Caches

April 4, 1996

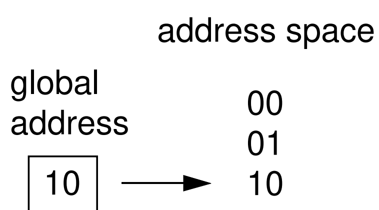
Topics:

- fun with bits
- basic cache organization
- direct mapped, fully associative, and k-way set-associative caches

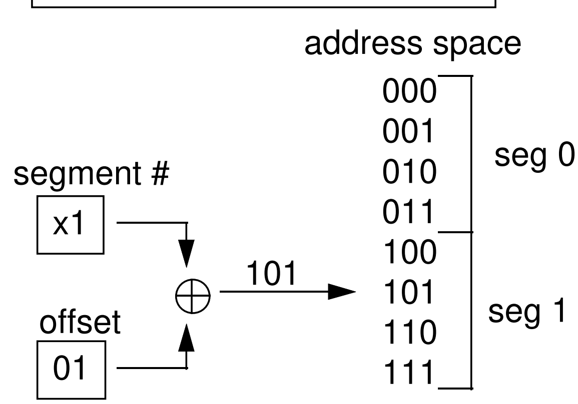
Address spaces

An n-bit address field defines an address space of 2^n items: $0,\ldots,2^{n-1}$.

Linear (flat) address space (e.g. MIPS)



Segmented address space (e.g. Intel x86)



Pro: extends the address space.
Con: ptrs are two words, data size might exceed offset, destroys programmer's will to live.

Partitioning address spaces

Key idea: partitioning the address bits partitions the address space.

In general, an address partitioned into sets of t (tag), s (set index), and b (block offset) bits, e.g.,

address =

t	s	b
tag	set index	offset

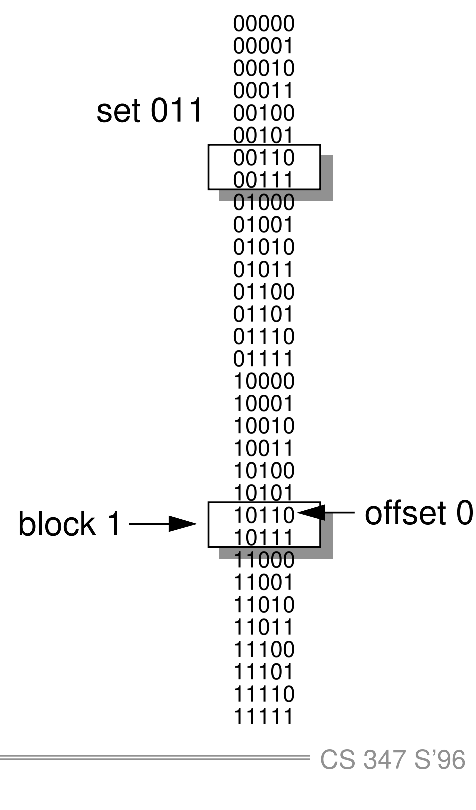
belongs to one of 2^s equivalence classes (sets), where each set consists of 2^t blocks of addresses, and each block consists of 2^b addresses.

The s bits uniquely identify an equivalence class.
The t bits uniquely identify each block in the equivalence class.
The b bits define the offset of an address within a block (block offset).
The t bits uniquely identify an address at a particular block offset.

Partitioning address spaces

t=1	s=3	b=1
1	011	0

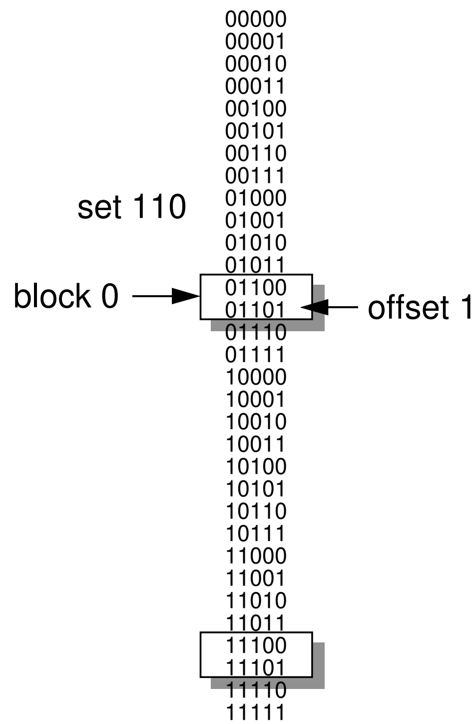
2^s = 8 sets of blocks
2^t = 2 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=1	s=3	b=1
0	110	1

