M6 – Memory Hierarchy

Module Outline

- CPU – Memory interaction
- Organization of memory modules
- Cache memory – Mapping and replacement policies.

Events on a Cache Miss

Events on a Cache Miss

- Stall the pipeline.

Events on a Cache Miss

- Stall the pipeline.
- Steps on an I-Cache miss:

Events on a Cache Miss

- Stall the pipeline.
- Steps on an I-Cache miss:
  1. Send the PC value to the memory.

Events on a Cache Miss

- Stall the pipeline.
- Steps on an I-Cache miss:
  1. Send the PC value to the memory.
  2. Instruct main memory to perform a read and wait for the memory to complete its access.

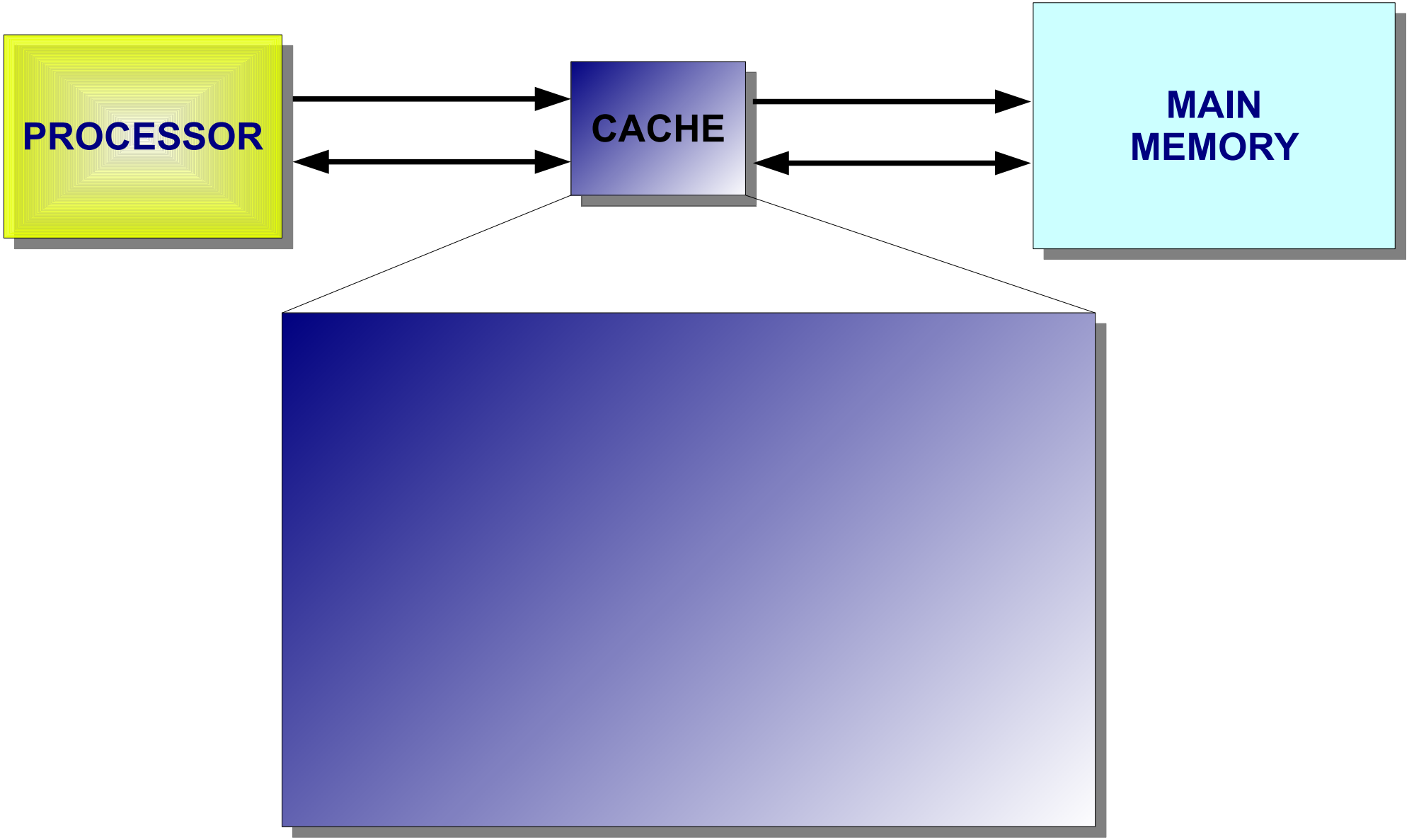
Events on a Cache Miss

- Stall the pipeline.
- Steps on an I-Cache miss:
  1. Send the PC value to the memory.
  2. Instruct main memory to perform a read and wait for the memory to complete its access.
  3. Fill the cache entry: write the data from memory into the cache block, fill the tag field from the address, turn the valid bit on.

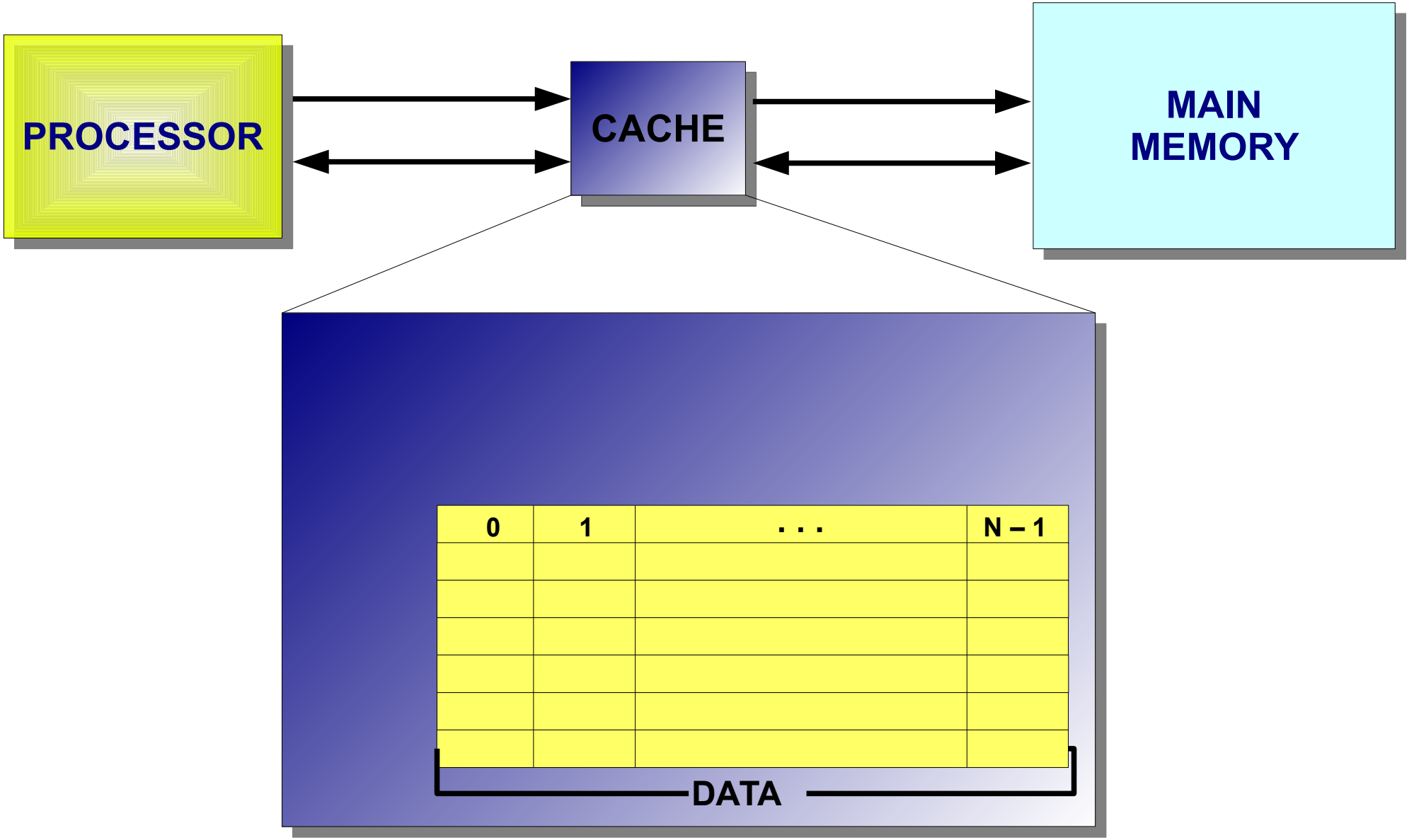
Events on a Cache Miss

- Stall the pipeline.
- Steps on an I-Cache miss:
  1. Send the PC value to the memory.
  2. Instruct main memory to perform a read and wait for the memory to complete its access.
  3. Fill the cache entry: write the data from memory into the cache block, fill the tag field from the address, turn the valid bit on.
  4. Restart the instruction execution at the first step, which will refetch the instruction, this time finding it in the cache.

