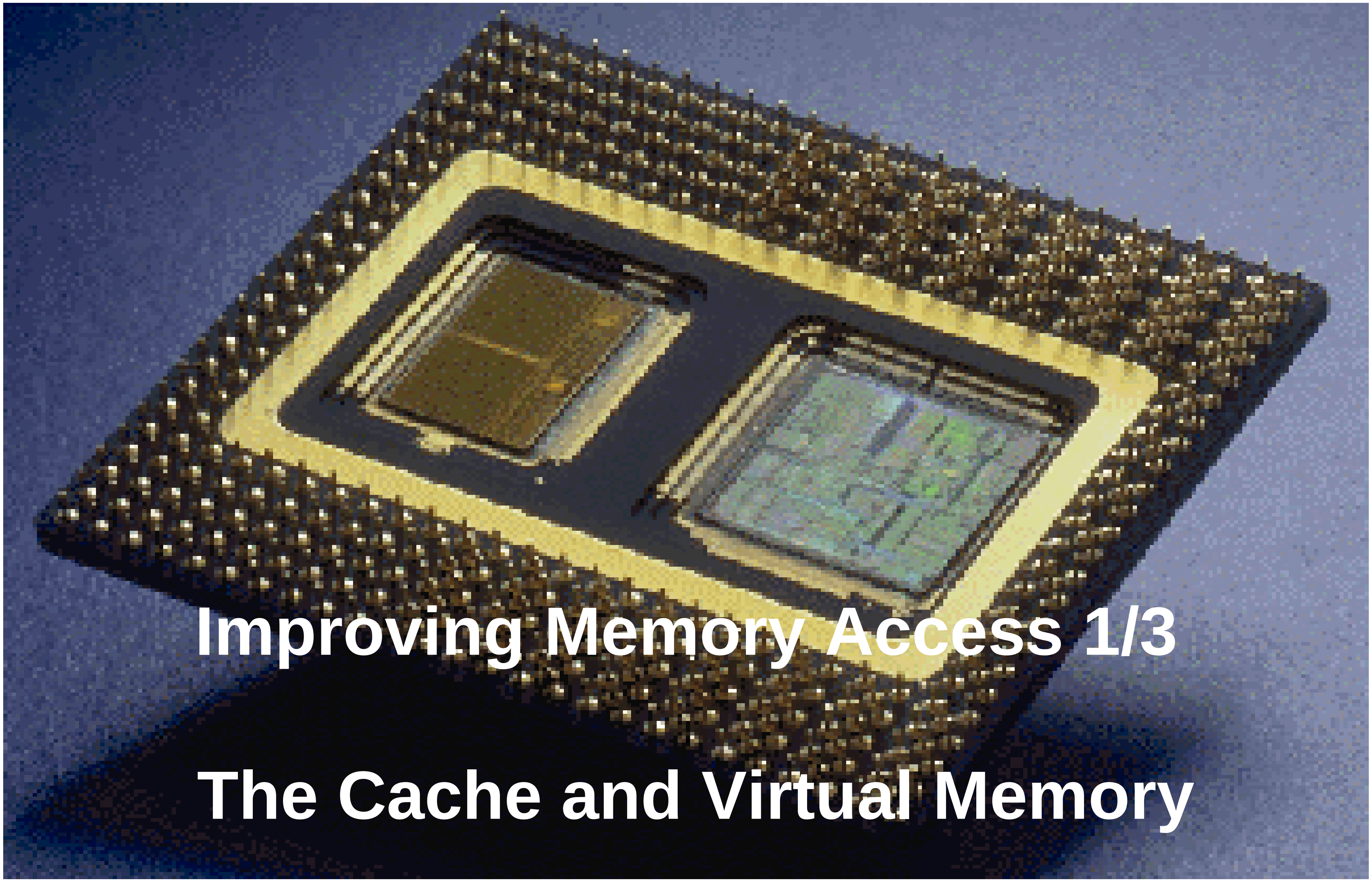
EECS 322 Computer Architecture



Improving Memory Access 1/3

The Cache and Virtual Memory

Principle of Locality

- Principle of Locality

  states that programs access a relatively small portion of their address space at <u>any instance of time</u>

- <u>Two types of locality</u>

  - <u>Temporal locality</u> (locality in time)

    If an item is referenced, then

    <u>the same</u> item will tend to be referenced soon

    "the tendency to reuse recently accessed data items"

