Busses
&
Virtual Memory

COMP375 Computer Architecture and Organization

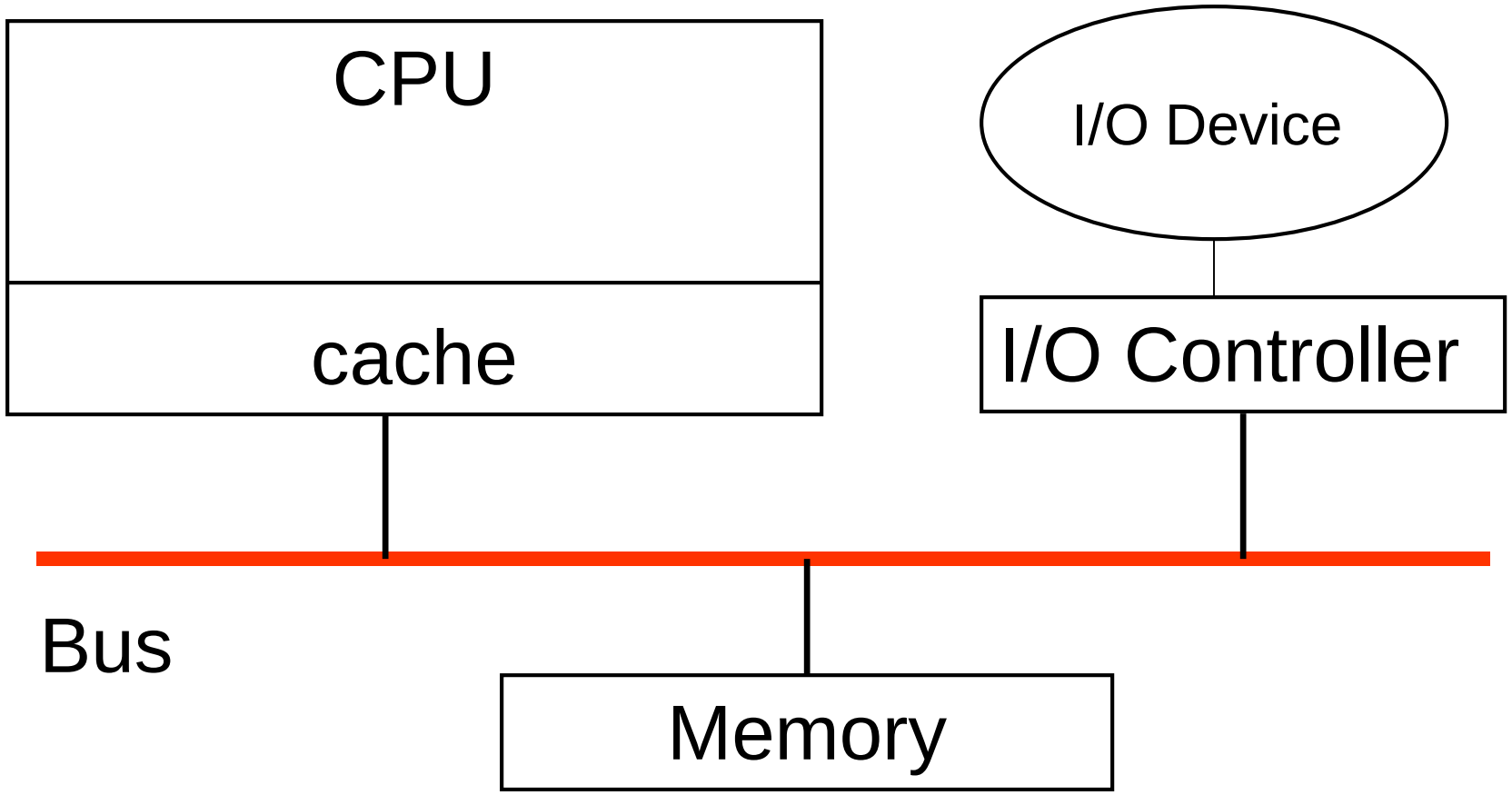
“The question of whether a computer can think is no more interesting than the question of whether a submarine can swim.”

Edsger Dijkstra

Practice Coding Interviews with Google

- Join the Google in Residence Interview Coaching program
- Google is hosting two on-campus interview prep and mock interview group sessions where you'll practice with a Googler and learn how to think through problems.
- Interview Prep Session: (6:00-9:00 PM in 207 Cherry Hall) November 11
- Group Mock Interview Session November 12
- RSVP @ goo.gle/2IX1E2D

Basic Computer Components



Bus Design Issues

- Dedicated or Multiplexed
- Width
- Access Protocol
- Arbitration
- Timing
- Operation

Fetch-Store Paradigm

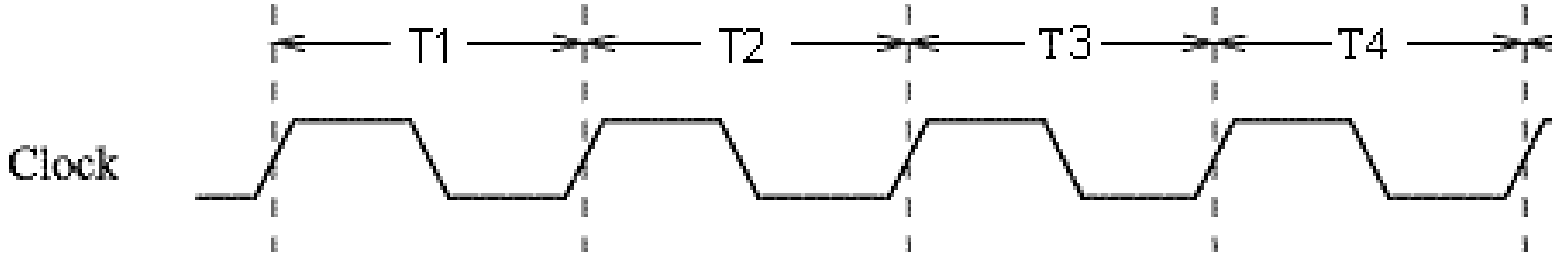
- A processor can fetch a value from memory or store a value to memory
- Fetch and store are also used to transfer data to an I/O device
- Only one device at a time can put a value on the bus data or address lines

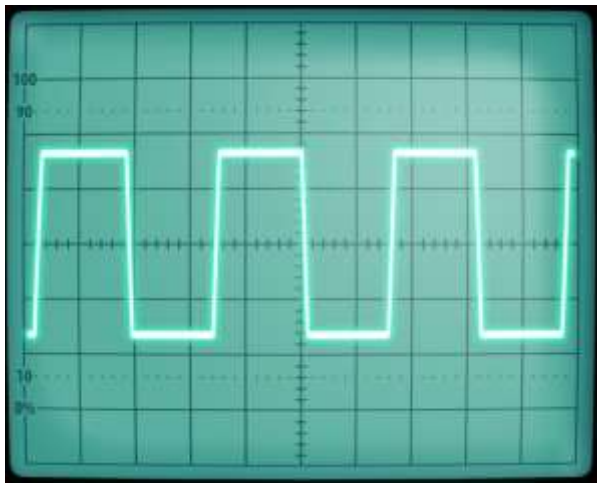
Block Transfers

- With cache systems, memory requests to the RAM are for a whole line of data
- The CPU requests an address and the RAM provides a series of data values
- I/O controllers may still communicate with the CPU or the memory with arbitrarily sized data

Timing Diagrams - Clock

- There is a horizontal line for each signal
- A synchronous bus has a clock signal that synchronizes operations
- Time goes from left to right

