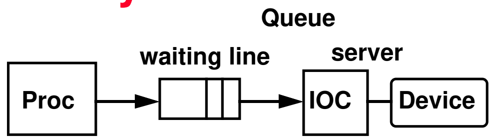
Lecture 22: I/O—I/O Busses

Professor Randy H. Katz
Computer Science 252
Spring 1995

Review: Storage System Issues

- Historical Context of Storage I/O
- Storage I/O Performance Measures
- Secondary and Tertiary Storage Devices
- A Little Queuing Theory
- Processor Interface Issues
- I/O & Memory Buses
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance

Review: A Little Queuing Theory: Litttle's Theorem



- **Queuing models assume state of equilibrium : input rate = output rate**
- **Notation:**

r	average number of arriving customers/second
T_s	average time to service a customer
u	server utilization (0..1): $u = r \times T_s$
T_w	average time/customer in waiting line
T_q	average time/customer in queue: $T_q = T_w + T_s$
L_w	average length of waiting line: $L_w = r \times T_w$
L_q	average length of queue: $L_q = r \times T_q$

- **Little's Law:** $r = L_q / T_q = L_w / T_w = u / T_s$
  Mean number customers = arrival rate x mean response time

Review: A Little Queuing Theory—M/G/1 and M/M/1

- **Assumptions so far**
  - equilibrium
  - time between two success arrivals in line are random
  - server can start on next customer immediately after finish prior
  - no limit to the waiting line: works First-In-First-Out
  - Afterward, all customers in line must complete; each avg. T_s
- **Described Markovian request arrival**
  - M for C=1 exponentially random, General service distribution (no restrictions), and 1 server: *M/G/1 queue*
- **When Service times have C =1, *M/M/1 queue***
  $T_w = T_s \times u \times (1 + C) / (2 \times (1 - u)) = T_s \times u / (1 - u)$

T_s	average time to service a customer
u	server utilization (0..1): $u = r \times T_s$
T_w	average time/customer in waiting line

- **Some confusion: waiting time = queue delay?**

Review: Processor Interface Issues

- Processor interface
  - interrupts
  - memory mapped I/O
- I/O Control Structures
  - polling
  - interrupts
  - DMA
  - I/O Controllers
  - I/O Processors

Review: Relationship to Processor Architecture

- I/O instructions and busses have disappeared
- Interrupt vectors have been replaced by jump tables
- Interrupt stack replaced by shadow registers
- Interrupt types reduced in number
- Caches required for processor performance cause problems for I/O
- Virtual memory frustrates DMA
- Load/store architecture at odds with atomic operations
- Stateful processors hard to context switch

Interconnect Trends

- Interconnect = glue that interfaces computer system components
- High speed hardware interfaces + logical protocols
- Networks, channels, backplanes

	Network	Channel	Backplane
Distance	>1000 m	10 - 100 m	1 m
Bandwidth	10 - 100 Mb/s	40 - 1000 Mb/s	320 - 1000+ Mb/s
Latency	high (>ms)	medium	low (<µs)
Reliability	low Extensive CRC	medium Byte Parity	high Byte Parity

message-based ←→ memory-mapped
narrow pathways ←→ wide pathways
distributed arb ←→ centralized arb

RHK.S96 7

Backplane Architectures

Metric	VME	FutureBus	MultiBus II	SCSI-I
Bus Width (signals)	128	96	96	25
Address/Data Multiplexed?	No	Yes	Yes	na
Data Width	16 - 32	32	32	8
Xfer Size	Single/Multiple	Single/Multiple	Single/Multiple	Single/Multiple
# of Bus Masters	Multiple	Multiple	Multiple	Multiple
Split Transactions	No	Optional	Optional	Optional
Clocking	Async	Async	Sync	Either
Bandwidth, Single Word (0 ns mem)	25	37	20	5, 1.5
Bandwidth, Single Word (150 ns mem)	12.9	15.5	10	5, 1.5
Bandwidth Multiple Word (0 ns mem)	27.9	95.2	40	5, 1.5
Bandwidth Multiple Word (150 ns mem)	13.6	20.8	13.3	5, 1.5
Max # of devices	21	20	21	7
Max Bus Length	.5 m	.5 m	.5 m	25 m
Standard	IEEE 1014	IEEE 896	ANSI/IEEE 1296	ANSI X3.131

