Miniature Processor

{ Design and contribution by :

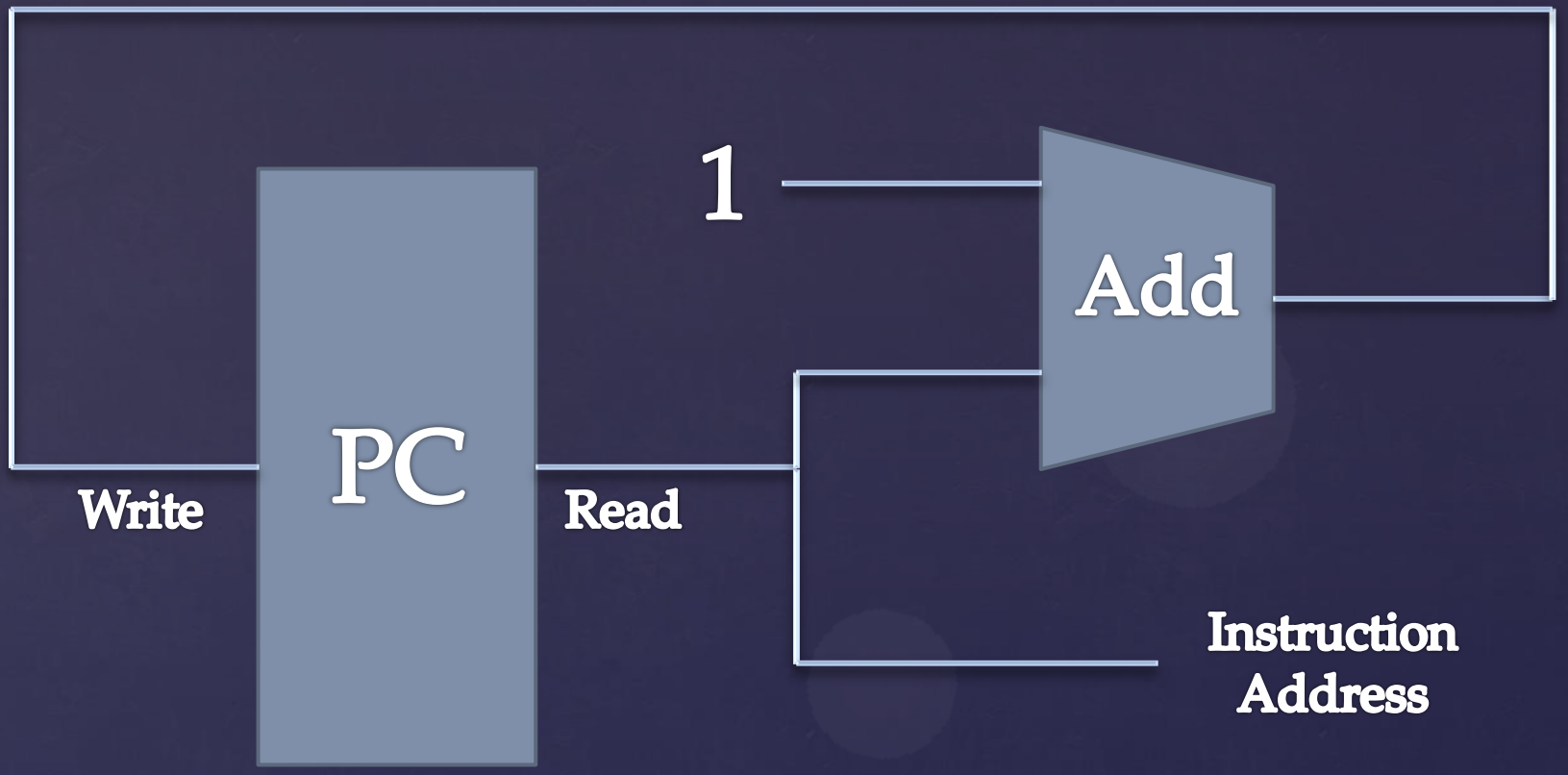
- *Mohsen Dehbashi*
- *Mohammad Hashemi*

```
1  lw 1,0,five
2  lw 2,1,2
3  start add 1,1,2
4  beq 0,1,done
5  j start
6
7  #done halt
8  done halt # redundant
9  five .fill 5
10 neg1 .fill -1
11 stAddr .fill start
```

