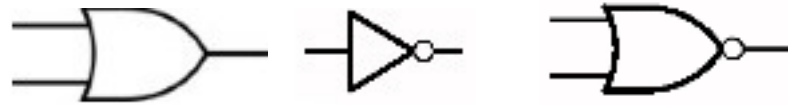


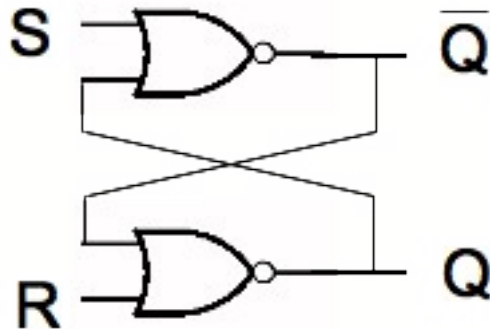
# A 1-Bit Memory



This stuff is truly unforgettable!

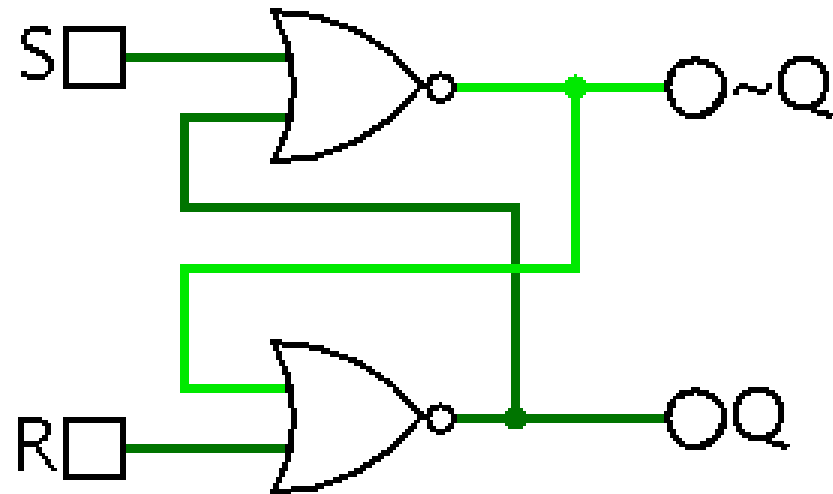


OR + NOT = NOR



# A 1-Bit Memory

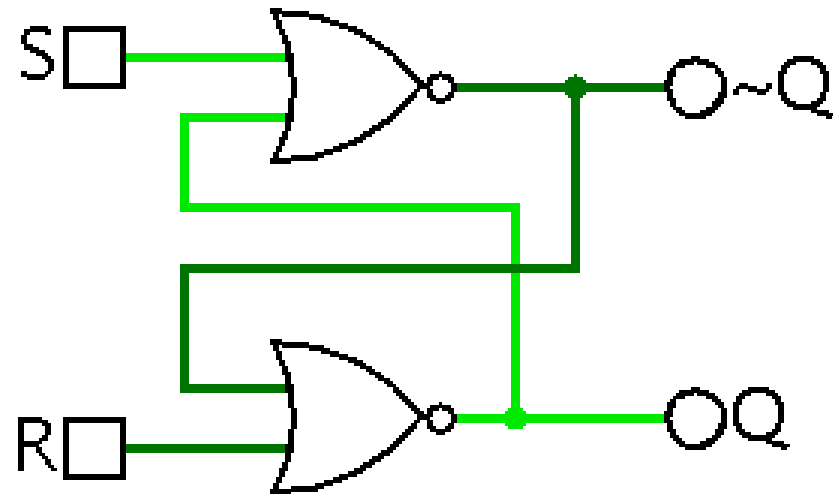
---



# Setting a 1-Bit Memory

---

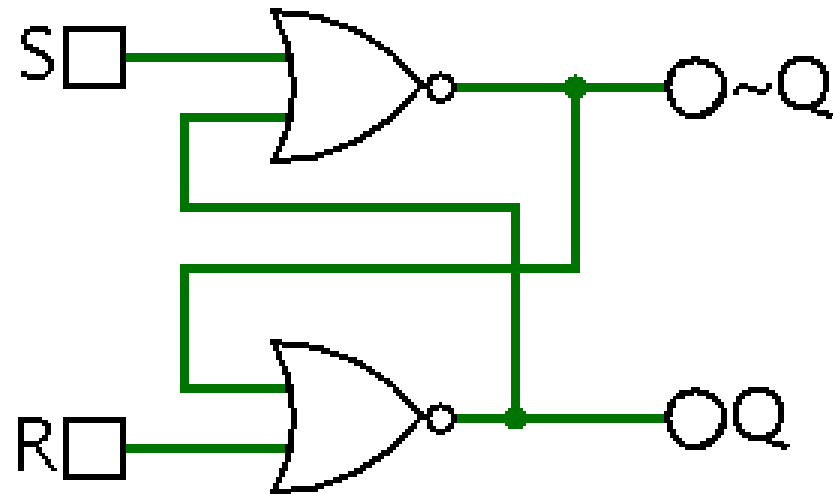
---



# Initializing a 1-Bit Memory

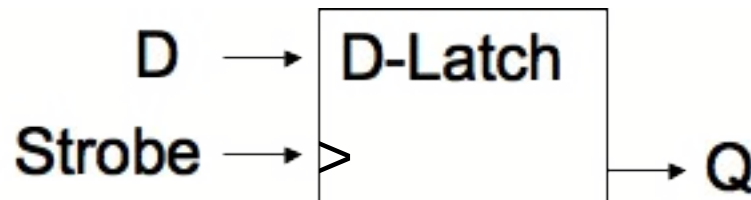
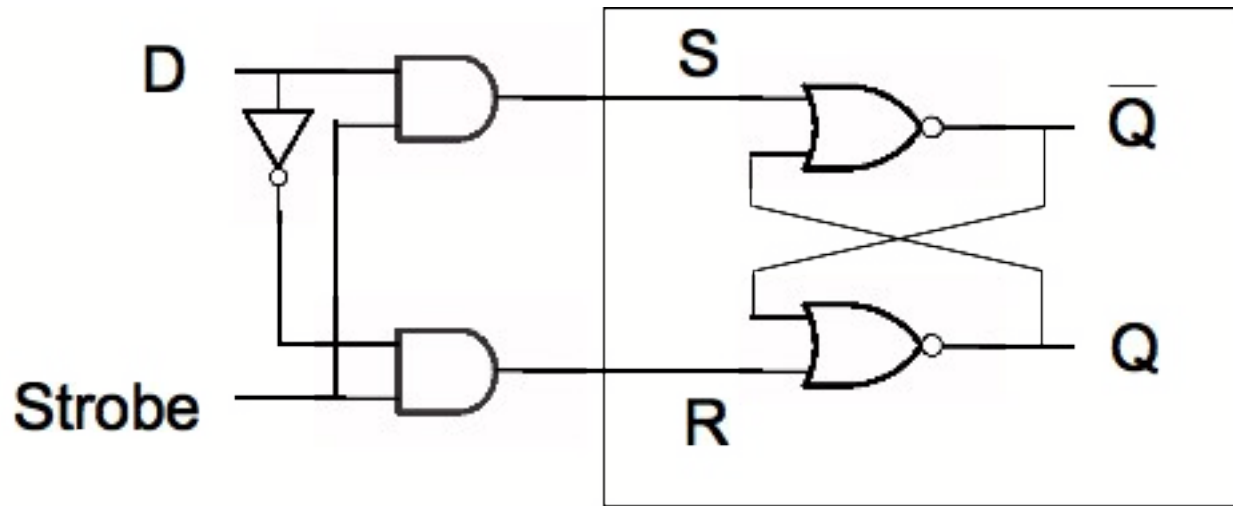
---

---



# From S-R Latches to D-Latches

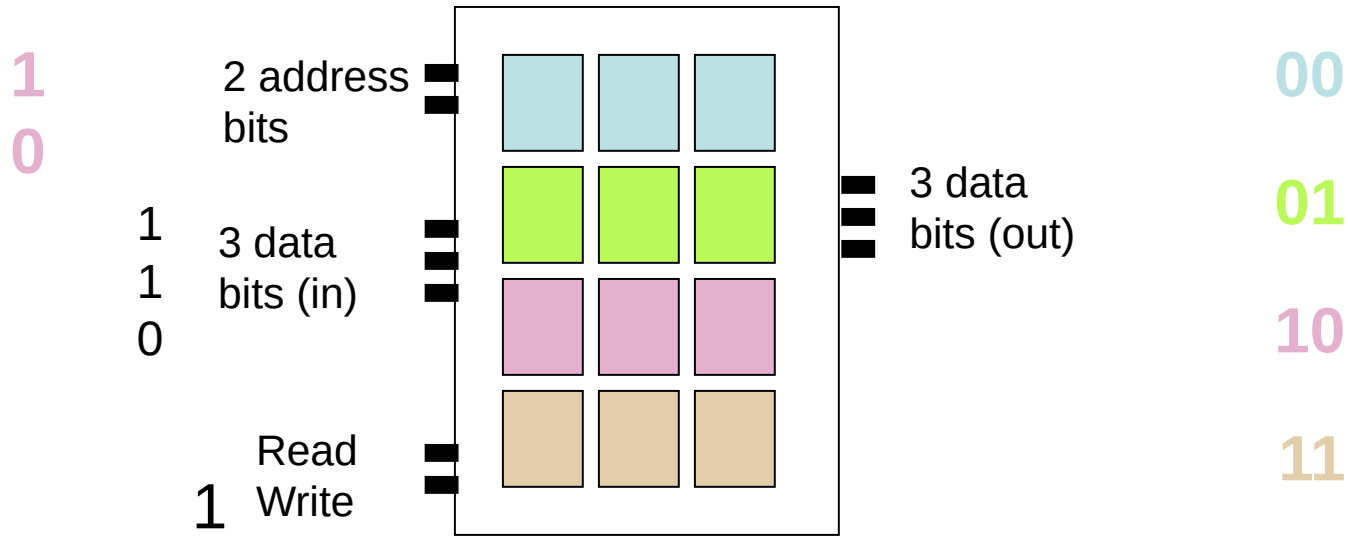
---



Worksheet!

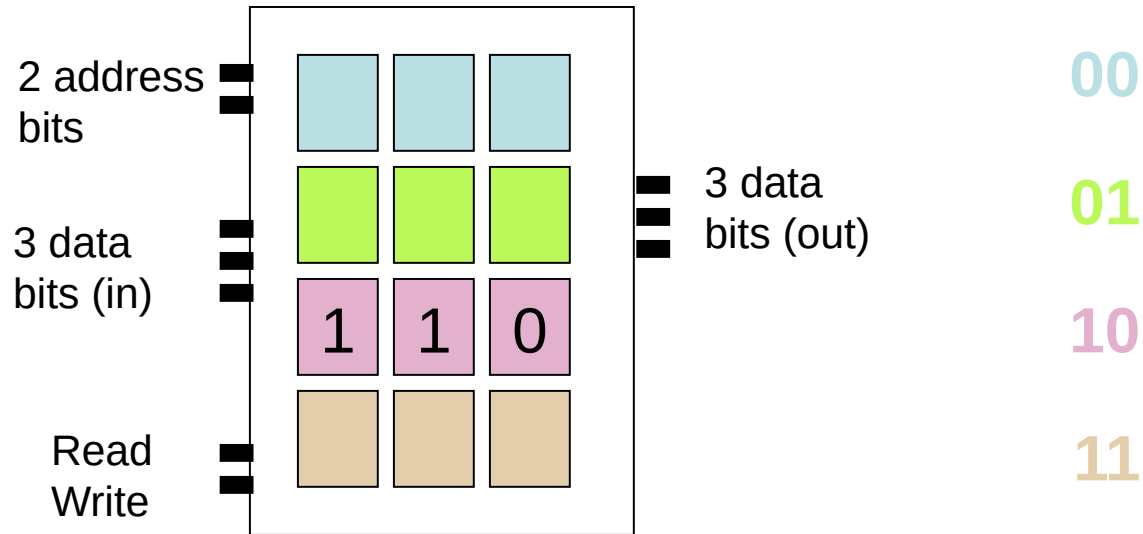
# A Random Access Memory (RAM)

---



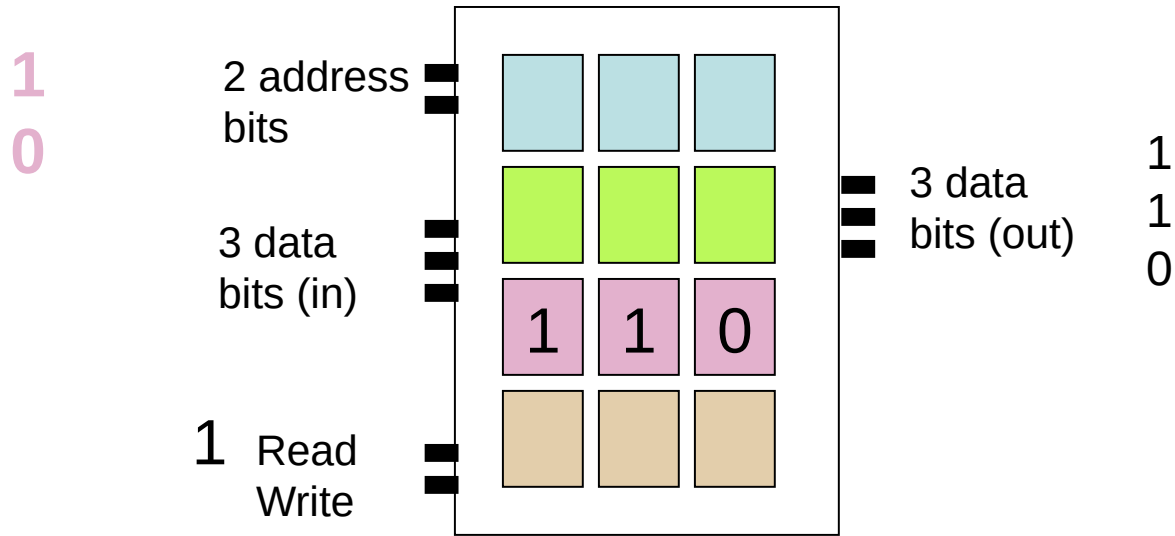
# A Random Access Memory (RAM)

---



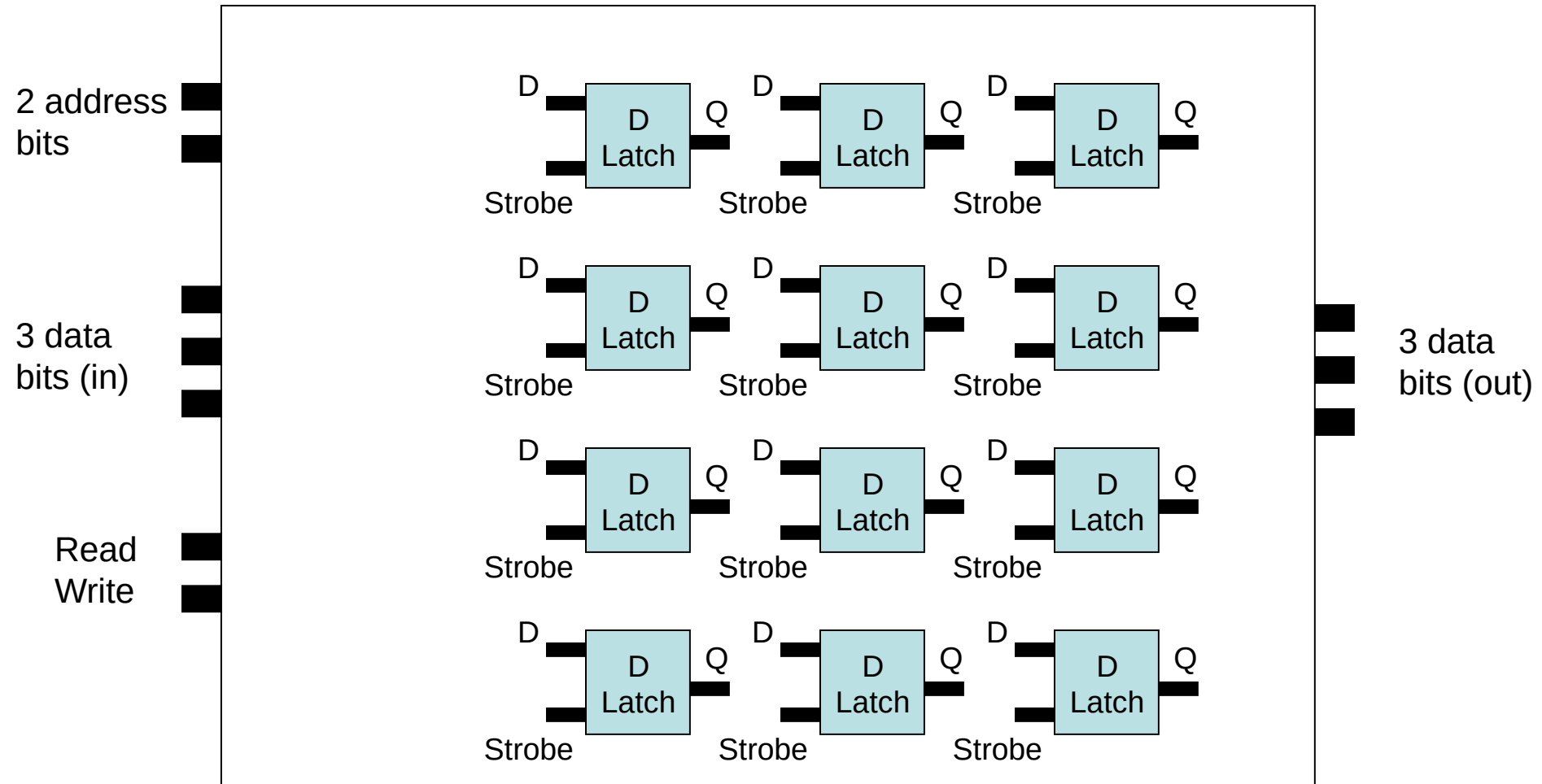
# A Random Access Memory (RAM)