Distinctions begin to blur:

SCSI channel is like a bus

FutureBus is like a channel (disconnect/reconnect)

HIPPI forms links in high speed switching fabrics

RHK.S96 8

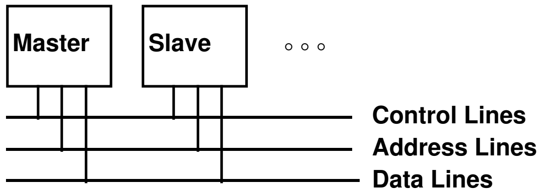
Bus-Based Interconnect

- **Bus: a shared communication link between subsystems**
  - Low cost: a single set of wires is shared multiple ways
  - Versatility: Easy to add new devices & peripherals may even be ported between computers using common bus
- **Disadvantage**
  - A communication bottleneck, possibly limiting the maximum I/O throughput
- **Bus speed is limited by physical factors**
  - the bus length
  - the number of devices (and, hence, bus loading).
  - these physical limits prevent arbitrary bus speedup.

Bus-Based Interconnect

- **Two generic types of busses:**
  - I/O busses: lengthy, many types of devices connected, wide range in the data bandwidth), and follow a bus standard (sometimes called a *channel*)
  - CPU–memory buses: high speed, matched to the memory system to maximize memory–CPU bandwidth, single device (sometimes called a *backplane*)
  - To lower costs, low cost (older) systems combine together
- **Bus transaction**
  - Sending address & receiving or sending data

Bus Protocols



Multibus: 20 address, 16 data, 5 control, 50ns Pause

Bus Master: has ability to control the bus, initiates transaction

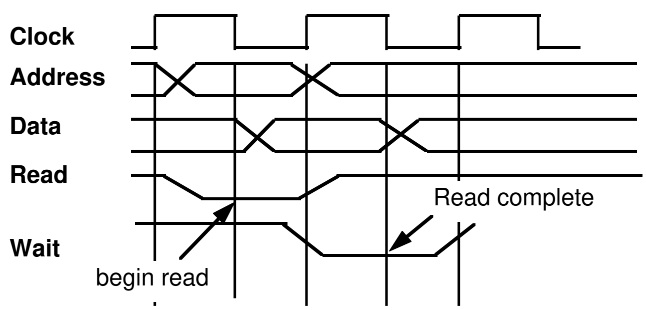
Bus Slave: module activated by the transaction

Bus Communication Protocol: specification of sequence of events and timing requirements in transferring information.

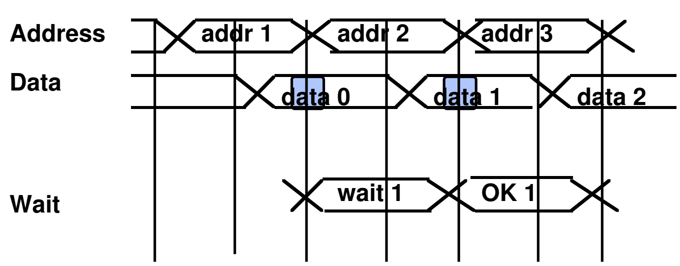
Asynchronous Bus Transfers: control lines (req., ack.) serve to orchestrate sequencing

