

Communication-Centric Design

Robert Mullins

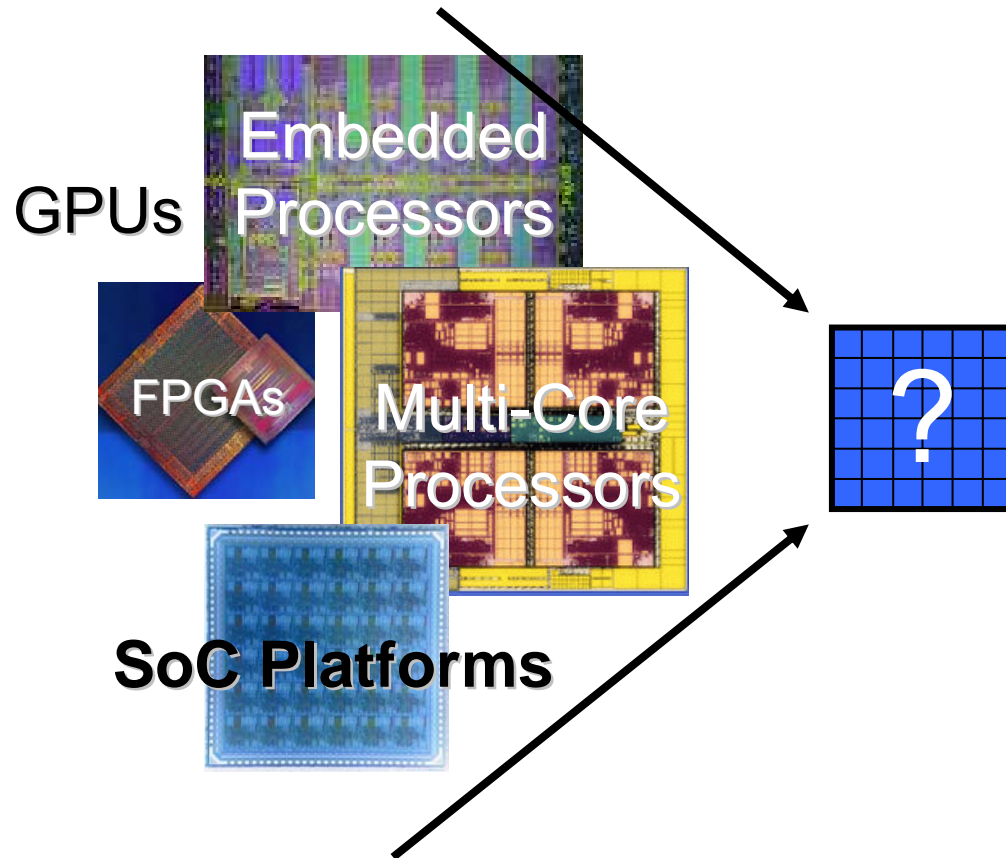
Computer Architecture Group

Computer Laboratory, University of Cambridge

*Workshop on On- and Off-Chip Interconnection Networks for
Multicore Systems, 6-7 Dec. 2006, Stanford.*

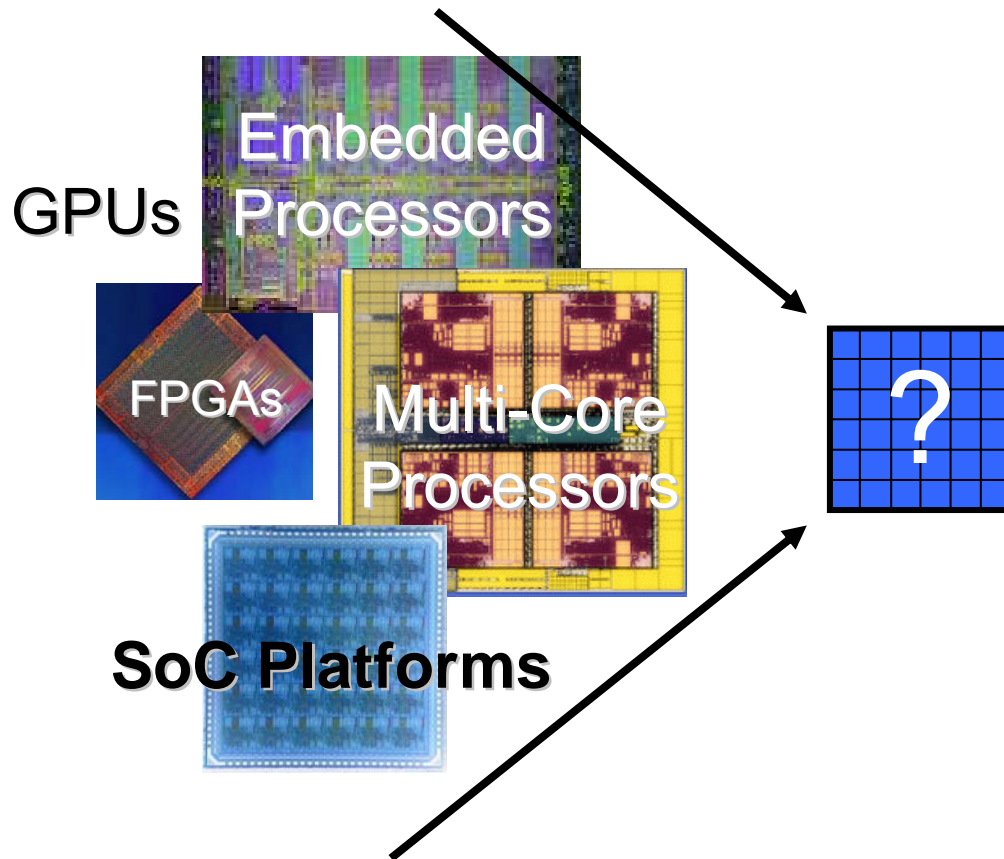


Convergence to flexible parallel architectures



- **Power Efficient**
 - Better match application characteristics (streaming, coarse-grain parallelism...)
 - Constraint-driven execution
- **Simple**
 - Increased regularity
 - S/W programmable
 - Limited core/tile set
 - Ease verification issues
- **Flexible**
 - Multi-use platform

Our Group's Research



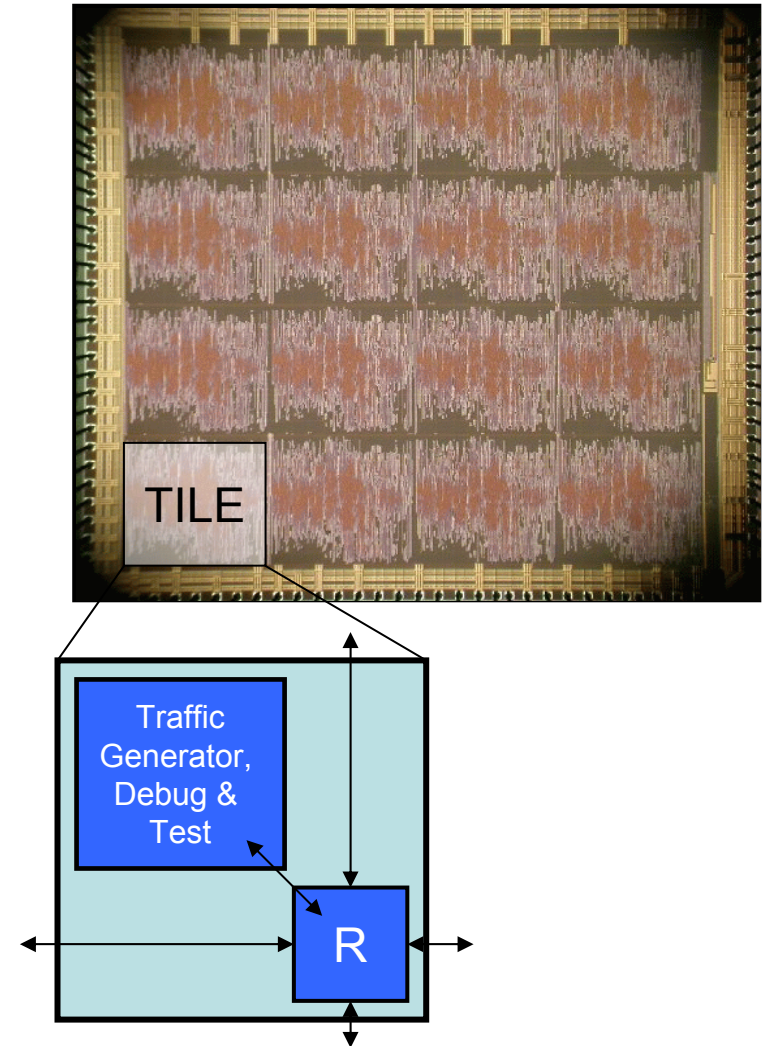
- **Now: support evolution of existing platforms**
 - Low-latency and low-power on-chip networks
 - System-timing considerations
 - Networking communications within FPGAs
 - Flexible networked SoC systems, virtual IP
 - On-chip serial interconnects
 - Multi-wavelength optical communication (off-chip)
 - Fault tolerant design
- **Future:**
 - Networks of processors to processing networks
 - Processing Fabrics

