CS152
Computer Architecture and Engineering
Lecture 22

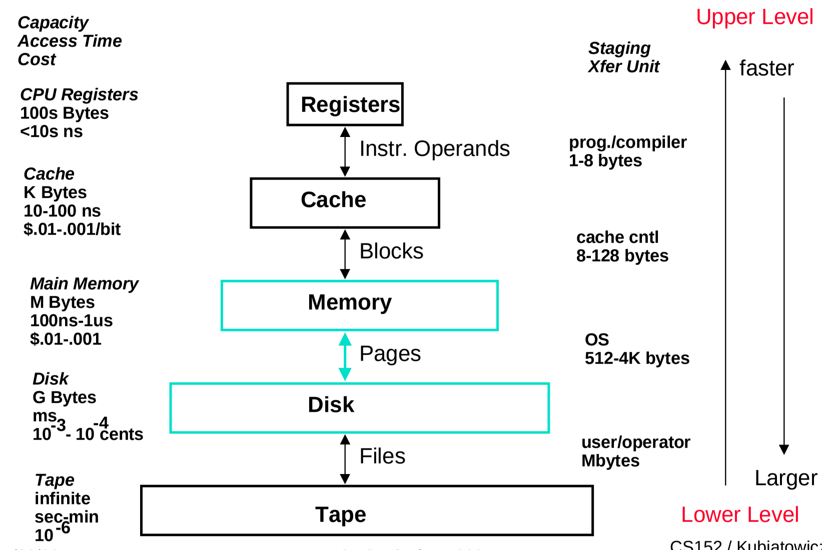
Buses and I/O
#1

April 26, 1999

John Kubiatowicz (http.cs.berkeley.edu/~kubitron)

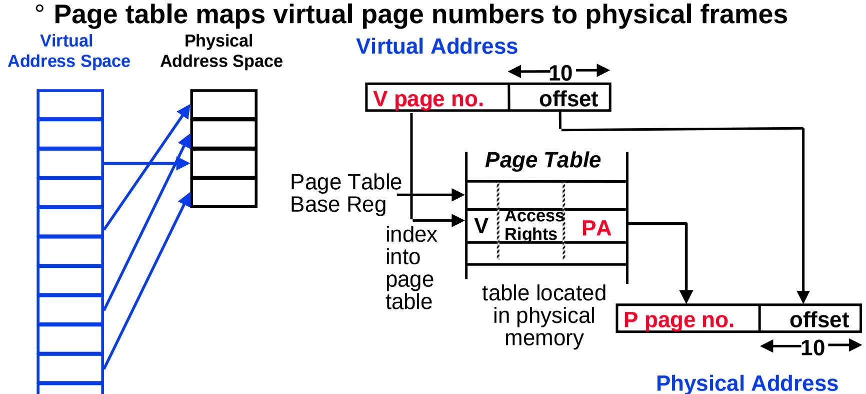
lecture slides: http://www-inst.eecs.berkeley.edu/~cs152/

Recap: Levels of the Memory Hierarchy



Recap: What is virtual memory?

- Virtual memory => treat memory as a cache for the disk
- Terminology: blocks in this cache are called "Pages"
- Typical size of a page: 1K — 8K
- Page table maps virtual page numbers to physical frames

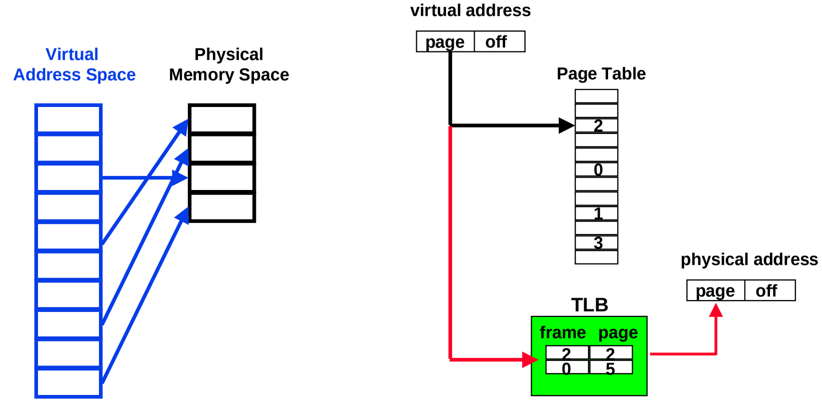


Recap: Three Advantages of Virtual Memory

- **Translation:**
  - Program can be given consistent view of memory, even though physical memory is scrambled
  - Makes multithreading reasonable (now used a lot!)
  - Only the most important part of program ("Working Set") must be in physical memory.
  - Contiguous structures (like stacks) use only as much physical memory as necessary yet still grow later.
- **Protection:**
  - Different threads (or processes) protected from each other.
  - Different pages can be given special behavior
    - (Read Only, Invisible to user programs, etc).
  - Kernel data protected from User programs
  - Very important for protection from malicious programs => Far more "viruses" under Microsoft Windows
- **Sharing:**
  - Can map same physical page to multiple users ("Shared memory")