---

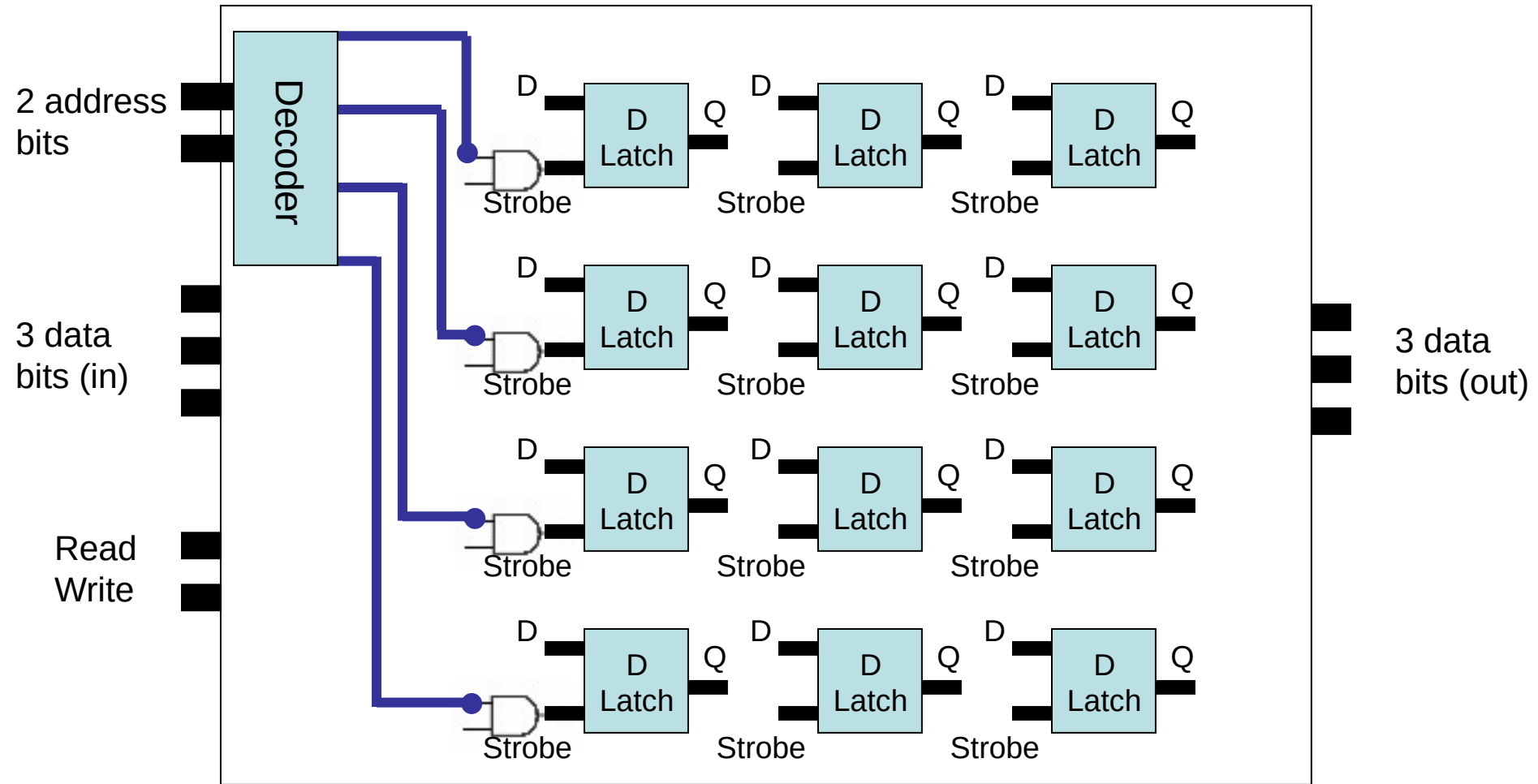




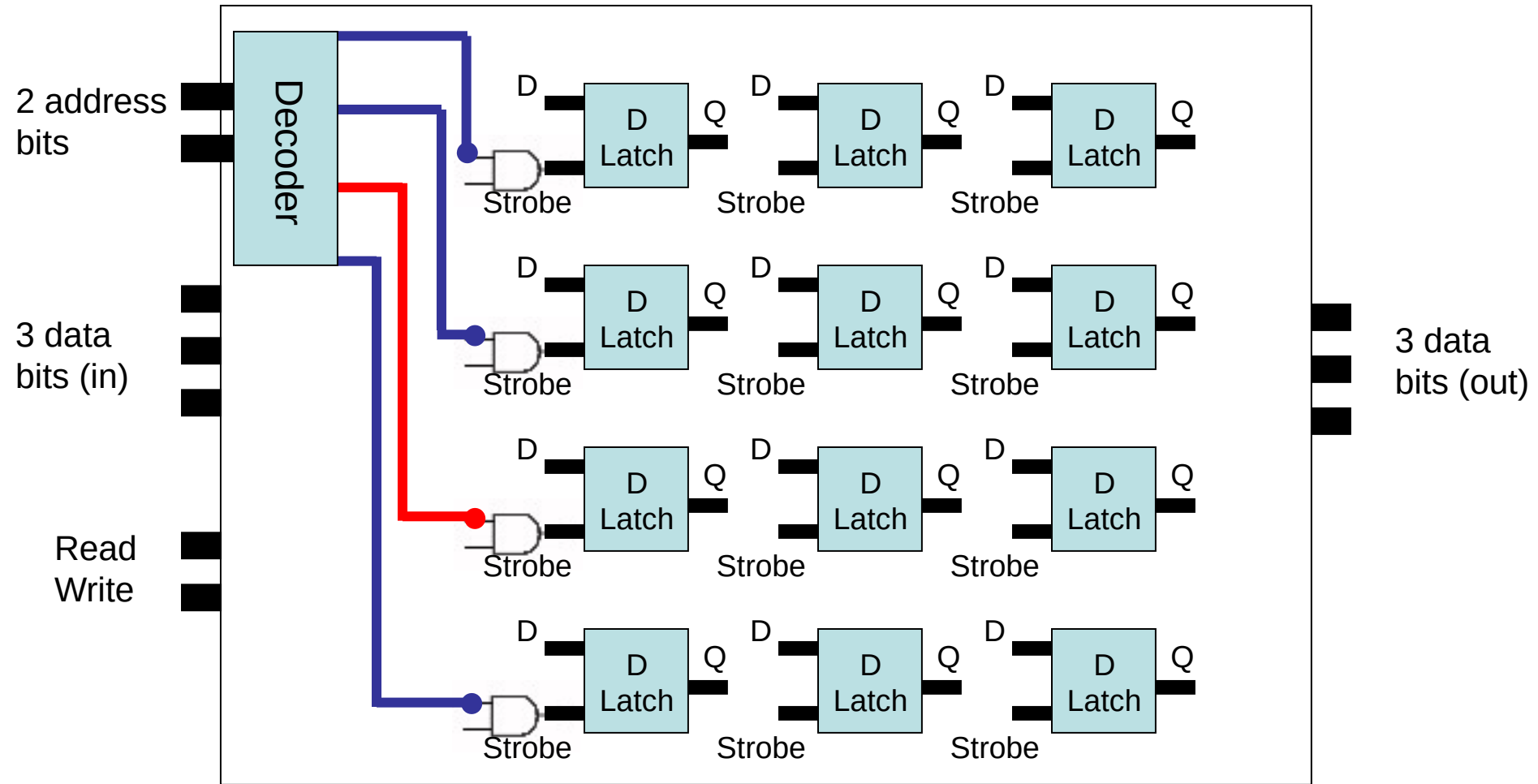
# A Random Access Memory (RAM)



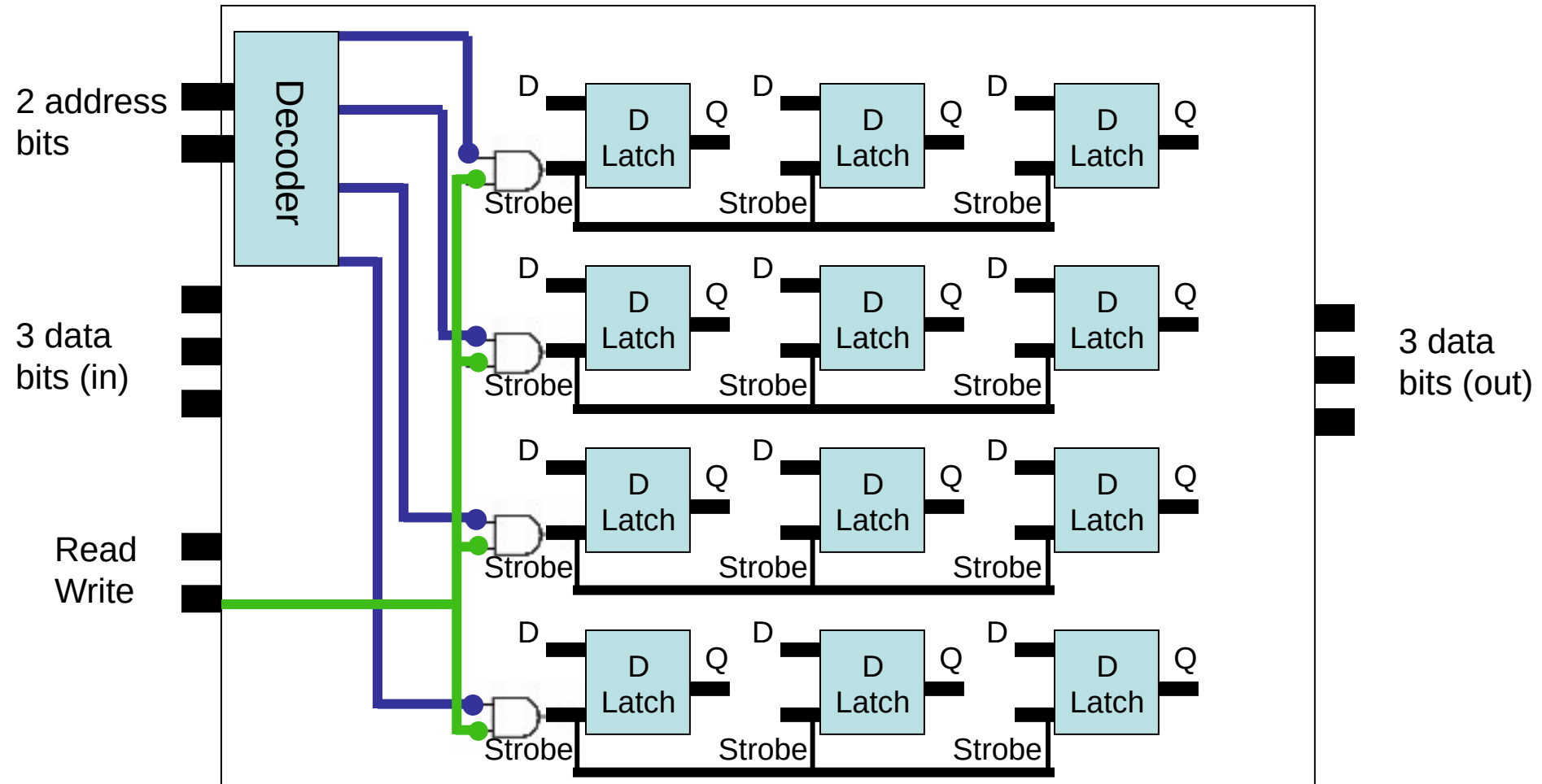
# A Random Access Memory (RAM)



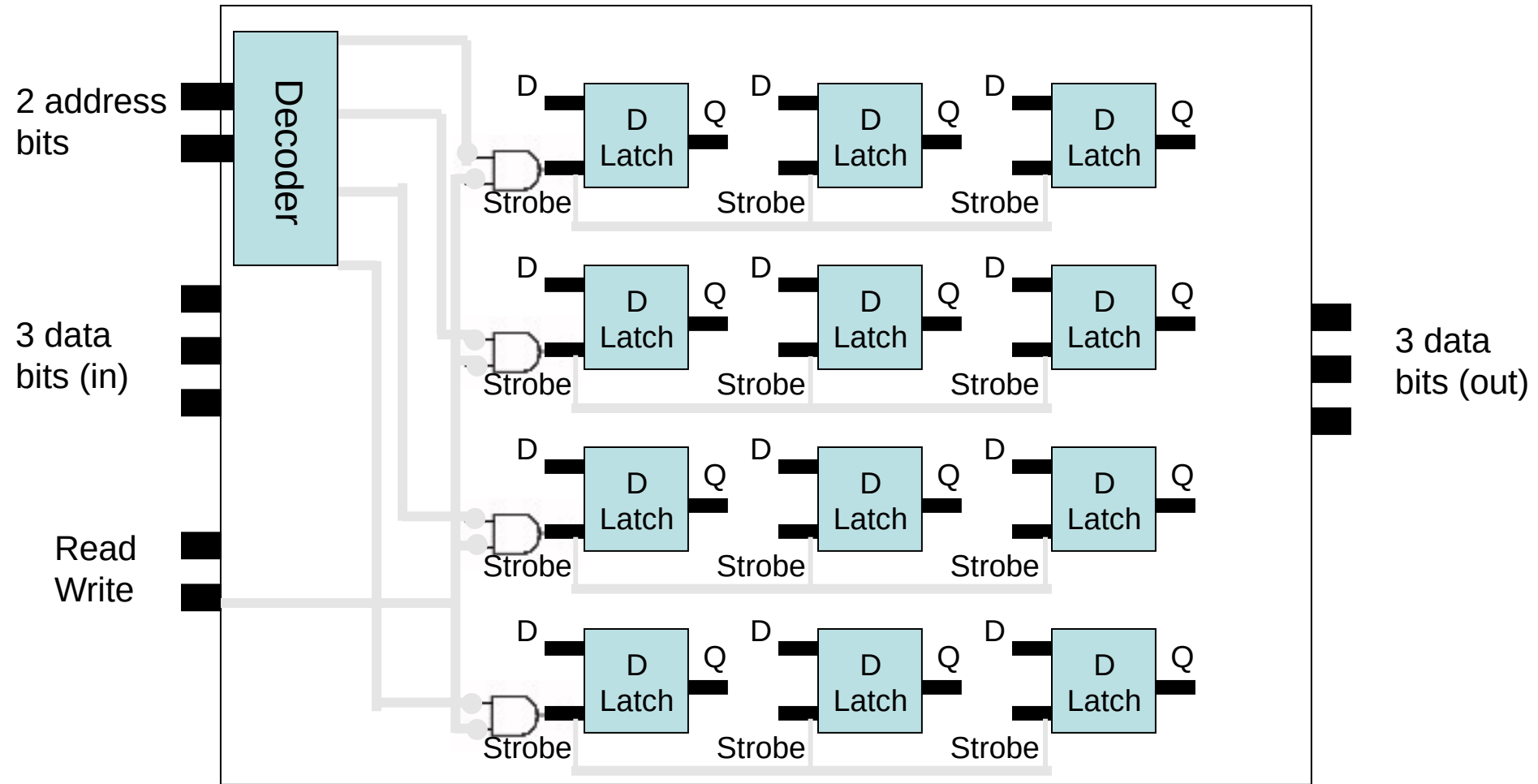
# A Random Access Memory (RAM)