Low-Latency Virtual-Channel Packet-Switched Routers

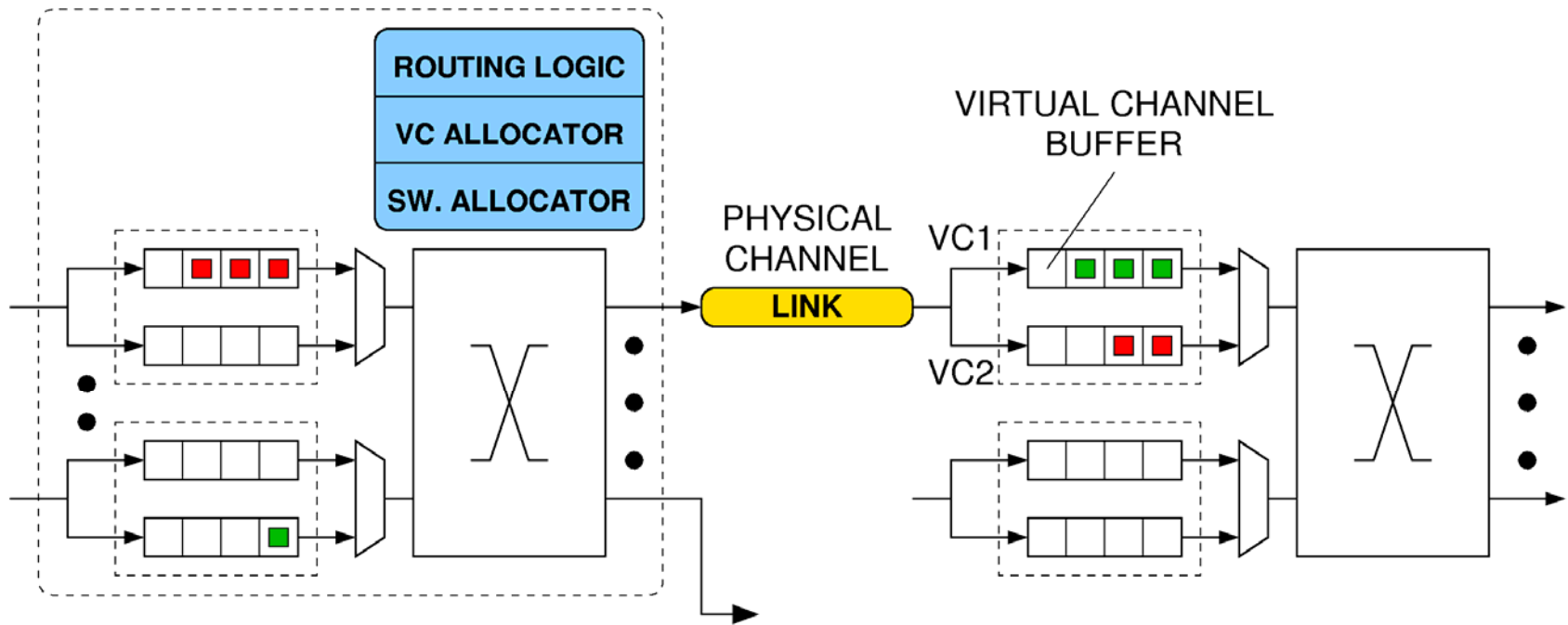
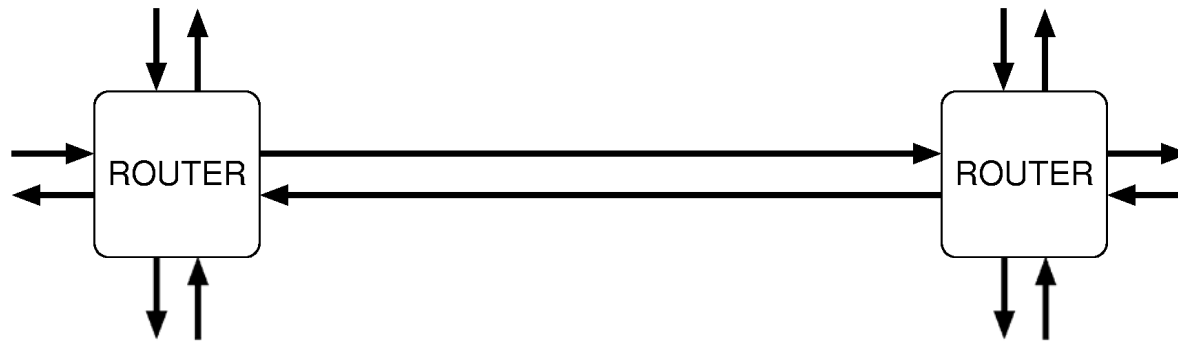
- Goal was to develop a virtual-channel network for a tiled processor architecture
- Collaboration with Krste Asanović's SCALE group at MIT
- Problem faced is rising interconnect costs
- Networking communications can increase communication latencies by an order of magnitude or more!

The Lochside test chip (2004/5)

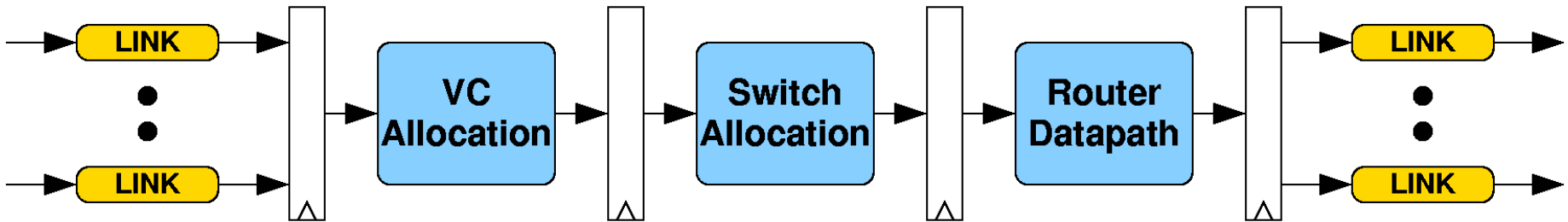
- UMC 0.18um Process
- 4x4 mesh network, 25mm²
- Single Cycle Routers (router + link = 1 clock)
- May be clocked by both traditional H-tree and DCG
- 4 virtual-channels/input
- 80-bit links
 - 64-bit data + 16-bit control
- 250MHz (worst-case PVT)
16Gb/s/channel (~35 FO4)
- Approx 5M transistors



