

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Information and Communication Technology Department



- REPORT LABWORK 3-

Name : Phạm Phú Hưng

StudentID : BA12-81

Major : Cyber Security

Practice-1

[I. Configuration Cache -01](#)

[III. Configuration Cache -02](#)

[V. Configuration Cache -03](#)

Practice-2

[I. First simulation](#)

[II. Second simulation](#)

[III. Third simulation](#)

[IV. Fourth simulation](#)

Practice-1

I. Configuration Cache -01

System Parameters:

Address width:	<input type="text" value="8"/> <input type="button" value="▼"/>	bits
Cache size:	<input type="text" value="32"/> <input type="button" value="▼"/>	bytes
Block size:	<input type="radio"/> 2 <input type="radio"/> 4 <input checked="" type="radio"/> 8 bytes	
Associativity:	<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 4 way(s)	
Write Hit:	<input type="button" value="Write back"/> <input type="button" value="▼"/>	
Write Miss:	<input type="button" value="Write-allocate"/> <input type="button" value="▼"/>	
Replacement:	<input type="button" value="Least Recently Used"/> <input type="button" value="▼"/>	

Explain

Analysis of Cache Structure

- The cache seems to use the Least Recently Used (LRU) replacement policy.
 - Cache is divided into sets, and each set has lines (associative).
 - Write-back policy is used for writes.
 - Data is loaded into cache blocks from memory addresses.

II. Address Breakdown and Cache Operations

1) Read from address 0x09

- **Split address:** Set 1, Tag 0
 - **Cache Check:** MISS (Set 1, Tag 0)
 - **Action:** Load block from memory address 0x08 into cache.
 - **Result:** Cache updated, data 0xd0 read.

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2, 4, 8 bytes
- Associativity: 1, 2, 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

Explain Read Addr: 0x 09
 Write Addr: 0x _____, Byte: 0x _____
 Flush

Tag	Index	Offset	Cache Hits	Cache Misses
000	01	001	0	1

Simulation Messages:

Block read into cache from memory at address 0x8.
LRU statuses updated.
Data: 0xd0

Physical Memory

Physical Address	Value
0x00	20 f6 ef ea a2 5e 9f 1a
0x08	a2 d0 4f c4 a0 0c f7 27
0x10	b8 bd 1a ca 35 95 cb 80
0x18	84 3f 02 4f 8e f3 f6 e5
0x20	cd 4a f6 48 1a 6f 7e 63
0x28	e9 36 ae 32 0d 37 bc c9
0x30	93 dc b8 7a 3b 1a b2 0c
0x38	d3 a6 a4 71 e2 23 9c 59
0x40	60 15 68 76 d3 e6 25 be
0x48	a4 a5 db 5e 5f ad 11 2e
0x50	17 1f 95 c4 24 63 d2 62
0x58	b1 7a 44 58 c7 c4 03 81
0x60	54 84 69 8c ab cc ff d9

2) Read from address 0x13

- **Split address:** Set 2, Tag 0
 - **Cache Check:** MISS (Set 2, Tag 0)
 - **Action:** Load block from memory address 0x10 into cache.
 - **Result:** Cache updated, data 0xca read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2 4 8 bytes
- Associativity: 1 2 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x13	Write Addr: 0x13	Byte: 0x	
<input type="button" value="Flush"/>				
Tag	Index	Offset	Cache Hits	Cache Misses
000	10	011	0	2

Simulation Messages:

Block read into cache from memory at address 0x10.
LRU statuses updated.
Data: 0xca

History:

```
R(0x09) = M
> R(0x13) = M
```

Physical Memory

V	D	T	Cache	Data	Physical Memory
Set 0	0	0	- - - - -	- - - - -	0x00 20 f6 ef ea a2 5e 9f 1a
Set 1	1	0	0 a2 d0 4f c4 a0 0c f7 27	1	0x08 a2 d0 4f c4 a0 0c f7 27
Set 2	1	0	0 b8 bd 1a ca 35 95 cb 80	1	0x10 b8 bd 1a ca 35 95 cb 80
Set 3	0	0	- - - - -	- - - - -	0x18 84 3f 02 4f 8e f3 f6 e5
					0x20 cd 4a f6 48 1a 6f 7e 63
					0x28 e9 36 ae 32 0d 37 bc c9
					0x30 93 dc b8 7a 3b 1a b2 0c
					0x38 d3 a6 a4 71 e2 23 9c 59
					0x40 60 15 68 76 3d e6 25 be
					0x48 a4 a5 db be 56 af 1d 2e
					0x50 17 1f 95 c4 24 63 d2 62
					0x58 b1 7a 44 58 c7 c4 03 81
					0x60 54 84 69 8c ab cc 1f d9

3) Read from address 0x21

- **Split address:** Set 0, Tag 1
 - **Cache Check:** MISS (Set 0, Tag 1)
 - **Action:** Load block from memory address 0x20 into cache.
 - **Result:** Cache updated, data 0x4a read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2, 4, 8 bytes
- Associativity: 1, 2, 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x 21
<input type="checkbox"/> Write Addr: 0x _____	Byte: 0x _____
Flush	

Tag	Index	Offset	Cache Hits	Cache Misses
001	00	001	0	3

Simulation Messages:

Block read into cache from memory at address 0x20.
LRU statuses updated.
Data: 0x4a

Physical Memory

V	D	T	C	Cache Data	Physical Address	Data
Set 0	1	0	1	cd 4a f6 48 1a 6f 7e 63	0x00 20	f6 ff ea a2 5e 9f 1a
Set 1	1	0	0	a2 d0 4f c4 a0 0c f7 27	0x08	a2 d0 4f c4 a0 0c f7 27
Set 2	1	0	0	b8 bd 1a ca 35 95 cb 80	0x10	b8 bd 1a ca 35 95 cb 80
Set 3	0	0	-	- - - - - - - - - - - - - - - -	0x18	84 3f 02 4f 8e f3 f6 e5
					0x20	cd 4a f6 48 1a 6f 7e 63
					0x28	e9 36 ae 32 0d 37 bc c9
					0x30	93 dc b8 7a 3b 1a b2 0c
					0x38	d3 a6 a4 71 e2 23 9c 59
					0x40	60 15 68 76 d3 e6 25 be
					0x48	a4 a5 db be 56 af d1 2e
					0x50	17 1f 95 c4 24 d3 d2 62
					0x58	b1 7a 44 58 c7 c4 03 81
					0x60	54 84 69 8c ab cc 1f d9

4) Read from address 0x2a

- **Split address:** Set 1, Tag 1
 - **Cache Check:** MISS (Set 1, Tag 1)
 - **Action:** Load block from memory address 0x28 into cache, replace the least recently used line in Set 1.
 - **Result:** Cache updated, data 0xae read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2, 4, 8 bytes
- Associativity: 1, 2, 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x 2a	Write Addr: 0x _____, Byte: 0x _____		
Flush				
Tag	Index	Offset	Cache Hits	Cache Misses
001	01	010	0	4

Simulation Messages:

Block read into cache from memory at address 0x28.
LRU statuses updated.
Data: 0x05

