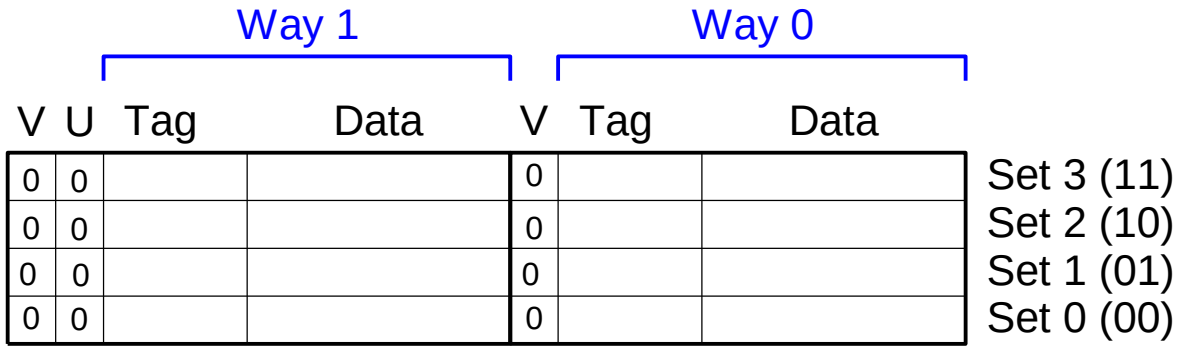
Cache Replacement Policy

Replacement Policy

- Cache is too small to hold all data of interest at once
- If cache full: program accesses data X and evicts data Y
- ***Capacity miss*** when access Y again
- How to choose Y to minimize chance of needing it again?
  - **Least recently used (LRU) replacement:** the least recently used block in a set evicted

LRU Replacement

```
# RISC-V assembly
lw s1, 0x04(zero)
lw s2, 0x24(zero)
lw s3, 0x54(zero)
```



Pseudo LRU Replacement

- For set associativity of higher then two ways
  - Divide the ways into two groups
  - U indicates which group of ways was least recently used
  - The new block replaces a random block within the least recently used group

Write Policy

Memory stores

- If the store word operation results in a miss, the cache block is fetched from main memory into the cache, and then the appropriate word in the cache block is written
- If the store word operation results in a hits, the word is simply written to the cache block.

Write through Vs Write back

- In a write through cache, the data written to a cache block is simultaneously written to main memory.
- In a write-back cache, a Dirty bit (*D*) is associated with each cache block.
  - *D* is 1 when the cache block has been written and 0 otherwise
  - Dirty cache blocks are written back to main memory only when they are evicted from the cache.
- Modern caches are usually write-back because main memory access time is so large.

Example

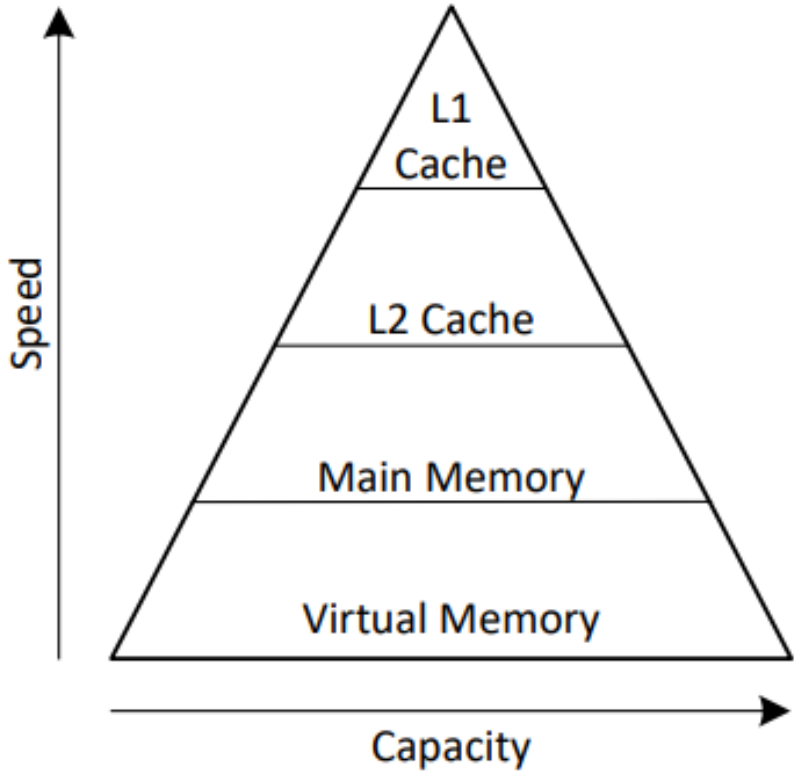
- Assuming a block size of four words, discuss main memory accesses for a write-through versus a write-back policy

```
addi t5, zero, 0
sw t1, 0(t5)
sw  t2, 12(t5)
sw  t3, 8(t5)
sw t4, 4 (t5)
```

- Trade-off: extra Dirty bit for every block.