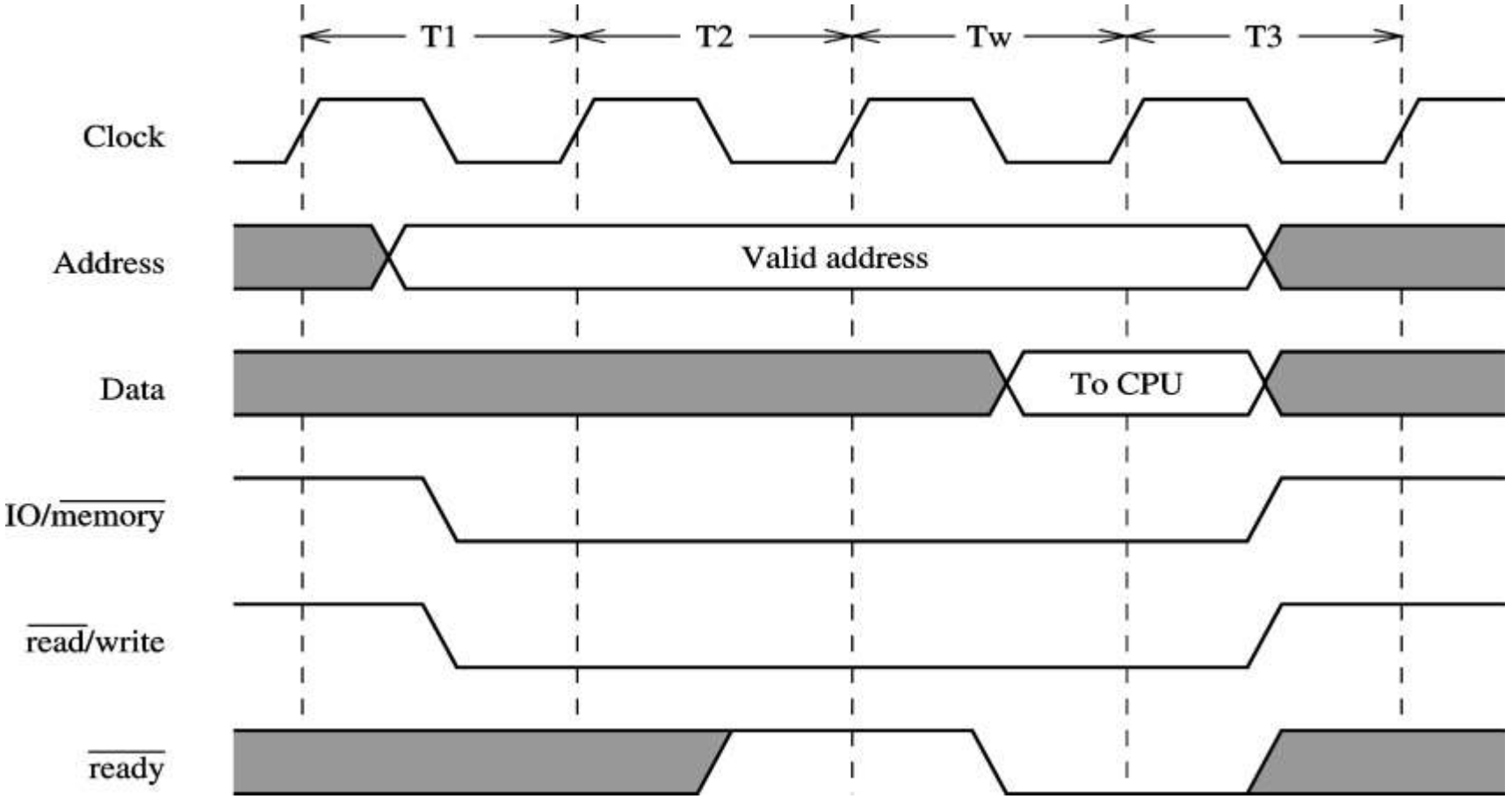




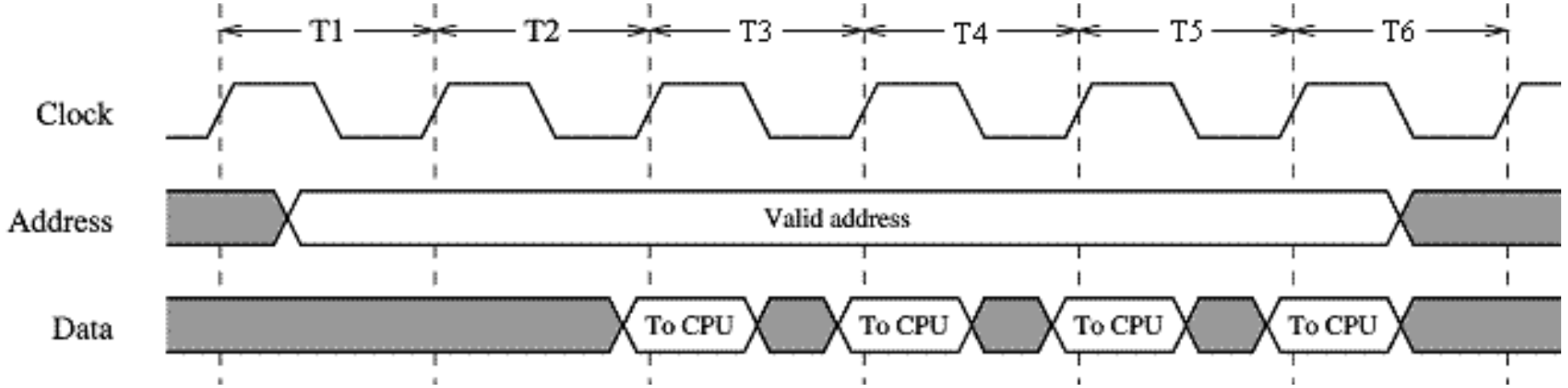
Timing Diagrams – Data Lines

- Multiple data or address lines are often shown as a block
- Data or address lines can be
  - all one or all zero
  - mixed with some zero and some one
  - undetermined, value is unimportant

Memory read with a wait state



Block transfer of data



If the bus has a 1.0 GHz clock, how long is each clock cycle?

A. 1.0 microsecond
B. 1.0 millisecond
C. 1.0 nanosecond
D. 0.5 nanosecond

Cycles and Bus Width

- If a bus is **n** bits wide, it can transfer **n** bits (or **n**/8 bytes) every cycle
- If you need to transfer more than **n** bits, it will take multiple cycles
- To transfer **x** bytes over an **n** bit wide bus, it takes **8x / n** cycles plus any cycles previous to the data transfer

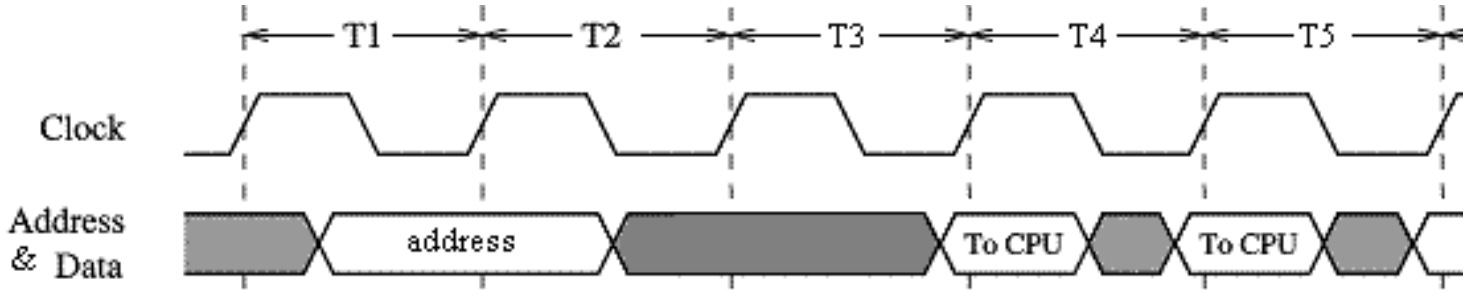
How many clock cycles does it take to transfer 96 bytes over a 128 bit wide bus?

- A 128 bit wide bus can transfer 128/8 = 16 bytes each cycle
- 96 bytes / 16 bytes/cycle = 6 cycles
- If there was a 4 cycle delay before the data was available, the total time would be 6 + 4 = 10 cycles

How many clock cycles does it take to transfer 64 bytes over a 64 bit wide bus?

A. 1 cycle
B. 2 cycles
C. 4 cycles
D. 8 cycle
E. 16 cycles

How many bus clock cycles does it take from when the CPU first requests the data until 32 bytes have been read?



A bus protocol requires the CPU to put the address on the 64 bit wide bus during the first clock period.

On the fourth clock period and every clock period afterwards data is retrieved from the RAM.

A. 2 cycles
B. 3 cycles
C. 4 cycles
D. 7 cycles
E. 8 cycles

How much time does it take for 7 cycles on a 1.0 GHz bus?

A. 7.0 nsec
B. 14.0 nsec
C. 3.5 nsec
D. 7.0 µsec

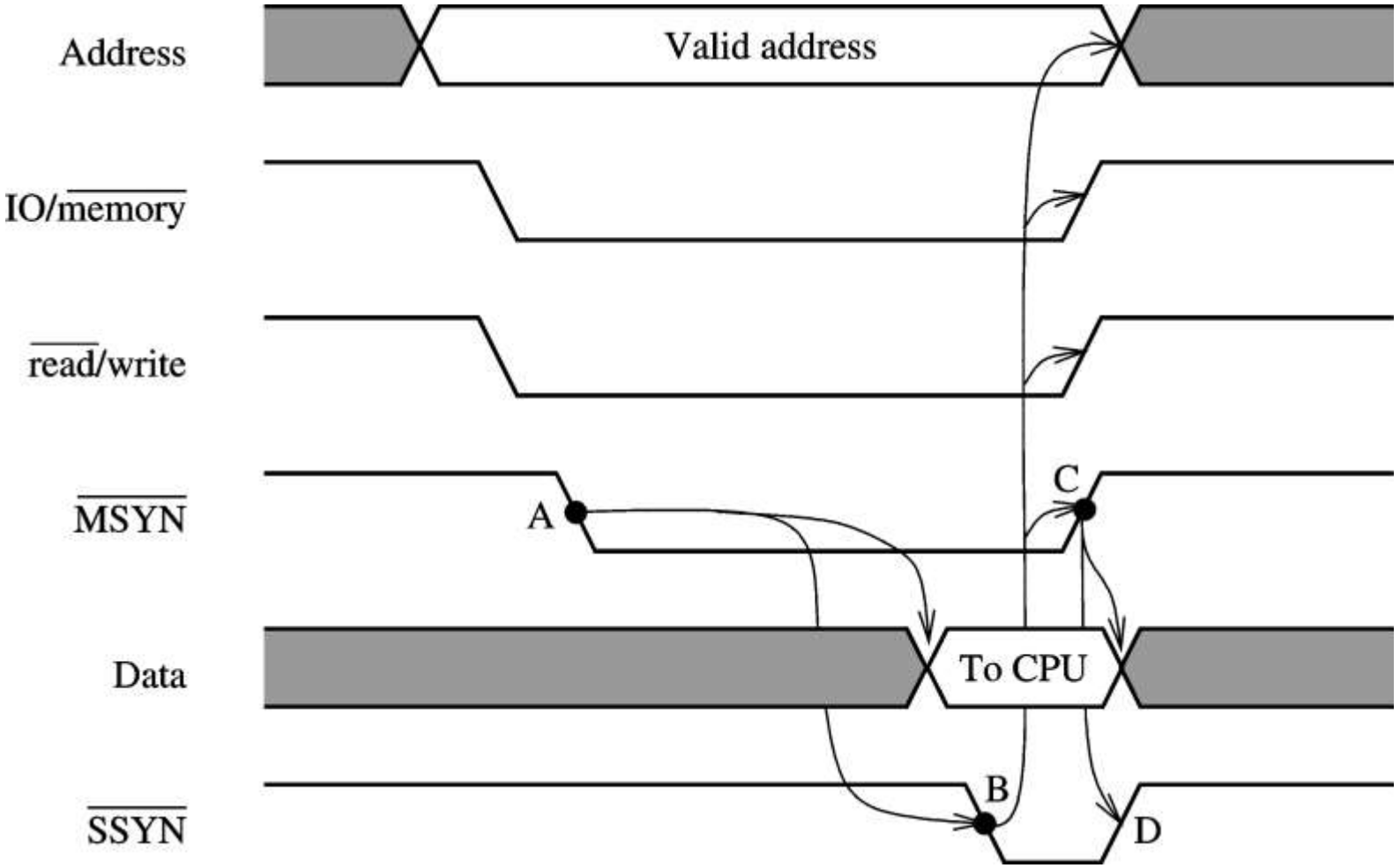
If every instruction requires two memory access of 7.0 nsec, what is the maximum execution speed in MIPS?

