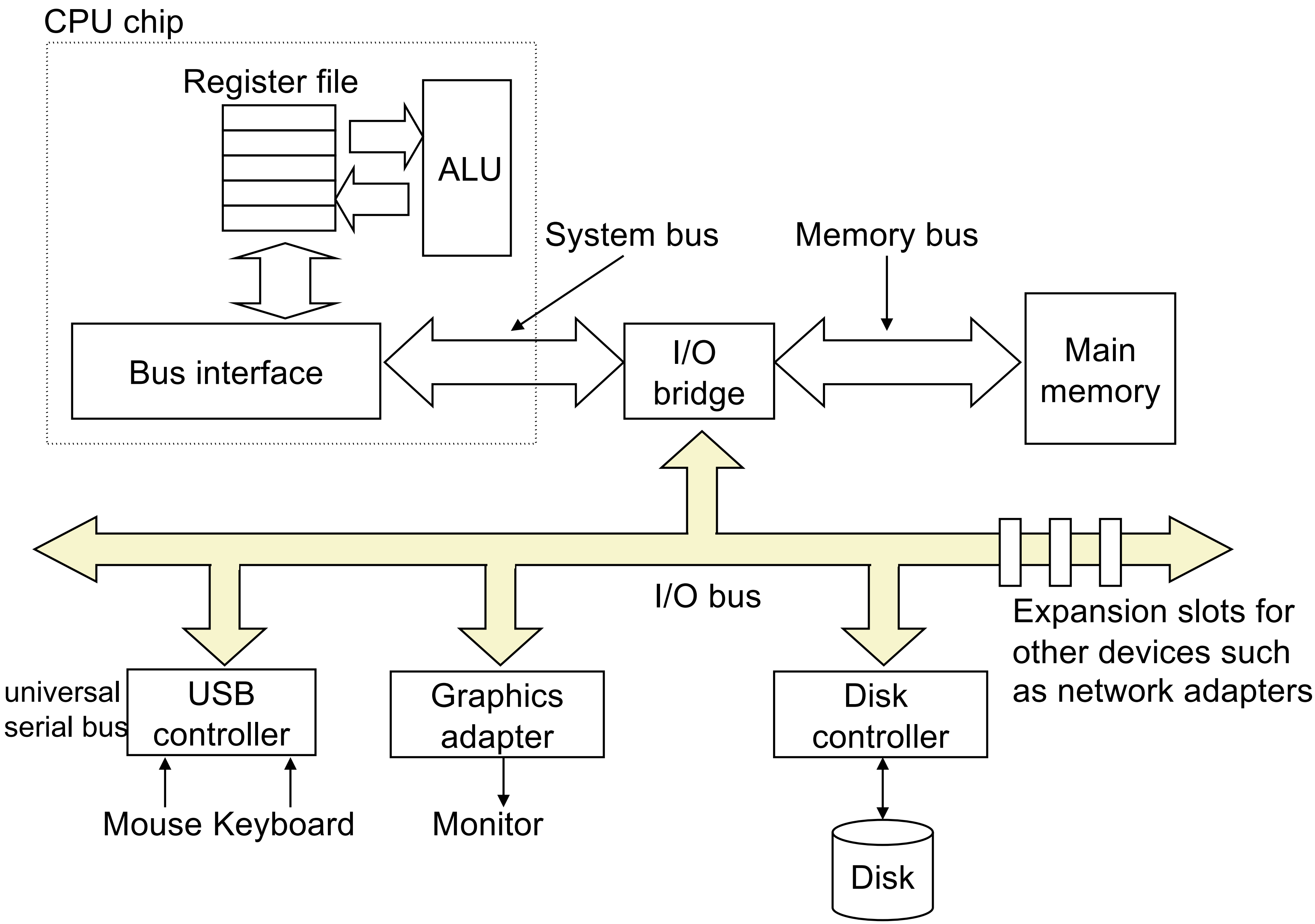