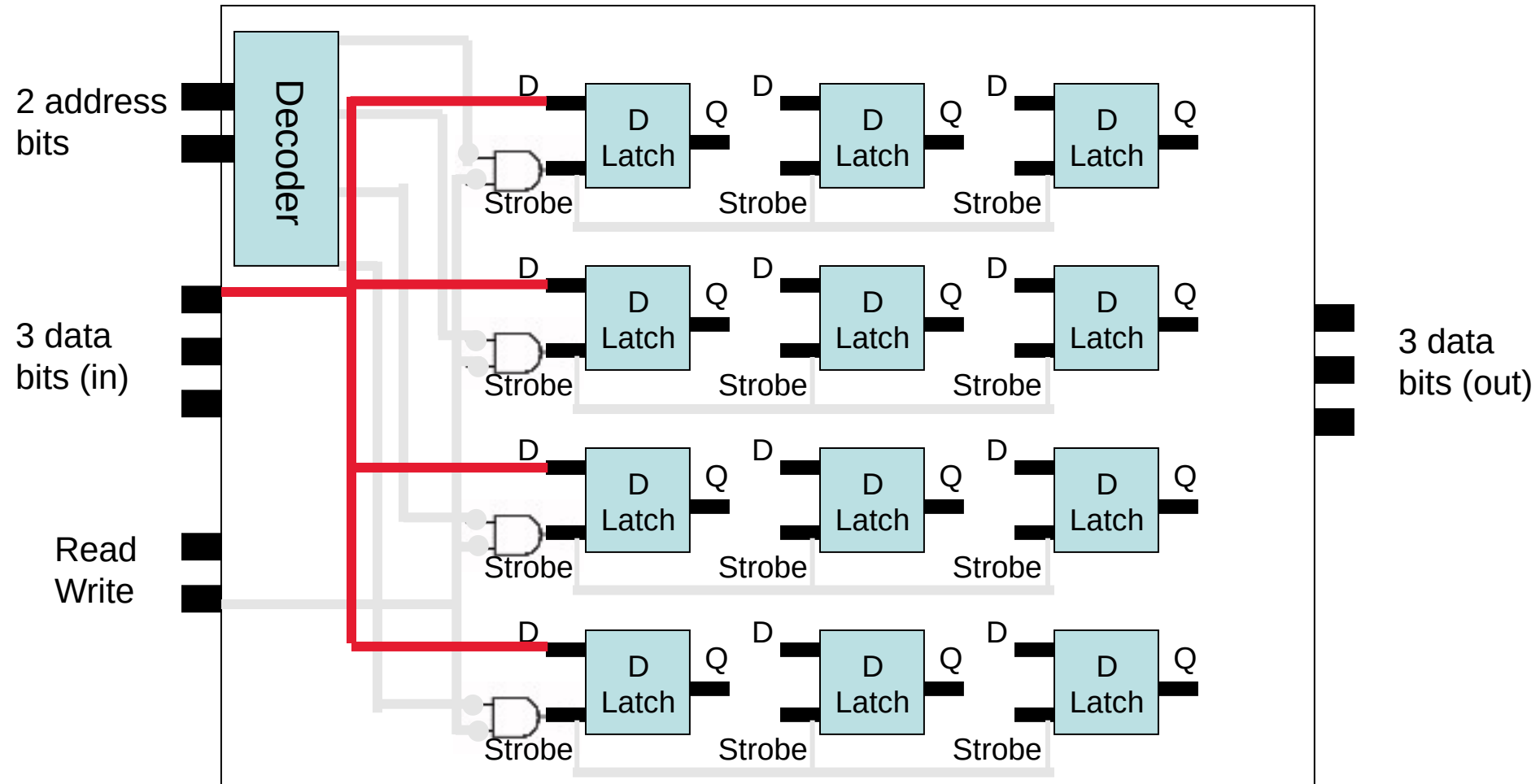
# A Random Access Memory (RAM)



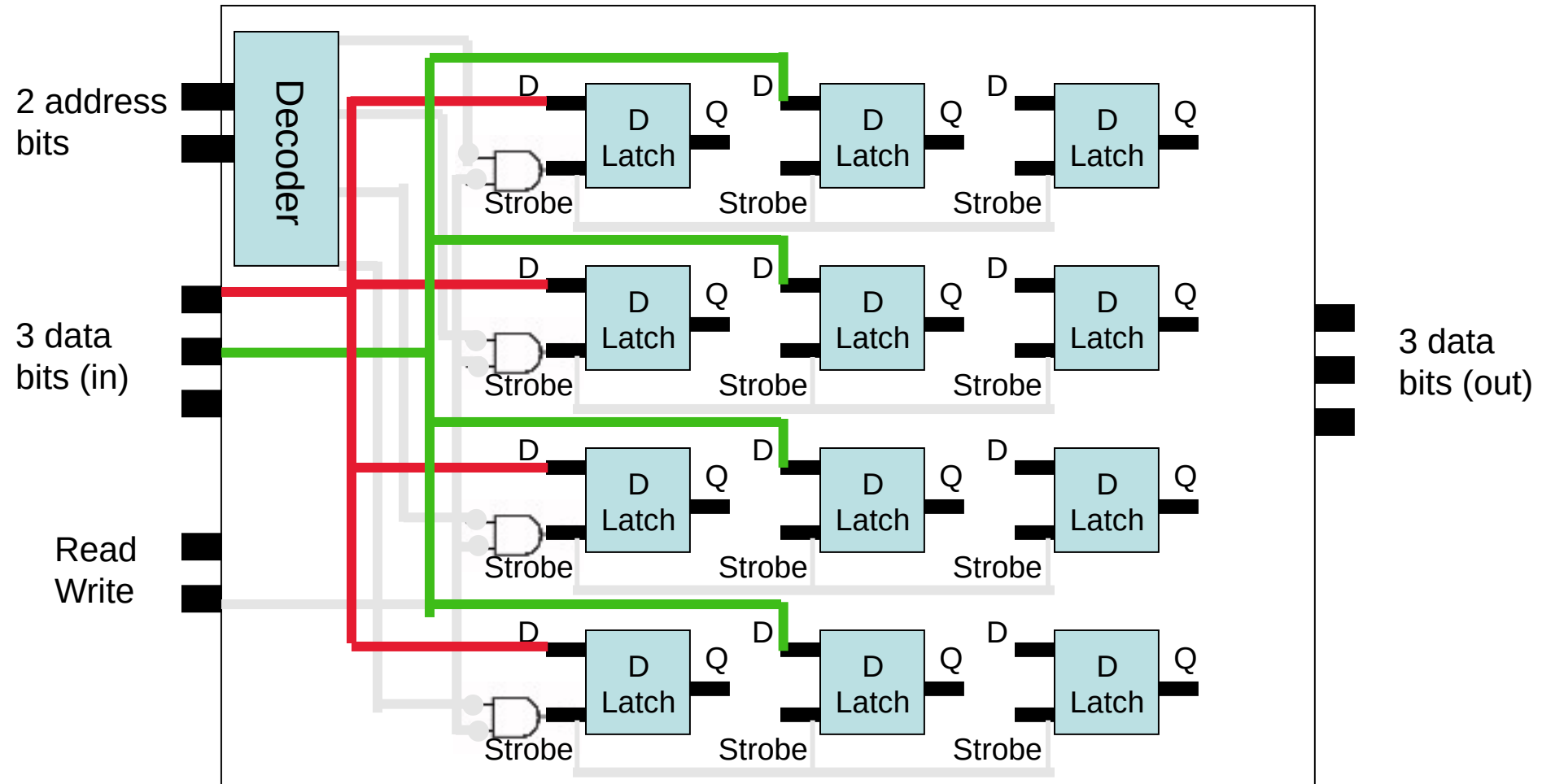
# A Random Access Memory (RAM)



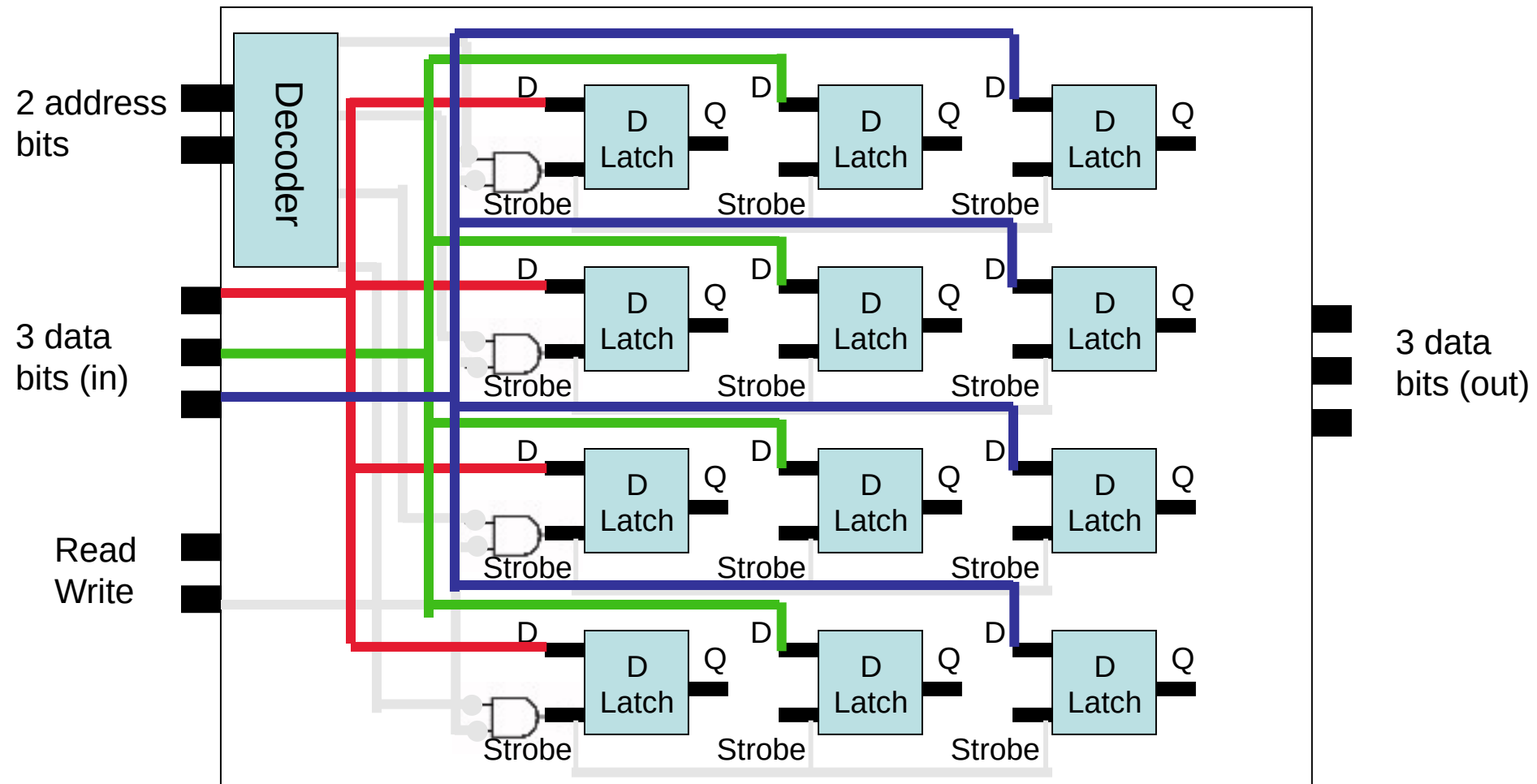
# A Random Access Memory (RAM)



# A Random Access Memory (RAM)



# A Random Access Memory (RAM)

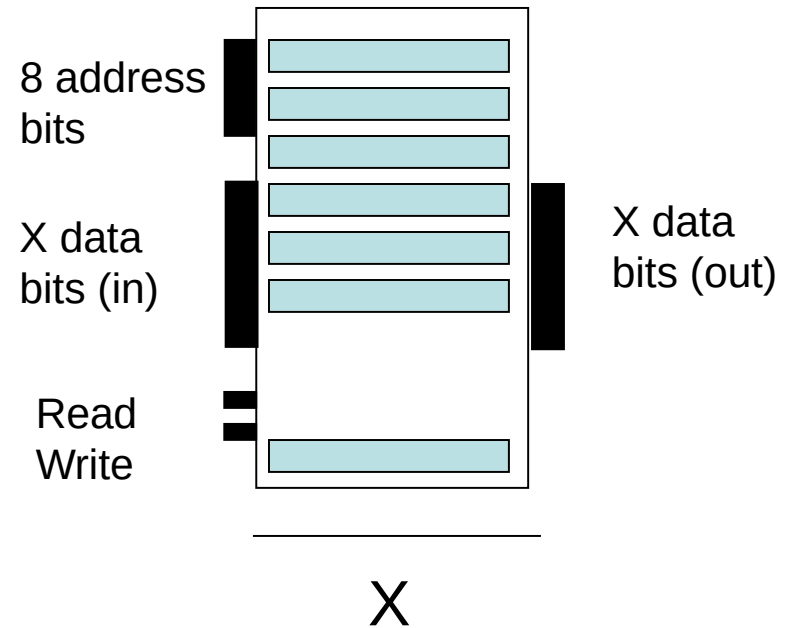
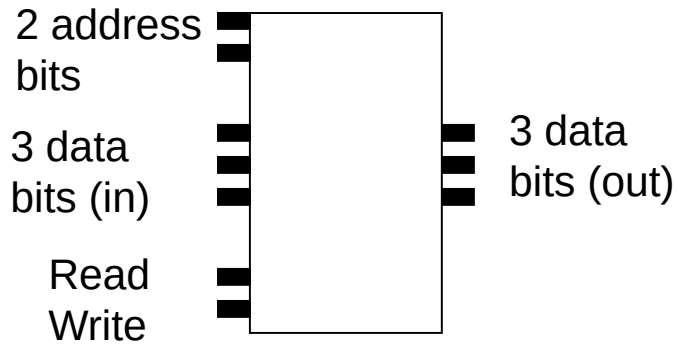




# Small Memory, “Big” Memory...

---

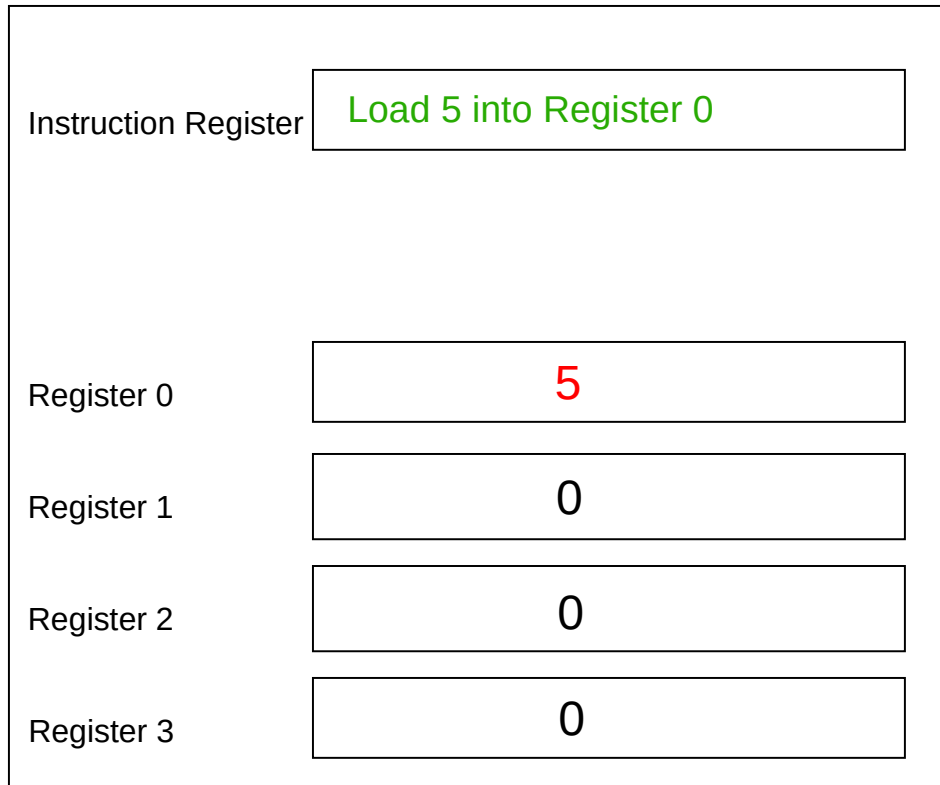
---



# A “Calculator”

Instruction Register	Load 5 into Register 0
Register 0	0
Register 1	0
Register 2	0
Register 3	0