  - <u>Spatial locality</u> (locality in space)

    If an item is referenced, then

    <u>nearby</u> items will be referenced soon

    "the tendency to reference nearby data items"

Cache Terminology

A hit if the data requested by the CPU is in the upper level

Hit rate or Hit ratio

is the fraction of accesses found in the upper level

Hit time

is the time required to access data in the upper level
= <detection time for hit or miss> + <hit access time>

A miss if the data is not found in the upper level

Miss rate or (1 – hit rate)

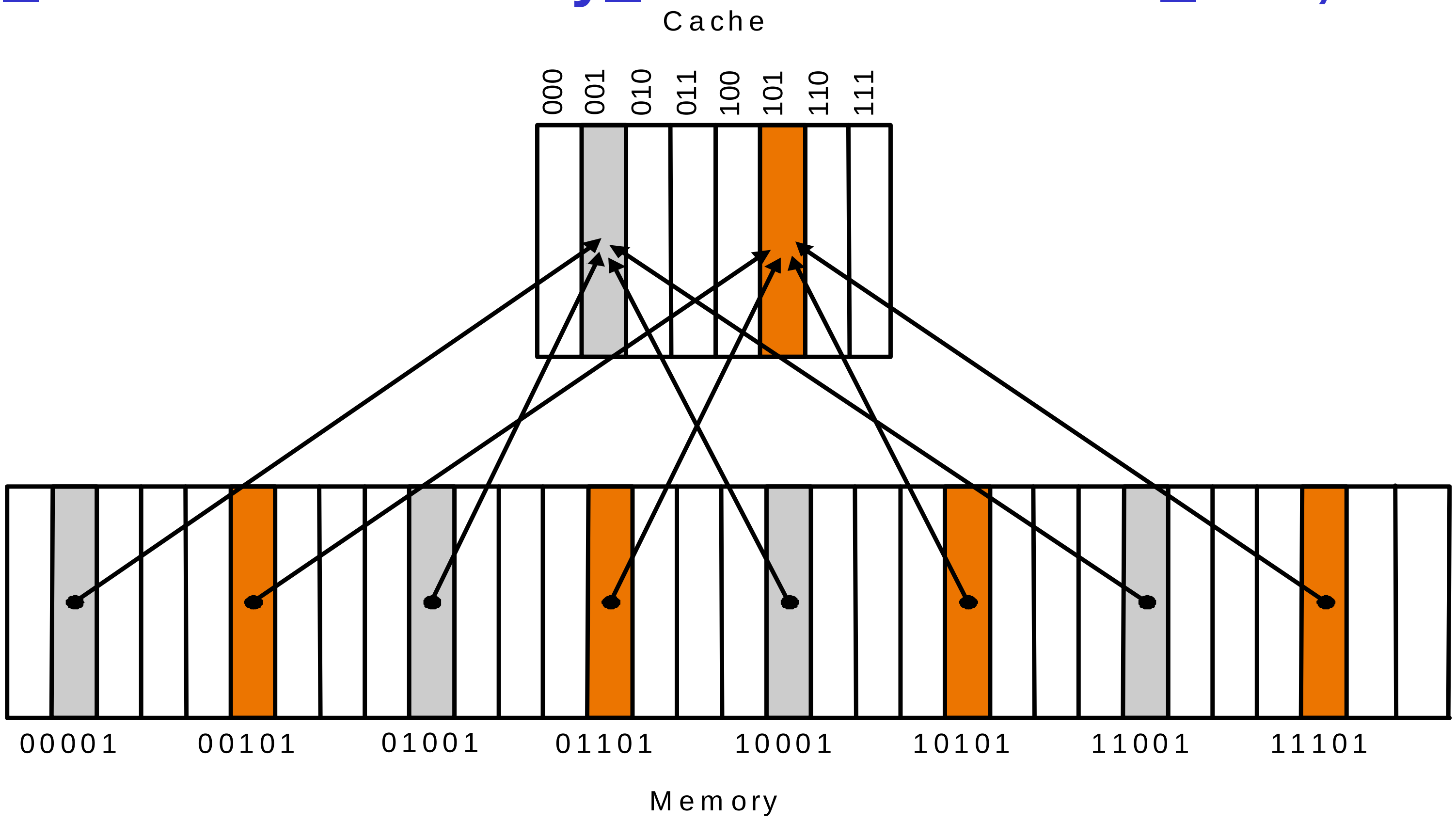
is the fraction of accesses not found in the upper level