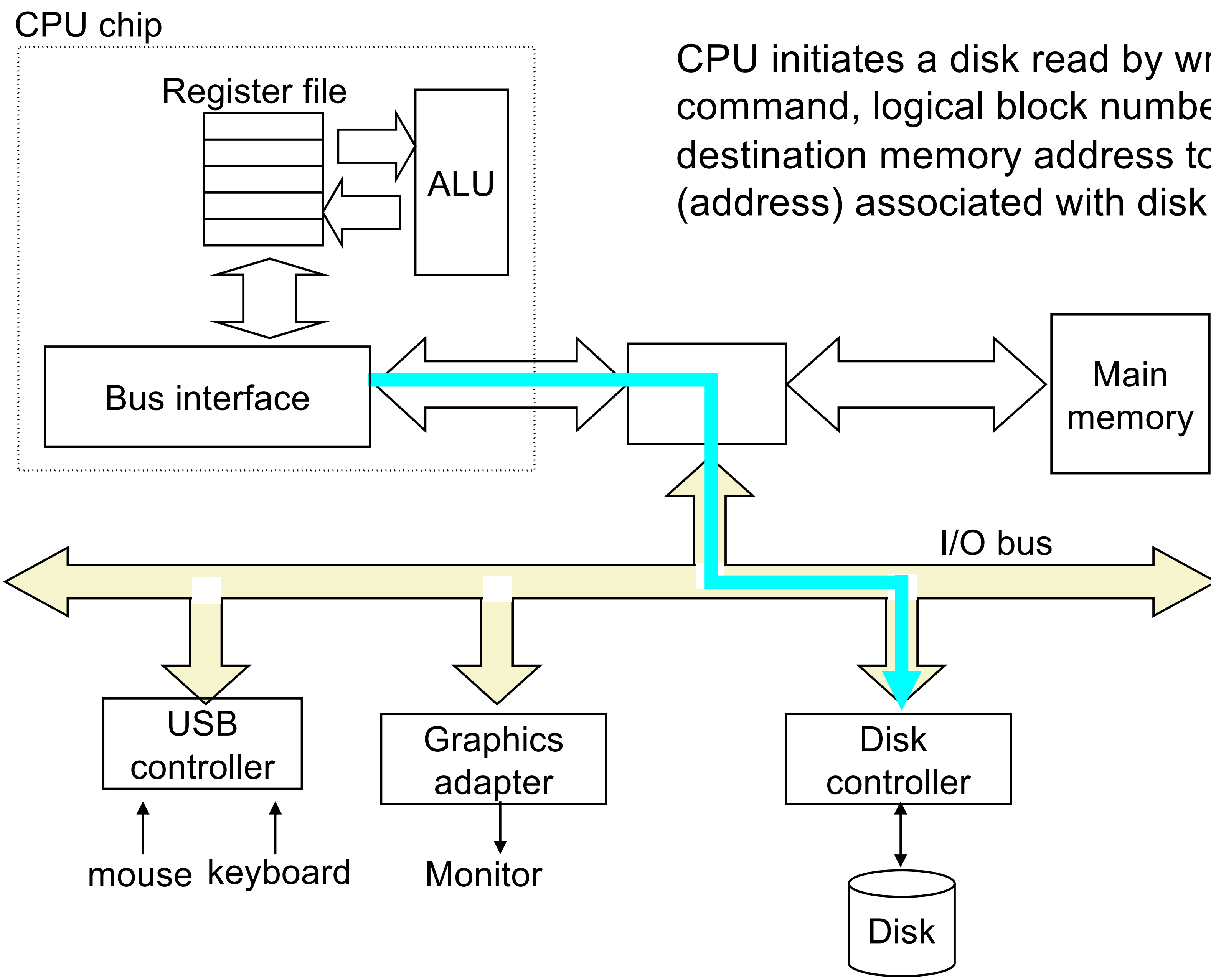
I/O Bus