Physical Memory:

```
m = 8, C = 32      V D T Cache Data
K = 8, E = 1        Set 0 1 0 1 cd 4a f6 48 1a 6f 7e 63 1  0x00 20 f6 ef ea a2 5e 9f 1a
Write back          Set 1 1 0 1 e9 36 ae 32 0d 37 bc c9 1  0x08 a2 d0 4f c4 a0 0c f7 27
Write-allocate      Set 2 1 0 0 b8 bd 1a ca 35 95 cb 80 1  0x10 b8 bd 1a ca 35 95 cb 80
Eviction: LRU      Set 3 0 0 - - - - - - - - - - - - - - - - - 1  0x18 84 3f 02 4f 8e f3 f6 e5
                                         0x20 cd 4a f6 48 1a 6f 7e 63
                                         0x28 e9 36 ae 32 0d 37 bc c9
                                         0x30 93 dc b8 7a 3b 1a b2 0c
                                         0x38 d3 a6 a4 71 e2 23 9c 59
                                         0x40 60 15 68 76 d3 e6 25 be
                                         0x48 a4 a5 db be 56 af d1 2e
                                         0x50 17 1f 95 c4 24 63 d2 62
                                         0x58 b1 7a 44 58 c7 c4 03 81
                                         0x60 54 84 69 8c ab cc 1f d9
```

History:

```
R(0x09) = M
R(0x11) = M
R(0x21) = M
R(0x2a) = M
>
```

5) Read from address 0x15

- Split address:** Set 2, Tag 0
- Cache Check:** HIT (Set 2, Tag 0)
- Action:** Update LRU.
- Result:** Data 0x95 read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2, 4, 8 bytes
- Associativity: 1, 2, 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x 15	Write Addr: 0x _____, Byte: 0x _____		
Flush				
Tag	Index	Offset	Cache Hits	Cache Misses
000	10	101	1	4

Simulation Messages:

Looking for Tag 0... HIT in Line 0!
LRU statuses updated.
Data: 0x95

Physical Memory:

```
m = 8, C = 32      V D T Cache Data
K = 8, E = 1        Set 0 1 0 1 cd 4a f6 48 1a 6f 7e 63 1  0x00 20 f6 ef ea a2 5e 9f 1a
Write back          Set 1 1 0 1 e9 36 ae 32 0d 37 bc c9 1  0x08 a2 d0 4f c4 a0 0c f7 27
Write-allocate      Set 2 1 0 0 b8 bd 1a ca 35 95 cb 80 1  0x10 b8 bd 1a ca 35 95 cb 80
Eviction: LRU      Set 3 0 0 - - - - - - - - - - - - - - - - - 1  0x18 84 3f 02 4f 8e f3 f6 e5
                                         0x20 cd 4a f6 48 1a 6f 7e 63
                                         0x28 e9 36 ae 32 0d 37 bc c9
                                         0x30 93 dc b8 7a 3b 1a b2 0c
                                         0x38 d3 a6 a4 71 e2 23 9c 59
                                         0x40 60 15 68 76 d3 e6 25 be
                                         0x48 a4 a5 db be 56 af d1 2e
                                         0x50 17 1f 95 c4 24 63 d2 62
                                         0x58 b1 7a 44 58 c7 c4 03 81
                                         0x60 54 84 69 8c ab cc 1f d9
```

History:

```
R(0x09) = M
R(0x11) = M
R(0x21) = M
R(0x2a) = M
R(0x15) = H
>
```

6) Write to address 0x11

- Split address:** Set 2, Tag 0
- Cache Check:** HIT (Set 2, Tag 0)
- Action:** Update LRU, set Dirty bit.
- Result:** Data written.

351 Cache Simulator

System Parameters:

Address width: 8 bits

Cache size: 32 bytes

Block size: 2 4 8 bytes

Associativity: 1 2 4 way(s)

Write Hit: Write back

Write Miss: Write-allocate

Replacement: Least Recently Used

[Reset System](#)

[Explain](#)

Manual Memory Access:

Read Addr: 0x 11, Byte: 0x 11

Explain Write Flush

Tag	Index	Offset	Cache Hits	Cache Misses
000	10	001	2	4

Simulation Messages:

Checking Set 2
Looking for Tag 0... HIT in Line 0!
LRU step 1 updated.
Write back: set Dirty bit.

History:

```
R(0x09) = M
R(0x09) = M
R(0x21) = M
R(0x2a) = M
R(0x11) = H
> W(0x11, 0x11) = H
```

Load ↑ ↓

m = 8, C = 32	V D T Cache Data	Physical Memory
K = 8, E = 1	Set 0 1 0 1 cd 4a f6 48 1a 6f 7e 63 1	0x00 20 f6 fe ea a2 5e 9f 1a
Write back	Set 1 1 0 1 e9 36 ae 32 0d 37 bc c9 1	0x08 a2 d0 4f c4 a0 0c f7 27
Write-allocate	Set 2 1 1 0 b8 11 1a ca 35 95 cb 80 1	0x10 b8 bd 1a ca 35 95 cb 80
Eviction: LRU	Set 3 0 0 ----- 1	0x18 84 3f 02 4f 8e f3 f6 e5
		0x20 cd 4a f6 48 1a 6f 7e 63
		0x28 e9 36 ae 32 0d 37 bc c9
		0x30 93 dc b8 7a 3b 1a b2 0c
		0x38 d3 a6 a4 71 e2 23 9c 59
		0x40 60 15 68 76 d3 e6 25 be
		0x48 a4 a5 db be 56 af d1 2e
		0x50 17 1f 95 c4 24 63 d2 62
		0x58 b1 7a 44 58 c7 c4 03 81
		0x60 54 84 69 8c ab cc 1f d9

7) Read from address 0x33

- **Split address:** Set 2, Tag 1
 - **Cache Check:** MISS (Set 2, Tag 1)
 - **Action:** Write back dirty block to memory, load block from memory address 0x30 into cache, replace the least recently used line in Set 2.
 - **Result:** Cache updated, data 0x7a read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2 bytes
- Associativity: 1 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x33			
<input type="checkbox"/> Write	Addr: 0x <input type="text"/> , Byte: 0x <input type="text"/>			
<input type="button" value="Flush"/>				
Tag	Index	Offset	Cache Hits	Cache Misses
001	10	011	2	5

Simulation Messages:

Block read into cache from memory at address 0x30.
LRU statuses updated.
Data: 0x7a

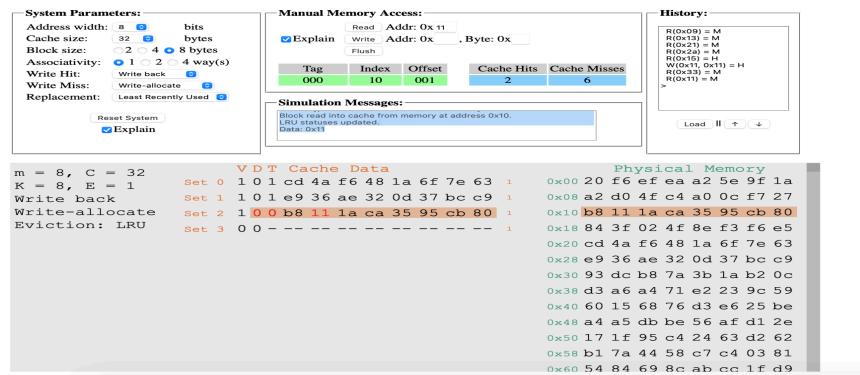
Physical Memory

V	D	T	Cache Data	Physical Memory
Set 0	1 0 1	cd 4a f6 48 1a 6f 7e 63	i	0x00 20 f6 ef ea a2 5e 9f 1a
Set 1	1 0 1	e9 36 ae 32 0d 37 bc c9	i	0x08 a2 d0 4f fc a4 0c f7 27
Set 2	1 0 1	93 dc b8 7a 3b 1a b2 0c	i	0x10 b8 11 1a ca 35 95 cb 80
Set 3	0 0	---	i	0x18 84 3f 02 4f 8e f3 f6 e5 0x20 cd 4a f6 48 1a 6f 7e 63 0x28 e9 36 ae 32 0d 37 bc c9 0x30 93 dc b8 7a 3b 1a b2 0c 0x38 d3 a6 a4 71 e2 23 9c 59 0x40 60 15 68 76 d3 e6 25 be 0x48 a4 a5 db be 56 af d1 2e 0x50 17 1f 95 c4 24 63 d2 62 0x58 b1 7a 44 58 c7 c4 03 81 0x60 54 84 69 8c ab cc 1f d9

8) Read from address 0x11

- **Split address:** Set 2, Tag 0
 - **Cache Check:** MISS (Set 2, Tag 0)
 - **Action:** Load block from memory address 0x10 into cache, replace the least recently used line in Set 2.
 - **Result:** Cache updated, data 0x11 read.

