LAB 4

Cache Geometry
Objective

• Find the number of bytes per cache block
• Using cache block size, find cache capacity
• Using only cache capacity, find associativity
Tools

• `void cache_init`
  • Creates the virtual cache you’re examining
• `bool_t cache_access(int_t address)`
  • True if address is a hit, otherwise false
• `void flush_cache`
  • Empties the cache
  • Next access is a guaranteed miss
• `Control structures (eg. if, while)`
Review from Lecture

• (10, miss), (11, hit), (12, miss)
  • 10 = 0b1010, 11 = 0b1011, 12 = 0b1100

• Direct mapped?
  • Block size = 2 or 4 bytes

• Two-way associative?
  • Block size = Still 2 or 4 bytes
  • No replacement occurred
Review From Lecture

• (10, miss); (12, miss); (10, miss)
  • Associativity?
    • Must be direct mapped
Practice

- 0x000, miss
- 0x001, hit
- 0x010, hit
- 0x011, hit
- 0x100, miss

What can we assume?
  - Block size is 4; can’t assume anything else
Practice

- 0x01101, miss
- 0x01110, miss
- 0x01111, hit
- 0x01100, hit

What can we assume?
  - 2 byte block size, direct mapped
Practice

- Given: block size = 8 (3 offset bits), 2 index bits, 2 tag bits
- 0x1101101, miss
- 0x0001101, miss
- 0x0111101, miss
- 0x1101100, hit

What can we assume?
- Associativity >= 2

How could we determine if associativity is 2 or 4?
Practice

- Given cache size = 32 bytes
- How many tag bits are there if we use 16 bit addresses?
  - $2^5 = 32$, $16 - 5 = 11$
- What are the other bits used for?
  - Either index or offset
- Think in powers of two