CCCS217 PROJECT

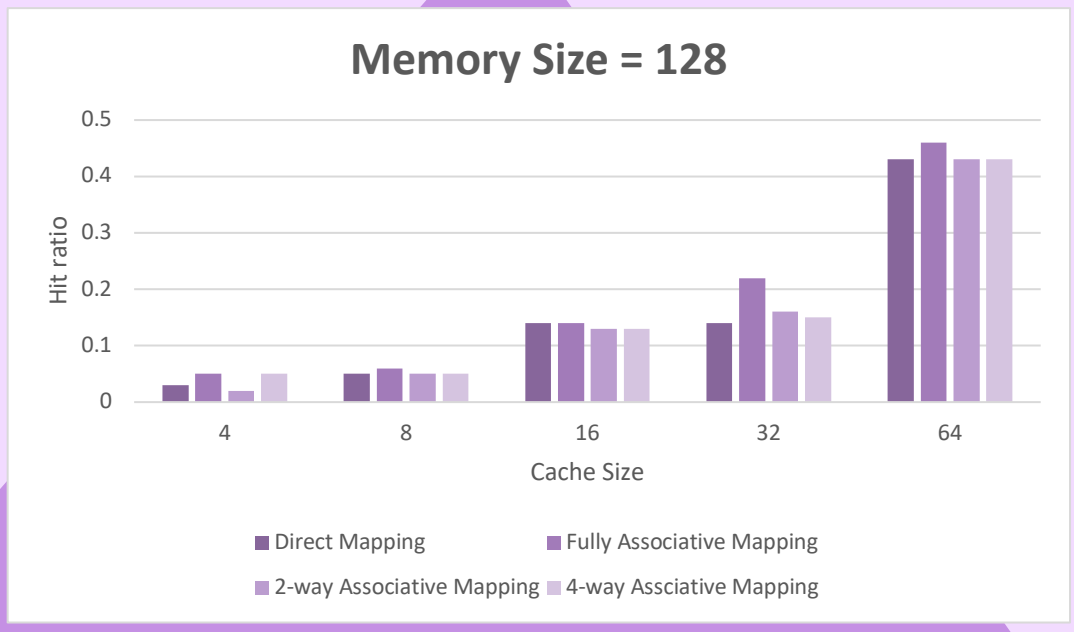
Team members:

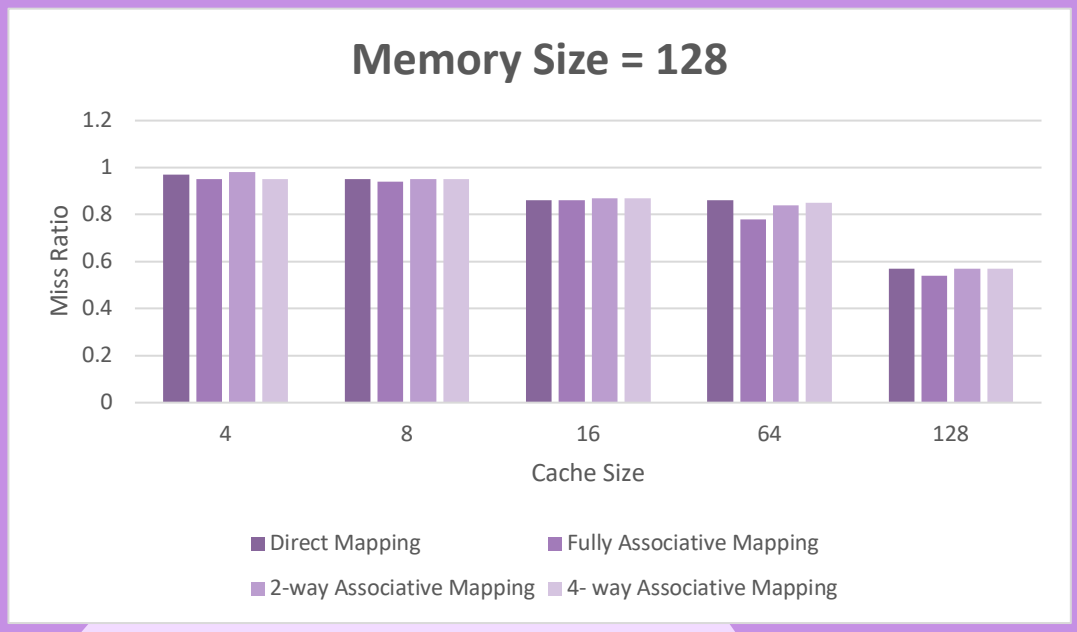
Lama Youssef Alghamdi

Joud Abdullah Aljehani

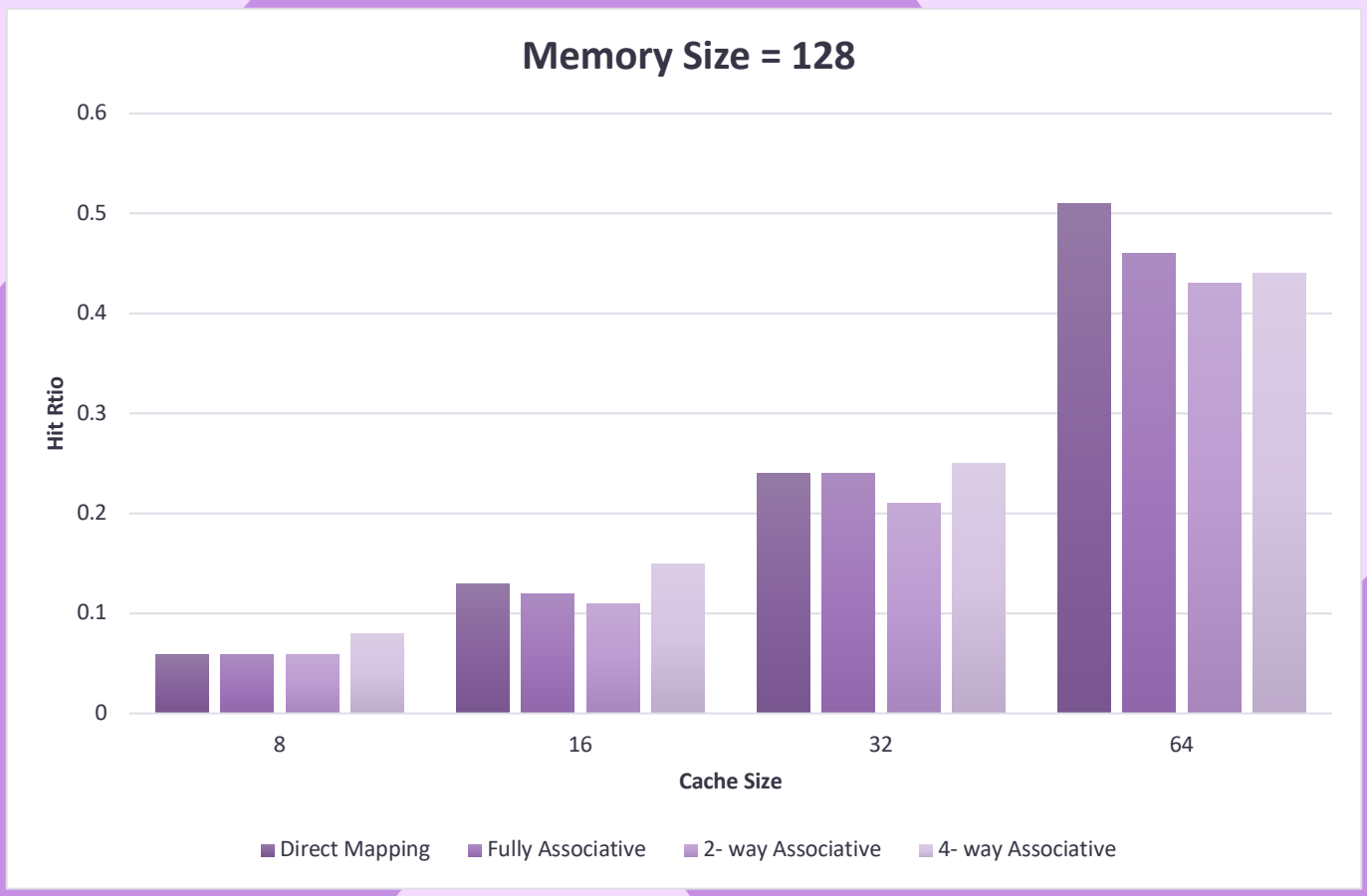
Jana Yasser Ghorab

a)