Recap: Making address translation practical: TLB

- Translation Look-aside Buffer (TLB) is a cache of recent translations
- Speeds up translation process “most of the time”
- TLB is typically a fully-associative lookup-table



Recap: TLB organization: include protection

Virtual Address	Physical Address	Dirty	Ref	Valid	Access	ASID
0xFA00	0x0003	Y	N	Y	R/W	34
0x0040	0x0010	N	Y	Y	R	0
0x0041	0x0011	N	Y	Y	R	0

- TLB usually organized as fully-associative cache
  - Lookup is by Virtual Address
  - Returns Physical Address + other info
- Dirty => Page modified (Y/N)?
  Ref => Page touched (Y/N)?
  Valid => TLB entry valid (Y/N)?
  Access => Read? Write?
  ASID => Which User?

Recap: MIPS R3000 pipelining of TLB

MIPS R3000 Pipeline

Inst Fetch	Dcd/ Reg	ALU / E.A	Memory	Write Reg
TLB | I-Cache	RF	Operation		WB
		E.A. | TLB	D-Cache	

TLB

64 entry, on-chip, fully associative, software TLB fault handler

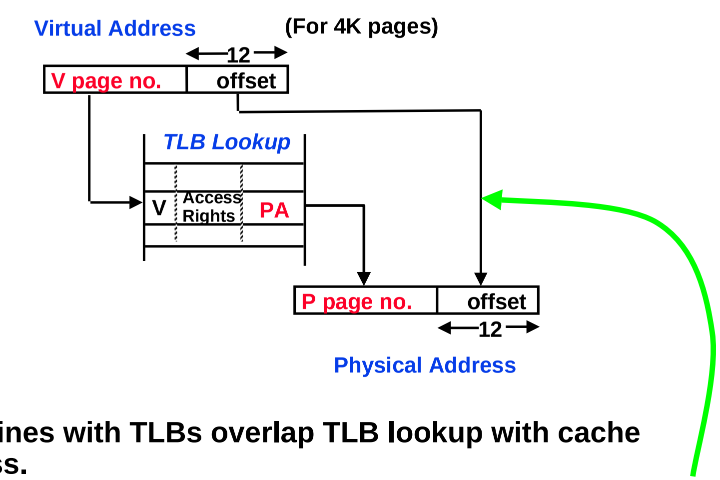
Virtual Address Space

ASID		V. Page Number	Offset
6		20	12

0xx User segment (caching based on PT/TLB entry)
100 Kernel physical space, cached
101 Kernel physical space, uncached
11x Kernel virtual space

Allows context switching among
64 user processes without TLB flush

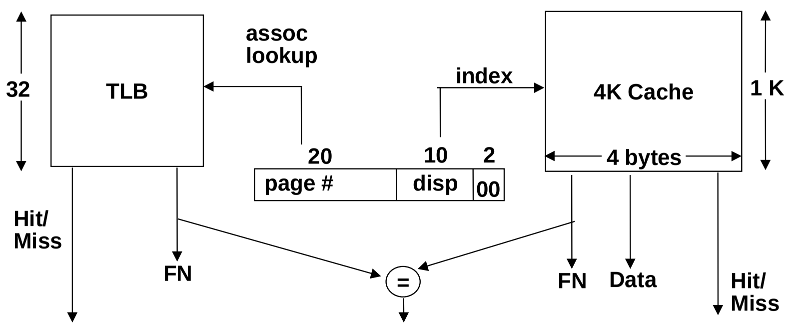
Reducing Translation Time I: Overlapped Access



- Machines with TLBs overlap TLB lookup with cache access.
  - Works because lower bits of result (offset) available early

Overlapped TLB & Cache Access

- If we do this in parallel, we have to be careful, however:



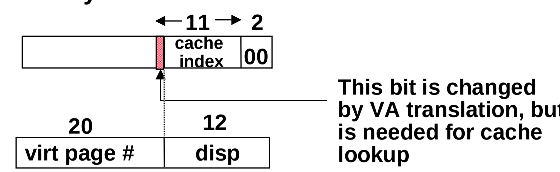
- With this technique, size of cache can be up to same size as pages.

⇒ What if we want a larger cache???

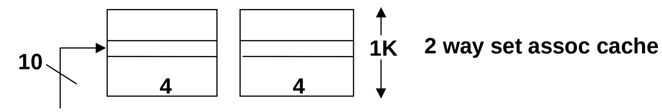
Problems With Overlapped TLB Access

- Overlapped access only works as long as the address bits used to index into the cache *do not change* as the result of VA translation

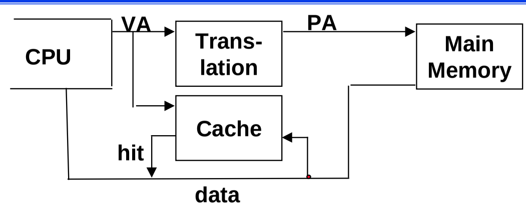
Example: suppose everything the same except that the cache is increased to 8 K bytes instead of 4 K:



- Solutions:
  - ⇒ Go to 8K byte page sizes;
  - ⇒ Go to 2 way set associative cache; or
  - ⇒ SW guarantee VA[13]=PA[13]



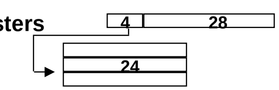
Reduced Translation Time II: Virtually Addressed Cache



- Only require address translation on cache miss!
  - Very fast as result (as fast as cache lookup)
  - No restrictions on cache organization
- *Synonym problem:* two different virtual addresses map to same physical address ⇒ two cache entries holding data for the same physical address!
- *Solutions:*
  - Provide associative lookup on physical tags during cache miss to enforce a single copy in the cache (potentially expensive)
  - Make operating system enforce one copy per cache set by selecting virtual⇒physical mappings carefully. This only works for direct mapped caches.
- Virtually Addressed caches currently out of favor because of synonym complexities

Survey

- **R4000**
  - 32 bit virtual, 36 bit physical
  - variable page size (4KB to 16 MB)
  - 48 entries mapping page pairs (128 bit)
- **MPC601 (32 bit implementation of 64 bit PowerPC arch)**
  - 52 bit virtual, 32 bit physical, 16 segment registers
  - 4KB page, 256MB segment
  - 4 entry instruction TLB
  - 256 entry, 2-way TLB (and variable sized block xlate)
  - overlapped lookup into 8-way 32KB L1 cache
  - hardware table search through hashed page tables



- **Alpha 21064**
  - arch is 64 bit virtual, implementation subset: 43, 47,51,55 bit
  - 8,16,32, or 64KB pages (3 level page table)
  - 12 entry ITLB, 32 entry DTLB
  - 43 bit virtual, 28 bit physical octword address

Alpha VM Mapping

- "64-bit" address divided into 3 segments
  - seg0 (bit 63=0) user code/heap
  - seg1 (bit 63 = 1, 62 = 1) user stack
  - kseg (bit 63 = 1, 62 = 0) kernel segment for OS
- 3 level page table, each one page
  - Alpha only 43 unique bits of VA
  - (future min page size up to 64KB => 55 bits of VA)
- PTE bits; valid, kernel & user read & write enable (No reference, use, or dirty bit)

Administrivia

- Important: Design for Test
  - You should be testing from the very start of your design
  - Consider adding special monitor modules at various points in design => I have asked you to label trace output from these modules with the current clock cycle #
  - The time to understand how components of your design should work is while you are designing!
- Question: Oral reports on 5/11?
  - Proposal: 10 — 12 am and 2 — 4 pm
- Pending schedule:
  - Friday 4/30: Update on design
  - Today 4/26 and Wednesday 4/28
  - Monday 5/3: no class
  - Wednesday 5/5: Multiprocessing
  - Monday 5/10 Last class (wrap up, evaluations, etc)
  - Tuesday 5/11 by 5pm: final project reports due.
  - Friday 5/14 grades should be posted.

Administrivia II

- Major organizational options:
  - 2-way superscalar (18 points)
  - 2-way multithreading (20 points)
  - 2-way multiprocessor (18 points)
  - out-of-order execution (22 points)
  - Deep Pipelined (12 points)
- Test programs will include multiprocessor versions
- Both multiprocessor and multithreaded must implement synchronizing "Test and Set" instruction:
  - use "coprocessor zero" load instruction (`lwc0 $1,$2,$3`)
  - Reads and returns old value of memory location at specified address, while setting the value to one (stall memory stage for one extra cycle).
  - For multiprocessor, this instruction must make sure that all updates to this address are suspended during operation.
  - For multithreaded, switch to other processor if value is already non-zero (like a cache miss).

Computers in the News: Sony Playstation 2000

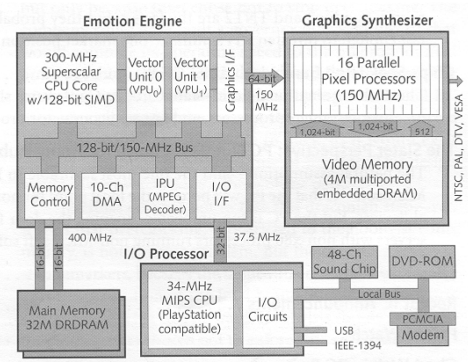


Figure 1. PlayStation 2000 employs an unprecedented level of parallelism to achieve workstation-class 3D performance.

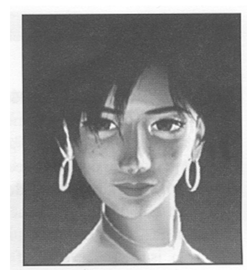


Figure 2. PlayStation 2000 screenshot. (Source: Namco)

- (as reported in Microprocessor Report, Vol 13, No. 5)
  - Emotion Engine: 6.2 GFLOPS, 75 million polygons per second
  - Graphics Synthesizer: 2.4 Billion pixels per second
  - Claim: *Toy Story* realism brought to games!

Playstation 2000 Continued

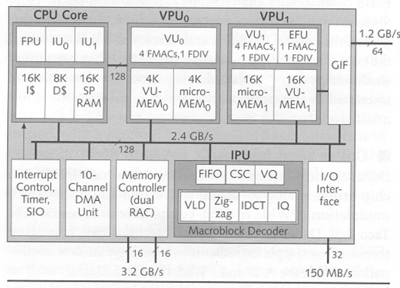


Figure 3. The PSX2's Emotion Engine provides ten floating-point multiplier-accumulators, four floating-point dividers, and an MPEG-2 decoder to deliver killer multimedia performance.

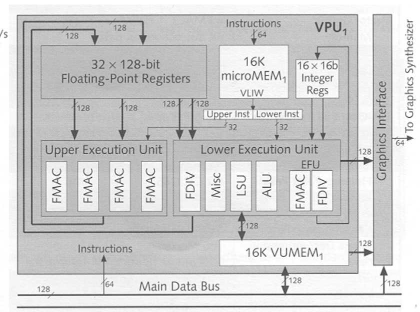
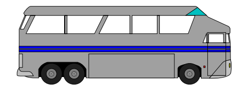


Figure 5. Each vector unit has enough parallelism to complete a vertex operation (19 mul-adds + 1 divide) every seven cycles.

° Emotion Engine:
- Superscalar MIPS core
- Vector Coprocessor Pipelines
- RAMBUS DRAM interface

° Sample Vector Unit
- 2-wide VLIW
- Includes Microcode Memory
- High-level instructions like matrix-multiply

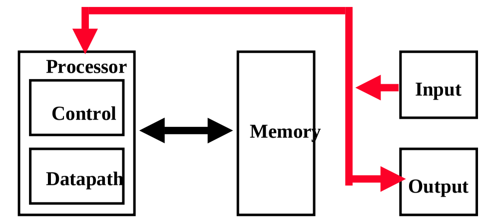
What is a bus?



A Bus Is:

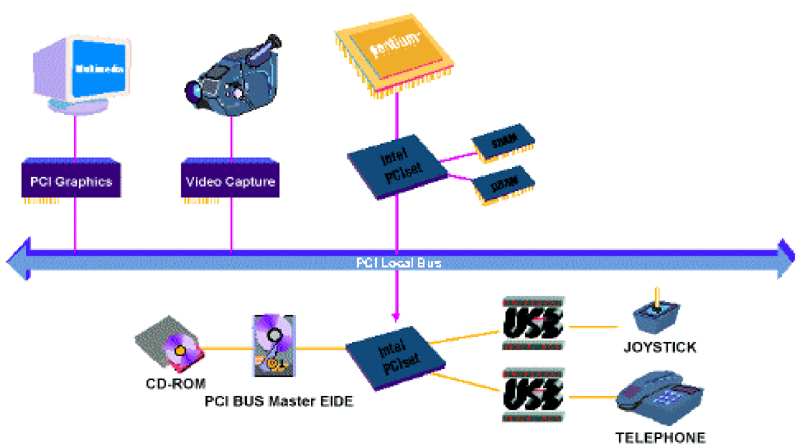
° shared communication link

° single set of wires used to connect multiple subsystems

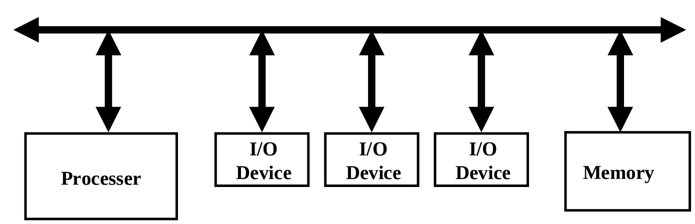


° A Bus is also a fundamental tool for composing large, complex systems
- systematic means of abstraction

Buses



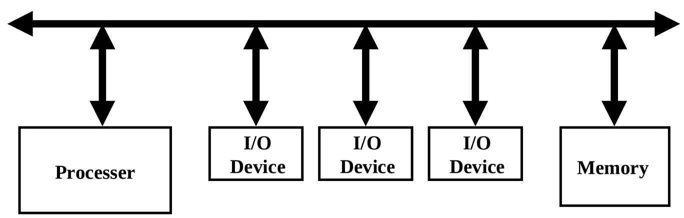
Advantages of Buses



° Versatility:
- New devices can be added easily
- Peripherals can be moved between computer systems that use the same bus standard

° Low Cost:
- A single set of wires is shared in multiple ways

Disadvantage of Buses

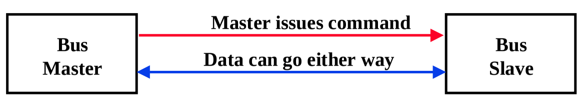


- It creates a communication bottleneck
  - The bandwidth of that bus can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus
  - The need to support a range of devices with:
    - Widely varying latencies
    - Widely varying data transfer rates

The General Organization of a Bus

- Control lines:
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- Data lines carry information between the source and the destination:
  - Data and Addresses
  - Complex commands