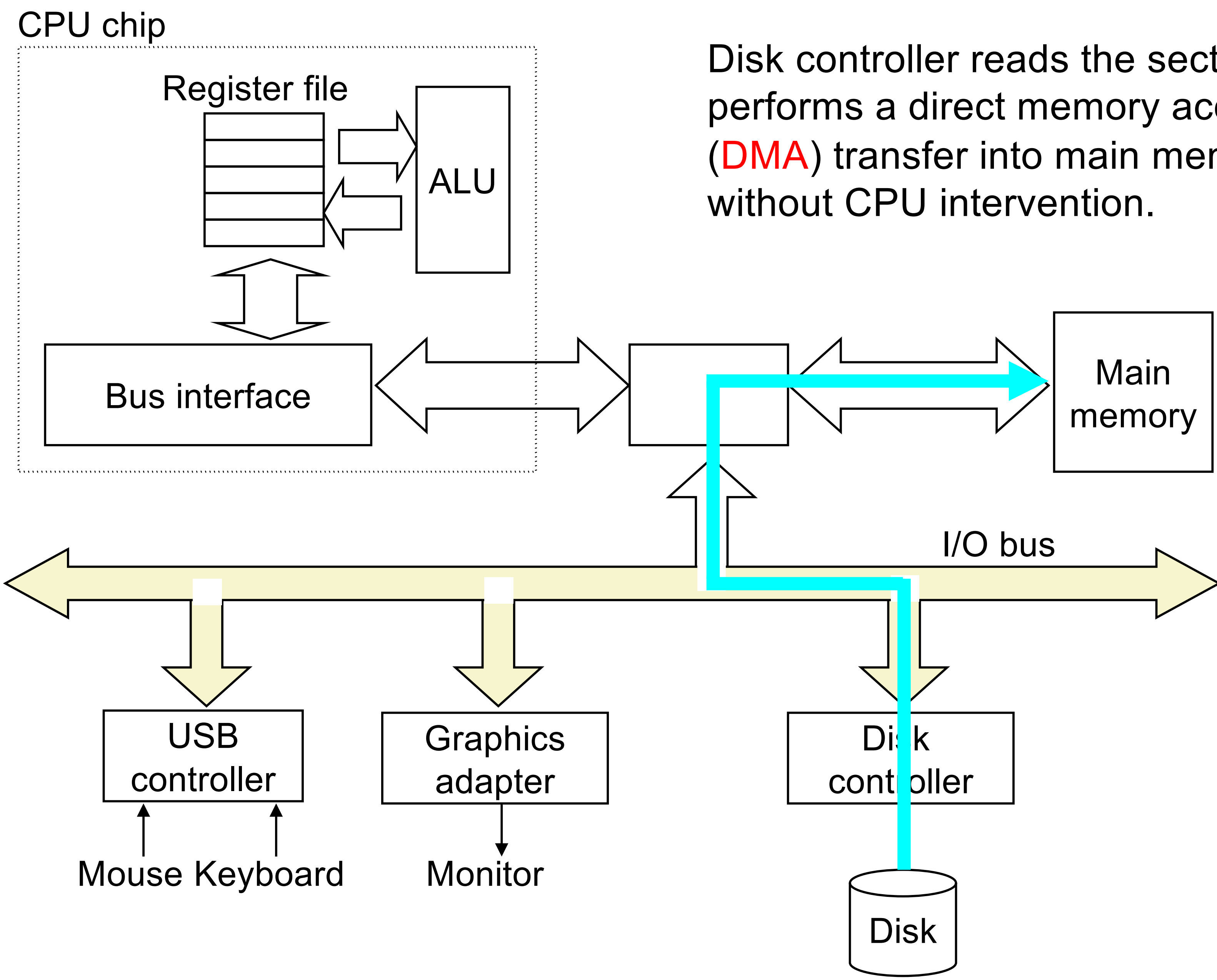

CPU chip
Register file
ALU
System bus
Memory bus
Bus interface
I/O
bridge
Main
memory
I/O bus
Expansion slots for
other devices such
as network adapters.
universal
serial bus
USB
controller
Graphics
adapter
Disk
controller
Mouse Keyboard
Monitor
Disk

Reading a Disk Sector (1) (fig. 6.12)



CPU initiates a disk read by writing a command, logical block number, and destination memory address to a port (address) associated with disk controller.