Virtual-Channel Flow Control

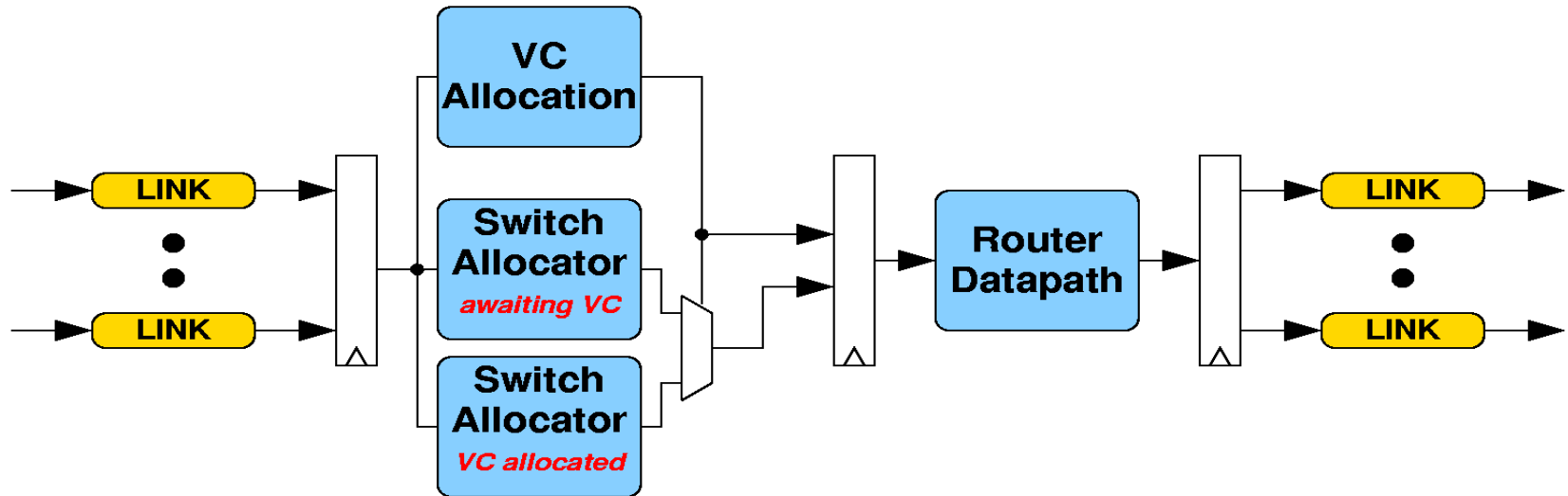


Typical Router Pipeline



- Router pipeline depth limits minimum latency
 - Even under low traffic conditions
 - Can make packet buffers less effective
 - Incurs pipelining overheads

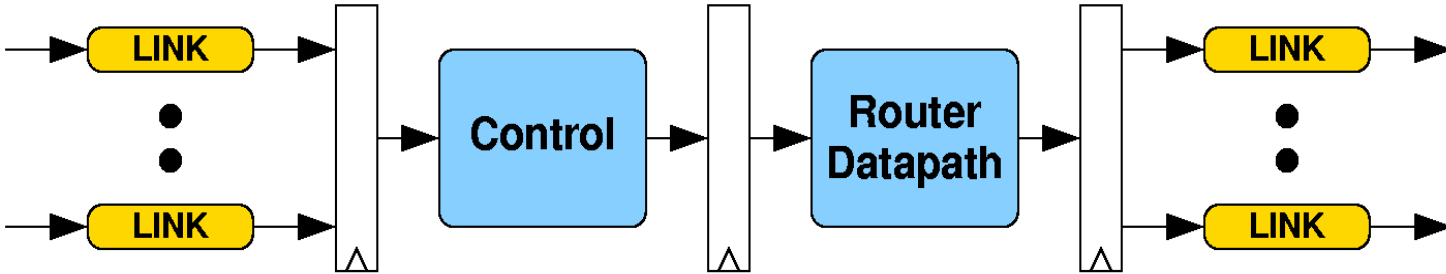
Speculative Router Architecture



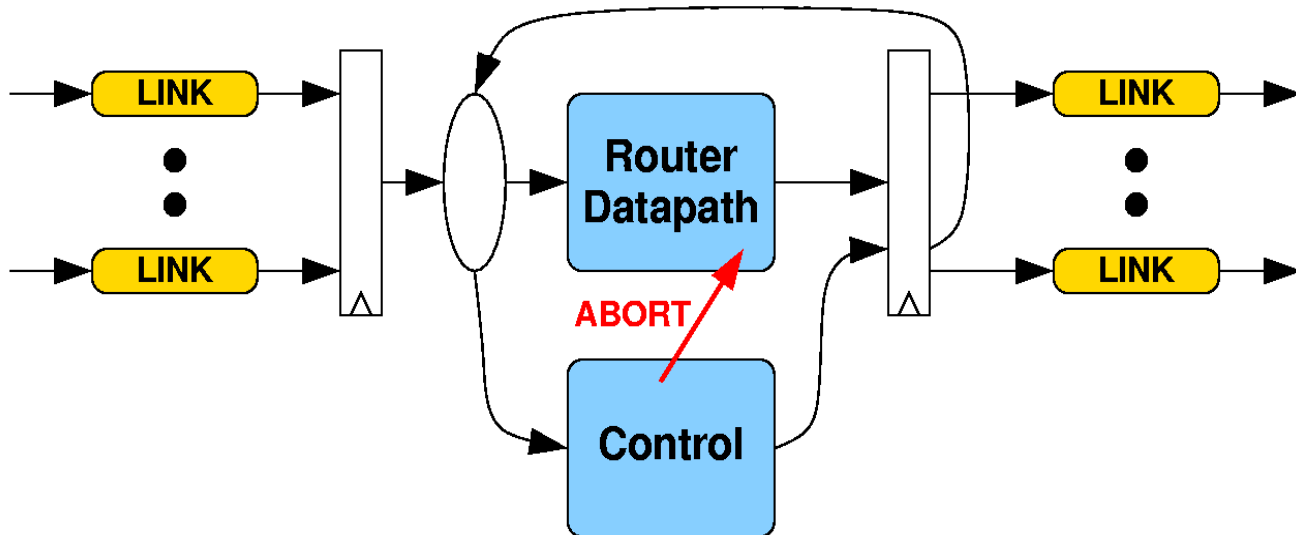
- VC and switch allocation may be performed concurrently:
 - Speculate that waiting packets will be successful in acquiring a VC
 - Prioritize non-speculative requests over speculative ones