Miss penalty

is the time required to access data in the lower level
= <lower access time>+<reload processor time>

Direct Mapped Cache

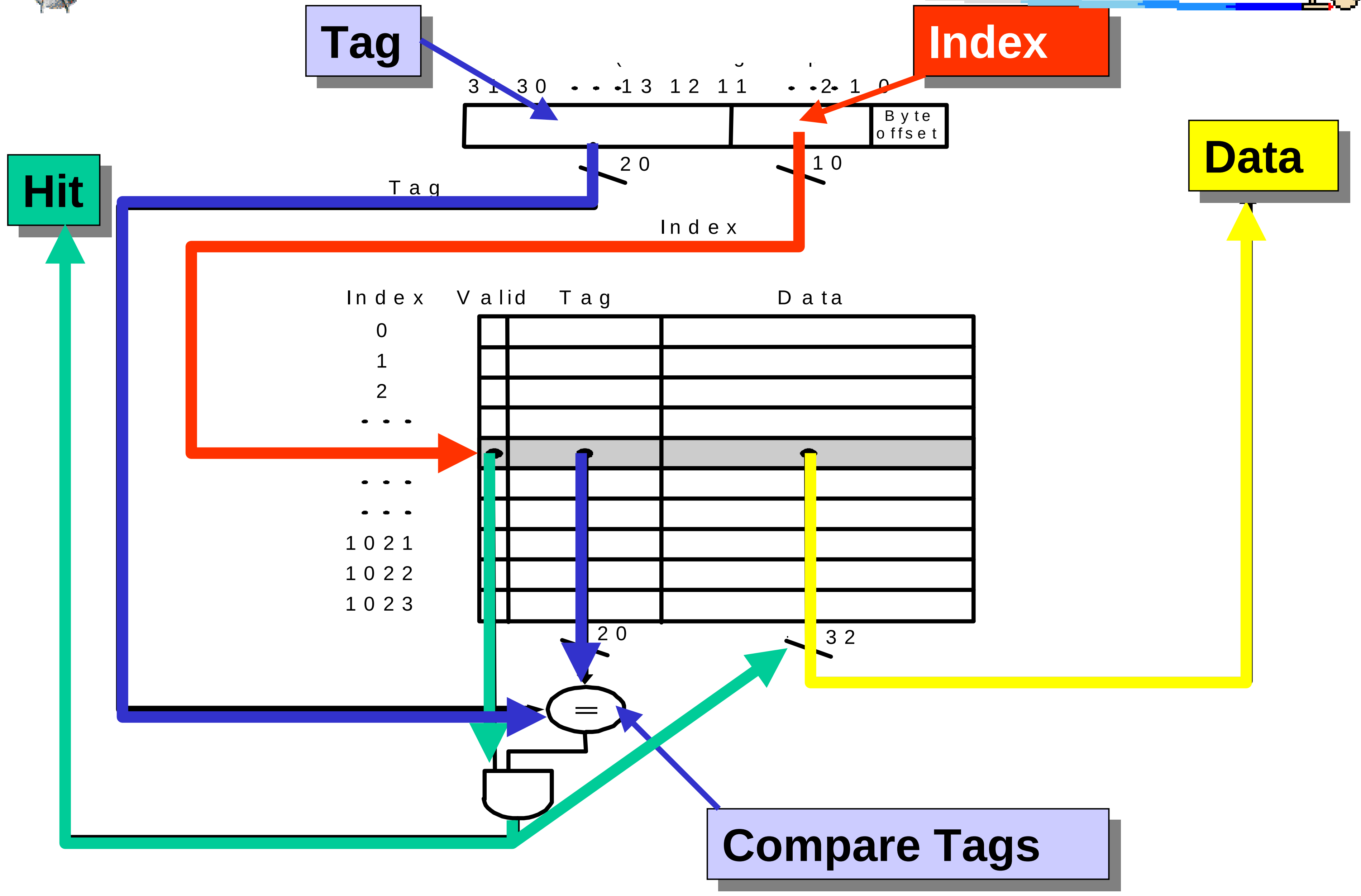
- **Direct Mapped:** **assign the cache location based on the address of the word in memory**
- **cache_address = memory_address % cache_size;**



Observe there is a Many-to-1 memory to cache relationship

Figure 7.7

Direct Mapped Cache: Mips Architecture



Bits in a Direct Mapped Cache

How many total bits are required for a direct mapped cache
with 64KB (= 2^{16} KiloBytes) of data
and one word (=32 bit) blocks
assuming a 32 bit byte memory address?