351 Cache Simulator



Cache Performance

- Cache Hits:** 2 (Addresses 0x15 and 0x11 during the write)
- Cache Misses:** 6 (Addresses 0x09, 0x13, 0x21, 0x2A, 0x33, 0x11)
- Write Policy:** Write-back, with Dirty bit set for write operations.
- LRU Replacement:** Least Recently Used policy is followed, ensuring that the least accessed cache line is replaced upon a miss.

Analysis of Cache Block Loading

- Cache blocks are loaded from memory addresses based on the set and tag derived from the address.
- For each miss, a new block is read from memory into the appropriate set.
- Write-backs occur when a dirty block is replaced during a miss.

III. Configuration Cache -02

System Parameters:

Address width:	8	bits	
Cache size:	32	bytes	
Block size:	<input type="radio"/> 2	<input type="radio"/> 4	<input checked="" type="radio"/> 8 bytes
Associativity:	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 4 way(s)
Write Hit:	Write back		
Write Miss:	Write-allocate		
Replacement:	Least Recently Used		

Explain

Analysis of Cache Structure

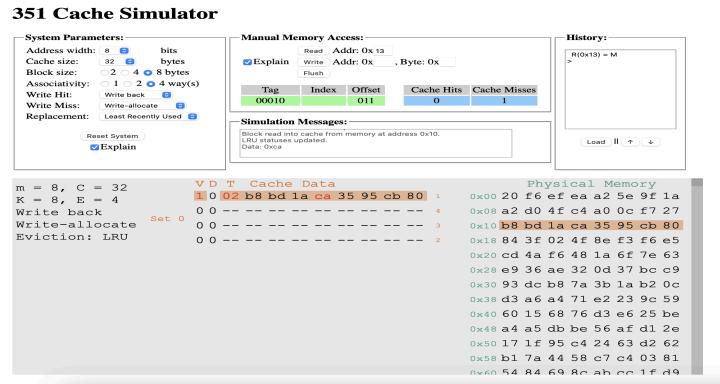
- Associativity:** The modified cache structure seems to be a fully associative cache, with all memory accesses directed to a single set (Set 0), and multiple lines within the set (line 0, line 1, line 2, etc.).
- Write Policy:** Remains write-back with a dirty bit.

- **Replacement Policy:** Still LRU.

IV. Address Breakdown and Cache Operations

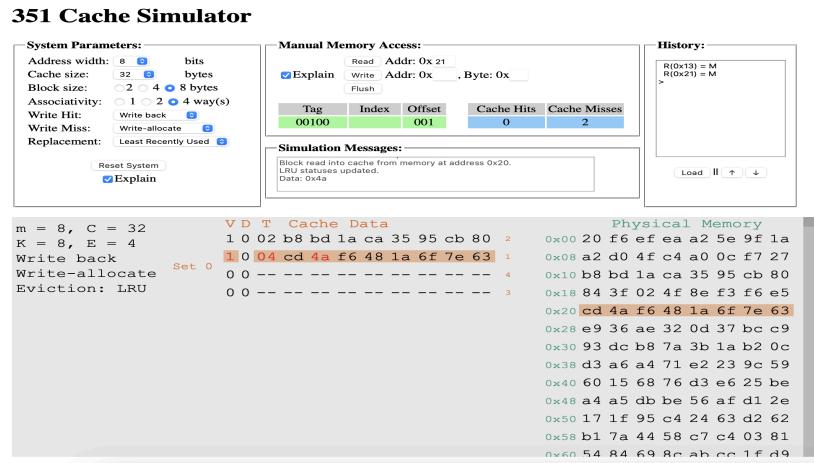
1) Read from address 0x13

- **Split address:** Set 0, Tag 02
- **Cache Check:** MISS (Set 0, Tag 02)
- **Action:** Load block from memory address 0x10 into cache (line 0).
- **Result:** Cache updated, data 0xca read.



2) Read from address 0x21

- **Split address:** Set 0, Tag 04
- **Cache Check:** MISS (Set 0, Tag 04)
- **Action:** Load block from memory address 0x20 into cache (line 1).
- **Result:** Cache updated, data 0xa4 read.



3) Read from address 0x2A

- **Split address:** Set 0, Tag 05
- **Cache Check:** MISS (Set 0, Tag 05)
- **Action:** Load block from memory address 0x28 into cache (line 2).
- **Result:** Cache updated, data 0xae read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2 4 8 bytes
- Associativity: 1 2 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

Read	Addr: 0x 2a	Write	Addr: 0x , Byte: 0x	Flush
<input checked="" type="checkbox"/> Explain				

History:

```
R(0x13) = M
R(0x21) = M
R(0x2a) = M
>
```

Simulation Messages:

```
Block read into cache from memory at address 0x28.
LRU statuses updated.
Data: 0xae
```

Physical Memory:

V	D	T	Cache Data	Physical Memory
1	0	02	b8 bd 1a ca 35 95 cb 80	0x00 20 f6 ef ea a2 5e 9f 1a
1	0	04	cd 4a f6 48 1a 6f 7e 63	0x08 a2 d0 4f c4 a0 0c f7 27
1	0	05	e9 36 ae 32 0d 37 bc c9	0x10 b8 bd 1a ca 35 95 cb 80
0	0	--	--	0x18 84 3f 02 4f 8e f3 f6 e5
0	0	--	--	0x20 cd 4a f6 48 1a 6f 7e 63
0	28	e9 36 ae 32 0d 37 bc c9		0x28 e9 36 ae 32 0d 37 bc c9
0	30	93 dc b8 7a 3b 1a b2 0c		0x30 93 dc b8 7a 3b 1a b2 0c
0	38	d3 a6 a4 71 e2 23 9c 59		0x38 d3 a6 a4 71 e2 23 9c 59
0	40	60 15 68 76 d3 e6 25 be		0x40 60 15 68 76 d3 e6 25 be
0	48	a4 a5 db be 56 af d1 2e		0x48 a4 a5 db be 56 af d1 2e
0	50	17 1f 95 c4 24 63 d2 62		0x50 17 1f 95 c4 24 63 d2 62
0	58	b1 7a 44 58 c7 c4 03 81		0x58 b1 7a 44 58 c7 c4 03 81
0	60	54 84 69 8c ab cc 1f d9		0x60 54 84 69 8c ab cc 1f d9

4) Read from address 0x15

- Split address:** Set 0, Tag 02
- Cache Check:** HIT (Set 0, Tag 02)
- Action:** Update LRU.
- Result:** Data 0x95 read.