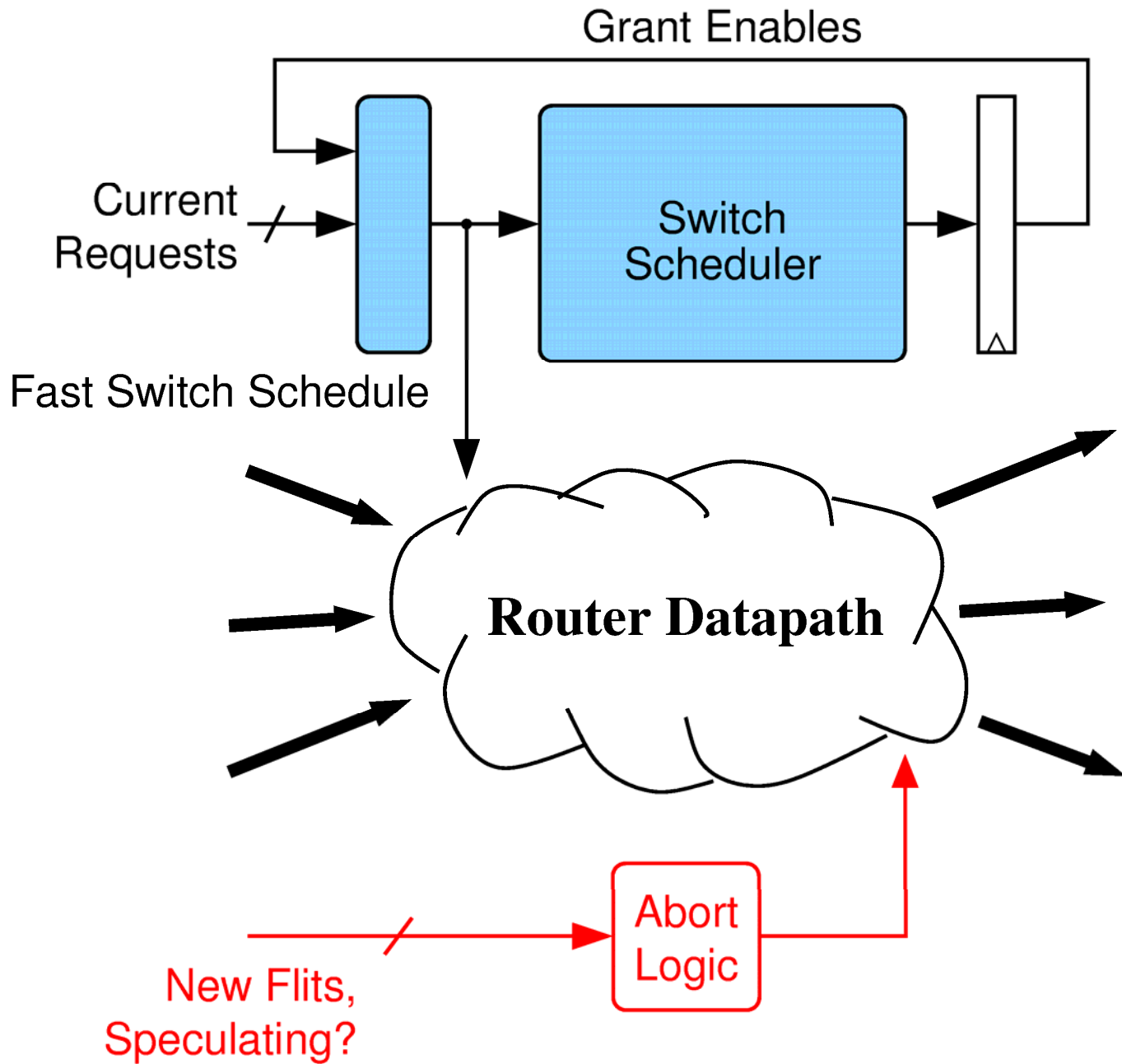
Li-Shiuan Peh and William J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers", In Proceedings HPCA'01, 2001.

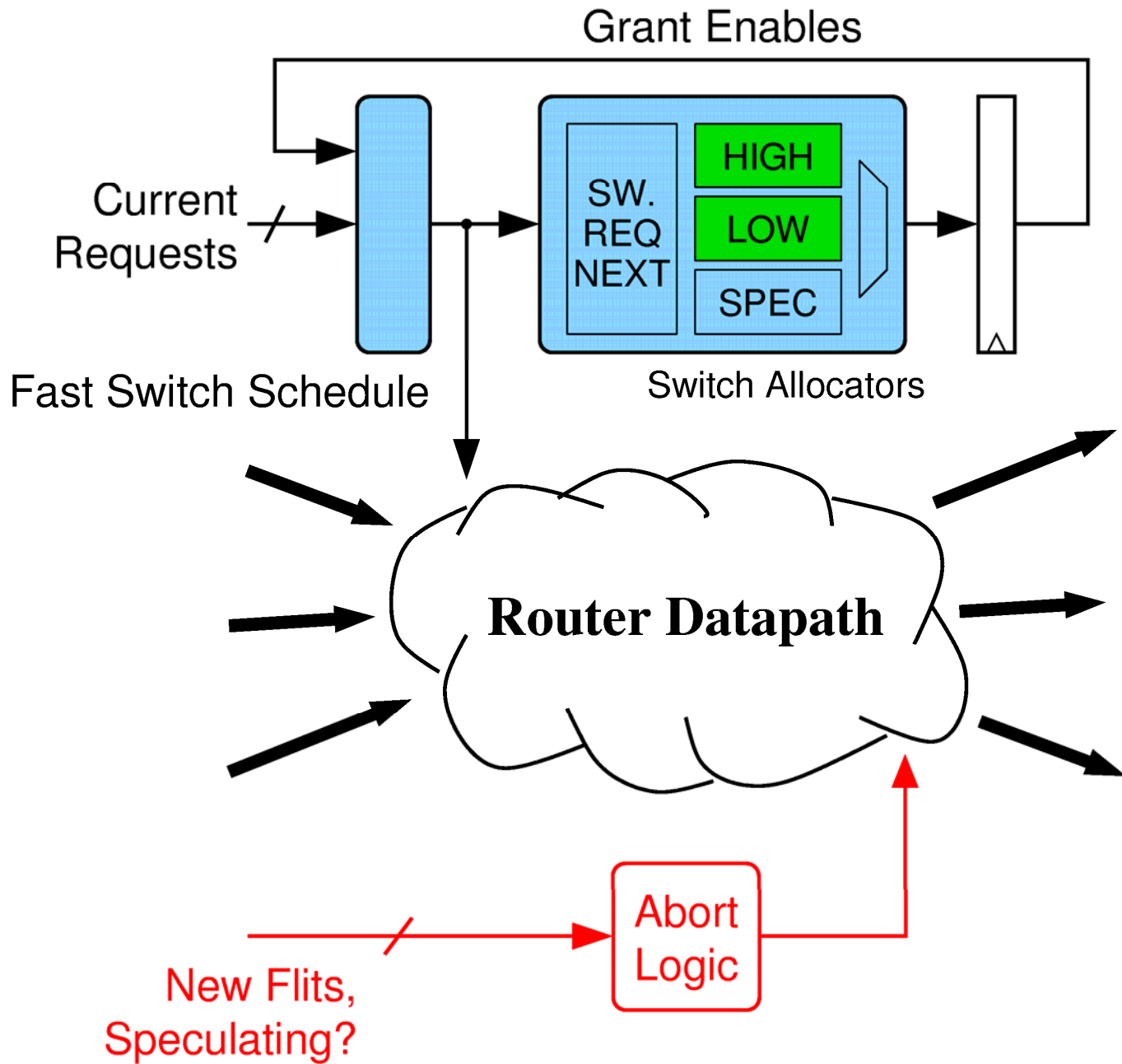
Single Cycle Speculative Router



SPECULATIVE CONTROL SIGNALS







Single Cycle Router Architecture

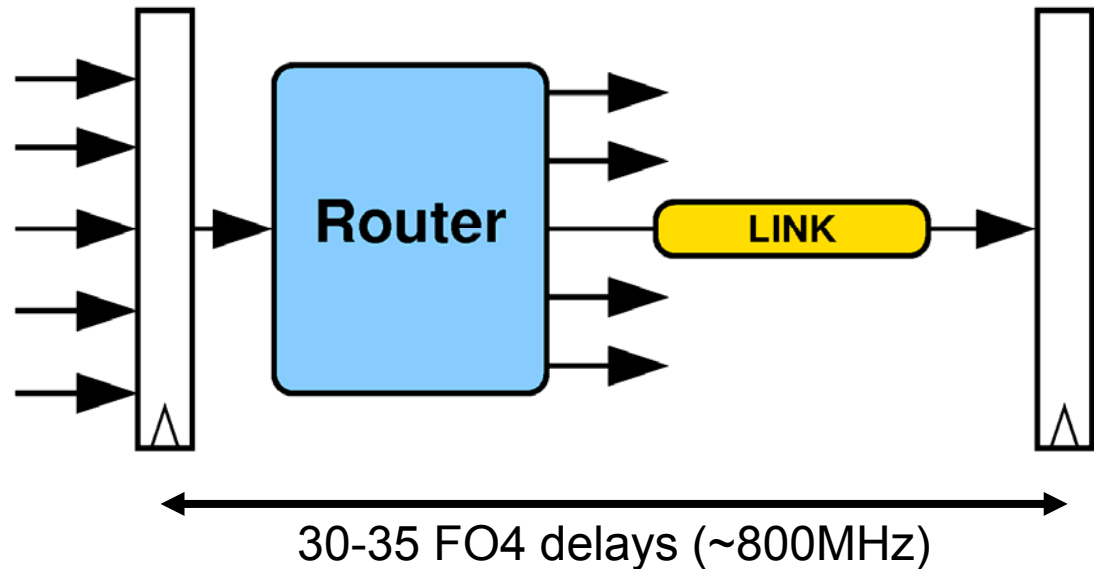
- Once speculation mechanism is in place a range of accuracy/cycle-time trade-offs can be made
 - Blocked VC, pipeline and speculate – use low priority switch scheduler
 - Switch and VC next request calculation
 - Don't bother calculating next switch requests just use current set. Safe to be pessimistic about what has been granted.
 - Need to be more accurate for VC allocation
 - Abort logic accuracy