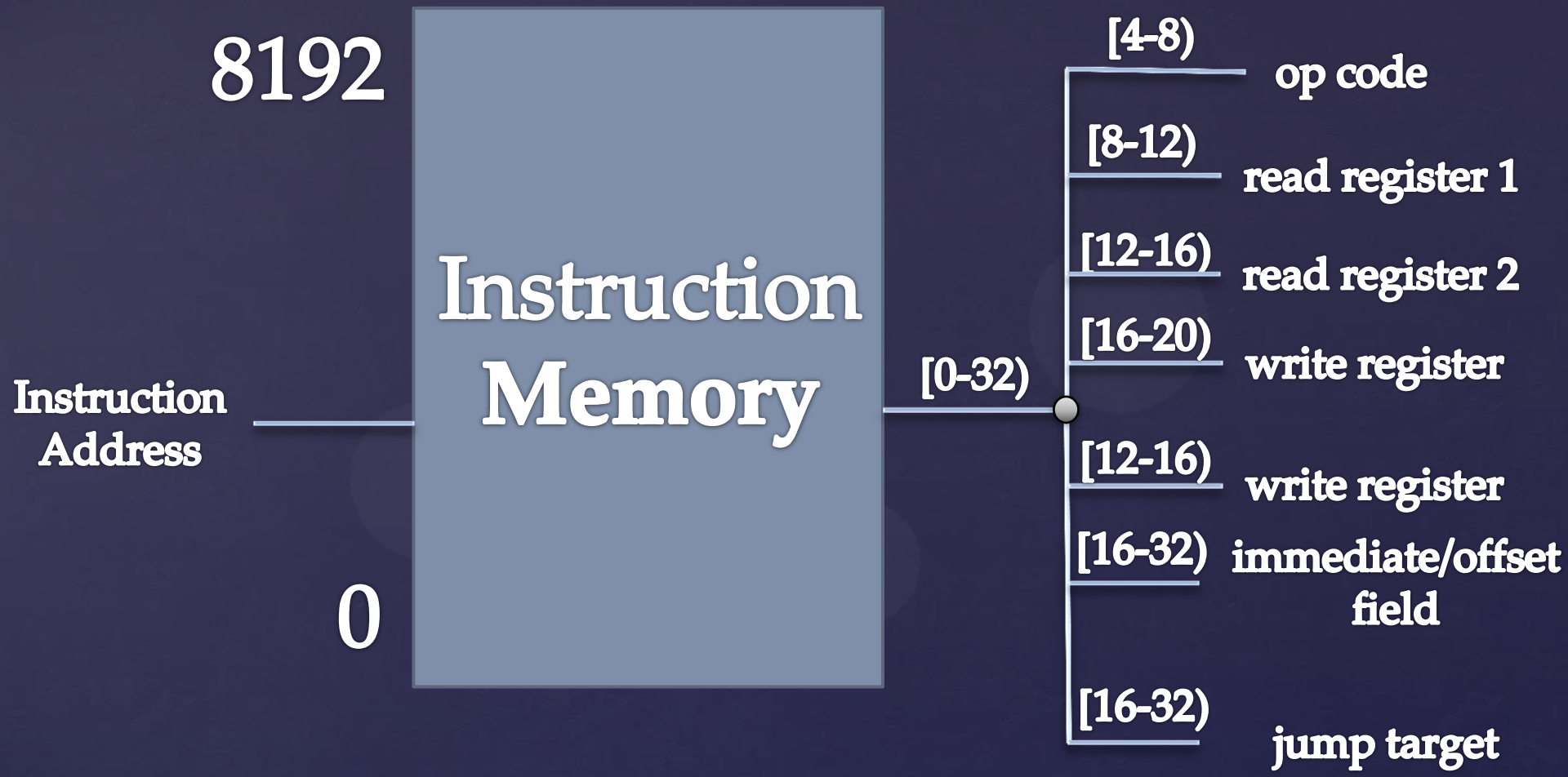
```
1 00001001000000010000000000000101
2 00001001000100100000000000000010
3 00000000000100100001000000000000
4 00001011000100000000000000000111
5 00001101000000000000000000000010
6 00001110000000000000000000000000
7 00000000000000000000000000000101
8 11111111111111111111111111111111
9 00000000000000000000000000000010
```

```
python assemble.py program.as program.mc
```