Cache index width = $\log_2$ words
= $\log_2 2^{16}/4$ = $\log_2 2^{14}$ words = 14 bits

Block address width = <byte address width> – $\log_2$ word
= 32 – 2 = 30 bits

Tag size = <block address width> – <cache index width>
= 30 – 14 = 16 bits

Cache block size = <valid size>+<tag size>+<block data size>
= 1 bit + 16 bits + 32 bits = 49 bits

Total size = <Cache word size> $\times$ <Cache block size>
= 2^{14} words $\times$ 49 bits = 784 $\times$ 2^{10} = 784 Kbits = 98 KB
= 98 KB/64 KB = 1.5 times overhead

The DECStation 3100 cache

write-through cache

Always write the data into both the cache and memory and then wait for memory.

DECStation uses a write-through cache

- 128 KB total cache size (=32K words)
  - = 64 KB instruction cache (=16K words)
  - + 64 KB data cache (=16K words)

- 10 processor clock cycles to write to memory

In a gcc benchmark, 13% of the instructions are stores.

- Thus, CPI of 1.2 becomes 1.2+13%x10 = 2.5
- Reduces the performance by more than a factor of 2!

Cache schemes

write-through cache

Always write the data into both the
cache and memory and then wait for memory.

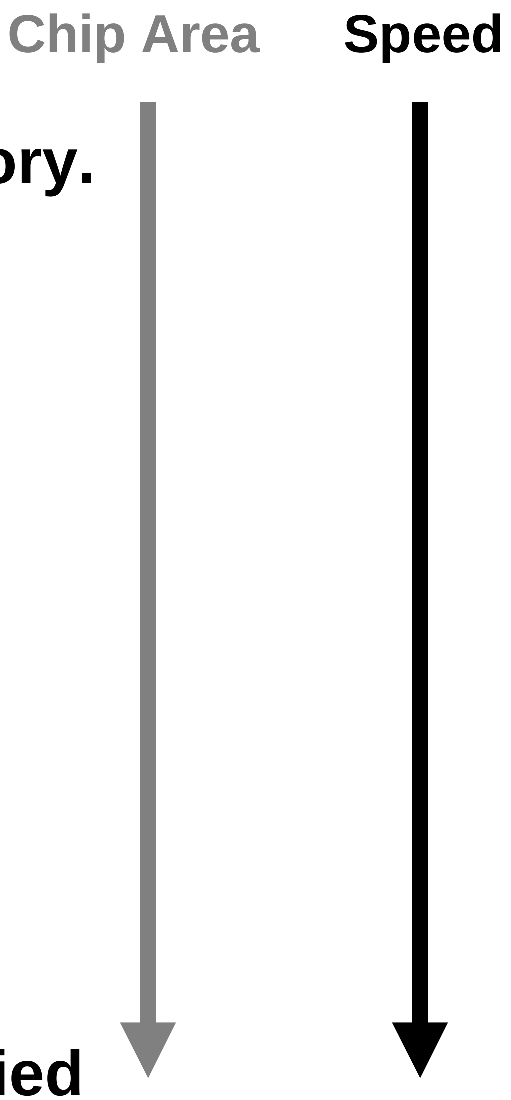
write buffer

write data into cache and write buffer.
If write buffer full processor must stall.

No amount of buffering can help
if writes are being generated faster
than the memory system can accept them.

write-back cache

Write data into the cache block and
only write to memory when block is modified
but complex to implement in hardware.



Hits vs. Misses

- Read hits
  - this is what we want!
- Read misses
  - stall the CPU, fetch block from memory,
    deliver to cache, and restart.
- Write hits
  - write-through: can replace data in cache and memory.
  - write-buffer: write data into cache and buffer.
  - write-back: write the data only into the cache.
- Write misses
  - read the entire block into the cache, then write the word.

Figure 7.9

The DECStation 3100 miss rates

- A split instruction and data cache increases the bandwidth

Benchmark Program	gcc	spice
Instruction miss rate	6.1%	1.2%
Data miss rate	2.1%	1.3%
Effective split miss rate	5.4%	1.2%
Combined miss rate	4.8%	

Why a lower miss rate?

