RISC Design
Memory System

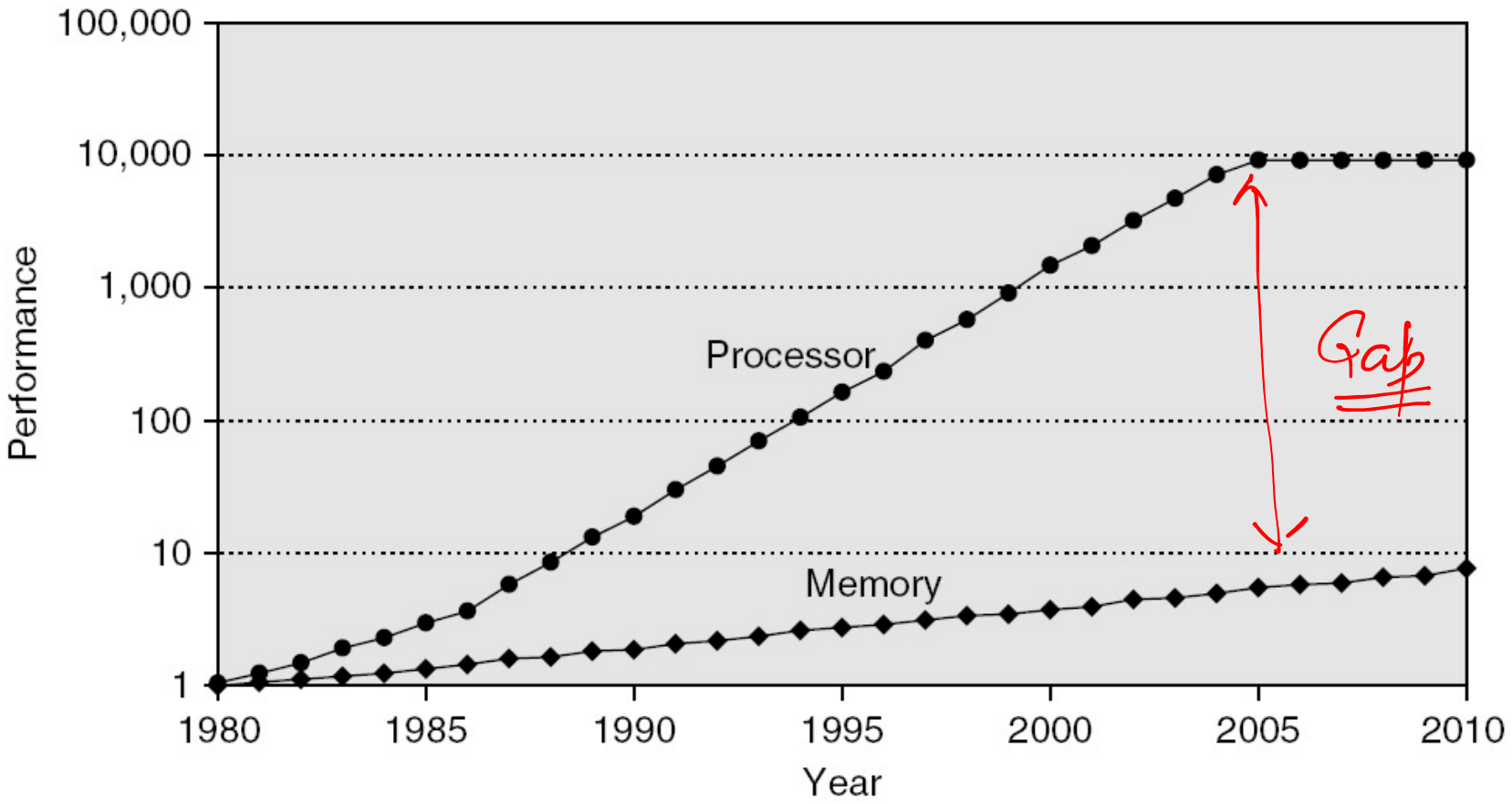
Virendra Singh
Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

EE-739: Processor Design

Lecture 25 (18 March 2021)

CADSL

Memory Performance Gap



Why Memory Hierarchy?

- Need lots of bandwidth

$$BW = \frac{1.0inst}{cycle} \times \left[\frac{1Ifetch}{inst} \times \frac{4B}{Ifetch} + \frac{0.2Dref}{inst} \times \frac{4B}{Dref} \right] \times \frac{1Gcycles}{\text{sec}}$$

$$= \frac{4.8GB}{\text{sec}}$$

- Need lots of storage
  - 64MB (minimum) to multiple TB
- Must be cheap per bit
  - (TB x anything) is a lot of money!
- These requirements seem incompatible

Memory Hierarchy Design

- Memory hierarchy design becomes more crucial with recent multi-core processors:
  - Aggregate peak bandwidth grows with # cores:
    - Intel Core i7 can generate two references per core per clock
    - Four cores and 3.2 GHz clock
      - 25.6 billion 64-bit data references/second +
      - 12.8 billion 128-bit instruction references
      - = 409.6 GB/s!
    - DRAM bandwidth is only 6% of this (25 GB/s)
    - Requires:
      - Multi-port, pipelined caches
      - Two levels of cache per core
      - Shared third-level cache on chip

⇒ Why Locality?

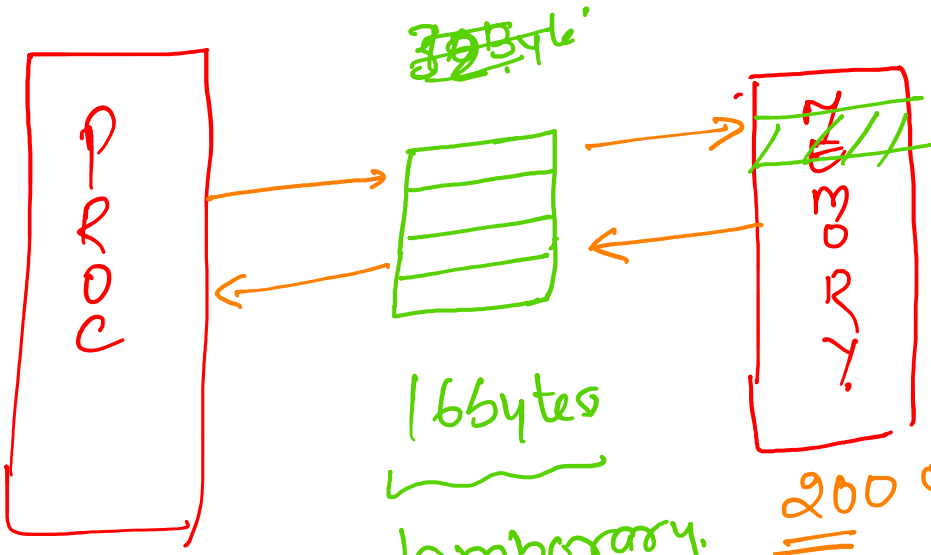
{ temporal locality
spatial locality

- Analogy:
  - Library (Disk)
  - Bookshelf (Main memory)
  - Stack of books on desk (off-chip cache)
  - Opened book on desk (on-chip cache)
- Likelihood of:
  - Referring to same book or chapter again?
    - Probability decays over time
    - Book moves to bottom of stack, then bookshelf, then library
  - Referring to chapter n+1 if looking at chapter n?

Why Does a Hierarchy Work?

- Locality of reference
  - Temporal locality
    - Reference same memory location repeatedly
  - Spatial locality
    - Reference near neighbors around the same time
- Empirically observed
  - Significant!
  - Even small local storage (8KB) often satisfies >90% of references to multi-MB data set

Instruction



chunk

Bring a chunk from memory & place it in temporary

166 bytes temporary

200 cycle

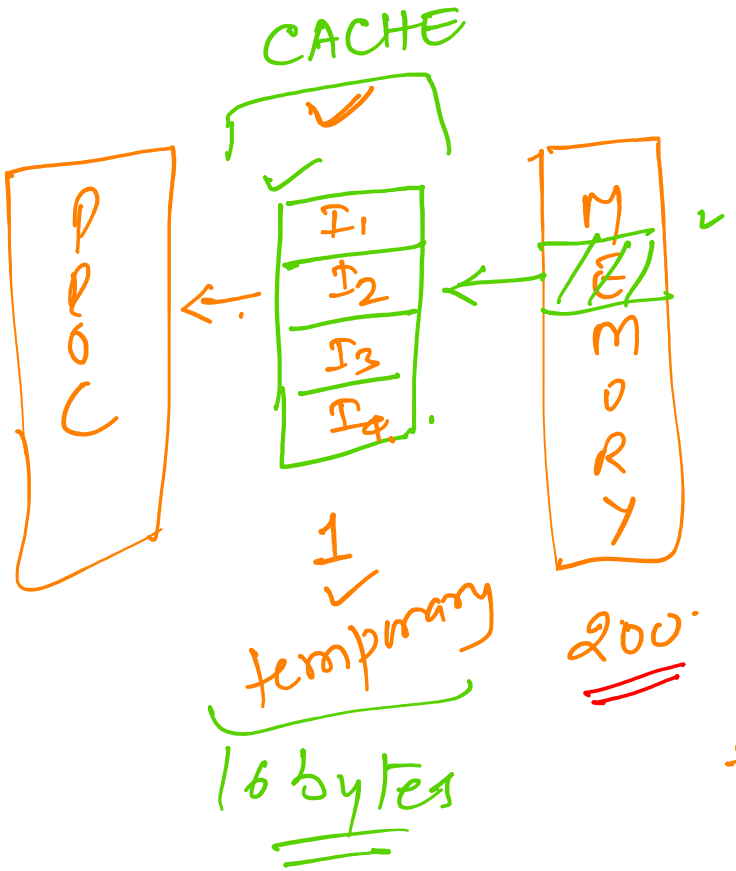
```
for (i=0; i<100; i++)
{
I1  a = b+c
I2  d = a+e
I3  p = q+r
I4  s = p+t
}
```

F-D-E-MA-WB

200+10+200+10+200+10+200+10

I1 I2 I3 I4

time = (840) X100 cycles.




```
for(i=0; i<100; i++)
{ a = b+c
  d = a+e
  p = f+r
  g = p+t
}
```

Spatial locality

200 + 10 + 1 + 10 + 1 + 10 + 1 + 10

I_1 + 1 + 10 + 1 + 10 + 1 + 10 + 1 + 10 (temporal)

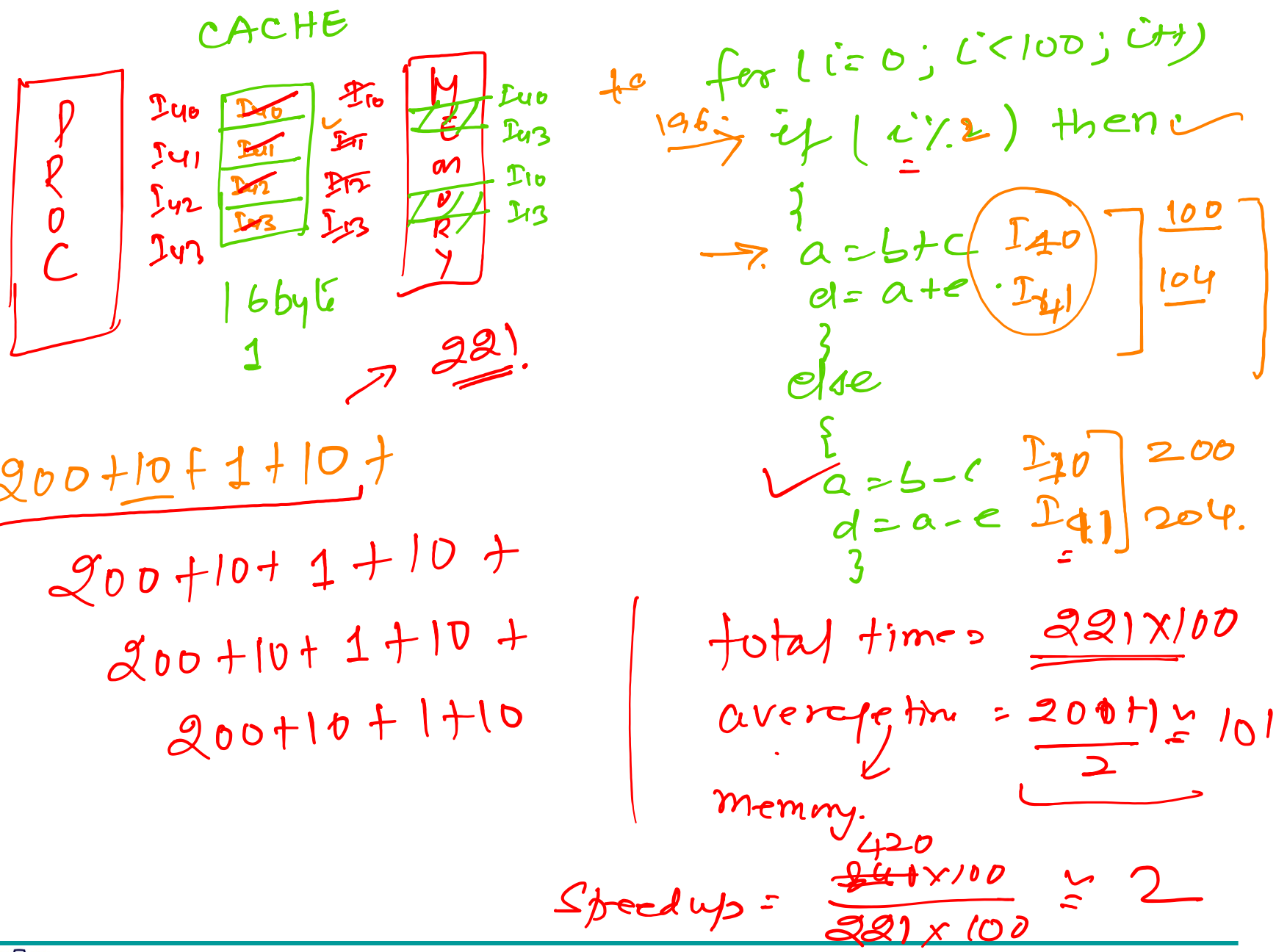
243 + 44 × 99

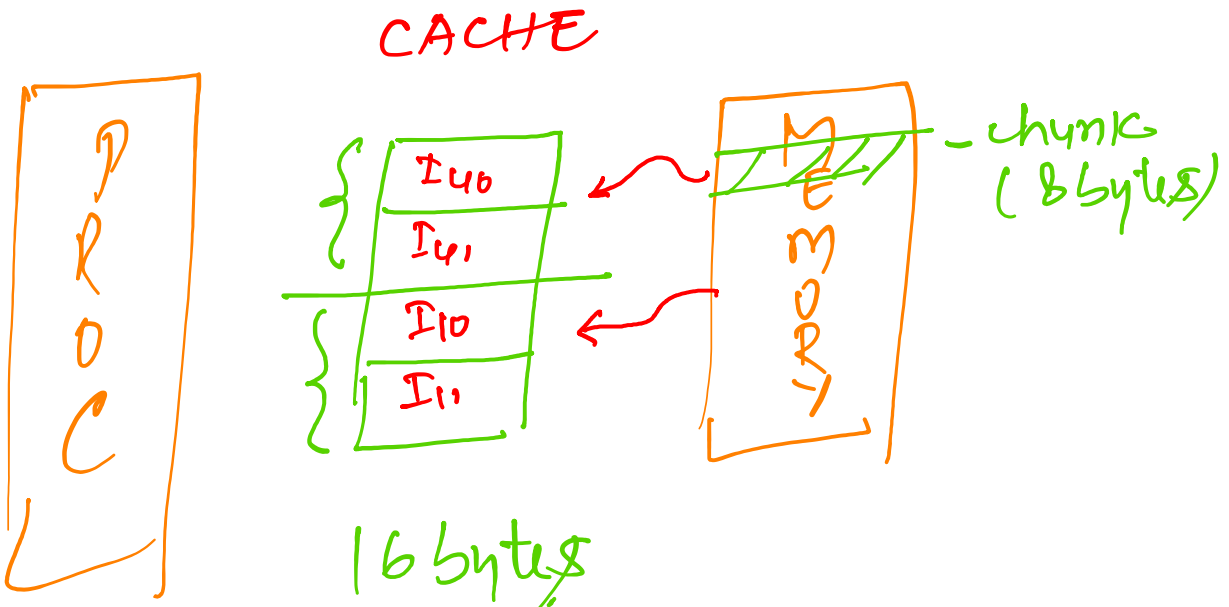
Speedup: $\frac{840 \times 100}{243 + 44 \times 99}$

Average time: fetch instr (without temp memory) = 203

$= \frac{200 + 1 + 1 + 1 + 4 \times 99}{400}$ ⇐ with temp memory

≈ ~~1.203~~ + 1.49 ≈ 1.5


CACHE
PROC
I40
I41
I42
I43
I10
I11
I12
I13
MEMORY
I40
I43
I10
I13
16bytes
1
221
for (i=0; i<100; i++)
196:
if (i%2)
{
a=b+c
d=a+e
I40
I41
100
104
}
else
{
a=b-c
d=a-e
I10
I11
200
204.
}
200+10+1+10+
200+10+1+10+
200+10+1+10
total time = 221x100
average time = (200+1)/2 = 101
memory.
Speedup = 420x100 / 221x100 ≈ 2




```
for (i=0; i<100; i++)
{
if (i%2) then
  a = b+c - I40
  d = a+e - I41
else
  a = b-c - I10
  d = a-e - I11
}
```

16 bytes

1

2 chunks

$\underbrace{200+10+1+10}_{Itr\,1} + \underbrace{200+10+1+10}_{Itr\,2} + \underbrace{1+10+1+10}_{Itr\,3} + \underbrace{1+10+1+10}_{Itr\,4}$

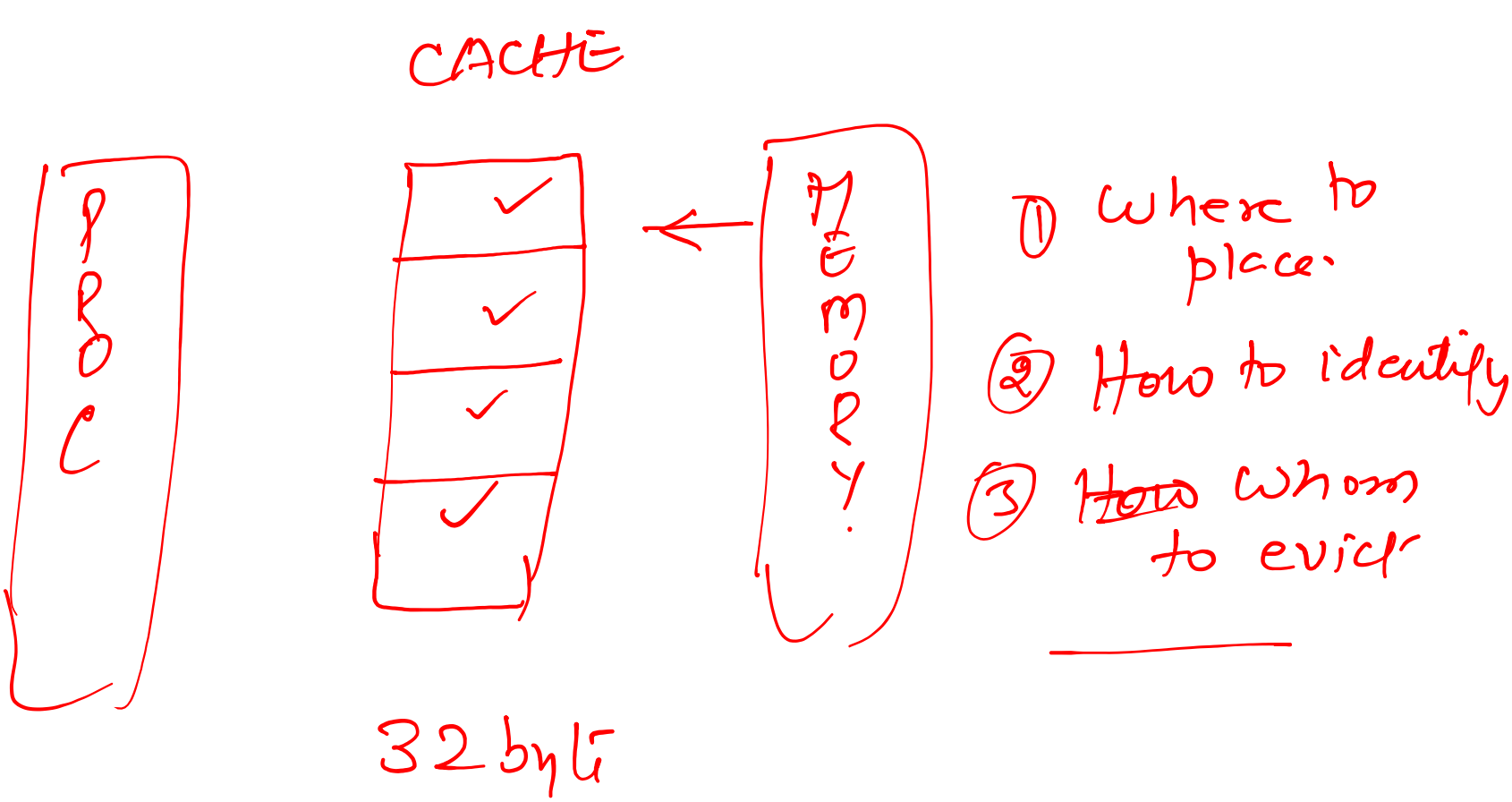
$\Rightarrow$ $221 * 2 = 221 + 22 \times 98 = 442 + 2156 = 2598$.

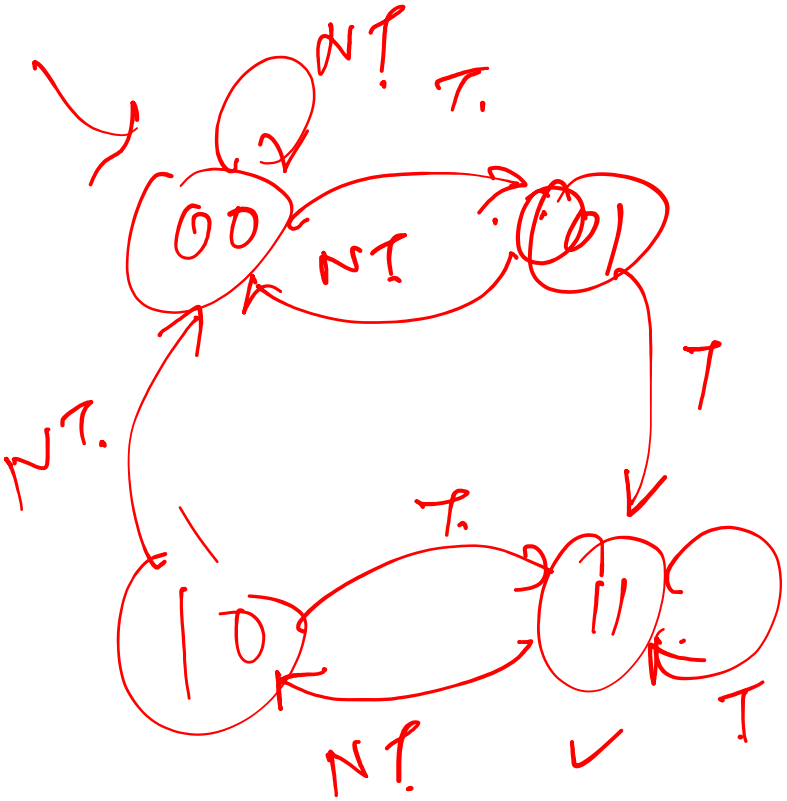
Speedup: $\frac{420 \times 100}{2598} = \frac{420 \times 100}{2600} = \frac{420}{26}$

Average memory access time

$$= \frac{200 + 200 + 1 + (1+1) \times 98}{200}$$

$$=$$


CACHE
PROC
MEMORY
32 byte
① where to place.
② How to identify
③ ~~How~~ whom to evict