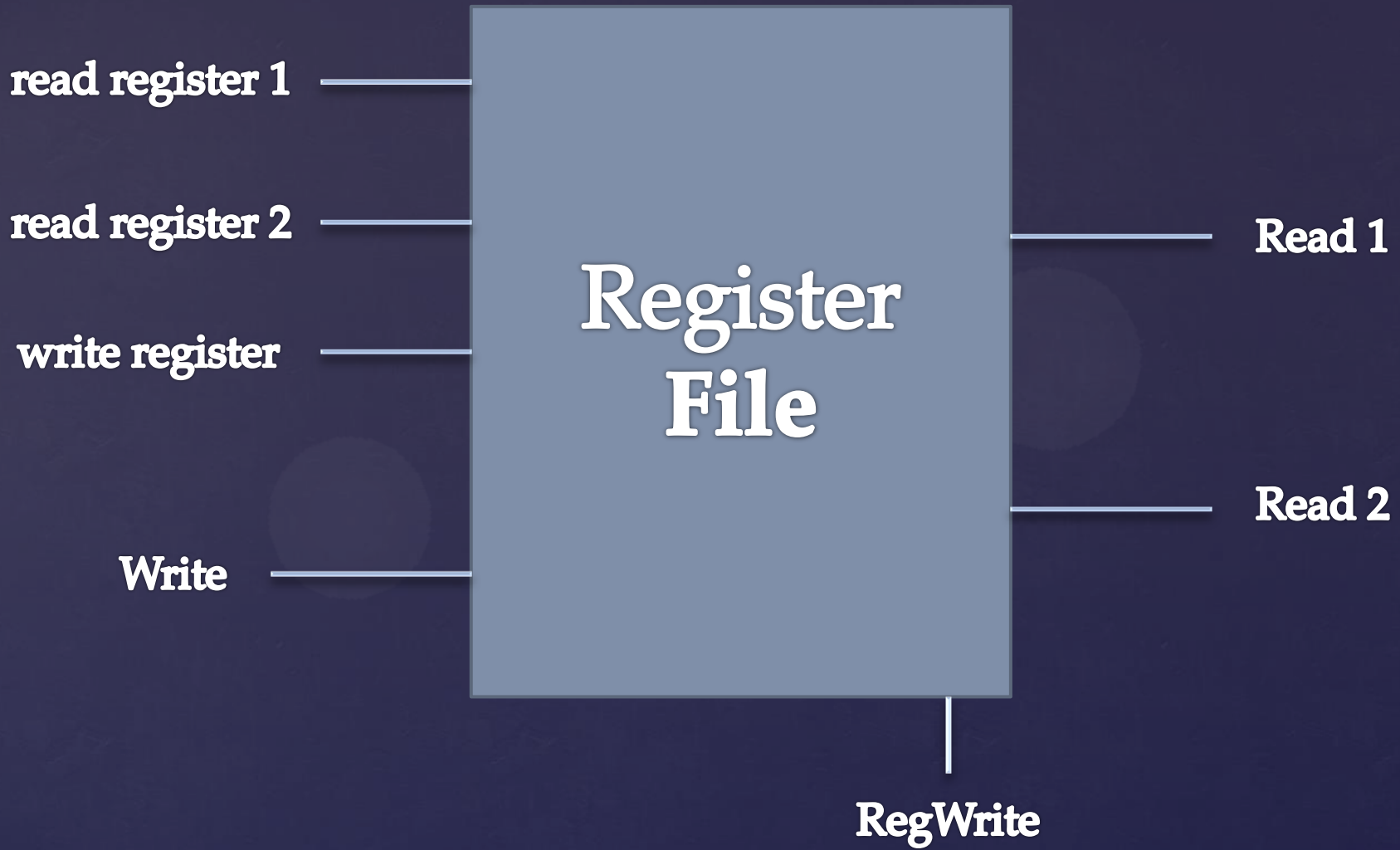
Miniature Assembler



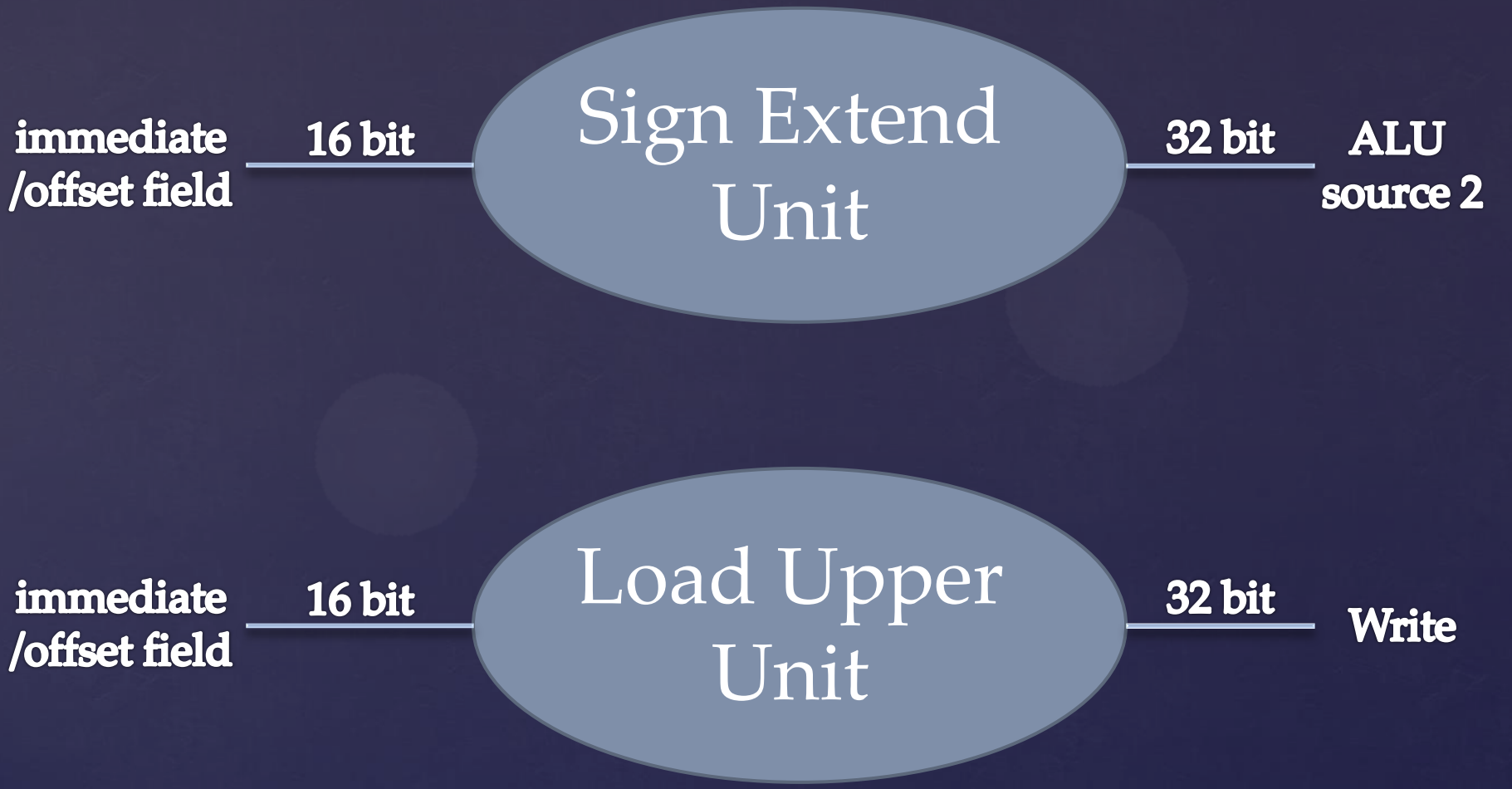