Numerical programs tend to consist of a lot of small program loops

split cache has slightly worse miss rate

Spatial Locality

- Temporal only cache

  cache block contains only one word (No spatial locality).

- Spatial locality

  Cache block contains multiple words.

  - When a miss occurs, then fetch multiple words.
  - Advantage

    Hit ratio increases because there is a high probability that the adjacent words will be needed shortly.

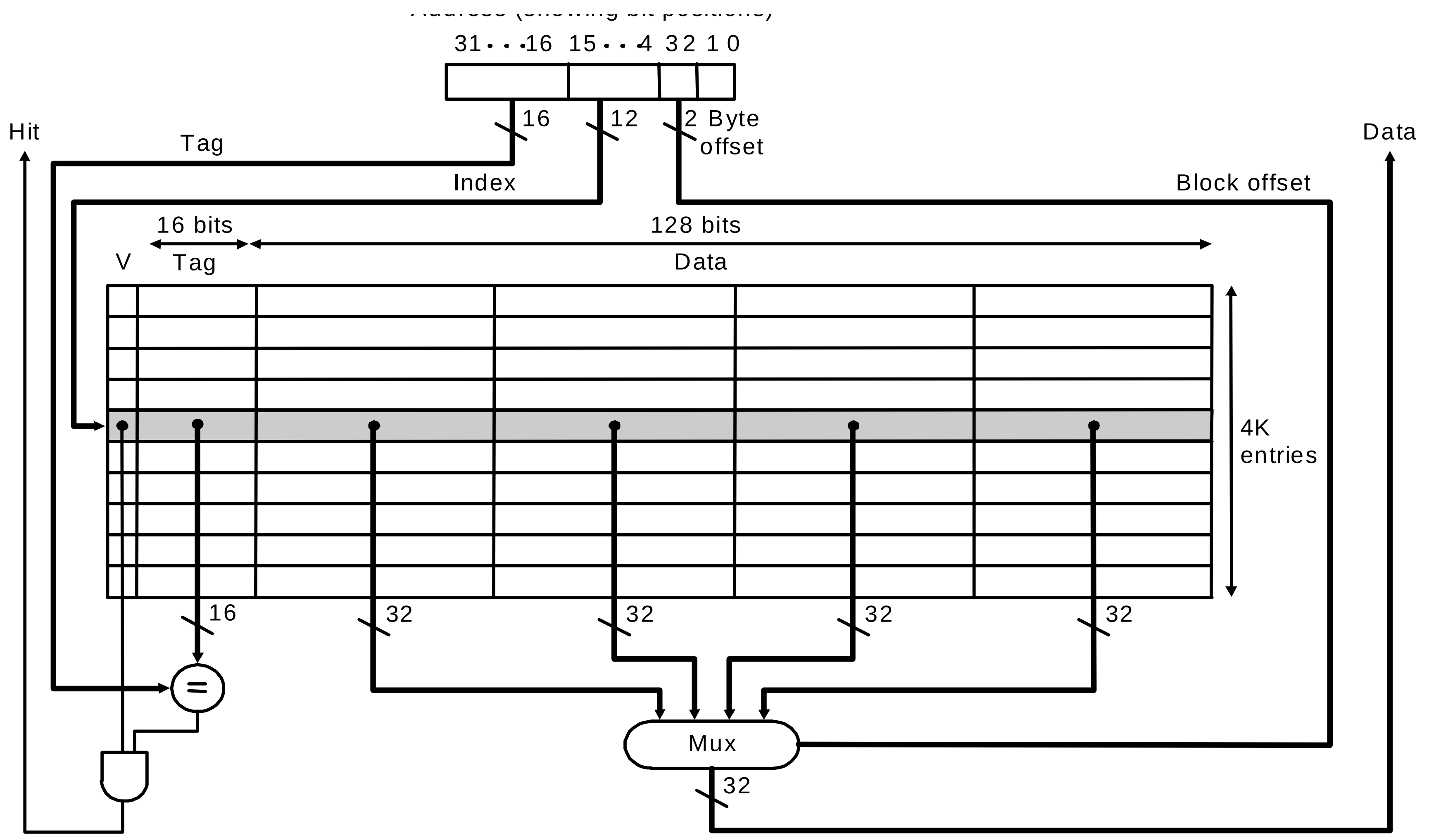
  - Disadvantage

    Miss penalty increases with block size

Figure 7.10

Spatial Locality: 64 KB cache, 4 words

- 64KB cache using four-word (16-byte word)
- 16 bit tag, 12 bit index, 2 bit block offset, 2 bit byte offset.