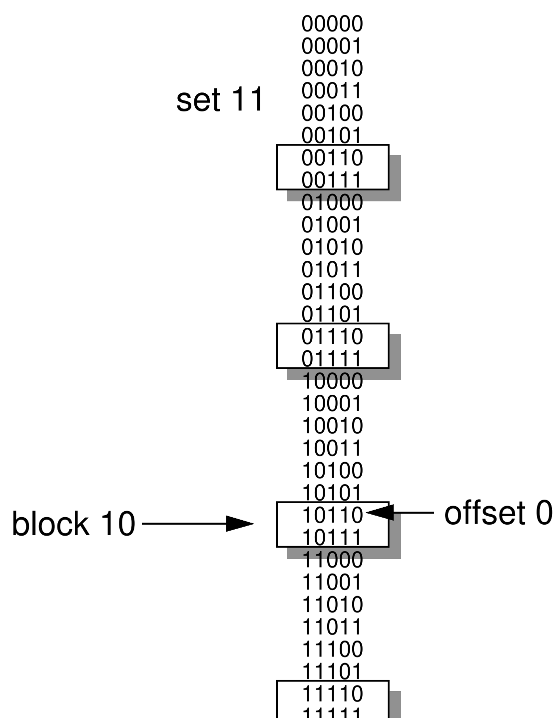
2^s = 8 sets of blocks
2^t = 2 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=2	s=2	b=1
10	11	0

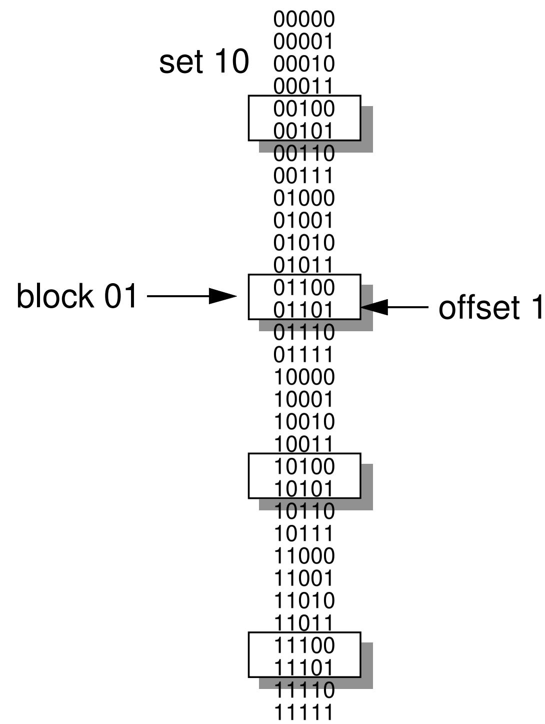
2^s = 4 sets of blocks
2^t = 4 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=2	s=2	b=1
01	10	1