Program Counter
32-bit Register

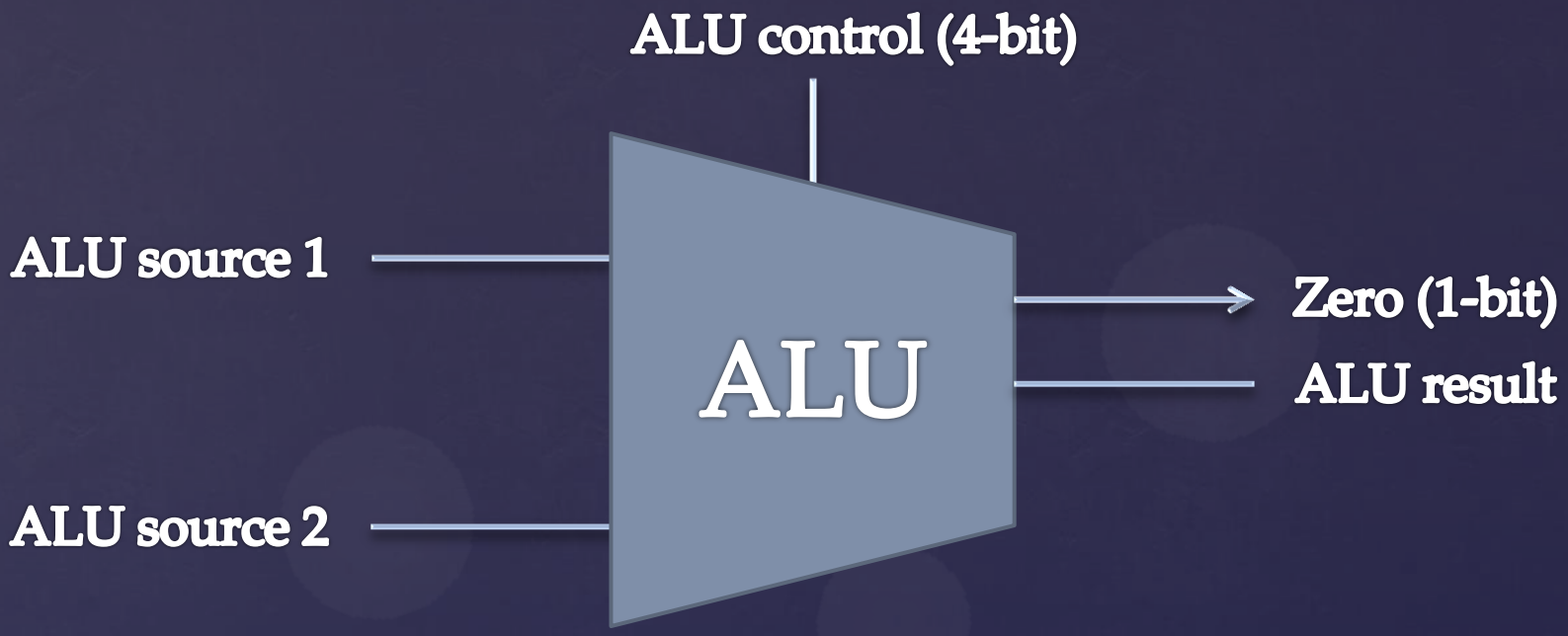


32-bit Instruction Field

