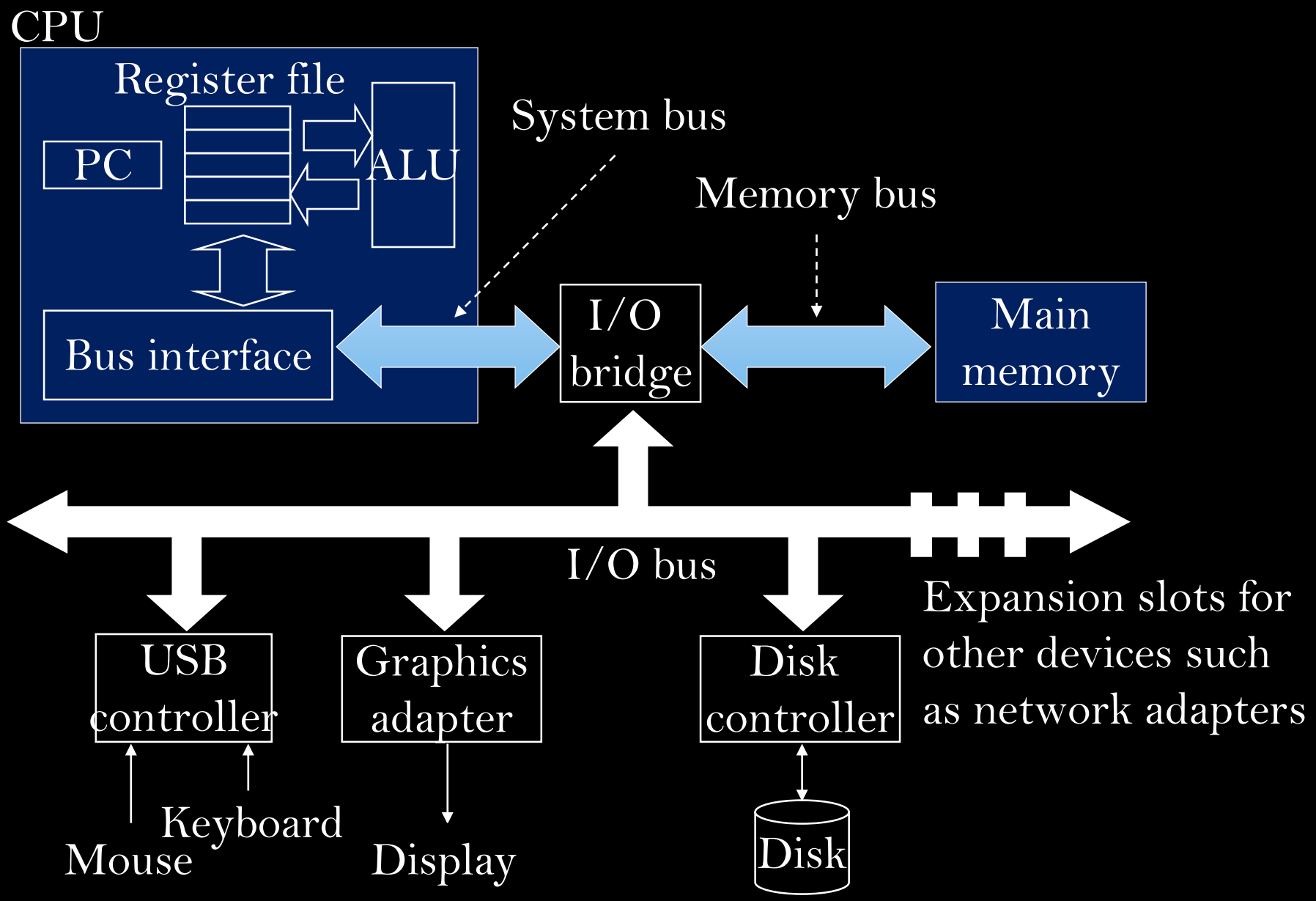
TRUONG DAI HOC KHOA HOC TU NHIEN
KHTN
TP. HO CHI MINH

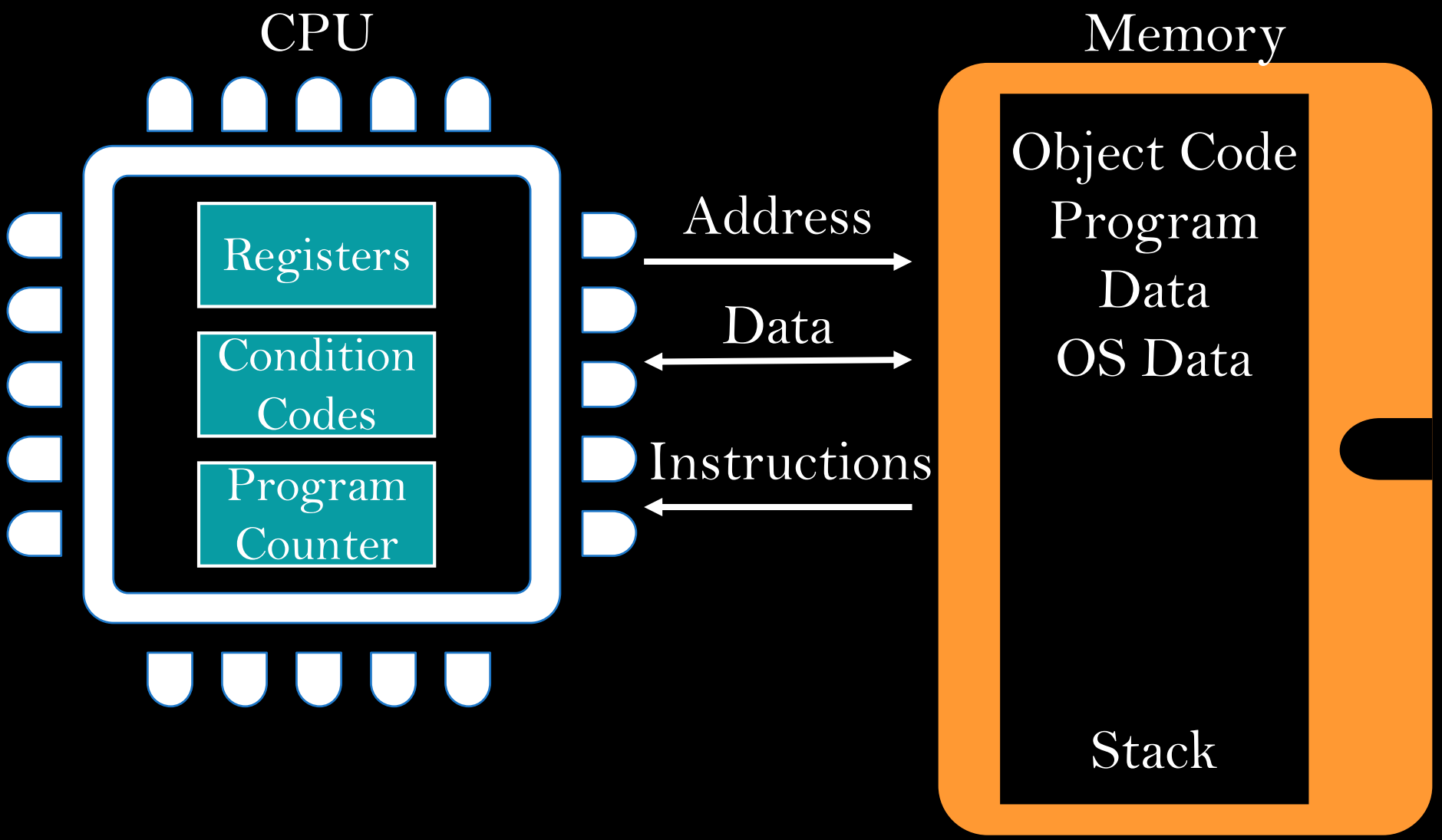
Bits and Bytes

1. Bits and Bytes
2. Addressing
3. Boolean Algebra

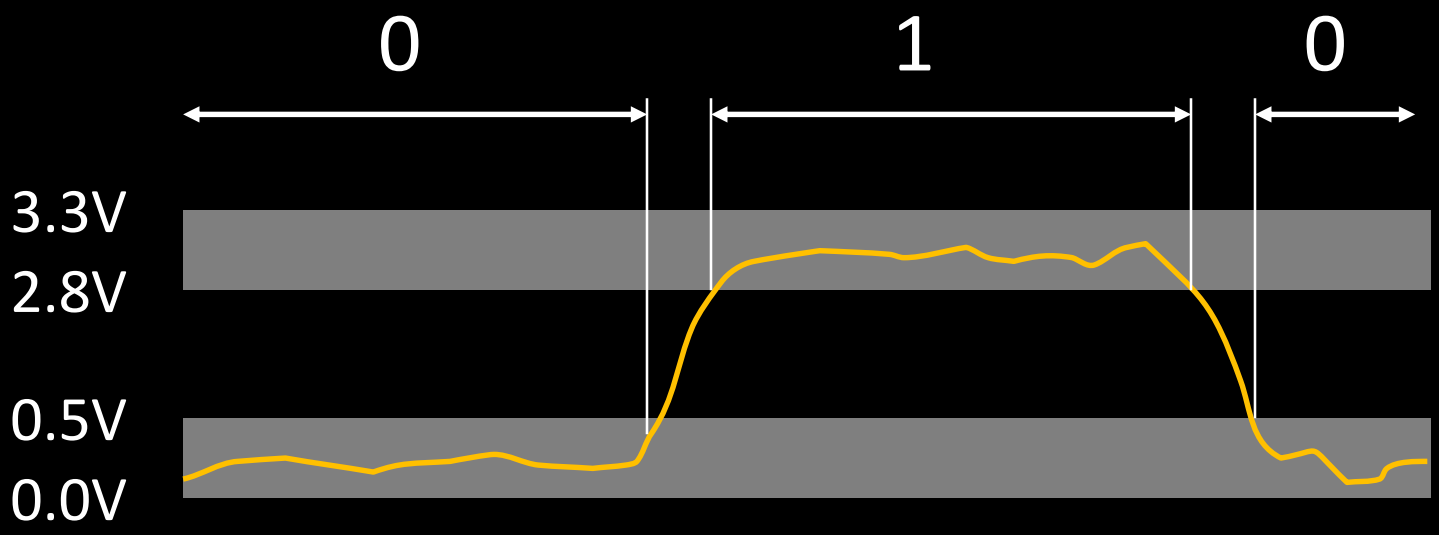
Hardware: Logical View



