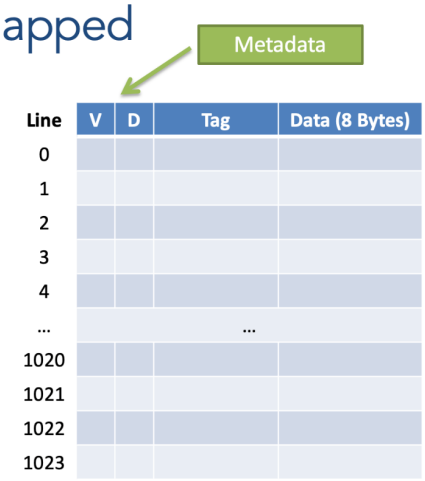
CS31 Worksheet: Week 9: Caching

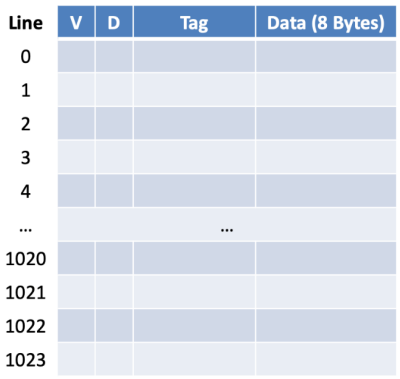
Direct-Mapped

1024 cache locations (every block mapped to one)
Block size of 8 bytes



Address division: Direct-Mapped

- Identify byte in block
  - How many bits do we need to represent each byte uniquely?
- Identify which row (line)
  - How many bits do we need to represent each line uniquely?



A. Block 8 bits Row 1024 bits
B. Block 3 bits Row 10 bits
C. Block 10 bits Row 10 bits
D. Block 32 bits Row 32 bits

Direct-Mapped Example

- Let's say we access memory at address:
  - 01101011 0011 0100
- Step 2:
  - Use index to find line (row)
  - 0011 -> 3

Line	V	D	Tag	Data (16 Bytes)
0				
1				
2				
3				
4				
5				
...				
15				

Suppose we had 8-bit addresses, a cache with 8 lines, and a block size of 4 bytes.

- How many bits would we use for:
  - Tag?
  - Index?
  - Offset?

How would the cache change if we performed the following memory operations?

Memory address

Read 01000100 (Value: 5)
Read 11100010 (Value: 17)
Write 01110000 (Value: 7)
Read 10101010 (Value: 12)
Write 01101100 (Value: 2)

Line	V	D	Tag	Data (4 Bytes)
0	1	0	111	17
1	1	0	011	9
2	0	0	101	15
3	1	1	001	8
4	1	0	011	4
5	0	0	111	6
6	0	0	101	32
7	1	0	110	3

How would the cache change if we performed the following memory operations? (2-way set)

Read 01000100 (Value: 5)
Read 11100010 (Value: 17)
Write 01100100 (Value: 7)
Read 01000110 (Value: 5)
Write 01100000 (Value: 2)

LRU = 0 means: the left line in the set was least recently used.
LRU = 1 means: the right line in the set was least recently used.

Set #	LRU	V	D	Tag	Data (4 Bytes)	V	D	Tag	Data (4 Bytes)
0	1	0	0	111	4	1	0	001	17
1	0	1	1	111	9	1	0	010	5
2				...	...			...	...
3									
4									
5									
6									
7									