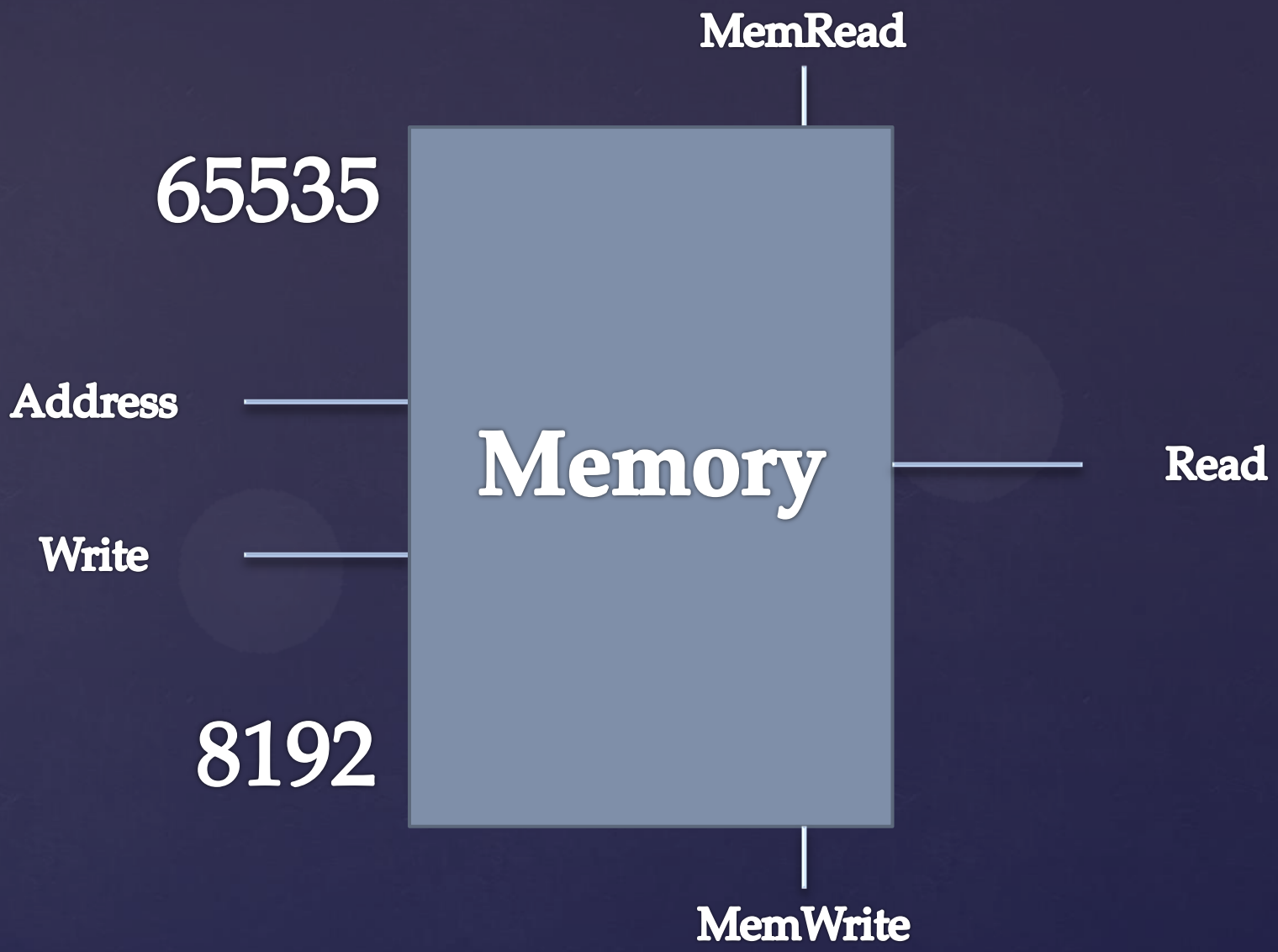


16 Registers

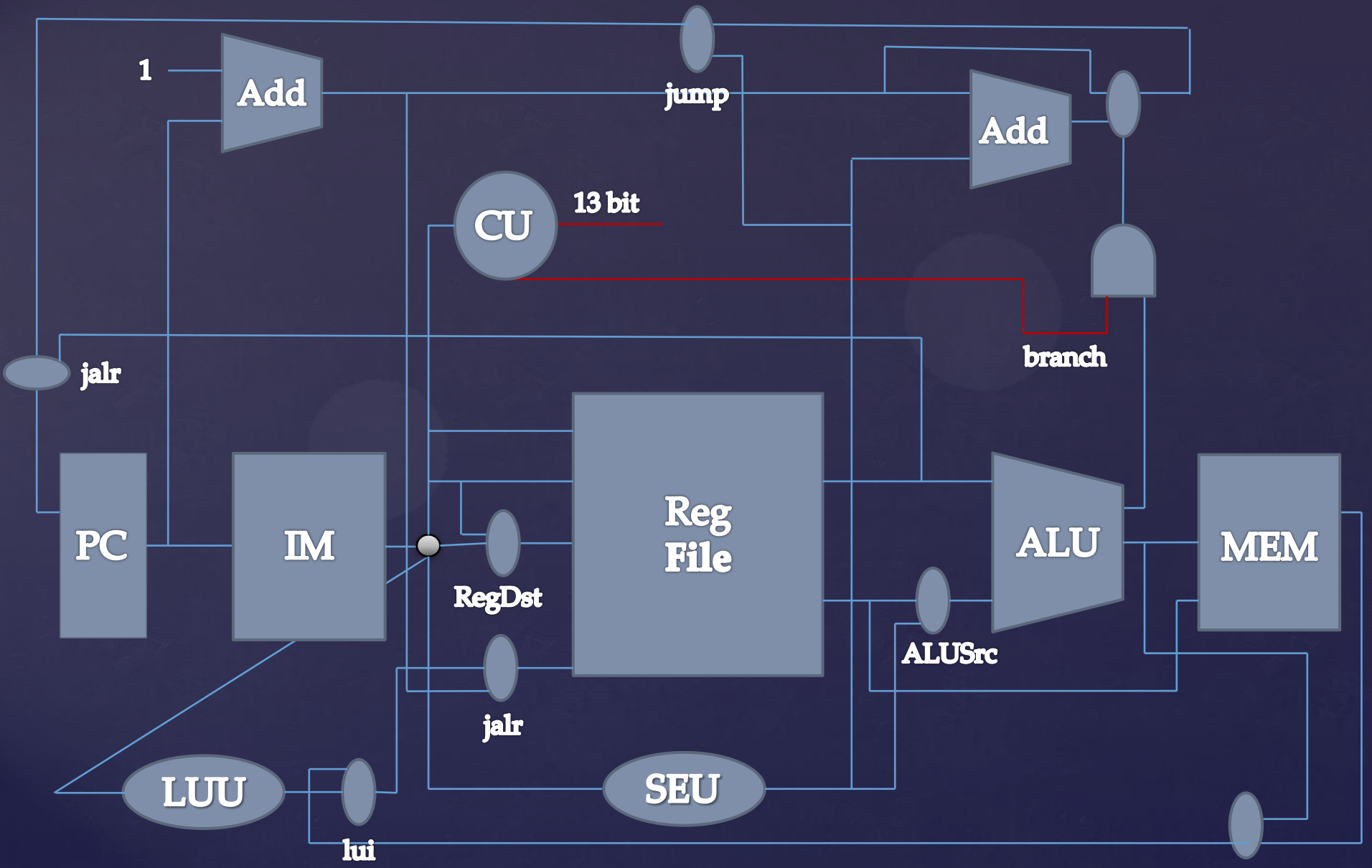

immediate /offset field
16 bit
Sign Extend Unit
32 bit
ALU source 2
immediate /offset field
16 bit
Load Upper Unit
32 bit
Write



