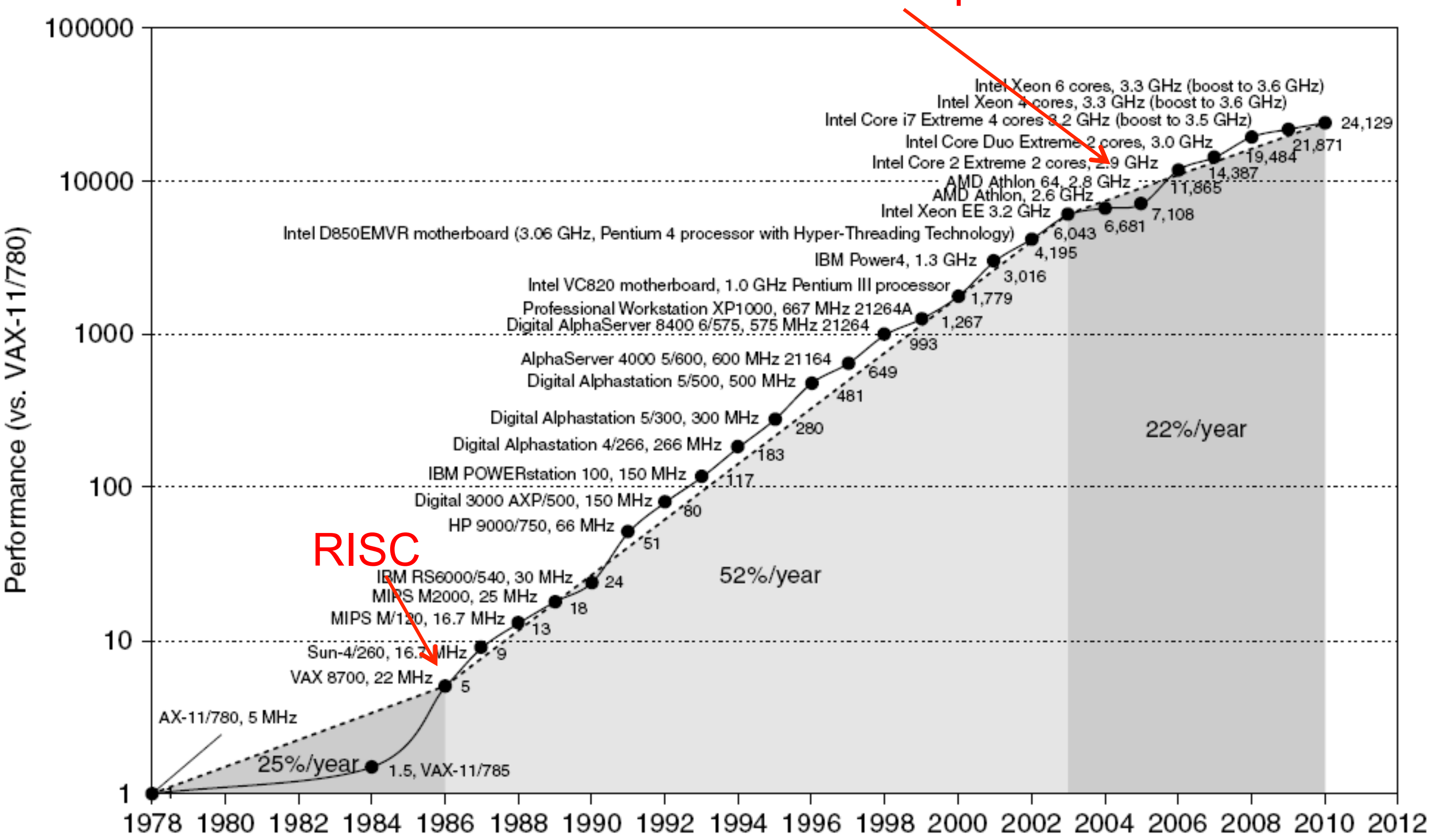
Multicore Architectures

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-739: Processor Design

Single Processor Performance



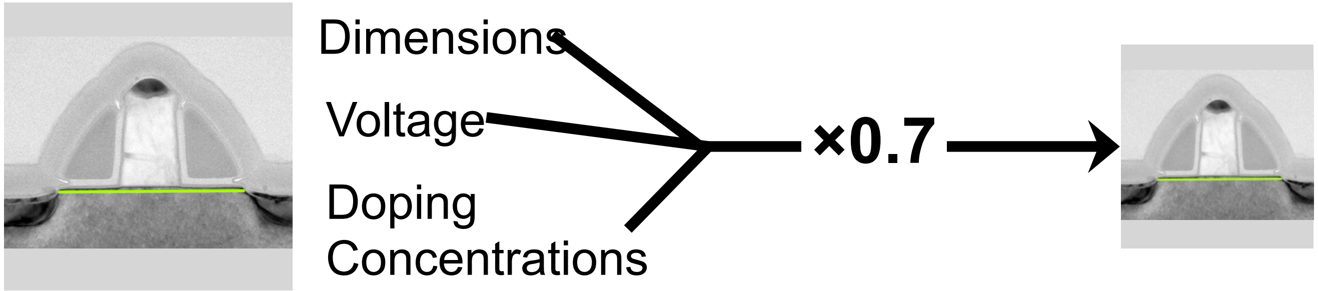
Move to multi-processor

RISC

Dennard Scaling:

Doubling the transistors; scale their power down

Transistor: 2D Voltage-Controlled Switch



Area ——— 0.5×↓ ———→

Capacitance ——— 0.7×↓ ———→

Frequency ——— 1.4×↑ ———→

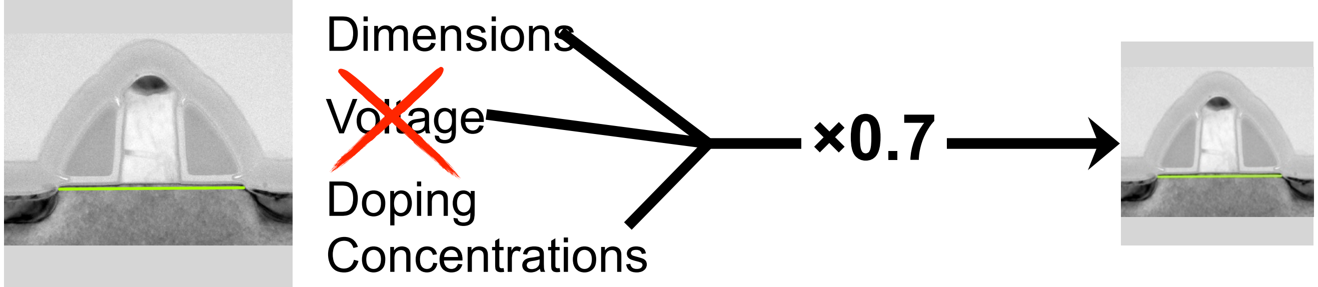
Power = Capacitance × Frequency × Voltage^2

Power ——— 0.5×↓ ———→

Dennard Scaling Broke:

Double the transistors; still scale their power down

Transistor: 2D Voltage-Controlled Switch



Dimensions

~~Voltage~~

Doping Concentrations

×0.7

Area 0.5×↓ →

Capacitance ~~0.7×~~↓ →

Frequency ~~1.4×~~↑ →

Power = Capacitance × Frequency × Voltage^2

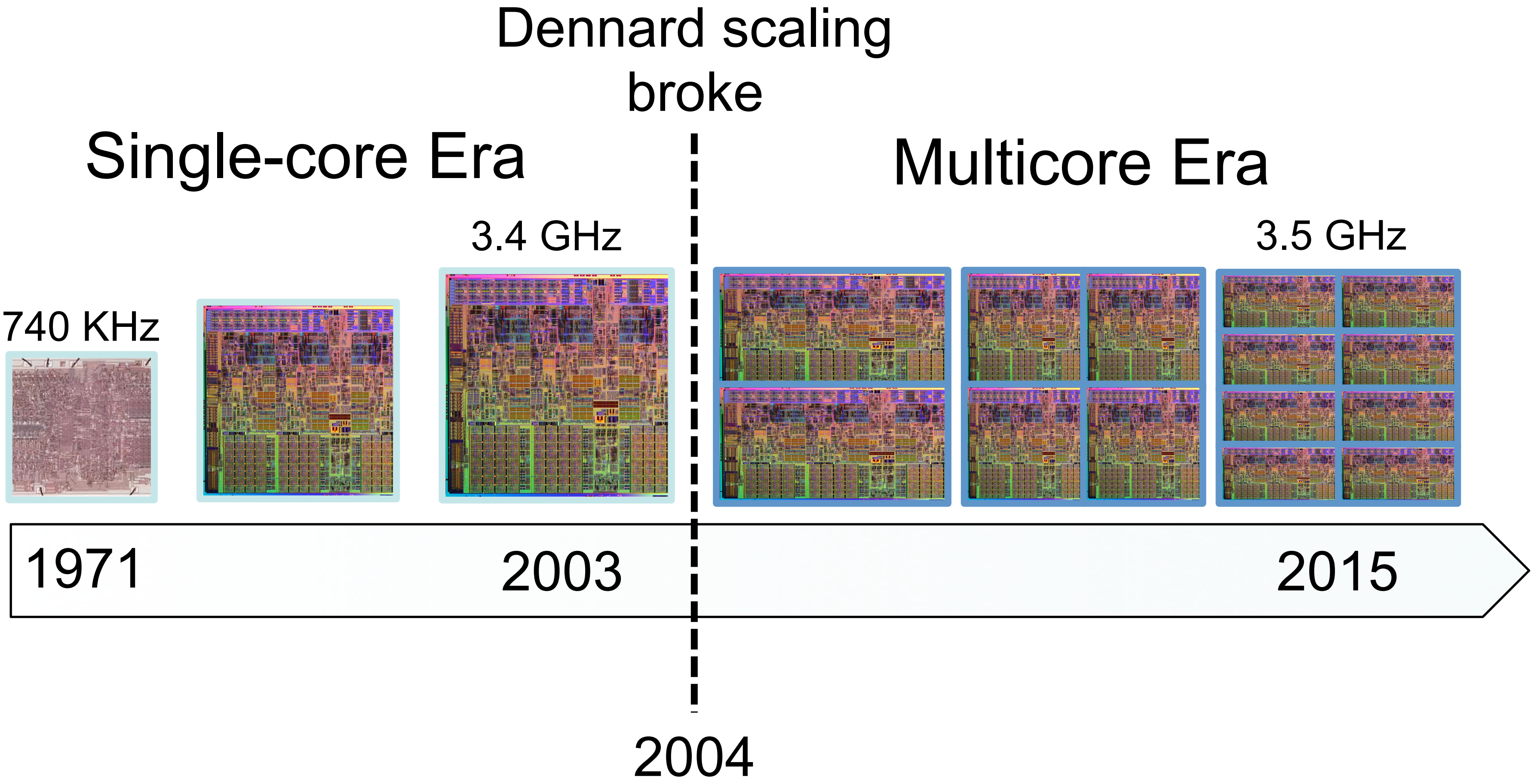
Power ~~0.5×~~↓ →

Transistor Scaling Model

From 45 nm to 8 nm

	[Dennard, 1974] **Historical Scaling**	[ITRS, 2010] **Optimistic Scaling Model**	[VLSI-DAT, 2010] **Conservative Scaling Model**
Area	32× ↓	32× ↓	32× ↓
Power	32× ↓	8.3× ↓	4.5× ↓
Speed	5.7× ↑	3.9× ↑	1.3× ↑