# A “Calculator”



# A “Calculator”

---

---

Instruction Register

Register2=Register0+Register1

Register 0

5

Register 1

0

Register 2

0

Register 3

0

# A “Calculator”

---

---

Instruction Register

Register2=Register0+Register1

Register 0

5

Register 1

0

Register 2

5

Register 3

0

# A “Calculator”

Instruction Register

Register2=Register0+Register2

Register 0

5

Register 1

0

Register 2

5

Register 3

0

# A “Calculator”

Instruction Register

Register2=Register0+Register2

Register 0

5

Register 1

0

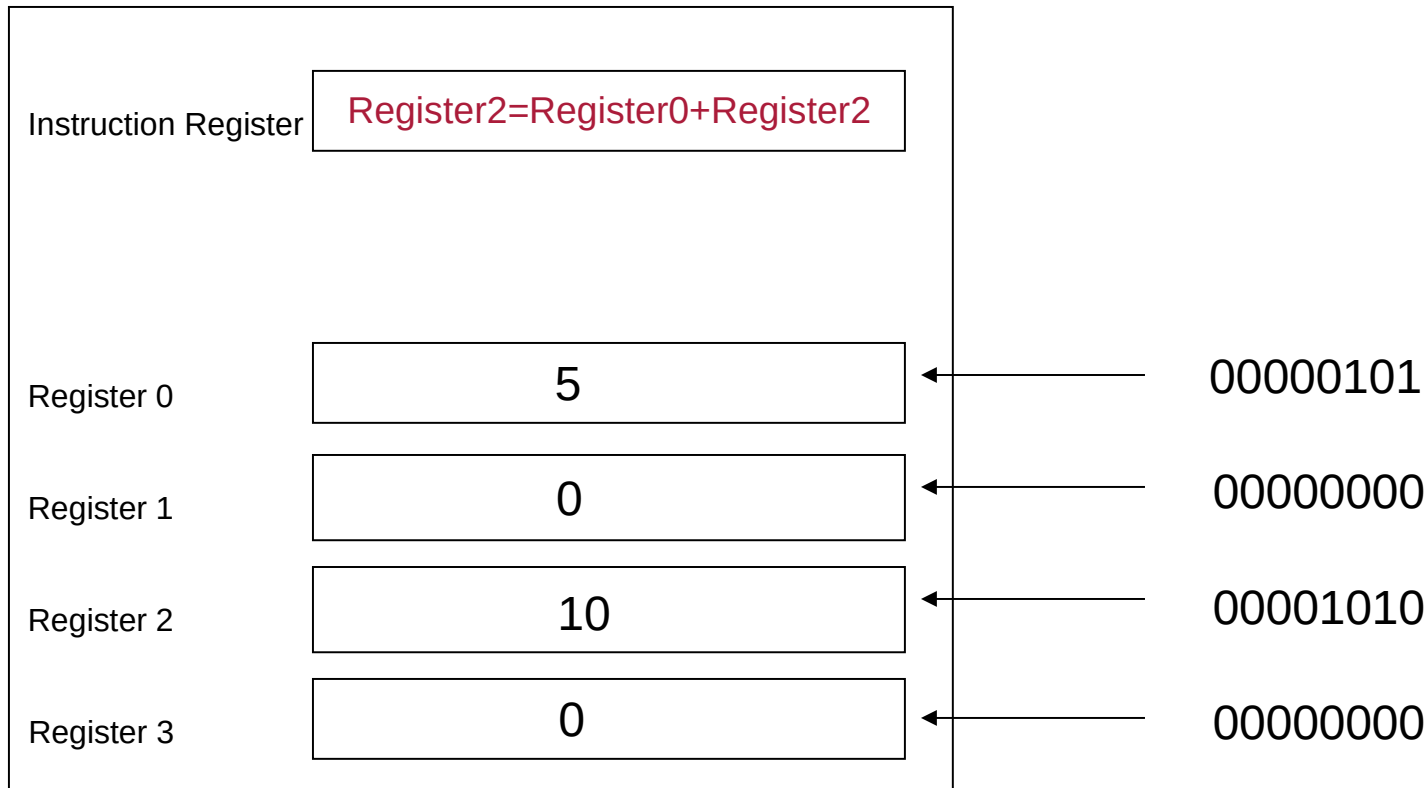
Register 2

10

Register 3

0

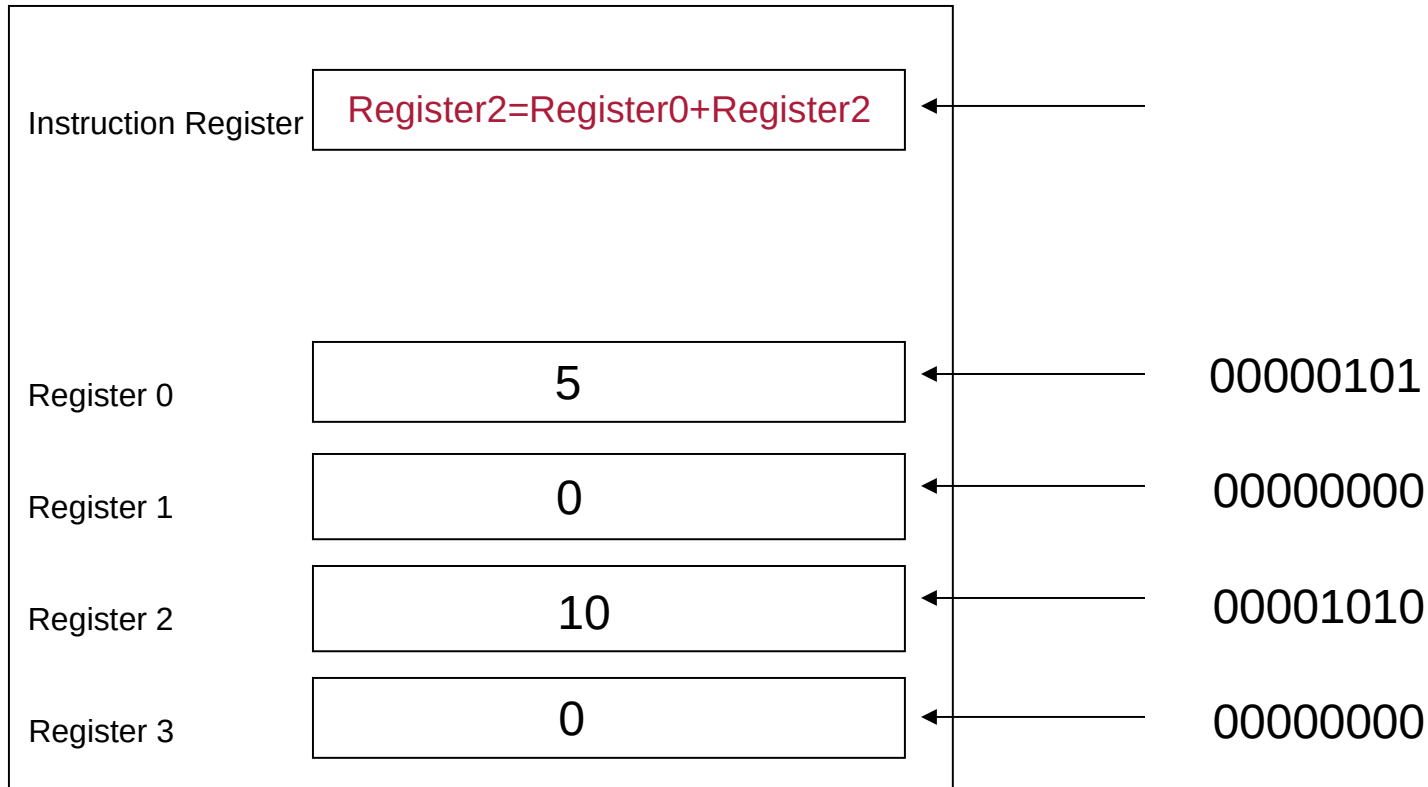
# A “Calculator”





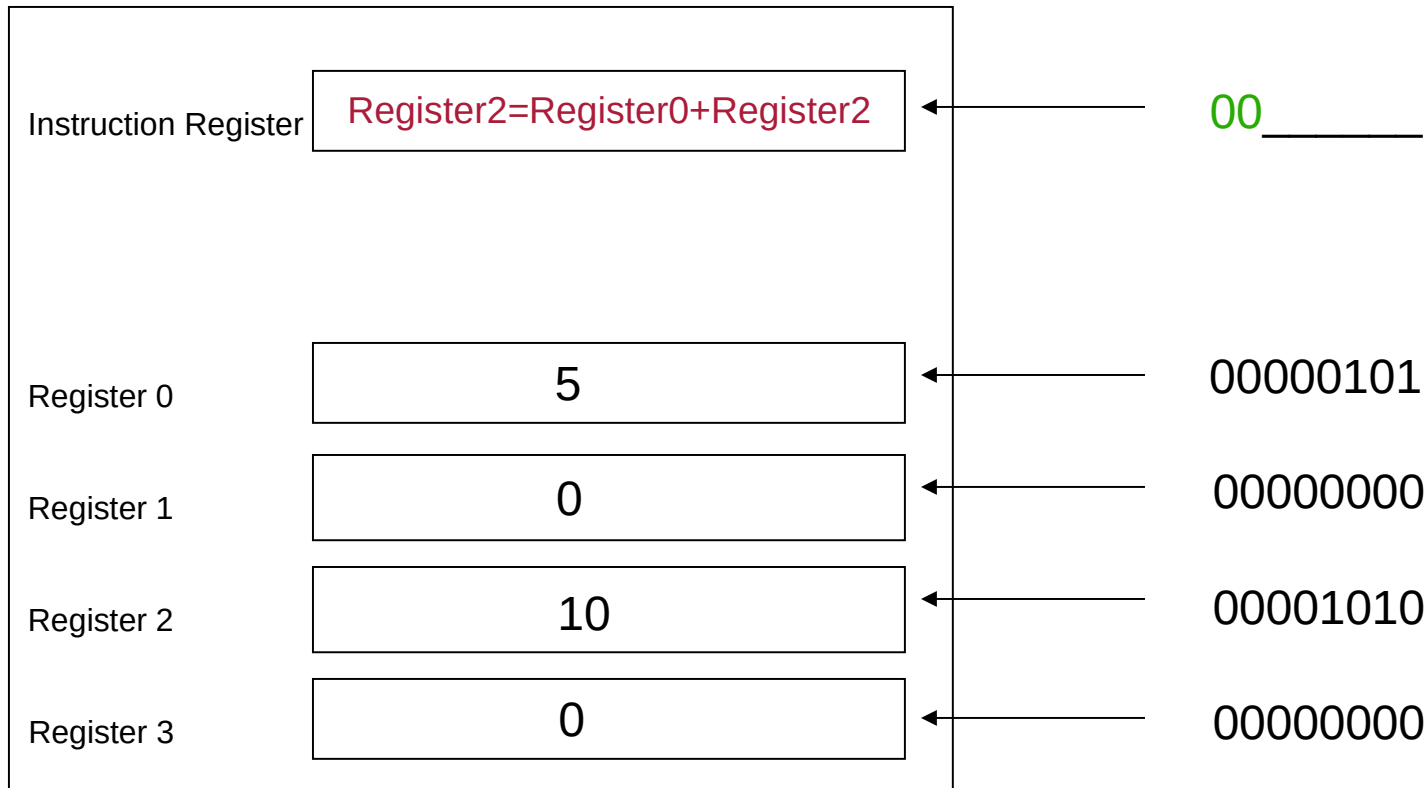
# A “Calculator”

00	add
01	subtract
10	multiply
11	divide



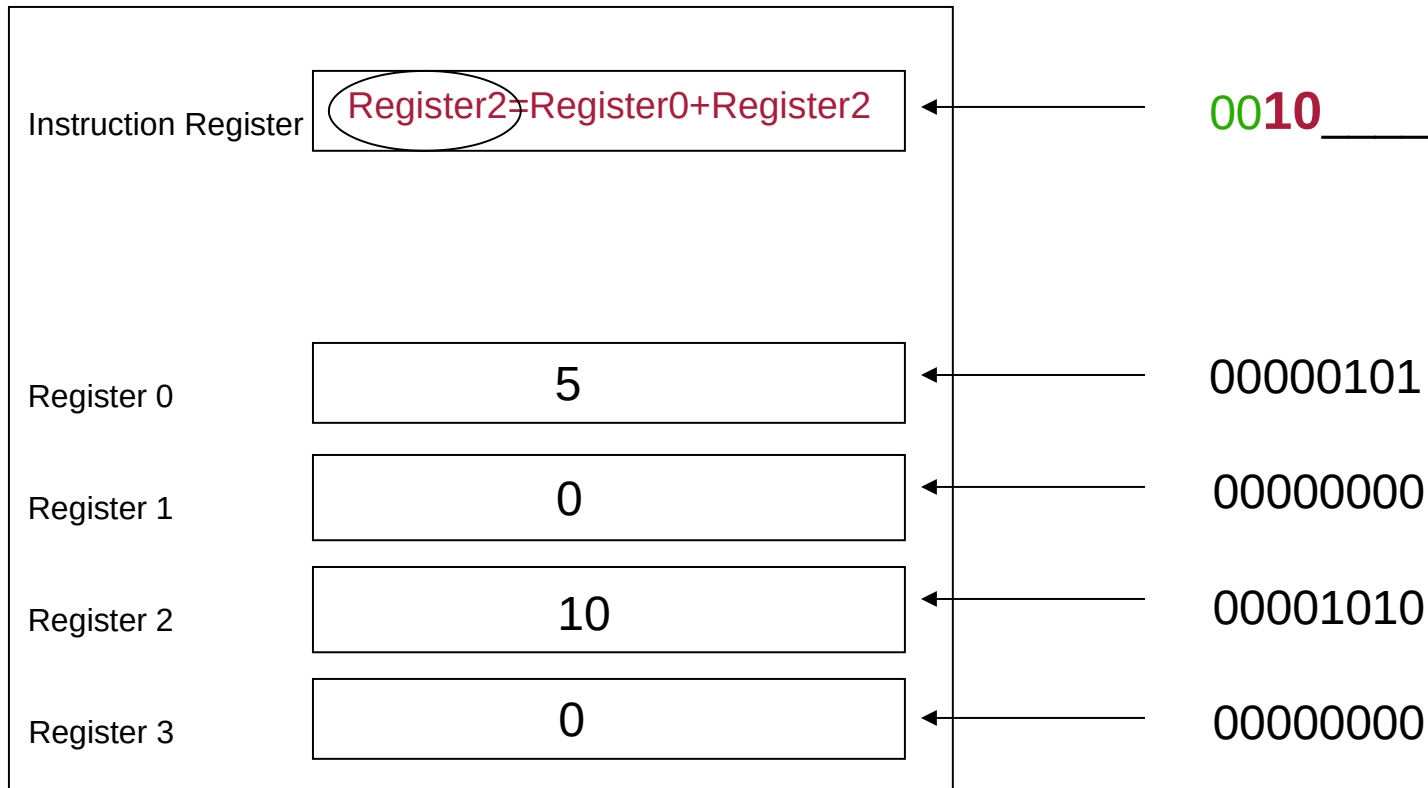
# A “Calculator”

00	add
01	subtract
10	multiply
11	divide



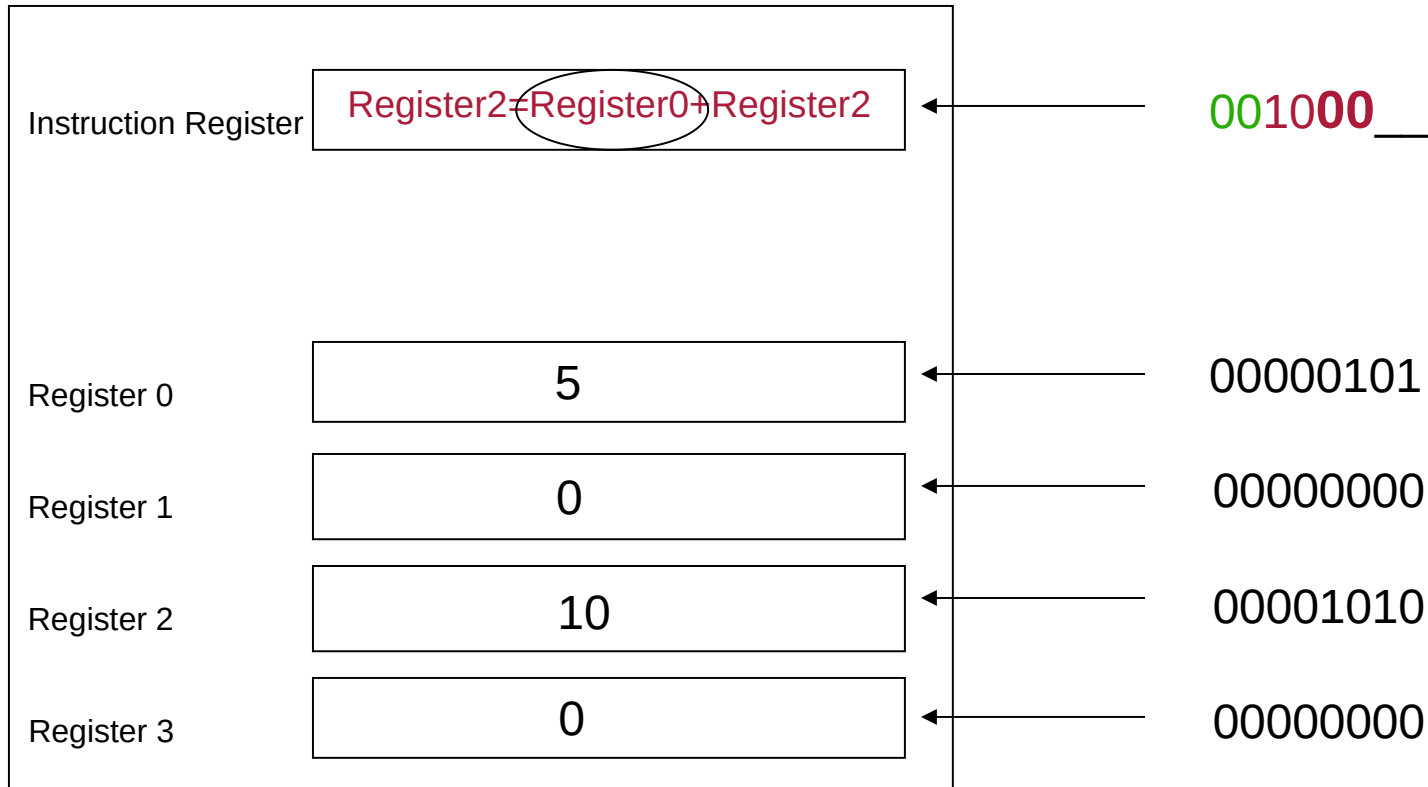
# A “Calculator”

