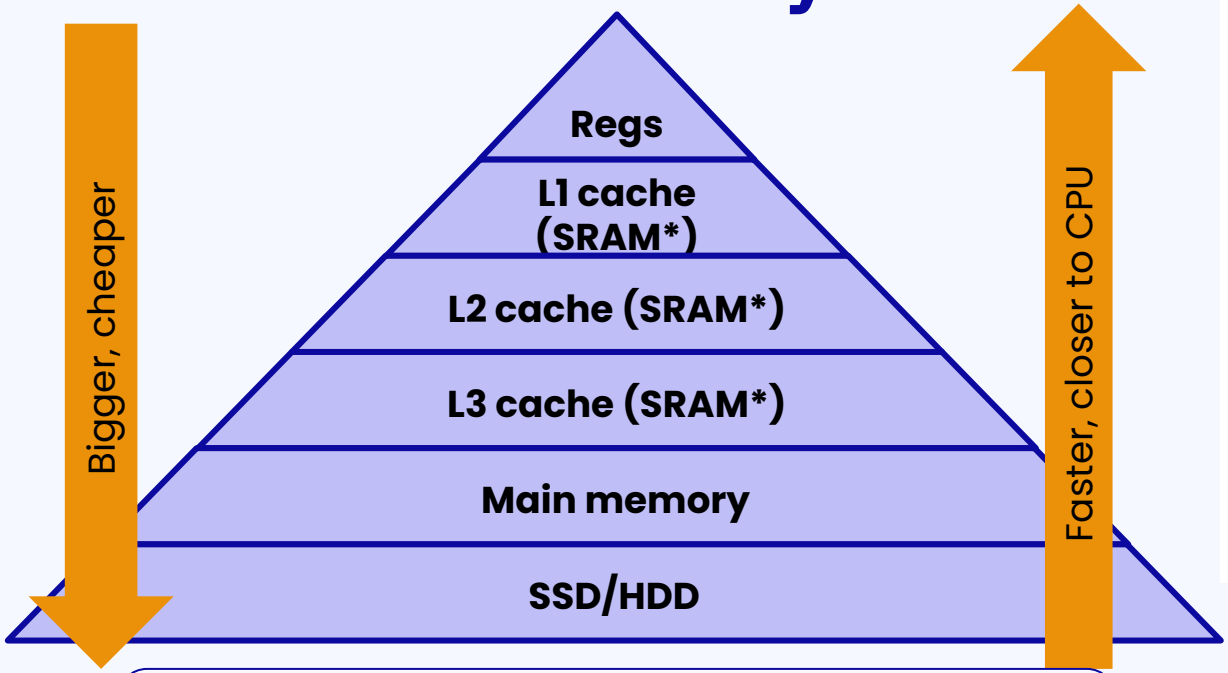
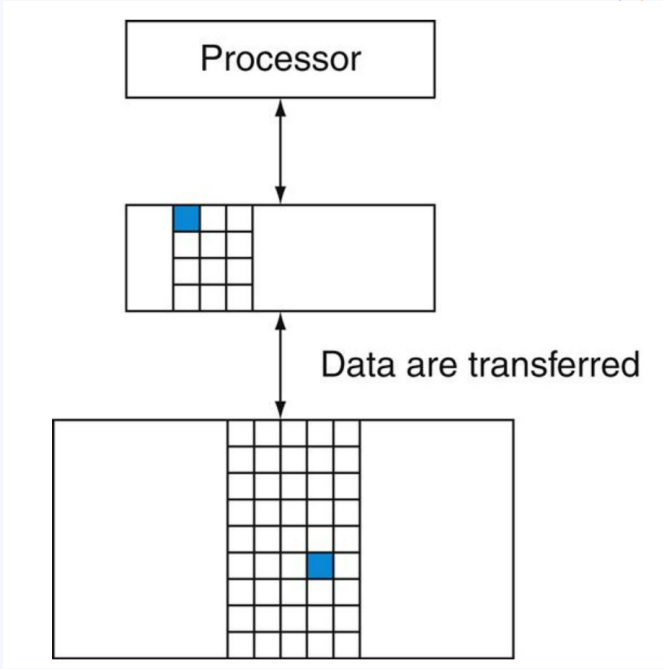
Cache controllers

Mem. hierarchy review



Each level stores (**caches**) a subset of the one below it, for faster access of specific data



P&H 5.2

What questions do we need to ask when designing a cache?

- How do we decide what goes where in a cache?
- What control information do we need to keep track of in order to implement our cache?
- How do we decide what data to evict?
- What is our scheme of maintaining consistent data?
- How do we build an efficient memory hierarchy? (What is “efficient”)

How do we decide what goes where?

For now: assume 1 word (4 byte) **block** size (reminder: minimum unit of information transferred between levels of memory hierarchy)

Assume one cache that holds 256 (2^8) blocks (1KB)

Assume register a0 holds the value 0x20000004

```
lb t0, 1(a0)

lw t1, 4(a0)
```

Block address – the address of the block of memory we care about
Why do we bring things in to cache before loading them into a register?
Where in the cache do these blocks go?

What control information do we need?

Assume register a0 holds the value 0x20000004

```
lb t0, 1(a0)

lw t1, 4(a0)

lhu t2, 2(a0)
```

How do we tell that the block with the data we need already exists in the cache?

How do we decide what data to evict?

Assume register a0 holds the value 0x20000004

```
lb t0, 1(a0)
lw t1, 4(a0)
lhu t2, 2(a0)
lw t0, 1024(a0)
```

Address maps to a cache block already in use:
how do we know that it's already in use?

Terminology

Block: minimum unit of information that can be present/not present in a cache

Valid bit: indicates whether data has been pulled in to that block of the cache

Tag: the upper bits of an address, used to uniquely identify which data is in the cache

What is our scheme of maintaining consistent data?

Assume register a0 holds the value 0x20000004

```
lb t0, 1(a0)
lw t1, 4(a0)
lhu t2, 2(a0)
lw t0, 1024(a0)
addi t1, t1, 1
sw t1, 4(a0)
sh t1, 8(a0)
```

What happens when data that exists in the cache is modified?
What happens when data that does not yet exist in the cache is modified?

Write through vs write back

Write through: every time data is changed in cache, change is done to lower level in hierarchy

Pro:

Con:

Write back: changes to lower level in hierarchy are only done when data is evicted from cache

Pro:

Con: