June 2 schedule

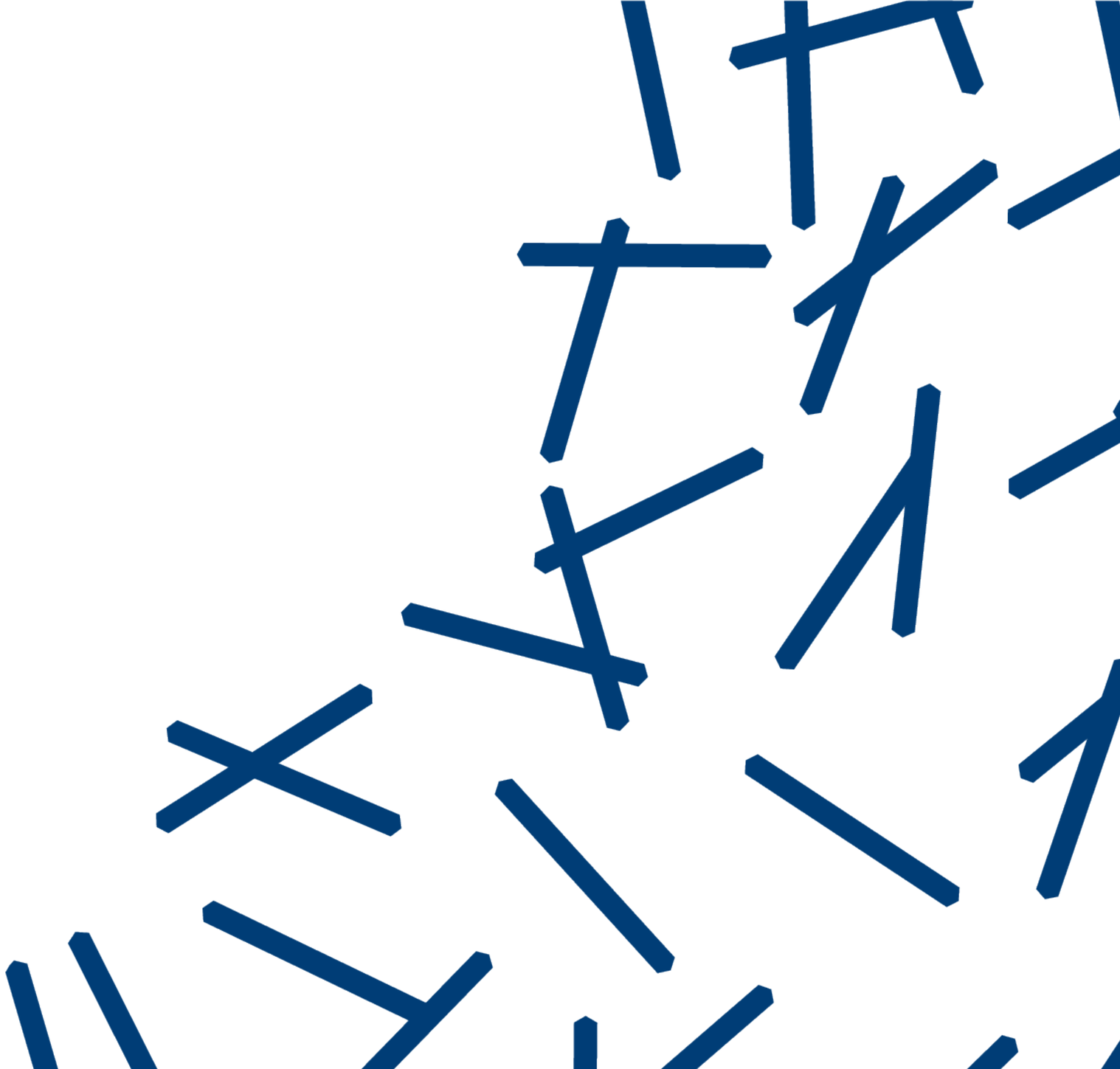
9:00 AM	Timing constraints	Rajit Manohar
9:30 AM	ASIC implementation flow I	Ole Richter
10:45 AM	Coffee break	
11:00 AM	ASIC implementation flow II	Rajit Manohar
12:10 PM	Lunch break	
1:10 PM	Petri net primer	Alex Yakovlev
2:00 PM	Controller design using Petri nets	Alex Yakovlev
3:00 PM	Coffee break	
3:30 PM	Custom circuit implementation	Rajit Manohar