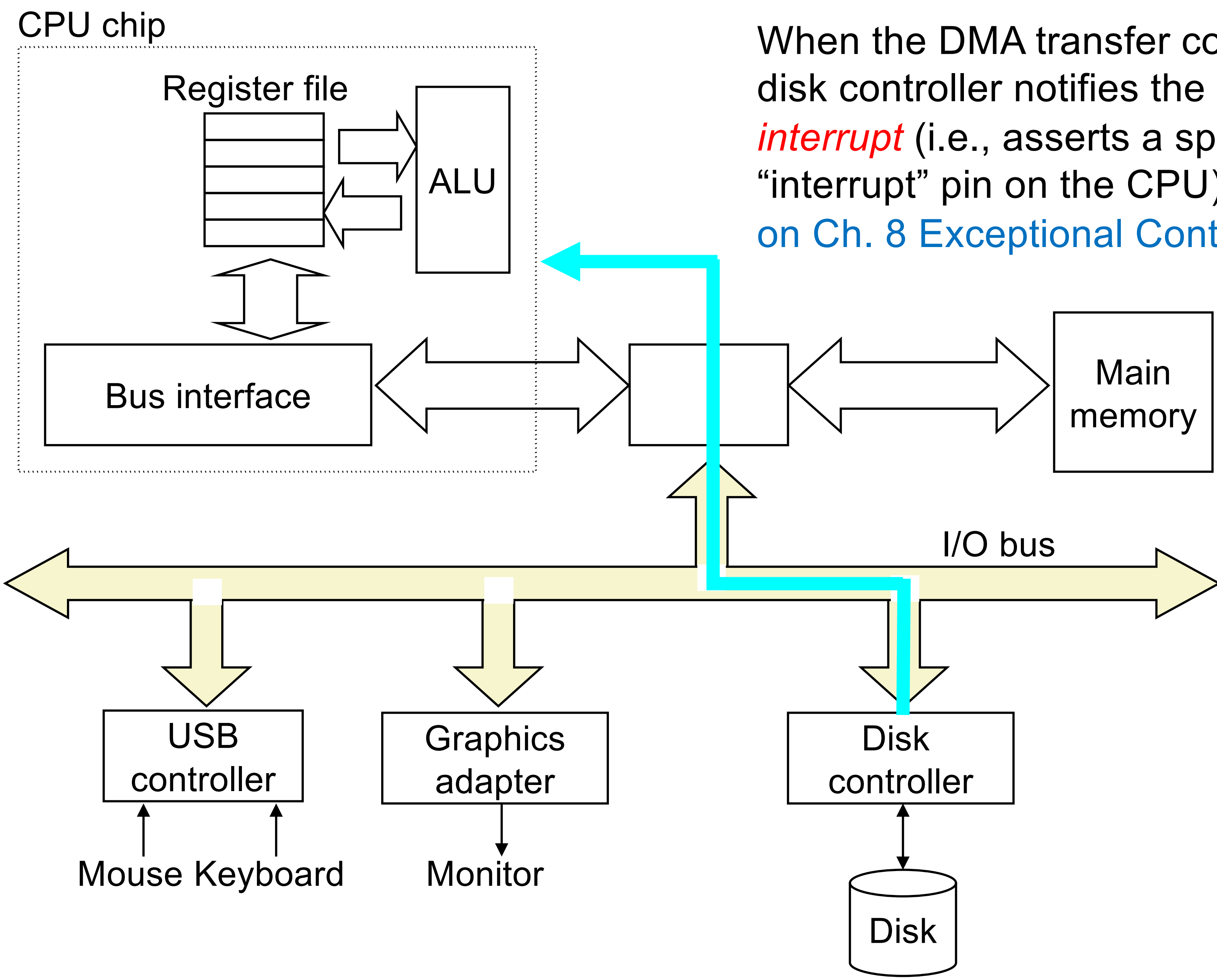
Reading a Disk Sector (2)