351 Cache Simulator

System Parameters:

- Address width: 8 bits
- Cache size: 32 bytes
- Block size: 2 4 8 bytes
- Associativity: 1 2 4 way(s)
- Write Hit: Write back
- Write Miss: Write-allocate
- Replacement: Least Recently Used

Manual Memory Access:

Read	Addr: 0x 15	Write	Addr: 0x , Byte: 0x	Flush
<input checked="" type="checkbox"/> Explain				

History:

```
R(0x13) = M
R(0x21) = M
R(0x2a) = M
R(0x15) = H
>
```

Simulation Messages:

```
Looking for Tag 02... HIT in Line 0!
LRU statuses updated.
Data: 0x95
```

Physical Memory:

V	D	T	Cache Data	Physical Memory
1	0	02	b8 bd 1a ca 35 95 cb 80	0x00 20 f6 ef ea a2 5e 9f 1a
1	0	04	cd 4a f6 48 1a 6f 7e 63	0x08 a2 d0 4f c4 a0 0c f7 27
1	0	05	e9 36 ae 32 0d 37 bc c9	0x10 b8 bd 1a ca 35 95 cb 80
0	0	--	--	0x18 84 3f 02 4f 8e f3 f6 e5
0	0	--	--	0x20 cd 4a f6 48 1a 6f 7e 63
0	28	e9 36 ae 32 0d 37 bc c9		0x28 e9 36 ae 32 0d 37 bc c9
0	30	93 dc b8 7a 3b 1a b2 0c		0x30 93 dc b8 7a 3b 1a b2 0c
0	38	d3 a6 a4 71 e2 23 9c 59		0x38 d3 a6 a4 71 e2 23 9c 59
0	40	60 15 68 76 d3 e6 25 be		0x40 60 15 68 76 d3 e6 25 be
0	48	a4 a5 db be 56 af d1 2e		0x48 a4 a5 db be 56 af d1 2e
0	50	17 1f 95 c4 24 63 d2 62		0x50 17 1f 95 c4 24 63 d2 62
0	58	b1 7a 44 58 c7 c4 03 81		0x58 b1 7a 44 58 c7 c4 03 81
0	60	54 84 69 8c ab cc 1f d9		0x60 54 84 69 8c ab cc 1f d9

5) Write to address 0x11

- Split address:** Set 0, Tag 02
- Cache Check:** HIT (Set 0, Tag 02)
- Action:** Update LRU, set Dirty bit.
- Result:** Data written.

351 Cache Simulator

System Parameters:

Address width:	8 bits
Cache size:	32 bytes
Block size:	2 4 8 bytes
Associativity:	1 2 4 way(s)
Write Hit:	Write back
Write Miss:	Write-allocate
Replacement:	Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x	Write Addr: 0x	Byte: 0x	Flush
Tag	Index	Offset	Cache Hits	Cache Misses
00010	001		2	3

Simulation Messages:

Checking Set 0
 LRU statuses updated.
 Write back: set Dirty bit.

History:

R(0x10) = M
 R(0x21) = M
 R(0x22) = M
 R(0x10) = M
 W(0x11, 0x11) = H

Physical Memory

V	D	T	Cache Data	Physical Memory
1	1	02	b8 11 1a ca 35 95 cb 80	0x00 20 f6 ef ea a2 5e 9f 1a
1	0	04	cd 4a f6 48 1a 6f 7e 63	0x08 a2 d0 4f c4 a0 0c f7 27
1	0	05	e9 36 ae 32 0d 37 bc c9	0x10 b8 bd 1a ca 35 95 cb 80
0	0	--	--	0x18 84 3f 02 4f 8e f3 f6 e5
				0x20 cd 4a f6 48 1a 6f 7e 63
				0x28 e9 36 ae 32 0d 37 bc c9
				0x30 93 dc b8 7a 3b 1a b2 0c
				0x38 d3 a6 a4 71 e2 23 9c 59
				0x40 60 15 68 76 d3 e6 25 be
				0x48 a4 a5 db be 56 af d1 2e
				0x50 17 1f 95 c4 24 63 d2 62
				0x58 b1 7a 44 58 c7 c4 03 81
				0x60 54 84 69 8c ab cc 1f d9

6) Read from address 0x33

- Split address:** Set 0, Tag 06
- Cache Check:** MISS (Set 0, Tag 06)
- Action:** Load block from memory address 0x30 into cache (line 3).
- Result:** Cache updated, data 0x7a read.

351 Cache Simulator

System Parameters:

Address width:	8 bits
Cache size:	32 bytes
Block size:	2 4 8 bytes
Associativity:	1 2 4 way(s)
Write Hit:	Write back
Write Miss:	Write-allocate
Replacement:	Least Recently Used

Manual Memory Access:

<input checked="" type="checkbox"/> Explain	Read Addr: 0x 33	Write Addr: 0x	Byte: 0x	Flush
Tag	Index	Offset	Cache Hits	Cache Misses
00110	011		2	4

Simulation Messages:

Block read into cache from memory at address 0x30.
 LRU statuses updated.
 Data: 0x7a

History:

R(0x10) = M
 R(0x21) = M
 R(0x22) = M
 R(0x10) = M
 W(0x11, 0x11) = H
 R(0x33) = M

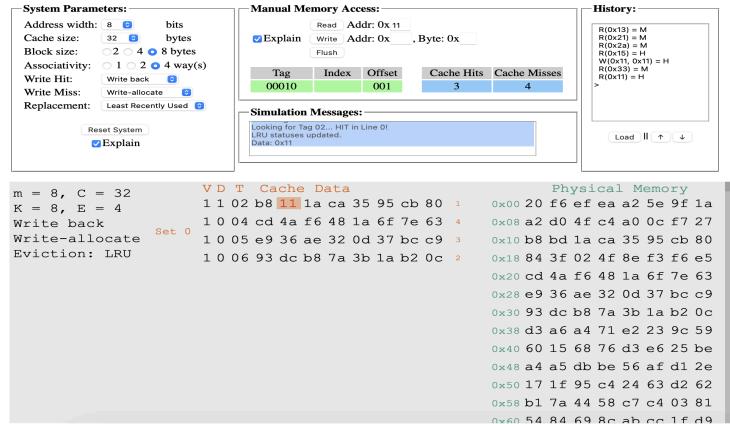
Physical Memory

V	D	T	Cache Data	Physical Memory
1	1	02	b8 11 1a ca 35 95 cb 80	0x00 20 f6 ef ea a2 5e 9f 1a
1	0	04	cd 4a f6 48 1a 6f 7e 63	0x08 a2 d0 4f c4 a0 0c f7 27
1	0	05	e9 36 ae 32 0d 37 bc c9	0x10 b8 bd 1a ca 35 95 cb 80
1	0	06	93 dc b8 7a 3b 1a b2 0c	0x18 84 3f 02 4f 8e f3 f6 e5
0	0	--	--	0x20 cd 4a f6 48 1a 6f 7e 63
				0x28 e9 36 ae 32 0d 37 bc c9
				0x30 93 dc b8 7a 3b 1a b2 0c
				0x38 d3 a6 a4 71 e2 23 9c 59
				0x40 60 15 68 76 d3 e6 25 be
				0x48 a4 a5 db be 56 af d1 2e
				0x50 17 1f 95 c4 24 63 d2 62
				0x58 b1 7a 44 58 c7 c4 03 81
				0x60 54 84 69 8c ab cc 1f d9

7) Read from address 0x11

- Split address:** Set 0, Tag 02
- Cache Check:** HIT (Set 0, Tag 02)
- Action:** Update LRU.
- Result:** Data 0x11 read.