Performance

Figure 7.11

- Use split caches because there is more spatial locality in code:

Program Block size	gcc =1	gcc =4	spice =1	spice =4
Instruction miss rate	6.1%	2.0%	1.2%	0.3%
Data miss rate	2.1%	1.7%	1.3%	0.6%
Effective split miss rate	5.4%	1.9%	1.2%	0.4%
Combined miss rate	4.8%	4.8%		

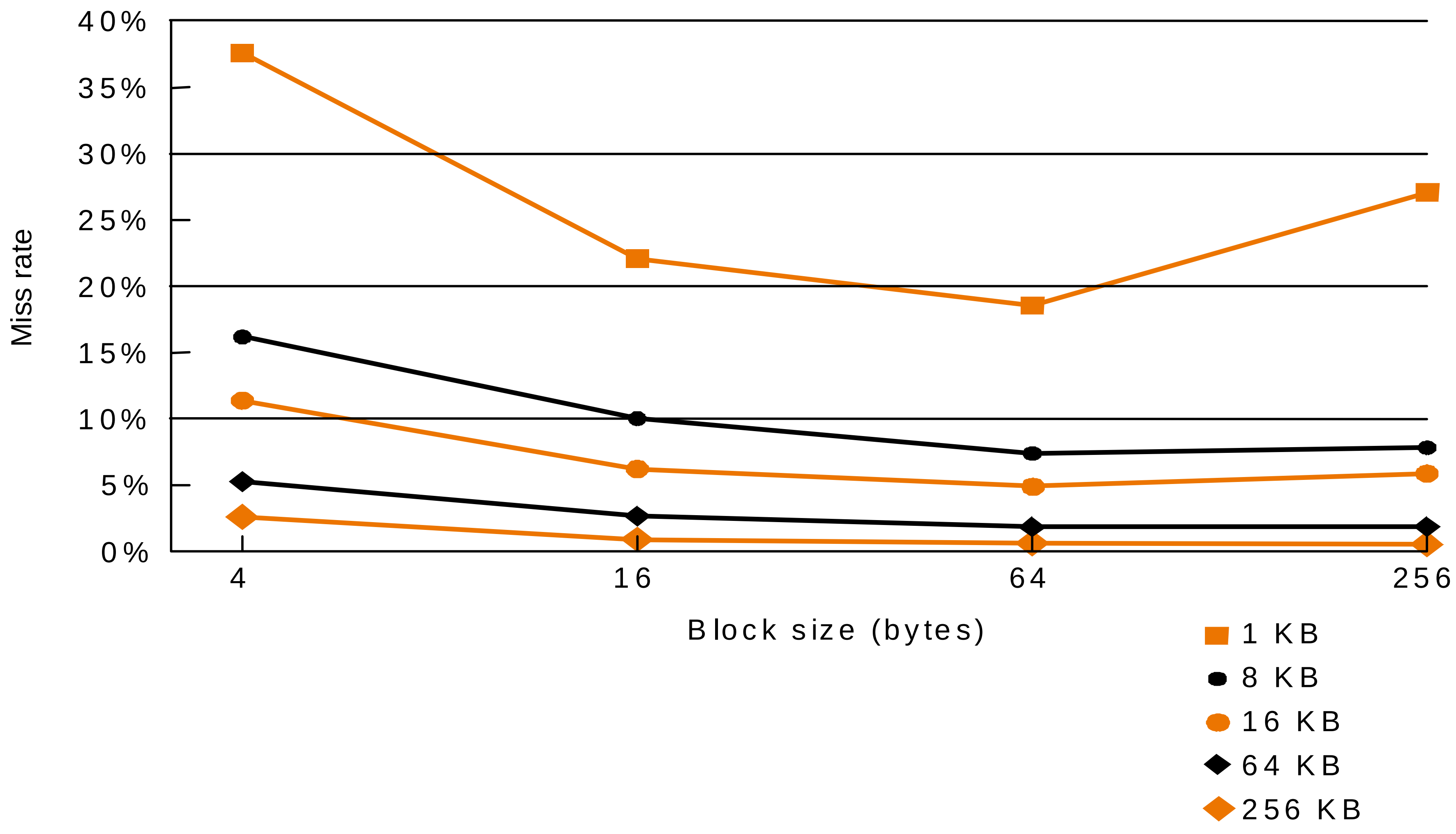
Temporal only split cache: has slightly worse miss rate