A. 71 MIPS
B. 11 MIPS
C. 45 MIPS
D. 22 MIPS
E. Cannot be determined

Synchronous or Asynchronous

- In a synchronous bus, a clock signal provides timing for all operations
- A device presents the address on a given clock pulse and expects the data during another predefined clock pulse
- In an asynchronous bus, a device waits for a ready signal to know when data is available

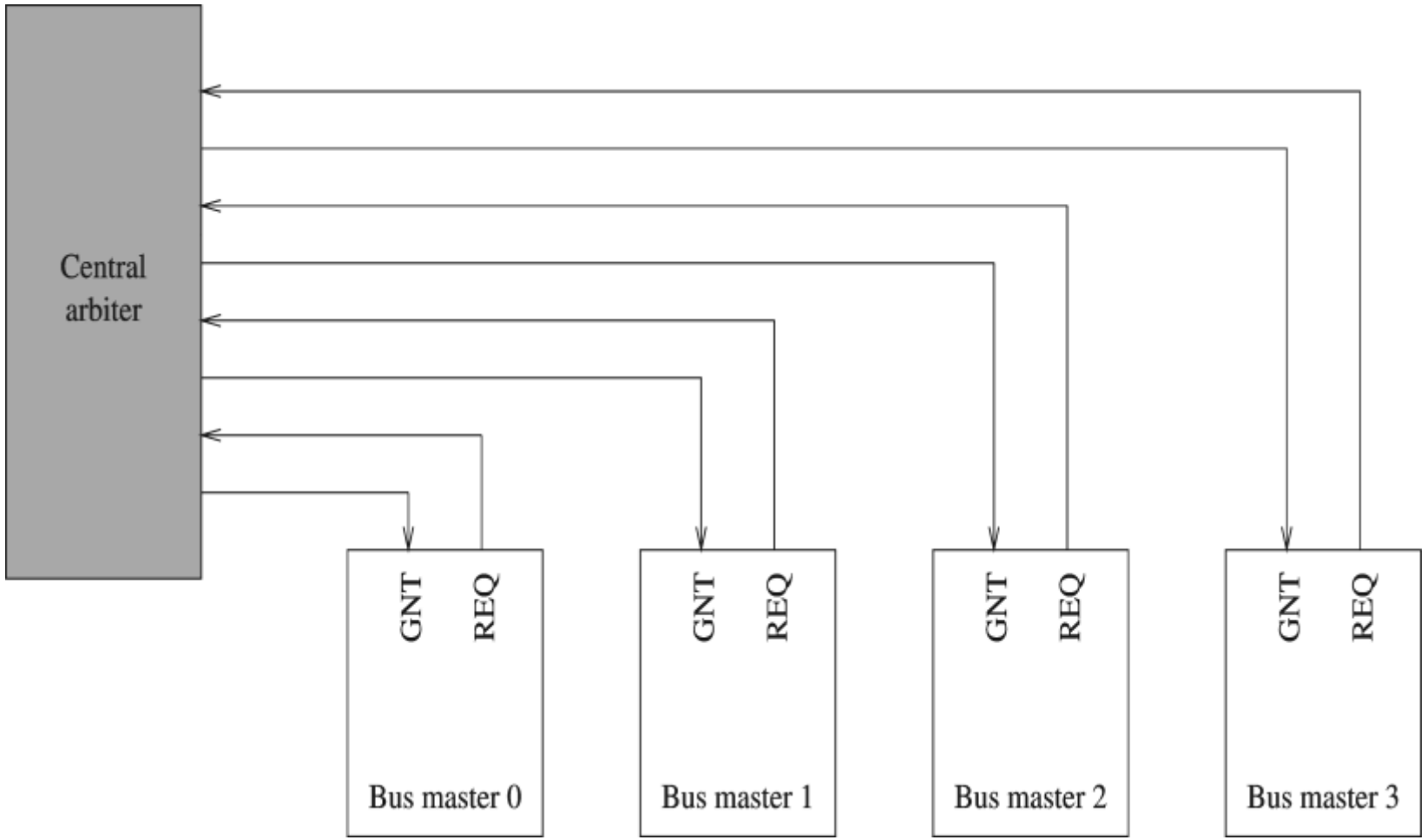
Asynchronous Bus



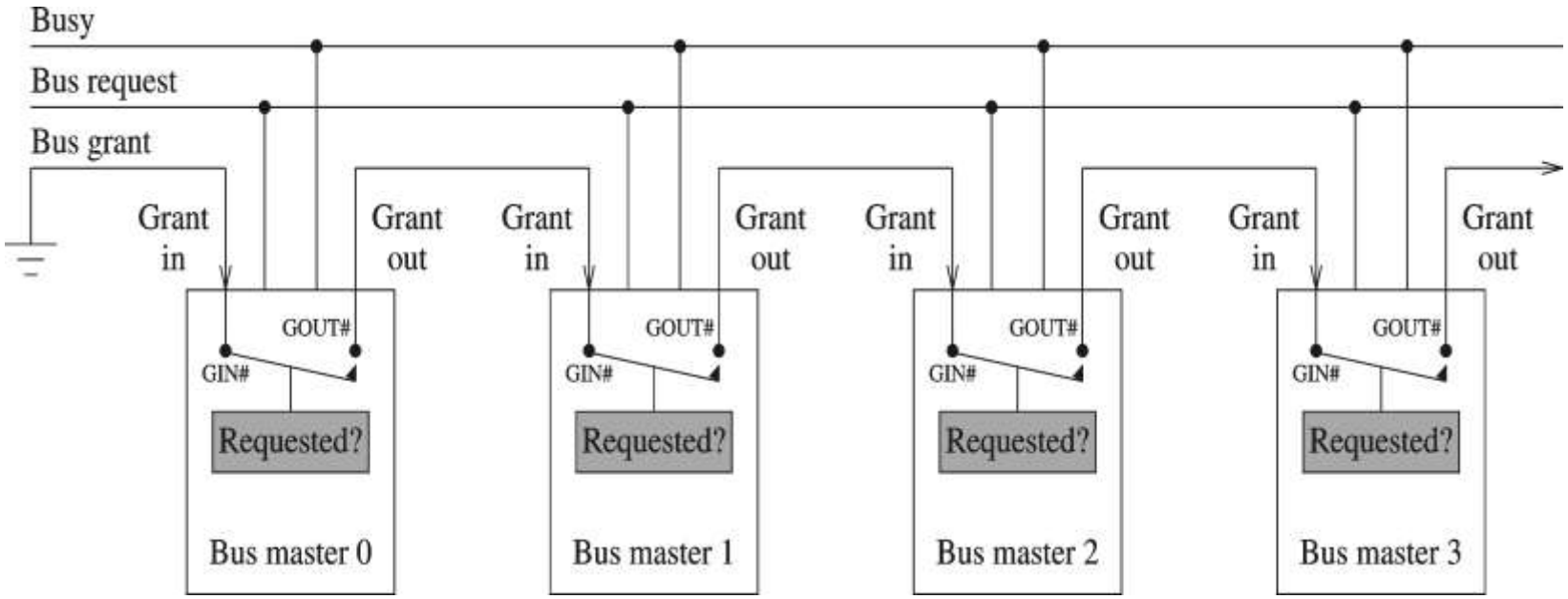
Arbitration

- Only one device can put data on the bus at a time. Many devices can sense the data, but only one can assert it
- The bus arbitration protocol determines which device gets to use the bus at any given time
- Bus arbitration can be centralized or distributed

Centralized Arbitration



Daisy Chain Bus



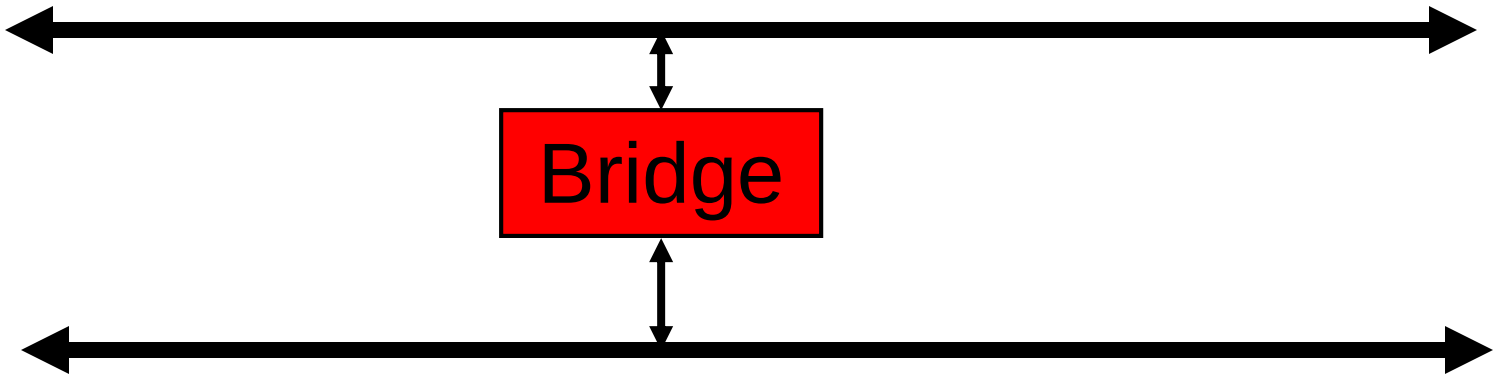
The devices determine who gets to use the bus