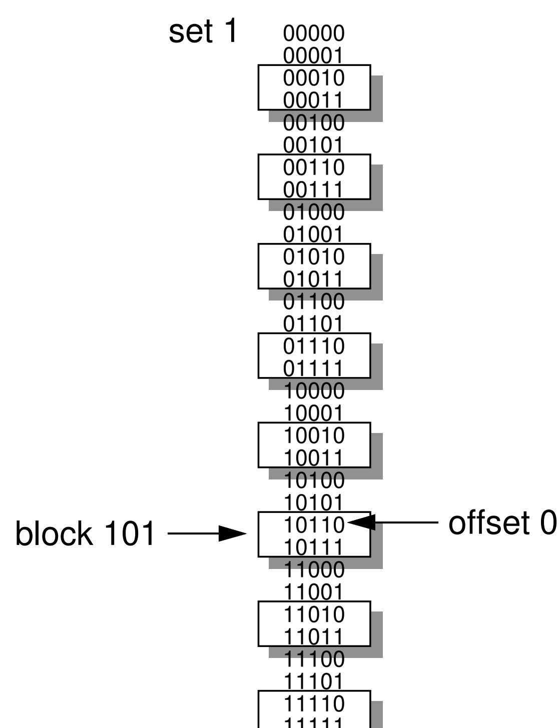
2^s = 4 sets of blocks
2^t = 4 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=3	s=1	b=1
101	1	0

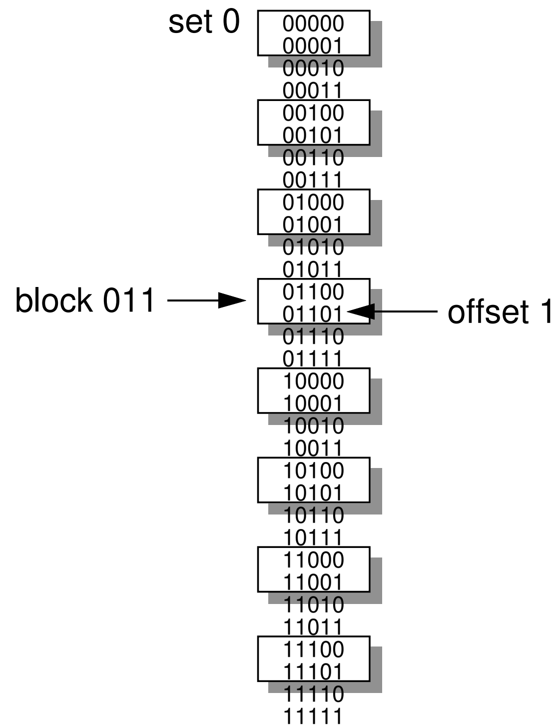
2^s = 2 sets of blocks
2^t = 8 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=3	s=1	b=1
011	0	1

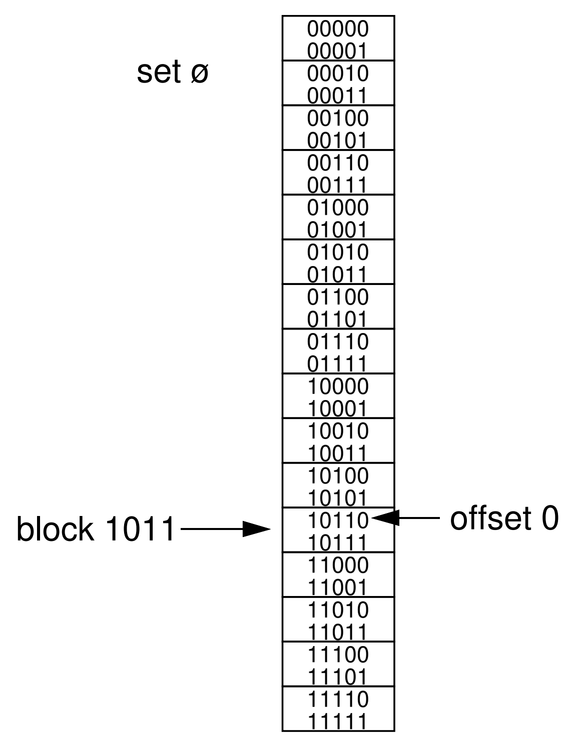
2^s = 2 sets of blocks
2^t = 8 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=4	s=0	b=1
1011		0