Spatial split cache: has lower miss rate

Cache Block size Performance

Figure 7.12

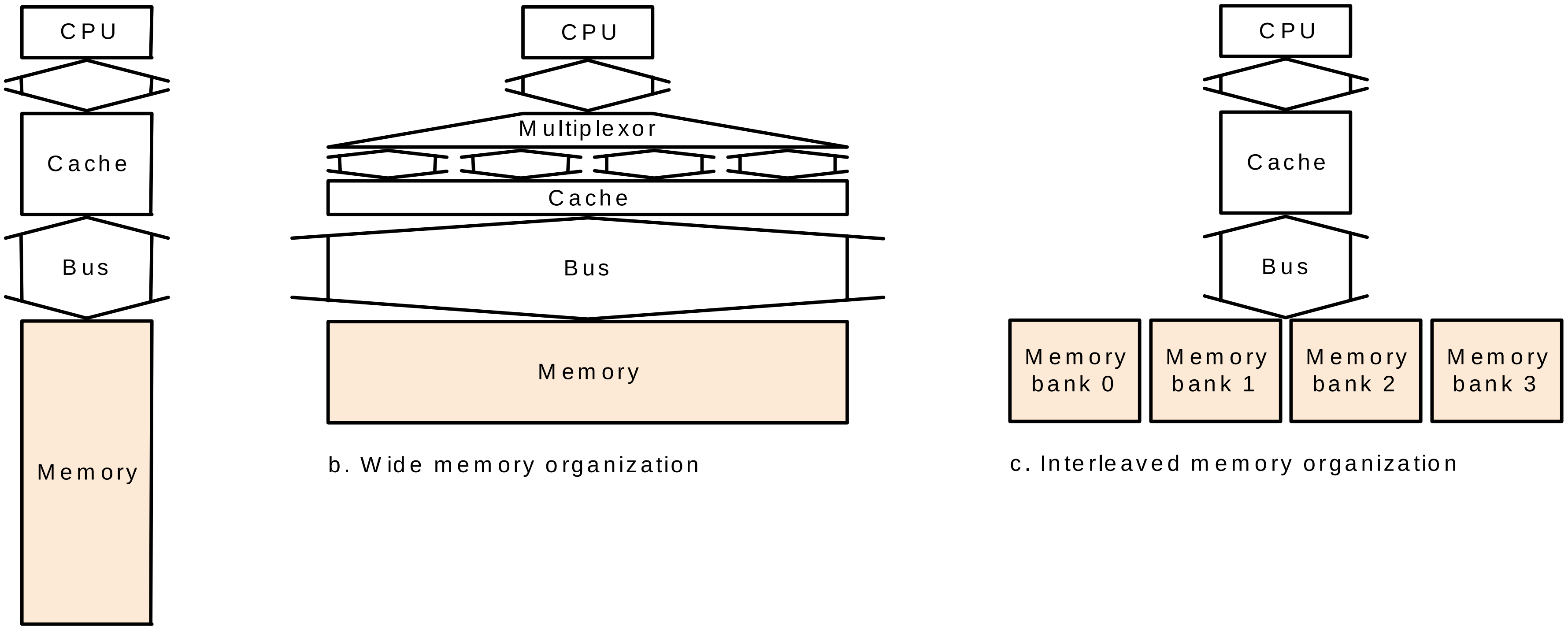
- **Increasing the block size tends to decrease miss rate:**



Designing the Memory System

Figure 7.13

- **Make reading multiple words easier by using banks of memory**



a. One-word-wide memory organization

b. Wide memory organization

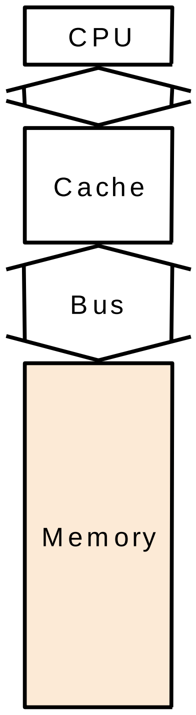
c. Interleaved memory organization

1-word-wide memory organization

Figure 7.13

Suppose we have a system as follows

- 1-word-wide memory organization
- 1 cycle to send the address
- 15 cycles to access DRAM
- 1 cycle to send a word of data



a. One-word-wide memory organization

If we have a cache block of 4 words

Then the miss penalty is

=(1 address send) + 4×(15 DRAM reads)+4×(1 data send)