351 Cache Simulator



Performance Comparison

First Configuration

- Cache Hits:** 2 (Addresses 0x15 and 0x11 during the write)
- Cache Misses:** 6 (Addresses 0x09, 0x13, 0x21, 0x2A, 0x33, 0x11)

Modified Configuration

- Cache Hits:** 3 (Addresses 0x15, 0x11 during write, and 0x11)
- Cache Misses:** 4 (Addresses 0x13, 0x21, 0x2A, 0x33)

Analysis

Cache Hits and Misses

- The modified configuration has fewer cache misses compared to the initial configuration due to its fully associative nature, which allows for more flexible placement and replacement of blocks.

Cache Block Loading

- Blocks are loaded into different lines within Set 0, taking advantage of the fully associative cache to reduce conflicts.
- Blocks are placed in lines based on the LRU policy, which ensures that the least recently used block is replaced.

Write Policy

- The write-back policy with a dirty bit remains the same. In both configurations, data is written back to memory only when a dirty block is replaced.
- During a write operation, the cache updates the LRU status and sets the dirty bit, which indicates the block has been modified.

V.Configuration Cache -03

System Parameters:

Address width:	8 <input checked="" type="radio"/>	bits
Cache size:	32 <input checked="" type="radio"/>	bytes
Block size:	<input type="radio"/> 2 <input type="radio"/> 4 <input checked="" type="radio"/> 8 bytes	
Associativity:	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 4 way(s)	
Write Hit:	Write back <input checked="" type="radio"/>	
Write Miss:	Write-allocate <input type="radio"/>	
Replacement:	Least Recently Used <input type="radio"/>	

Explain

Analysis of Cache Structure

- **Cache Size:** 32 bytes
- **Associativity:** 2-way set associative
- **Write Policy:** Write-back with a dirty bit
- **Replacement Policy:** Least Recently Used (LRU)

VI. Address Breakdown and Cache Operations

1) Read from address 0x13

- **Split address:** Set 0, Tag 02
- **Cache Check:** MISS (Set 0, Tag 02)
- **Action:** Load block from memory address 0x10 into cache (line 0).
- **Result:** Cache updated, data 0xca read.

351 Cache Simulator

System Parameters:

Address width:	8 <input checked="" type="radio"/>	bits
Cache size:	32 <input checked="" type="radio"/>	bytes
Block size:	<input type="radio"/> 2 <input type="radio"/> 4 <input checked="" type="radio"/> 8 bytes	
Associativity:	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 4 way(s)	
Write Hit:	Write back <input checked="" type="radio"/>	
Write Miss:	Write-allocate <input type="radio"/>	
Replacement:	Least Recently Used <input type="radio"/>	

Manual Memory Access:

<input type="button" value="Read Addr: 0x13"/>	<input type="button" value="Write Addr: 0x_____ , Byte: 0x_____"/>			
<input checked="" type="checkbox"/> Explain	<input type="button" value="Flush"/>			
Tag	Index	Offset	Cache Hits	Cache Misses
00010	011		0	1

History:

```
R(0x13) = M
>
```

Simulation Messages:

```
Read: 0x13
Split address into TIO breakdown.
Checking Set 0
Looking for Tag 0... MISS!
```

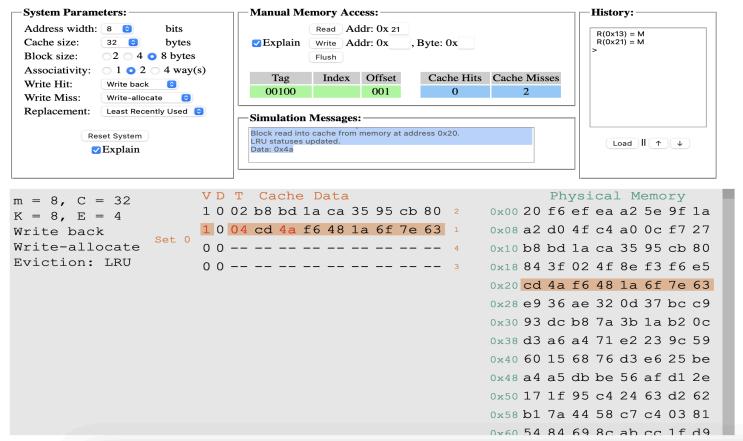
Physical Memory:

V	D	T	Cache Data	Physical Memory
1	0	02	b8 bd 1a ca 35 95 cb 80	0x00 20 f6 ef ea a2 5e 9f 1a
0	0	---	-----	0x08 a2 d0 4f c4 a0 0c f7 27
0	0	---	-----	0x10 b8 bd 1a ca 35 95 cb 80
0	0	---	-----	0x18 84 3f 02 4f 8e f3 f6 e5
0	0	---	-----	0x20 cd 4a f6 48 1a 6f 7e 63
0	0	---	-----	0x28 e9 36 ae 32 0d 37 bc c9
0	0	---	-----	0x30 93 dc b8 7a 3b 1a b2 0c
0	0	---	-----	0x38 d3 a6 a4 71 e2 23 9c 59
0	0	---	-----	0x40 60 15 68 76 d3 e6 25 be
0	0	---	-----	0x48 a4 a5 db be 56 af d1 2e
0	0	---	-----	0x50 17 1f 95 c4 24 63 d2 62
0	0	---	-----	0x58 b1 7a 44 58 c7 c4 03 81
0	0	---	-----	0x60 54 84 69 8c ab cc 1f d9

2) Read from address 0x21

- **Split address:** Set 0, Tag 04
- **Cache Check:** MISS (Set 0, Tag 04)
- **Action:** Load block from memory address 0x20 into cache (line 1).
- **Result:** Cache updated, data 0x4a read.

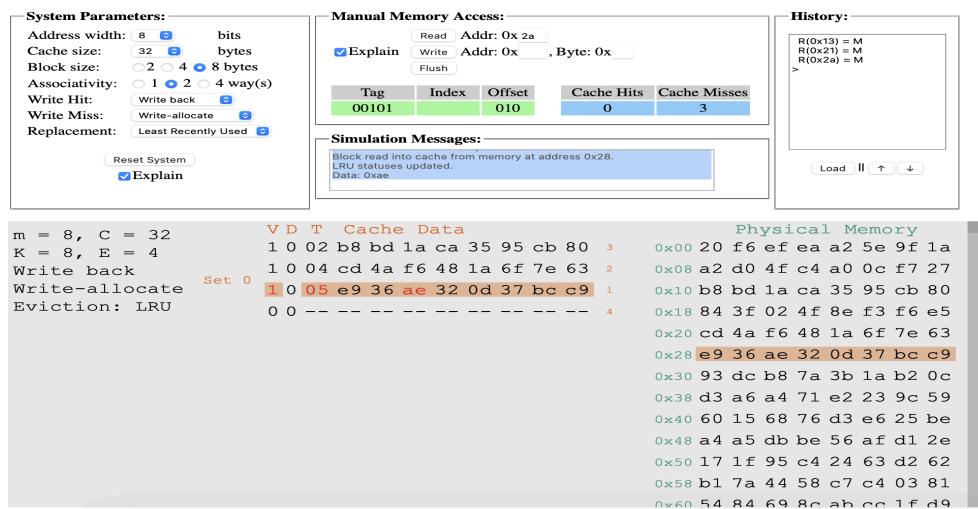
351 Cache Simulator



3) Read from address 0x2A

- **Split address:** Set 0, Tag 05
- **Cache Check:** MISS (Set 0, Tag 05)
- **Action:** Load block from memory address 0x28 into cache (line 2).
- **Result:** Cache updated, data 0xae read.