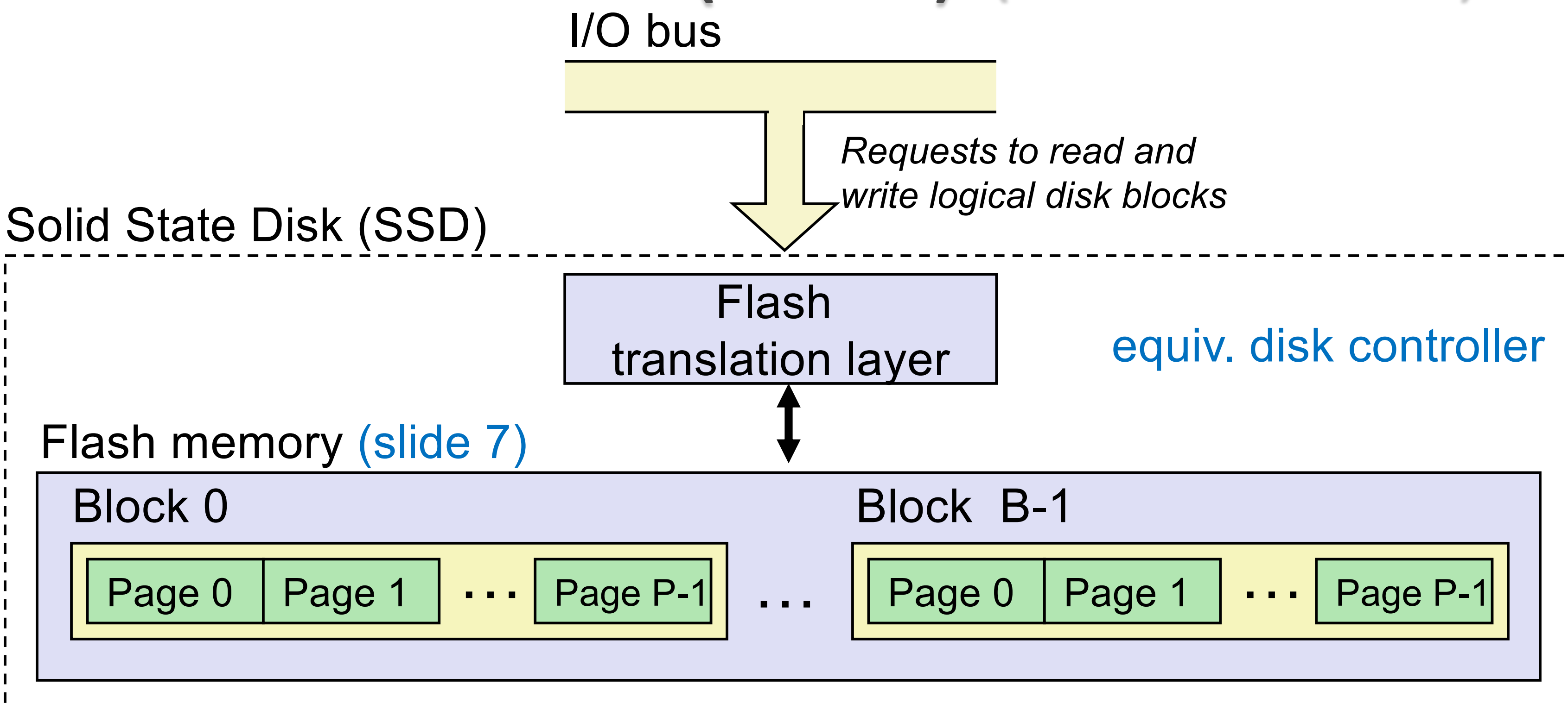
Disk controller reads the sector and performs a direct memory access (DMA) transfer into main memory, without CPU intervention.

