delay through combinational logic

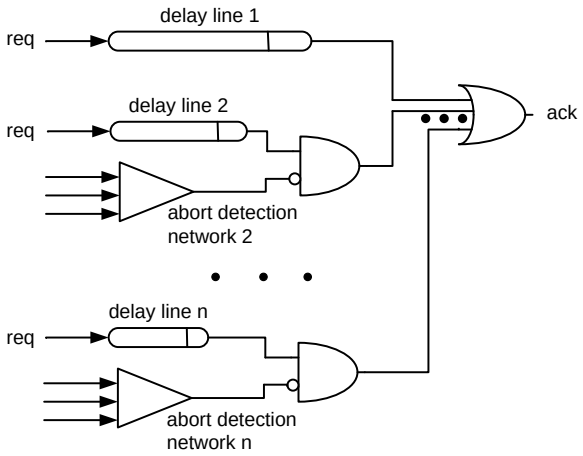


- Asymmetric
  - Faster reset implemented by replacing some INVs with NANDs



- Other
  - May also use re-create path of gates along critical path to better match delay

Speculative Completion Sensing/Variable Latency



- *Enables data-dependent component delays in bundled-data environment*

Micropipeline Pitfalls