Programmer's View

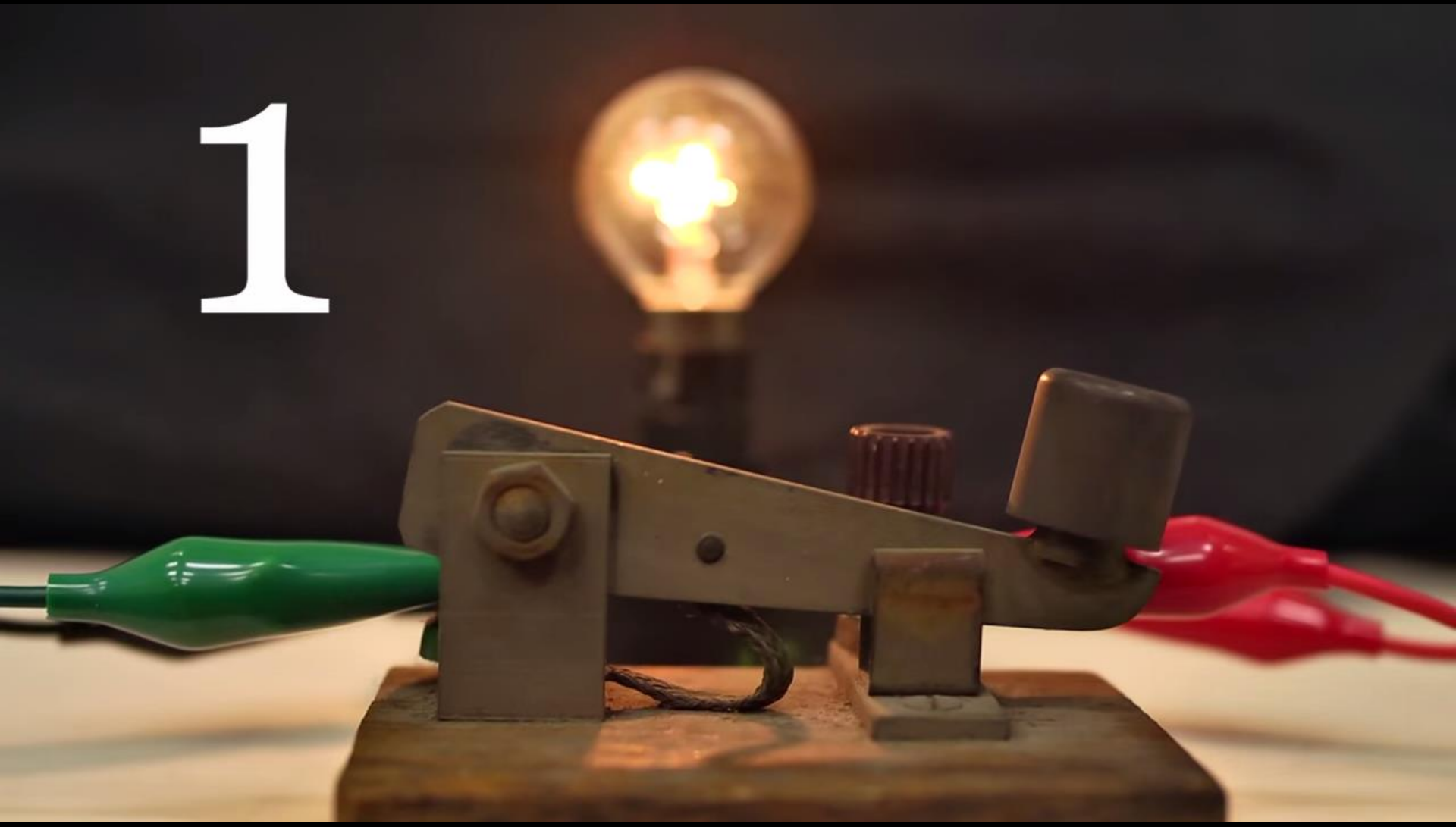


Binary Representation



Binary digit

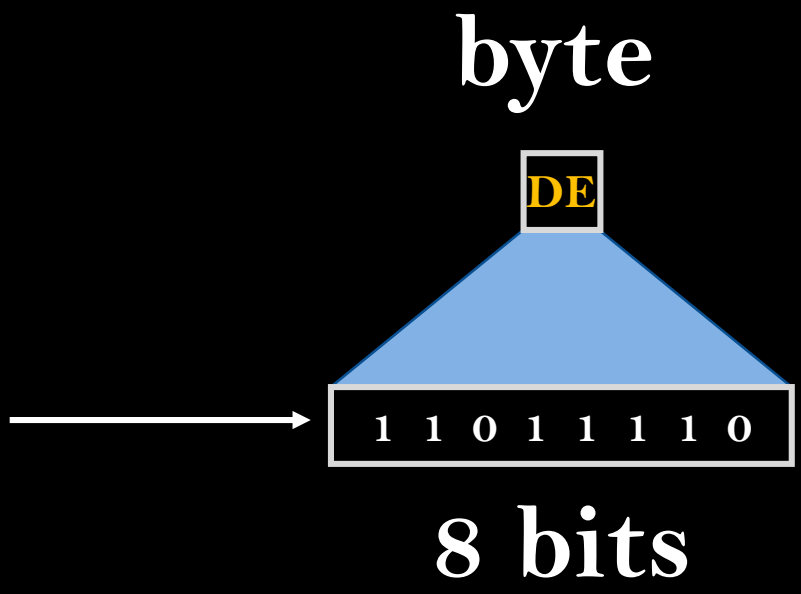

O


1

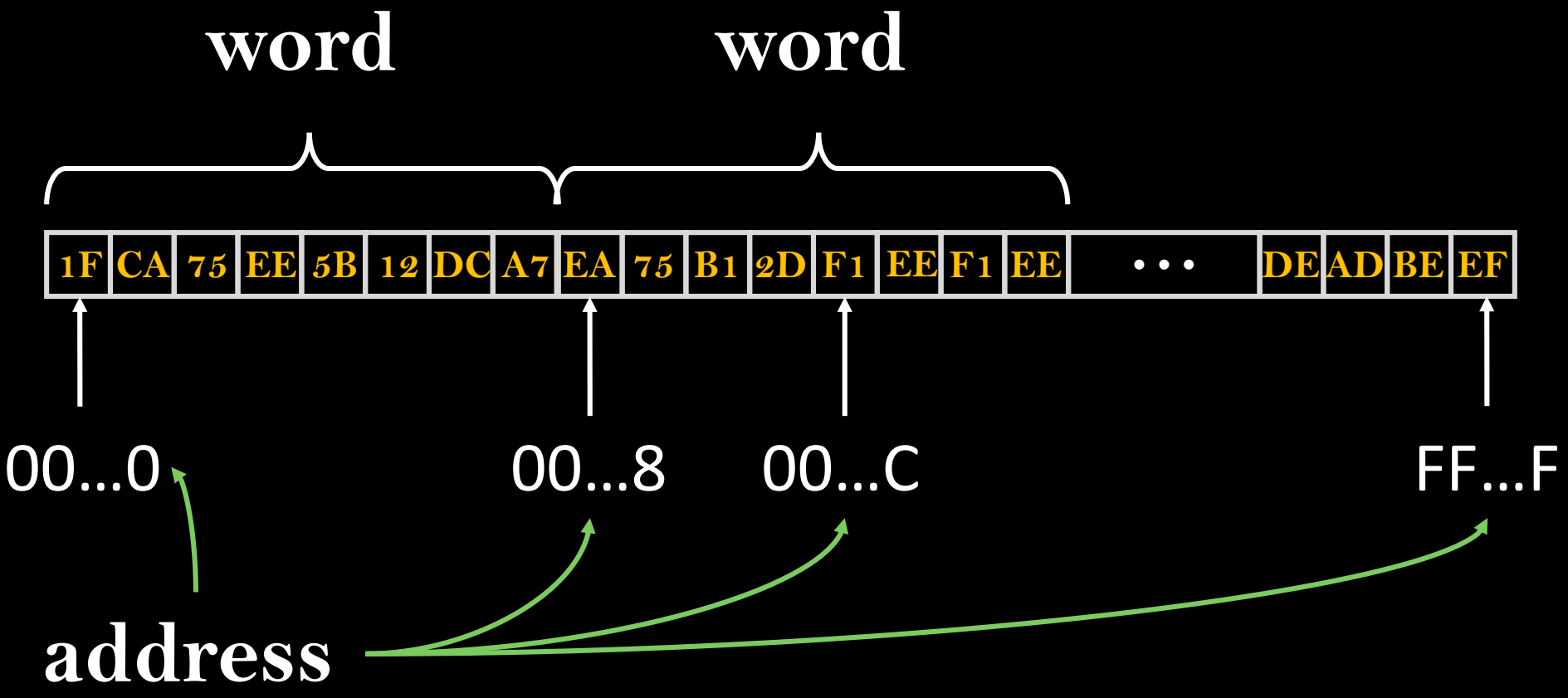
Binary				Dec	Hex
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7

Binary				Dec	Hex