Reading a Disk Sector (3)

When the DMA transfer completes, the disk controller notifies the CPU with an *interrupt* (i.e., asserts a special "interrupt" pin on the CPU) (More on this on Ch. 8 Exceptional Control Flow)



Solid State Disks (SSDs) (section 6.1.3)



- Pages: 512KB to 4KB, Blocks: 32 to 128 pages
- Data read/written in units of pages.
- Page can be written only after its block has been erased
- A block wears out after about 100,000 repeated writes.

SSD Performance Characteristics

Sequential read thruput	550 MB/s	Sequential write thruput	470 MB/s
Random read thruput	365 MB/s	Random write thruput	303 MB/s
Avg seq read time	50 µs	Avg seq write time	60 µs

- Sequential access faster than random access
  - Common theme in the memory hierarchy
- Random writes are somewhat slower
  - Erasing a block takes a long time (~1 ms).
  - Modifying a block page requires all other pages to be copied to new block.
  - In earlier SSDs, the read/write gap was much larger.

Source: Intel SSD 730 product specification.

SSD vs rotating disks (2)

- Even though SSD are faster than rotating disks, compared with the main memory and caches SSD are still at least 3 orders of magnitude slower.
- There is also overhead on translating the logical block number to the physical address which includes physical block and page.
- ms miliseconds — 1 ms = 1,000,000 ns
- µs microseconds — 1 µs = 1,000 ns
- ns nanoseconds — 1 µs = 1,000 ns