Evolution of Processors



Dennard scaling broke

Single-core Era

Multicore Era

740 KHz

3.4 GHz

3.5 GHz

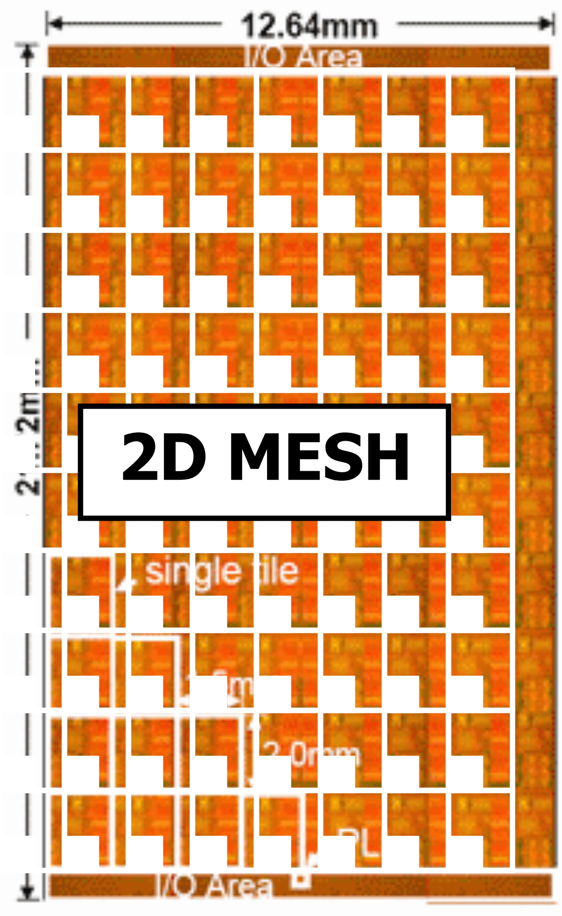
1971

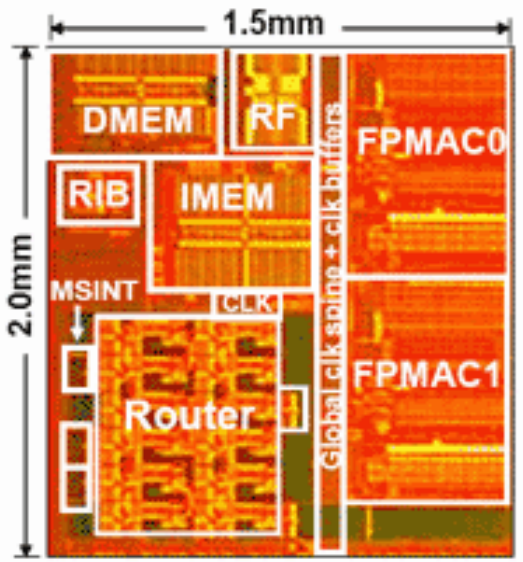
2003

2015

2004

Many Core Example





Technology	65nm CMOS Process
Interconnect	1 poly, 8 metal (Cu)
Transistors	100 Million
Die Area	275mm²
Tile area	3mm²
Package	1248 pin LGA, 14 layers, 343 signal pins

- Intel Polaris
  - 80 core prototype
- Academic Research ex:
  - MIT Raw, TRIPs
    - 2-D Mesh Topology
    - Scalar Operand Networks

CMP Examples

- Chip Multiprocessors (CMP)
- Becoming very popular

Processor	Cores/ chip	Multi-threaded?	Resources shared
IBM Power 4	2	No	L2/L3, system interface
IBM Power 5	2	Yes (2T)	Core, L2/L3, system interface
Sun Ultrasparc	2	No	System interface
Sun Niagara	8	Yes (4T)	Everything
Intel Pentium D	2	Yes (2T)	Core, nothing else
AMD Opteron	2	No	System interface (socket)

Multi-core Designs

- Use available transistors efficiently
  - Provide better perf, perf/cost, perf/watt
- Effectively share expensive resources
  - Socket/pins:
    - DRAM interface
    - Coherence interface
    - I/O interface

Multi-core Designs

- How to connect ?
- Where to Connect ?
- Shared caches
- Cache coherence

On-Chip Bus/Crossbar

- Used widely (Power4/5/6, Piranha, Niagara, etc.)
  - Assumed not scalable
  - Is this really true, given on-chip characteristics?

- Simple, straightforward, nice ordering properties
  - Wiring is a nightmare (for crossbar)
  - Bus bandwidth is weak (even multiple busses)
  - Compare piranha 8-lane bus (32GB/s) to Power4 crossbar (100+GB/s)

On-Chip Ring

- Point-to-point ring interconnect
  - Simple, easy
  - Nice ordering properties (unidirectional)
  - Every request a broadcast (all nodes can snoop)
  - Scales poorly: O(n) latency, fixed bandwidth

On-Chip Mesh

- Widely assumed in academic literature
- Tilera, Intel 80-core prototype
- Not symmetric, so have to watch out for load imbalance on inner nodes/links
  - 2D torus: wraparound links to create symmetry
    - Not obviously planar
    - Can be laid out in 2D but longer wires, more intersecting links
- Latency, bandwidth scale well
- Lots of existing literature

Switching/Flow Control Overview

- Topology: determines connectivity of network
- Routing: determines paths through network
- Flow Control: determine allocation of resources to messages as they traverse network
  - Buffers and links
  - Significant impact on throughput and latency of network

Packets

- Messages: composed of one or more packets
  - If message size is <= maximum packet size only one packet created
- Packets: composed of one or more flits
- Flit: flow control digit
- Phit: physical digit
  - Subdivides flit into chunks = to link width
  - In on-chip networks, flit size == phit size.
    - Due to very wide on-chip channels