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

No. of bit	Max Value	C C++
1	1	
2	3	
4	15	
8	255	char
16	65,535	short
32	~4.3B	int
64	~18BB	long



Memory Organization



location in memory

Memory Organization

Byte Ordering

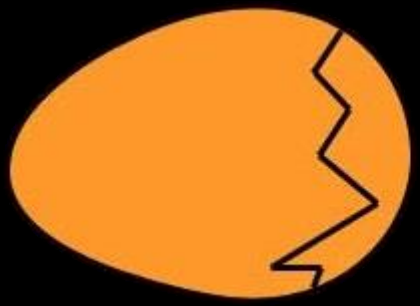
Big-Endian

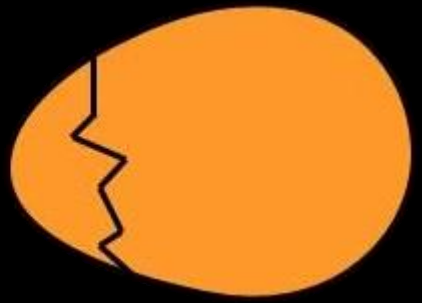
1FCA75EE5B12DCA7

A7DC125BEE75CA1F

00…00	1F	CA	75	EE	5B	12	DC	A7
00…08	EA	75	B1	2D	F1	EE	F1	EE
00…10	00	00	00	00	00	00	00	08
	08	00	00	00	00	00	00	00
	…							
	49	6C	6F	76	65	55	00	00
	18	00	00	00	00	00	00	00
	DE	AD	BE	EF	DE	AD	BE	EF