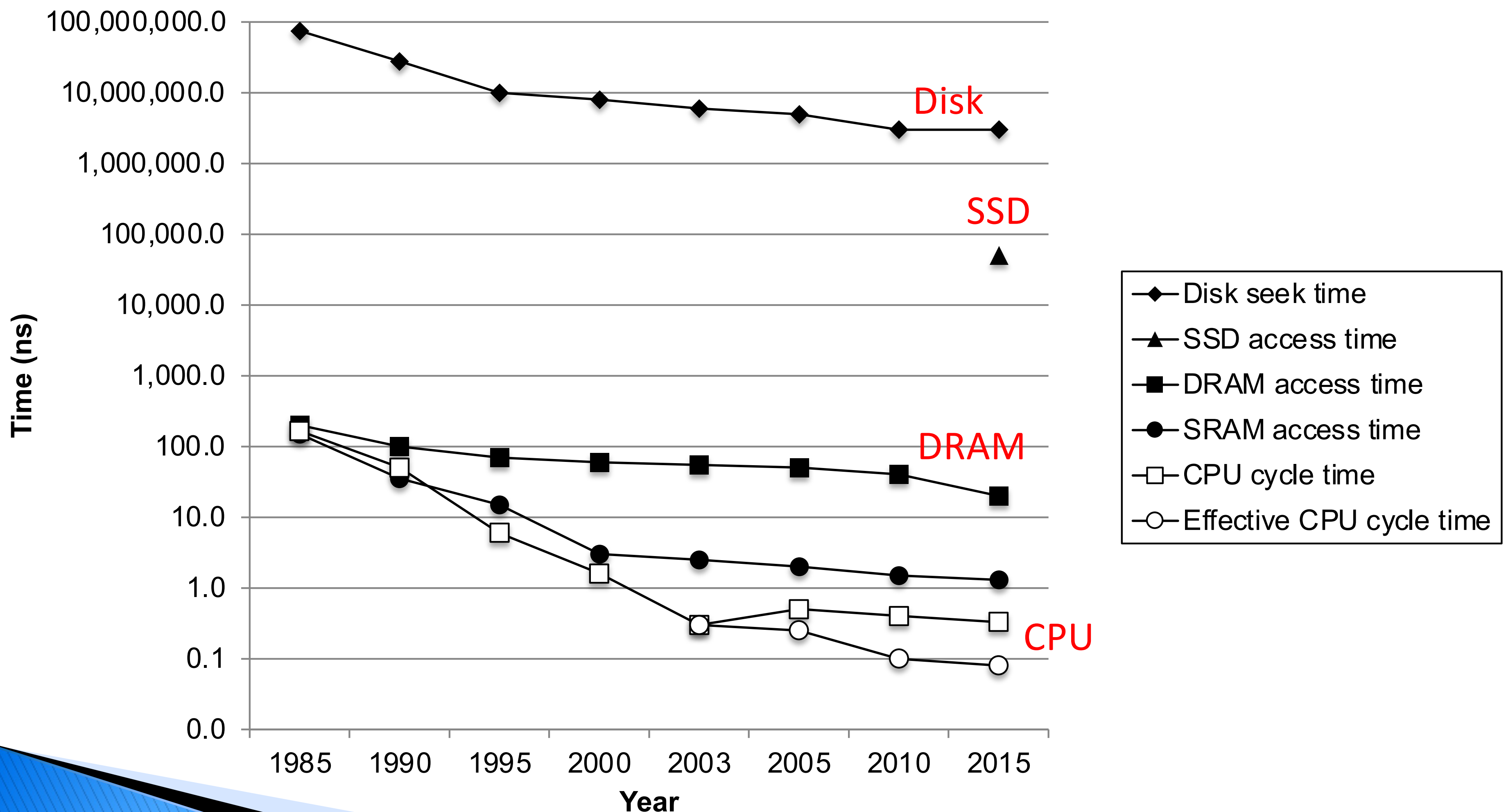
1 ms = 1,000 µs

SSD Tradeoffs vs Rotating Disks

- Advantages
  - No moving parts → faster, less power, more rugged
- Disadvantages
  - Have the potential to wear out
    - Mitigated by "wear leveling logic" in flash translation layer
    - E.g. Intel SSD 730 guarantees 128 petabyte (128×10^{15} bytes) of writes before they wear out
  - In 2015, about 30 times more expensive per byte
- Applications
  - MP3 players, smart phones, laptops
  - Beginning to appear in desktops and servers

The CPU-Memory Gap (trend details sec. 6.1.4, p. 602)

The gap widens between disk, DRAM, and CPU speeds.



Advances

- On Canvas - week 10, article from 2019 with recent developments on disk storage technologies:
  - SSD increased capacity but lowered speed
  - 3D Xpoint - “new type of non-volatile memory that sits somewhere between the two.” (SSD and DRAM)

Locality to the Rescue!

The key to bridging this CPU-Memory gap is a fundamental property of computer programs known as locality

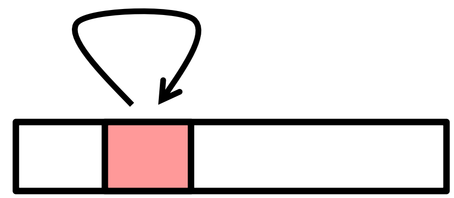
As we just saw in problem 6.4

Comparing the total time to transfer a 1MB file stored on a rotating disk

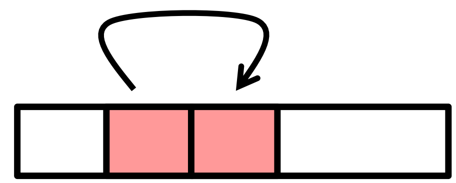
- A. When all the blocks of the file are in contiguous locations
  - 20 msec
- B. When the blocks of the file are randomly placed in the disk
  - 16,000 msec = 16 seconds!

Locality

- Principle of Locality: Programs tend to use data and instructions with addresses near or equal to those they have used recently



- Temporal locality:
  - Recently referenced items are likely to be referenced again in the near future



- Spatial locality:
  - Items with nearby addresses tend to be referenced close together in time

Locality Example

```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

- Data references
  - Reference array elements in succession (stride-1 reference pattern). — Spatial locality
  - Reference variable `sum` each iteration. — Temporal locality
- Instruction references
  - Reference instructions in sequence. — Spatial locality
  - Cycle through loop repeatedly. — Temporal locality

Stride-k pattern definition

```
sum = 0;
for (i = 0; i < n; i+=k)
    sum += a[i];
return sum;
```

- Visiting the k-th element of a contiguous vector is called a stride-k reference pattern
- The smaller the k, the better the Spatial locality
- The program shown has a stride-k reference pattern.