00	add
01	subtract
10	multiply
11	divide



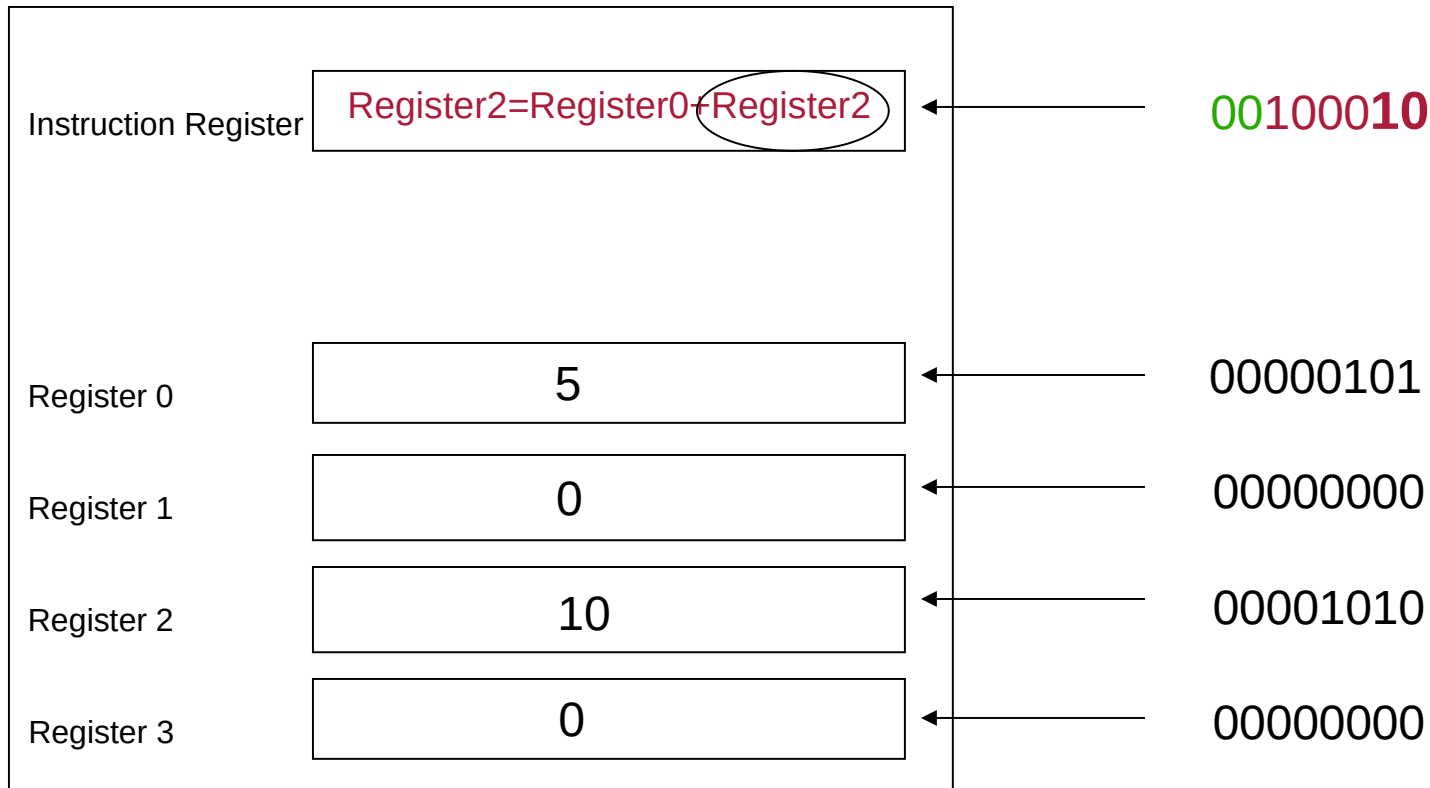
# A “Calculator”

00	add
01	subtract
10	multiply
11	divide



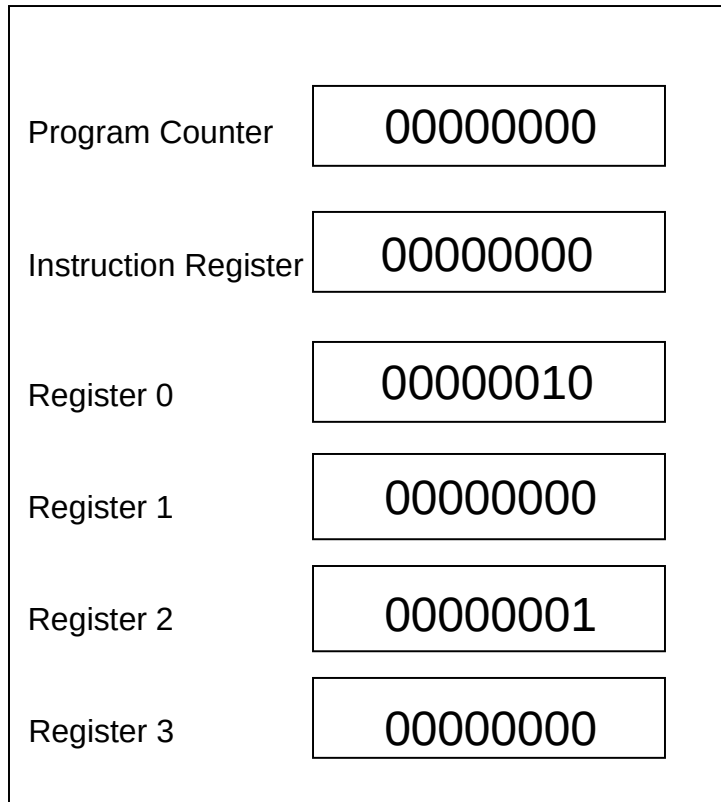
# A “Calculator”

00	add
01	subtract
10	multiply
11	divide

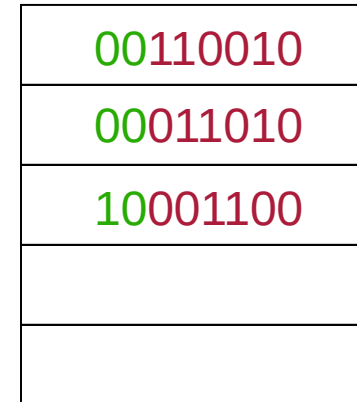


# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)



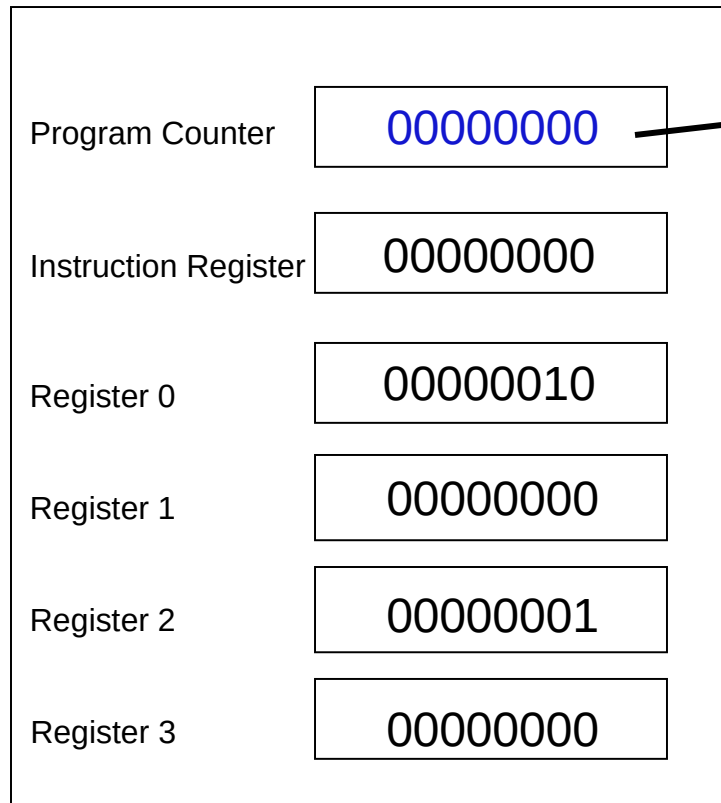
Memory

## Memory Location

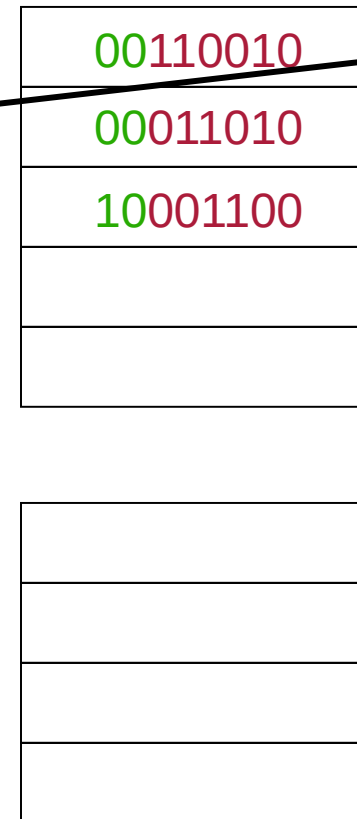
Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)



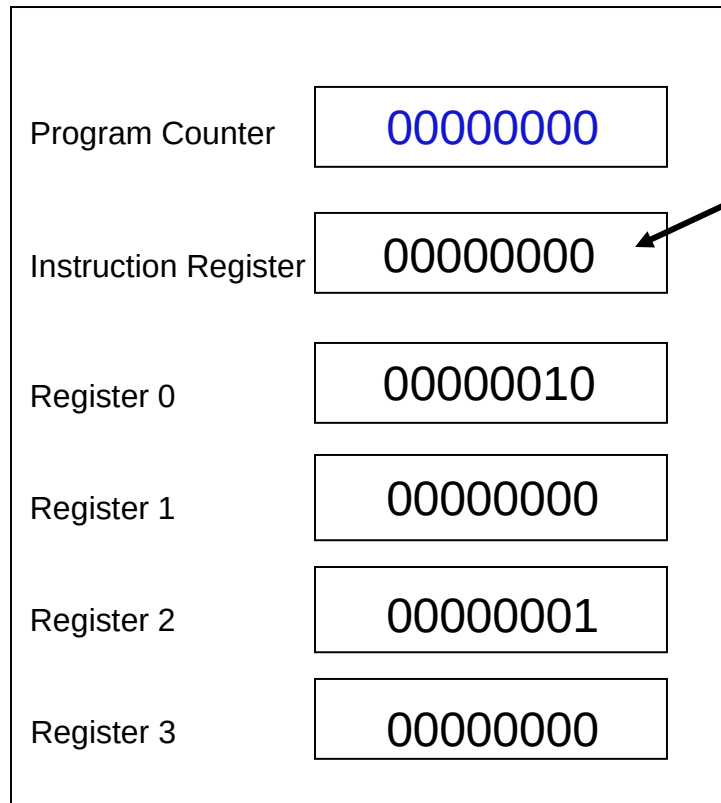
Memory

Memory Location

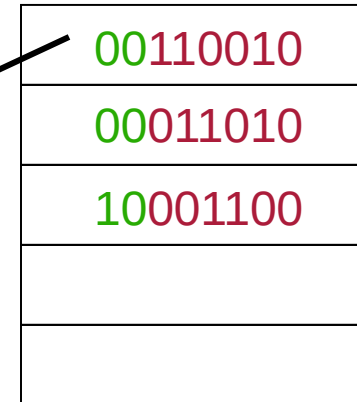
Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)



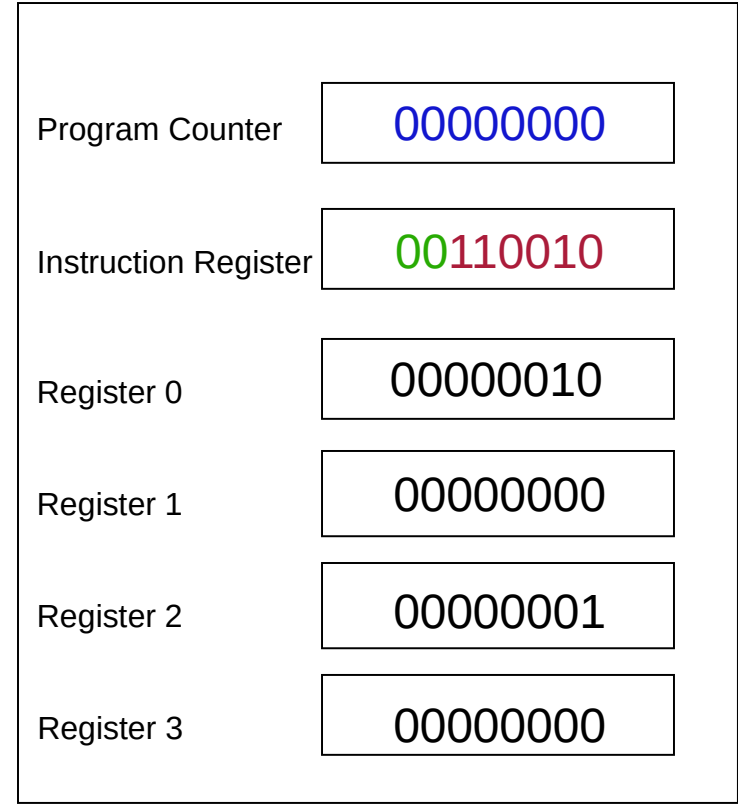
Memory

## Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255




add  
subtract  
multiply  
divide

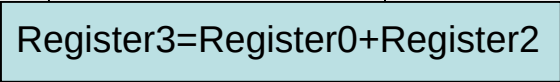


# Central Processing Unit (CPU)

Memory Location	
Binary	Base 10
00110010	0
00011010	1
10001100	2
	3
	4
11111111	255

# Memory


add  
subtract  
multiply  
divide



Register3=Register0+Register2

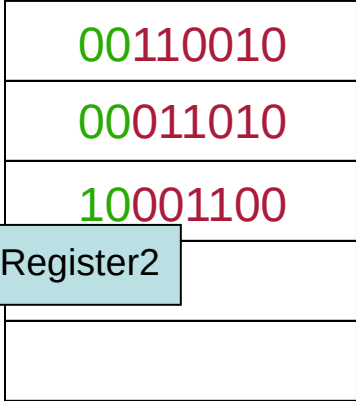


### Memory Location


add  
subtract  
multiply  
divide



# Central Processing Unit (CPU)



# Memory

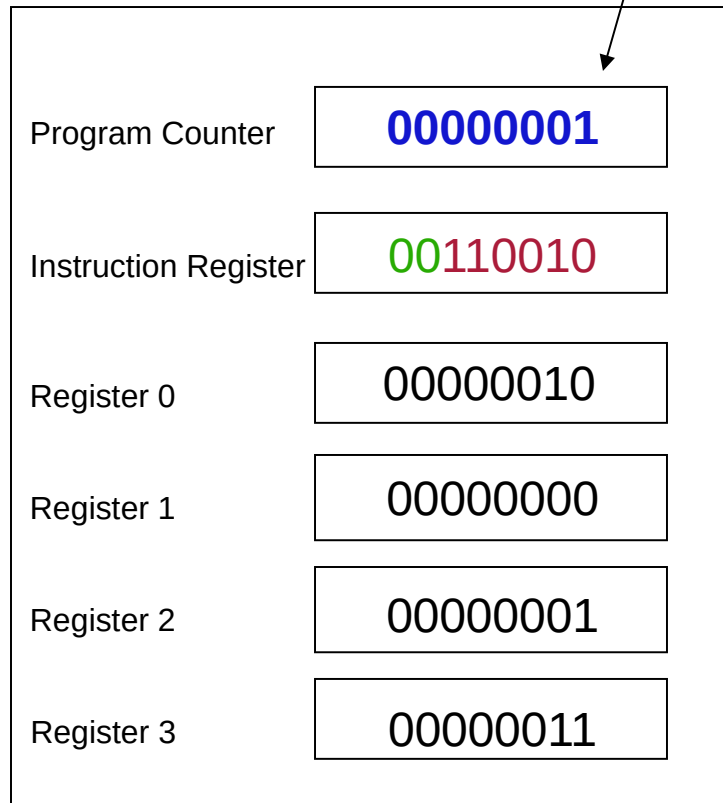
## Memory Location

[illegible]

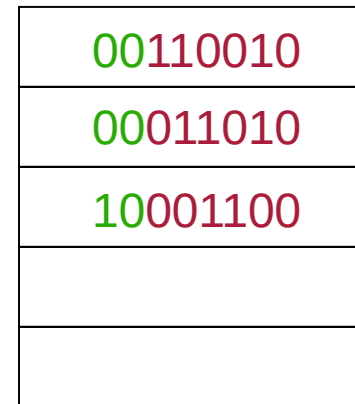
# A Computer!