Synchronous Bus Transfers: sequence relative to common clock

Synchronous Bus Protocols

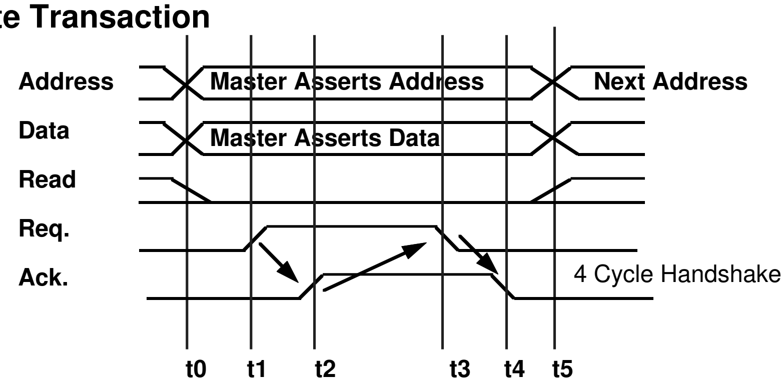


Pipelined/Split transaction Bus Protocol



Asynchronous Handshake

Write Transaction



t0 : Master has obtained control and asserts address, direction, data
Waits a specified amount of time for slaves to decode target\

t1: Master asserts request line

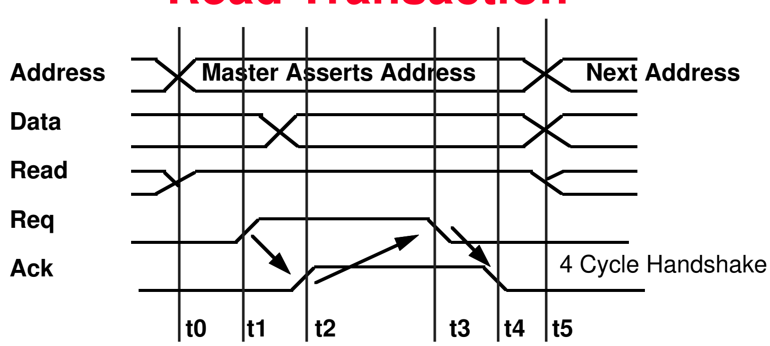
t2: Slave asserts ack, indicating data received

t3: Master releases req

t4: Slave releases ack

RHK.S96 13

Read Transaction



t0 : Master has obtained control and asserts address, direction, data
Waits a specified amount of time for slaves to decode target\

t1: Master asserts request line

t2: Slave asserts ack, indicating ready to transmit data

t3: Master releases req, data received

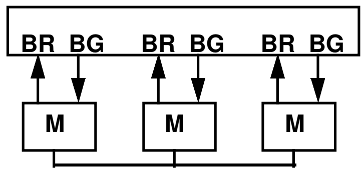
t4: Slave releases ack

Time Multiplexed Bus: address and data share lines

RHK.S96 14

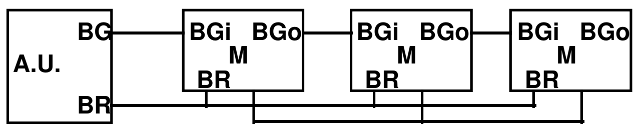
Bus Arbitration

Parallel (Centralized) Arbitration

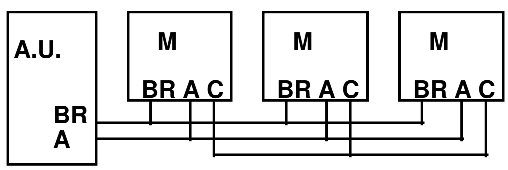


Bus Request
Bus Grant

Serial Arbitration (daisy chaining)



Polling



Bus Options

Option	*High performance*	*Low cost*
Bus width	Separate address & data lines	Multiplex address & data lines
Data width	Wider is faster (e.g., 32 bits)	Narrower is cheaper (e.g., 8 bits)
Transfer size	Multiple words has less bus overhead	Single-word transfer is simpler
Bus masters	Multiple (requires arbitration)	Single master (no arbitration)
Split transaction?	Yes—separate Request and Reply packets gets higher bandwidth (needs multiple masters)	No—continuous connection is cheaper and has lower latency
Clocking	Synchronous	Asynchronous