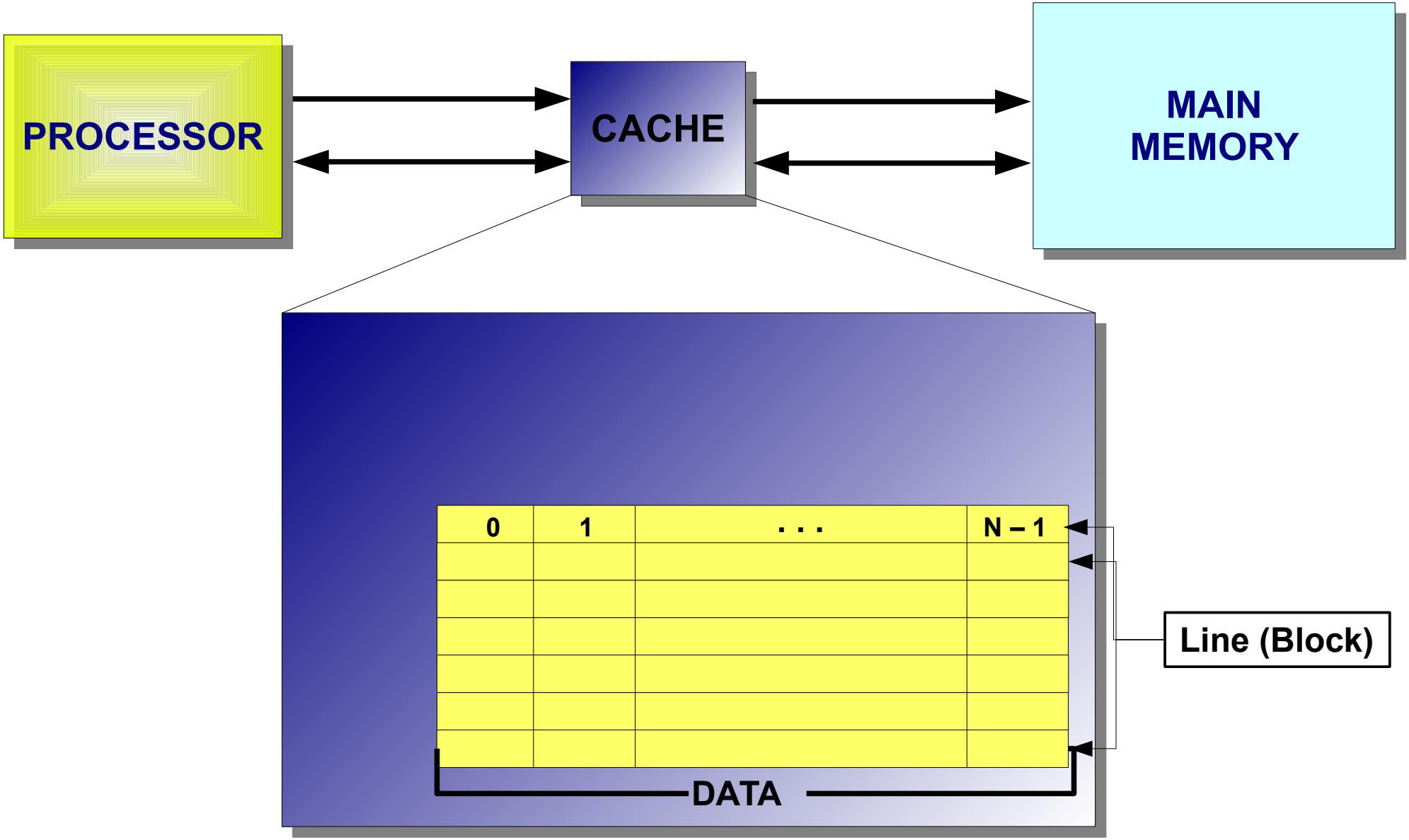
Inside a Cache



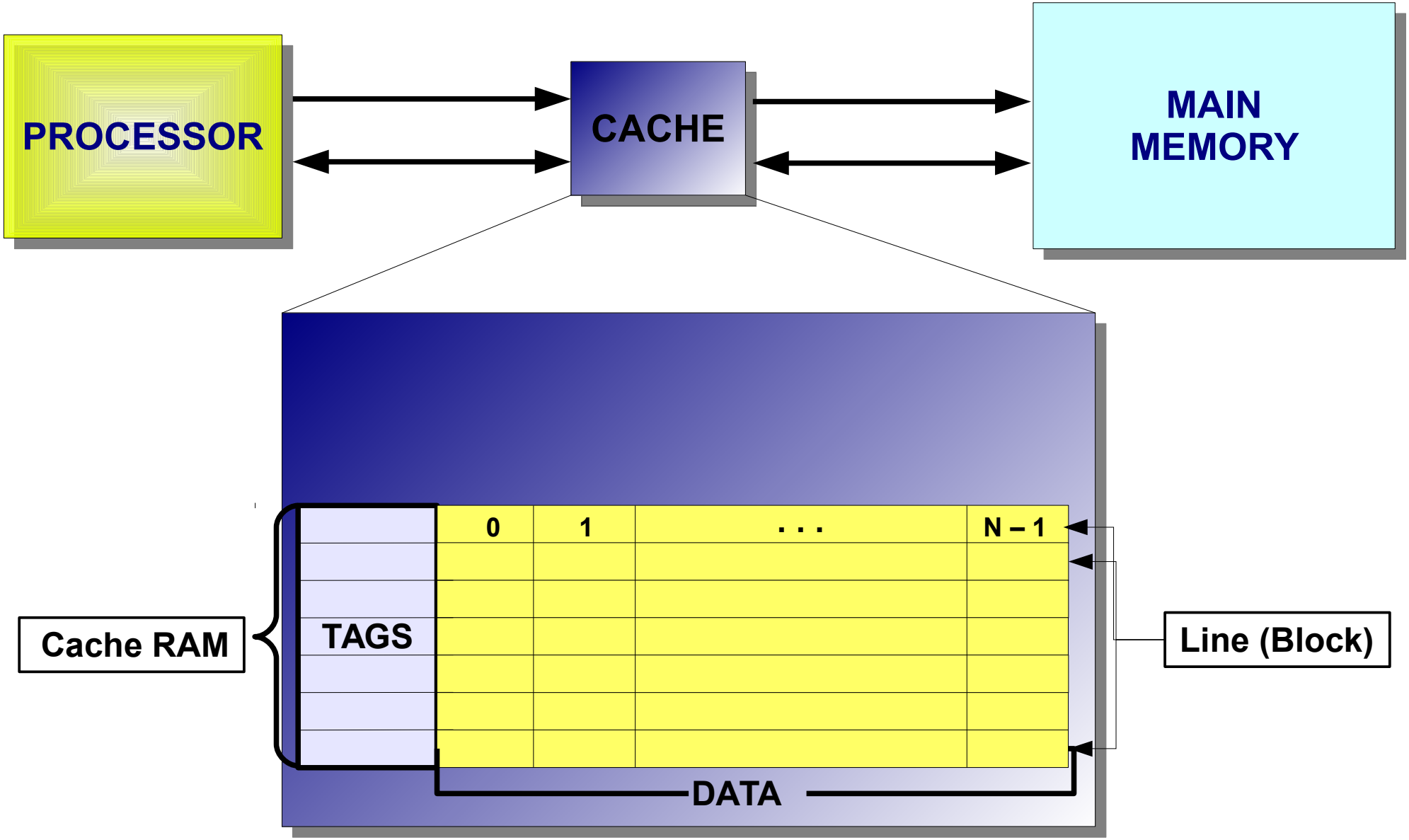
Inside a Cache



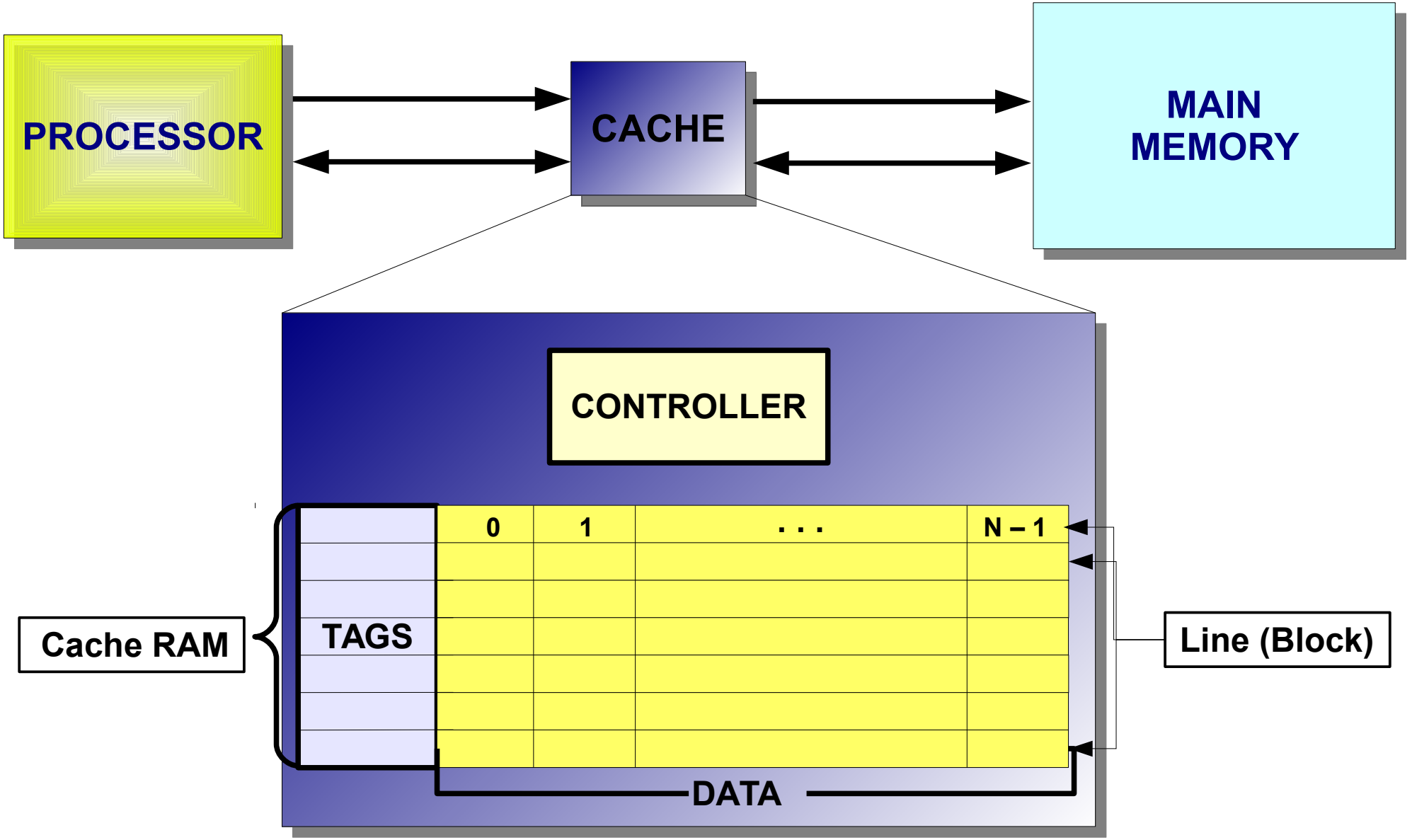
Inside a Cache



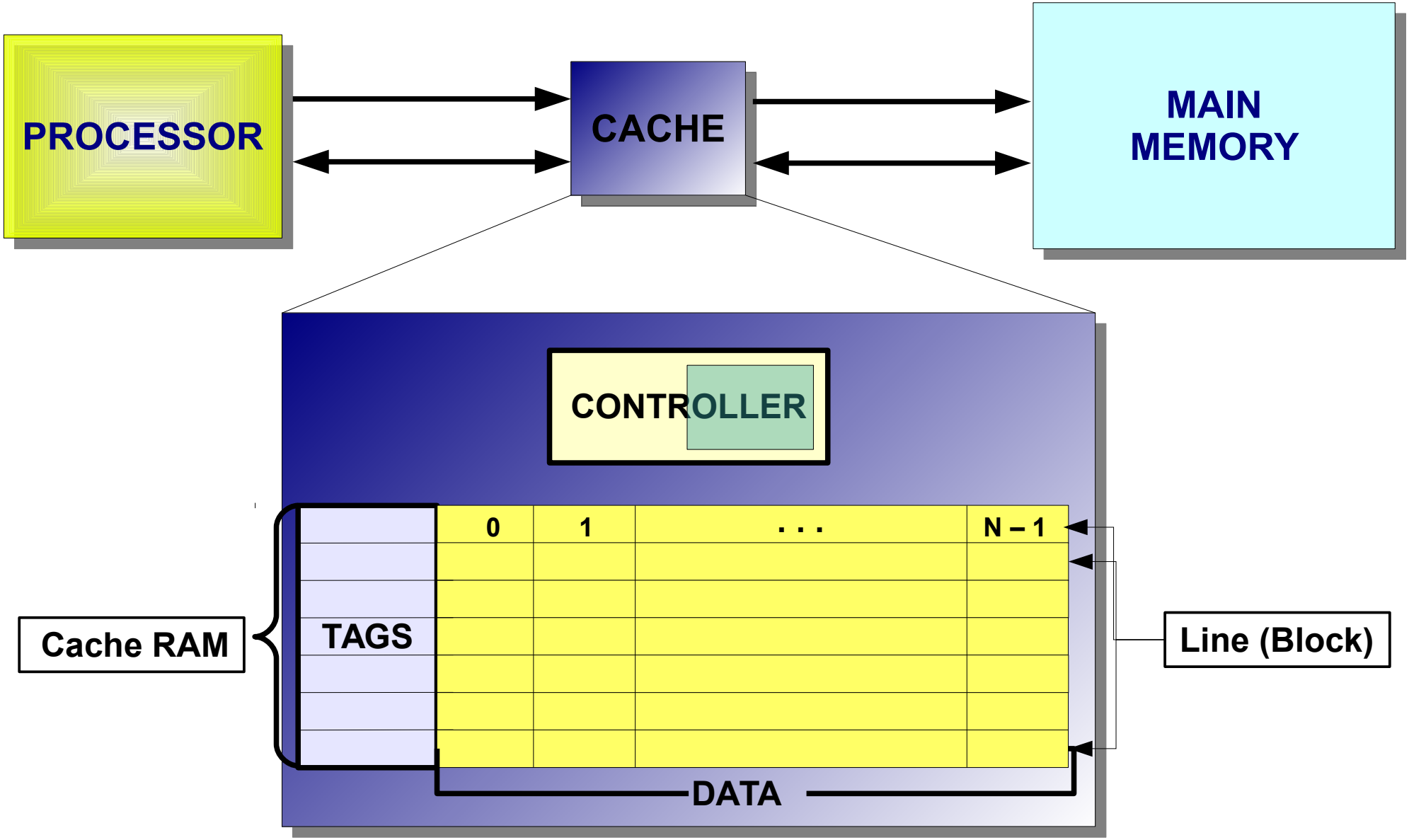
Inside a Cache



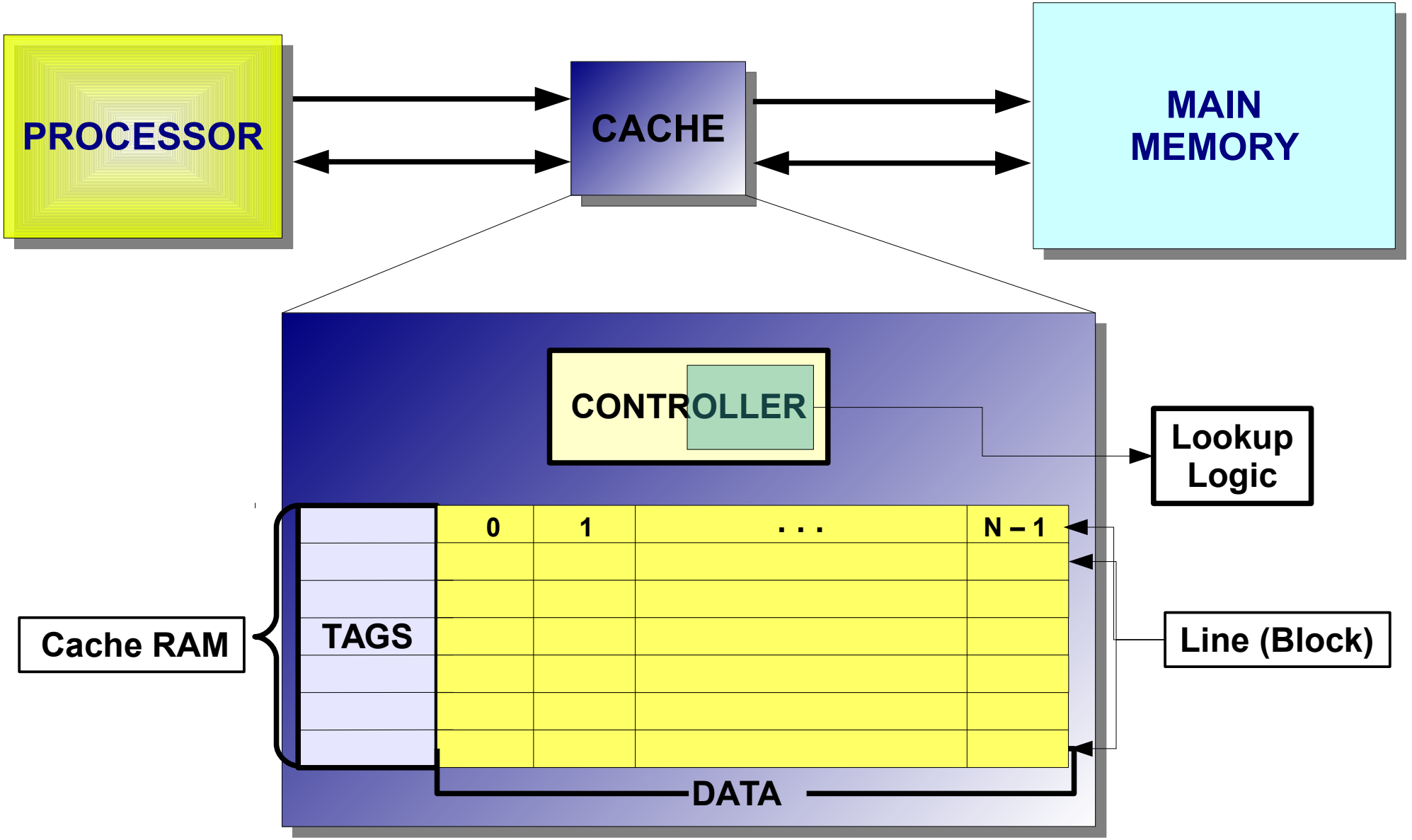
Inside a Cache



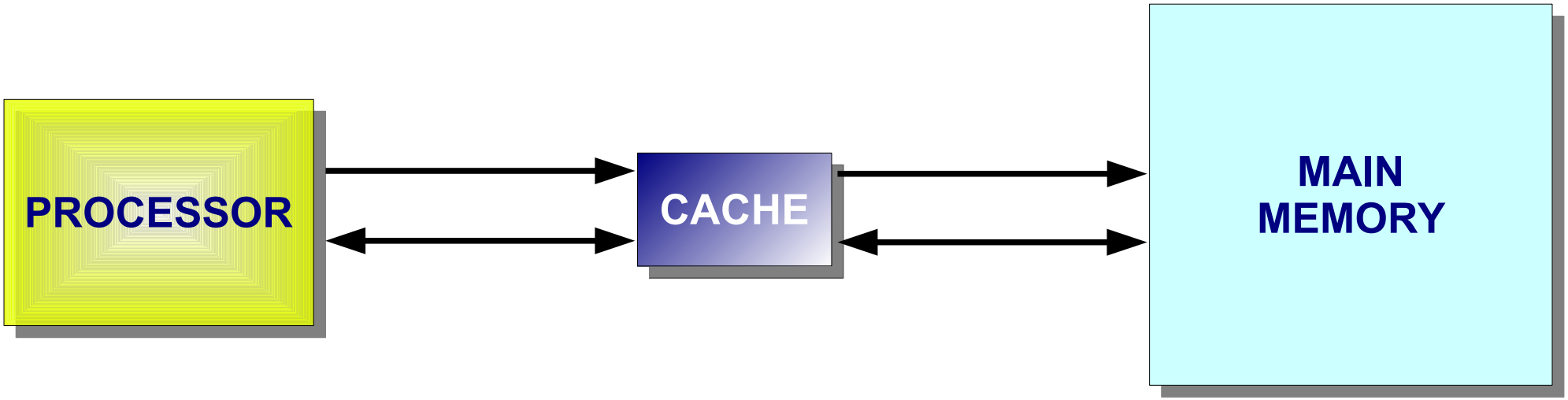
Inside a Cache



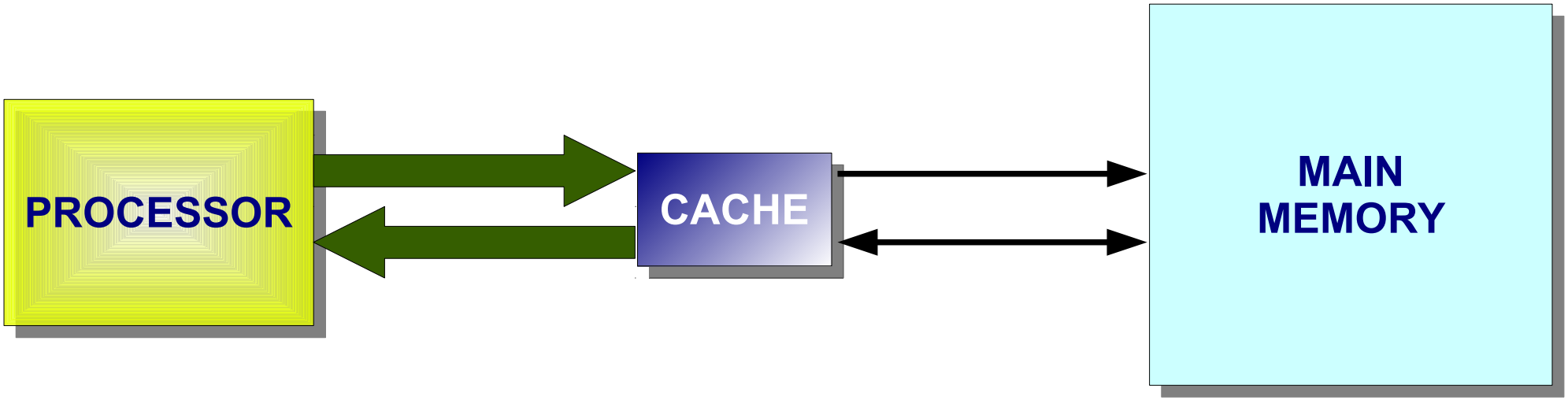
Inside a Cache



Cache Access Time

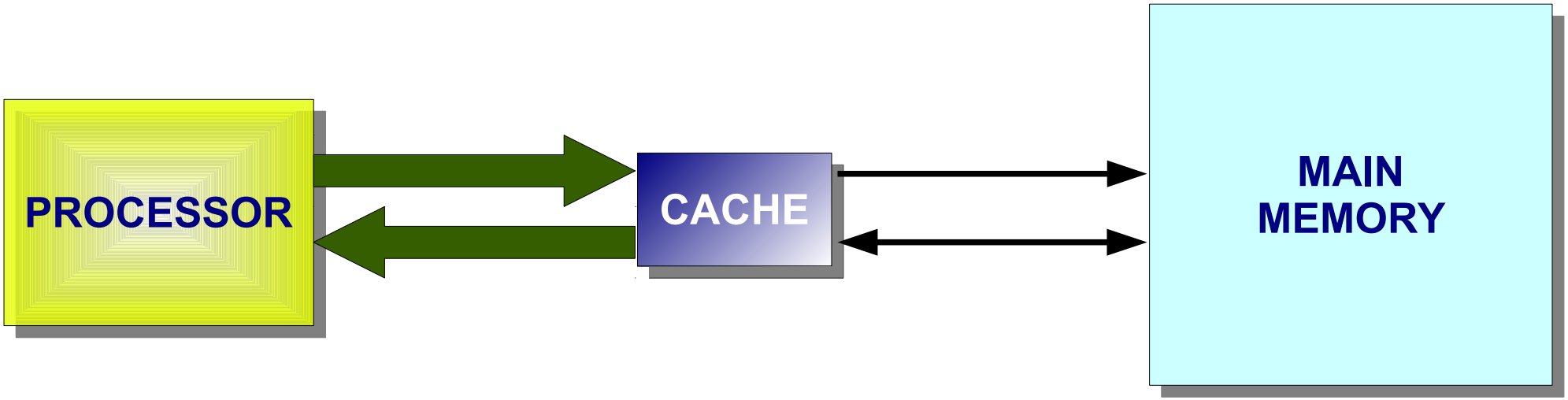


Cache Access Time

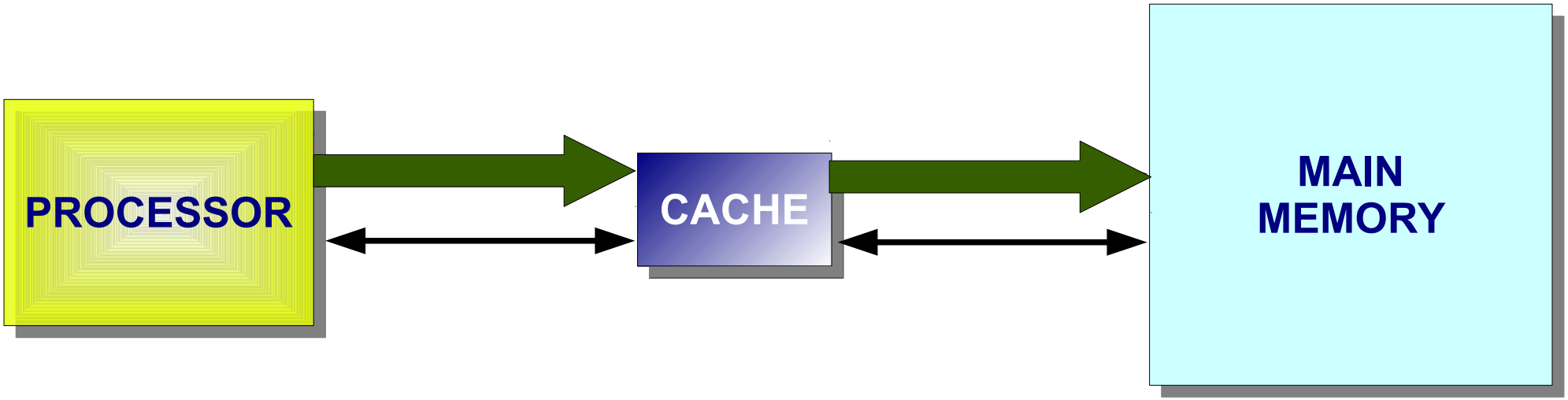


Cache Access Time

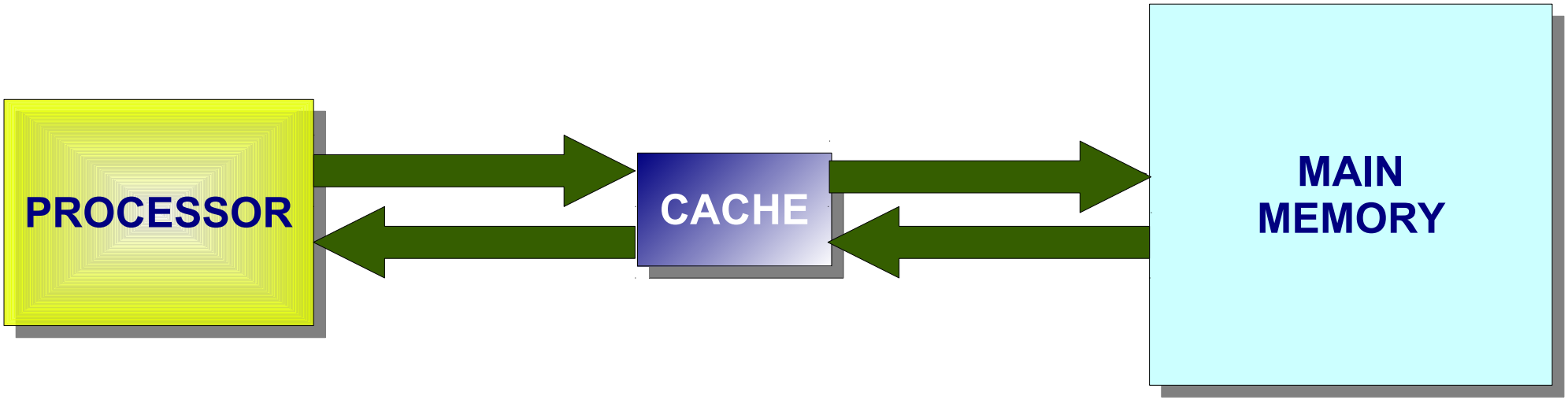
- Hit Time



Cache Access Time

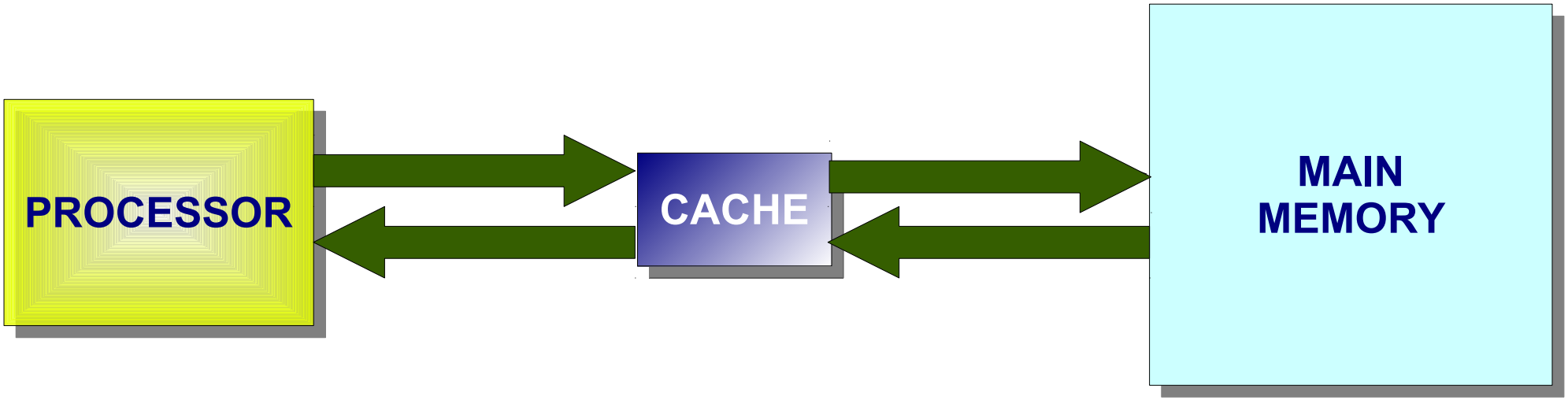


Cache Access Time



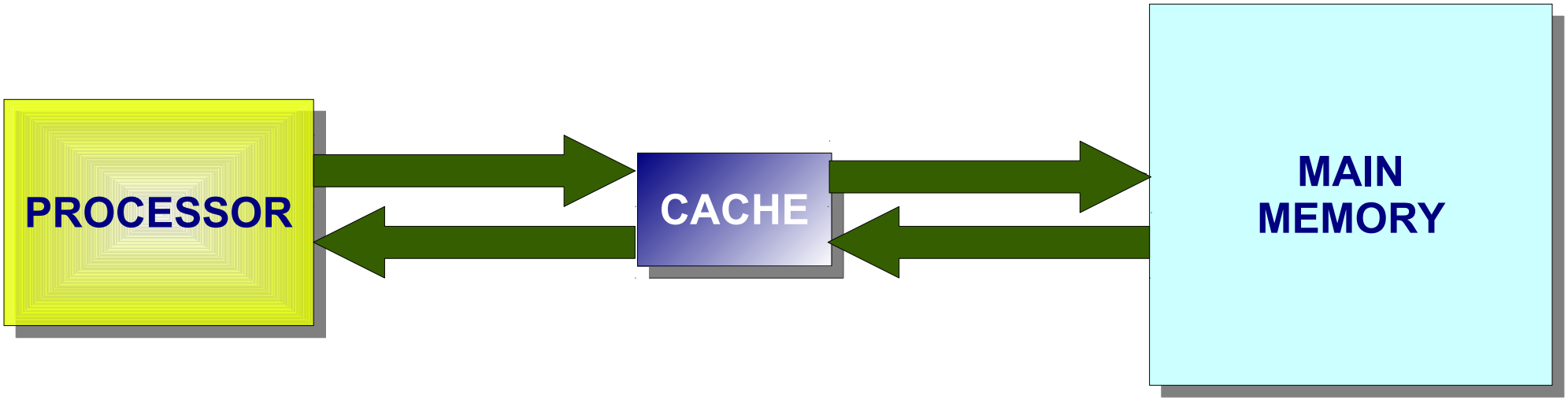
Cache Access Time

- Miss Penalty



Cache Access Time