00 add  
01 subtract  
10 multiply  
11 divide

Program Counter Incremented



Central Processing Unit (CPU)



Memory

Memory Location

Binary      Base 10

00000000      0

00000001      1

00000010      2

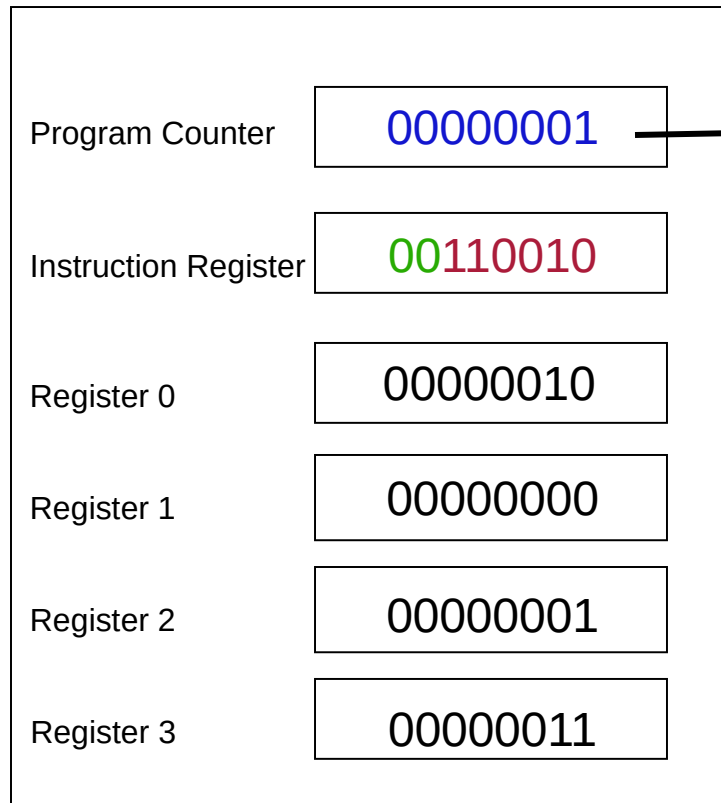
00000011      3

00000100      4

11111111      255

# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)



Memory

Memory Location

Binary      Base 10

00000000      0

00000001      1

00000010      2

00000011      3

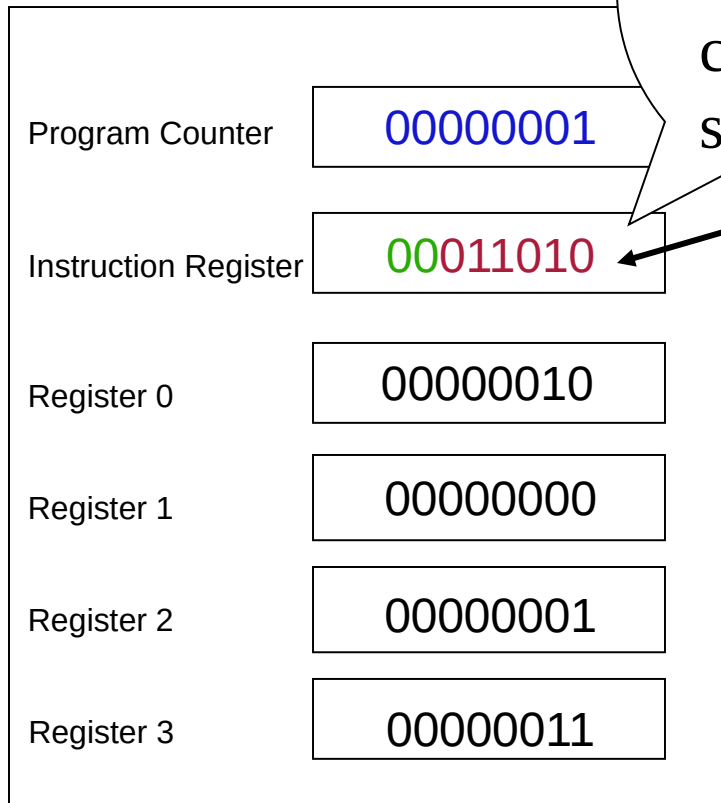
00000100      4

11111111      255

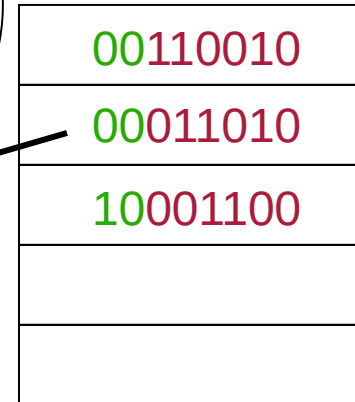
# A Computer!

00 add  
01 subtract  
10 multiply  
11 divide

What is  
this  
command  
saying?



Central Processing Unit (CPU)



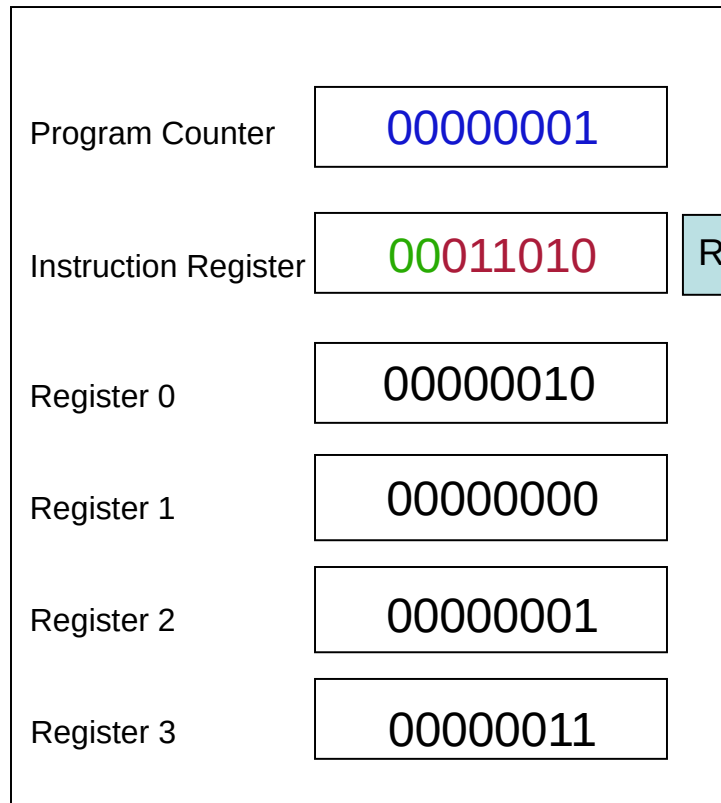
Memory

## Memory Location

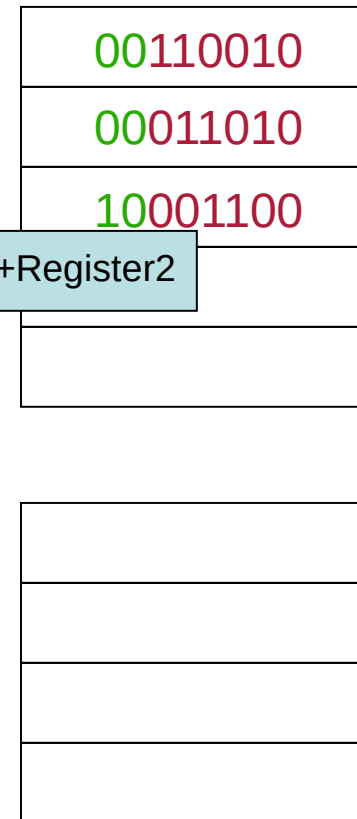
Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)



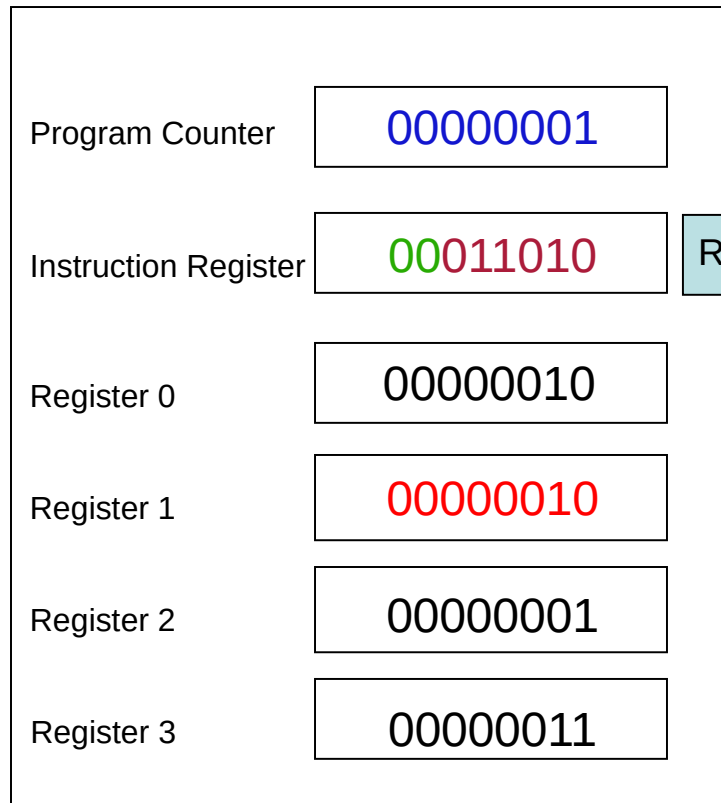
Memory

## Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

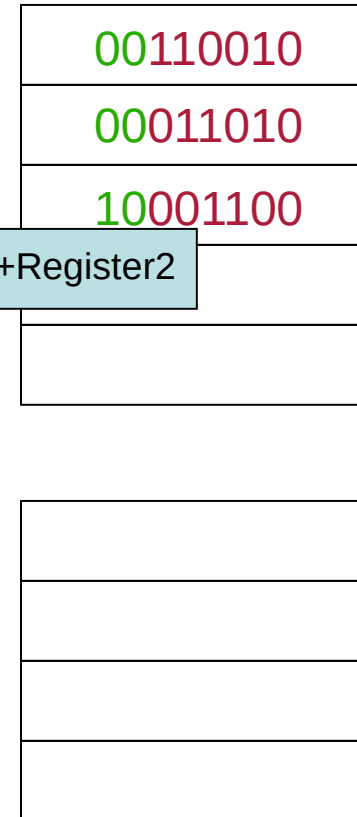
# A Computer!

00	add
01	subtract
10	multiply
11	divide



Central Processing Unit (CPU)

Register1=Register2+Register2



Memory

Memory Location

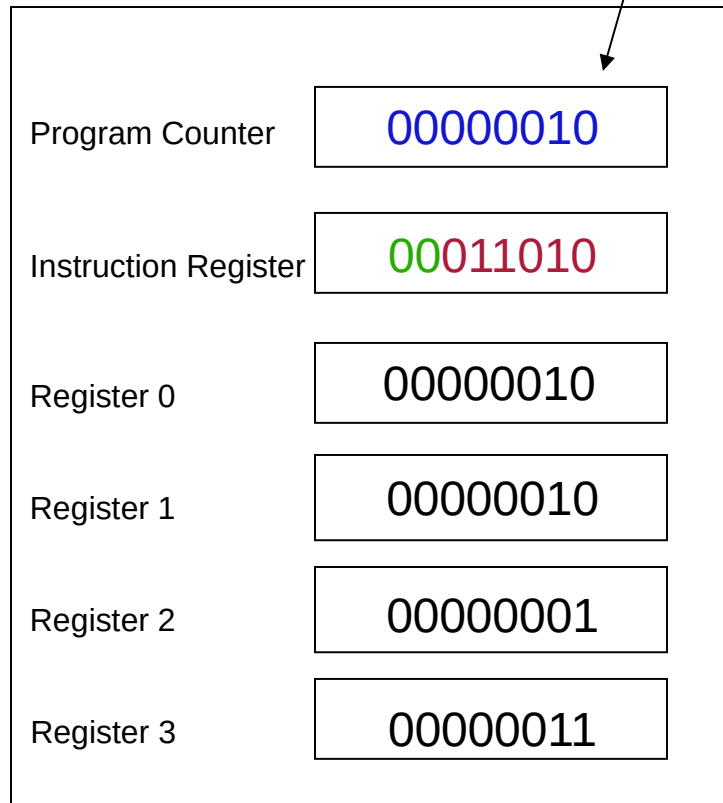
Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255



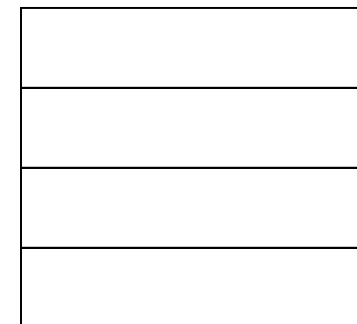
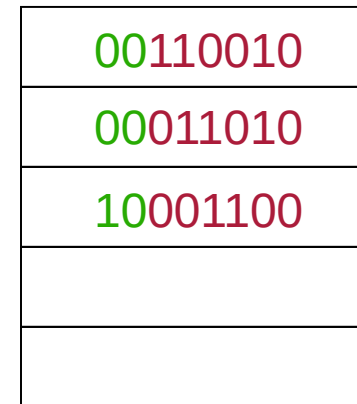
# A Computer!

00 add  
01 subtract  
10 multiply  
11 divide

Program Counter Incremented



Central Processing Unit (CPU)



Memory

Memory Location

Binary      Base 10

00000000      0

00000001      1

00000010      2

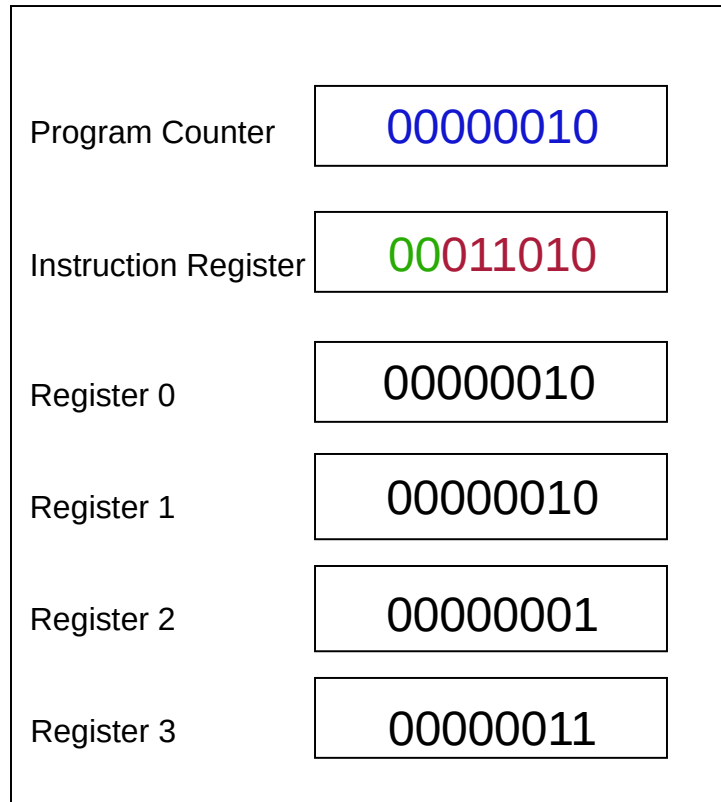
00000011      3

00000100      4

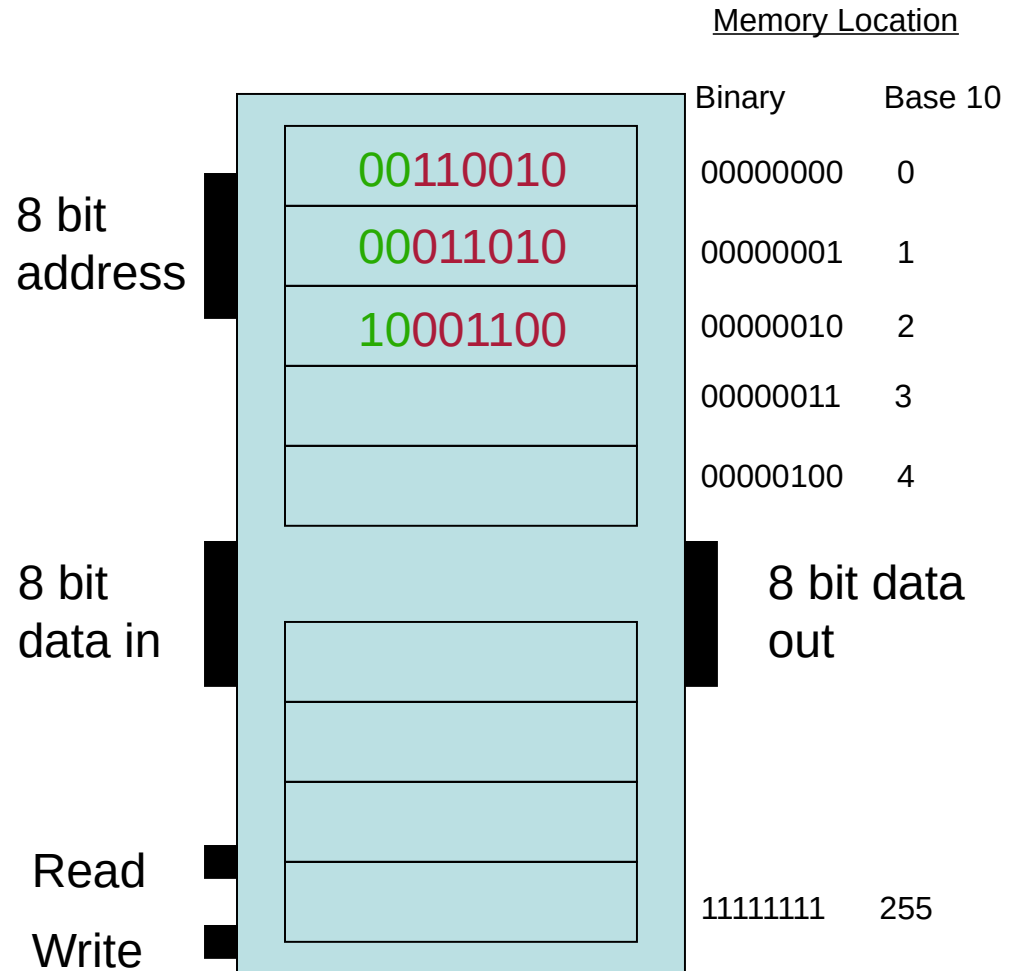
11111111      255

# A Computer!

00	add
01	subtract
10	multiply
11	divide



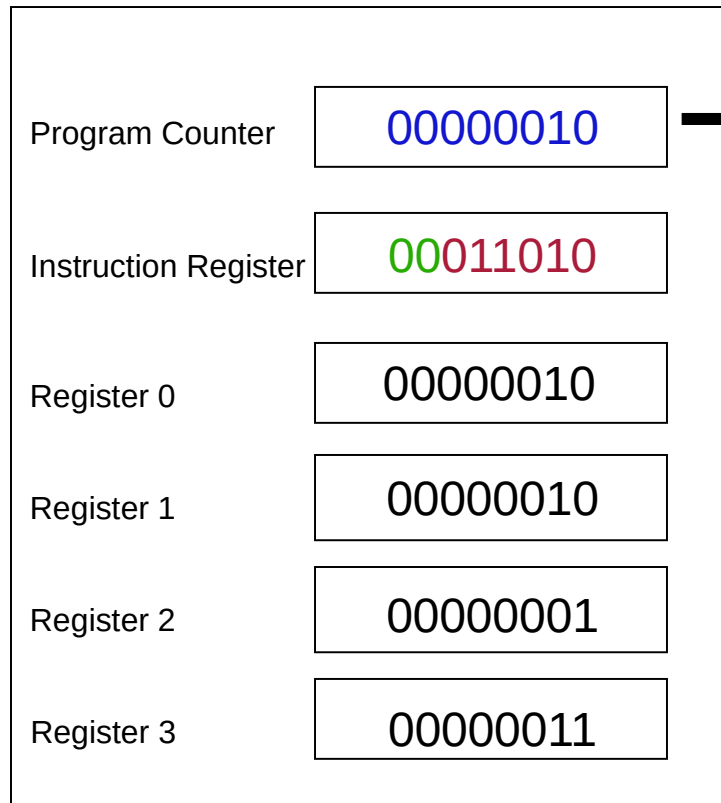
Central Processing Unit (CPU)