= 65 clocks per block read

Thus the number of bytes transferred per clock cycle

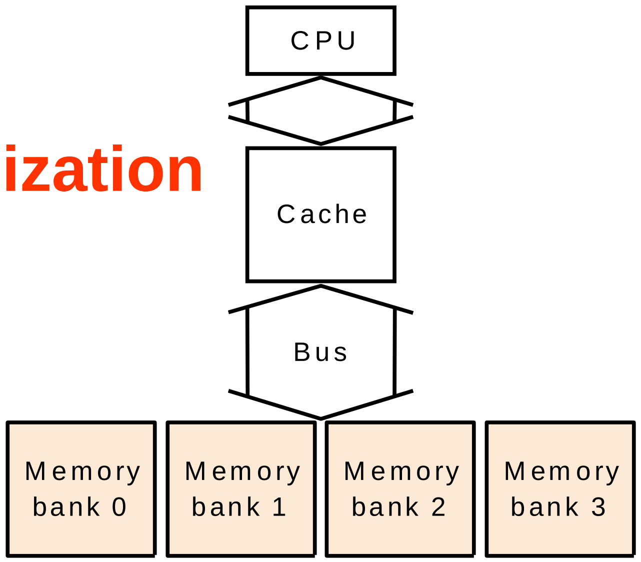
= 4 bytes/word x 4 words/65 clocks = 0.25 bytes/clock

Interleaved memory organization

Figure 7.13

Suppose we have a system as follows

- 4-bank memory interleaving organization
- 1 cycle to send the address
- 15 cycles to access DRAM
- 1 cycle to send a word of data



c. Interleaved memory organization

If we have a cache block of 4 words

Then the miss penalty is
= (1 address send) + 1×(15 DRAM reads)+ 4×(1 data send)
= 20 clocks per block read

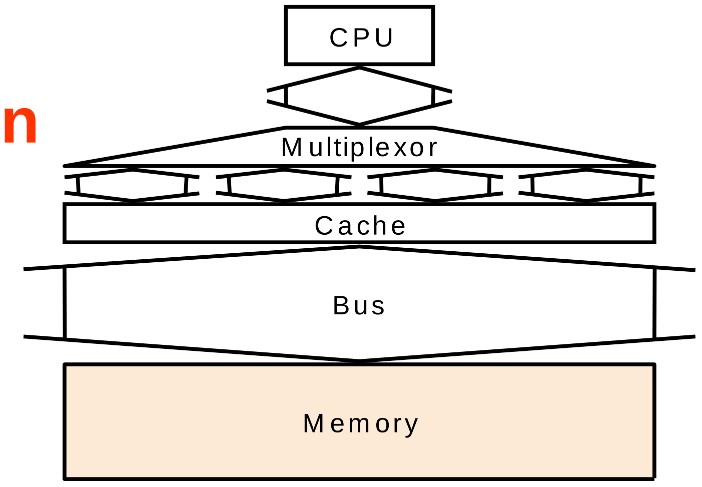
Thus the number of bytes transferred per clock cycle
= 4 bytes/word x 4 words/17 clocks = 0.80 bytes/clock
we improved from 0.25 to 0.80 bytes/clock!

Wide bus: 4-word-wide memory organization

Figure 7.13

Suppose we have a system as follows

- 4-word-wide memory organization
- 1 cycle to send the address
- 15 cycles to access DRAM
- 1 cycle to send a word of data



b. Wide memory organization

If we have a cache block of 4 words

Then the miss penalty is

= (1 address send) + 1×(15 DRAM reads)+ 1×(1 data send)

= 17 clocks per block read

Thus the number of bytes transferred per clock cycle

= 4 bytes/word x 4 words/17 clocks = 0.94 bytes/clock

we improved from 0.25 to 0.80 to 0.94 bytes/clock!

Memory organizations

Figure 7.13

Chip Area Speed

One word wide memory organization

Advantage

Easy to implement, low hardware overhead

Disadvantage

Slow: 0.25 bytes/clock transfer rate

Interleave memory organization

Advantage

Better: 0.80 bytes/clock transfer rate

Banks are valuable on writes: independently

Disadvantage

more complex bus hardware

Wide memory organization

Advantage

Fastest: 0.94 bytes/clock transfer rate

Disadvantage

Wider bus and increase in cache access time