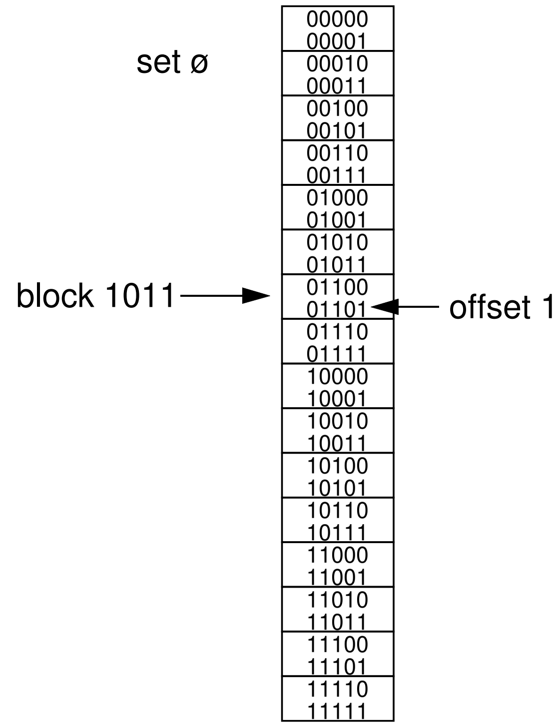
2^s = 1 set of blocks
2^t = 16 blocks/set
2^b = 2 addresses/block.



Partitioning address spaces

t=4	s=0	b=1
0110		1

2^s = 1 set of blocks
2^t = 16 blocks/set
2^b = 2 addresses/block.



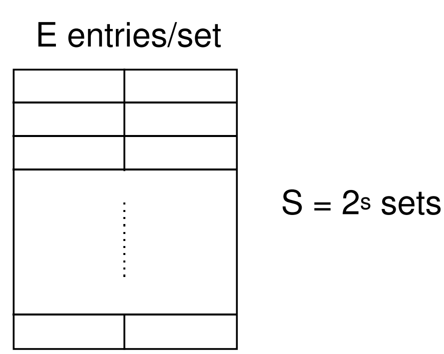
Basic cache organization

Address space (N = 2^n bytes)

t	s	b

t + s + b = n

Cache (C = S x E x B bytes)



Cache entry

1 bit	t bits	B = 2^b bytes (line size)
V	tag	data block (line)

Direct mapped cache (E = 1)

N = 16 addresses (n=4)

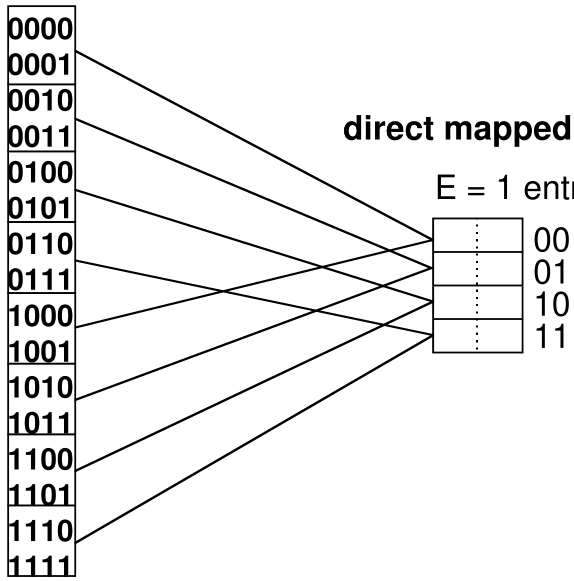
t=1	s=2	b=1
x	xx	x

cache size:
C = 8 data bytes

line size:
$B = 2^b = 2$ bytes/line
(assume a 2 byte word)

direct mapped cache

E = 1 entry/set



$S = 2^s = 4$ sets

Direct mapped cache simulation

N=16 addresses B=2 bytes/line S=4 sets E=1 entry/set

Address trace (reads): 0(miss) 1(hit) 13(miss) 8(miss) 0(miss)