Multiple Buses

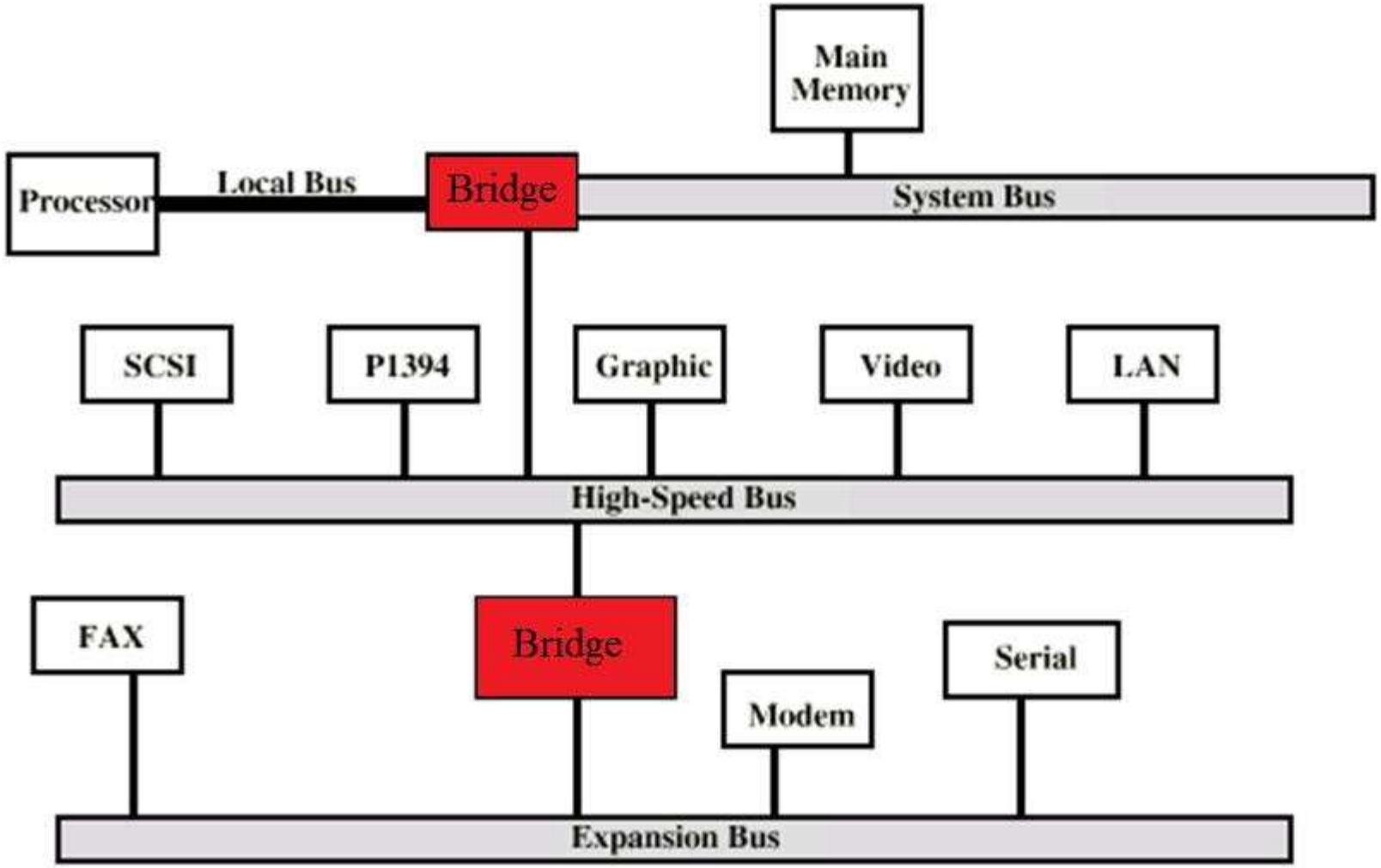
- A single computer usually has several buses.
- Different devices have different requirements
- A 56K modem only needs about 7 KB/sec bandwidth while a graphics device may need 70 MB/sec or more
- Multiple buses allow devices using different technologies to connect to the same computer

Bridging Buses

- A bridge is a device that connects two buses
- A bridge converts the addresses and protocols of one bus to another



Bus Hierarchy



Chipsets

- The chipset controls the bus
- Intel Pentium® processors are available with system bus speeds of 400, 533, 800, 1066 MHz and more
- The Intel X38 Express chipset operates at 1333 MHz and can transfer data at up to 21.2 GB/s

PC advertisement

Extreme Processing Power
Choose from the latest Intel® processors, which use the new **Intel X38 Express chipset** – state-of-the-art processing designed to drive your system and digital entertainment experience further and faster than ever before.

Bus Standards

- To allow different equipment to connect together, devices need to follow standards
- Bus designs are often developed by individual companies and then standardized by industry organizations

ISA Bus

- The Industry Standard Architecture (ISA) bus was used in the first 8088 PCs
- It was originally a 8 bit data bus with 20 dedicated address lines
- Bus design similar to 8088 local bus
- The bus was updated to have 16 data lines and operate at 8.33 MHz providing 8 MB/second bandwidth
- Updated again to the Extended ISA (EISA)

PCI Bus

- The Peripheral Component Interconnection (PCI) bus was developed by Intel in the early 1990s
- PCI has 64 data lines running at 66 MHz providing up to 528 MB/sec bandwidth
- Data and address lines are multiplexed
- Centralized arbitration

PCI Express

- Now on version 3.0, **PCIe** is a high speed replacement for PCI that can transfer up to 16 GB/s
- Structured around point-to-point serial links instead of a shared bus
- Allows more than one pair of devices to communicate with each other at the same time

SCSI Bus

- Pronounced “scuzzy”
- Small Computer System Interface
  - Supports both internal and external connection
- Comes in multiple bus widths
- Allows the connection of up to 16 devices. Each device has a bus ID
- Popular for hard drive

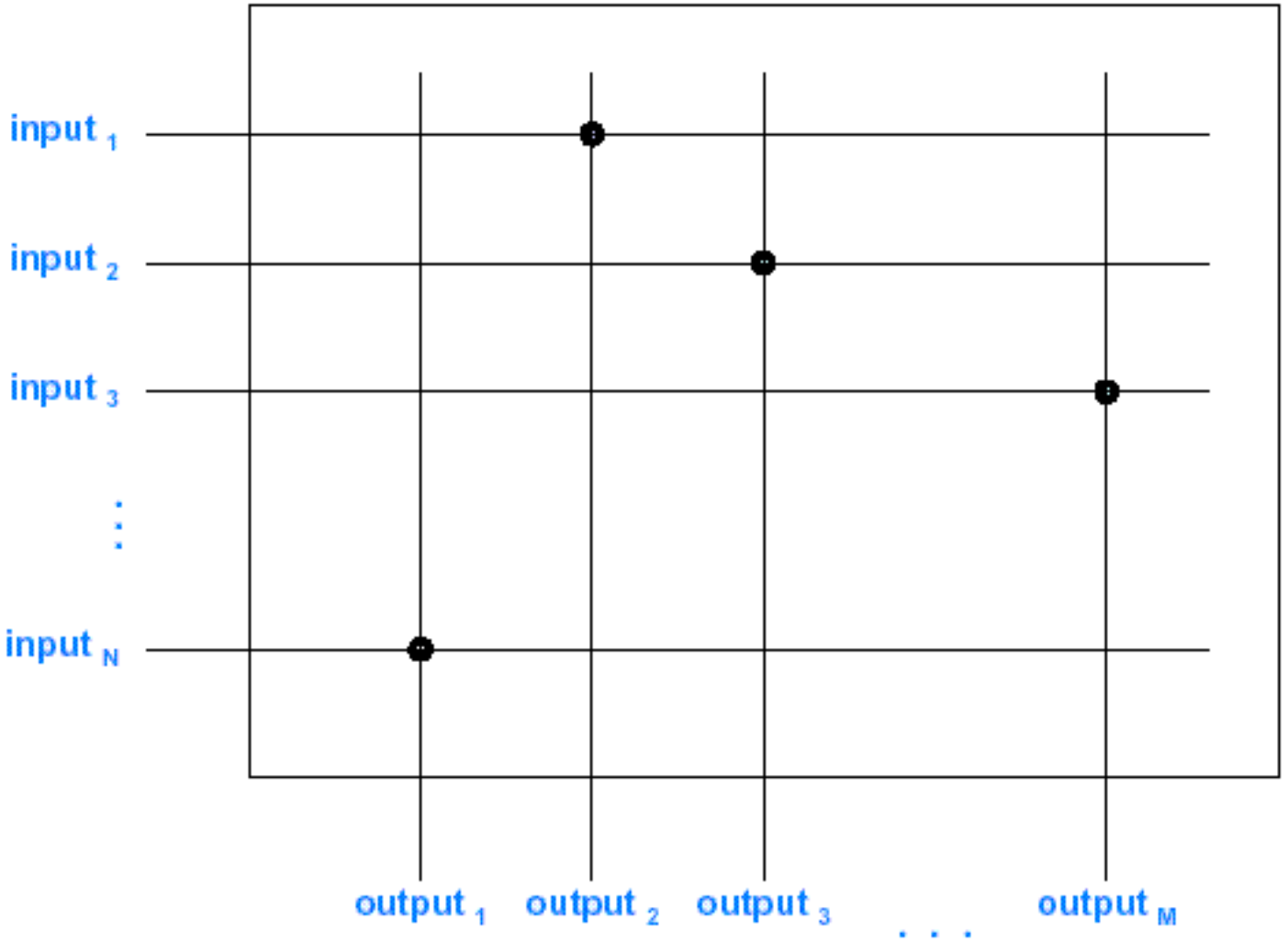
Fiber Optic SCSI Buses

- The original SCSI bus systems involved multiple parallel wires
- Some newer SCSI standards run over fiber optic cables
- Fiber optics uses only one "wire"
- Data is transmitted serially