Master versus Slave



- A bus transaction includes two parts:
  - Issuing the command (and address) – request
  - Transferring the data – action
- Master is the one who starts the bus transaction by:
  - issuing the command (and address)
- Slave is the one who responds to the address by:
  - Sending data to the master if the master ask for data
  - Receiving data from the master if the master wants to send data

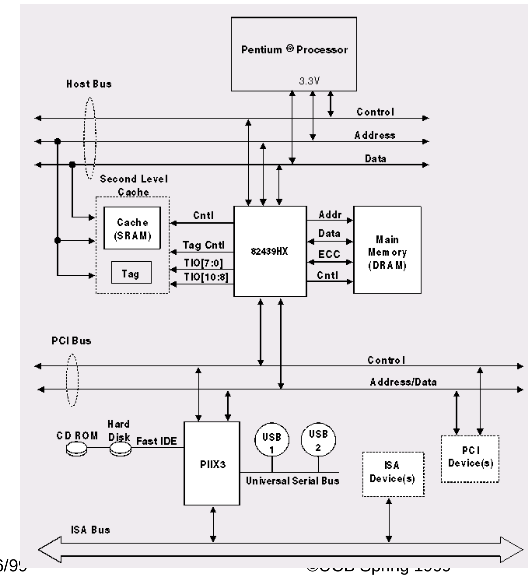
What is DMA (Direct Memory Access)?

- Typical I/O devices must transfer large amounts of data to memory of processor:
  - Disk must transfer complete block (4K? 16K?)
  - Large packets from network
  - Regions of frame buffer
- DMA gives external device ability to write memory directly: much lower overhead than having processor request one word at a time.
  - Processor (or at least memory system) acts like slave
- Issue: Cache coherence:
  - What if I/O devices write data that is currently in processor Cache?
    - The processor may never see new data!
  - Solutions:
    - Flush cache on every I/O operation (expensive)
    - Have hardware invalidate cache lines (remember "Coherence" cache misses?)

Types of Buses

- Processor-Memory Bus (design specific)
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
  - Optimized for cache block transfers
- I/O Bus (industry standard)
  - Usually is lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus
- Backplane Bus (standard or proprietary)
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one bus for all components

Example: Pentium System Organization

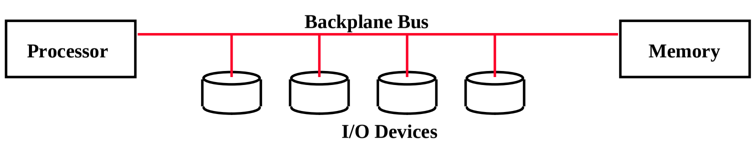


Processor/Memory Bus

PCI Bus

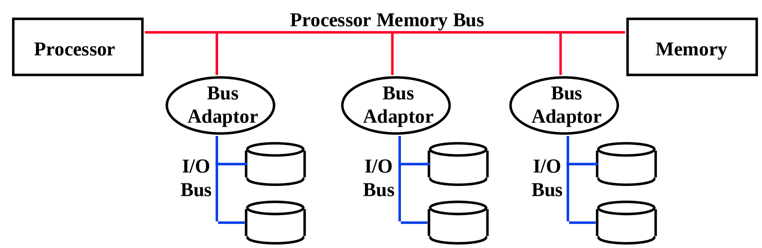
I/O Busses

A Computer System with One Bus: Backplane Bus



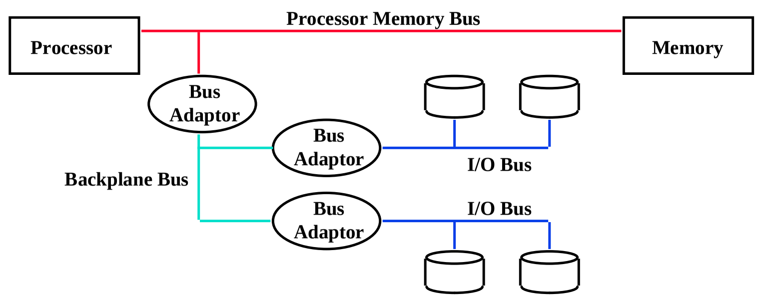
- A single bus (the backplane bus) is used for:
  - Processor to memory communication
  - Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: IBM PC - AT

A Two-Bus System



- I/O buses tap into the processor-memory bus via bus adaptors:
  - Processor-memory bus: mainly for processor-memory traffic
  - I/O buses: provide expansion slots for I/O devices
- Apple Macintosh-II
  - NuBus: Processor, memory, and a few selected I/O devices
  - SCCI Bus: the rest of the I/O devices

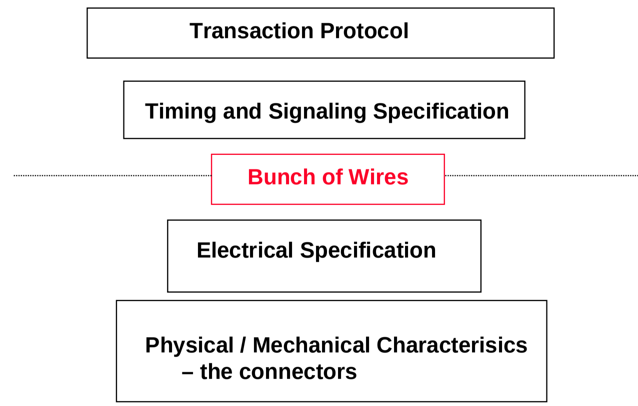
A Three-Bus System



° A small number of backplane buses tap into the processor-memory bus
  • Processor-memory bus is only used for processor-memory traffic
  • I/O buses are connected to the backplane bus

° Advantage: loading on the processor bus is greatly reduced

What defines a bus?