Multiple level Caches

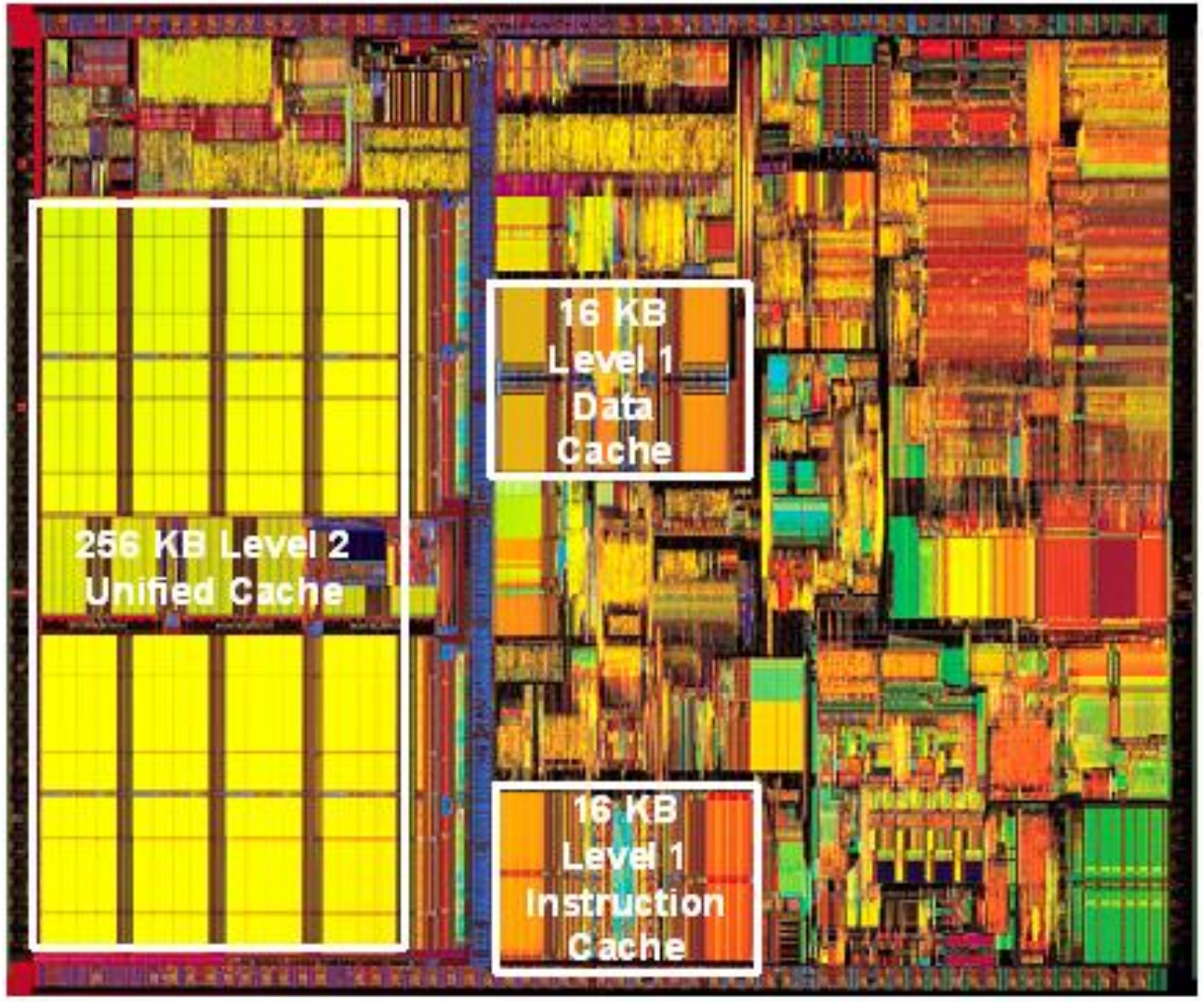
Multilevel Caches



Multilevel Caches

- Larger caches have lower miss rates, longer access times
- Expand memory hierarchy to multiple levels of caches
- Level 1: small and fast (e.g. 16 KB, 1 cycle)
- Level 2: larger and slower (e.g. 256 KB, 2-6 cycles)
- Most modern PCs have L1, L2, and L3 cache

Intel Pentium III Die



© Intel Corp.

Example

- Access time:
  - L1: 1 cycyle
  - L2: 10 cycle
  - Virtual memory: 100 cycle
- Miss rate:
  - L1: 5%
  - L2: 20%
- AMAT=1cycle+0.05(10 cycles +0.2(100 cycles))=2.5 cycles