351 Cache Simulator



4) Read from address 0x15

- **Split address:** Set 0, Tag 02
- **Cache Check:** HIT (Set 0, Tag 02)
- **Action:** Update LRU.
- **Result:** Data 0x95 read.

351 Cache Simulator

The screenshot shows the 351 Cache Simulator interface. In the 'System Parameters' section, the cache is configured with 8 bits for address width, 32 bytes for cache size, 8 bytes for block size, and 4-way associativity. The 'Write Back' option is selected. The 'Manual Memory Access' section shows a 'Write' operation to address 0x15, with the tag 00010 and index 101. The 'Cache Data' table shows the entry for line 0 (Set 0) has been updated. The 'Physical Memory' section displays the full memory dump. The 'History' section shows previous operations: R(0x13)=M, R(0x21)=M, R(0x2a)=M, R(0x10)=H, and W(0x10, 0x11)=H. The 'Simulation Messages' box indicates a hit in line 0.

5) Write to address 0x11

- **Split address:** Set 0, Tag 02
- **Cache Check:** HIT (Set 0, Tag 02)
- **Action:** Update LRU, set Dirty bit.
- **Result:** Data written.

351 Cache Simulator

The screenshot shows the 351 Cache Simulator interface. In the 'System Parameters' section, the cache is configured with 8 bits for address width, 32 bytes for cache size, 8 bytes for block size, and 4-way associativity. The 'Write Back' option is selected. The 'Manual Memory Access' section shows a 'Write' operation to address 0x11, with the tag 00010 and index 001. The 'Cache Data' table shows the entry for line 0 (Set 0) has been updated. The 'Physical Memory' section displays the full memory dump. The 'History' section shows previous operations: R(0x13)=M, R(0x21)=M, R(0x2a)=M, R(0x10)=H, and W(0x11, 0x11)=H. The 'Simulation Messages' box indicates a hit in line 0.

6) Read from address 0x33

- **Split address:** Set 0, Tag 06
- **Cache Check:** MISS (Set 0, Tag 06)
- **Action:** Load block from memory address 0x30 into cache (line 3).
- **Result:** Cache updated, data 0x7a read.

351 Cache Simulator

The screenshot shows the 351 Cache Simulator interface. On the left, 'System Parameters' are set to: Address width: 8 bits, Cache size: 32 bytes, Block size: 4 bytes, Associativity: 4 way(s), Write Hit: Write back, Write Miss: Write-allocate, Replacement: Least Recently Used. Under 'Manual Memory Access', a read operation is performed at address 0x33. The cache state shows a hit for tag 011 (index 011, offset 0) with a value of 0x11. The 'History' panel shows previous operations: R(0x13) = M, R(0x21) = M, R(0x2A) = M, R(0x15) = H, W(0x11, 0x11) = H, R(0x33) = M. The 'Physical Memory' dump shows the memory layout with address 0x33 highlighted.

7) Read from address 0x11

- **Split address:** Set 0, Tag 02
- **Cache Check:** HIT (Set 0, Tag 02)
- **Action:** Update LRU.
- **Result:** Data 0x11 read.

351 Cache Simulator

The screenshot shows the 351 Cache Simulator interface. On the left, 'System Parameters' are set to: Address width: 8 bits, Cache size: 32 bytes, Block size: 4 bytes, Associativity: 4 way(s), Write Hit: Write back, Write Miss: Write-allocate, Replacement: Least Recently Used. Under 'Manual Memory Access', a read operation is performed at address 0x11. The cache state shows a hit for tag 011 (index 011, offset 0) with a value of 0x11. The 'History' panel shows previous operations: R(0x13) = M, R(0x21) = M, R(0x2A) = M, R(0x15) = H, W(0x11, 0x11) = H, R(0x33) = M. The 'Physical Memory' dump shows the memory layout with address 0x11 highlighted.

Performance Comparison

New Configuration

- **Cache Hits:** 3 (Addresses 0x15, 0x11 during write, and 0x11)
- **Cache Misses:** 4 (Addresses 0x13, 0x21, 0x2A, 0x33)

Previous Configurations

- **Initial Configuration:**
 - **Cache Hits:** 2
 - **Cache Misses:** 6
- **Modified Fully Associative Configuration:**
 - **Cache Hits:** 3

- **Cache Misses:** 4

Analysis

Cache Hits and Misses

- The new 2-way set associative cache configuration provides the same number of hits and misses as the fully associative configuration.
- This configuration also reduces cache misses compared to the initial direct-mapped cache configuration.

Block Loading

- Blocks are loaded into different sets and lines based on the set and tag derived from the address.
- Blocks are placed in lines within a set according to the LRU policy, which ensures that the least recently used block is replaced.

Write Policy

- The write-back policy with a dirty bit is retained. Data is written back to memory only when a dirty block is replaced.
- During a write operation, the cache updates the LRU status and sets the dirty bit.

Practice-2

I. First simulation

Cache Configuration

Configuration of the cache simulator
Clic on the blue lines to open the configuration zones

General configuration

Show the access trace : Oui ; Non

Configuration of the number of levels of caches : 1

Cache level 1 unified : Yes ; No
Cache level 2 unified : yes ; No
Cache level 3 unified : Yes ; No

Main Memory size : 1kmots

Main memory time access : 100 ns

Memory access trace : Choose File MemoryTrac...ample.mem

Simulate

Level 1 Cache Configuration

Data cache or unified cache :
Cache size : 128 mots
Block size : 16 mots
Cache access time : 10 ns
Associativity : Direct associativity ; Completely associative ; n Ways associative nb ways
Replacement policy : FIFO ; LRU ; Random
Write policy : Write back ; Write through
Allocation policy for write : Write allocate ; Write no allocate

Instruction cache :
Cache size : 128 mots
Block size : 16 mots
Cache access time : 10 ns
Associativity : Direct associativity ; Completely associative ; n Ways associative nb ways
Replacement policy : FIFO ; LRU ; Random

```
--Dinero IV cache simulator, version 7
--Written by Jan Edler and Mark D. Hill
--Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
--All rights reserved.
--Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
--See -copyright option for details.

--Summary of options (-help option gives usage information).

-l1-usize 128
-l1-uhsiz 16
-l1-uhsiz 16
-l1-uassoc 8
-l1-uassoc 8
-l1-l1l 1
-l1-ufetch d
-l1-ualloc a
-l1-uwbact a
-l1-ahsize 8
-l1-flushcount 0
-maxcount 0
-stat-interval 0
-informat d
-on-trigger 0x0
-off-trigger 0x0

--Simulation begins.
--Simulation complete.
l1-ucache
Metrics      Total      Instrn      Data      Read      Write      Misc
-----      -----
Demand Fetches    1000       351       649       321       328        0
Fraction of total   1.0000     0.3510     0.6490     0.3210     0.3280     0.0000

Demand Misses     873        311       562       277       285        0
Demand miss rate  0.8730     0.8860     0.8659     0.8629     0.8689     0.0000

Multi-block refs      0
Bytes From Memory  13968
( / Demand Fetches) 13.9680
Bytes To Memory    4912
( / Demand Writes) 14.9156
Total Bytes r/w Mem 18880
( / Demand Fetches) 18.8800

--Execution complete.
```

Analysis

- **Demand Fetches:** 1000 total, with 351 for instructions and 649 for data.
- **Fraction of Total Fetches:** 35.10% for instructions and 64.90% for data.
- **Demand Misses:** 873 total, with 311 for instructions and 562 for data.
- **Demand Miss Rate:** 87.30% overall, with 88.60% for instructions and 86.59% for data.
- **Read/Write Breakdown:** 321 reads and 328 writes for data accesses.
- **Bytes Transferred:**
 - **From Memory:** 13,968 bytes.
 - **To Memory:** 4,912 bytes.

- **Total Bytes Read/Written:** 18,880 bytes.

Conclusions

- **High Demand Miss Rate:**
 - The overall demand miss rate of 87.30% is quite high.
 - Both instruction and data caches have high miss rates (88.60% for instructions and 86.59% for data).
- **Cache Effectiveness:**
 - The high miss rates indicate that the cache configuration may not be optimal for the given workload.
 - Possible reasons for the high miss rates could be due to the random access pattern, which reduces the cache's ability to effectively reuse cached data.
- **Memory Traffic:**
 - A significant amount of data is transferred to and from memory.
 - This suggests that the cache misses are leading to frequent memory accesses, which can significantly impact performance due to the higher latency of memory accesses compared to cache accesses.

II. Second simulation

```

---Dinero IV cache simulator, version 7
---Written by Jan Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-l1-usize 128
-l1-ubsize 4
-l1-uassoc 4
-l1-uassoc 1
-l1-urepl 1
-l1-ufetch d
-l1-uwalloc a
-l1-uwback a
-skipcount 0
-flushcount 0
-maxcount 0
-stat-interval 0
-informat d
-on-trigger 0x0
-off-trigger 0x0

---Simulation begins.
---Simulation complete.

l1-ucache
Metrics          Total      Instrn      Data      Read      Write      Misc
-----|-----|-----|-----|-----|-----|-----|
Demand Fetches   1000       351        649       321       328        0
Fraction of total 1.0000    0.3510     0.6490    0.3210    0.3280    0.0000

Demand Misses    874        309        565       281       284        0
Demand miss rate 0.8740    0.8803     0.8706    0.8754    0.8659    0.0000

Multi-block refs 0
Bytes From Memory 2360
( / Demand Fetches) 2.3600
Bytes To Memory 1240
( / Demand Writes) 3.7805
Total Bytes r/w Mem 3600
( / Demand Fetches) 3.6000

---Execution complete.

```

Simulation Results

- **Demand Fetches:** 1000 total (351 for instructions and 649 for data).
- **Demand Misses:** 874 total (309 for instructions and 565 for data).
- **Demand Miss Rate:** 87.40% overall (88.60% for instructions and 86.59% for data).

Calculations

1. **Number of Misses:**
 - Total misses: 874.
 - Instruction misses: 309.
 - Data misses: 565.
2. **Number of Hits:**
 - Total hits = Total fetches - Total misses = $1000 - 874 = 126$.
3. **Miss Time:**
 - Miss time = Cache access time + Memory access time = $10 \text{ ns} + 100 \text{ ns} = 110 \text{ ns}$.
4. **Hit Time:**
 - Hit time = Cache access time = 10 ns.
5. **Total Time of Memory Access:**
 - Total time = (Number of hits * Hit time) + (Number of misses * Miss time).
 - Total time = $(126 \text{ hits} * 10 \text{ ns}) + (874 \text{ misses} * 110 \text{ ns})$.
 - Total time = 1260 ns + 96140 ns.
 - Total time = 97300 ns.

Summary

- **Number of Misses (Defaults):** 874
- **Number of Hits:** 126
- **Miss Time:** 110 ns
- **Hit Time:** 10 ns
- **Total Time of Memory Access:** 97300 ns

III. Third simulation

Cache memory organisation

Change the block size to 8 words

```
---Dinero IV cache simulator, version 7
---Written by Jim Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-l1-size 128
-l1-blocksize 8
-l1-subblocksize 8
-l1-uassoc 1
-l1-urepl 1
-l1-wlatch c
-l1-walloc d
-l1-wback a
-l1-ablock 0
-flushcount 0
-maxcount 0
-stat-interval 0
-infos 0
-on-trigger 0x0
-off-trigger 0x0

---Simulation begins.
---Simulation complete.
l1-wcache
Metrics      Total       Instrn      Data      Read      Write      Misc
-----  -----
Demand Fetches    1000        351       649       321       328        0
Fraction of total 1.0000     0.3510     0.6490     0.3210     0.3280     0.0000

Demand Misses     884        315       569       282       287        0
Demand miss rate 0.8840     0.8974     0.8767     0.8785     0.8750     0.0000

Multi-block refs      0
Bytes From Memory   7072
(/ Demand Fetches) 7.0720
Bytes Written       2496
(/ Demand Writes)  7.6098
Total Bytes r/w Mem 9568
(/ Demand Fetches) 9.5680

---Execution complete.
```

Simulation Results

- **Demand Fetches:** 1000 total (351 for instructions and 649 for data).
- **Demand Misses:** 884 total (315 for instructions and 569 for data).
- **Demand Miss Rate:** 88.40% overall (89.74% for instructions and 87.67% for data).

Calculations

1. Number of Misses:

- Total misses: 884.
- Instruction misses: 315.
- Data misses: 569.

2. Number of Hits:

- Total hits = Total fetches - Total misses = $1000 - 884 = 116$.

3. Miss Time:

- Miss time = Cache access time + Memory access time = $10 \text{ ns} + 100 \text{ ns} = 110 \text{ ns}$.

4. Hit Time:

- Hit time = Cache access time = 10 ns.

5. Total Time of Memory Access:

- Total time = (Number of hits * Hit time) + (Number of misses * Miss time).
- Total time = $(116 \text{ hits} * 10 \text{ ns}) + (884 \text{ misses} * 110 \text{ ns})$.
- Total time = 1160 ns + 97240 ns.
- Total time = 98400 ns.

Comparison with Previous Simulations

1. Number of Misses and Hits:

- The number of misses increased slightly from 874 to 884 with the block size change.
- The number of hits decreased from 126 to 116.

2. Time of Miss and Time of Hit:

- Miss time remains the same at 110 ns (10 ns for cache access + 100 ns for memory access).
- Hit time remains the same at 10 ns.

3. Total Time of Memory Access:

- The total time of memory access increased from 97300 ns to 98400 ns with the block size change.

Change the block size to 16 words

```

---Dinero IV cache simulator, version 7
---Written by Jan Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-l1-usize 128
-l1-ubsize 16
-l1-uassoc 16
-l1-urepl 1
-l1-ufetch d
-l1-uwalloc a
-l1-uwback a
-skipcount 0
-flushcount 0
-maxcount 0
-stat-interval 0
-informat d
-on-trigger 0x0
-off-trigger 0x0

---Simulation begins.
---Simulation complete.
l1-ucache
Metric      Total      Instrn      Data      Read      Write      Misc
-----      -----      -----      -----      -----      -----      -----
Demand Fetches      1000       351       649       321       328       0
Fraction of total   1.0000    0.3510    0.6490    0.3210    0.3280    0.0000

Demand Misses      879        310       569       284       285       0
Demand miss rate   0.8790    0.8832    0.8767    0.8847    0.8689    0.0000

Multi-block refs     0
Bytes From Memory   14064
( / Demand Fetches) 14.0640
Bytes To Memory     4944
( / Demand Writes) 15.0732
Total Bytes r/w Mem 19008
( / Demand Fetches) 19.0080

```

---Execution complete.