- Hit Time, Miss Penalty



$$Average\ Memory\ Access\ Time = Hit\ Time + \underbrace{Miss\ rate \times Miss\ penalty}_{\text{Stall Time}}$$

Processor Performance – No Cache

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles
- Ignoring memory access, Clocks Per Instruction (CPI) =

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles
- Ignoring memory access, Clocks Per Instruction (CPI) = 1

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles
- Ignoring memory access, Clocks Per Instruction (CPI) = 1
- What is the CPI including memory accesses?

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles
- Ignoring memory access, Clocks Per Instruction (CPI) = 1
- What is the CPI including memory accesses?
- Assuming no memory data access:

Processor Performance – No Cache

- 5GHz processor, cycle time = 0.2ns
- Memory access time = 100ns = 500 cycles
- Ignoring memory access, Clocks Per Instruction (CPI) = 1
- What is the CPI including memory accesses?
- Assuming no memory data access:
  - $CPI_{no\text{-}cache}$ = 1 + #stall cycles
  - 1 + 500 = 501

Processor + Cache Performance

- Hit Rate = 0.95

Processor + Cache Performance

- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle

Processor + Cache Performance

- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle
- $CPI_{with\text{-}cache}$ = 1 + #stall cycles

Processor + Cache Performance

- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle
- $CPI_{with\text{-}cache}$ = 1 + #stall cycles
- #stall cycles = ?
  -
  -

Processor + Cache Performance

- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle
- $CPI_{with\text{-}cache}$ = 1 + #stall cycles
- #stall cycles = ?
  - $stall\ cycles = Miss\ Rate \cdot Miss\ Penalty$
  - $stall\ cycles = 0.05 \cdot 500 = 25$

Processor + Cache Performance

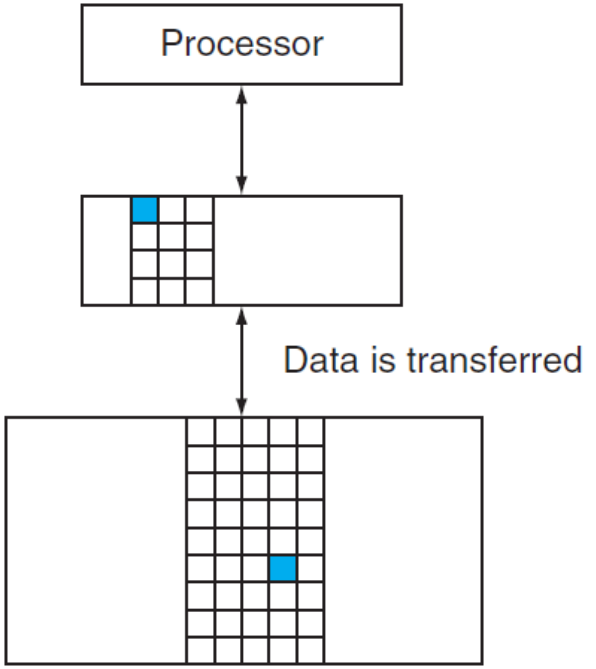
- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle
- $CPI_{with\text{-}cache}$ = 1 + #stall cycles
- #stall cycles = ?
  - $stall\ cycles = Miss\ Rate \cdot Miss\ Penalty$
  - $stall\ cycles = 0.05 \cdot 500 = 25$
- $CPI_{with\text{-}cache}$ = 26

Processor + Cache Performance

- Hit Rate = 0.95
- L1 Access Time = 0.2 ns = 1 cycle
- $CPI_{with\text{-}cache}$ = 1 + #stall cycles
- #stall cycles = ?
  - $stall\ cycles = Miss\ Rate \cdot Miss\ Penalty$
  - $stall\ cycles = 0.05 \cdot 500 = 25$
- $CPI_{with\text{-}cache}$ = 26
- Increase in performance = 501/26 = 19.3

