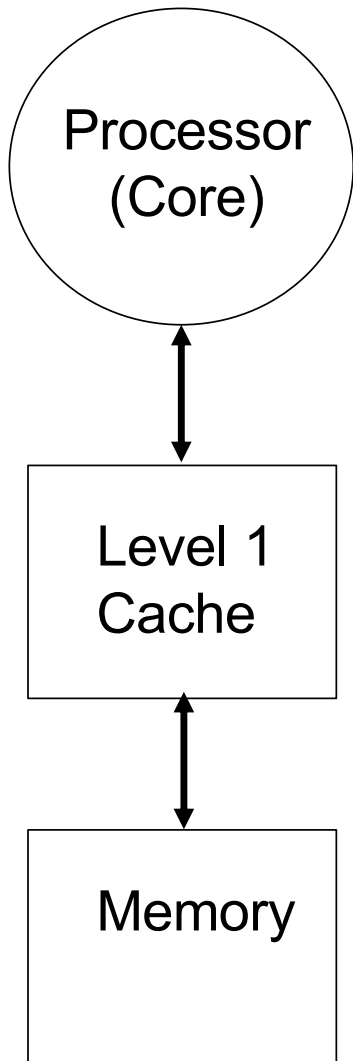


Lecture 5

Memory Hierarchies

- **Cache memory**
 - ✓ Classification of cache misses (4.3.5)
 - ✓ Cache hierarchy performance (4.3.4)
 - ✓ Memory inclusion (4.2.3)
 - ✓ Non-blocking (Lock-up free) caches (4.3.6)
 - ✓ Cache prefetching and preloading (4.3.7)

Memory System Performance



CPI_0

Ideal memory:

$$CPI = CPI_0$$

NOTE: Model assumes that processor stalls on cache misses. Not always true in modern processors

Impact on CPI:

$$CPI = CPI_0 + CPI_{Hit} + CPI_{Miss}$$

CPI_{Hit}

$$CPI_{Miss} = MPI \times MP$$

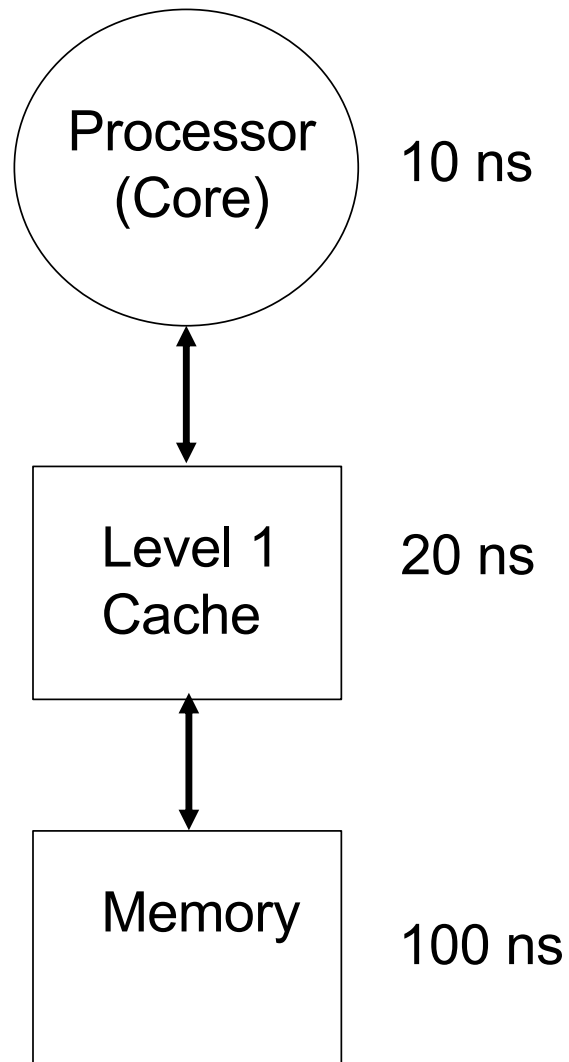
MPI = Miss Per Instruction
MP = Miss Penalty

Impact on execution time:

$$T = IC \times (CPI_0 + CPI_{Hit} + \underbrace{MPI \times MP}_{CPI_{Miss}}) \times TPC$$

CPI_{Miss}

CPI_{Miss}



Question:

What is the average number of cycles per instruction (CPI) assuming that the number of misses per instruction (MPI) is 1%?

Answer:

- $CPI_0 = 1$ where a clock cycle takes 10 ns.
- The level 1 cache is accessed in two cycles so it will add a cycle to each instruction: $CPI_{Hit} = 1$.
- The Miss penalty (MP) is $100/10 = 10$ cycles
- $MPI = 0.01$

$$CPI = CPI_0 + CPI_{Hit} + CPI_{Miss} = 1 + 1 + 0.01 \times 10 = 2.1$$

Cache Miss Classification (Section 4.3.5)

Compulsory Misses

Unavoidable misses even with an infinitely large cache

Question:

Why a fully associative cache and not a direct mapped cache?

Question:

How does a direct mapped cache cause such conflicts?

The 3-C Model

Capacity Misses

Misses caused by a limited capacity fully associative cache

Conflict Misses

Misses caused by limited associativity due to address mapping conflicts

Compulsory
Capacity
Conflict

Block access sequence: 0 1 2 3 4 0 5 1 8 4 9 5

Question:

How many compulsory misses are there in the sequence?

Answer:

The same as the number of unique block addresses:
Eight, marked red below

0 1 2 3 4 0 5 1 8 4 9 5

Compulsory
Capacity
Conflict

Block access sequence: 0 1 2 3 4 0 5 1 8 4 9 5

Question:

How many capacity misses are there in the sequence assuming a four-block cache?

Methodology:

- Assume a fully associative cache
- Replacement policy: Optimal
- Simulate fetches and replacement

Number of Capacity Misses

Compulsory
Capacity
Conflict

OPT Replacement victim: Replace block accessed farthest into the future

Block access sequence: 0 1 2 3 4 0 5 1 8 4 9 5

Victim:

				2		3		0		1	
0		0		0	2	0	4	0	4	8	4
		1		1	3	1	3	1	5	1	5
M	M	M	M	M	H	M	H	M	H	M	H

Access: 0 1 2 3 4 0 5 1 8 4 9 5

Total fully associative: 8

Compulsory misses: 8

Capacity misses = Total fully associative – compulsory = 0

Compulsory
Capacity
Conflict

Block access sequence: 0 1 2 3 4 0 5 1 8 4 9 5

Question:

How many conflict misses are there in the sequence assuming a four-block direct-mapped cache?

Methodology:

- Determine the number of misses
- Number of Conflict Misses = Total Number of Misses - Number of Capacity and Compulsory Misses

Number of Conflict Misses

Compulsory
Capacity
Conflict

Block access sequence: 0 1 2 3 4 0 5 1 8 4 9 5

Victim:

				0	4	1	5	0	8	1	9																																																
<table><tr><td>0</td><td></td></tr><tr><td></td><td></td></tr></table>	0				<table><tr><td>0</td><td></td></tr><tr><td>1</td><td></td></tr></table>	0		1		<table><tr><td>0</td><td>2</td></tr><tr><td>1</td><td></td></tr></table>	0	2	1		<table><tr><td>0</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	0	2	1	3	<table><tr><td>4</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	4	2	1	3	<table><tr><td>0</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	0	2	1	3	<table><tr><td>0</td><td>2</td></tr><tr><td>5</td><td>3</td></tr></table>	0	2	5	3	<table><tr><td>0</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	0	2	1	3	<table><tr><td>8</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	8	2	1	3	<table><tr><td>4</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	4	2	1	3	<table><tr><td>8</td><td>2</td></tr><tr><td>9</td><td>3</td></tr></table>	8	2	9	3	<table><tr><td>8</td><td>2</td></tr><tr><td>5</td><td>3</td></tr></table>	8	2	5	3
0																																																											
0																																																											
1																																																											
0	2																																																										
1																																																											
0	2																																																										
1	3																																																										
4	2																																																										
1	3																																																										
0	2																																																										
1	3																																																										
0	2																																																										
5	3																																																										
0	2																																																										
1	3																																																										
8	2																																																										
1	3																																																										
4	2																																																										
1	3																																																										
8	2																																																										
9	3																																																										
8	2																																																										
5	3																																																										
M	M	M	M	M	M	M	M	M	M	M	M																																																

Access: 0 1 2 3 4 0 5 1 8 4 9 5

Total for direct-mapped: 12

Compulsory misses: 8

Capacity misses = Total for fully associative – compulsory = 0

Conflict misses = Total – compulsory – capacity = 12-8-0 = 4

Effect on Misses of Cache Parameters

Larger Caches

Compulsory Misses?

Capacity Misses?

Conflict Misses?

Larger Blocks

Compulsory Misses?

Capacity Misses?

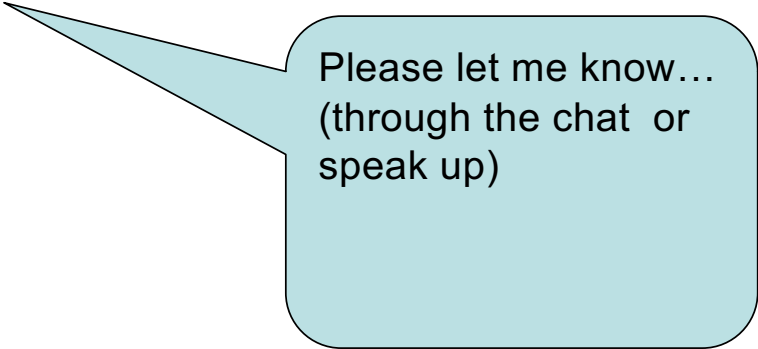
Conflict Misses?

Higher Associativity

Compulsory Misses?

Capacity Misses?

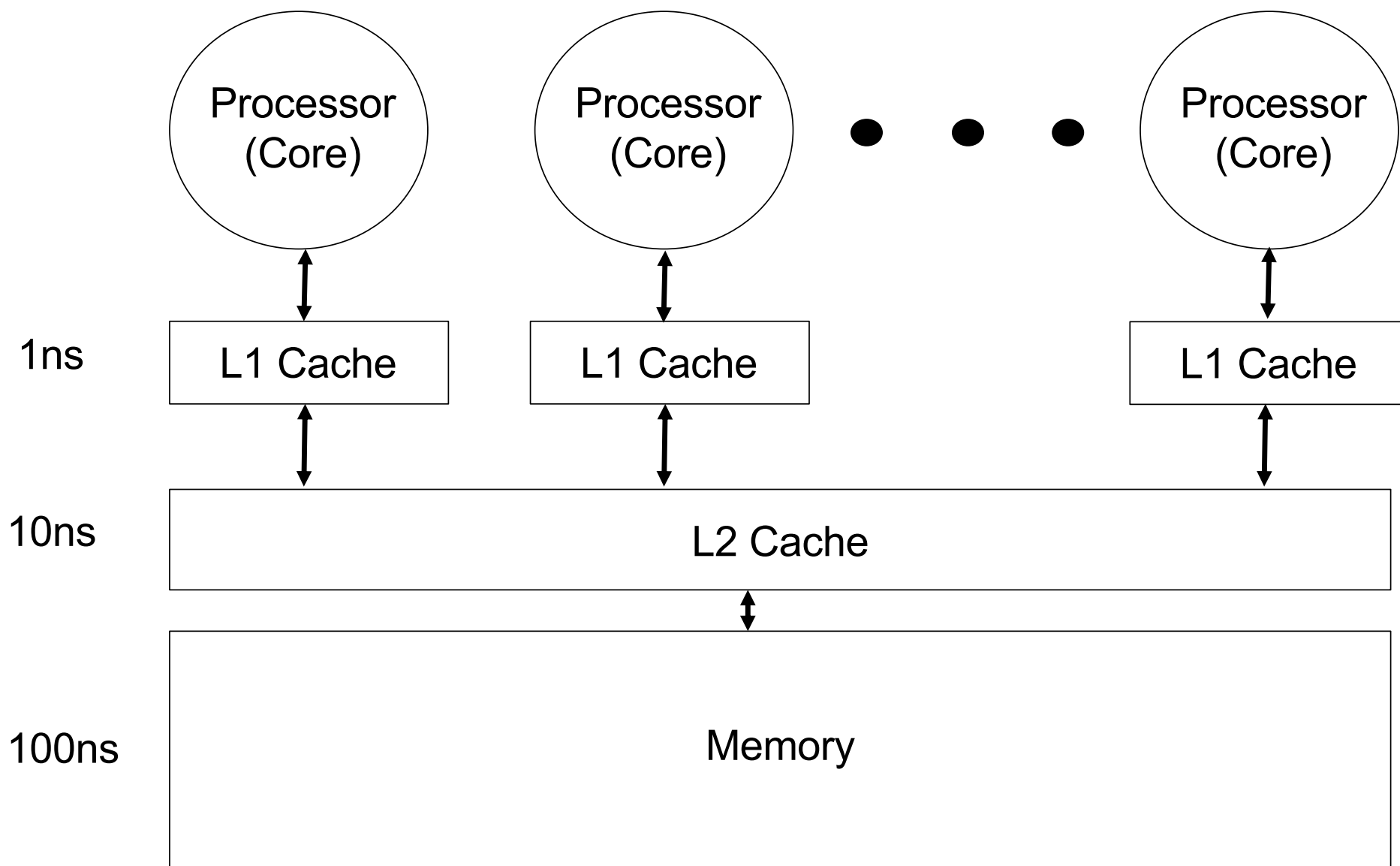
Conflict Misses?

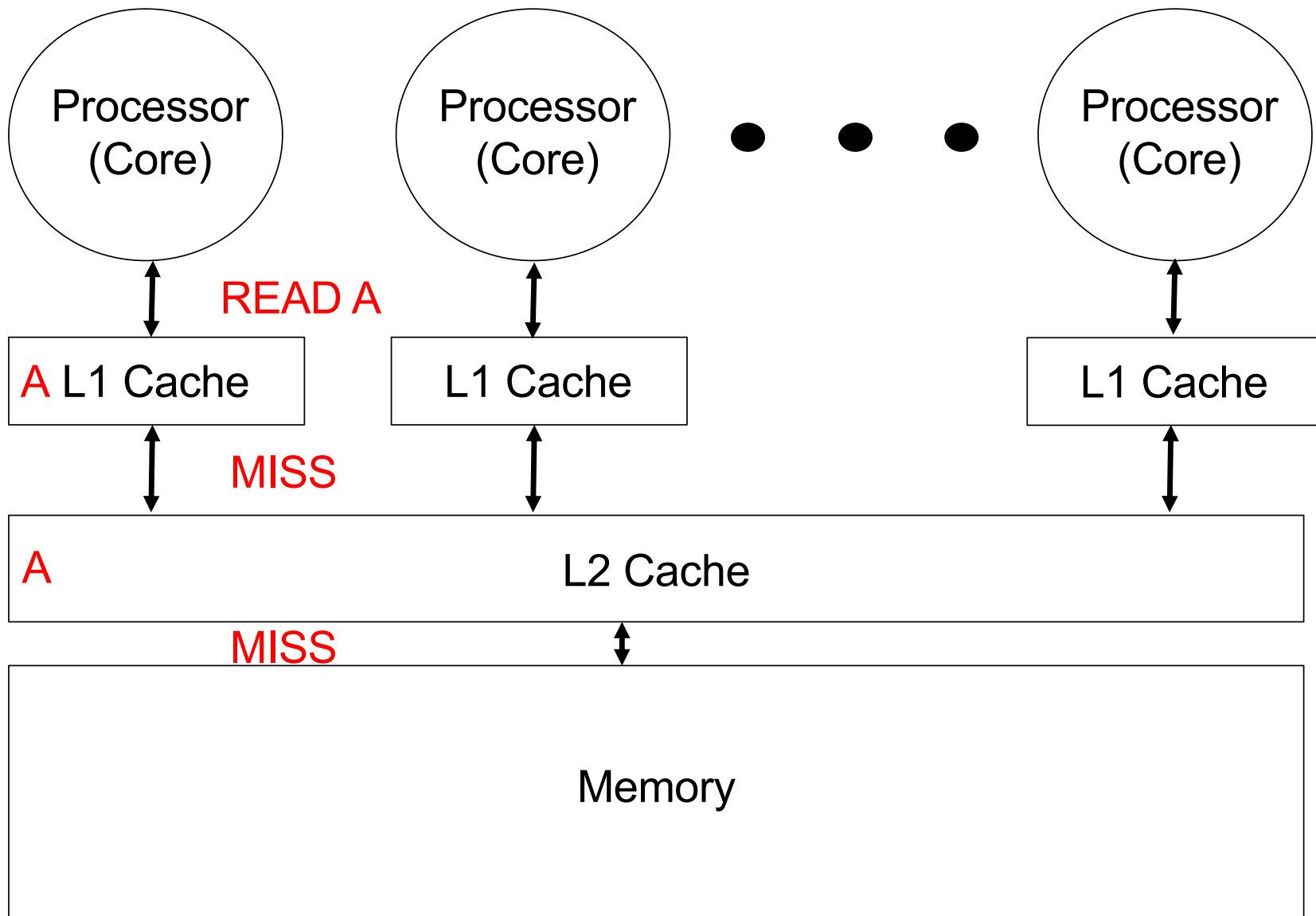


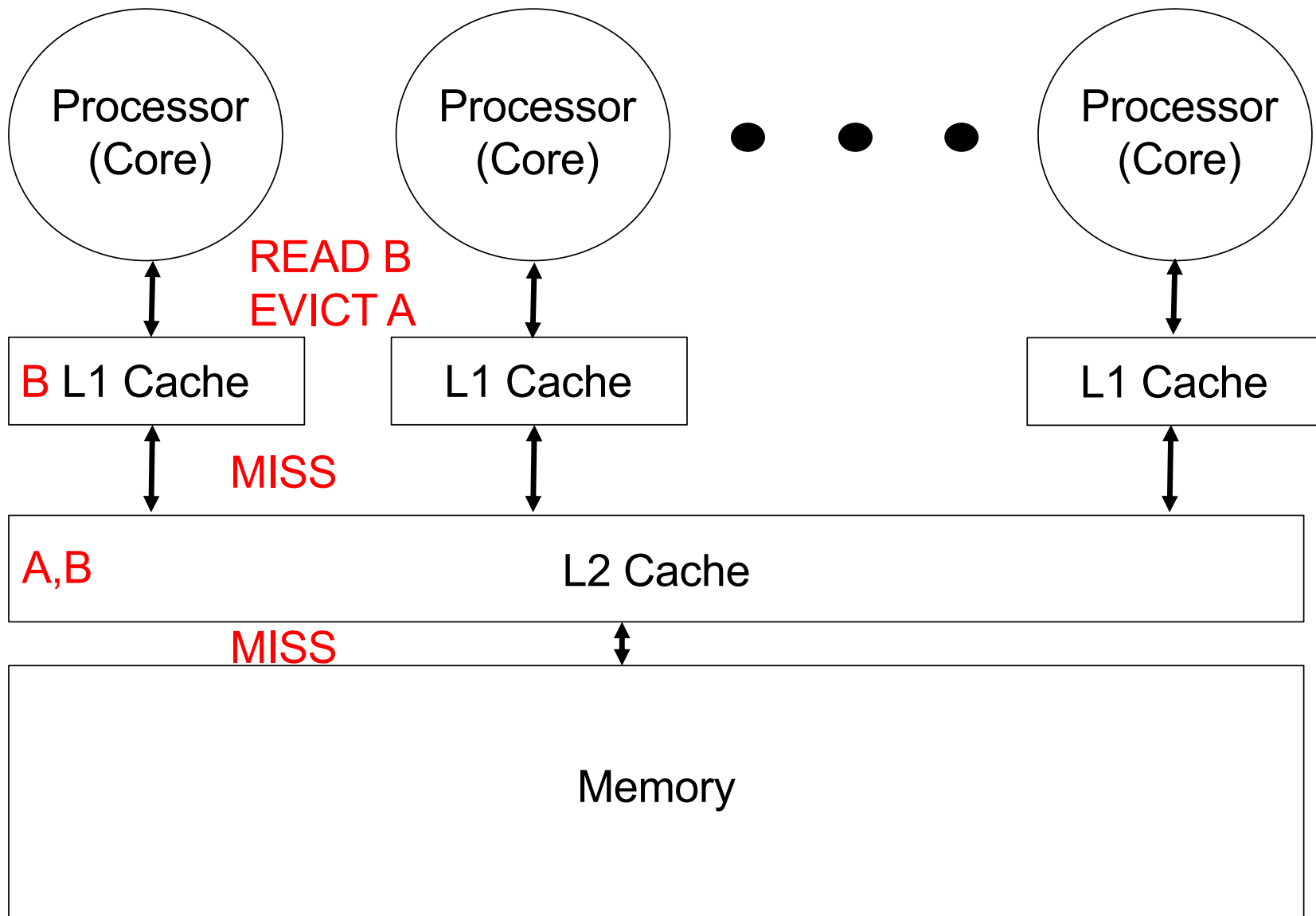
Please let me know...
(through the chat or
speak up)

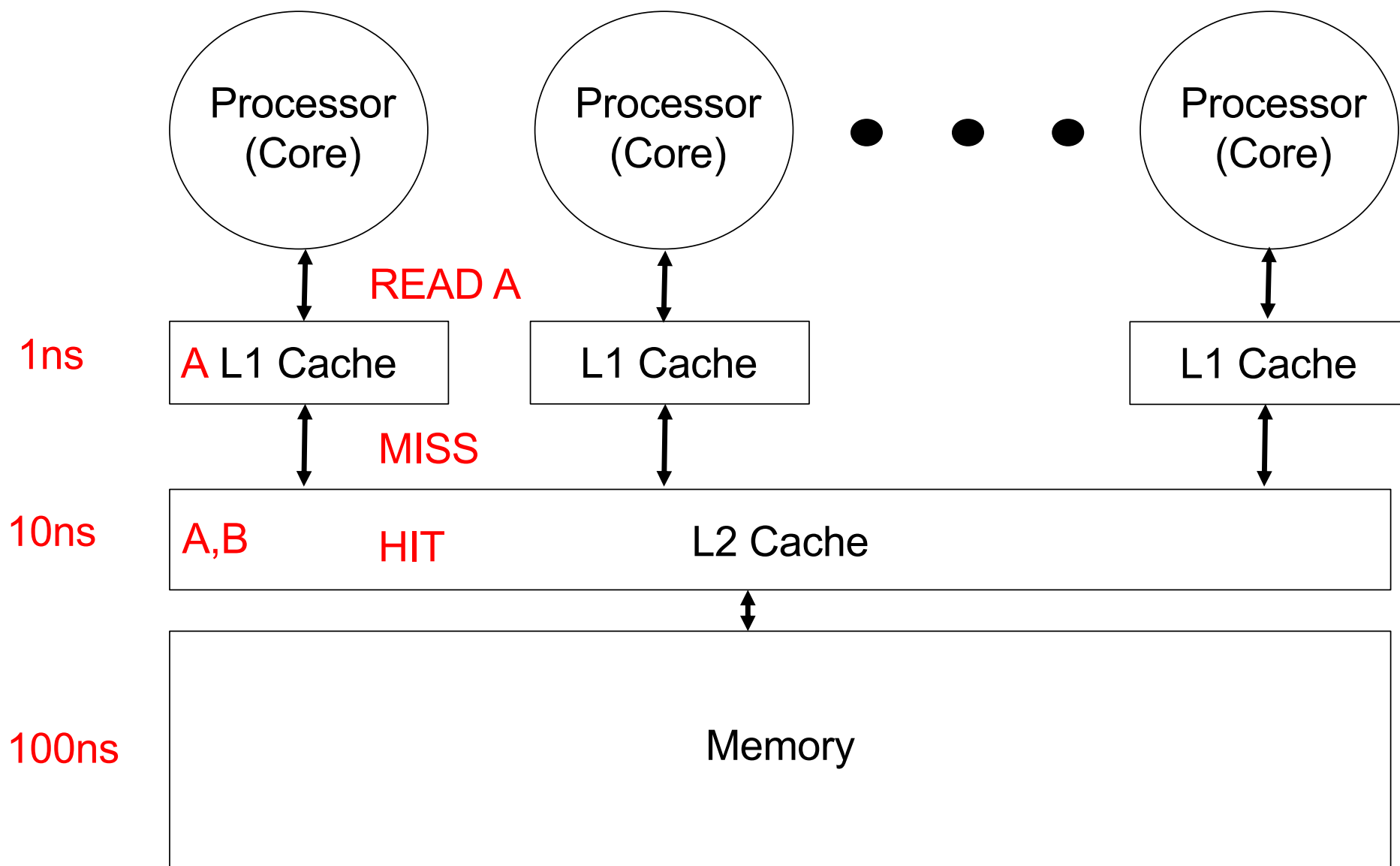
Cache Inclusion

Section 4.2.3









Impact on Execution Time

Two-level hierarchy

$$T = IC \times (CPI_0 + CPI_{L1} + MPI_{L1} \times CPI_{L2} + MPI_{L2} \times MEM) \times TPC$$

where MEM is processor/memory speedgap

= Memory access time/Processor cycle time

Question:

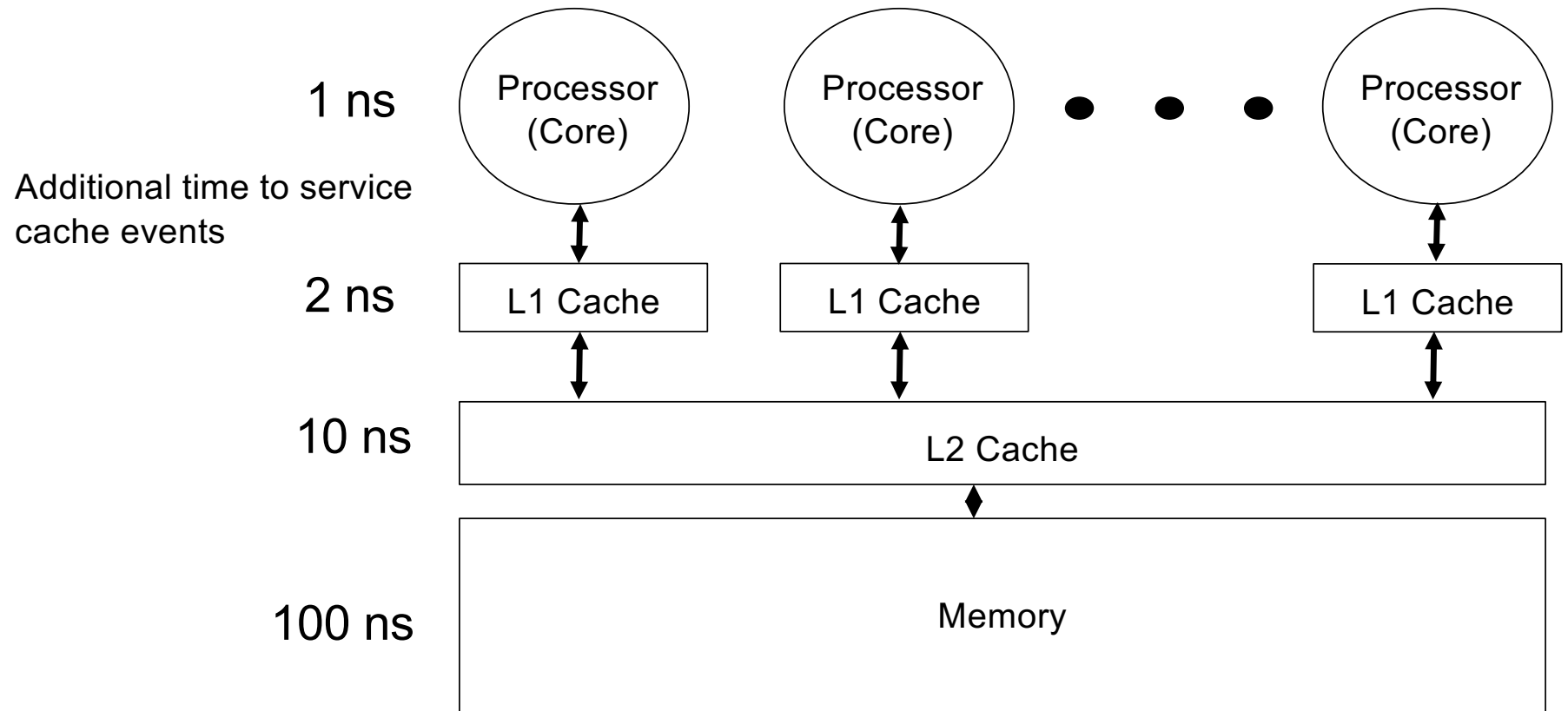
Assume $CPI_0=1$. What is CPI if $MPI_{L1}=0.02$ and $MPI_{L2}=0.01$?

Answer:

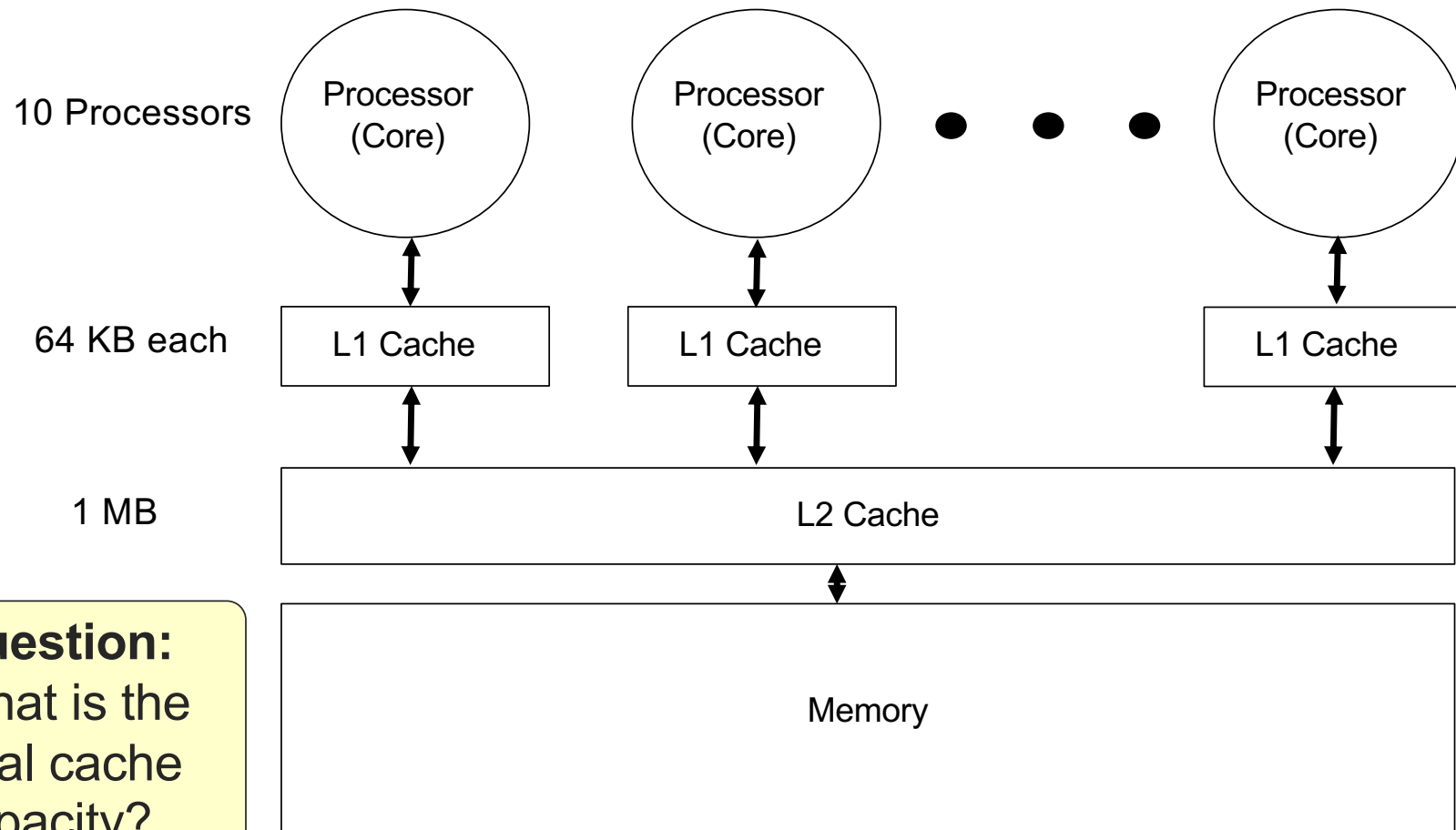
$$CPI = CPI_0 + CPI_{L1} + MPI_{L1} \times CPI_{L2} + MPI_{L2} \times MEM$$

$$CPI_0=1, CPI_{L1}=1, CPI_{L2}=10/1=10, MEM=100/1 = 100$$

$$CPI = 1+1 + 0.02 \times 10 + 0.01 \times 100 = 3.2$$



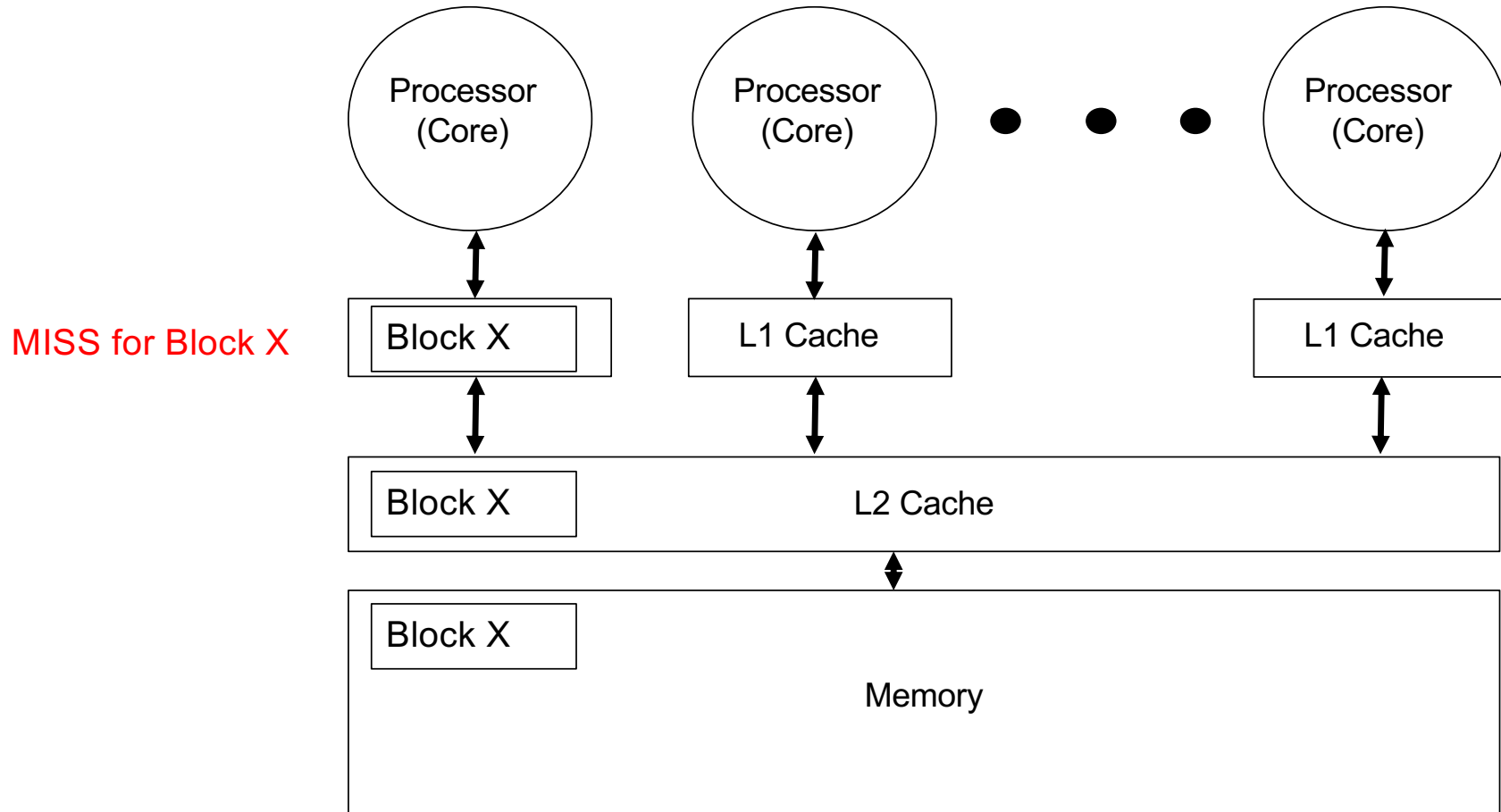
Cache Capacity



Question:
What is the
total cache
capacity?

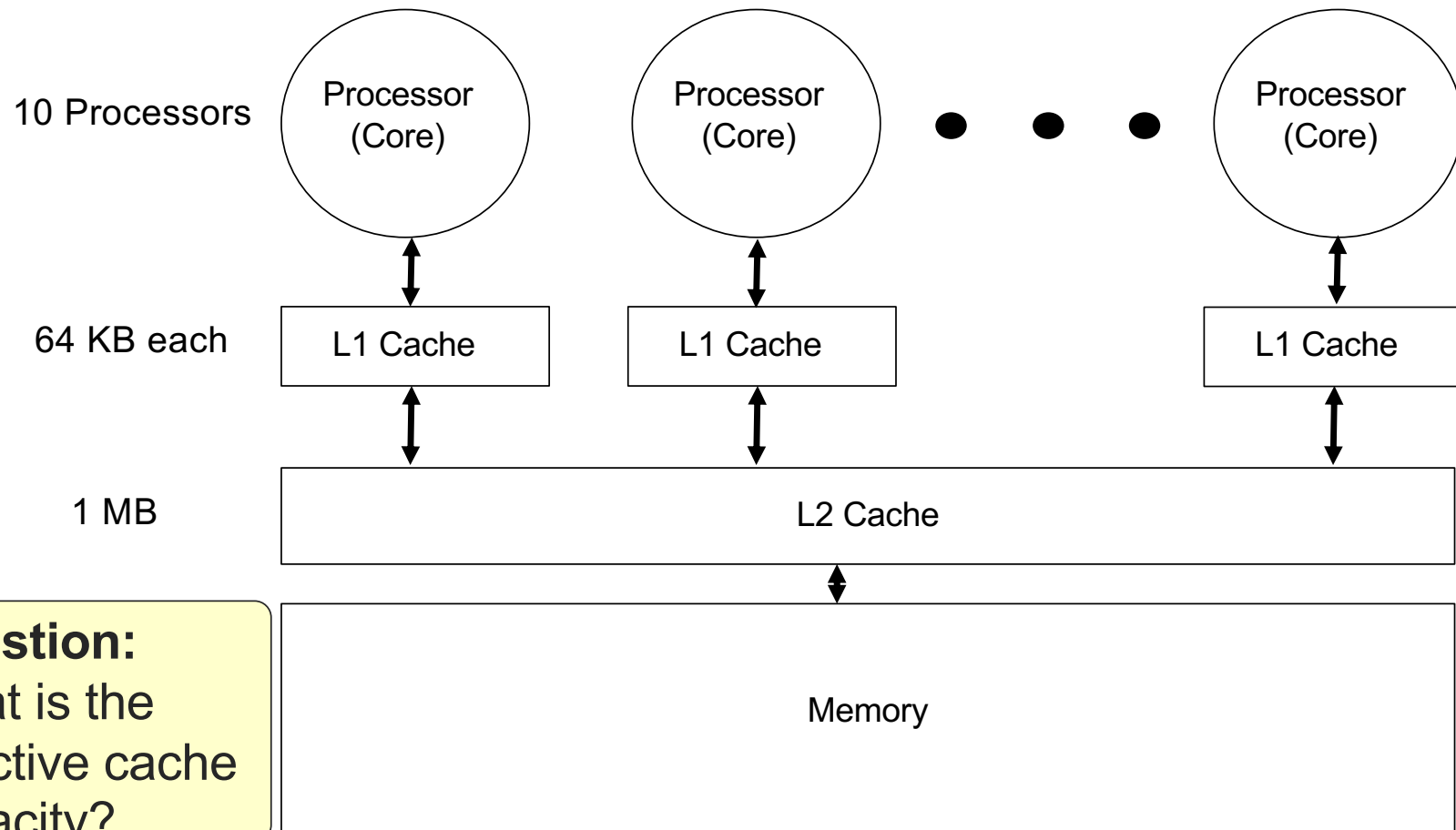
Answer:
 $64 \text{ KB} \times 10 +$
 $1 \text{ MB} = 1024$
 $\text{KB} + 640 \text{ KB}$
 $= 1684 \text{ KB}$

Inclusion Policy



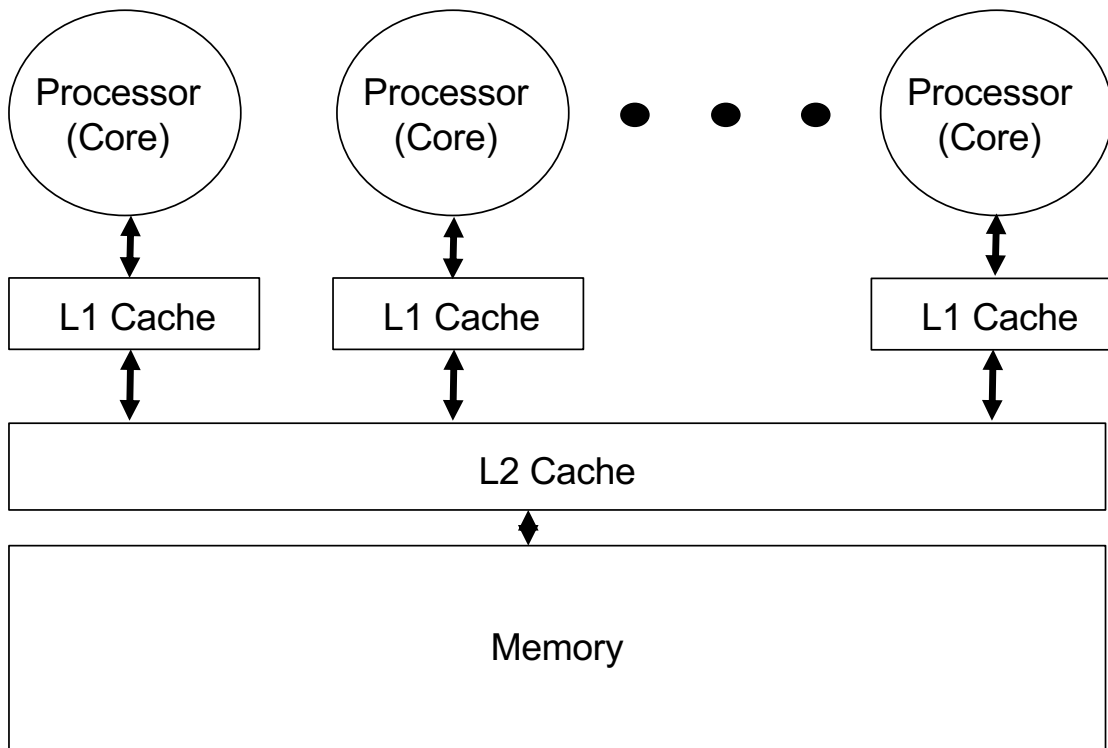
A block occupies space in both L1 and L2

Inclusion: Effective cache capacity?



Answer:
Size of L2:
1 MB

Inclusion Policy

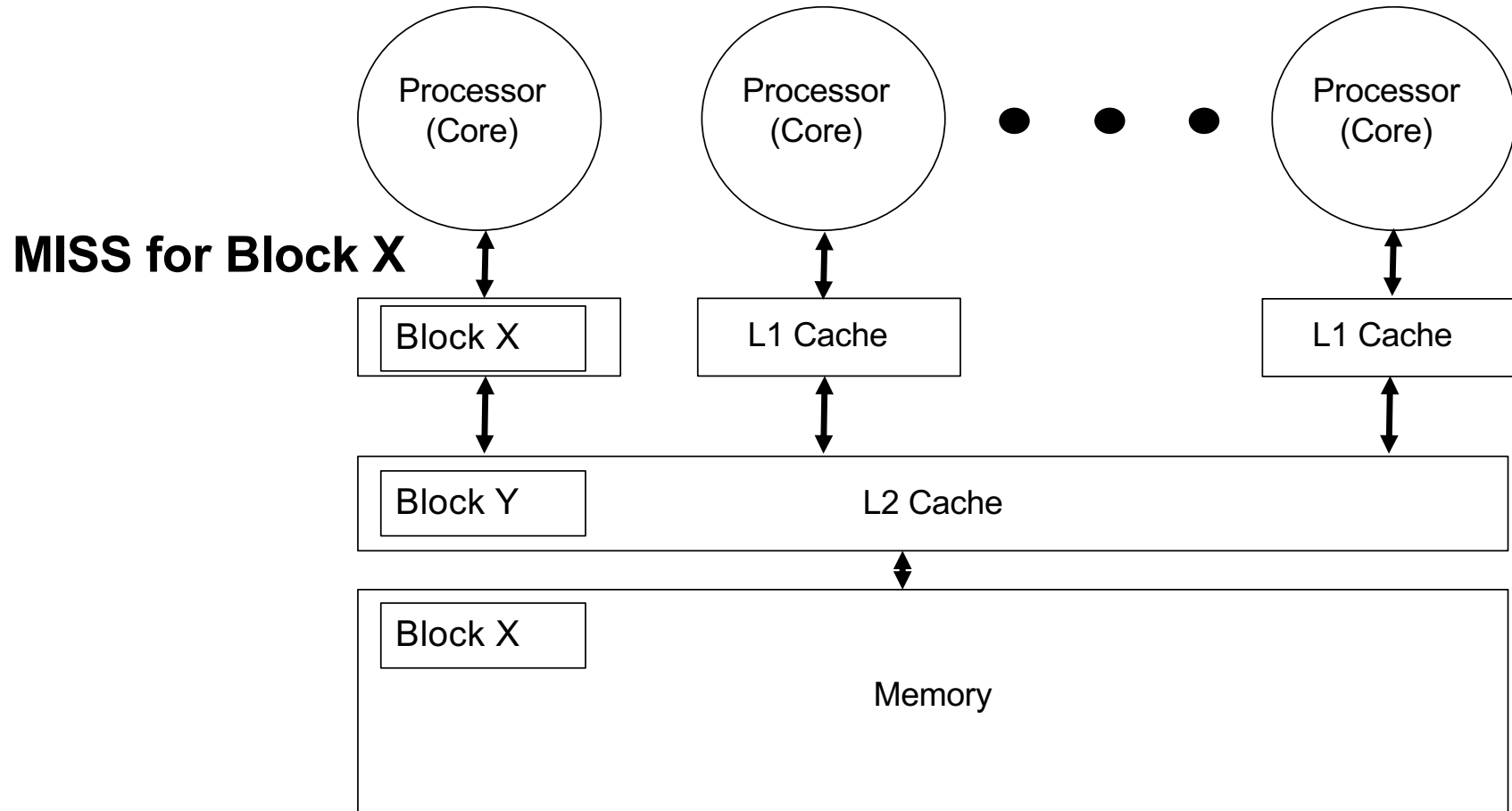


Cache inclusion:

- **L1/L2 miss: Fill L1 and L2**
- **L2 eviction (replacement): Evict block in L1**

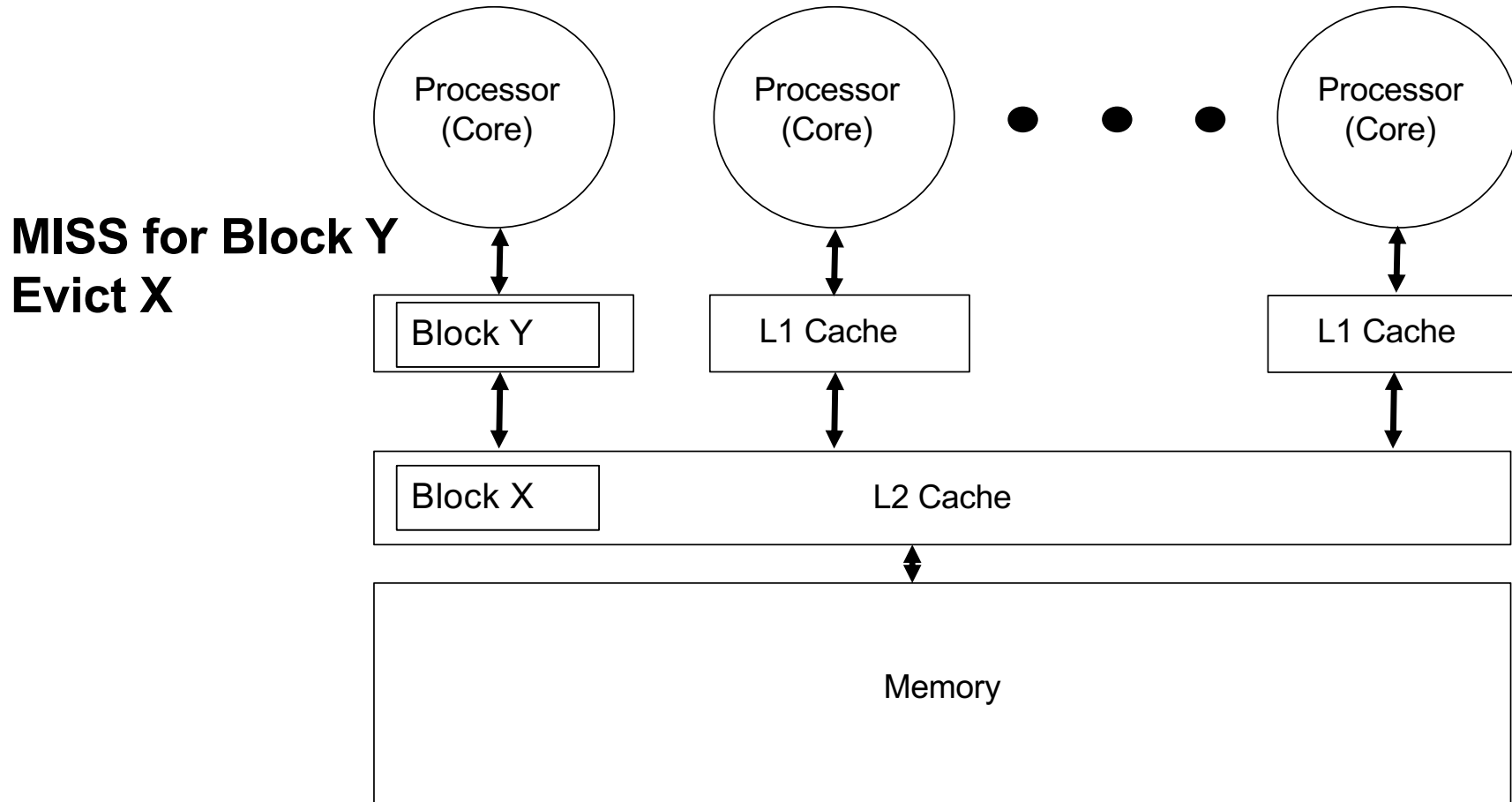
Cache capacity = L2 (not L1 + L2)

Exclusion Policy



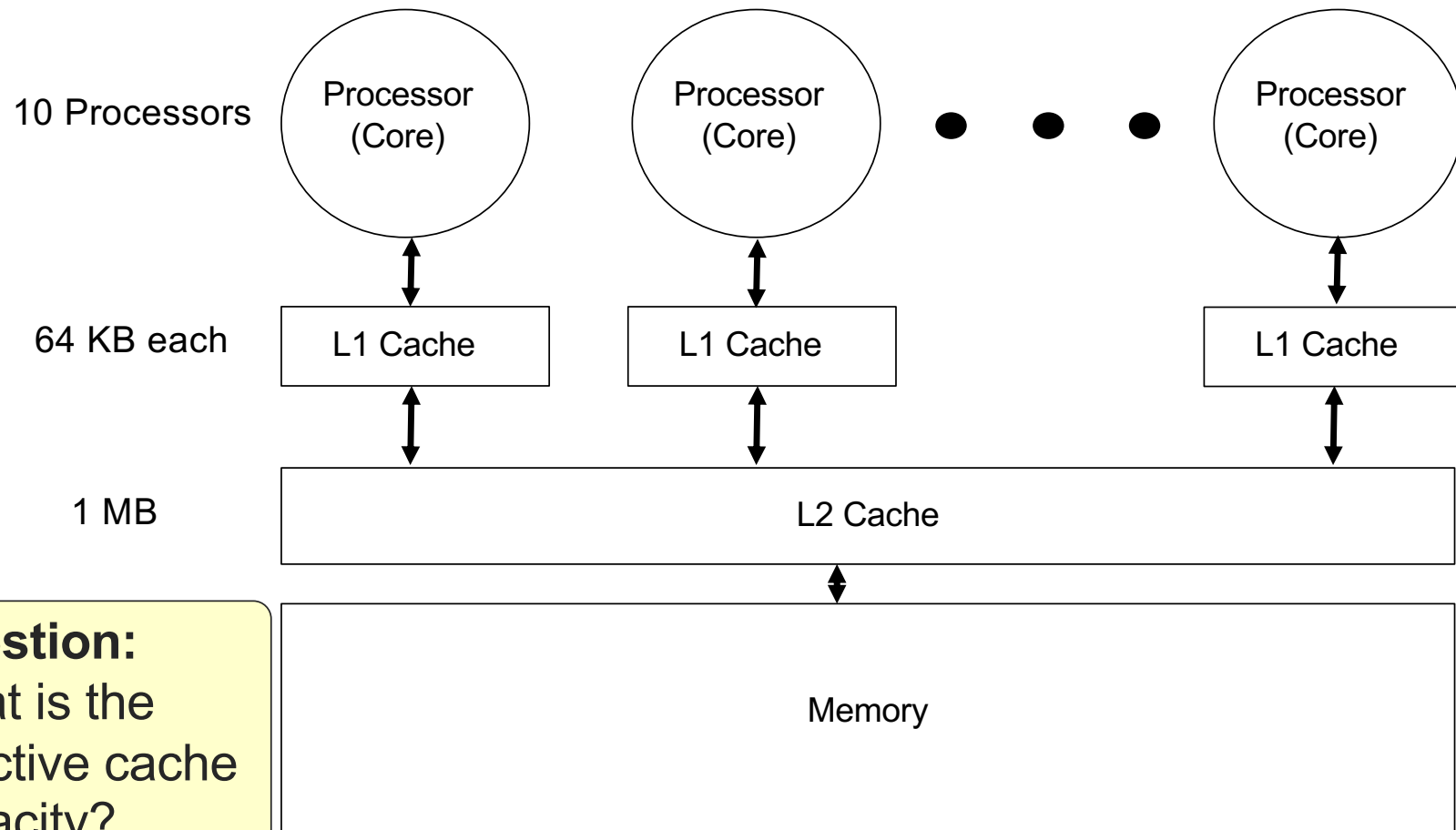
Cache space = L1 + L2 (not L2 only)

L1 Miss – L2 Hit



Swap blocks X and Y

Exclusion: Effective cache capacity?



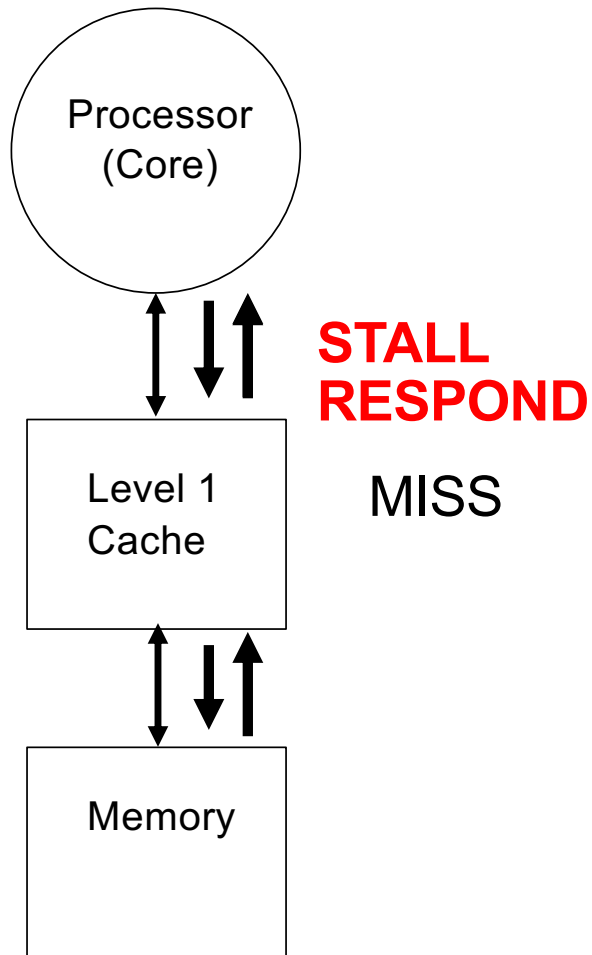
Question:
What is the
effective cache
capacity?

Answer:
Size of L1+L2:
~1.6 MB

Non-Blocking (Lockup-free) Caches

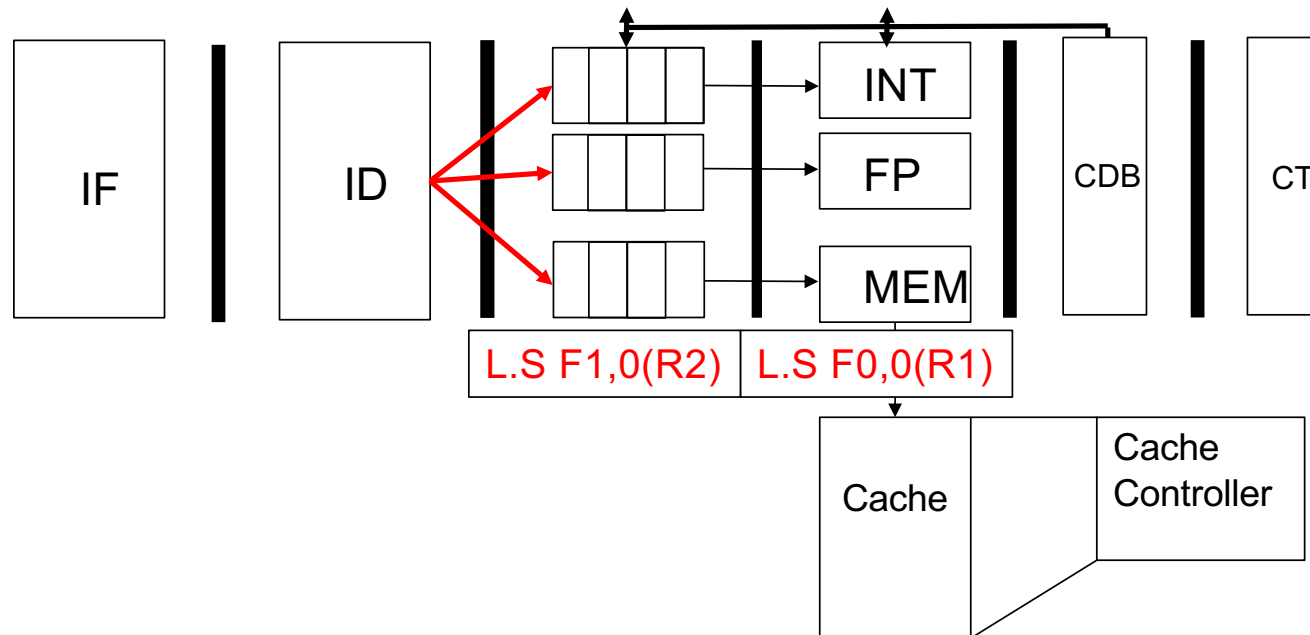
(Section 4.3.6)

Blocking Caches 1(2)

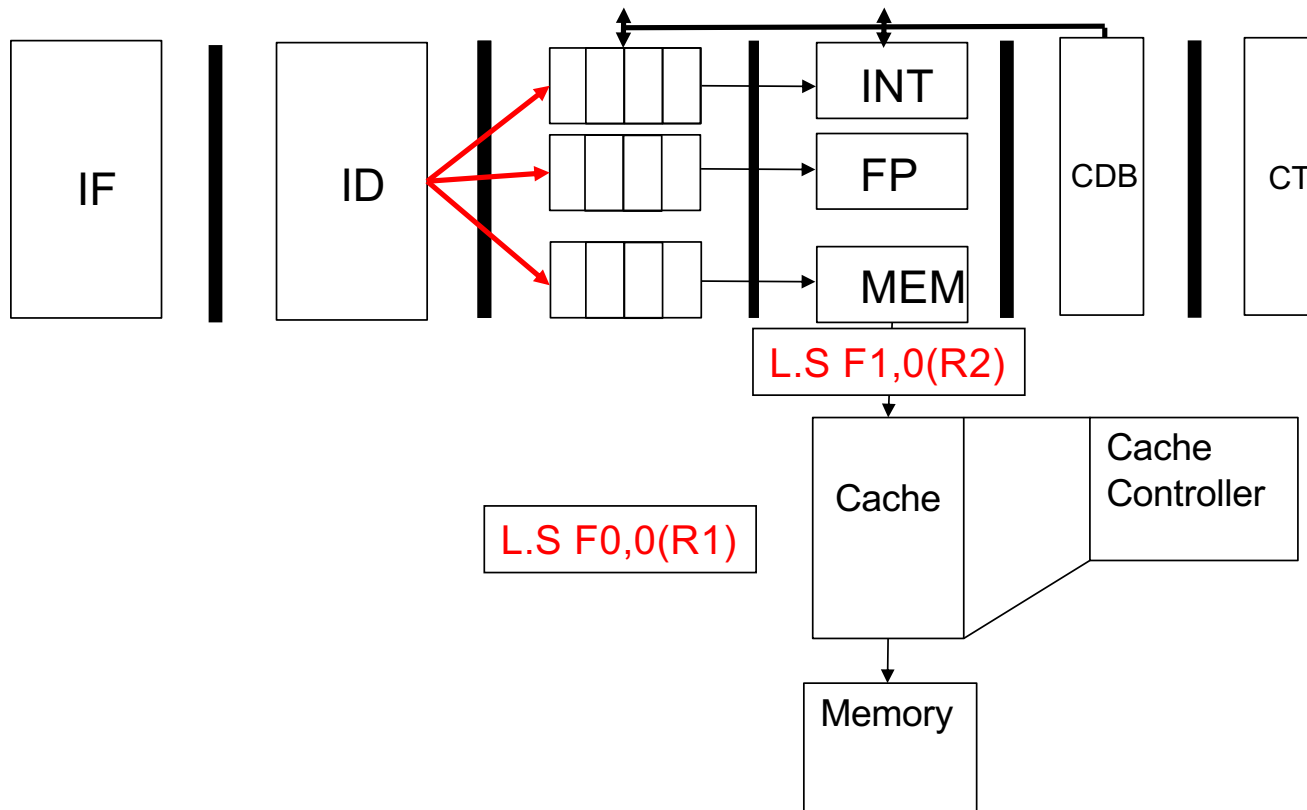


Processor stalls until cache has serviced a request (hit or miss)

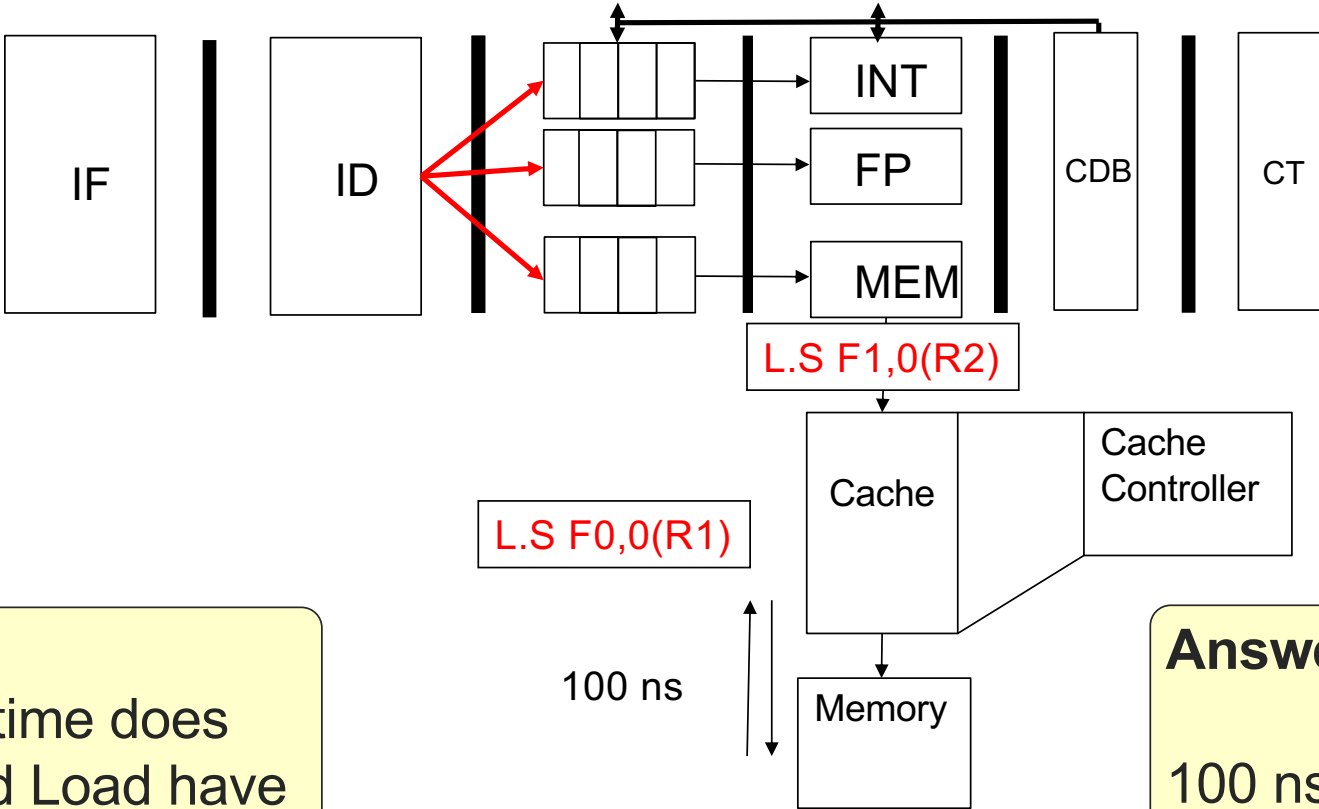
Anatomy of a Blocking Cache



Blocking Cache: Misses



Blocking Cache: Misses



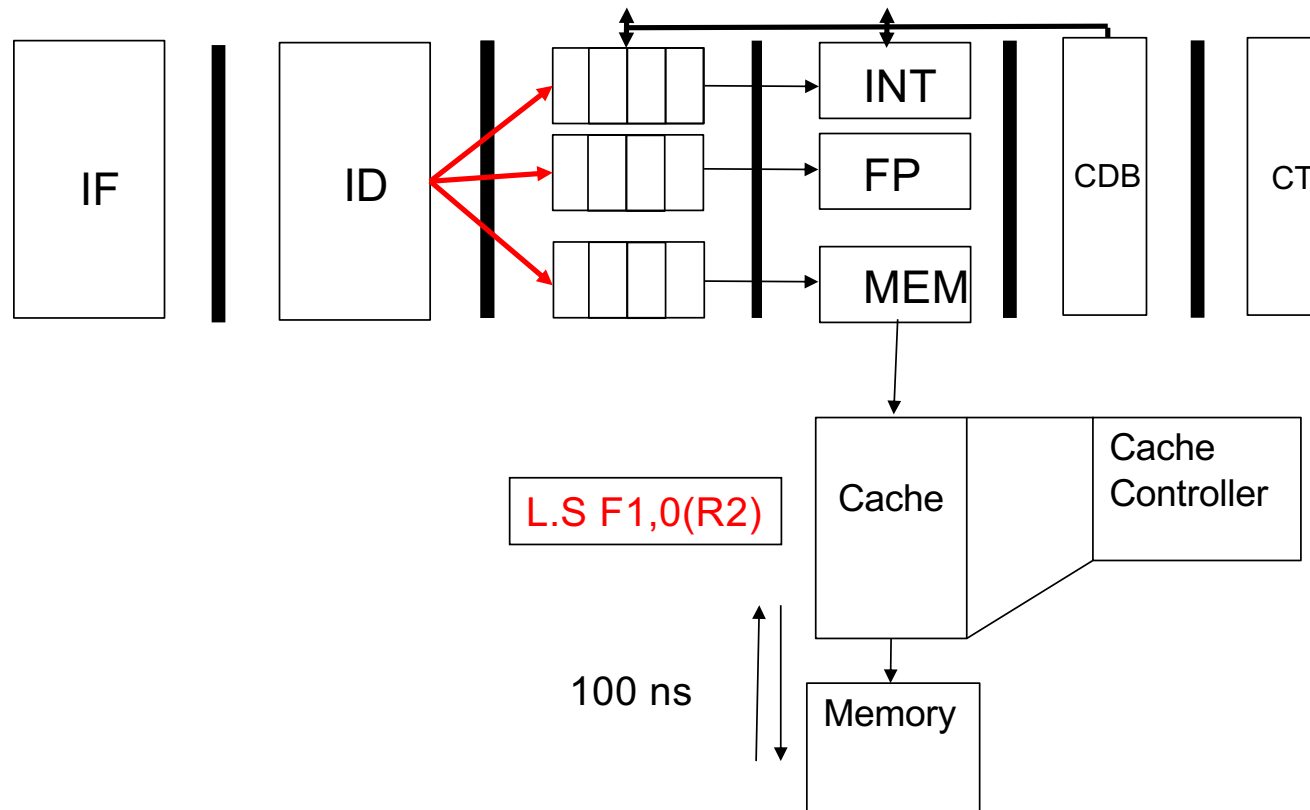
Question:

How long time does the second Load have to wait?

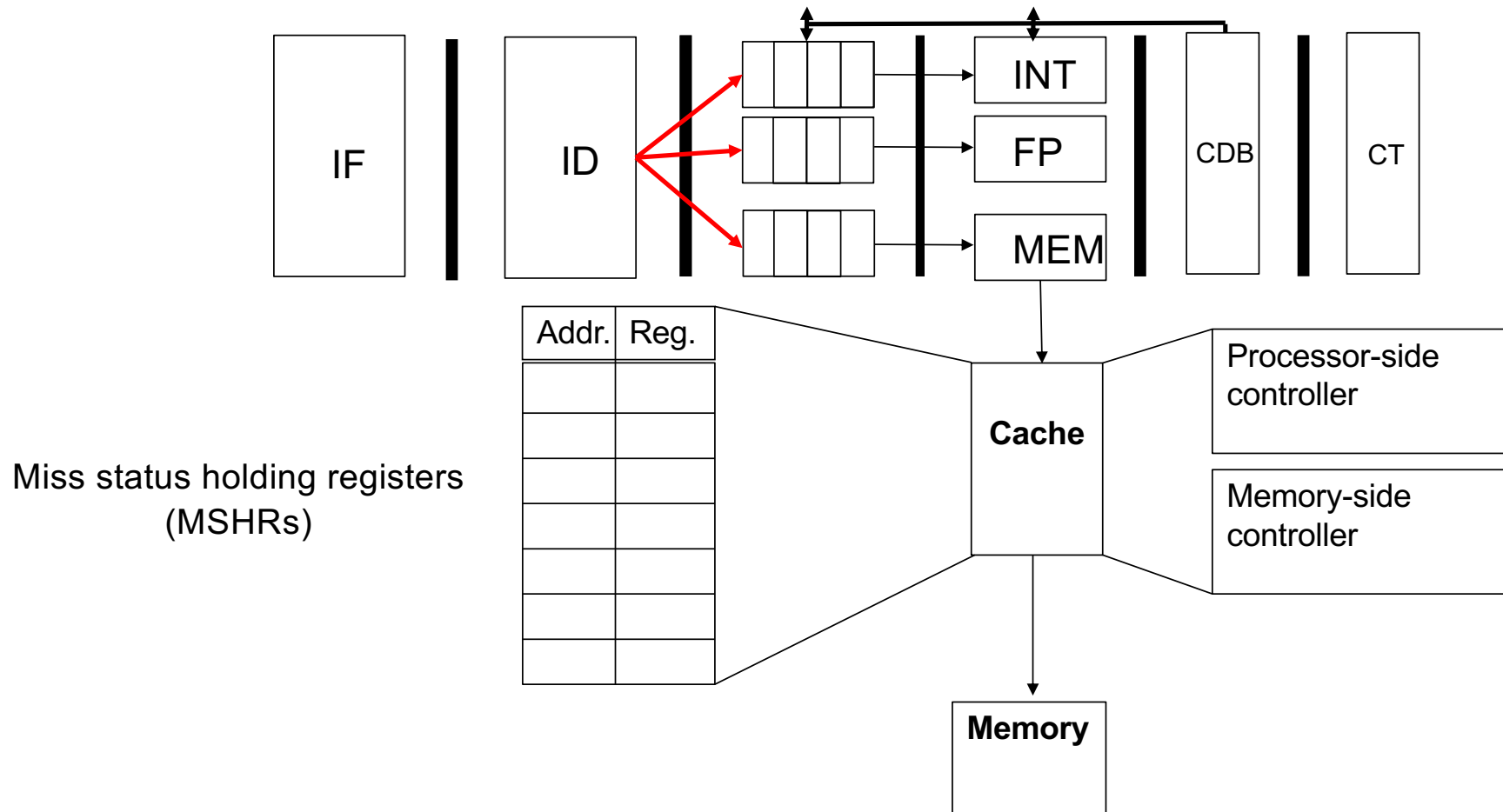
Answer:

100 ns

Blocking Cache: Misses

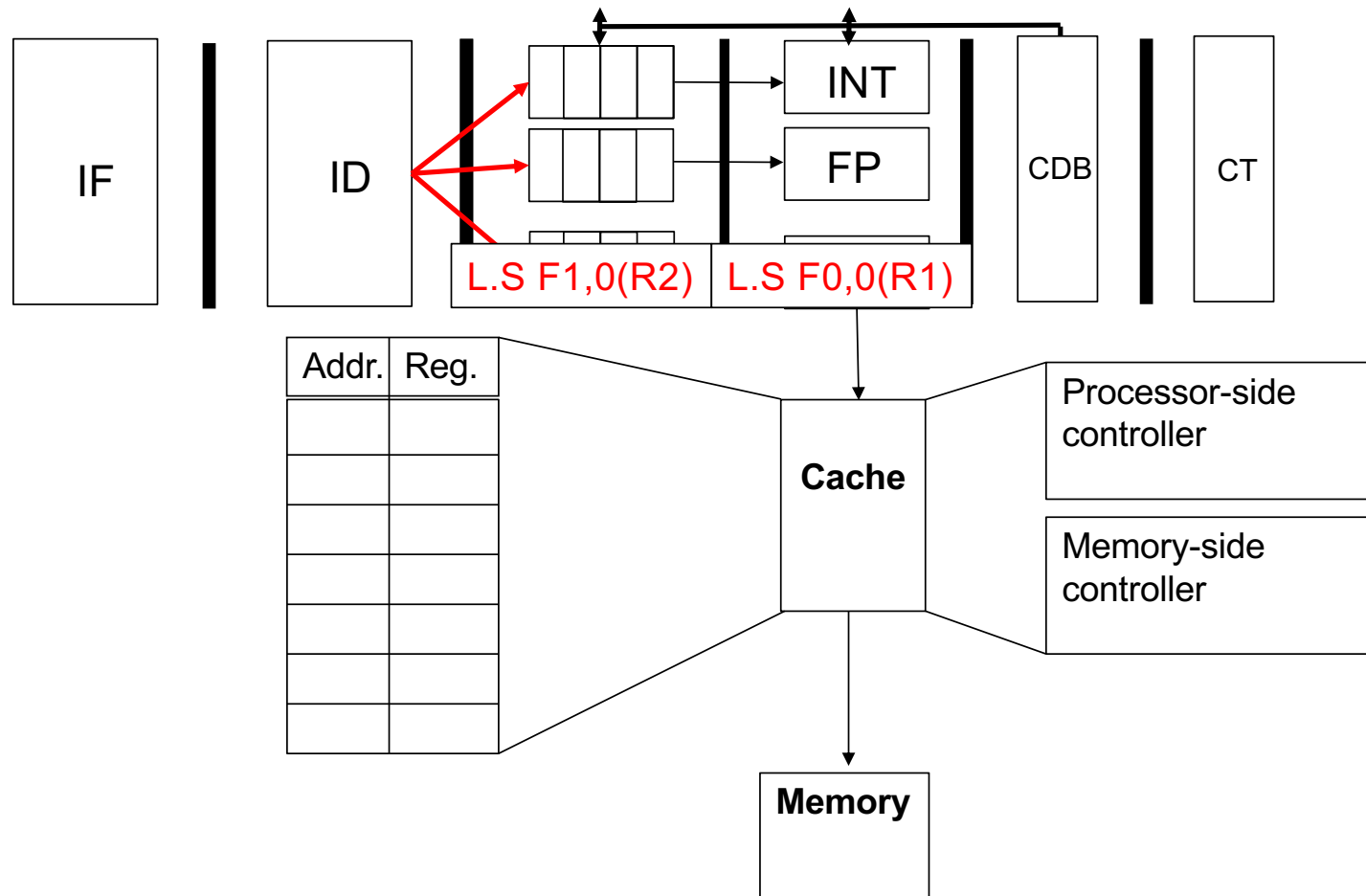


Anatomy of a Non-Blocking (or Lockup-free) Caches

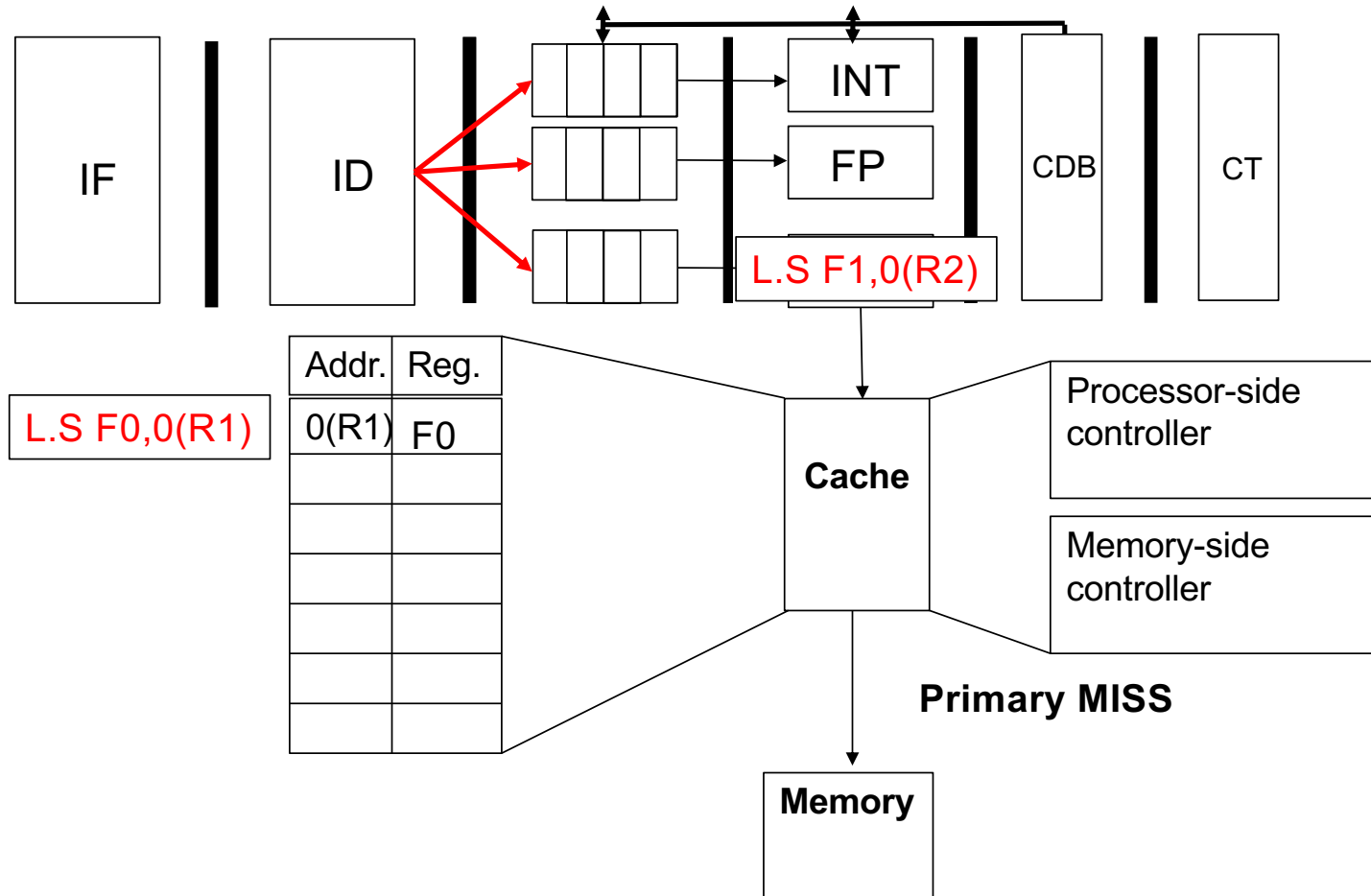


Non-Blocking Caches – Misses

Cycle: 1

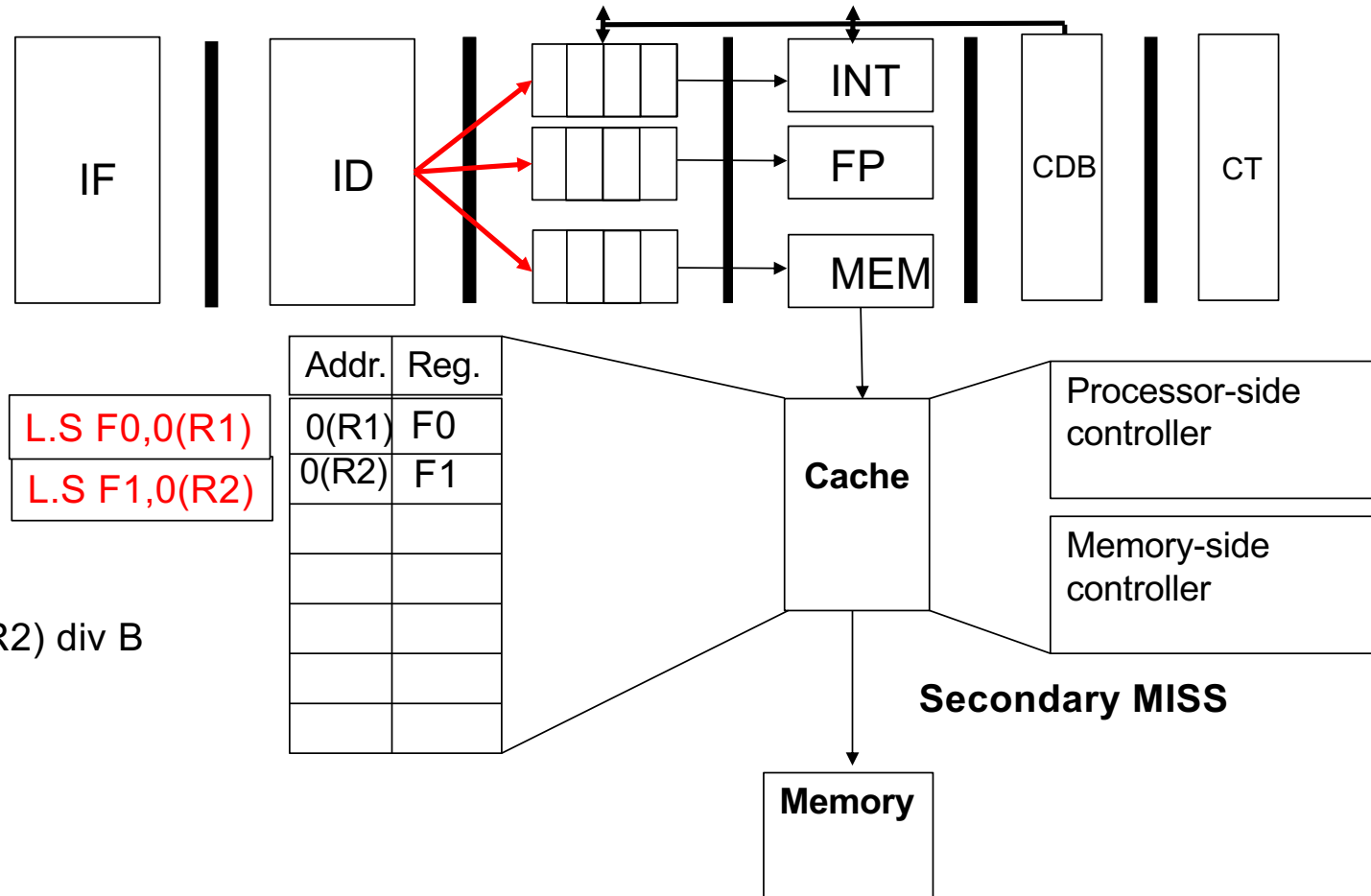


Cycle: 2



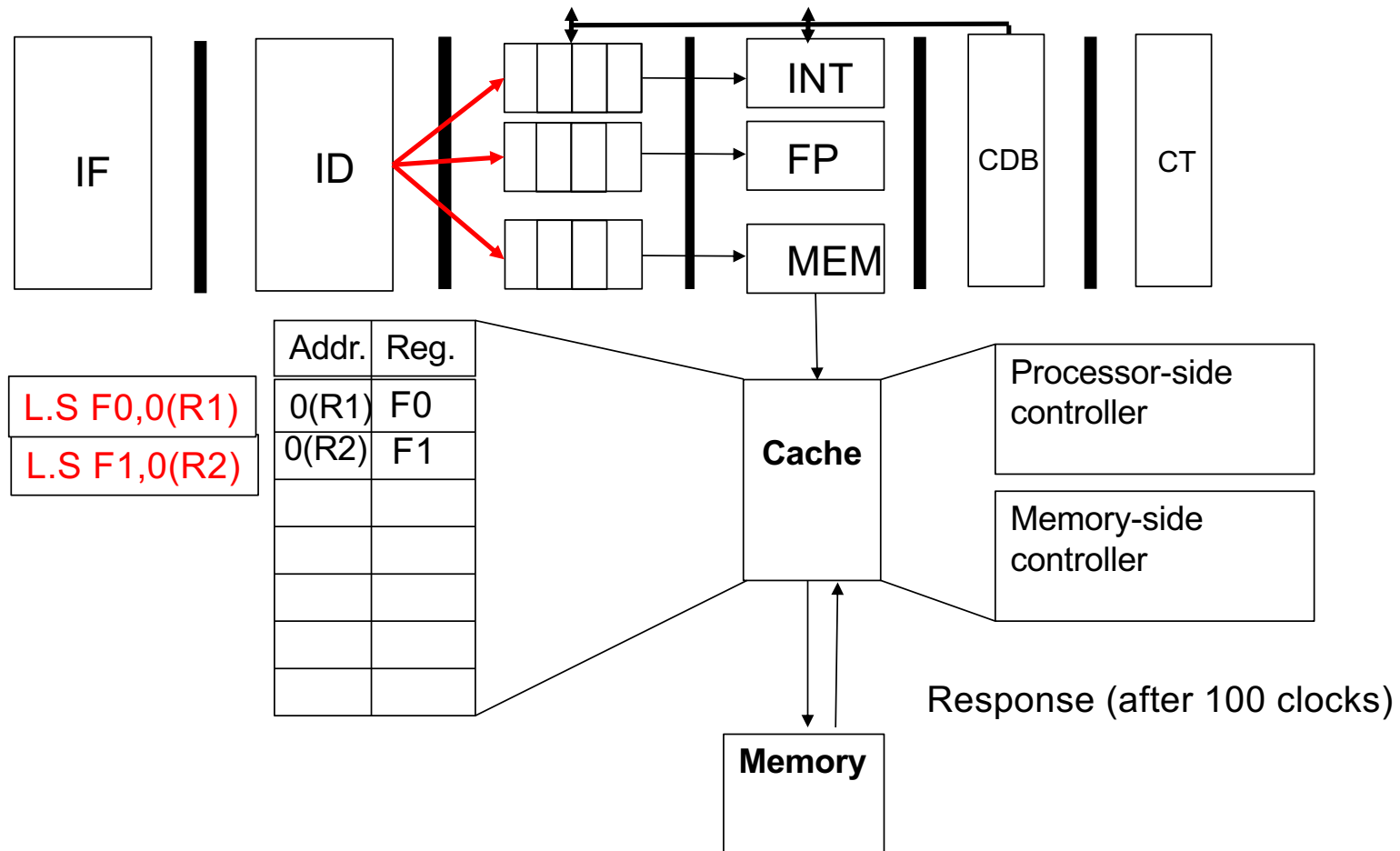
Non-Blocking Caches – Misses

Cycle: 3



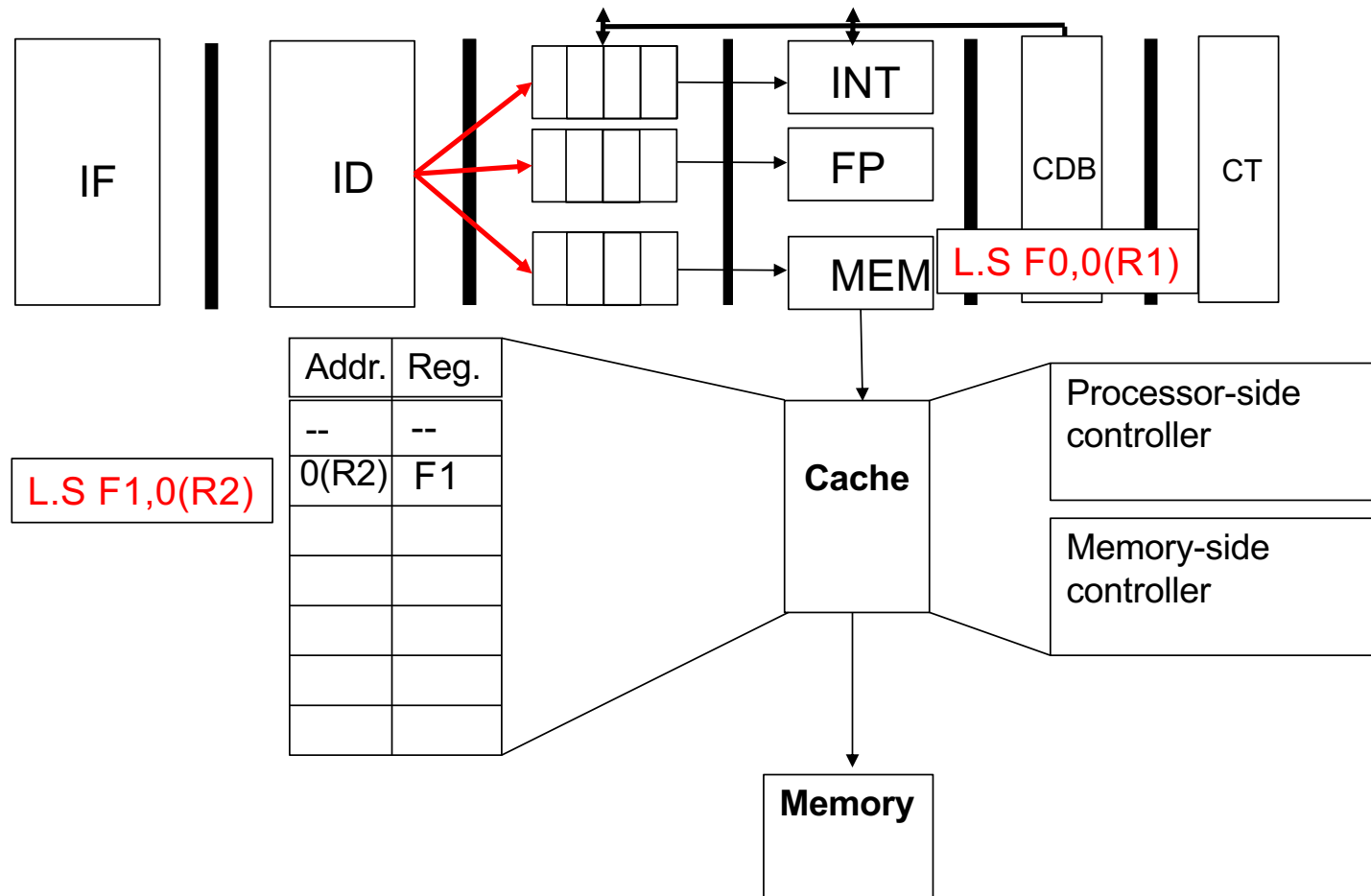
Non-Blocking Caches – Misses

Cycle: 103



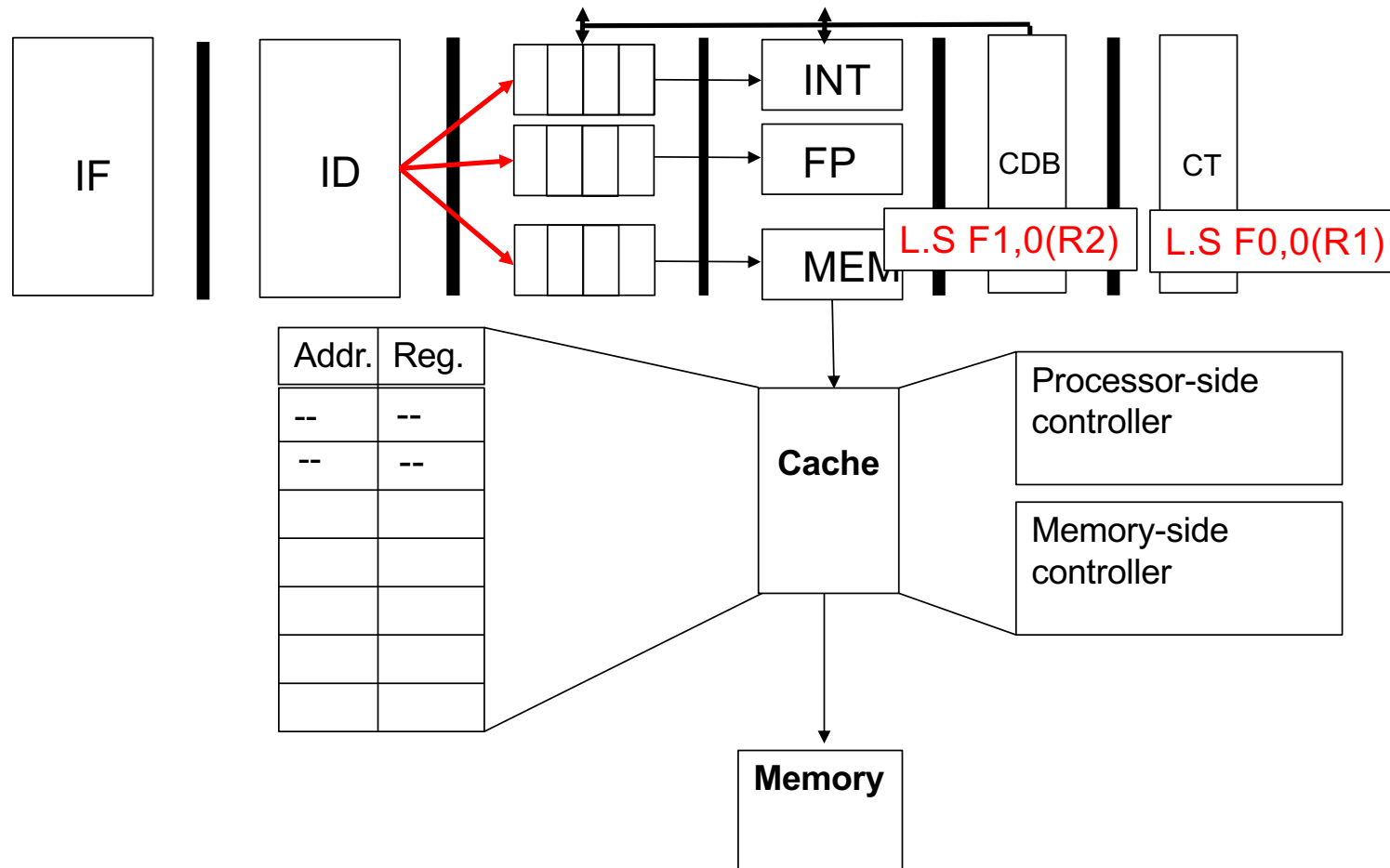
Non-Blocking Caches – Misses

Cycle: 104



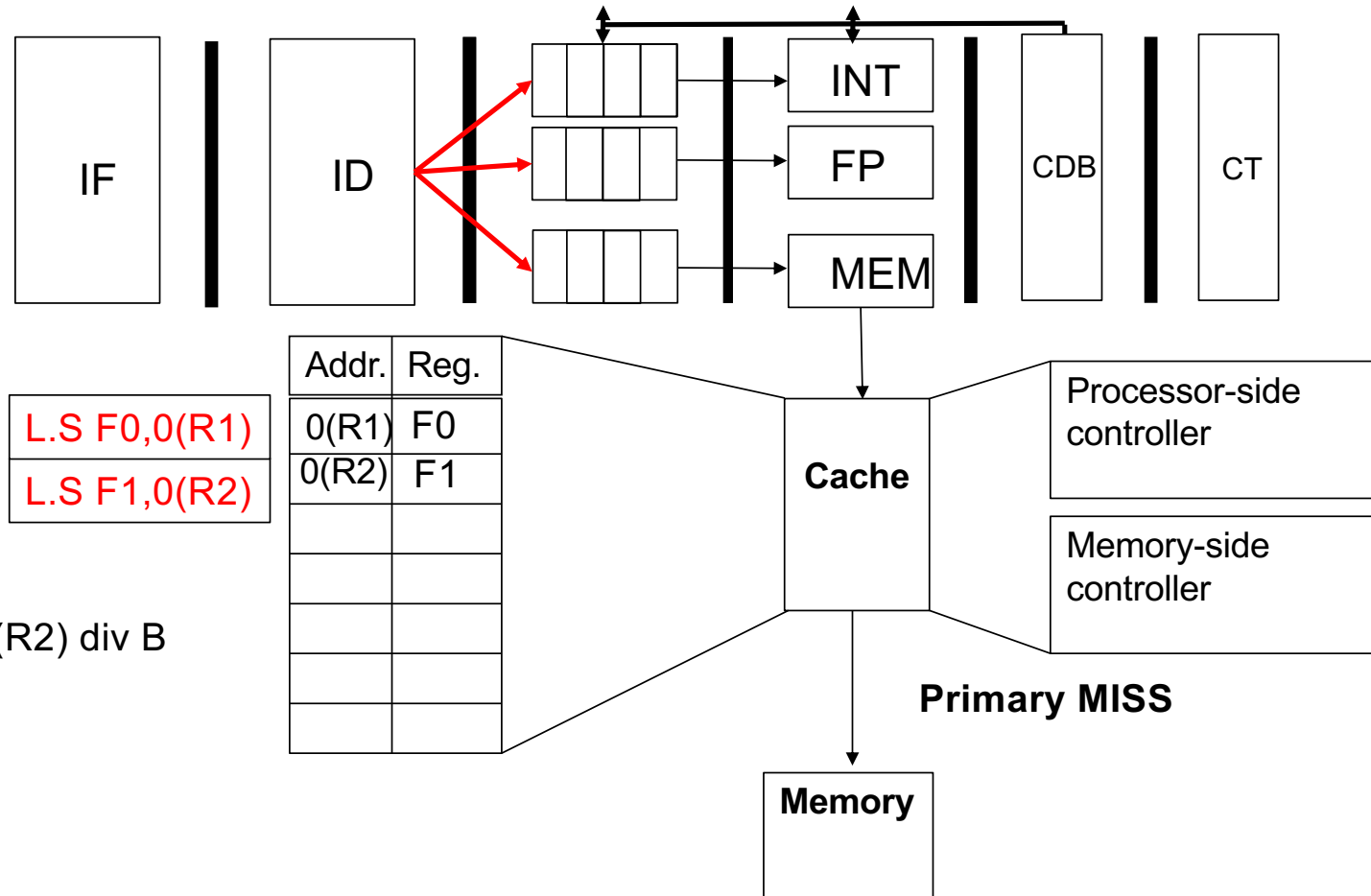
Non-Blocking Caches – Misses

Cycle: 105



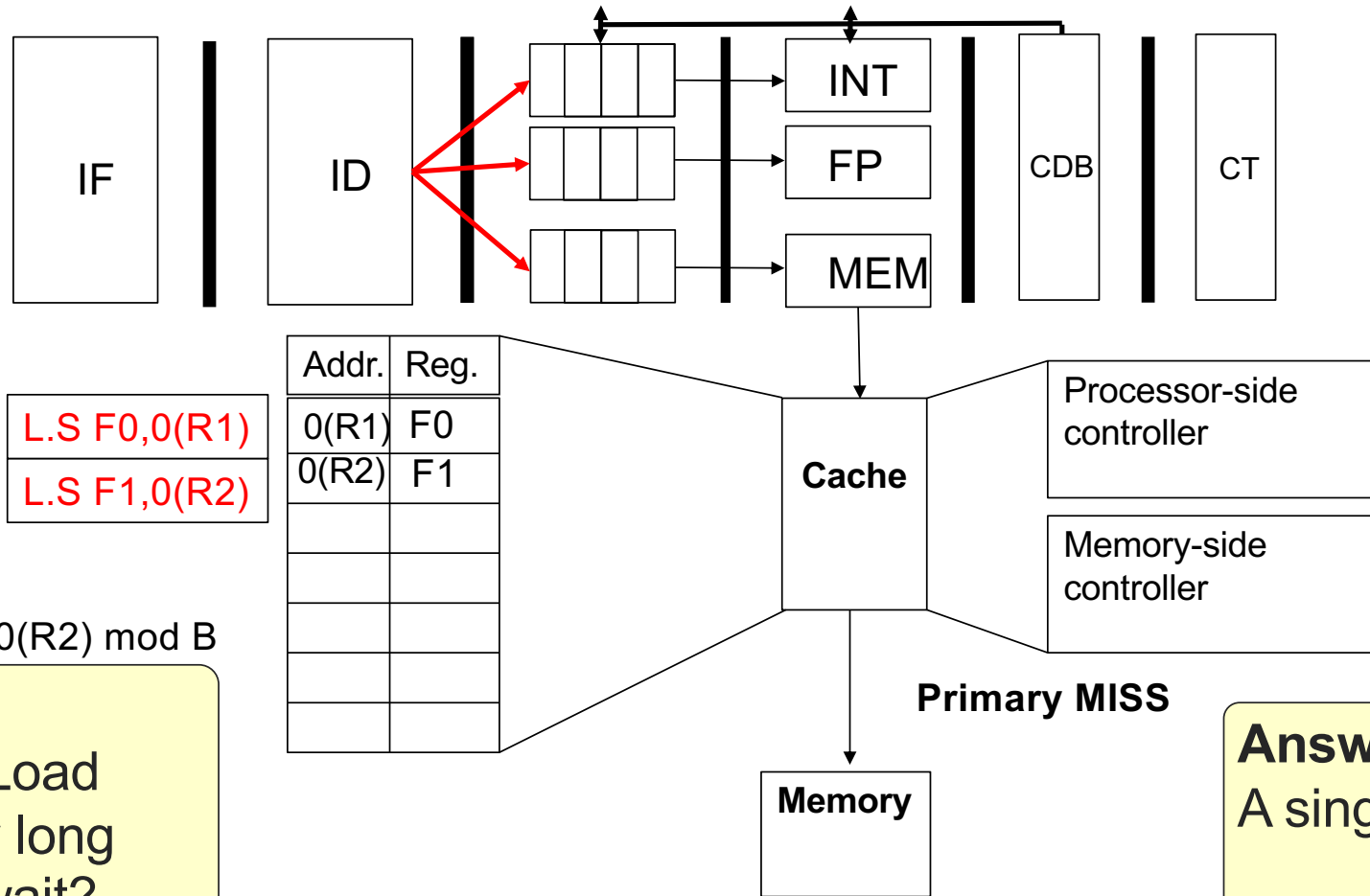
Non-Blocking Caches – Misses

Cycle: 3



Non-Blocking Caches – Misses

Cycle: 3



Question:

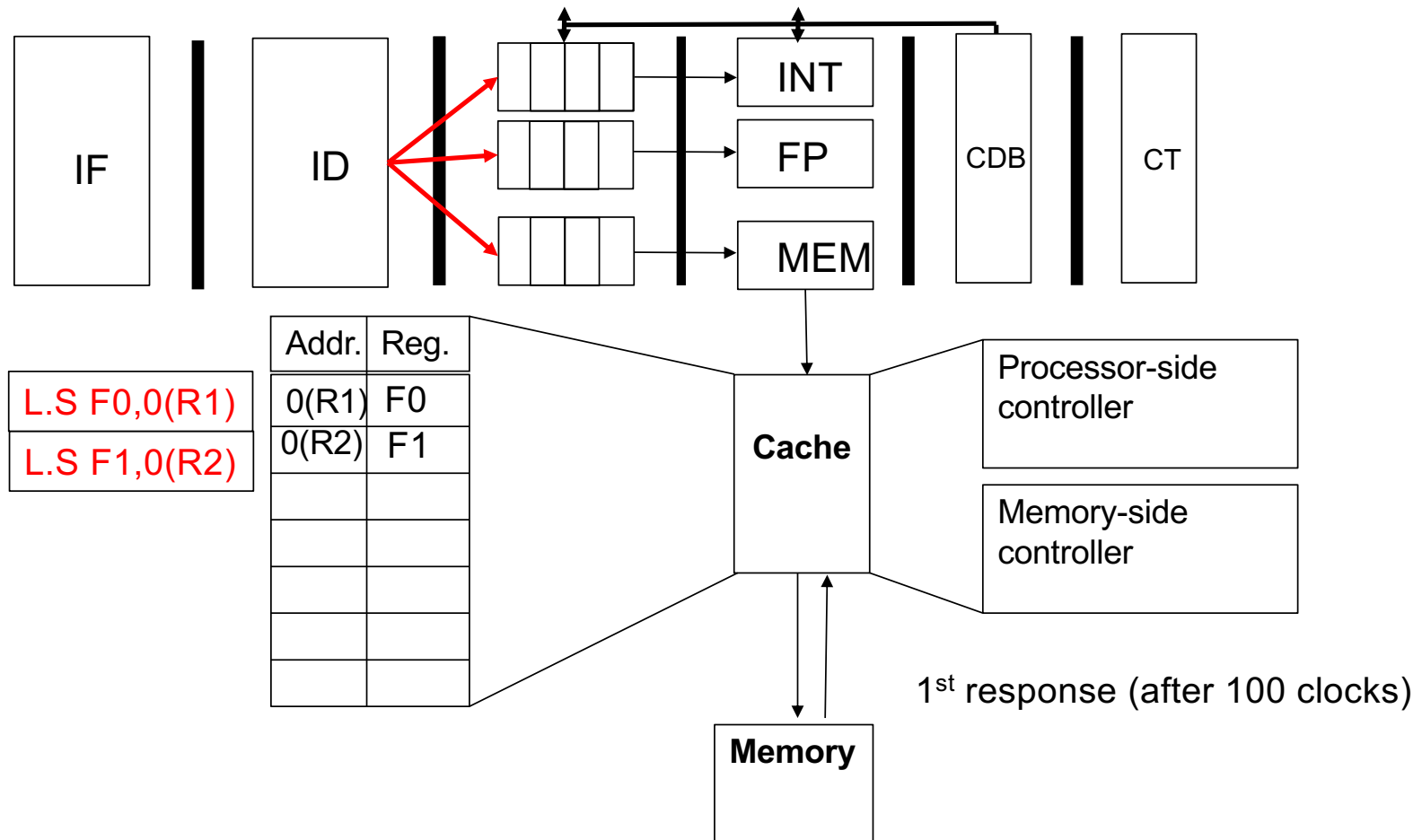
If the second Load would hit, how long time would it wait?

Answer:

A single cycle

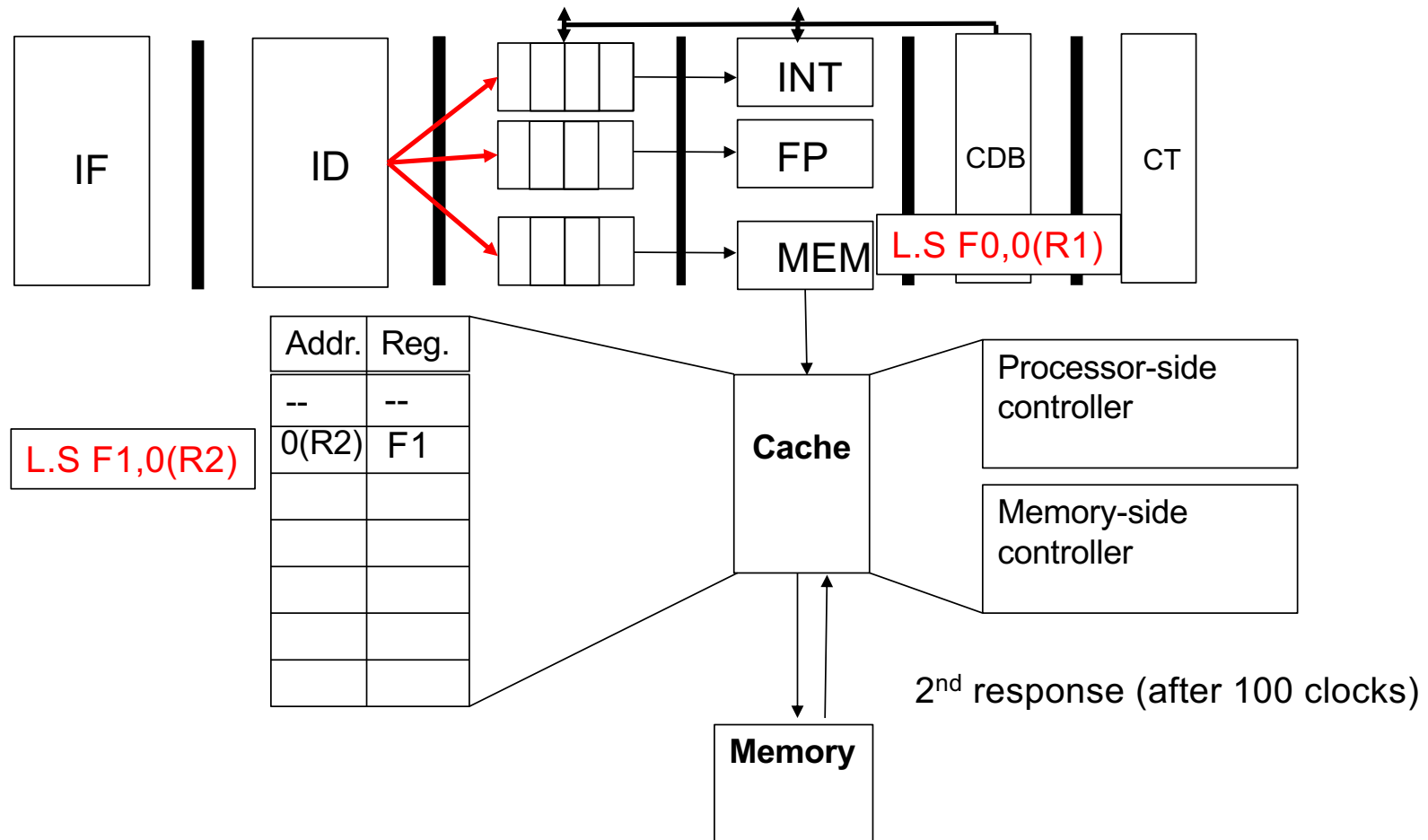
Non-Blocking Caches – Misses

Cycle: 103



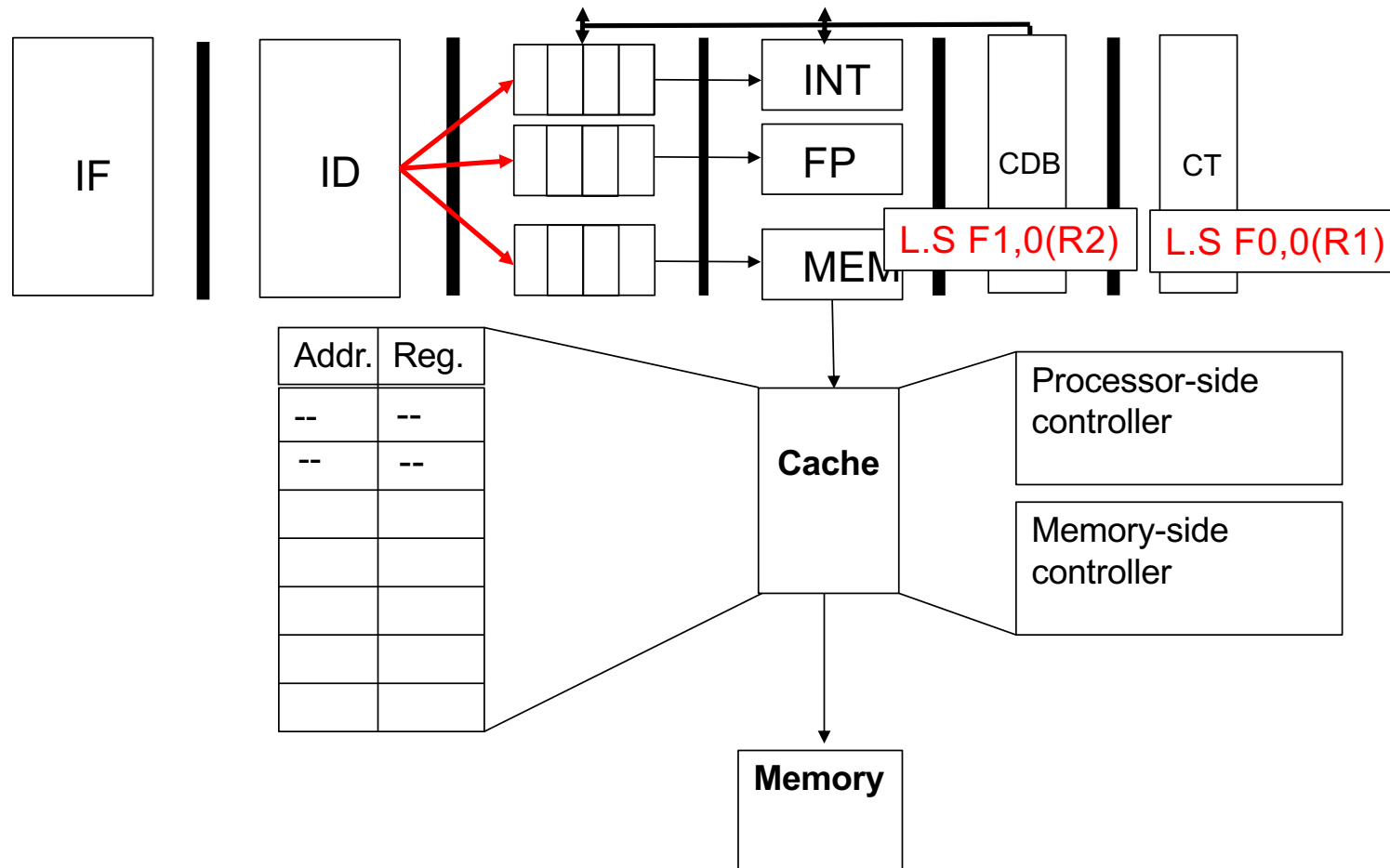
Non-Blocking Caches – Misses

Cycle: 104



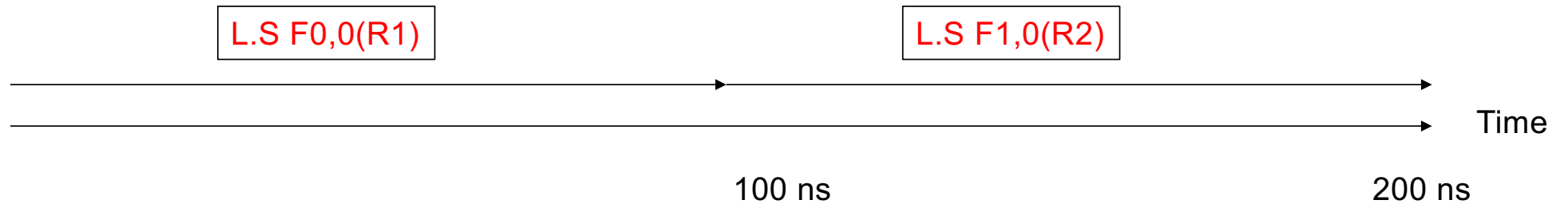
Non-Blocking Caches – Misses

Cycle: 105

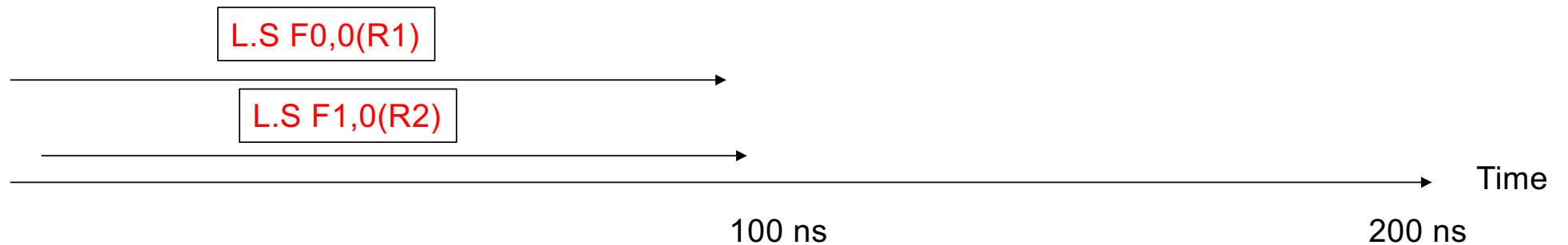


Blocking vs. Non-Blocking Caches

Blocking Cache



Non-Blocking Cache



Example

Question:

Assume that the block size is four words. Determine the number of Primary and Secondary Misses if the loop is executed 16 iterations. Assume that the number of MSHRs suffices.

```
LOOP: LW R1,0(R2)
      ADDI R2,R2,#4
      BNEZ R2,R4, LOOP
```

Example (Cont'd)

```
LOOP: LW R1,0(R2)
      ADDI R2,R2,#4
      BNEZ R2,R4, LOOP
```

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15	I16
Primary	X				X				X				X			
Secondary		X	X	X		X	X	X		X	X	X		X	X	X

Answer:

The first access to each block will generate a primary miss and the next three accesses will generate a secondary miss:

- Number of Primary Misses: 4
- Number of Secondary Misses: 12

Another Example

Question:

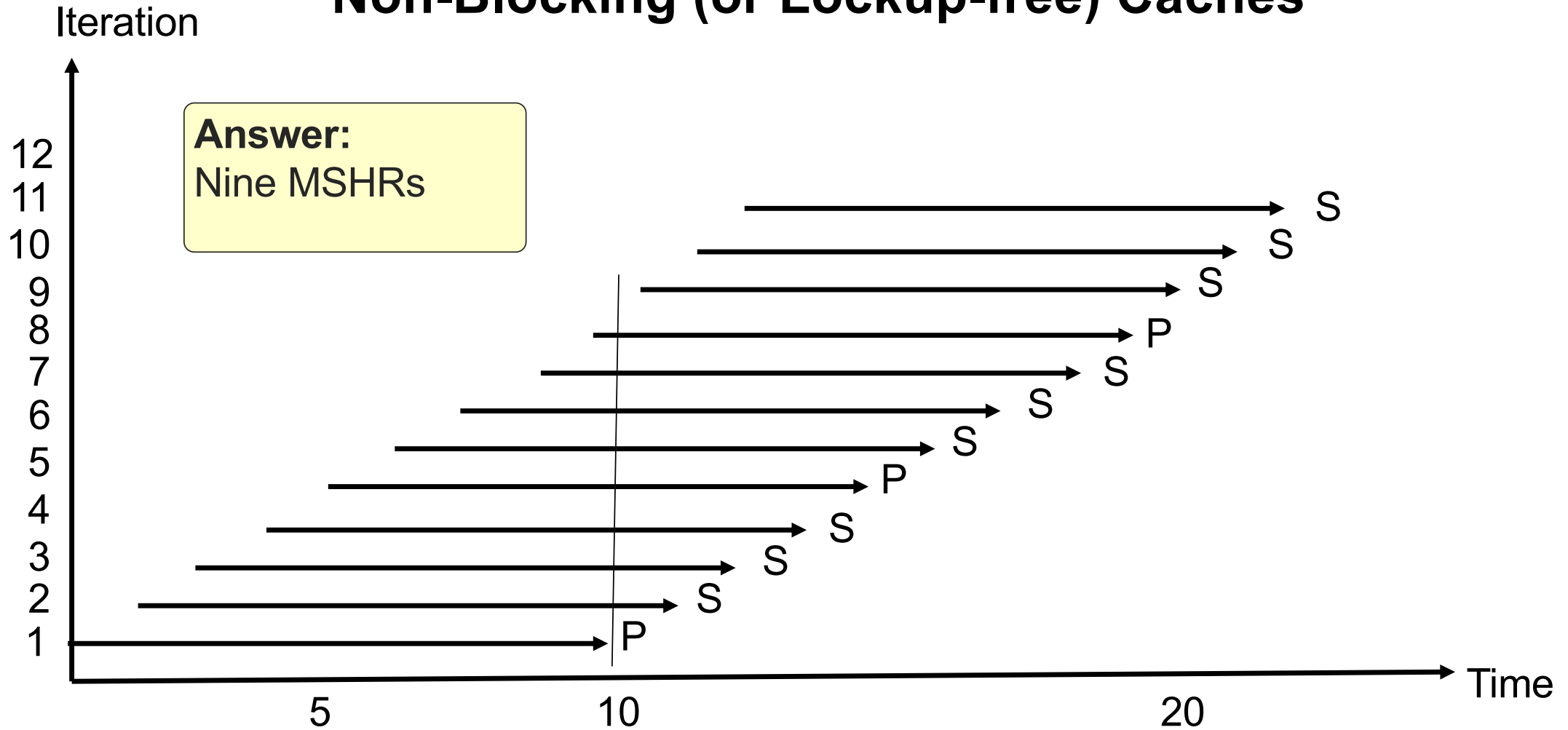
Assume that

- The block size is four words
- It takes 10 cycles to handle a miss.
- 12 iterations are executed

How many MSHRs are used?

```
LOOP: LW R1,0(R2)
      ADDI R2,R2,#4
      BNEZ R2,R4, LOOP
```

Non-Blocking (or Lockup-free) Caches



Cache Prefetching (Section 4.3.7)

Cache Prefetching

Concept: Bring instructions/data into cache prior to access

Two challenges:

- **WHAT:** What data will be accessed in future?
- **WHEN:** When will it be accessed?

Example

```
LOOP: L.D F2, 0(R1)
      ADD.D F4, F2, F0
      S.D F4, 0(R1)
      ADDI R1, R1, #8
      SUBI R2, R2, #1
      BNEZ R2, LOOP
```

Instruction access sequence: I, I+4, I+8...

Data access sequence: 100, 108, 116, ...

Highly predictable

WHAT to Prefetch?
WHEN to Prefetch?

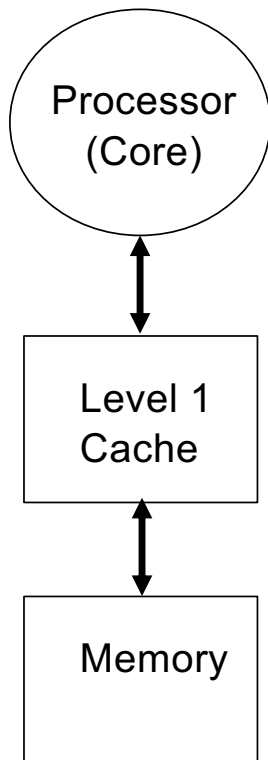
Sequential Prefetching

WHAT to Prefetch?

WHEN to Prefetch?

Concept: On a miss, fetch two blocks instead of one

Block size = 4 8-byte (long)words



Miss: Bring block B and B+1

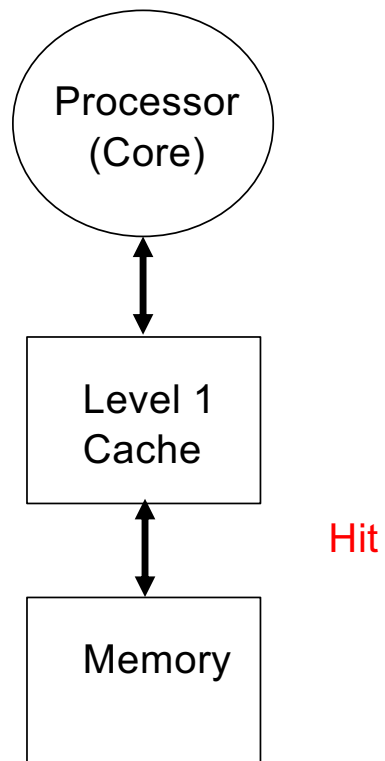
Iteration: 1



```
LOOP L.D F2,0(R1)
      ADD.D F4,F2,F0
      S.D F4,0(R1)
      ADDI R1,R1,#8
      SUBI R2,R2,#1
      BNEZ R2, LOOP
```

Sequential Prefetching

WHAT to Prefetch?
WHEN to Prefetch?

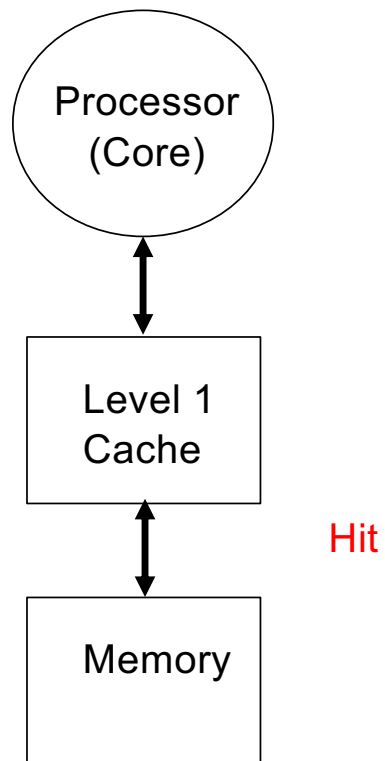


Iteration: 2

**→ LOOP L.D F2,0(R1)
ADD.D F4,F2,F0
S.D F4,0(R1)
ADDI R1,R1,#8
SUBI R2,R2,#1
BNEZ R2, LOOP**

Sequential Prefetching

WHAT to Prefetch?
WHEN to Prefetch?

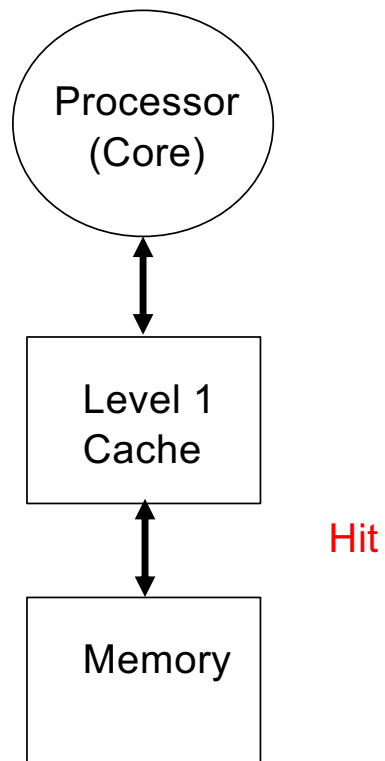


Iteration: 3

→ **LOOP L.D F2,0(R1)**
ADD.D F4,F2,F0
S.D F4,0(R1)
ADDI R1,R1,#8
SUBI R2,R2,#1
BNEZ R2, LOOP

Sequential Prefetching

WHAT to Prefetch?
WHEN to Prefetch?



Iteration: 4

➔ **LOOP L.D F2,0(R1)
ADD.D F4,F2,F0
S.D F4,0(R1)
ADDI R1,R1,#8
SUBI R2,R2,#1
BNEZ R2, LOOP**

Sequential Prefetching

WHAT to Prefetch?

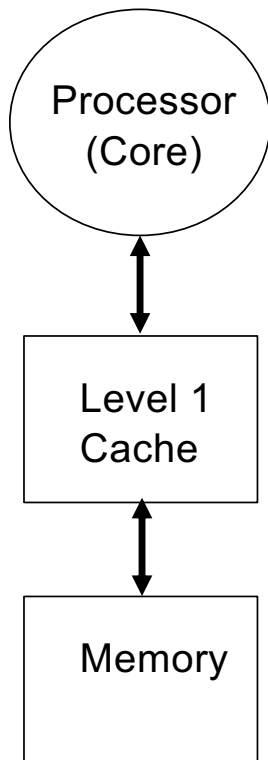
WHEN to Prefetch?

Question:

What will happen in Iterations 6-9?

Answer:

The Load in iterations 6-8 will hit whereas the Load in iteration 9 will miss.



Hit: But would miss without prefetching

Iteration: 5



```
LOOP L.D F2,0(R1)
      ADD.D F4,F2,F0
      S.D F4,0(R1)
      ADDI R1,R1,#8
      SUBI R2,R2,#1
      BNEZ R2, LOOP
```


Stride Prefetching

for (i=0; i<N ; i+=**K**)
A[i] = A[i] + Constant;

K is the stride

```
LOOP:L.D F2,0(R1)
      ADD.D F4,F2,F0
      S.D F4,0(R1)
      ADDI R1,R1,#K*8
      SUBI R2,R2,#1
      BNEZ R2, LOOP
```

WHAT to Prefetch?
WHEN to Prefetch?

Stride Prefetching: Get it to Work

WHAT to Prefetch?

WHEN to Prefetch?

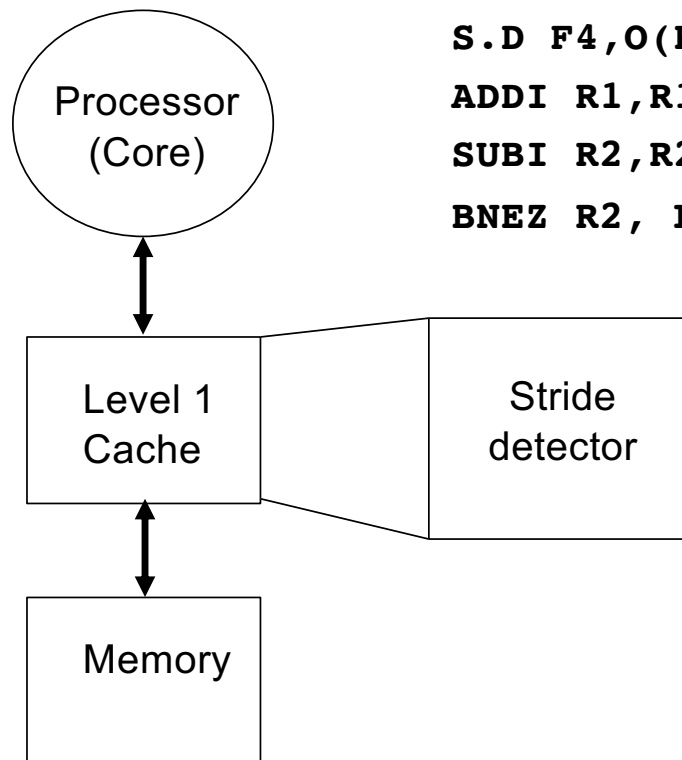
```
LOOP:L.D F2,0(R1)
      ADD.D F4,F2,F0
      S.D F4,0(R1)
      ADDI R1,R1,#K*8
      SUBI R2,R2,#1
      BNEZ R2, LOOP
```

Let $K=2$

Data access sequence: 100, 116, 132, ...

Stride detection:

- Access 1: 100
- Access 2: 116 (stride: $116 - 100 = 16$)
- Access 3: 132 (stride: $132 - 116 = 16$)
- Same stride twice: start prefetching



Example

WHAT to Prefetch?
WHEN to Prefetch?

Question:

Assume the following

- Access sequence (bytes): 100, 110, 120, 130, 140, 150, 160
- Block size 16 bytes
- Stride prefetcher brings prefetched block in zero time

How many misses will be generated.

WHAT to Prefetch?
WHEN to Prefetch?

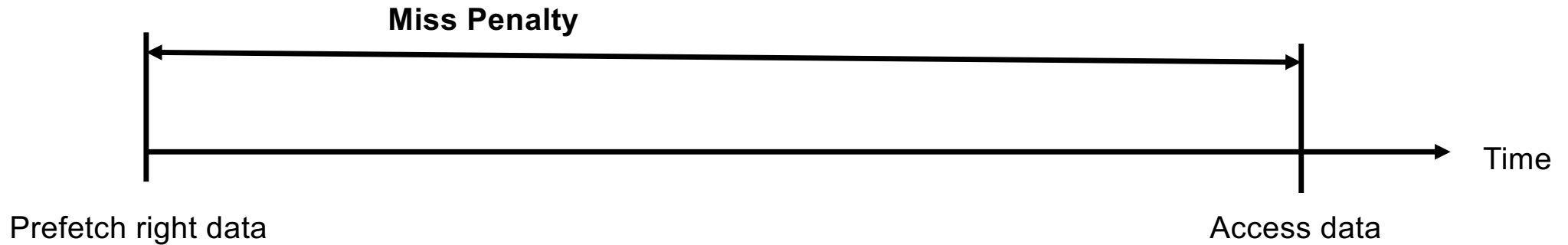
Answer:

- 100, 110,..., 160 comprises 4 blocks (16 bytes each)
- The first three accesses detect the stride. They span the first two blocks
- The last two blocks will be prefetched

So, two misses and two hits.

Prefetch Right Data on Time

WHAT to Prefetch?
WHEN to Prefetch?



Compiler-Controlled Prefetching

WHAT to Prefetch?

WHEN to Prefetch?

ISA Extension: PREF DISP(Rx)

Prefetches block at address $\text{MEM}[\text{DISP} + (\text{Rx})]$
for $(i=0; i<N; i++)$
 $A[i]=A[i] + \text{Constant};$

Question:

Assume the following:

- CPI =1
- Block size = 32 bytes
- Miss Penalty: 28 cycles

What displacement should be used, i.e., what is “?” ?

```
LOOP  PREF ?(R2)
      L.D F2,0(R2)
      ADD.D F4,F2,F0
      S.D F4,0(R2)
      ADDI R2,R2,#8
      SUBI R1,R1,#1
      BNEZ R1, LOOP
```

```
LOOP L.D F2,0(R2)
      PREF ?(R2)
      ADD.D F4,F2,F0
      S.D F4,0(R2)
      ADDI R2,R2,#8
      SUBI R1,R1,#1
      BNEZ R1, LOOP
```



Seven instructions in each iteration:
CPI = 1 yields seven cycles

Miss penalty = 28 cycles
Equivalent to $28/7 = 4$ iterations

In four iterations: $R2 = R2 + 4 \times 8 = 32$

Insert PREF 32(R2)

Answer:

Displacement is 32. Insert PREF 32(R2)