Yale ENGINEERING

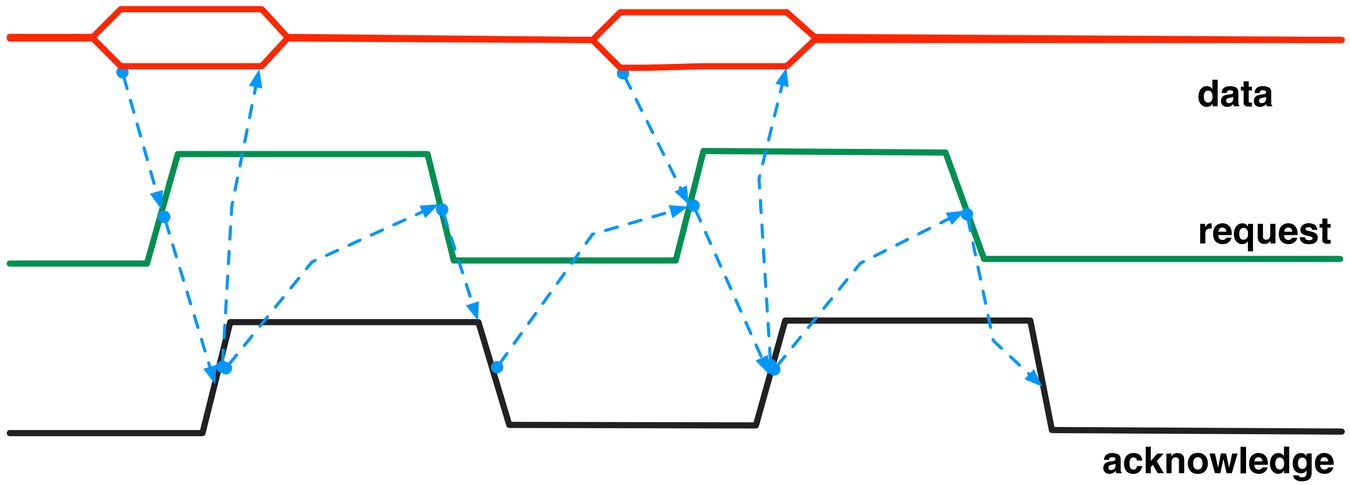
Timing

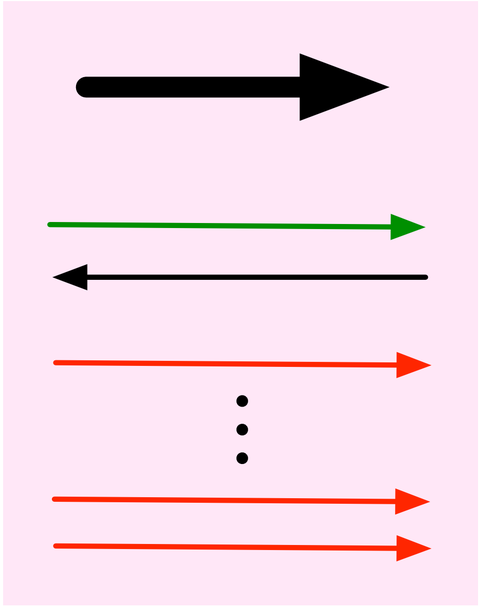
Rajit Manohar
Computer Systems Lab
Yale University

AVLSI



Encoding data: bundled data communication





- Protocol on request/acknowledge protocol can be ***any*** *of the ones seen earlier!*
  - Two wires (or one) for the control
  - N data wires for N-bit data communication
  - **Timing** requirement (“bundled data timing requirement”)

Why different circuit families?

DI control + datapath

No timing constraints!

QDI control + datapath

Minimal timing constraints, robust operation

QDI control + bundled-data datapath

Only bundled-data timing constraint

Click elements

Repurpose clocked standard-cell library, flip-flop based

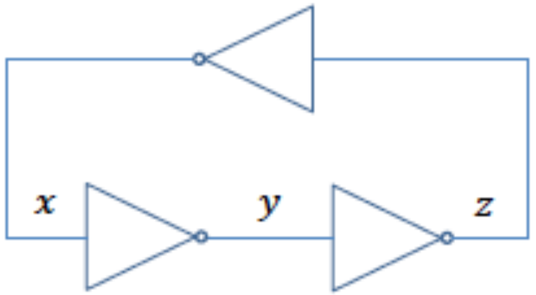
GasP pipeline

high-speed pipeline, single track handshake

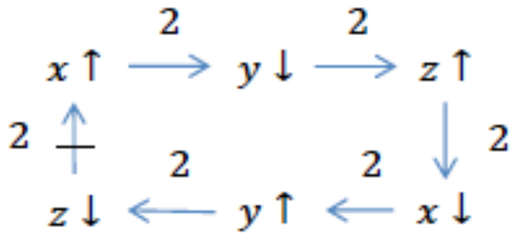
MOUSETRAP pipeline

high-speed pipeline

Static timing analysis with cycles



Circuit



RER system

$$\langle x\uparrow,0\rangle \xrightarrow{2} \langle y\downarrow,0\rangle \xrightarrow{2} \langle z\uparrow,0\rangle \xrightarrow{2} \langle x\downarrow,0\rangle \xrightarrow{2} \langle y\uparrow,0\rangle \xrightarrow{2} \langle z\downarrow,0\rangle \xrightarrow{2} \langle x\uparrow,1\rangle \ldots$$

- The ***timing simulation*** captures when events occur

$$\hat{t}(e) = \max\{\hat{t}(f) + \alpha | (f, e, \alpha) \in R\}$$

  - ❖ "R" captures dependencies ("rules" or timing arcs)

What can we say about timing?