Beyond Buses

- Although there are several devices that communicate over the bus, only one device can send data at a time
- Other interconnection schemes allow multiple simultaneous connections
- There are many designs of *switching fabrics* to interconnect devices
- Most switching fabrics can be expensive to implement

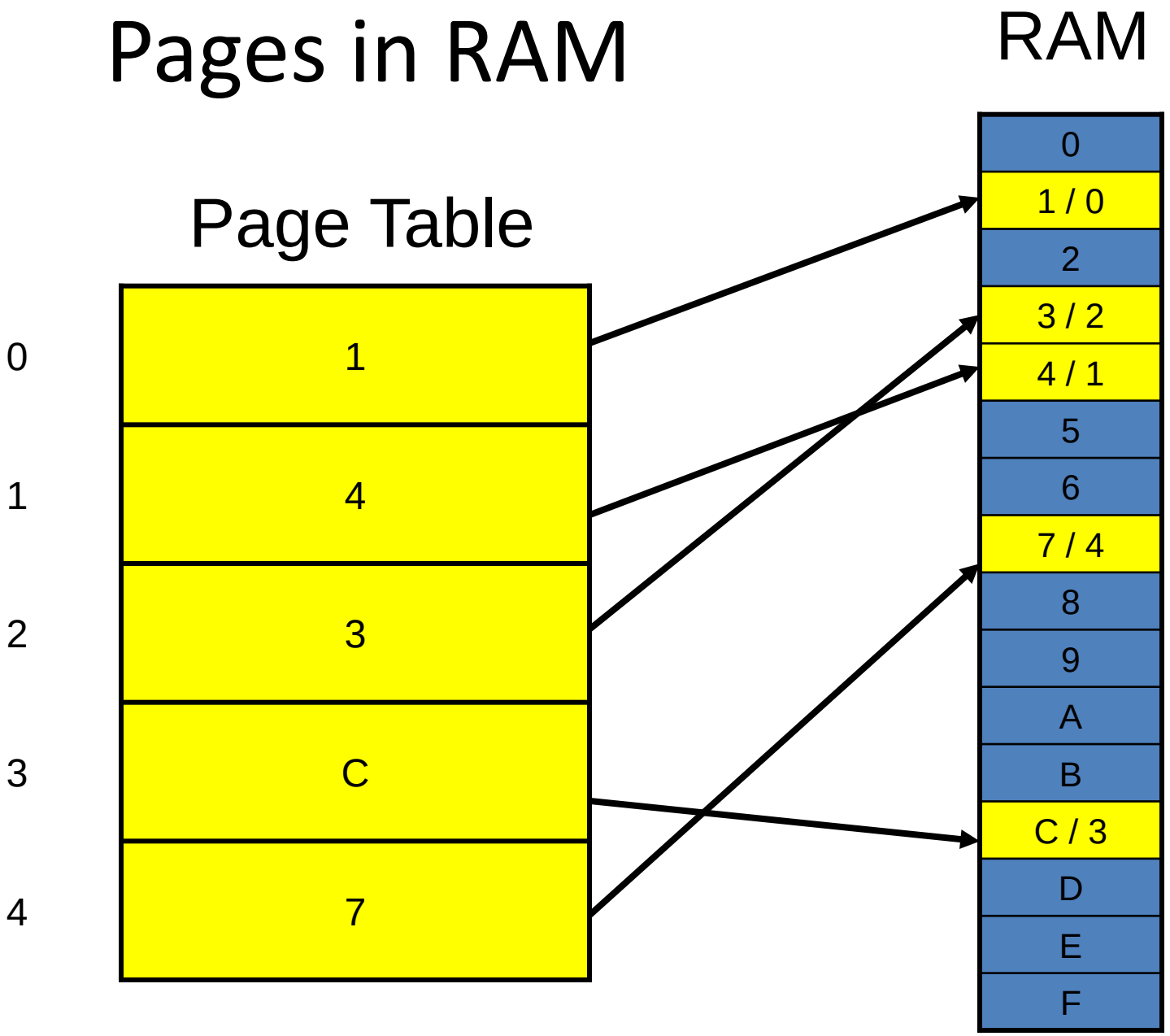
Crossbar Switch



Paged Memory

- RAM and programs are divided into fixed sized pages
- The page size is usually fixed for a given architecture, often between 512 -8K bytes
- The pages of a program can be put anywhere in RAM. They do not have to be contiguous.
- The page table keeps track of the physical location of pages
- The page table is indexed by the page number portion of a program address

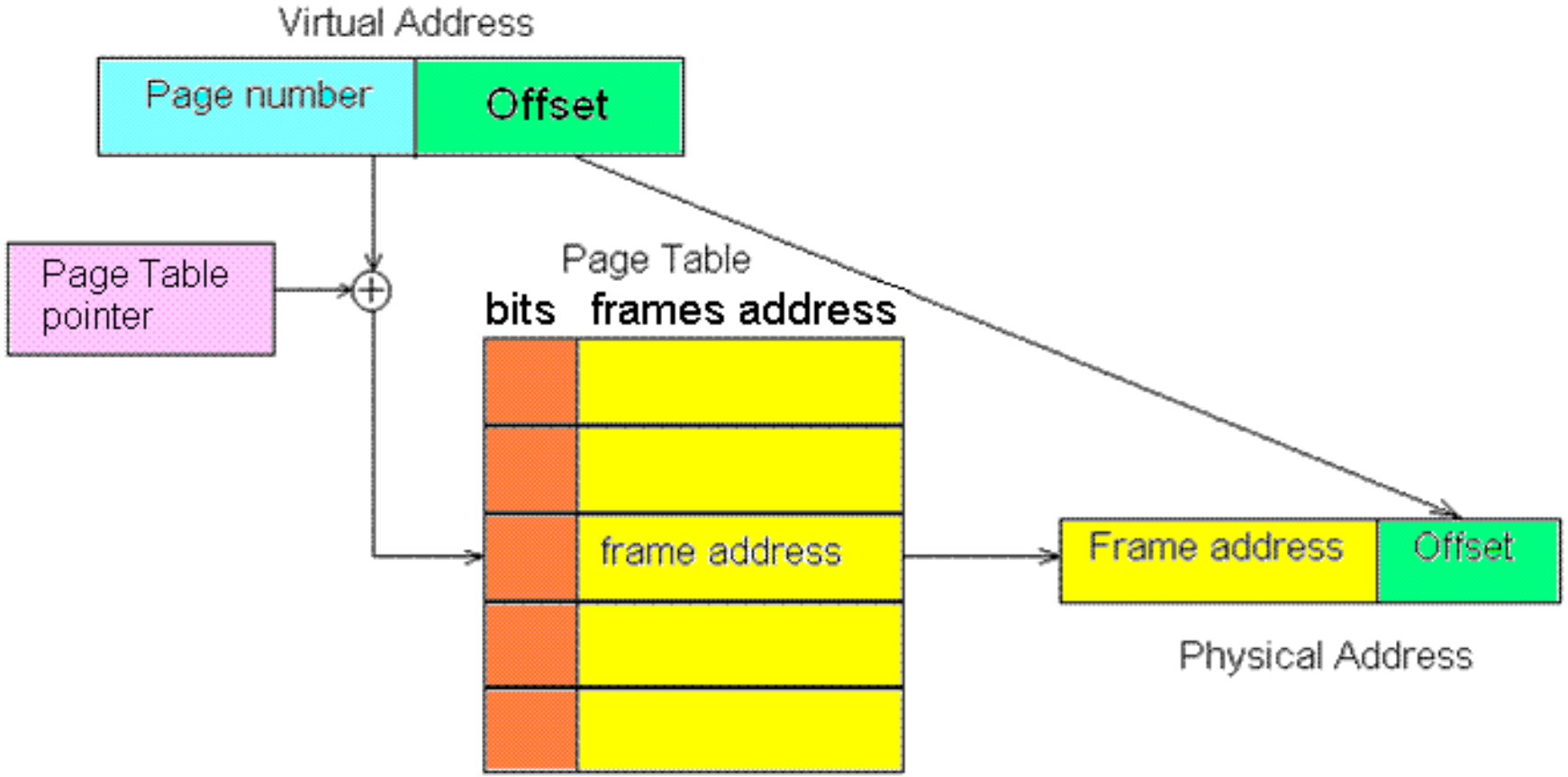
Pages in RAM



Virtual to Physical Address

- The upper bits of a program address are used as an index into a page table
- Each page table entry contains the physical address for that page of the program
- The lower bits of the program address (which indicate which byte in the page is desired) are concatenated to the end of the physical page address

Address Translation



Dividing the Address

- The offset portion of the address is always the lower $\log_2$(size of a page) bits of the address
- The page number portion are all of the address bits that are not part of the offset

How many bits are used for the page number and offset?

Assume **2 GB** address space and 4K byte pages.