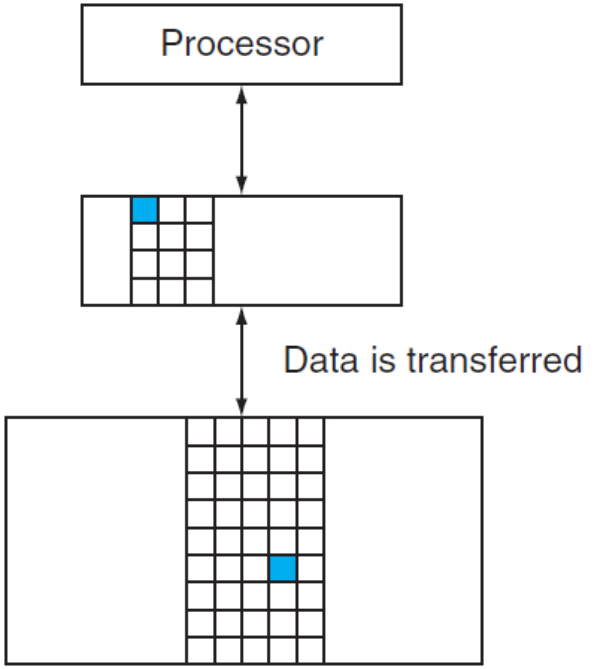
Cache Terms

- cache hit – An access where the data is found in the cache.



Cache Terms

- cache hit – An access where the data is found in the cache.
- cache miss -- an access which isn't
-



Cache Terms

- cache hit – An access where the data is found in the cache.
- cache miss -- an access which isn’t
- hit time -- time to access the cache
- miss penalty -- time to move data from lower level to upper, then to cpu

Cache Terms

- hit rate -- percentage of cache hits

Cache Terms

- hit rate -- percentage of cache hits
- miss rate -- (1 – hit rate)

Cache Terms

- cache block size or cache line size -- the amount of data that gets transferred on a cache miss.

Cache Terms

- instruction cache -- cache that only holds instructions.
- data cache -- cache that only caches data.

Cache Performance

CPU time =

Cache Performance

$$\text{CPU time} = (\text{CPU execution clock cycles} \times \text{Clock cycle time}$$

Cache Performance

$$\text{CPU time} = (\text{CPU execution clock cycles} + \text{Memory-stall clock cycles}) \times \text{Clock cycle time}$$

Cache Performance

$$\text{CPU time} = (\text{CPU execution clock cycles} + \text{Memory-stall clock cycles}) \times \text{Clock cycle time}$$

$$\text{Memory-stall clock cycles} =$$

Cache Performance

$$\text{CPU time} = (\text{CPU execution clock cycles} + \text{Memory-stall clock cycles}) \times \text{Clock cycle time}$$

$$\text{Memory-stall clock cycles} = \frac{\text{Memory accesses}}{\text{Program}} \times \text{Miss rate} \times \text{Miss penalty}$$

Cache Performance

- Assume the miss rate of an instruction cache is 2% and the miss rate of the data cache is 4%. If a processor has a CPI of 2 without any memory stalls and the miss penalty is 100 cycles for all misses, determine how much faster a processor would run with a perfect cache that never missed. Assume the frequency of all loads and stores is 36%.

Average memory stall cycles for (a) Instruction Cache (b) Data Cache?

Multilevel Caches

- Primary cache attached to CPU
  - Small, but fast

Multilevel Caches

- Primary cache attached to CPU
  - Small, but fast
- Level-2 cache services misses from primary cache
  - Larger, slower, but still faster than main memory

Multilevel Caches

- Primary cache attached to CPU
  - Small, but fast
- Level-2 cache services misses from primary cache
  - Larger, slower, but still faster than main memory
- Main memory services L-2 cache misses

Multilevel Caches

- Primary cache attached to CPU
  - Small, but fast
- Level-2 cache services misses from primary cache
  - Larger, slower, but still faster than main memory
- Main memory services L-2 cache misses
- Some high-end systems include L-3 cache

Multilevel Cache Example

- Given
  - CPU base CPI = 1, clock rate = 4GHz
  - Miss rate/instruction = 2%
  - Main memory access time = 100ns

Multilevel Cache Example

- Given
  - CPU base CPI = 1, clock rate = 4GHz
  - Miss rate/instruction = 2%
  - Main memory access time = 100ns
- With just primary cache
  - Miss penalty =
  - Effective CPI =

Multilevel Cache Example

- Given
  - CPU base CPI = 1, clock rate = 4GHz
  - Miss rate/instruction = 2%
  - Main memory access time = 100ns
- With just primary cache
  - Miss penalty = 100ns/0.25ns = 400 cycles
  - Effective CPI = 1 + 0.02 × 400 = 9

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty =

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles
- Primary miss with L-2 miss
  -

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles
- Primary miss with L-2 miss
  - Extra penalty =

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles
- Primary miss with L-2 miss
  - Extra penalty = 400 cycles
- CPI =

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles
- Primary miss with L-2 miss
  - Extra penalty = 400 cycles
- CPI = 1 + 0.02 × 20 + 0.005 × 400 = 3.4
- Performance ratio =

Example (cont.)

- Now add L-2 cache
  - Access time = 5ns
  - Global miss rate to main memory = 0.5%
- Primary miss with L-2 hit
  - Penalty = 5ns/0.25ns = 20 cycles
- Primary miss with L-2 miss
  - Extra penalty = 400 cycles
- CPI = 1 + 0.02 × 20 + 0.005 × 400 = 3.4
- Performance ratio = 9/3.4 = 2.6

Multilevel Cache Considerations

- Primary cache
  -
- L-2 cache
  -
  -
- Results
  -
  -

Multilevel Cache Considerations

- Primary cache
  - Focus on minimal hit time
- L-2 cache
  -
  -
- Results
  -
  -

Multilevel Cache Considerations

- Primary cache
  - Focus on minimal hit time
- L-2 cache
  - Focus on low miss rate to avoid main memory access
  - Hit time has less overall impact
- Results
  -
  -

Multilevel Cache Considerations

- Primary cache
  - Focus on minimal hit time
- L-2 cache
  - Focus on low miss rate to avoid main memory access
  - Hit time has less overall impact
- Results
  - L-1 cache usually smaller than a single cache
  - L-1 block size smaller than L-2 block size

Memory Hierachy – Recap

- What programmers want: Unlimited amounts of memory with low latency
  - –
  - –

Memory Hierachy – Recap

- What programmers want: Unlimited amounts of memory with low latency
  - Reality: Memory latency is huge
  - Reality: Fast memory technology is more expensive per bit than slower memory

Memory Hierachy – Recap

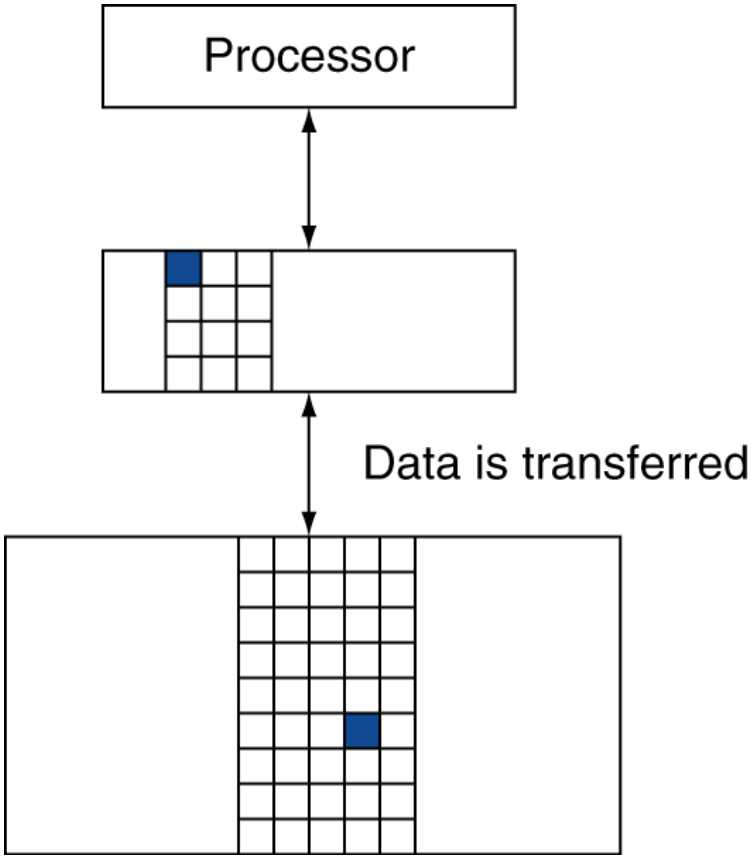
- Solution: organize memory system into a hierarchy
  -
  -