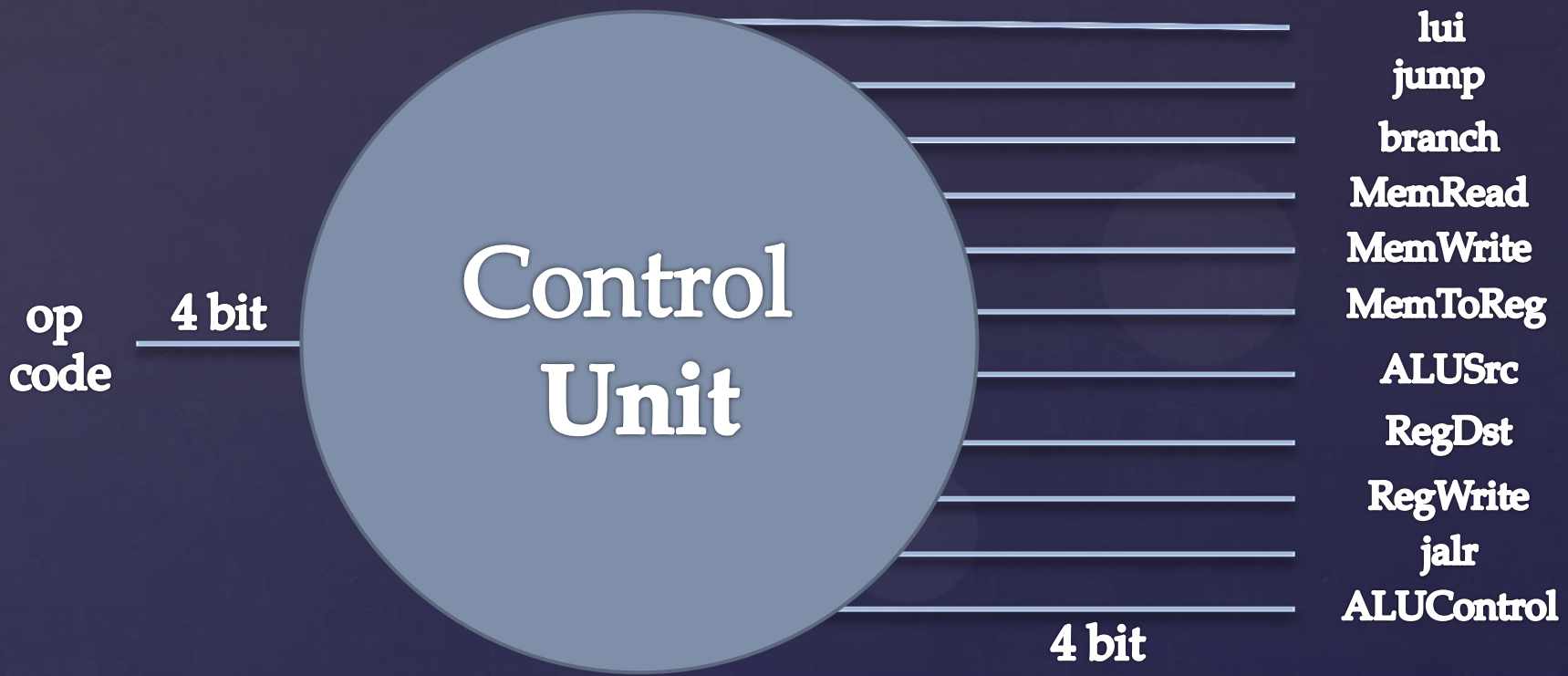
Arithmetic Logical Unit



Data Memory



Miniature Datapath



Control the Datapath

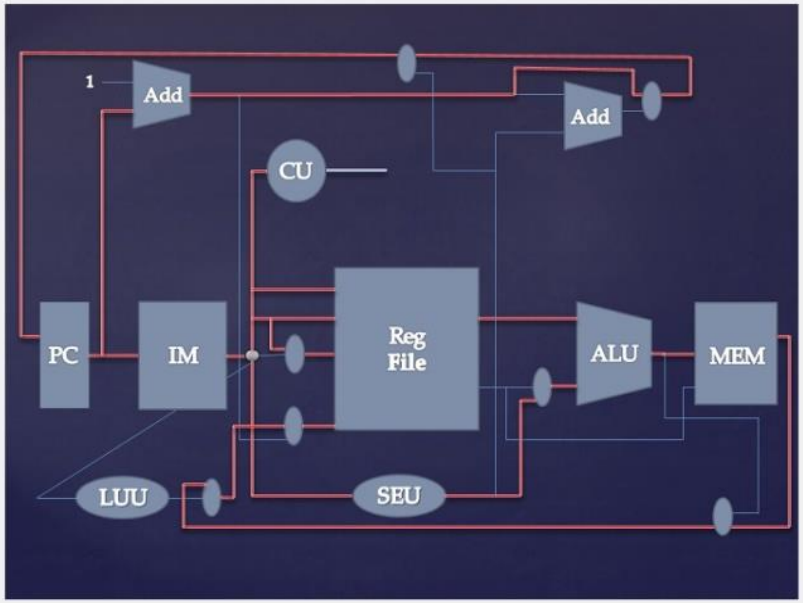
ALU control	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set-on-less-than
1100	NOR

ALU Control Unit

	op code	jump	branch	Mem Read	Mem Write	MemTo Reg	ALU Src	Reg Dst	Reg Write	jalr	lui	ALUControl
add	0000	0	0	0	0	0	0	1	1	0	0	0010
sub	0001	0	0	0	0	0	0	1	1	0	0	0110
slt	0010	0	0	0	0	0	0	1	1	0	0	0111
or	0011	0	0	0	0	0	0	1	1	0	0	0001
and	0100	0	0	0	0	0	0	1	1	0	0	0000
addi	0101	0	0	0	0	0	1	0	1	0	0	0010
slti	0110	0	0	0	0	0	1	0	1	0	0	0111
ori	0111	0	0	0	0	0	1	0	1	0	0	0001
lui	1000	0	0	0	0	X	X	0	1	0	1	XXXX
lw	1001	0	0	1	0	1	1	0	1	0	0	0010
sw	1010	0	0	0	1	X	1	X	0	0	0	0010
beq	1011	0	1	0	0	X	0	X	0	0	0	0110
jalr	1100	0	0	0	0	X	X	0	1	1	0	XXXX
j	1101	1	X	0	0	X	X	X	0	X	X	XXXX

PC : 0 Instruction : 00001001000000010000000000000101

lw 1,0,5



R0 = 0
R1 = 0
R2 = 0
R3 = 0
R4 = 0
R5 = 0
R6 = 0
R7 = 0
R8 = 0
R9 = 0
R10 = 0
R11 = 0
R12 = 0
R13 = 0
R14 = 0
R15 = 0

Memory : {}

```
python simulate.py program.mc
```

Miniature Simulator

Thanks!

{ Computer Organisations and Design Lesson

Dr. Mehran Rezaee

University of Isfahan, Fourth Grade

Spring 1397