A. page# = 21 offset = 11
B. page# = 22 offset = 10
C. page# = 19 offset = 11
D. page# = 20 offset = 12

Big Programs

- Even when using every byte of RAM, it is not always possible to load all the programs users would like
- Frequently large parts of programs are never executed. There are many features of Microsoft Word® you have never used.
- More programs could fit in memory if only the used portions were loaded into RAM

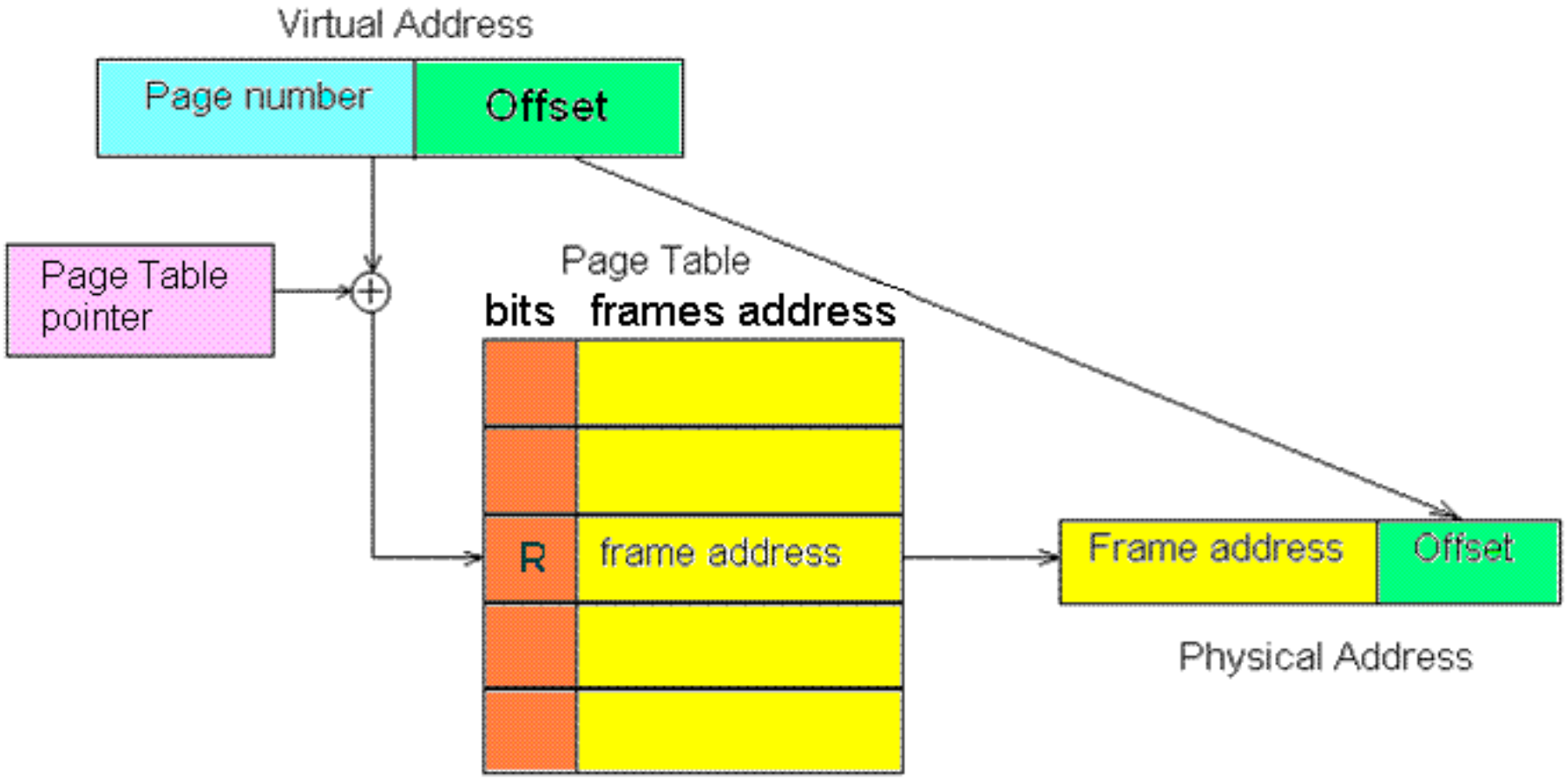
Virtual Memory

- Virtual Memory is an extension of paging
- Only the pages that are being used are in RAM
- A copy of all pages of a program are on the page file
- If a program accesses an address in a page not in RAM, the hardware creates a page fault interrupt and the OS copies the desired page into RAM

Virtual Memory Implementation

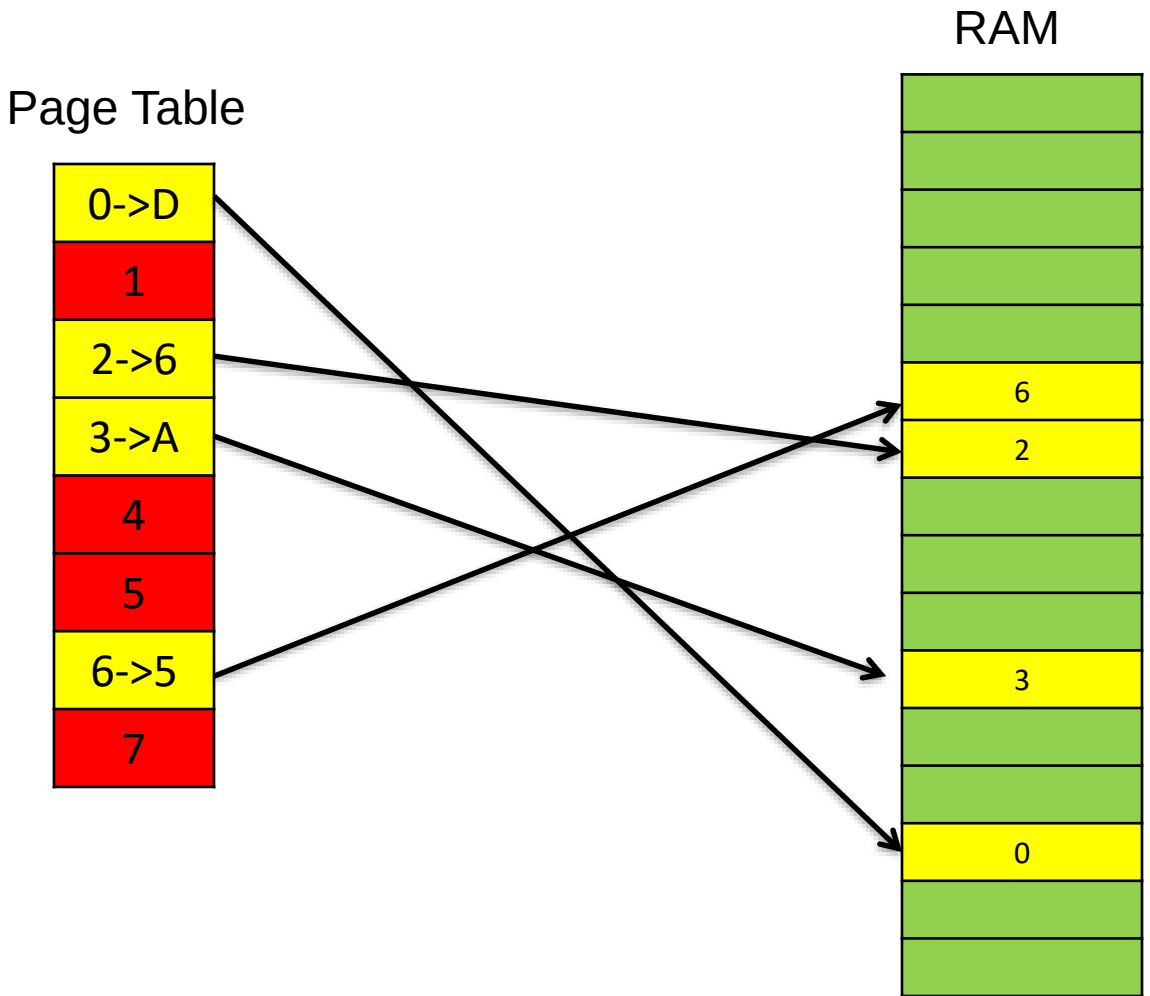
- Unused pages of a program do not need to be in RAM to execute the program
- A “resident” bit is added to the page table
  - Pages in RAM have the resident bit set
  - Pages not in RAM have the resident bit cleared
- All pages are stored on disk
- When a program references a page with the resident bit clear, the hardware creates a page fault interrupt

Address Translation



Only Necessary Pages in RAM

Program addresses showing pages that are referenced



Hardware and Operating System

- Virtual memory requires both hardware and operating system support
- When a page fault interrupt occurs, the OS reads the desired page from disk into an available page of RAM
- The user’s page table is updated to point to the newly loaded page
- The program is placed back on the ready list to be executed

User Programs and Virtual Memory

- Virtual memory is transparent to user programs
- Programs are unaware of page faults
- The only impact of virtual memory on user programs is performance
- If a program creates frequent page faults, the program will run very slowly