1990 Bus Survey (P&H, 1st Ed)

	VME	FutureBus	Multibus II	IPI	SCSI
Signals	128	96	96	16	8
Addr/Data mux	no	yes	yes	n/a	n/a
Data width	16 - 32	32	32	16	8
Masters	multi	multi	multi	single	multi
Clocking	Async	Async	Sync	Async	either
MB/s (0ns, word)	25	37	20	25	1.5 (asyn) 5 (sync)
150ns word	12.9	15.5	10	=	=
0ns block	27.9	95.2	40	=	=
150ns block	13.6	20.8	13.3	=	=
Max devices	21	20	21	8	7
Max meters	0.5	0.5	0.5	50	25
Standard	IEEE 1014	IEEE 896.1	ANSI/IEEE 1296	ANSI X3.129	ANSI X3.131

VME

- 3 96-pin connectors
- 128 defined as standard, rest customer defined
  - 32 address
  - 32 data
  - 64 command & power/ground lines

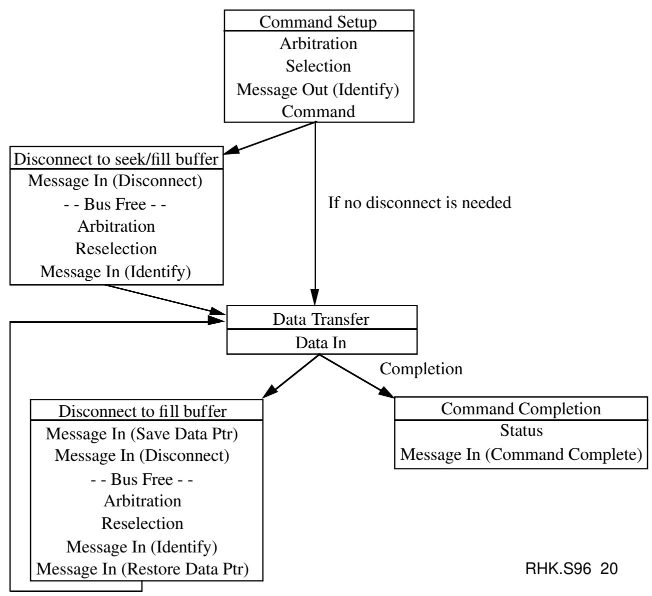
SCSI: Small Computer System Interface

- Up to 8 devices to communicate on a bus or “string” at sustained speeds of 4-5 MBytes/sec
- SCSI-2 up to 20 MB/sec
- Devices can be slave (“target”) or master(“initiator”)
- SCSI protocol: a series of ``phases", during which specific actions are taken by the controller and the SCSI disks
  - Bus Free: No device is currently accessing the bus
  - Arbitration: When the SCSI bus goes free, multiple devices may request (arbitrate for) the bus; fixed priority by address
  - Selection: informs the target that it will participate (Reselection if disconnected)
  - Command: the initiator reads the SCSI command bytes from host memory and sends them to the target
  - Data Transfer: data in or out, initiator: target
  - Message Phase: message in or out, initiator: target (identify, save/restore data pointer, disconnect, command complete)
  - Status Phase: target, just before command complete