Memory

# A Computer!

00	add
01	subtract
10	multiply
11	divide

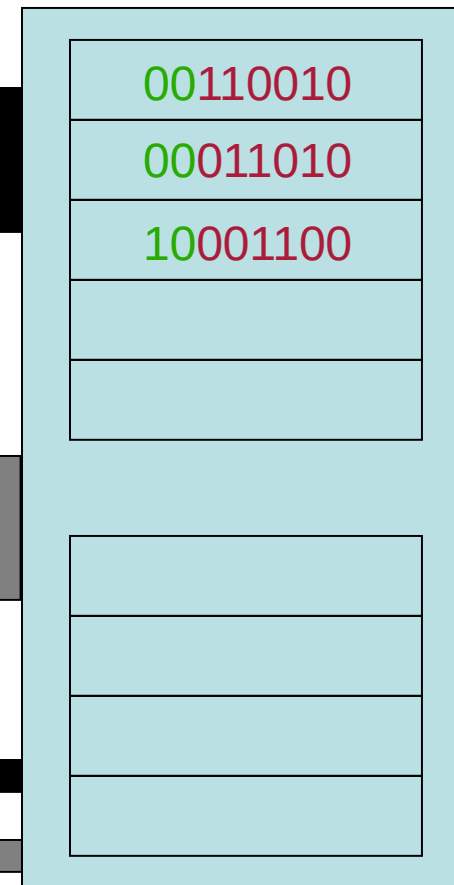


Central Processing Unit (CPU)

8 bit  
address

8 bit  
data in

1  
Read  
Write



Memory

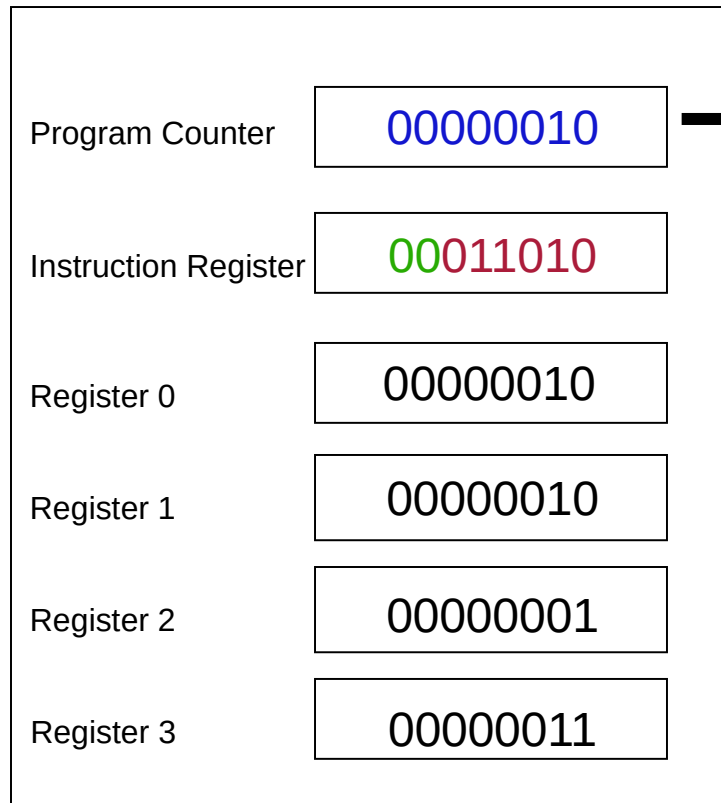
Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

8 bit data  
out

# A Computer!

00	add
01	subtract
10	multiply
11	divide

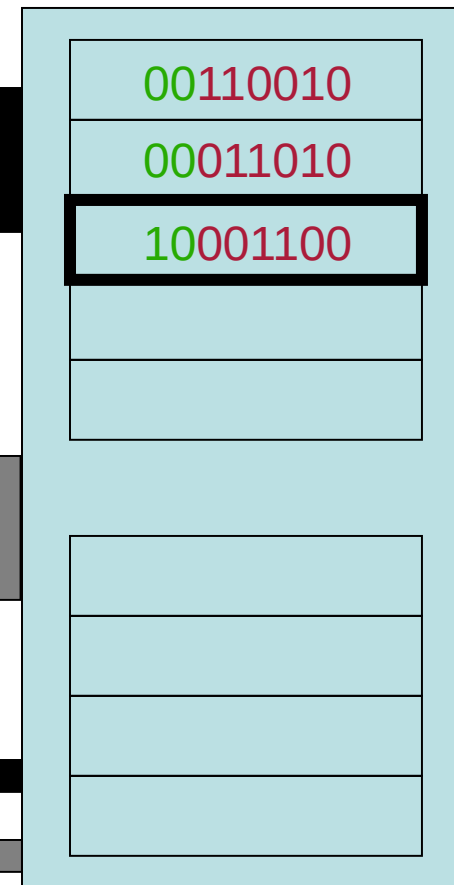


Central Processing Unit (CPU)

8 bit  
address

8 bit  
data in

1  
Read  
Write



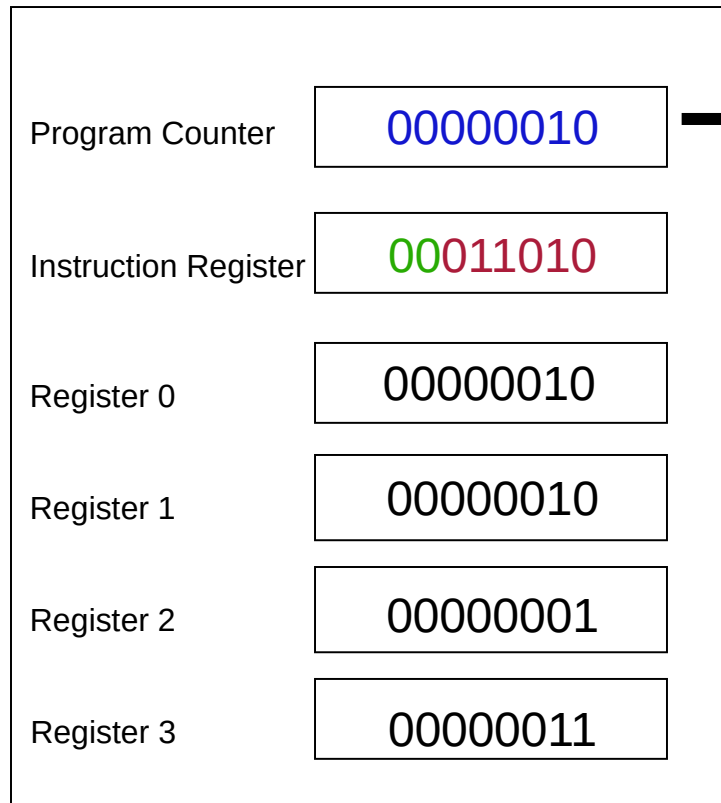
Memory

Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
11111111	255

# A Computer!

00	add
01	subtract
10	multiply
11	divide

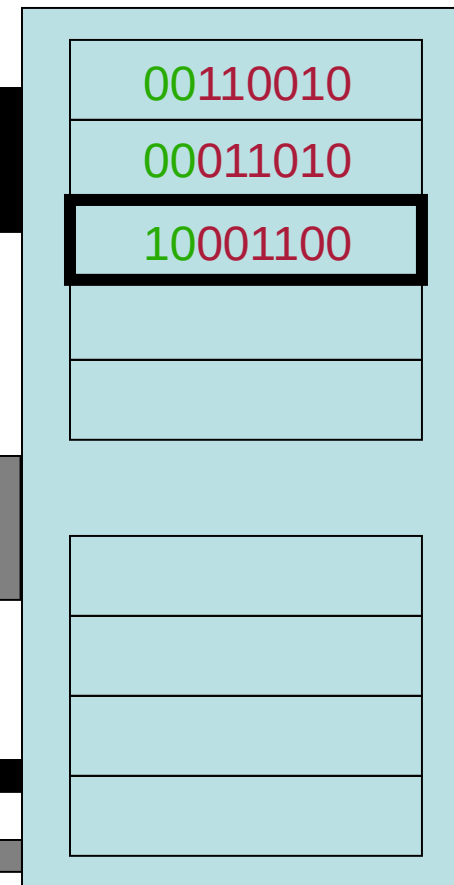


Central Processing Unit (CPU)

8 bit  
address

8 bit  
data in

1  
Read  
Write



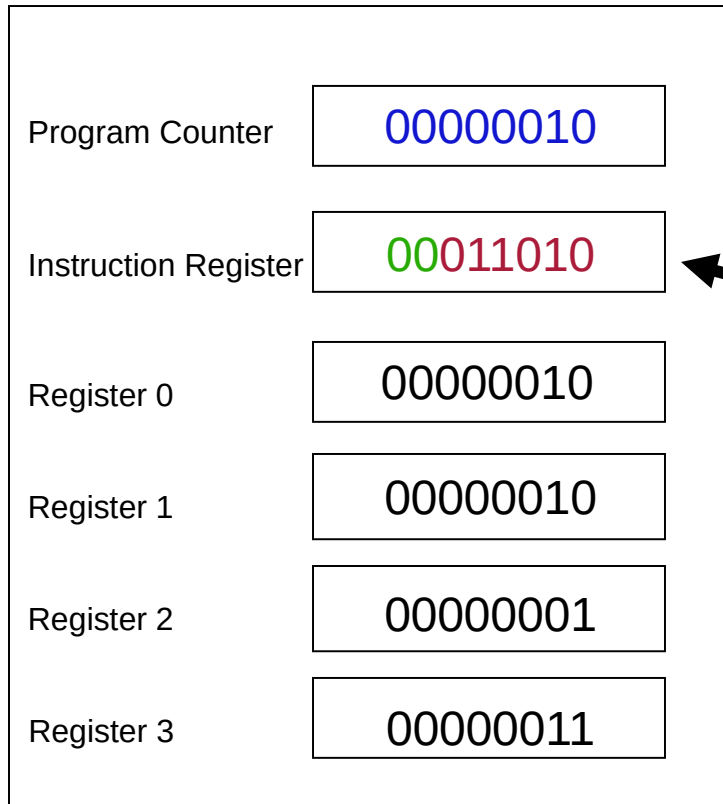
Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
...	...
11111111	255