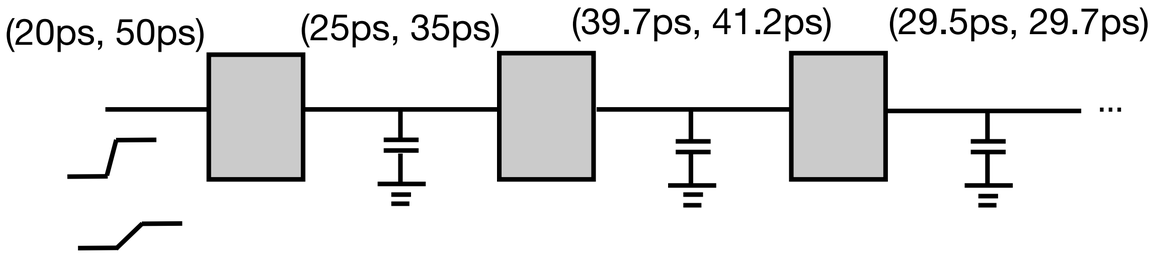
- Theoretical result: there is an M such that for all i ≥ N

$$\hat{t}(x, i+M) - \hat{t}(x, i) = Mp^*$$

critical cycle ratio

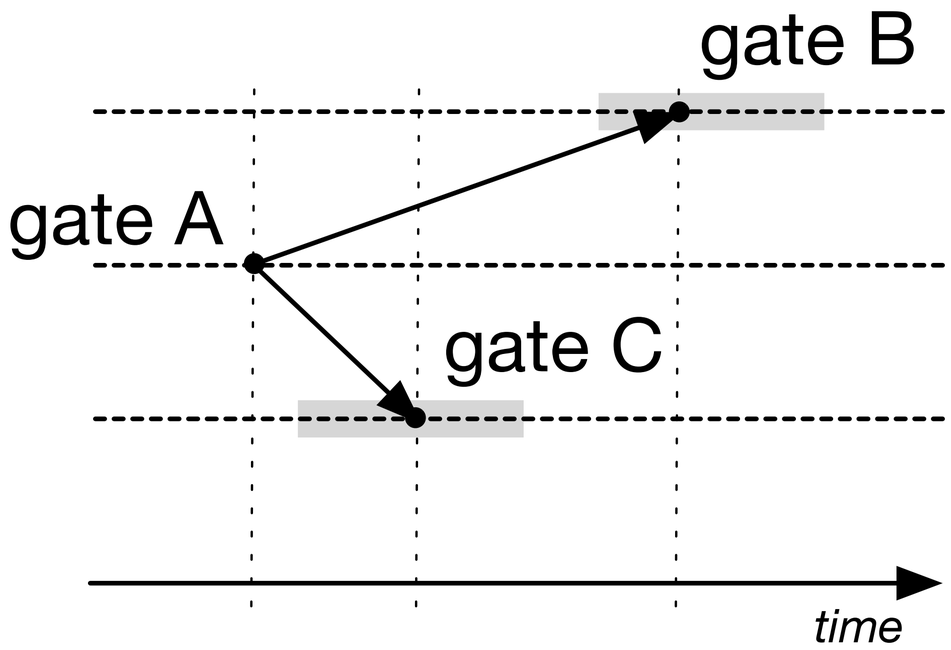
unfolding factor

- How do we calculate delays?
  - ❖ Standard “non-linear delay model”
    - ▸ delay is a function of (load, input slew)
    - ▸ output slew is a function of (load, input slew)
  - ❖ What is the slew rate?



  - ❖ Empirically, 3-4 stages suffices for convergence

Timing constraints in ACT



- **maximum** delay from A to C < **minimum** delay from A to B

```
spec {
   timing a+ : c+ < b-
}
```

```
spec {
   timing a+ : c+ < [4] b-
}
```

Timing constraints in ACT

- Example: bundled-data channel

```
template<pint M>
defchan bd (bool d[M]; bool r, a)
{
  spec {
    /* timing fork */
    timing a- : d* < r*+
  }
  …
}
```

The standard channel definitions in ***`std::channel`*** *include timing constraints, so those will be automatically included.*

`actsim` checks constraints during simulation.

Where do you get the timing for gates?

- Cell library
  - Covers all the gates needed for the design
    - NAND2X1, INVX4, NOR2X2, … etc
  - Any digital circuit component is mapped to a collection of cells
  - Cells + wiring = gate level design

- Cell library can then be *characterized*
  - Delays from input to output
    - … under different operating conditions
    - … under different loads
    - … under different input conditions
  - ″`.lib`″ file