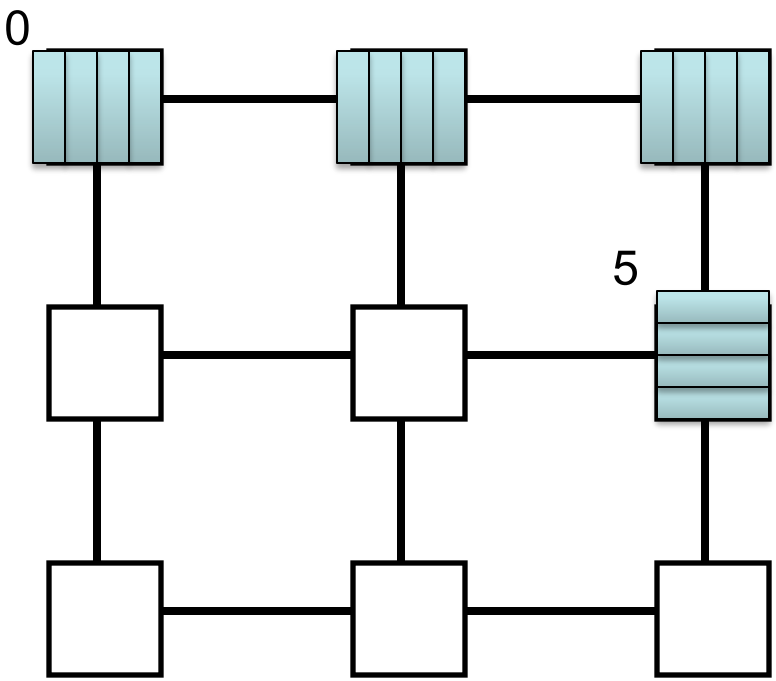
Switching

- Different flow control techniques based on granularity
- Circuit-switching: operates at the granularity of messages
- Packet-based: allocation made to whole packets
- Flit-based: allocation made on a flit-by-flit basis

Packet-Based Flow Control

- Store and forward
- Links and buffers are allocated to entire packet
- Head flit waits at router until entire packet is buffered before being forwarded to the next hop
- Not suitable for on-chip
  - Requires buffering at each router to hold entire packet
  - Incurs high latencies (pays serialization latency at each hop)

Store and Forward Example

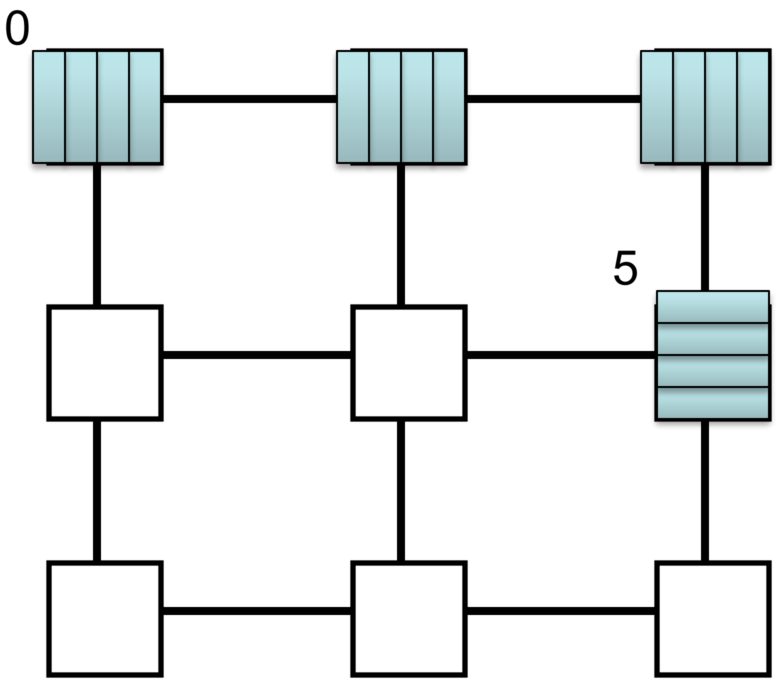


- High per-hop latency
- Larger buffering required

Virtual Cut Through

- Packet-based: similar to Store and Forward
- Links and Buffers allocated to entire packets
- Flits can proceed to next hop before tail flit has been received by current router
  - But only if next router has enough buffer space for entire packet
- Reduces the latency significantly compared to SAF
- But still requires large buffers
  - Unsuitable for on-chip