Single Cycle Router Architecture

- Decreasing accuracy often leads to poorer schedule and more aborts but reduces the router's cycle time
- Impact of speculation on single cycle router:
 - 10% more cycles on average
 - clock period reduced by factor of 1.6
 - Network latency reduced by a factor of ~ 1.5
- Need to be careful about updating arbiter state correctly after speculation outcome is known

Lochside Router Clock Period

5-port router
4 VCs per port
64-bit links, ~1.5mm
90nm technology



100% standard cell

FF/Clocking

FIFOs/Control/Datapath

Link

23% (8.3 FO4)

53% (19 FO4)

22% (7.9 FO4) range 4.6-7.9

- Could move to router/link pipeline
- Option to pipeline control - maintaining single cycle best case
- Impact of technology scaling
- Scalability: doubling VCs to 8, only adds ~10% to cycle time

Router Power Optimisation

- Local and global clock gating & signal gating
 - Global clock gating exploits *early-request* signals from neighbouring routers
 - Slightly pessimistic (based on what is requested not granted)
 - Factor 2-4 reduction power consumption
- Peak 0.15mW/Mhz (0.35 unopt.)
- Low Random 0.06mW/Mhz (0.27 unopt.)

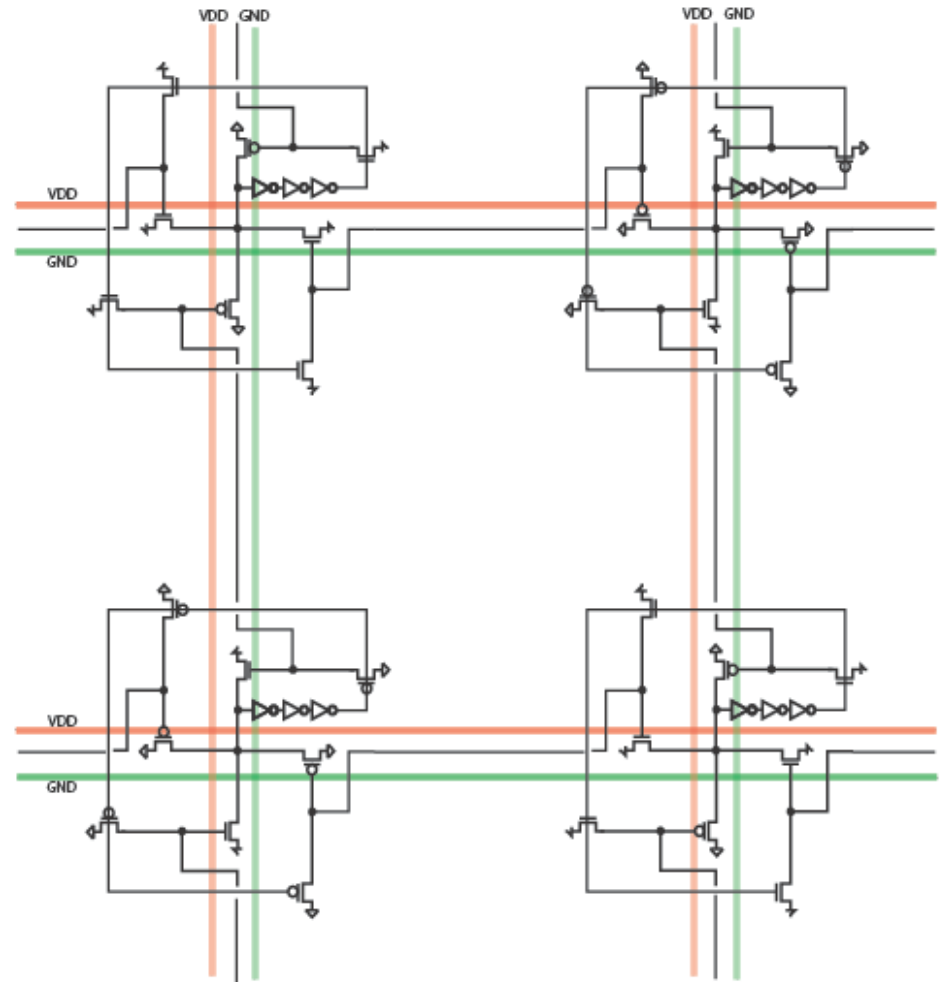
Analysis of Power Consumption

- 22% Static power
 - 11% Inter-Router Links
 - ~1% Global Clock tree
 - 65% Dynamic Power
 - Power Breakdown
 - ~50% local clock tree and input FIFOs
 - ~30% on router datapath
 - ~20% on scheduling and arbitration
- } Due to increase as % as technology scales

(Low random traffic case)

Distributed Clock Generator (DCG)

- Exploits self-timed circuitry to generate and a clock in a distributed fashion
- Low-skew and low-power solution to providing global synchrony
- Mesh topology
- Simple proof of concept provided by Lochside test chip