RHK.S96 19

SCSI “Bus”: Channel Architecture

peer-to-peer protocols
initiator/target
linear byte streams
disconnect/reconnect



RHK.S96 20

1993 I/O Bus Survey (P&H, 2nd Ed)

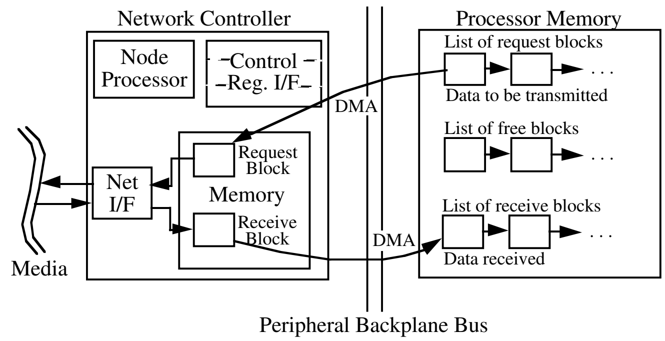
Bus	SBus	TurboChannel	MicroChannel	PCI
Originator	Sun	DEC	IBM	Intel
Clock Rate (MHz)	16-25	12.5-25	async	33
Addressing	Virtual	Physical	Physical	Physical
Data Sizes (bits)	8,16,32	8,16,24,32	8,16,24,32,64	8,16,24,32,64
Master	Multi	Single	Multi	Multi
Arbitration	Central	Central	Central	Central
32 bit read (MB/s)	33	25	20	33
Peak (MB/s)	89	84	75	111 (222)
Max Power (W)	16	26	13	25

1993 MP Server Memory Bus Survey

Bus	Summit	Challenge	XDBus
Originator	HP	SGI	Sun
Clock Rate (MHz)	60	48	66
Split transaction?	Yes	Yes	Yes?
Address lines	48	40	??
Data lines	128	256	144 (parity)
Data Sizes (bits)	512	1024	512
Clocks/transfer	4	5	4?
Peak (MB/s)	960	1200	1056
Master	Multi	Multi	Multi
Arbitration	Central	Central	Central
Addressing	Physical	Physical	Physical
Slots	16	9	10
Busses/system	1	1	2
Length	13 inches	12? inches	17 inches

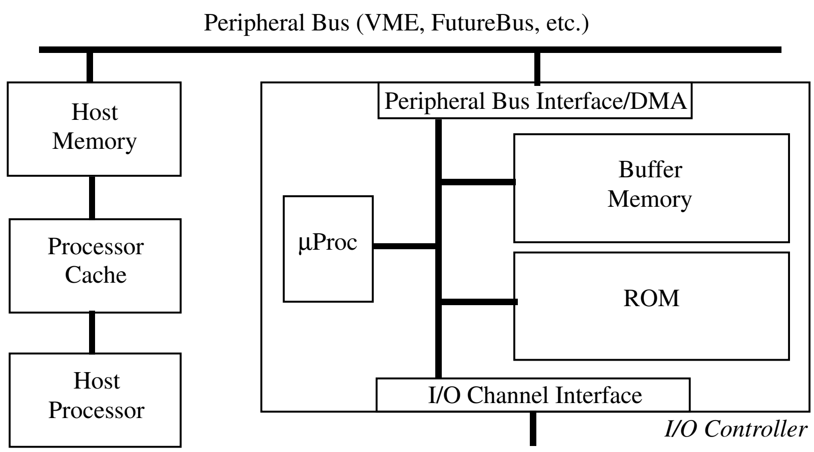
Communications Networks

Performance limiter is memory system, OS overhead, not protocols



- **Send/receive queues in processor memories**
- **Network controller copies back and forth via DMA**
- **No host intervention needed**
- **Interrupt host when message sent or received** RHK.S96 23

I/O Controller Architecture



Request/response block interface

Backdoor access to host memory

RHK.S96 24

I/O Data Flow

Impediment to high performance: multiple copies, complex hierarchy

	Application Address Space	Host Processor
Memory-to-Memory Copy		
	OS Buffers (>10 MByte)	
DMA over Peripheral Bus		
	HBA Buffers (1 M - 4 MBytes)	I/O Controller
Xfer over Disk Channel		
	Track Buffers (32K - 256KBytes)	Embedded Controller
Xfer over Serial Interface		
	I/O Device	Head/Disk Assembly