Page Table Flags

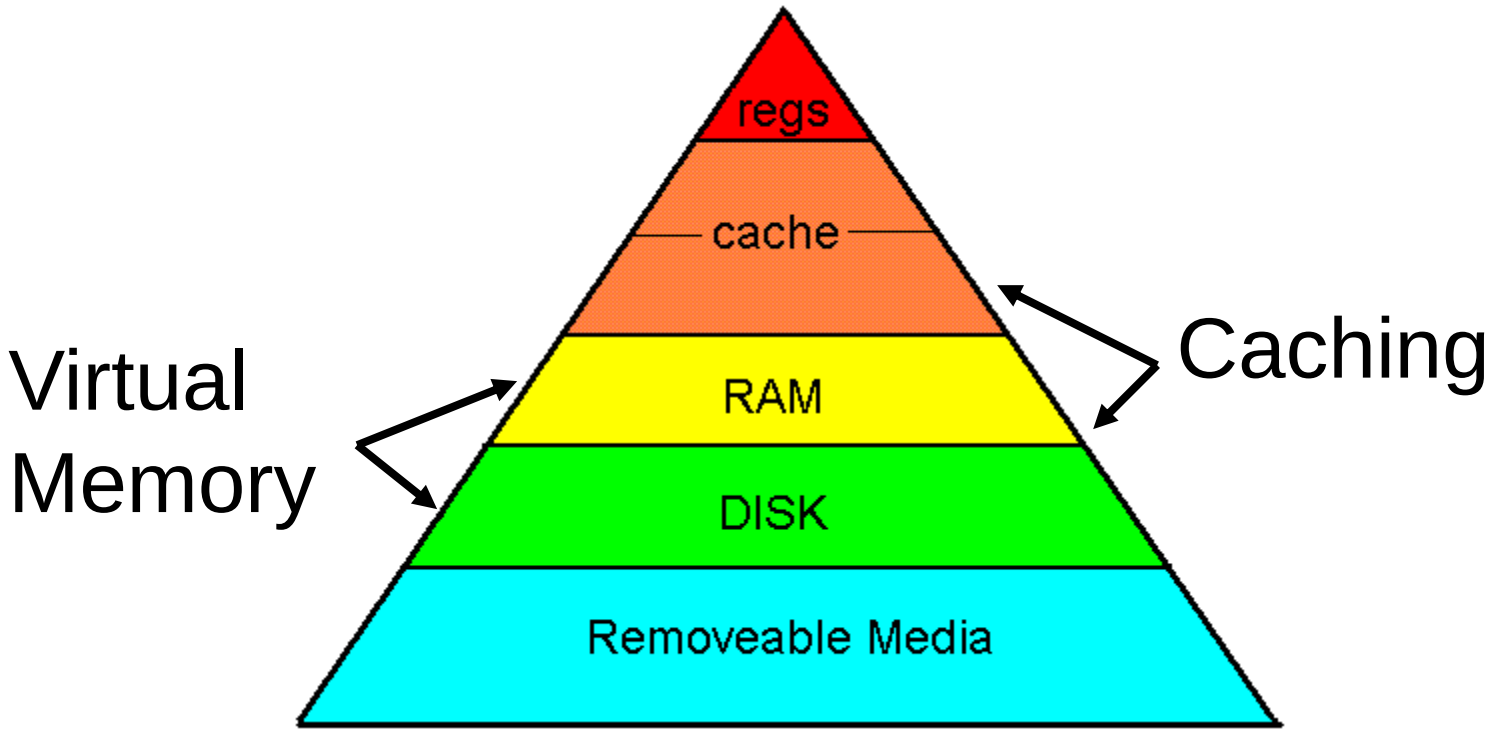
- Each page table entry has flag bits
  - **Resident** – set if the virtual address is in RAM and clear otherwise.
  - **Used** – set when the page is referenced
  - **Dirty** – set when the page is changed
  - **No Execute** – Instructions cannot be fetched from this page
- The Used and Dirty bits are set by the hardware and cleared by the OS

Locality of Reference

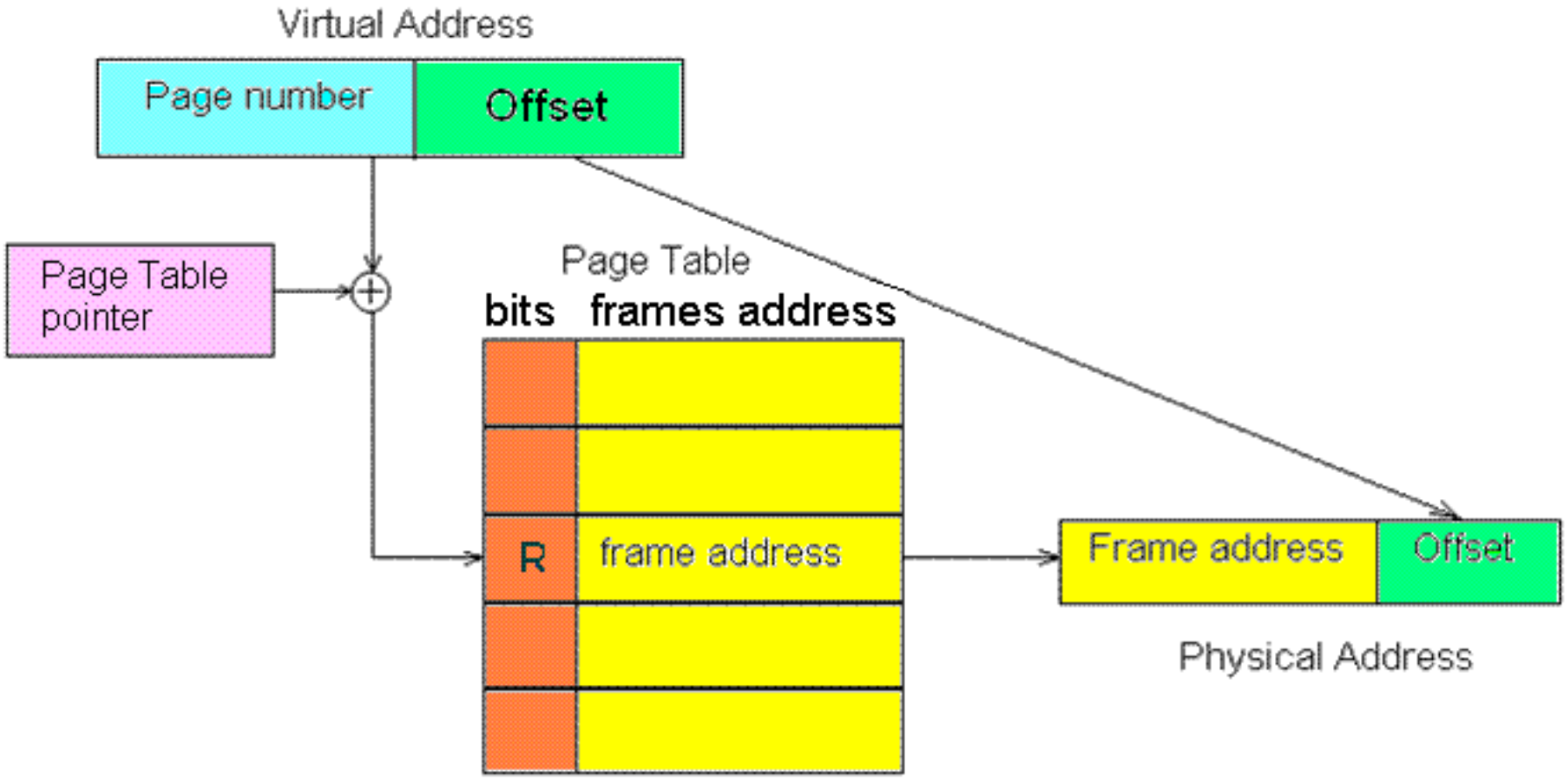
- **Temporal locality** - a referenced location is likely to be referenced again

- **Spatial locality** - nearby locations are likely to be referenced soon

Memory Hierarchy



Address Translation



Translation Lookaside Buffer

- The Translation Lookaside Buffer (TLB) is a special cache to hold recently used page table entries
- The CPU looks in the TLB for a page entry before looking in the page table
- Without the TLB each memory reference would require two memory accesses
- Most TLBs are small and use fully associative mapping

Performance Factors

- The hardware handles the normal address translation
- When a page fault occurs, the OS determines where the page will be loaded and what page will be overwritten
- The OS must minimize the number of page faults
- The OS determines which and how many pages of a program are in RAM

Virtual Memory Performance

- Computers can retrieve a word from RAM in about 60 ns (6×10^{-8} sec)
- A good disk drive can read a block of data in about 6 ms (6×10^{-3} sec)
- If the needed page is not in RAM and has to be read from the disk, it takes about 100,000 times longer to get the data.
- Page fault interrupts have to be kept to a minimum.

A High Hit Rate Is Important

- Consider RAM in access in 60 ns and disk drive access in 6 ms
- If the hit ratio is 99.99%

$$\text{avg access} = 0.9999 * 6 \times 10^{-8} \text{ sec} + 0.0001 * 6 \times 10^{-3} \text{ sec} = 660 \text{ nsec}$$

- The average memory access time is about 11 times slower

What is the average access time for 99.9% hit rate with 9.0 ms disk and 9.0 ns memory?

A. 8.1 ms
B. 8.1 ns
C. 9.0 ms
D. 9.0 µs
E. 9.0 ns

Program to Physical Translation Goals

- Convert program addresses to physical addresses
  - More complicated than base address. May require two memory accesses.
- Make it easy for the operating system to place the program in memory
  - Much easier than base address
- Allow sharing of data and instructions
  - Pages can be shared between programs

Program to Physical Translation Goals

- Detect out of range memory accesses
  - If the page table index is too large, a segmentation fault interrupt with occur
- Provide read-only and no-execute memory segments
  - Different pages can be marked read-only or no-execute

Virtual Memory Advantages

- Allows you to fit many large programs into a relatively small RAM
- Only part of a program needs to be loaded into memory
- Eliminates the need to “fit” programs into memory holes