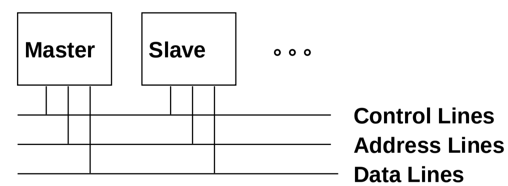


Synchronous and Asynchronous Bus

° Synchronous Bus:
  • Includes a clock in the control lines
  • A fixed protocol for communication that is relative to the clock
  • Advantage: involves very little logic and can run very fast
  • Disadvantages:
    - Every device on the bus must run at the same clock rate
    - To avoid clock skew, they cannot be long if they are fast

° Asynchronous Bus:
  • It is not clocked
  • It can accommodate a wide range of devices
  • It can be lengthened without worrying about clock skew
  • It requires a handshaking protocol

Busses so far



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.

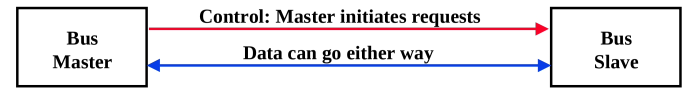
Asynchronous Bus Transfers: control lines (req, ack) serve to orchestrate sequencing.

Synchronous Bus Transfers: sequence relative to common clock.

Bus Transaction

- Arbitration: Who gets the bus
- Request: What do we want to do
- Action: What happens in response

Arbitration: Obtaining Access to the Bus

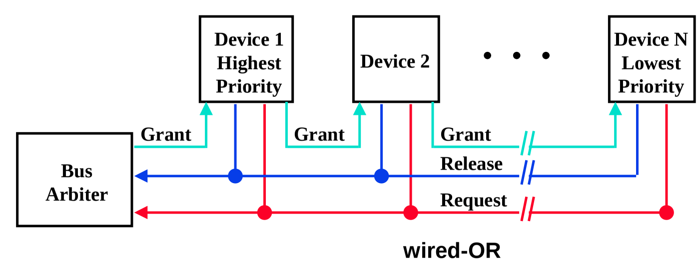


- One of the most important issues in bus design:
  - How is the bus reserved by a device that wishes to use it?
- Chaos is avoided by a master-slave arrangement:
  - Only the bus master can control access to the bus:
    It initiates and controls all bus requests
  - A slave responds to read and write requests
- The simplest system:
  - Processor is the only bus master
  - All bus requests must be controlled by the processor
  - Major drawback: the processor is involved in every transaction

Multiple Potential Bus Masters: the Need for Arbitration

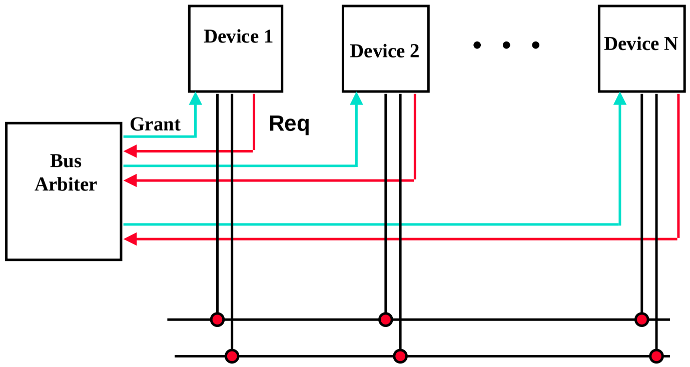
- Bus arbitration scheme:
  - A bus master wanting to use the bus asserts the bus request
  - A bus master cannot use the bus until its request is granted
  - A bus master must signal to the arbiter after finish using the bus
- Bus arbitration schemes usually try to balance two factors:
  - Bus priority: the highest priority device should be serviced first
  - Fairness: Even the lowest priority device should never
    be completely locked out from the bus
- Bus arbitration schemes can be divided into four broad classes:
  - Daisy chain arbitration
  - Centralized, parallel arbitration
  - Distributed arbitration by self-selection: each device wanting the bus places a code indicating its identity on the bus.
  - Distributed arbitration by collision detection: Ethernet uses this.