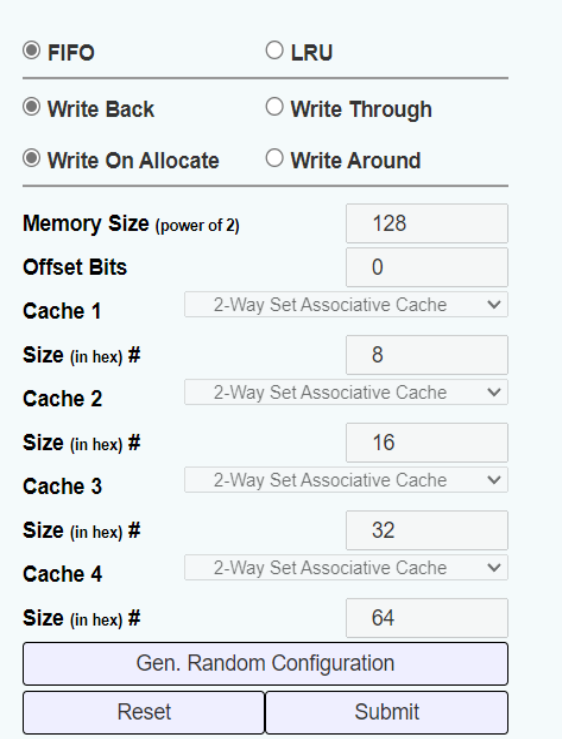

Memory Size = 128
0.5
0.4
0.3
0.2
0.1
0
Hit ratio
4
8
16
32
64
Cache Size
Direct Mapping
Fully Associative Mapping
2-way Associative Mapping
4-way Assciative Mapping

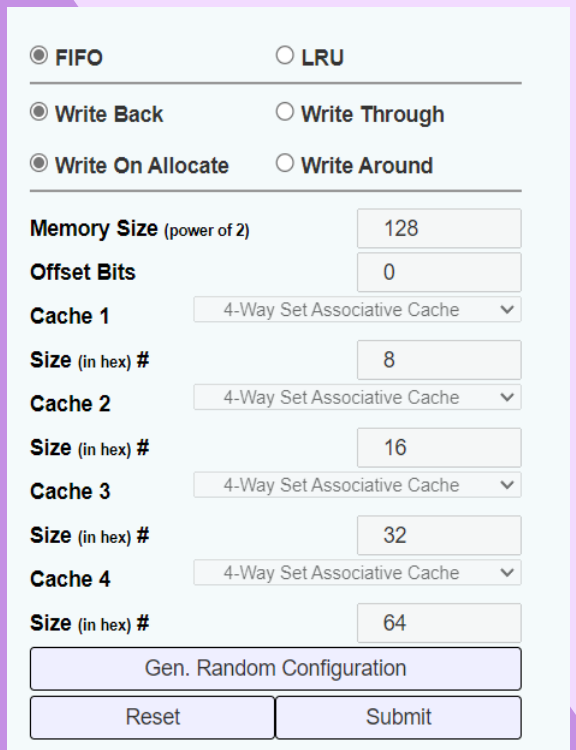

Memory Size = 128
1.2
1
0.8
0.6
0.4
0.2
0
Miss Ratio
4
8
16
64
128
Cache Size
Direct Mapping
Fully Associative Mapping
2-way Associative Mapping
4- way Associative Mapping

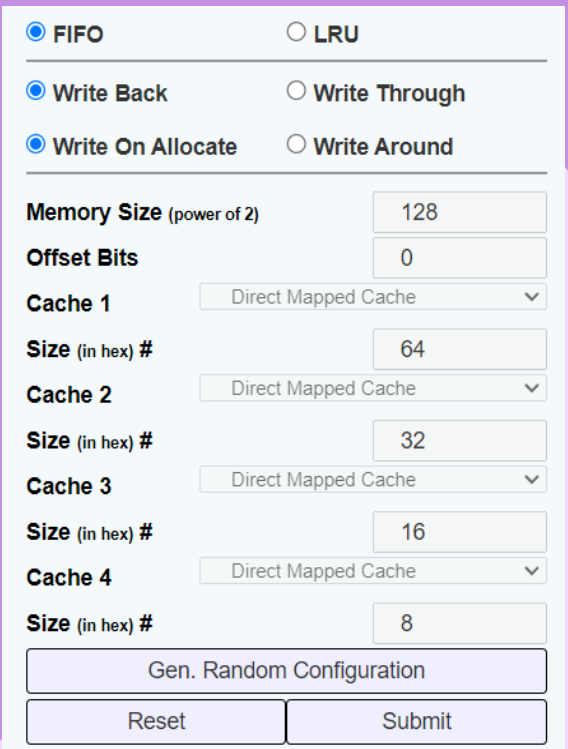
b)

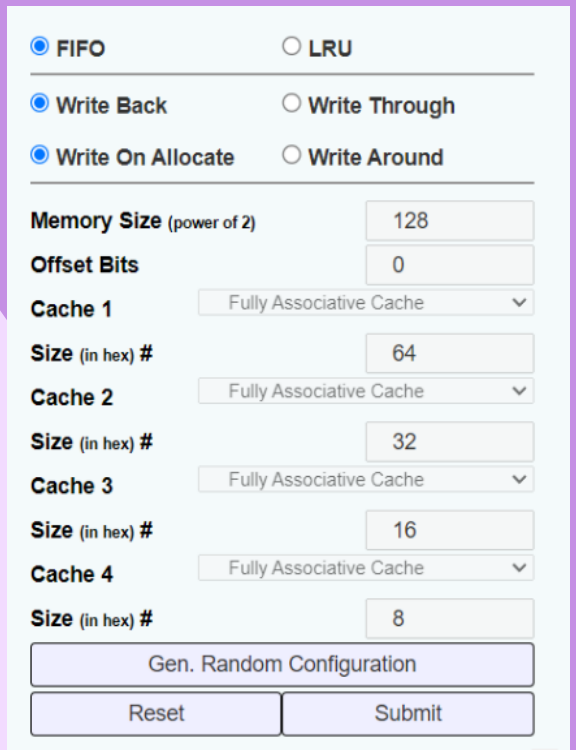

Memory Size = 128
0.6
0.5
0.4
0.3
0.2
0.1
0
Hit Rtio
8
16
32
64
Cache Size
Direct Mapping
Fully Associative
2- way Associative
4- way Associative

C) we've changed the memory size to 128 and by increasing the cache size the hit ratio would increase









d)

1 - Direct Mapping

Cache Size = 64

Instruction Breakdown

1	101100	0
1 bit	6 bit	0 bit

Cache Size = 64

Memory Size = $128 = 2^7$

PA = $\log_2 128 = \log_2 2^7 = 7$

Tag = S - r

= 7 - 6 = 1

Tag = 1

Tag	Line	word
1 bit	6 bit	0

2- Fully Associative Mapping

Cache Size = 64

Instruction Breakdown

1011101	0
7 bit	0 bit

Cache Size = 64

Memory Size = $128 = 2^7$

PA = $\log_2 128 = \log_2 2^7 = 7$

Tag = S

S = 7

Tag	word
7 bit	0

3- 4-way Associative Mapping

Cache Size = 64

Instruction Breakdown

010	0001	0
3 bit	4 bit	0 bit

Cache Size = 64

Memory Size = $128 = 2^7$

PA = $\log_2 128 = \log_2 2^7 = 7$

Tag = (S-D) bits

Tag = 7 - 4 = 3

Tag	Set	word
3 bit	4 bit	0

part2:

```
.data
massege: .asciiz "Please Enter Octal Number(accept 3-digit): "
massege2: .asciiz "\nThe Number in decimal: "
massege3: .asciiz "\nWrong Number "


.text
la $a0, massege
li $v0, 4
syscall    #print the first message
li $v0, 5
syscall    #read the number
move $t0, $v0 #stor the number in v0 to t0

addi $a0, $zero, 0 #argument
addi $t2, $zero, 1
addi $s1, $zero, 99
addi $s2, $zero, 1000
addi $s3, $zero, 10
addi $s4, $zero, 8
ble $t0, $s1, exit #if the number less than 3 digit exit
bge $t0, $s2, exit #if the number more than 3 digit exit

while:
div $t0, $s3  #Divide t0 by s3
mfhi $t1 #Store the remainder in t1
div $t0, $s3 #div the number
mflo $t0 #stor the div in t0
mult $t1, $t2  #Multiply t1 by t2
mflo $t1  #Store the multiply in t1
add $a0, $a0, $t1  #Add t1 to a0 and store it in a0
mul $t2, $t2, $s4  #multiply $t2 by s4
bne $t0, $zero, while #t0 not equal to zero then repeat the while loop
move $t0, $a0   #Stor a0 in t0

la $a0, massege2
li $v0, 4
syscall
move $a0, $t0
li $v0, 1
syscall
exit:
la $a0, massege3
li $v0, 4
syscall
li $v0, 10
syscall
```

- **3-digit**

```
Please Enter Octal Number(accept 3-digit): 140

The Number in decimal: 96
-- program is finished running --
```

- **Less than 3-digit**

```
Please Enter Octal Number(accept 3-digit): 77

Wrong Number
-- program is finished running --
```

- **More than 3-digit**

```
Please Enter Octal Number(accept 3-digit): 1900

Wrong Number
-- program is finished running --
```