S. Fairbanks and S. Moore "Self-timed circuitry for global clocking", ASYNC'05

Beyond global synchrony

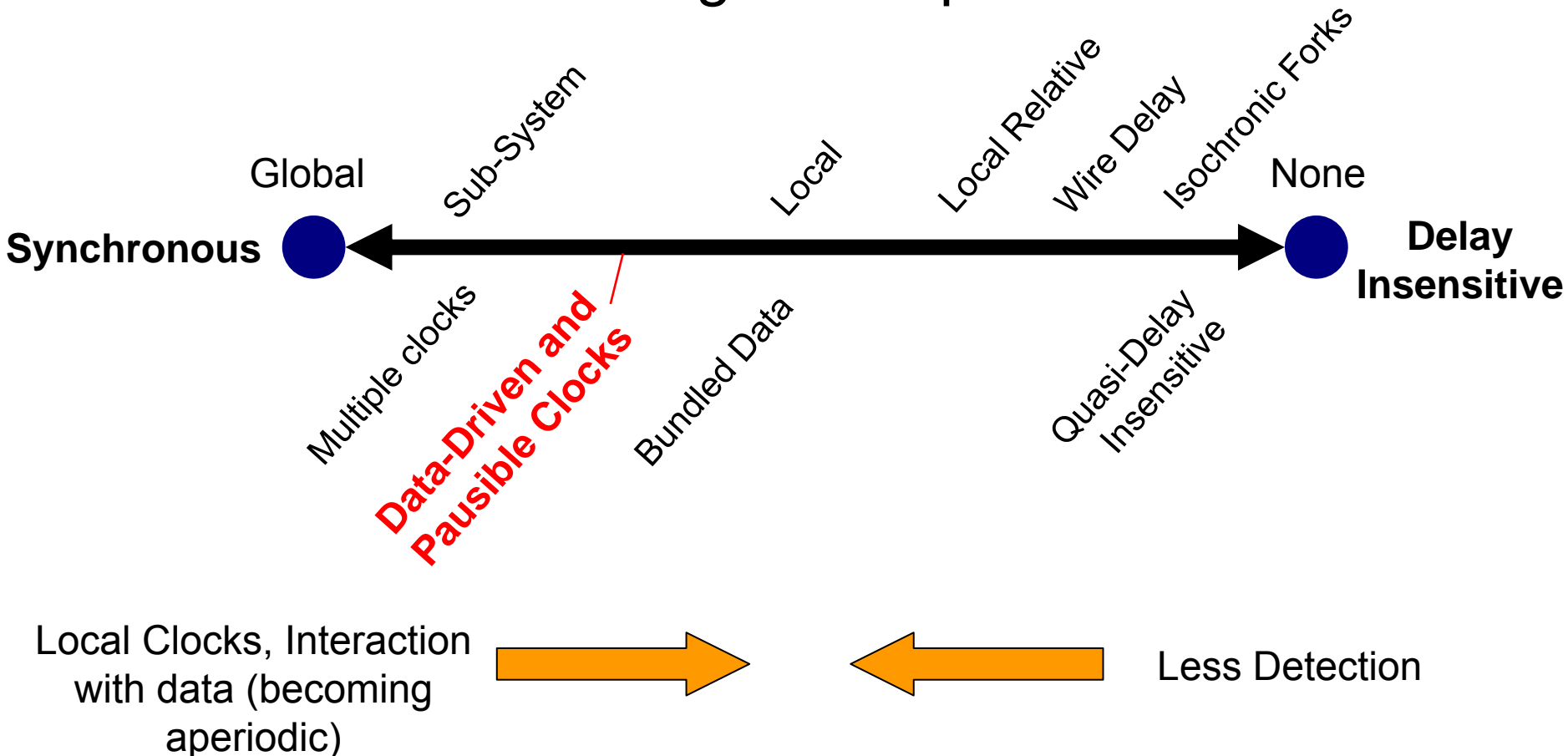


- Clock distribution issues
 - Challenge as network is physically distributed
 - Increasing process variation
- Synchronization
 - Core clock frequencies may vary, perhaps adaptively
 - Link and router DVS or other energy/perf. trade-offs
- Selecting a global network clock frequency
 - Run at maximum frequency continuously?
 - Use a multitude of network clock frequencies?
 - Select a global compromise?

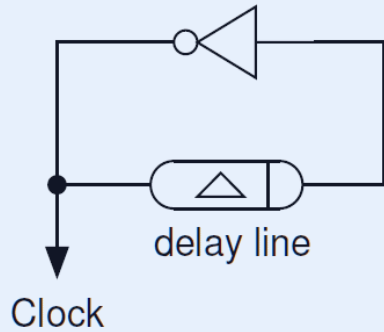
Beyond Global Synchrony

- A complete spectrum of approaches to system-timing exist

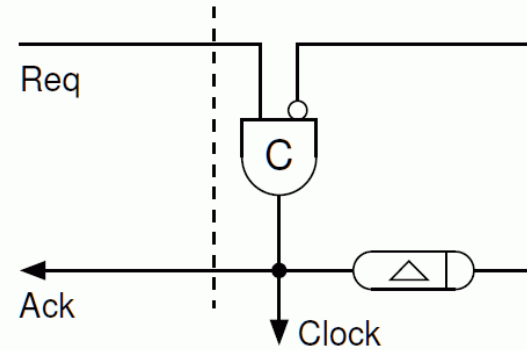
Timing Assumptions



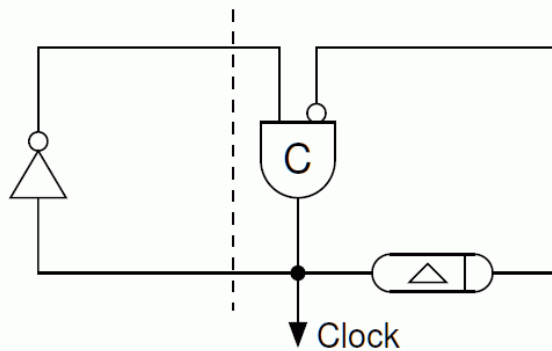
Data-Driven and Pausable Clocks



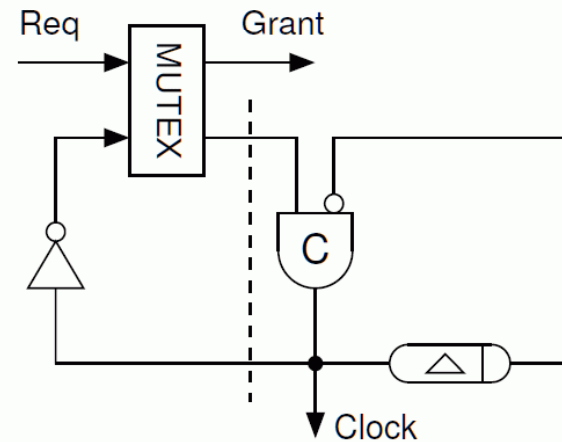
(a) Ring Oscillator



(b) Data-Driven Clock

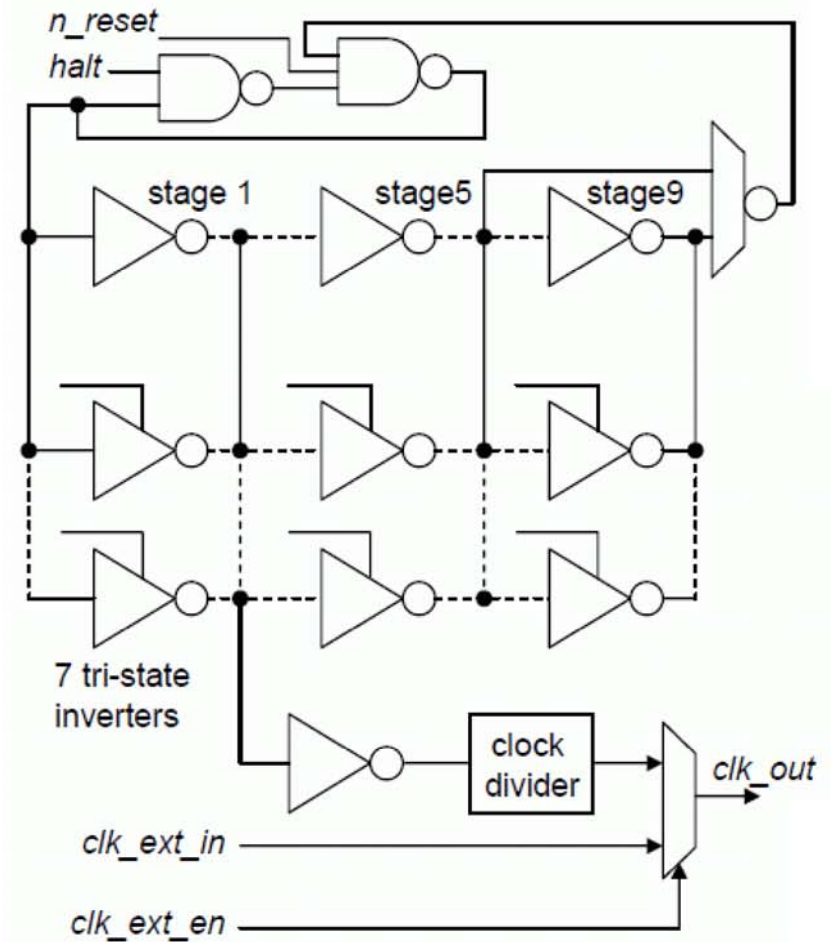
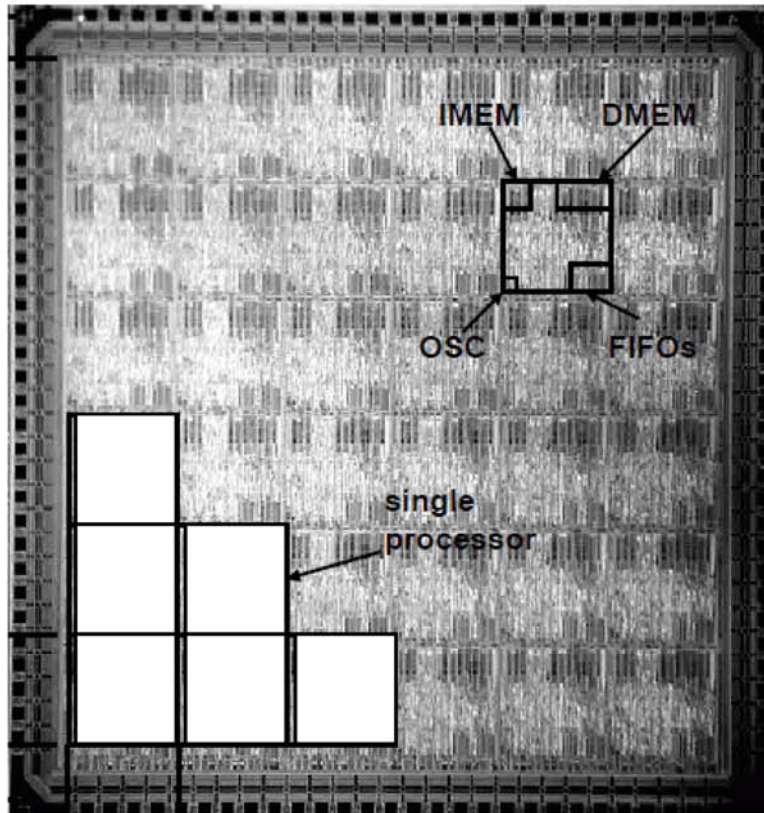


(c) Ring Oscillator (II)



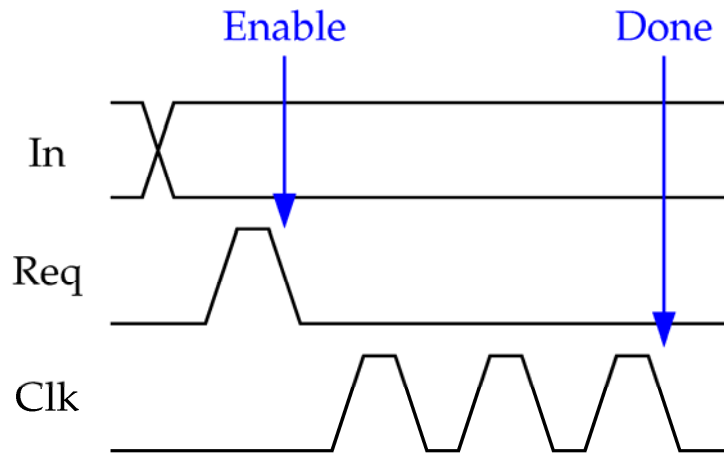
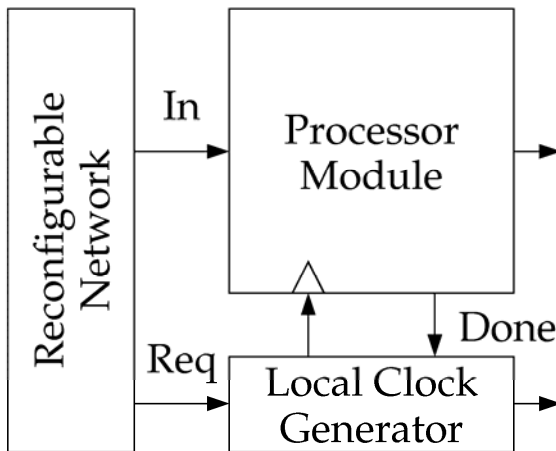
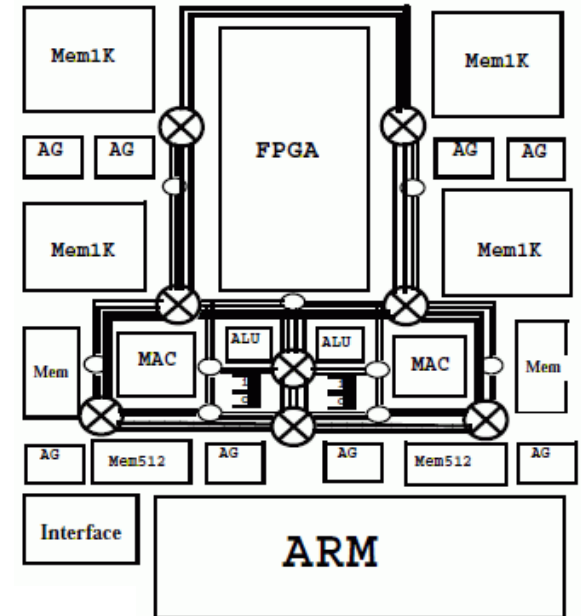
(d) Pausable Clock

Example: AsAP project (UC Davis, 2006)



Example: MAIA chip (Berkeley, 2000)

- GALS architecture, data-flow driven processing elements (“satellites”)