The Daisy Chain Bus Arbitrations Scheme



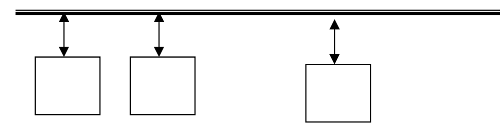
- Advantage: simple
- Disadvantages:
  - Cannot assure fairness:
    A low-priority device may be locked out indefinitely
  - The use of the daisy chain grant signal also limits the bus speed

Centralized Parallel Arbitration



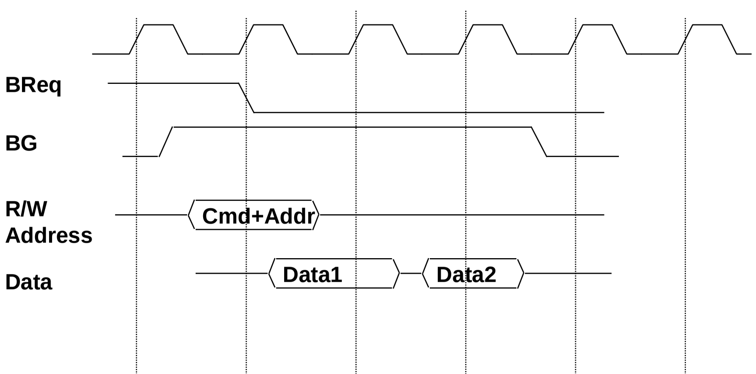
- Used in essentially all processor-memory busses and in high-speed I/O busses

Simplest bus paradigm



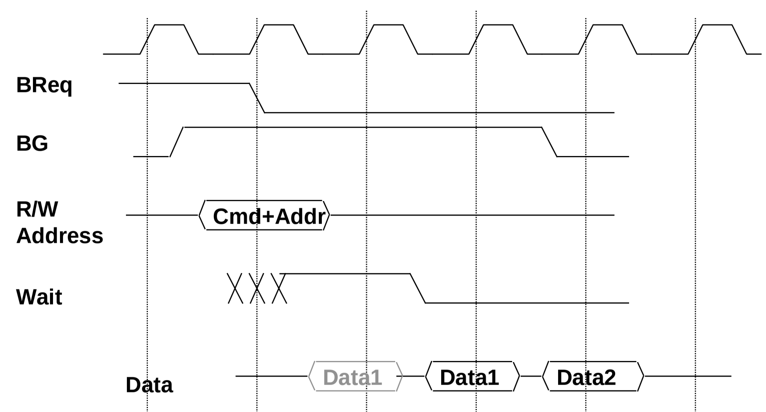
- All agents operate synchronously
- All can source / sink data at same rate
- => simple protocol
  - just manage the source and target

Simple Synchronous Protocol



- Even memory busses are more complex than this
  - memory (slave) may take time to respond
  - it may need to control data rate

Typical Synchronous Protocol



- Slave indicates when it is prepared for data xfer
- Actual transfer goes at bus rate

Increasing the Bus Bandwidth

- Separate versus multiplexed address and data lines:
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available
  - Cost: (a) more bus lines, (b) increased complexity
- Data bus width:
  - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles
  - Example: SPARCstation 20's memory bus is 128 bit wide
  - Cost: more bus lines
- Block transfers:
  - Allow the bus to transfer multiple words in back-to-back bus cycles
  - Only one address needs to be sent at the beginning
  - The bus is not released until the last word is transferred
  - Cost: (a) increased complexity
    (b) decreased response time for request

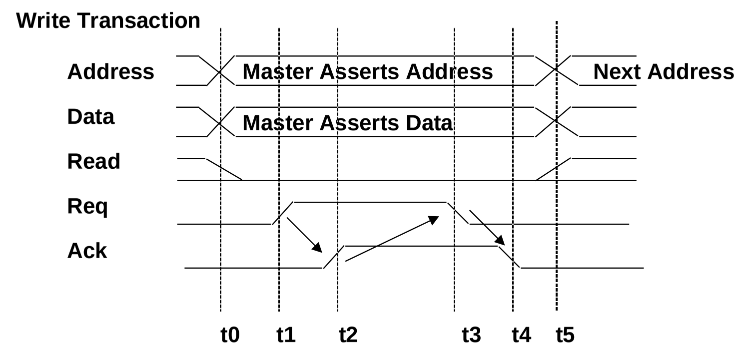
Increasing Transaction Rate on Multimaster Bus

- Overlapped arbitration
  - perform arbitration for next transaction during current transaction
- Bus parking
  - master can holds onto bus and performs multiple transactions as long as no other master makes request
- Overlapped address / data phases (prev. slide)
  - requires one of the above techniques
- Split-phase (or packet switched) bus
  - completely separate address and data phases
  - arbitrate separately for each
  - address phase yield a tag which is matched with data phase
- "All of the above" in most modern mem busses