Memory Hierachy – Recap

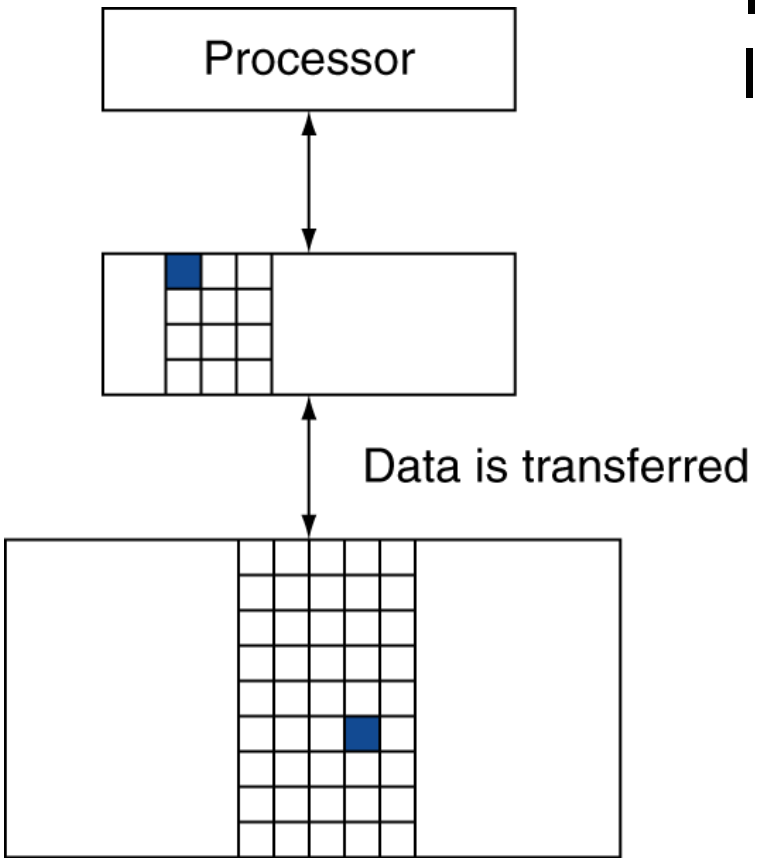
- Solution: organize memory system into a hierarchy
  - Entire addressable memory space available in largest, slowest memory
  - Incrementally smaller and faster memories, each containing a subset of the memory below it

Memory Hierarchy Levels

- Block (aka line): unit of copying

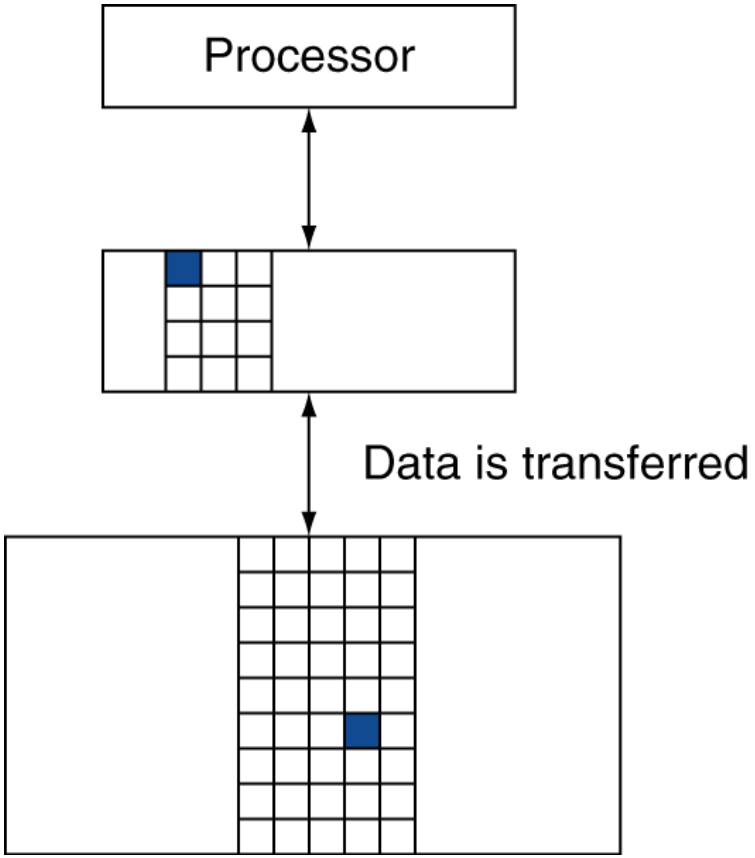


Memory Hierarchy Levels



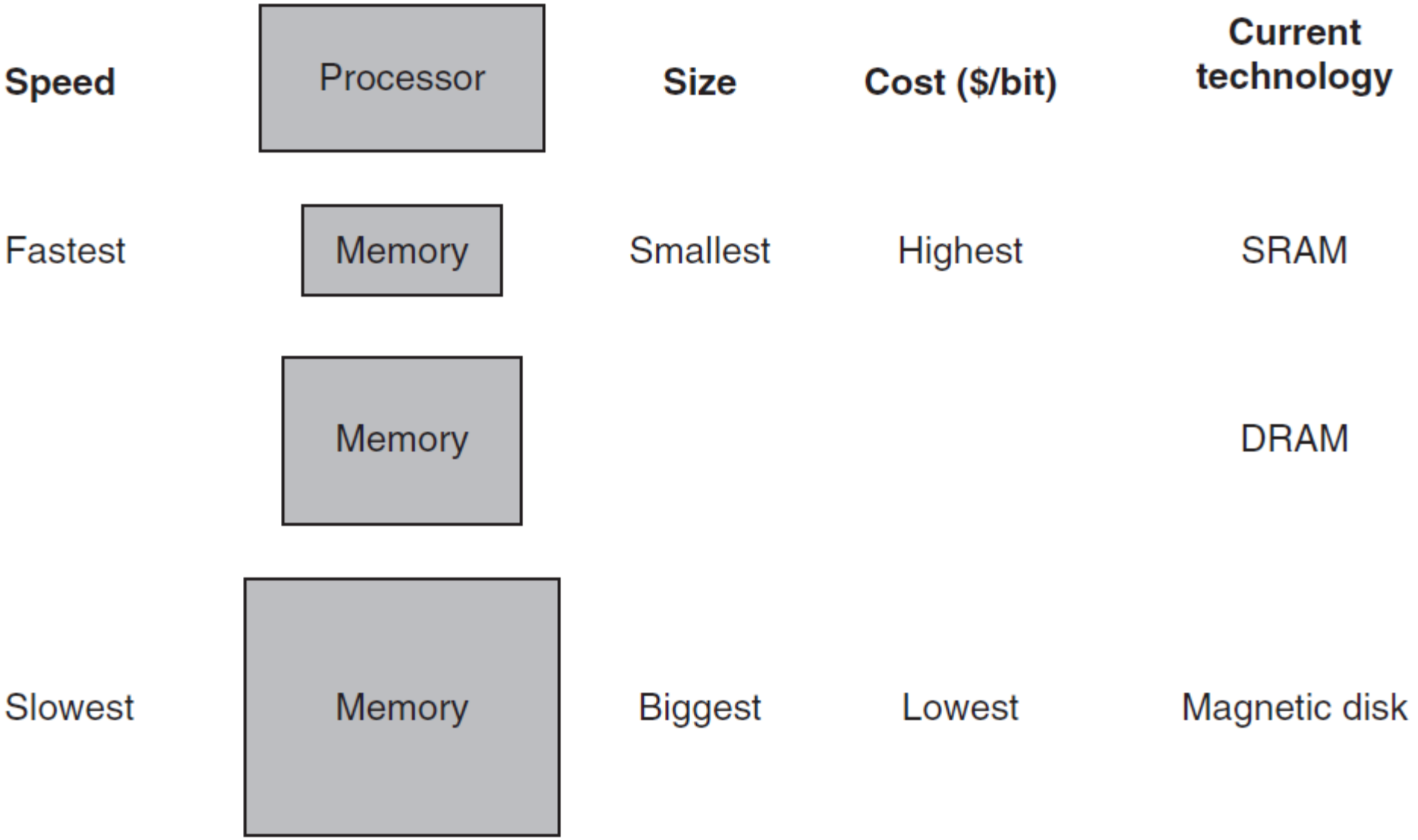
- If accessed data is present in upper level
  - Hit: access satisfied by upper level
  - Hit ratio: hits/accesses

Memory Hierarchy Levels

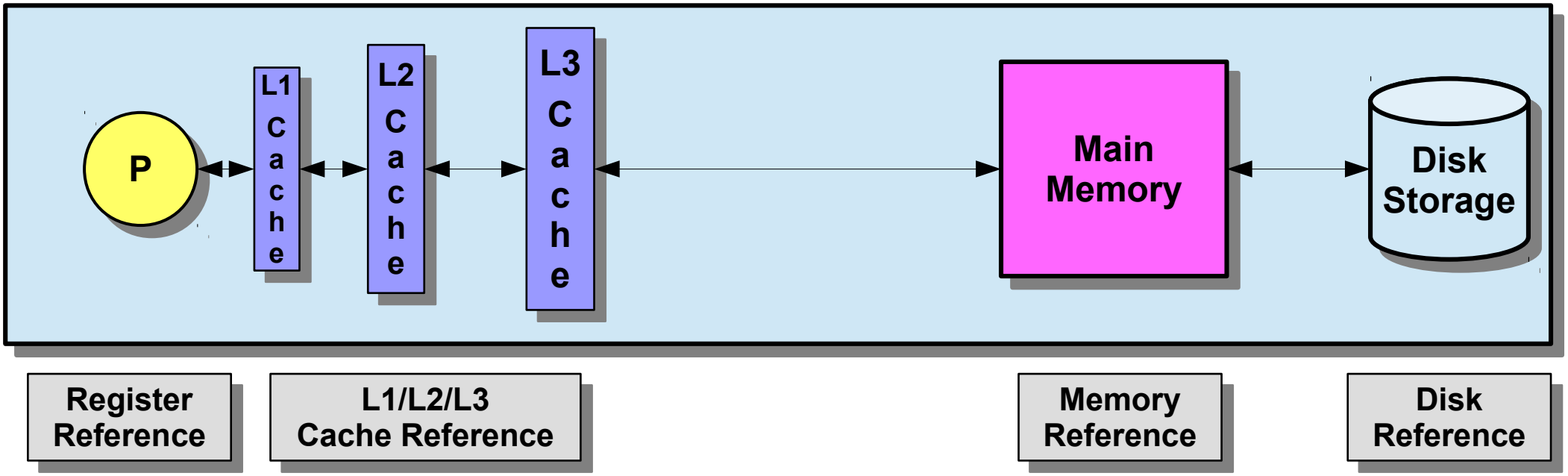


- If accessed data is absent
  - Miss: block copied from lower level
    - Time taken: miss penalty
    - Miss ratio: misses/accesses
      = 1 – hit ratio
  - Then accessed data supplied from upper level