Virtual Cut Through Example

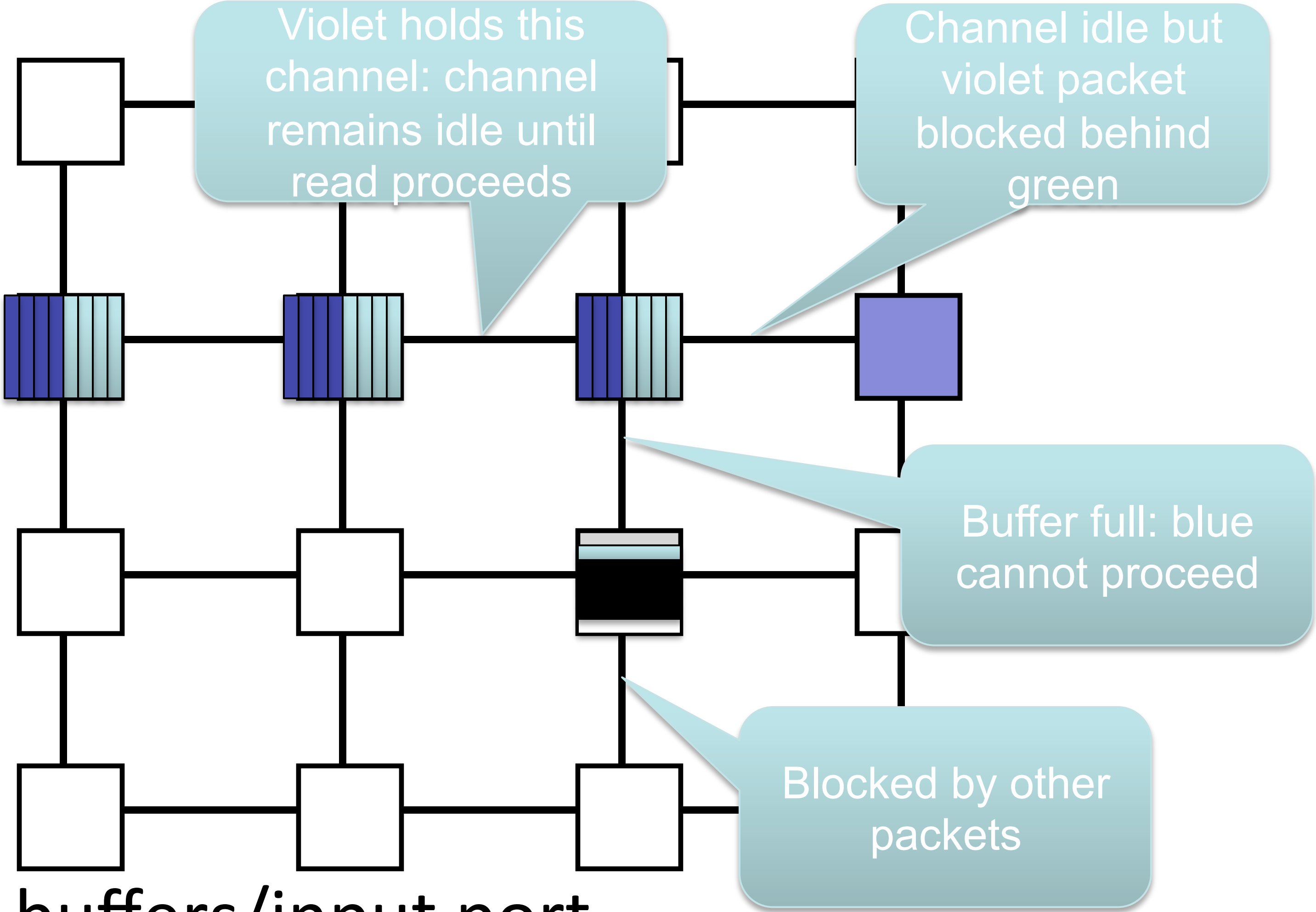


- Lower per-hop latency
- Larger buffering required

Flit Level Flow Control

- Wormhole flow control
- Flit can proceed to next router when there is buffer space available for that flit
  - Improved over SAF and VCT by allocating buffers on a flit-basis
- Pros
  - More efficient buffer utilization (good for on-chip)
  - Low latency
- Cons
  - Poor link utilization: if head flit becomes blocked, all links spanning length of packet are idle
    - Cannot be re-allocated to different packet
    - Suffers from head of line (HOL) blocking