1993 MP Server Memory Bus Survey: GTL revolution

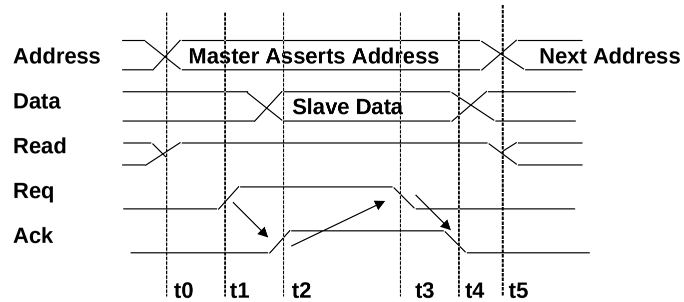
Bus	MBus	Summit	Challenge	XDBus
Originator	Sun	HP	SGI	Sun
Clock Rate (MHz)	40	60	48	66
Address lines	36	48	40	muxed
Data lines	64	128	256	144 (parity)
Data Sizes (bits)	256	512	1024	512
Clocks/transfer		4	5	4?
Peak (MB/s)	320(80)	960	1200	1056
Master	Multi	Multi	Multi	Multi
Arbitration	Central	Central	Central	Central
Slots		16	9	10
Busses/system	1	1	1	2
Length		13 inches	12? inches	17 inches

Asynchronous Handshake



- 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

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

1993 Backplane/IO Bus Survey

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

High Speed I/O Bus

- Examples
  - graphics
  - fast networks
- Limited number of devices
- Data transfer bursts at full rate
- DMA transfers important
  - small controller spools stream of bytes to or from memory
- Either side may need to squelch transfer
  - buffers fill up

PCI Read/Write Transactions

- All signals sampled on rising edge
- Centralized Parallel Arbitration
  - overlapped with previous transaction
- All transfers are (unlimited) bursts
- Address phase starts by asserting FRAME#
- Next cycle “initiator” asserts cmd and address
- Data transfers happen on when
  - IRDY# asserted by master when ready to transfer data
  - TRDY# asserted by target when ready to transfer data
  - transfer when both asserted on rising edge
- FRAME# deasserted when master intends to complete only one more data transfer

PCI Read Transaction

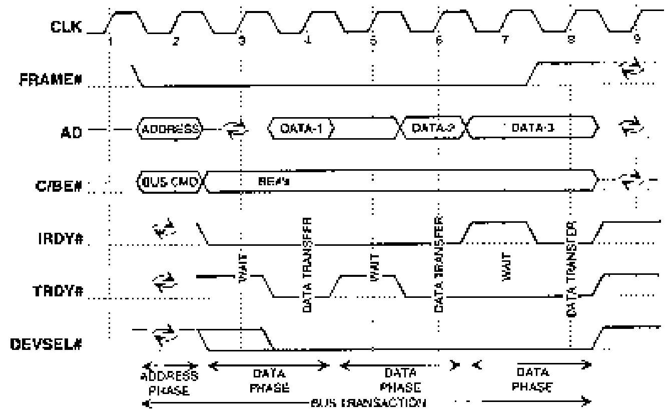


Figure 3-1: Basic Read Operation

– Turn-around cycle on any signal driven by more than one agent

PCI Write Transaction

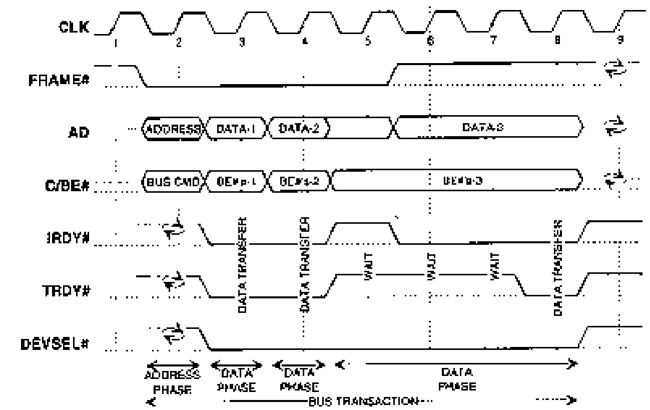


Figure 3-2: Basic Write Operation

PCI Optimizations

- Push bus efficiency toward 100% under common simple usage
  - like RISC
- Bus Parking
  - retain bus grant for previous master until another makes request
  - granted master can start next transfer without arbitration
- Arbitrary Burst length
  - initiator and target can exert flow control with xRDY
  - target can disconnect request with STOP (abort or retry)
  - master can disconnect by deasserting FRAME
  - arbiter can disconnect by deasserting GNT
- Delayed (pended, split-phase) transactions
  - free the bus after request to slow device

Summary

- Buses are an important technique for building large-scale systems
  - Their speed is critically dependent on factors such as length, number of devices, etc.
  - Critically limited by capacitance
  - Tricks: esoteric drive technology such as GTL
- Important terminology:
  - Master: The device that can initiate new transactions
  - Slaves: Devices that respond to the master
- Two types of bus timing:
  - Synchronous: bus includes clock
  - Asynchronous: no clock, just REQ/ACK strobing
- Direct Memory Access (dma) allows fast, burst transfer into processor's memory:
  - Processor's memory acts like a slave
  - Probably requires some form of cache-coherence so that DMA'ed memory can be invalidated from cache.

Summary of 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)**
Clocking	**Synchronous**	**Asynchronous**