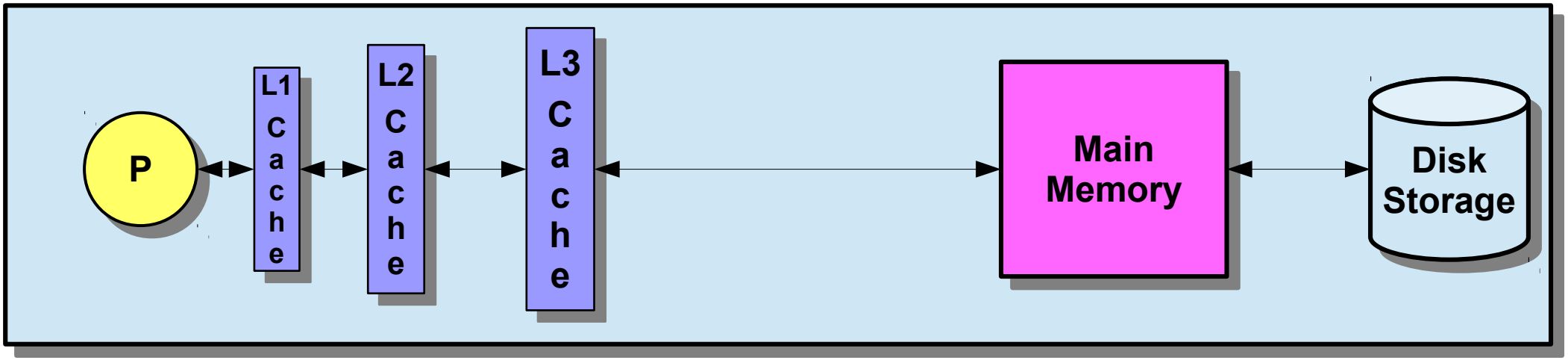
Memory Hierarchy



Memory Hierarchy



Memory Hierarchy



Register Reference

L1/L2/L3 Cache Reference

Memory Reference

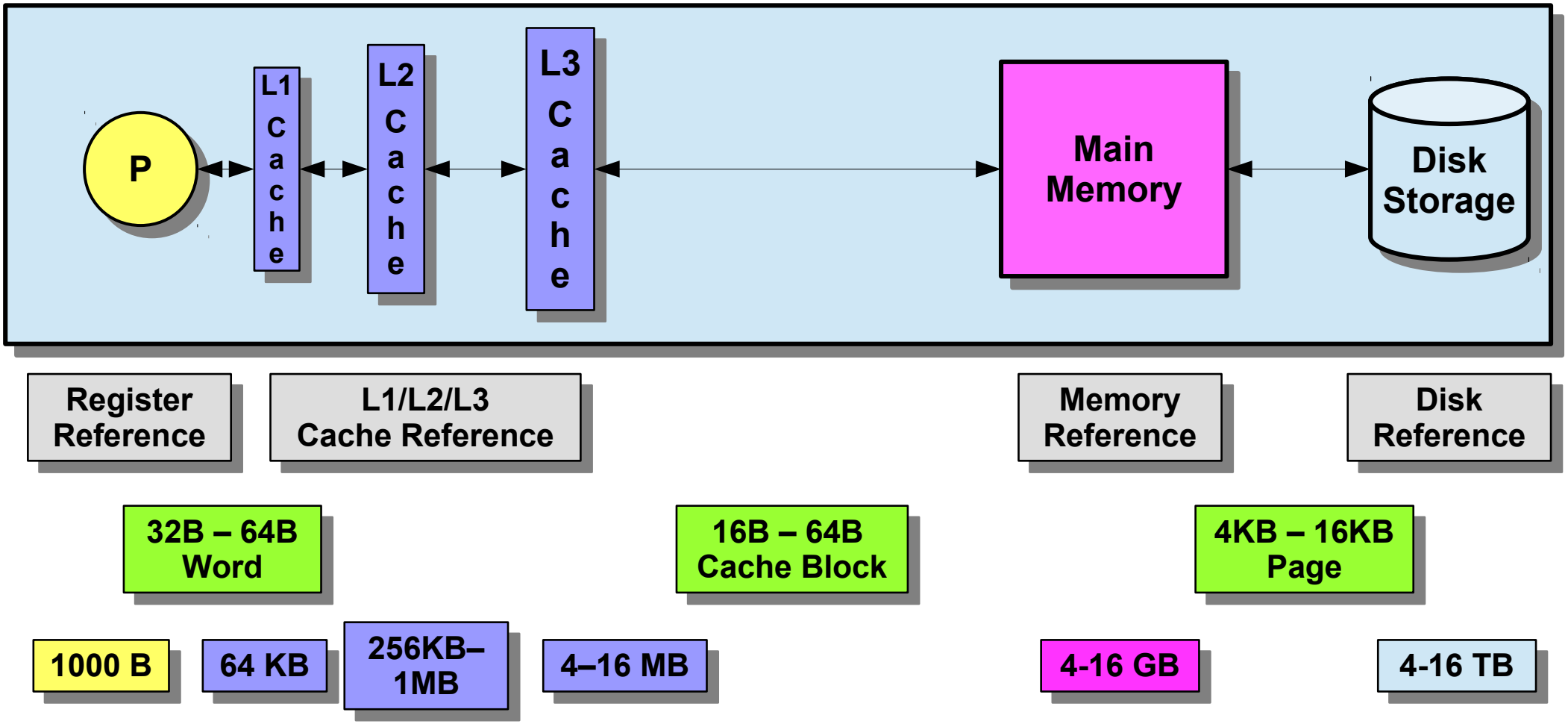
Disk Reference

32B – 64B Word

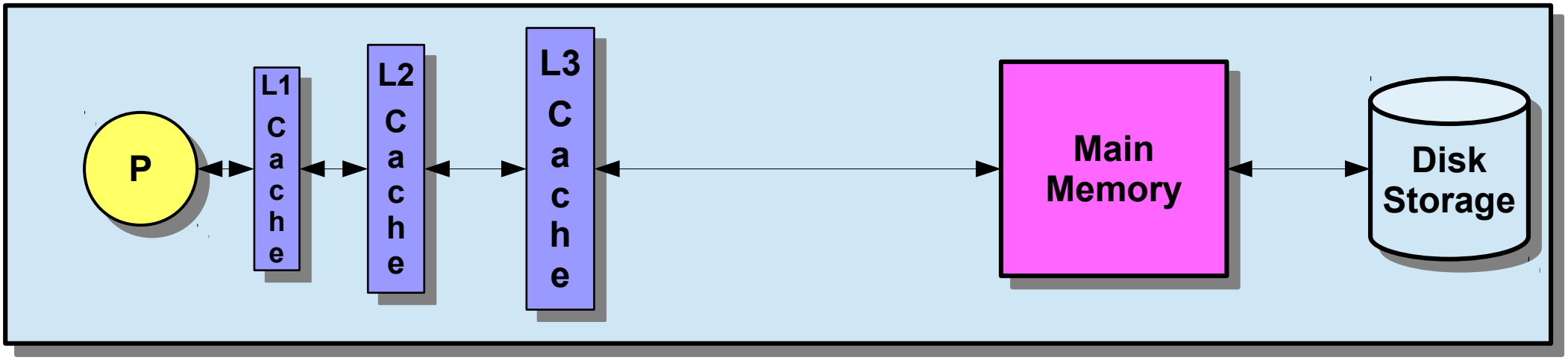
16B – 64B Cache Block

4KB – 16KB Page

Memory Hierarchy



Memory Hierarchy



Register Reference	L1/L2/L3 Cache Reference			Memory Reference	Disk Reference
32B – 64B Word		16B – 64B Cache Block		4KB – 16KB Page	
1000 B	64 KB	256KB–1MB	4–16 MB	4-16 GB	4-16 TB
300ps	1 ns	3 - 10 ns	10 - 25 ns	50 - 100 ns	5 - 10 ms

Memory Hierarchy – Other Topics

- Main Memory, DRAM
- Virtual Memory
- Non-Volatile Memory
- Persistent NVM

Module Outline

- CPU – Memory interaction
- Organization of memory modules
- Cache memory – Mapping and replacement policies.