Wormhole Example



Violet holds this channel: channel remains idle until read proceeds

Channel idle but violet packet blocked behind green

Buffer full: blue cannot proceed

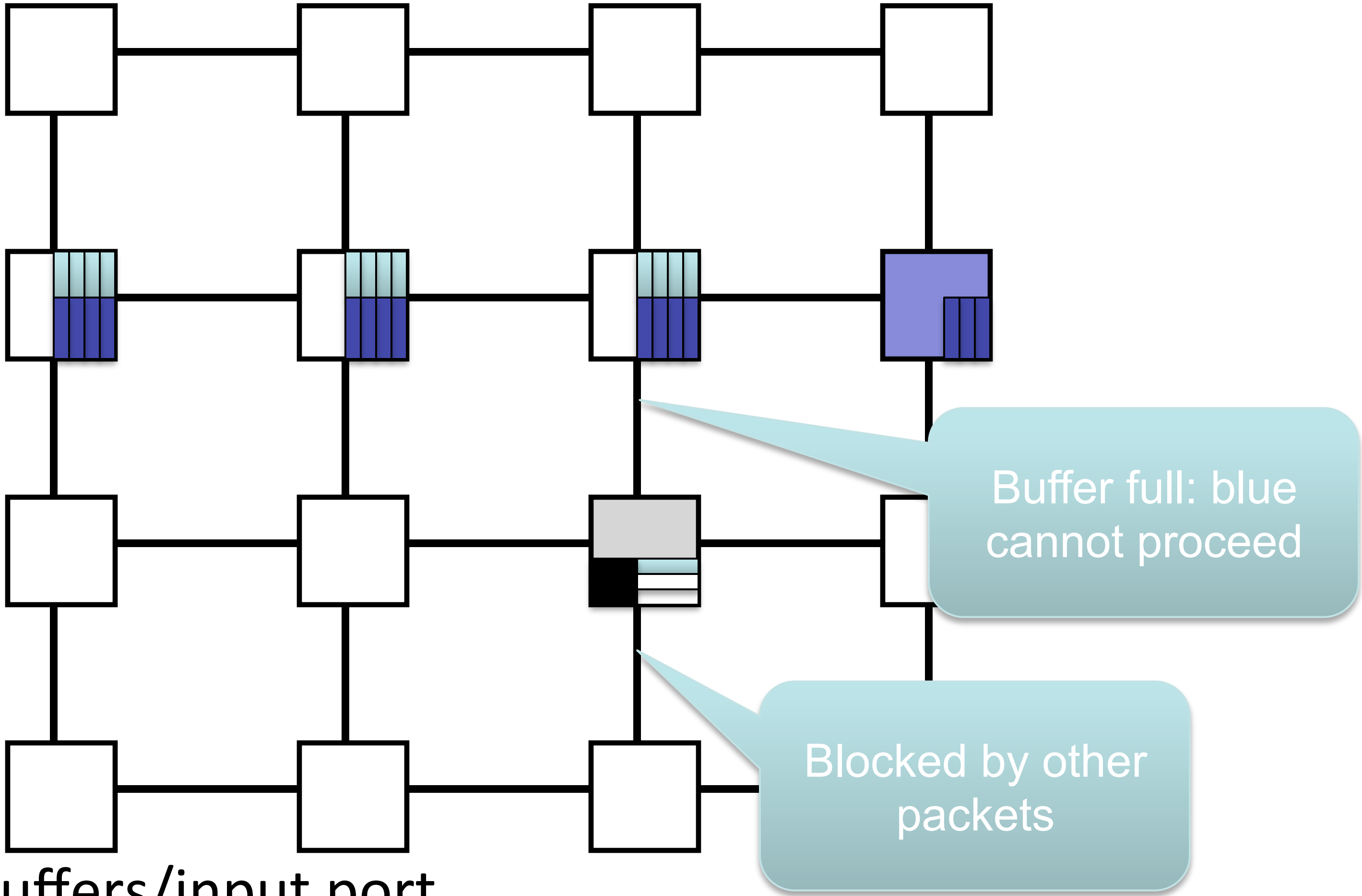
Blocked by other packets

- 6 flit buffers/input port

Virtual Channel Flow Control

- Virtual channels used to combat HOL block in wormhole
- Virtual channels: multiple flit queues per input port
  - Share same physical link (channel)
- Link utilization improved
  - Flits on different VC can pass blocked packet