Simulation Results

- **Demand Fetches:** 1000 total (351 for instructions and 649 for data).
- **Demand Misses:** 879 total (310 for instructions and 569 for data).
- **Demand Miss Rate:** 87.90% overall (88.32% for instructions and 87.67% for data).

Calculations

1. Number of Misses:

- Total misses: 879.
- Instruction misses: 310.
- Data misses: 569.

2. Number of Hits:

- Total hits = Total fetches - Total misses = $1000 - 879 = 121$.

3. Miss Time:

- Miss time = Cache access time + Memory access time = $10 \text{ ns} + 100 \text{ ns} = 110 \text{ ns}$.

4. Hit Time:

- Hit time = Cache access time = 10 ns .

5. Total Time of Memory Access:

- Total time = (Number of hits * Hit time) + (Number of misses * Miss time).
- Total time = $(121 \text{ hits} * 10 \text{ ns}) + (879 \text{ misses} * 110 \text{ ns})$.
- Total time = $1210 \text{ ns} + 96690 \text{ ns}$.
- Total time = 97900 ns .

Comparison with Previous

1. Number of Misses and Hits:

- Compared to the previous block size of 8 words, the number of misses slightly decreased from 884 to 879.

- The number of hits increased from 116 to 121.
- 2. Time of Miss and Time of Hit:**
- Miss time remains the same at 110 ns (10 ns for cache access + 100 ns for memory access).
 - Hit time remains the same at 10 ns.
- 3. Total Time of Memory Access:**
- The total time of memory access decreased slightly from 98400 ns to 97900 ns.

IV. Fourth simulation

Cache Memory Organisation

Change the associativity, Complete associativity

And redo the previous simulations

```
---Dinero IV cache simulator, version 7
---Written by Jan Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-ll-utime 128
-ll-lbusize 4
-ll-ubsize 16
-ll-uassoc 2
-ll-uasoc 1
-ll-uwpl 1
-ll-ufetch d
-ll-uwallc a
-ll-uwaloc a
-ll-uwatch a
-skipevict 0
-flushcount 0
-mexcoun 0
-stat-interval 0
-informat d
-out-format b6
-off-trigger 0x0

---Simulation begins.
---Simulation complete.

11-usage
Metrics          Total       Instrn      Data      Read      Write      Misc
-----+-----+-----+-----+-----+-----+-----+
Demand Patches   1000        351       649       321       328        0
Fraction of total 1.0000     0.3510    0.6490    0.3210    0.3280    0.0000

Demand Misses   870         307       563       282       281        0
Demand miss rate 0.8700    0.8746    0.8675    0.8785    0.8667    0.0000

Multi-block refs 0
Bytes From Memory 2356
Bytes To Memory 1228
Bytes Total (116) 3584
Total Bytes r/w Mem 3584
( / Demand Patches) 3.5840

---Execution complete.

---Dinero IV cache simulator, version 7
---Written by Jan Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-ll-utime 128
-ll-lbusize 8
-ll-ubsize 16
-ll-uassoc 16
-ll-uasoc 1
-ll-uwpl 1
-ll-ufetch d
-ll-uwallc a
-ll-uwaloc a
-skipevict 0
-flushcount 0
-mexcoun 0
-stat-interval 0
-informat d
-out-format b6
-off-trigger 0x0

---Simulation begins.
---Simulation complete.

11-usage
Metrics          Total       Instrn      Data      Read      Write      Misc
-----+-----+-----+-----+-----+-----+-----+
Demand Patches   1000        351       649       311       328        0
Fraction of total 1.0000     0.3510    0.6490    0.3210    0.3280    0.0000

Demand Misses   870         310       560       278       282        0
Demand miss rate 0.8700    0.8832    0.8629    0.8660    0.8598    0.0000

Multi-block refs 0
Bytes From Memory 6960
Bytes To Memory 6.9600
Bytes Total (116) 1444
Bytes Total (Demand Writes) 7.5122
Total Bytes r/w Mem 9424
( / Demand Patches) 9.4240

---Execution complete.
```

```

---Dinero IV cache simulator, version 7
---Written by Jan Edler and Mark D. Hill
---Copyright (C) 1997 NEC Research Institute, Inc. and Mark D. Hill.
---All rights reserved.
---Copyright (C) 1985, 1989 Mark D. Hill. All rights reserved.
---See -copyright option for details

---Summary of options (-help option gives usage information).

-l1-usize 128
-l1-ubsize 16
-l1-usbsize 16
-l1-lblock 8
-l1-urepl 1
-l1-ufetch d
-l1-uwallc a
-l1-uwallback a
-skipcount 0
-flushcount 0
-maxcycles 0
-ite-interval 0
-informat d
-on-trigger 0x0
-off-trigger 0x0

---Simulation begins.
---Simulation complete.
l1-cache
Metrics          Total      Instrn      Data      Read      Write      Misc
-----  -----
Demand Fetches   1000       351        649       321       328        0
Fraction of total 1.0000    0.3510     0.6490    0.3210    0.3280    0.0000

Demand Misses    873        311        562       277       285        0
Demand miss rate 0.8730    0.8860     0.8659    0.8629    0.8689    0.0000

Multi-block refs 0
Bytes From Memory 13968
( / Demand Fetches) 13.9680
Bytes To Memory 4912
( / Demand Writes) 14.9756
Total Bytes r/w Mem 18880
( / Demand Fetches) 18.8800

---Execution complete.

```

Calculations for Each Simulation

1. First Simulation:

- **Miss Time:** 110 ns (Cache access time + Memory access time)
- **Hit Time:** 10 ns (Cache access time)
- **Total Time of Memory Access:** Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)
Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)=(130×10 ns)+(870×110 ns)=1300 ns+95700 ns=97000 ns=(130×10 ns)+(870×110 ns)=1300 ns+95700 ns=97000 ns

2. Third Simulation(Change the block size to 8 words):

- **Miss Time:** 110 ns (Cache access time + Memory access time)
- **Hit Time:** 10 ns (Cache access time)
- **Total Time of Memory Access:** Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)
Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)=(130×10 ns)+(870×110 ns)=1300 ns+95700 ns=97000 ns=(130×10 ns)+(870×110 ns)=1300 ns+95700 ns=97000 ns

3. Third Simulation(Change the block size to 16 words):

- **Miss Time:** 110 ns (Cache access time + Memory access time)
- **Hit Time:** 10 ns (Cache access time)
- **Total Time of Memory Access:** Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)
Total Time=(Number of Hits×Hit Time)+(Number of Misses×Miss Time)=(127×10 ns)+(873×110 ns)=1270 ns+96030 ns=97300 ns=(127×10 ns)+(873×110 ns)=1270 ns+96030 ns=97300 ns

Summary and Comparison

1. First Simulation:

- **Number of Misses (Defaults):** 870
- **Number of Hits:** 130
- **Miss Time:** 110 ns

- **Hit Time:** 10 ns
- **Total Time of Memory Access:** 97000 ns

2. **Second Simulation:**

- **Number of Misses (Defaults):** 870
- **Number of Hits:** 130
- **Miss Time:** 110 ns
- **Hit Time:** 10 ns
- **Total Time of Memory Access:** 97000 ns

3. **Third Simulation:**

- **Number of Misses (Defaults):** 873
- **Number of Hits:** 127
- **Miss Time:** 110 ns
- **Hit Time:** 10 ns
- **Total Time of Memory Access:** 97300 ns