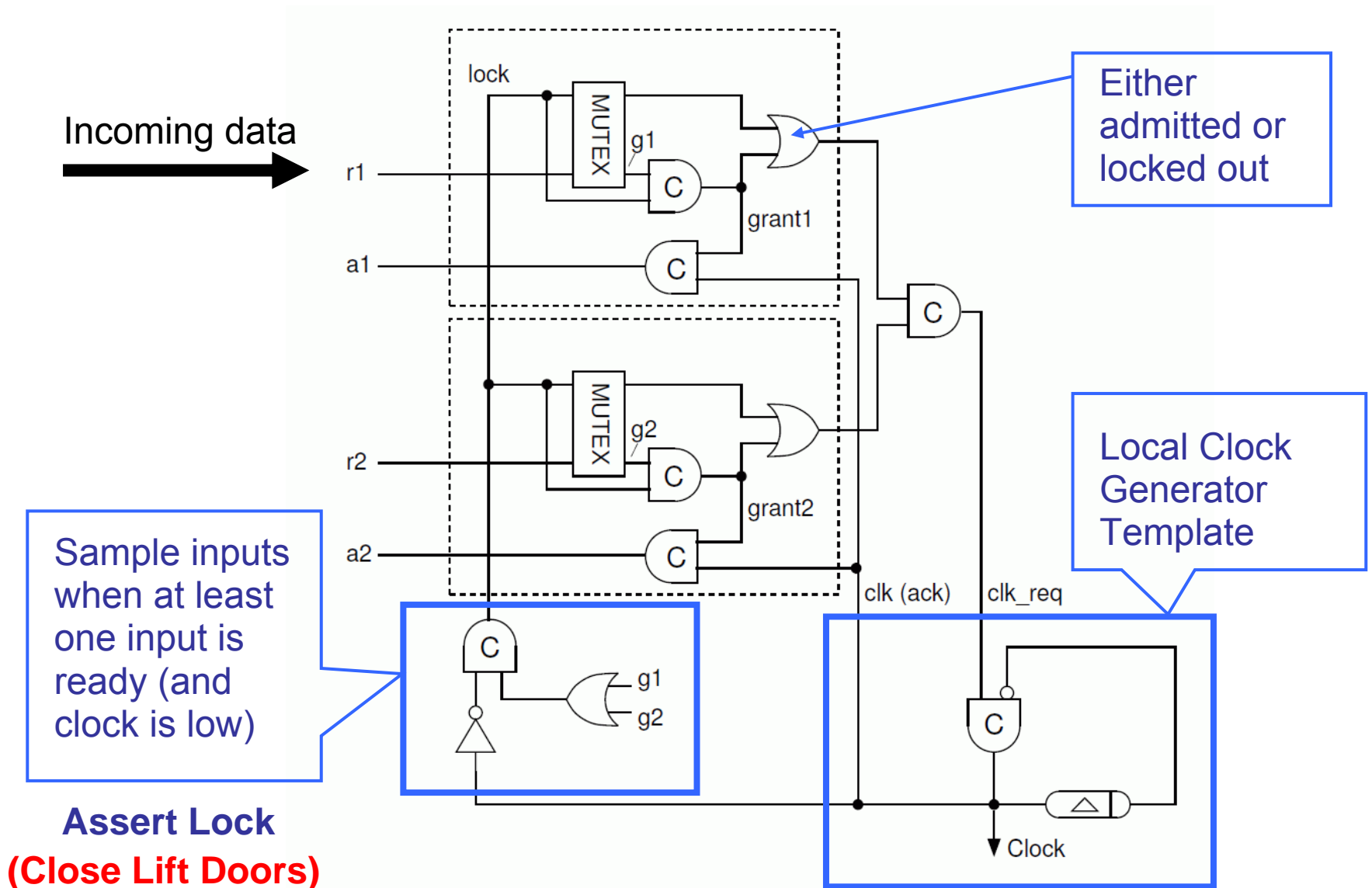


Zhang et al,
ISSCC'00

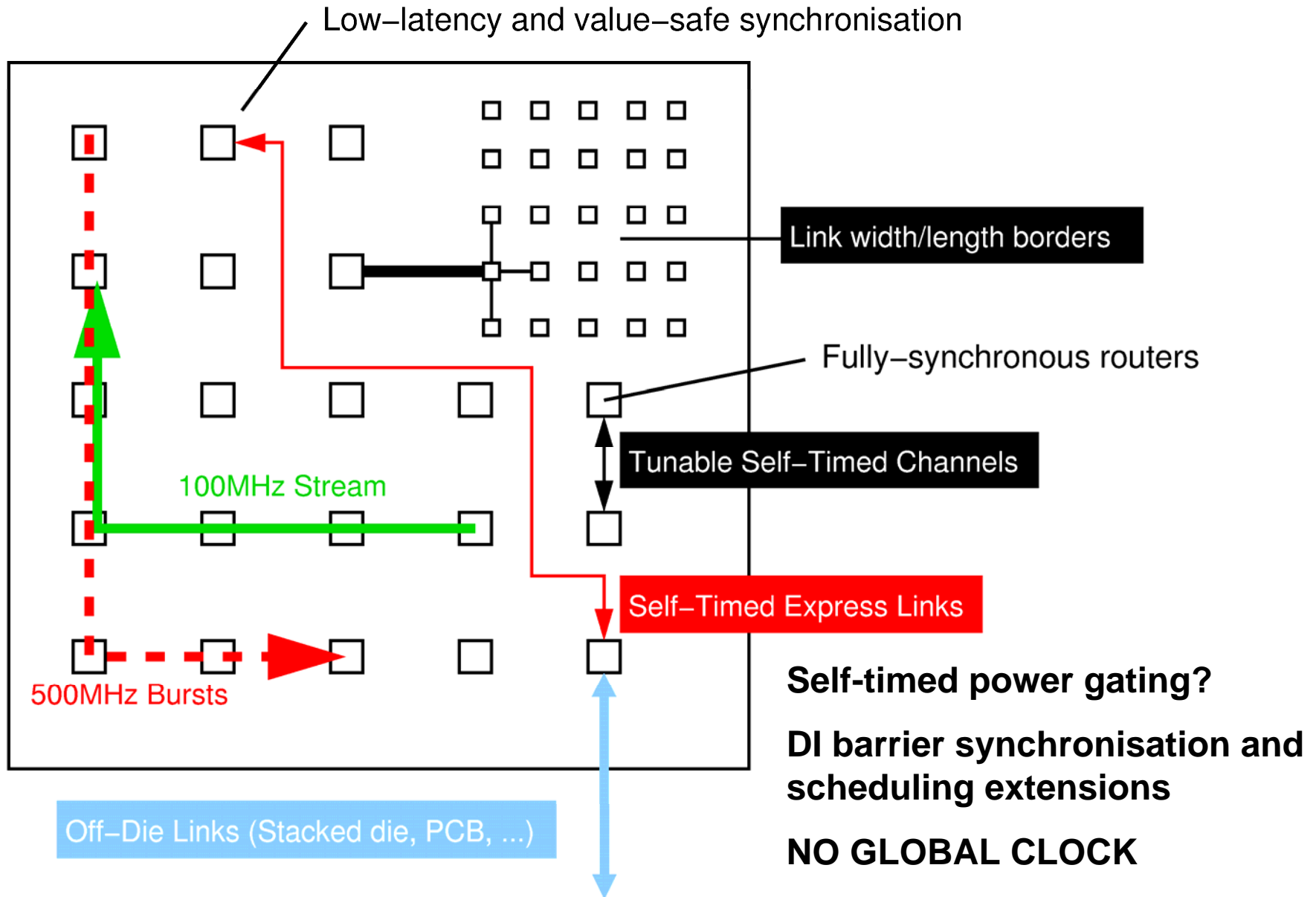
Data-Driven Clocking for On-Chip Routers

- Router should be clocked when one or more inputs are valid (or flits are buffered)
- Free running (paternoster) elevator
 - Chain of open compartments
 - Must synchronise before you jump on!
- Traditional elevator
 - Wait for someone to arrive
 - Close doors, decide who is in and who is out
 - Metastability issue again (potentially painful!)

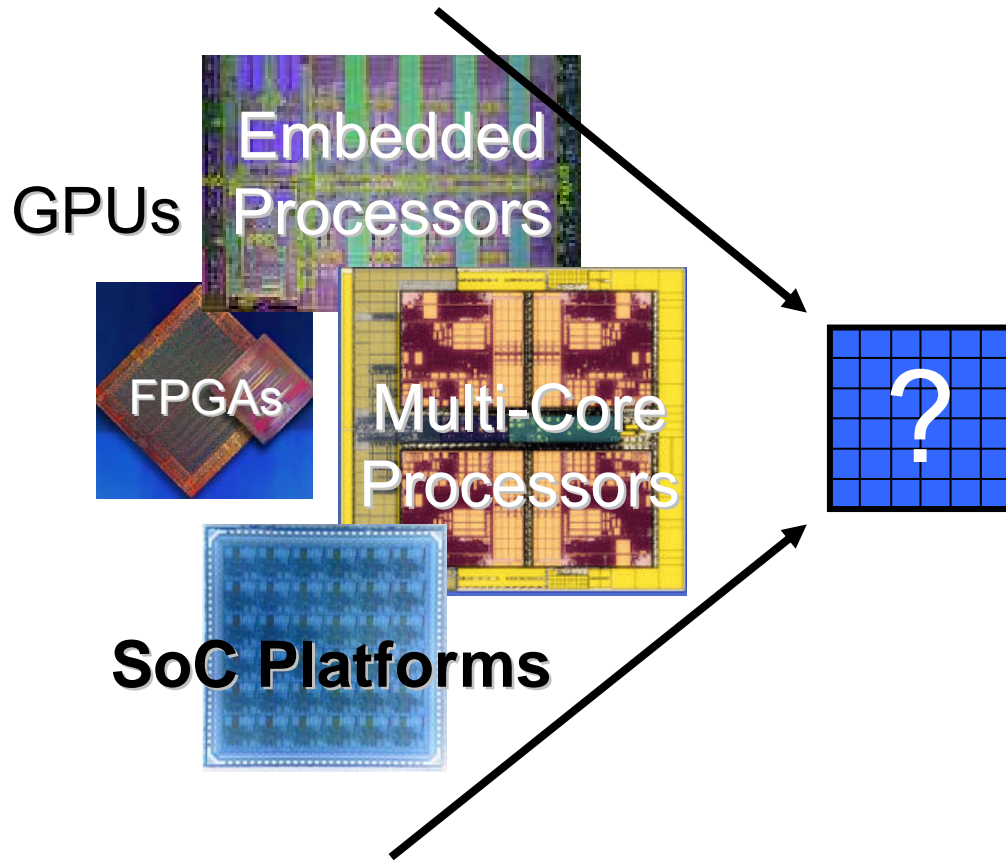
Data-Driven Clock Implementation



Data-driven clocking benefits



Networks of processors to processing networks



- Will a single universal parallel architecture be the eventual outcome of this convergence?

Current Focus

– **Network of Processors**

- Number of processors increase
- Core architectures tailored to “many-core” environment
- Remove hard tile boundaries
 - Why fix granularity of cores, communication and memory hierarchies?
- Move away from processor + router model
 - Everything is on the network
 - Richer interconnection of components, increased flexibility
- Add network-based services
 - Network aids collaboration, focuses resources, supports dynamic optimisations, scheduling, ...
 - Tailor virtual architecture to application

– **Processing Network** or **Fabric**

Increased flexibility to
optimise computation
and communication



Thank You.