Little-Endian





Examining Data Representations

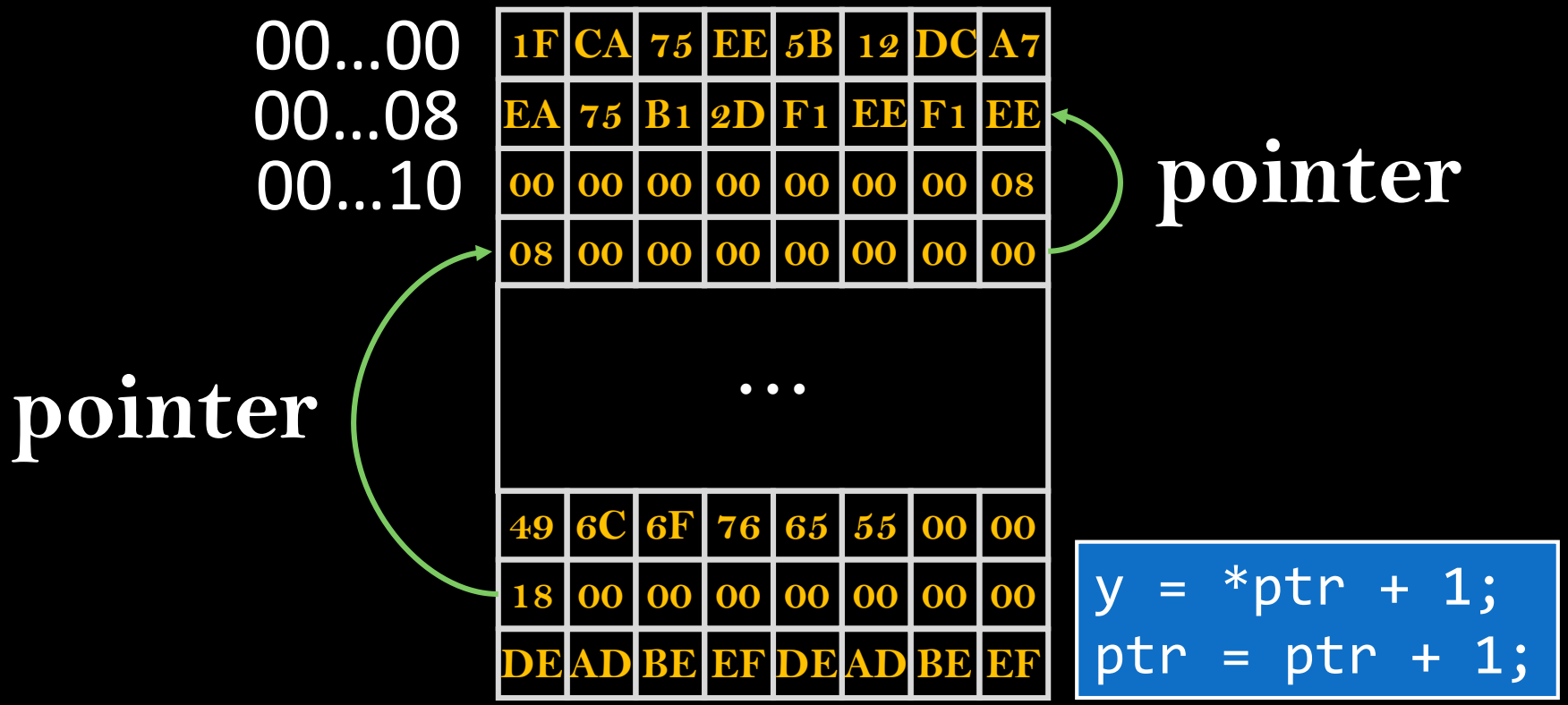
```
void show_bytes(char *start, int len) {
  int i;
  for (i = 0; i < len; i++)
    printf("%p\t0x%.2x\n",start+i,*(start+i));
  printf("\n");
}
```

```
void show_int (int x) {
  show_bytes( (char *) &x, sizeof(int));
}
```

```
int a = 12345; // 0x00003039
printf("int a = 12345;\n");
show_int(a);
```

Pointer

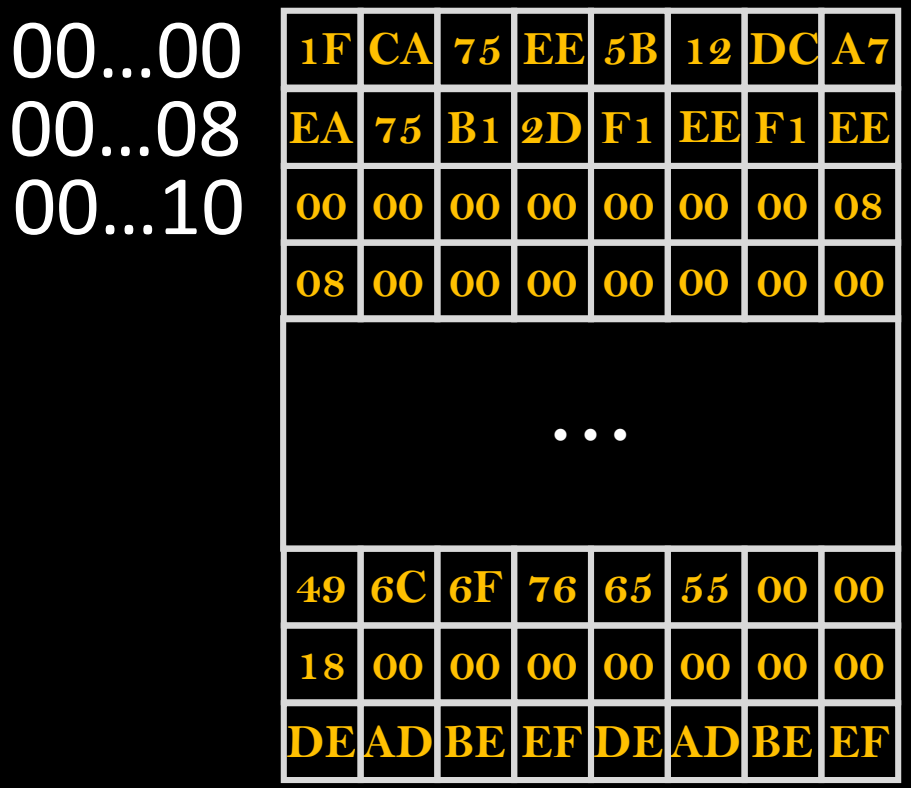
```
long x = 0xEEF1EEF12DB175EA;
long *ptr = &x;
```




```
y = *ptr + 1;
ptr = ptr + 1;
```