Virtual Channel Example

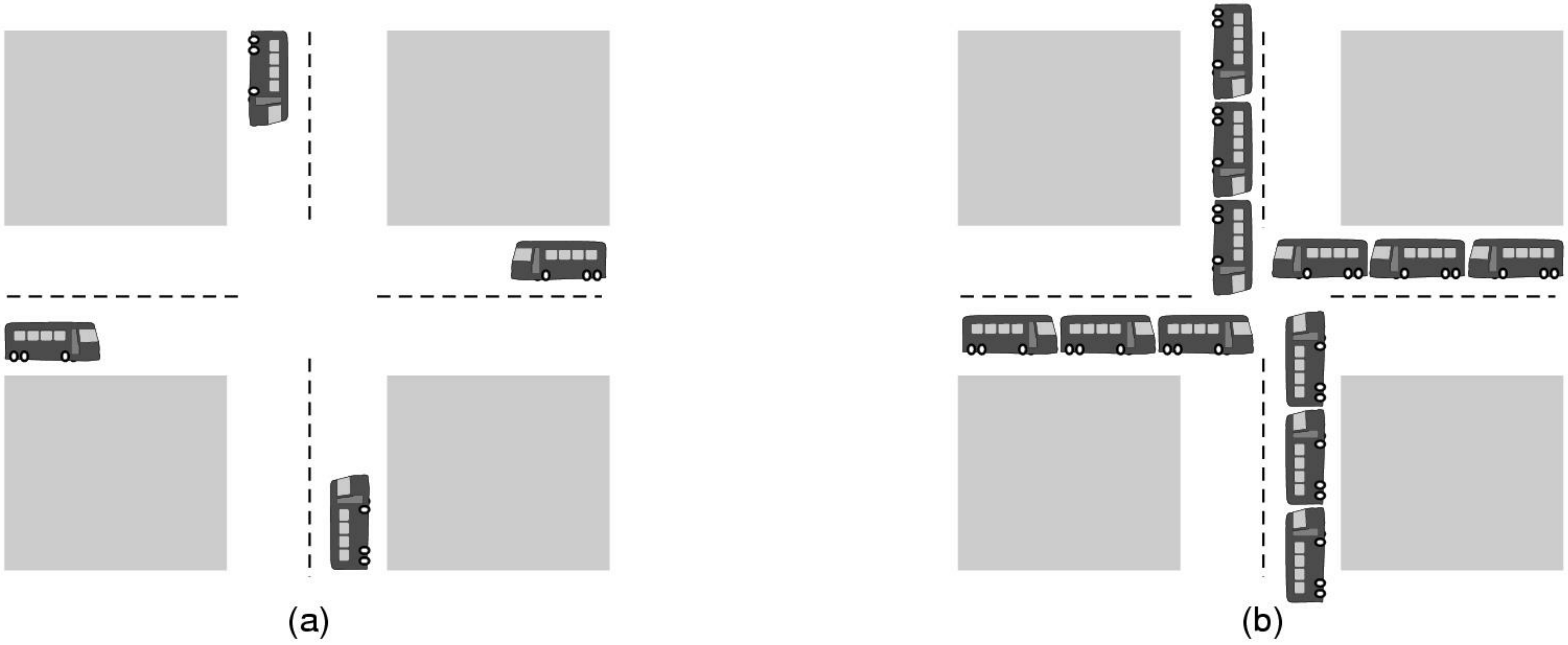


Buffer full: blue cannot proceed

Blocked by other packets

- 6 flit buffers/input port
- 3 flit buffers/VC

Deadlock



(a) A potential deadlock. (b) an actual deadlock.

Deadlock

- Using flow control to guarantee deadlock freedom give more flexible routing
- Escape Virtual Channels
  - If routing algorithm is not deadlock free
  - VCs can break resource cycle
  - Place restriction on VC allocation or require one VC to be DOR

Topology Overview

- Definition: determines arrangement of channels and nodes in network
- Analogous to road map
- Often first step in network design
- Routing and flow control build on properties of topology

Abstract Metrics

- Use metrics to evaluate performance and cost of topology
- Also influenced by routing/flow control
  - At this stage
    - Assume ideal routing (perfect load balancing)
    - Assume ideal flow control (no idle cycles on any channel)
- Switch Degree: number of links at a node
  - Proxy for estimating cost
    - Higher degree requires more links and port counts at each router

Latency

- Time for packet to traverse network
  - Start: head arrives at input port
  - End: tail departs output port
- Latency = Head latency + serialization latency
  - Serialization latency: time for packet with Length L to cross channel with bandwidth b (L/b)
- Hop Count: the number of links traversed between source and destination
  - Proxy for network latency
  - Per hop latency with zero load

Impact of Topology on Latency

- Impacts average minimum hop count
- Impact average distance between routers
- Bandwidth

Throughput

- Data rate (bits/sec) that the network accepts per input port
- Max throughput occurs when one channel saturates
  - Network cannot accept any more traffic
- Channel Load
  - Amount of traffic through channel c if each input node injects 1 packet in the network

Thank You