HR

NT { 00, 01

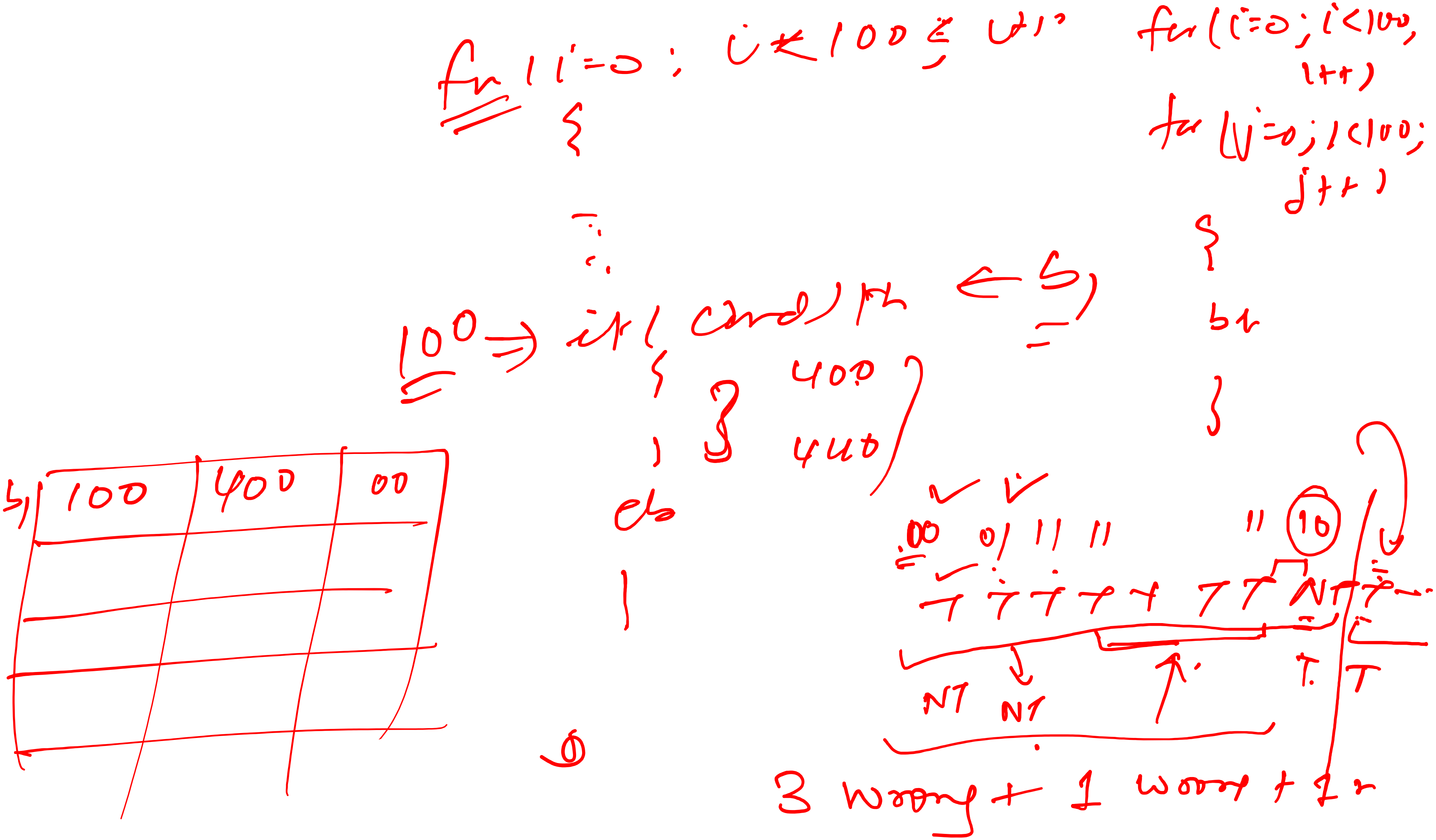
T { 10, 11

Fetch

Predict → fetch < T, NT

Validate → Execution { Pass — don't need to do anything

fail → flush instruction fetched after this (IF & ID stage)


for(i=0; i<100; i++)
for(j=0; j<100; j++)
br
100
400
00
400
440
NT
T
3 wrong + 1 wrong

Four Burning Questions

- These are:
  - Placement
    - Where can a block of memory go?
  - Identification
    - How do I find a block of memory?
  - Replacement
    - How do I make space for new blocks?
  - Write Policy
    - How do I propagate changes?
- Consider these for caches
  - Usually SRAM
- Will consider main memory, disks later

Thank You