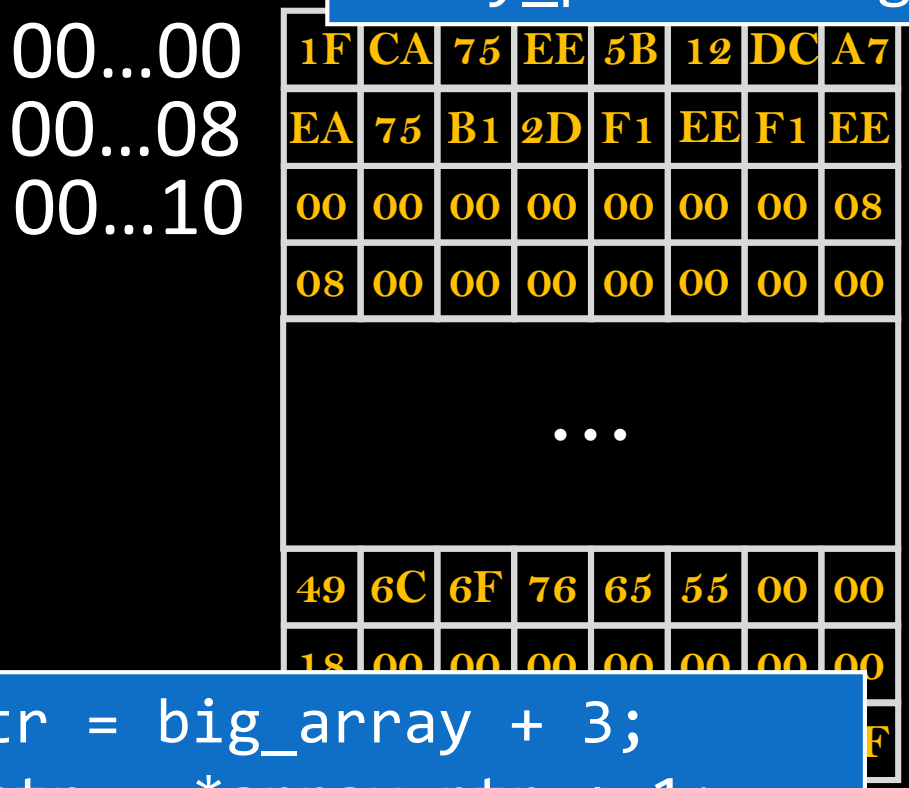
Assignment

```
int *x; int y;
x = &y + 3;
*x = y;
```



Array

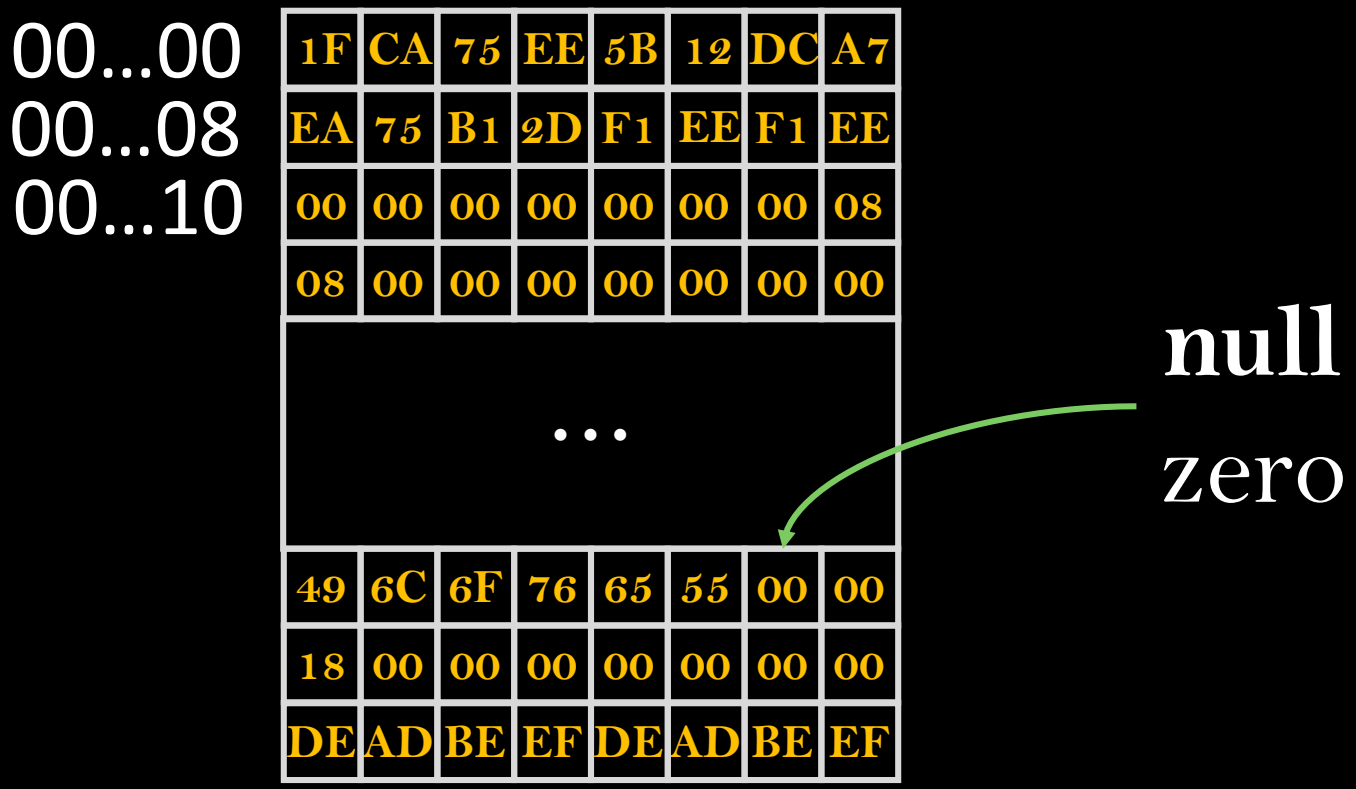
```
int *array_ptr;
array_ptr = big_array;
array_ptr = &big_array[0];
array_ptr = &big_array[3];
array_ptr = &big_array[0] + 3;
```




```
array_ptr = big_array + 3;
*array_ptr = *array_ptr + 1;
array_ptr = &big_array[130];
```