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

(1) 0 (miss)

v	tag	data
1	0	m[0-1]
0		
0		
0		

(2) 13 (miss)

v	tag	data
1	0	m[0-1]
0		
1	1	m[12-13]
0		

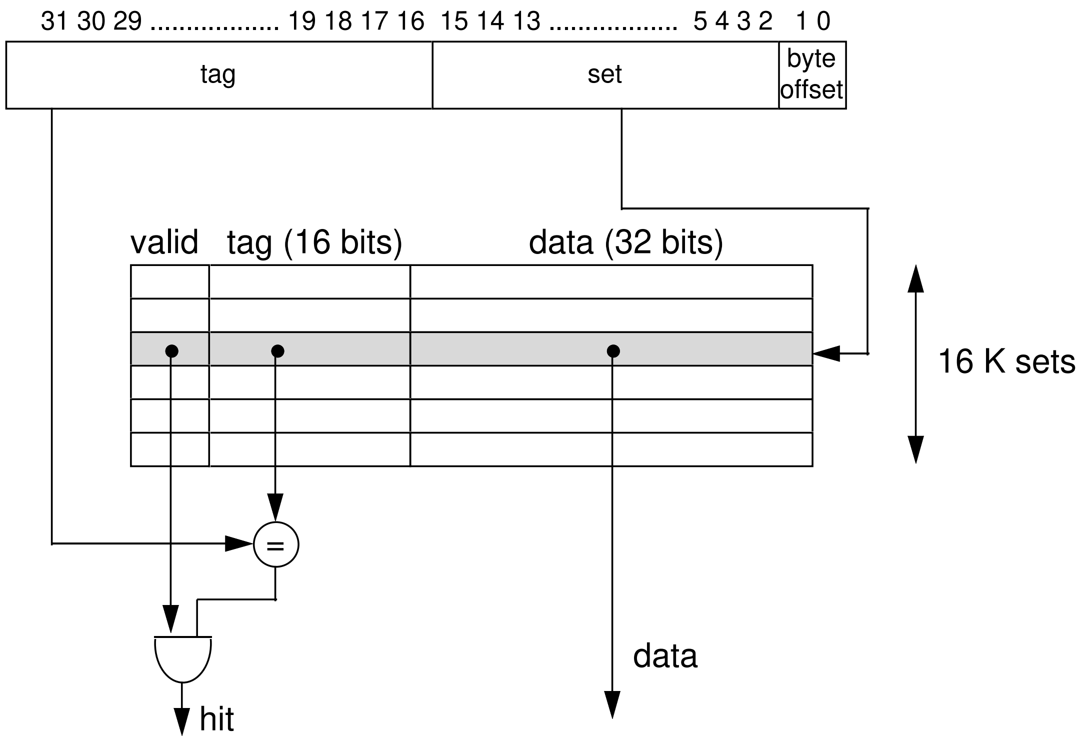
(3) 8 (miss)

v	tag	data
1	1	m[8-9]
0		
1	1	m[12-13]
0		

(4) 0 (miss)

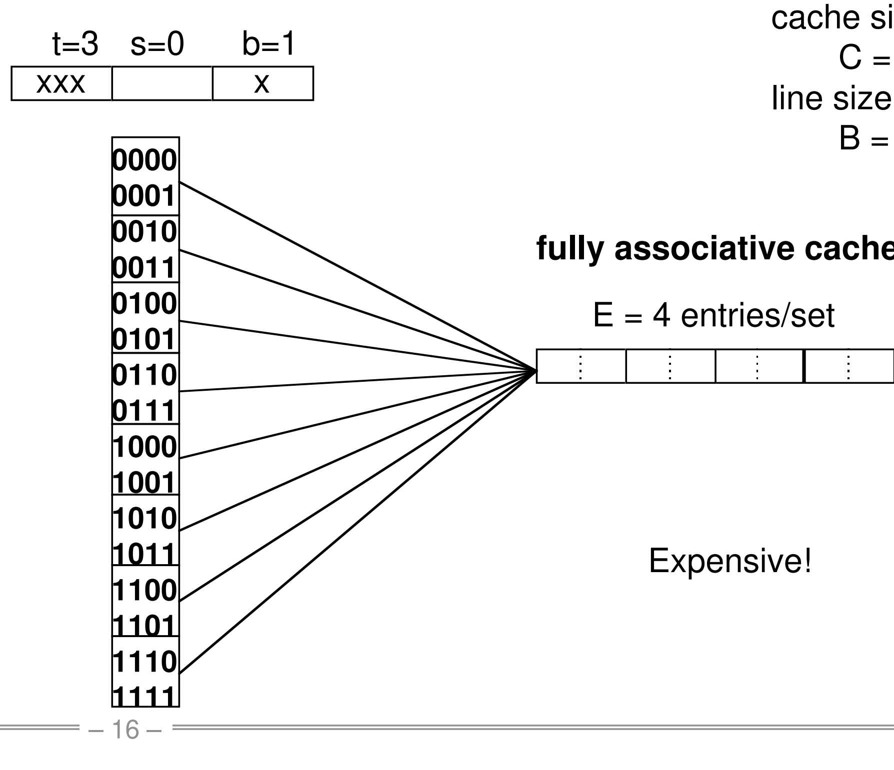
v	tag	data
1	0	m[0-1]
0		
1	1	m[12-13]
0		