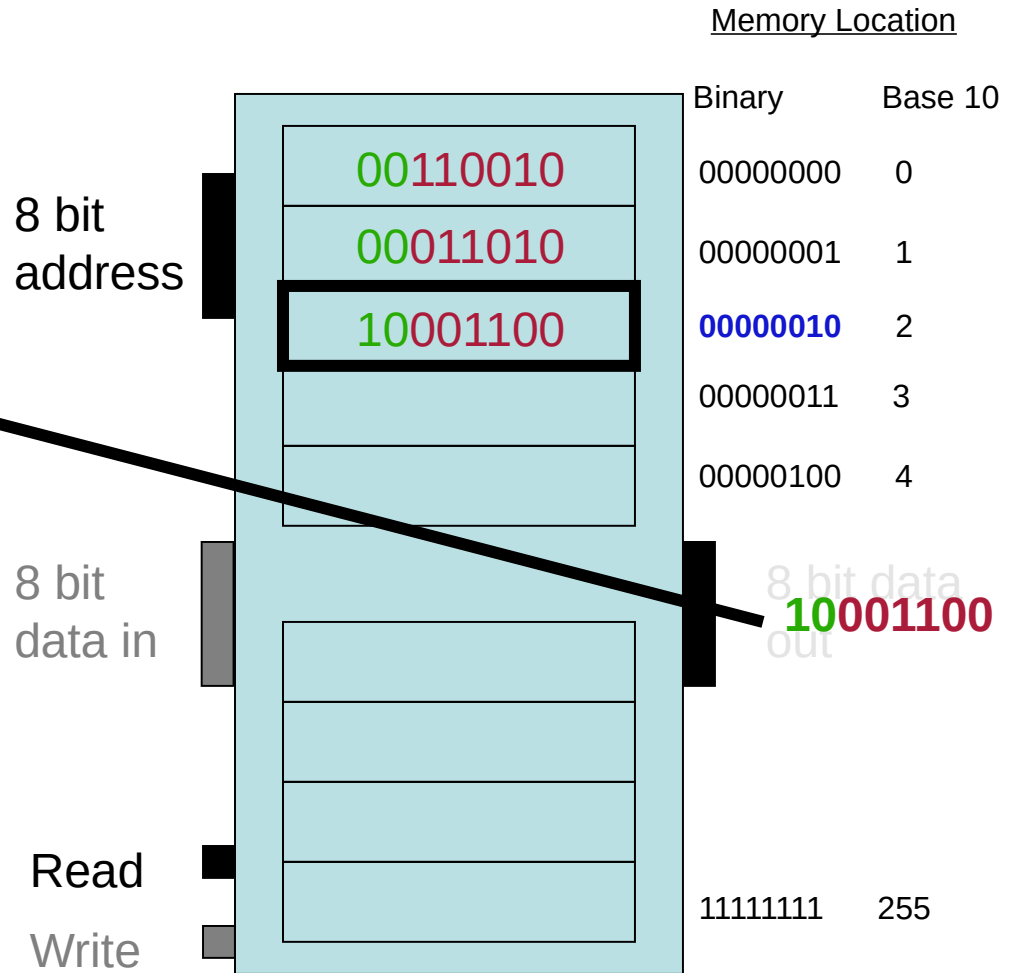
8 bit data  
out  
10001100

Memory


00	add
01	subtract
10	multiply
11	divide



# Central Processing Unit (CPU)

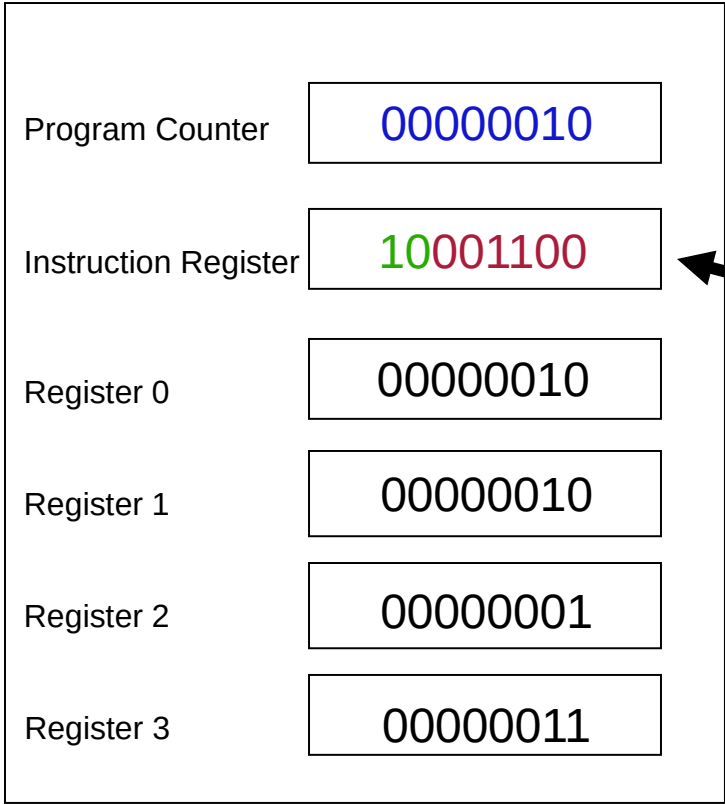


# Memory

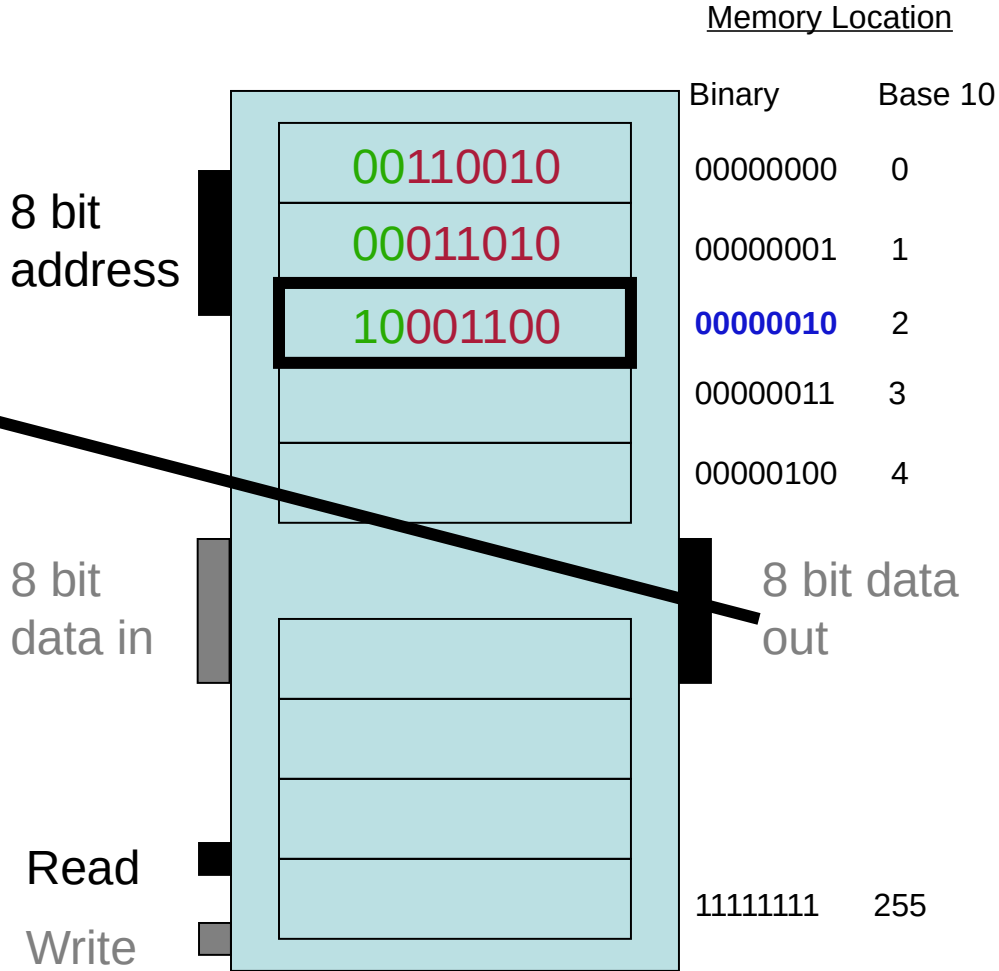
---

---

00	add
01	subtract
10	multiply
11	divide



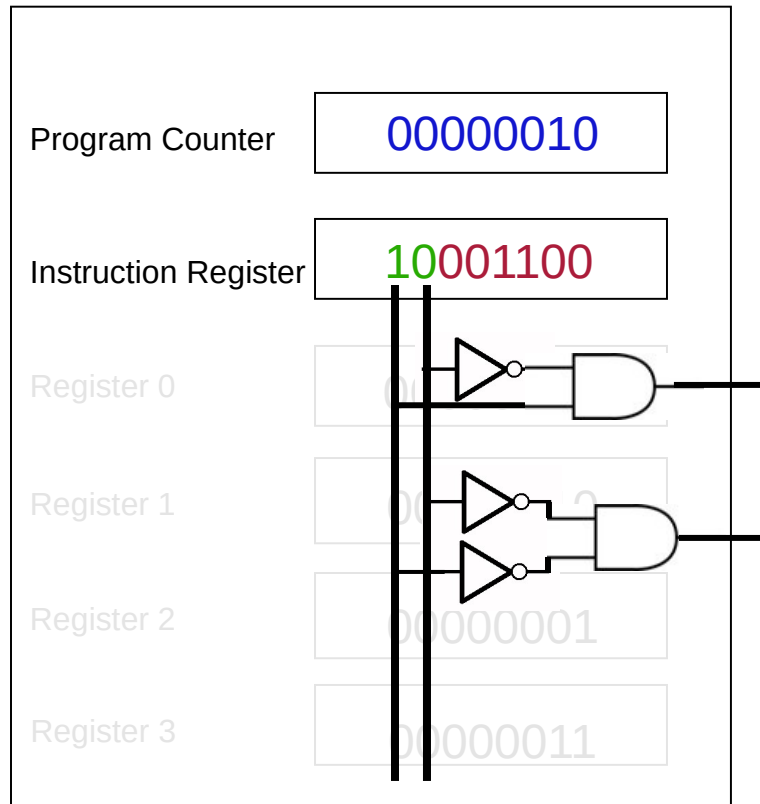
# Central Processing Unit (CPU)



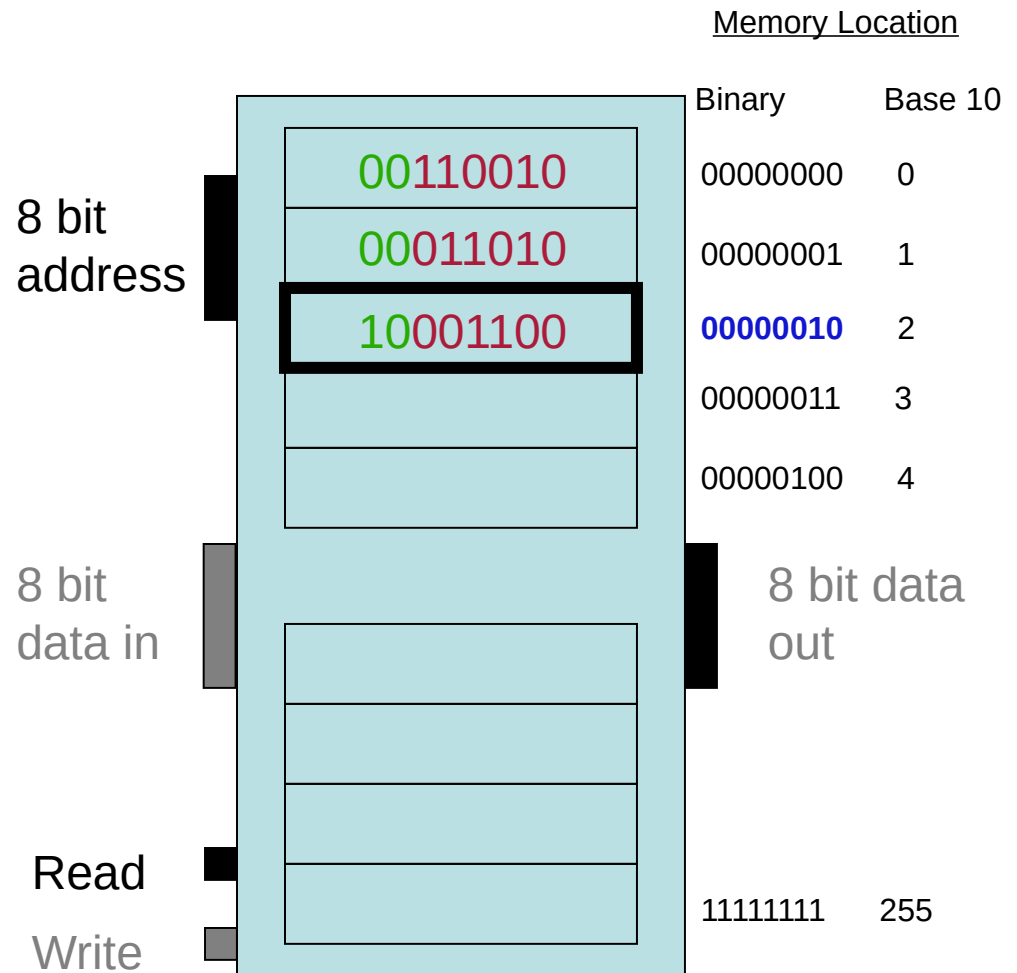
# Memory

# A Computer!

00 add  
01 subtract  
10 multiply  
11 divide



Central Processing Unit (CPU)

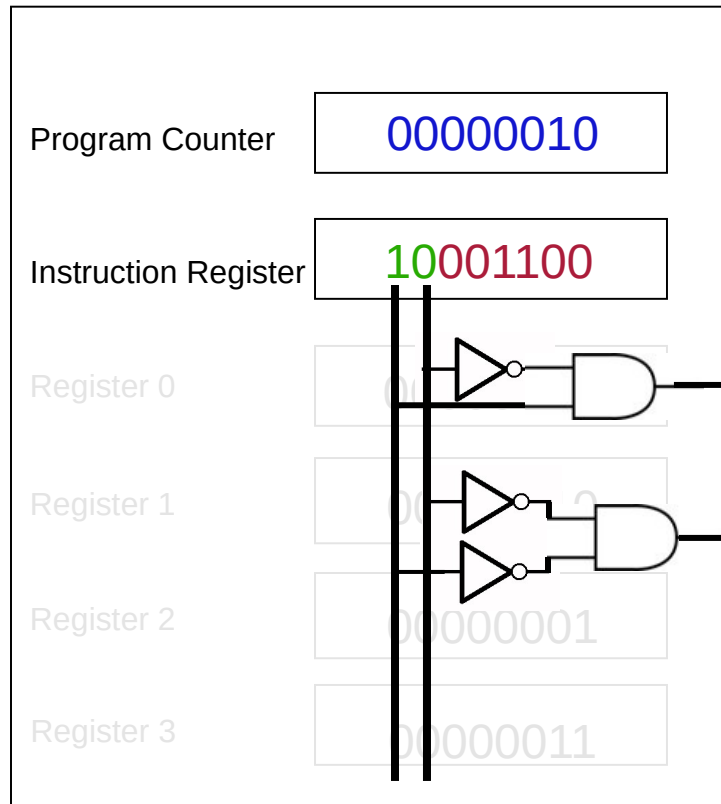


Memory



# A Computer!

00 add  
01 subtract  
10 multiply  
11 divide

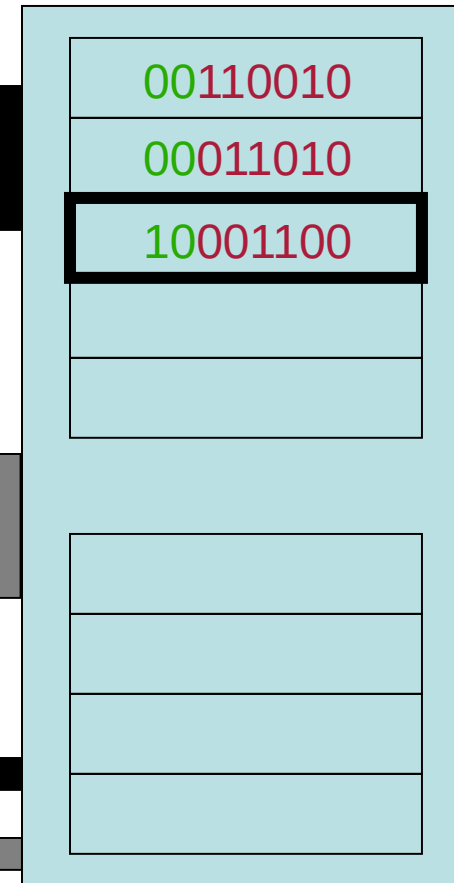


Central Processing Unit (CPU)

8 bit  
address

8 bit  
data in

Read  
Write



Memory

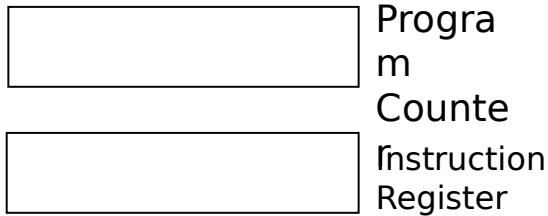
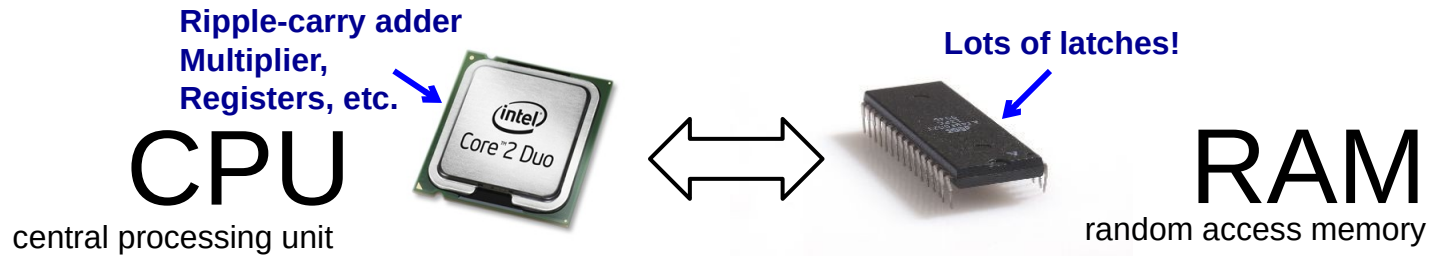
Memory Location

Binary	Base 10
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
...	...
11111111	255

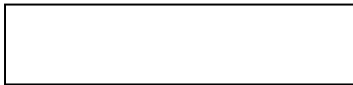
# The von Neumann "Architecture"



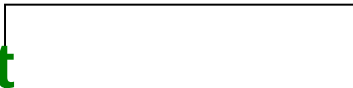
John von Neumann



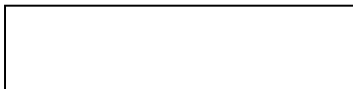
r0



r1

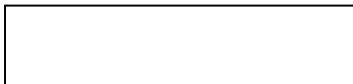


r2

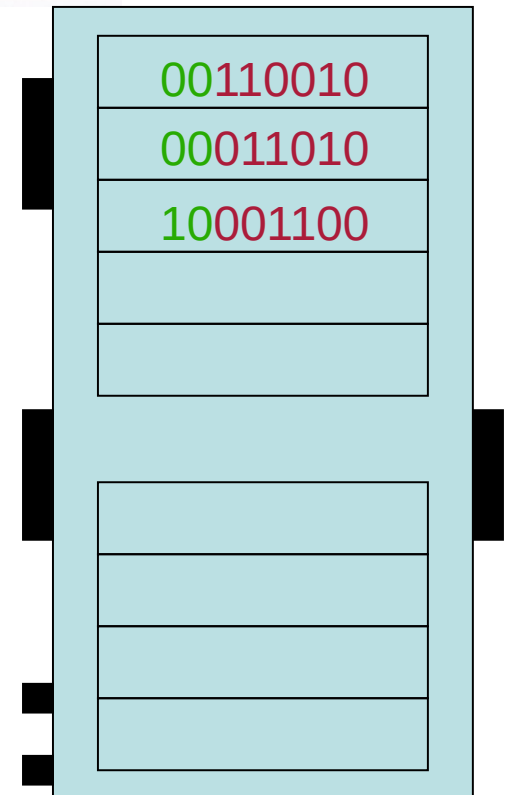


...

r15



**Large but  
slow memory**



**A few fast  
registers**







2006  
Intel Core 2 Duo  
3 GHz clock  
64-bit processor  
291 million transistors  
65 nm wires



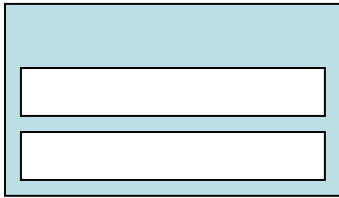
It doesn't  
look all that  
fast to me!





# A Short Aside...

CPU

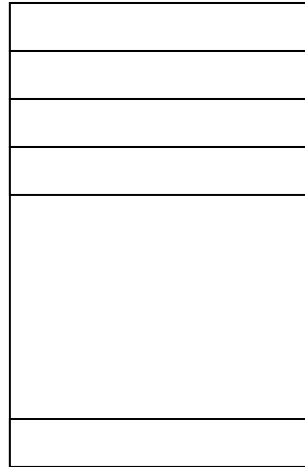


16 Registers (“Bytes”)

1 cycle

<10<sup>-9</sup> sec

Main Memory  
(RAM)



10<sup>9</sup> “Bytes” of  
memory

200 cycles

<10<sup>-7</sup> sec

Disk Drive



10<sup>12</sup> “Bytes” of  
memory

? cycles

<10<sup>-2</sup> sec

Actual  
time:

If “cycle”  
= 1 sec

1 sec