Strings

ASCII codes



Boolean Algebra

AND: A&B OR: A|B XOR: A^B NOT: ~A

&	0	1
0	0	0
1	0	1

|	0	1
0	0	1
1	1	1

^	0	1
0	0	1
1	1	0

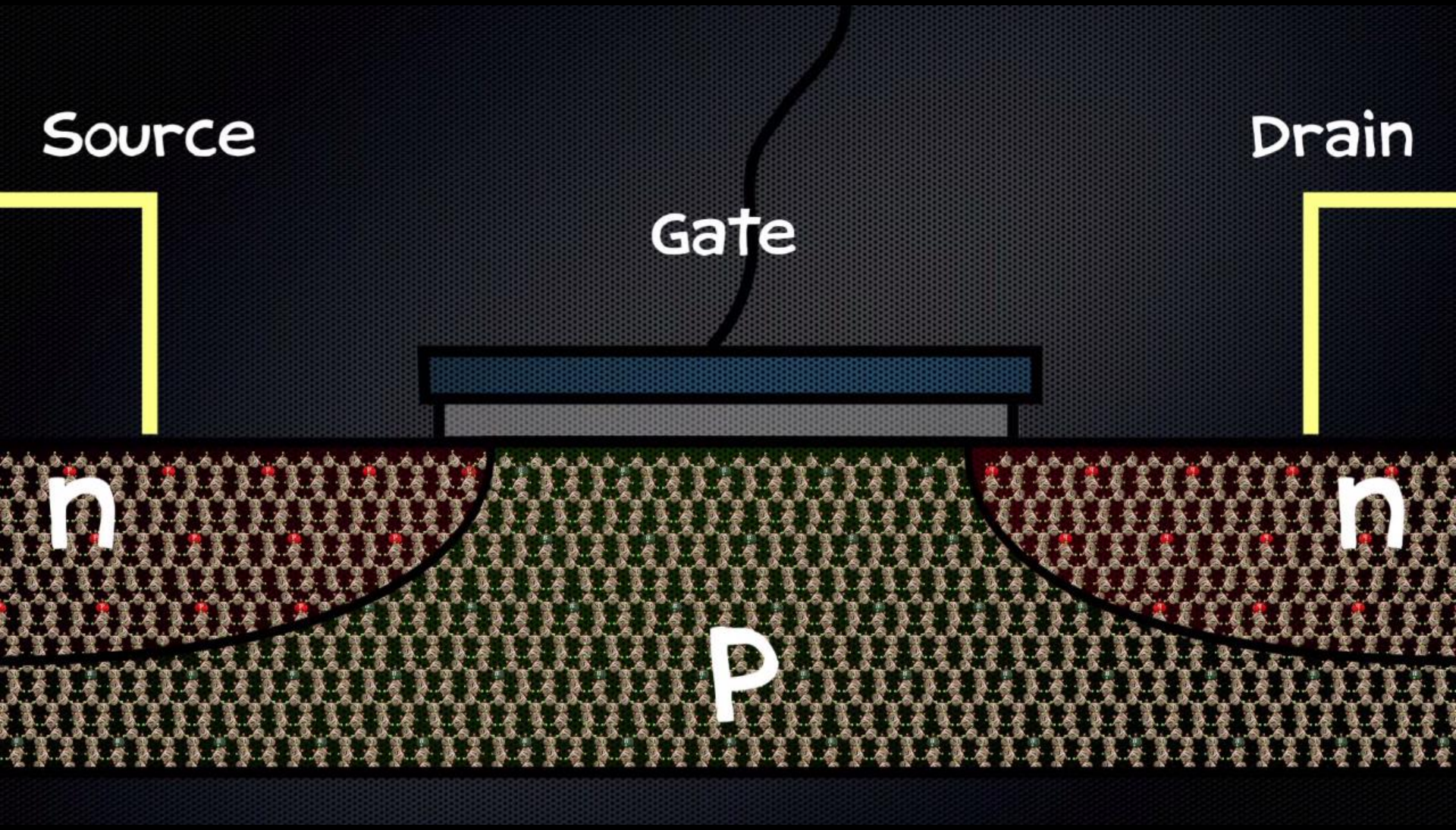
~	
0	1
1	0

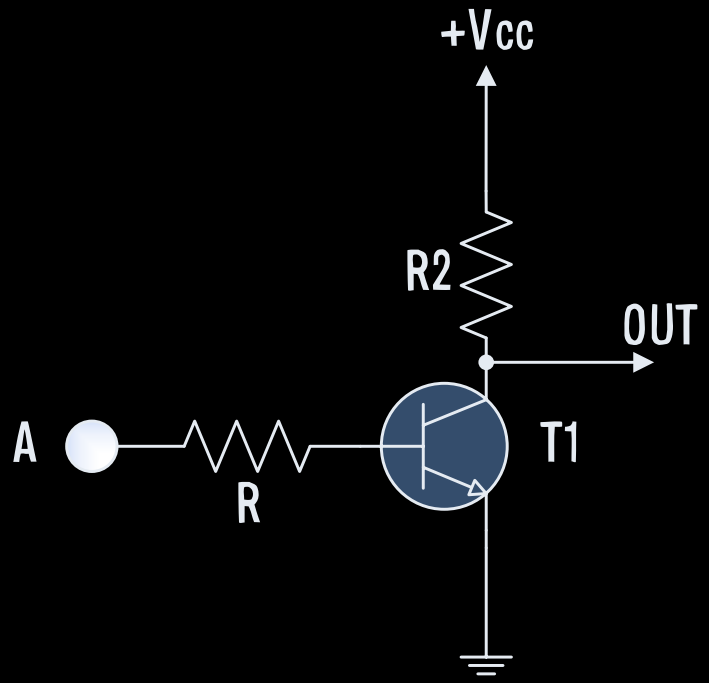
DeMorgan's Law: ~(A | B) = ~A & ~B

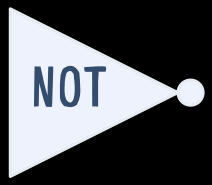
Boolean Algebra

AND: A&B OR: A|B XOR: A^B NOT: ~A

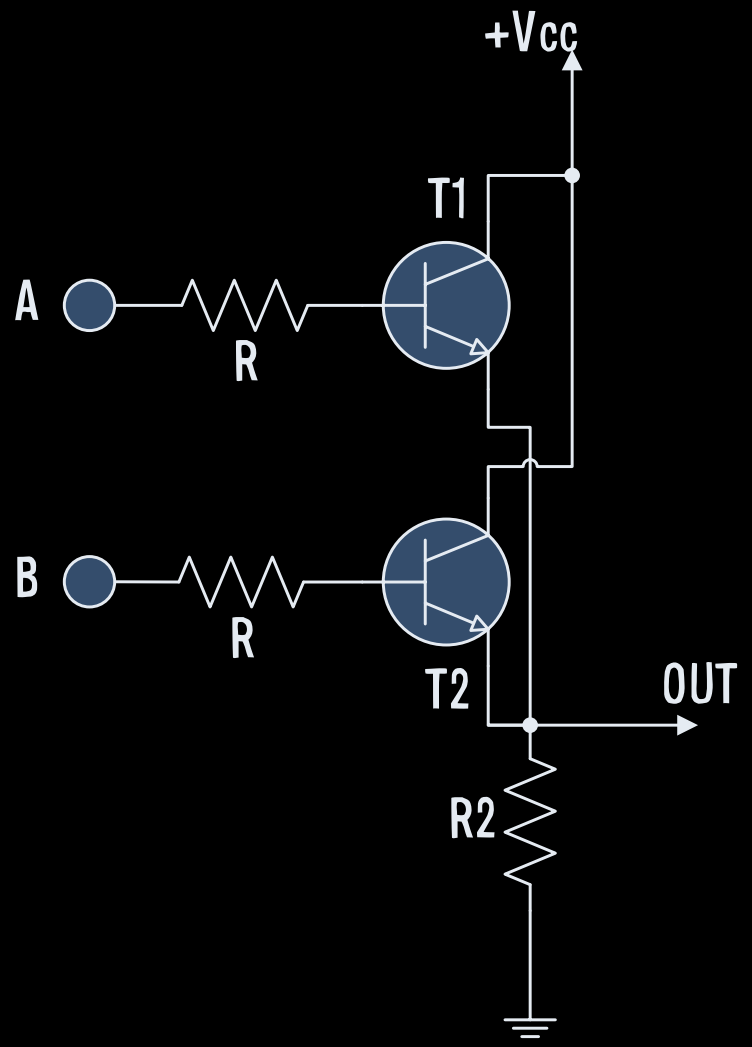
```
  01101001     01101001     01101001
& 01010101   | 01010101   ^ 01010101   ~ 01010101
----------   ----------   ----------   ----------
  01000001     01111101     00111100     10101010
```

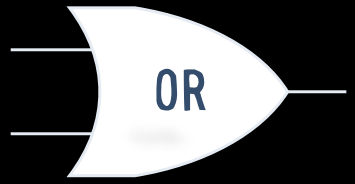

Source
Drain
Gate
n
n
P



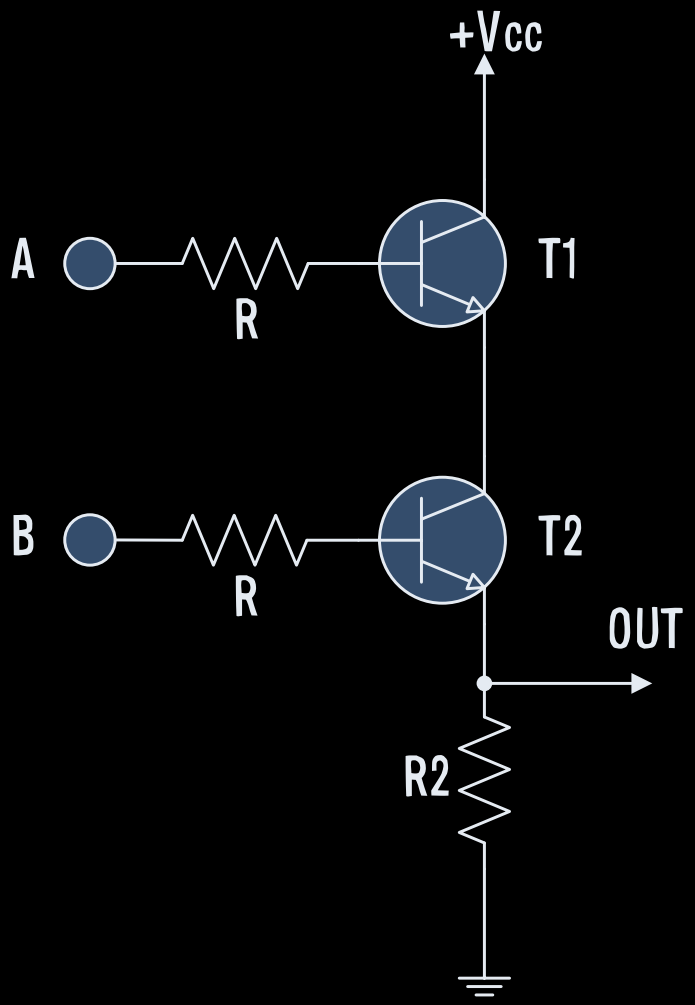


A	OUT
0	1
1	0

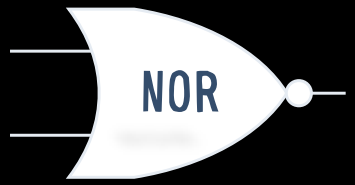




A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	1



A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1



A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	0

A	B	OUT
0	0	1
0	1	1
1	0	1
1	1	0

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0

A	B	OUT
0	0	1
0	1	0
1	0	0
1	1	1

Shift Operations

x << y

Shift Operations

x >> y

Shift Operations

Arithmetic x >> y

To do

1. Read section 2.1
2. Watch the lectures in advance “Bits, Bytes, and Ints: Part 1”

   http://www.cs.cmu.edu/afs/cs/academic/class/15213-f15/www/schedule.html

Augustus De Morgan

“I was x years of age in the year x^2.”