Direct mapped cache (DECStation 3100)



Fully associative cache (E = C/B)

N = 16 addresses (n=4)



cache size:
$C = 8$ data bytes
line size:
$B = 2^b = 2$ bytes/line

fully associative cache

$E = 4$ entries/set

$S = 2^s = 1$ set

Expensive!

Fully associative cache simulation

N=16 addresses B=2 bytes/line S=1 sets E=4 entries/set

Address trace (reads): 0(miss) 1(hit) 13(miss) 8(miss) 0(hit)

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

(1) 0 (miss)

v	tag	data
1	000	m[0-1]
0		
0		
0		

set ø

(2) 13 (miss)

v	tag	data
1	000	m[0-1]
1	110	m[12-13]
0		
0		

(3) 8 (miss)

v	tag	data
1	000	m[0-1]
1	110	m[12-13]
1	100	m[8-9]
0		

E-way set-associative cache

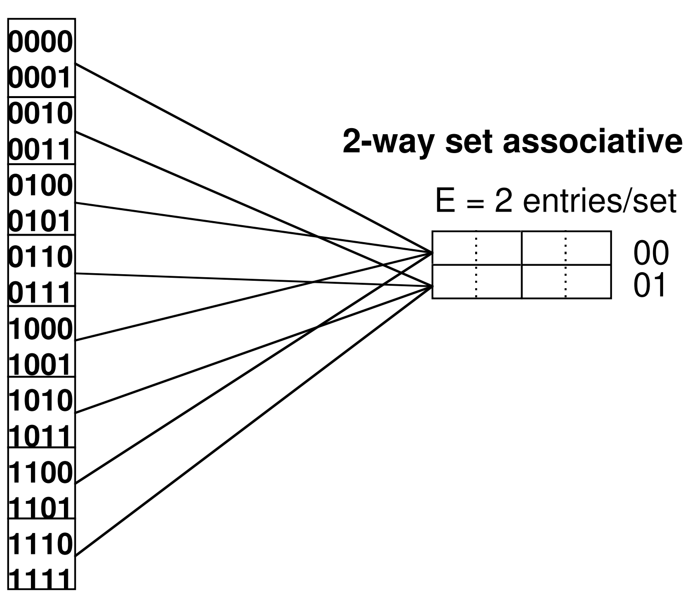
N = 16 addresses (n=4)

t=2	s=1	b=1
xx	x	x

cache size:
C = 8 data bytes
line size:
$B = 2^b = 2$ bytes/line

2-way set associative cache

E = 2 entries/set



$S = 2^1 = 2$ sets

2-way set associative simulation

N=16 addresses B=2 bytes/line S=2 sets E=2 entries/set

Address trace (reads): 0(miss) 1(hit) 13(miss) 8(miss) 0(miss)

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

(1) 0 (miss)

v	tag	data	
1	00	m[0-1]	set 0
0			
0			set 1
0			

(2) 13 (miss)

v	tag	data
1	00	m[0-1]
1	11	m[12-13]
0		
0		

(3) 8 (miss)

v	tag	data
1	10	m[8-9]
1	11	m[12-13]
0		
0		

(LRU replacement)

(3) 0 (miss)

v	tag	data
1	10	m[8-9]
1	00	m[0-1]
0		
0		

(LRU replacement)