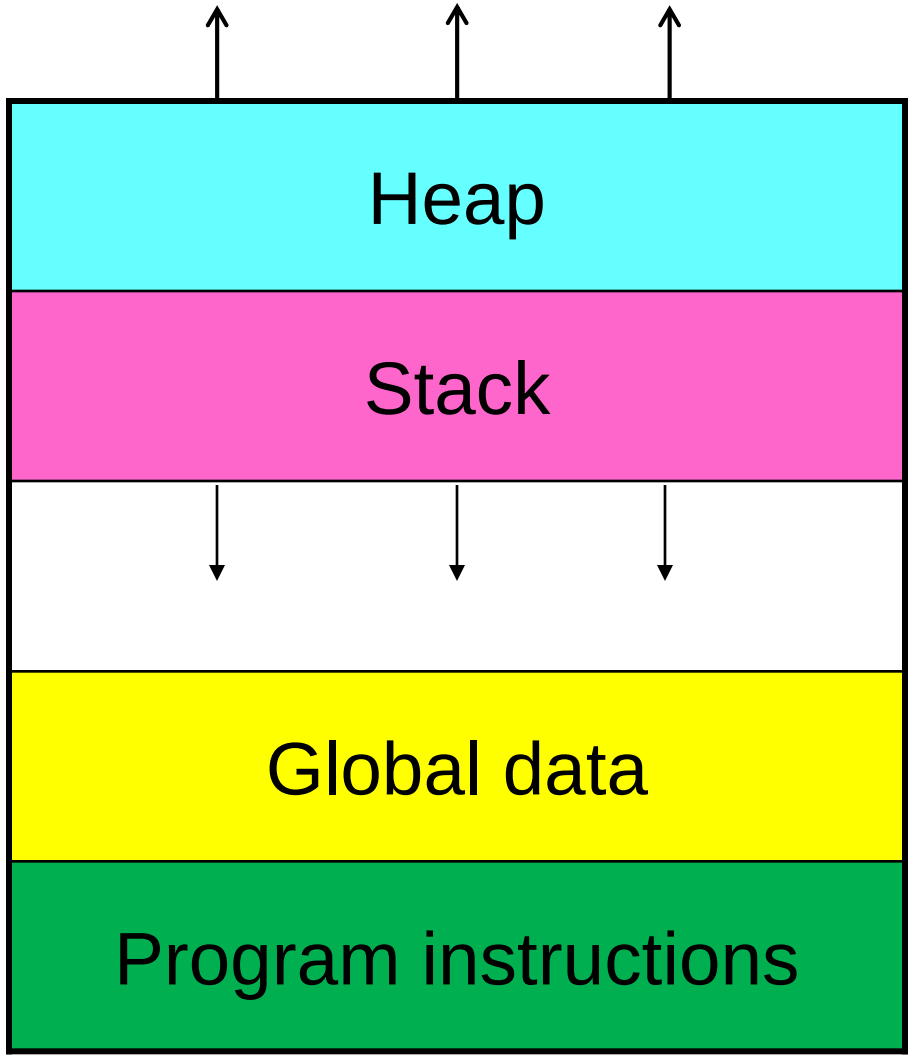
Virtual Memory Disadvantages

A. Makes address translation much more complicated
B. Can reduce performance
C. Makes program execution time less predictable
D. All of the above
E. None of the above

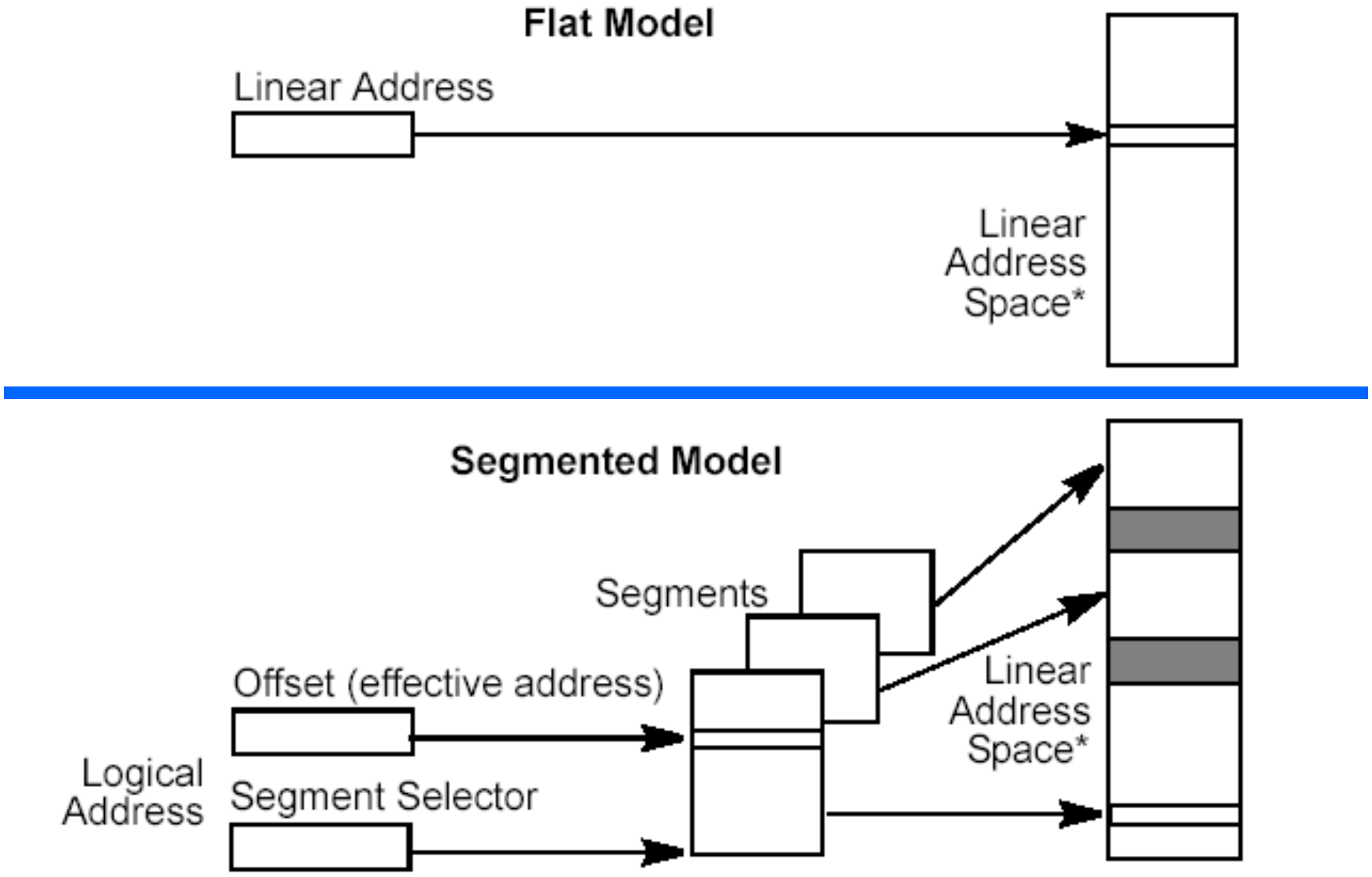
Segmented Memory

- Memory can be separated into segments based on the program
  - instruction segments for each function
  - data segments for global data and heap
  - stack segments
- Segmentation provides greater function isolation and makes linking easier
- Segments can have virtual memory pages

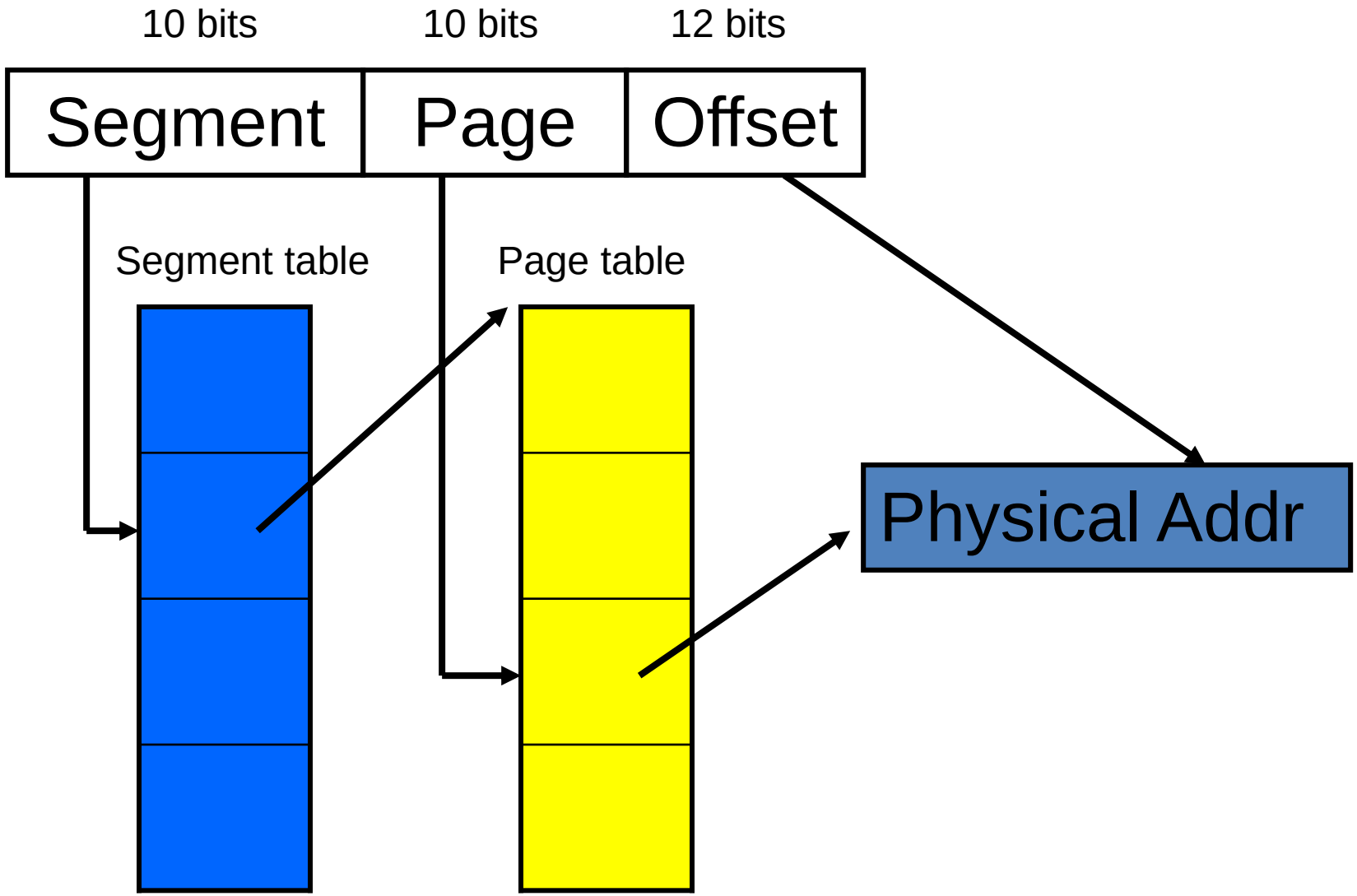
Program Memory Organization



Memory Models



Intel Segment Addresses



Practice Coding Interviews with Google

- Join the Google in Residence Interview Coaching program
- Google is hosting two on-campus interview prep and mock interview group sessions where you'll practice with a Googler and learn how to think through problems.
- Interview Prep Session: (6:00-9:00 PM in 207 Cherry Hall) November 11
- Group Mock Interview Session November 12
- RSVP @ goo.gle/2IX1E2D