1+1 = 2
… in hardware!

CS 3410: Computer System Organization and Programming

[K. Bala, A. Bracy, E. Sirer, Z. Susag, and H. Weatherspoon]

What is 2^{10}?

0

1024

0%

$2^5 \times 2^2$

0%

$2^5 \times 2^5$

0%

The number of distinct binary numbers you can express in 10 digits.

0%

The number of distinct decimal numbers you can express in 10 digits.

0%

Goals for Today

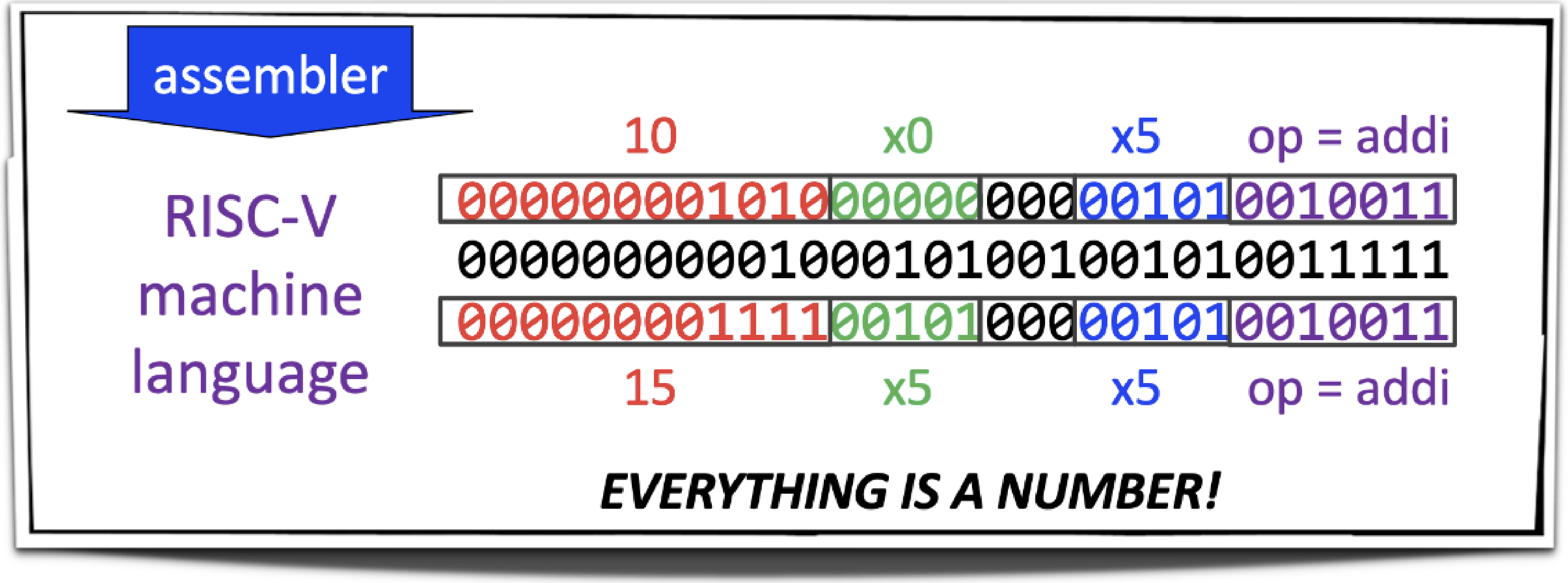
- Number representations
  - How to translate between different **bases**
- Addition
  - How to construct simple **binary** adders

Number Representations

Most computers represent exactly 2 symbols:

- "high voltage" = 1 = true; ”low voltage" = 0 = false

Recall:



Number Representations

Most computers represent exactly 2 symbols:

- "high voltage" = 1 = true; ”low voltage" = 0 = false

How do we represent numbers in Binary? (base 2)

How do we do it for decimal? (base 10)

Base 10

hundreds *tens* *ones*

$637_{10} = 6 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0 = 637$

Base 2

fours *twos* *ones*

$101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$

Counting in Different Bases

DEC (Base 10)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
BIN (Base 2)	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	1000 0	1000 1	1001 0
OCT (Base 8)	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22
HEX (Base 16)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12

Counting in Different Bases

DEC (Base 10)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
BIN (Base 2)	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010
OCT (Base 8)	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22
HEX (Base 16)	0	1	2	3	4	5	6	7	8	9	A	B	C						12

0b 1111 1111 = 255

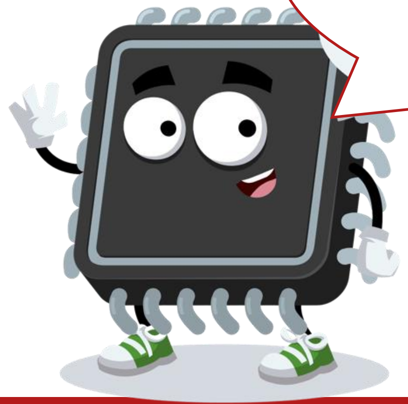
0b 1 0000 0000 = 256

0o 77 = 63

0o 100 = 64

0x ff = 255

0x 100 = 256



PRO TIP!

4 binary digits = 1 hex digit

3 binary digits = 1 octal digit

Converting between bases: $637_{10} \rightarrow$ octal

Approach #1: Left to Right

637 – 1 x 512 = 125
125 – 1 x 64 = 61
61 – 7 x 8 = 5

8^3	8^2	8^1	8^0
512	64	8	1
1	1	7	5

Approach #2: Right to Left (repetitive division)

637 / 8 = 79 remainder 5
79 / 8 = 9 remainder 7
9 / 8 = 1 remainder 1
1 / 8 = 0 remainder 1

lsb (least significant bit*)

↓

msb (most significant bit*)

637 = 0o1175

Try $637_{10} \rightarrow$ binary! (using both methods)

Poll Everywhere Question #2:

Convert the number 657_{10} to base 16
What is the least significant digit of this number?

a) D
b) F
c) 0
d) 1
e) 11

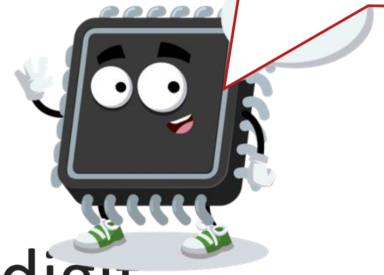
Convert the number 657_{10} to base 16. What is the least significant digit?

0

Nobody has responded yet.

Hang tight! Responses are coming in.

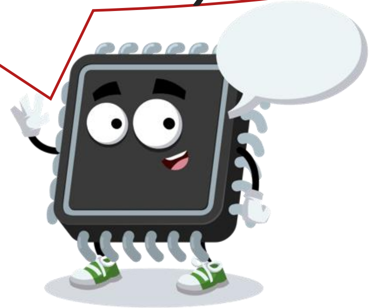
Convert from Binary to other powers of 2



- Binary to Octal
  - 3 bits (000—111) have values 0…7 = 1 octal digit

    example: 0b 1 001 111 101

    1 1 7 5 → 0o1175



- Binary to Hexadecimal
  - **Nibble** (0000—1111) has values 0…15 = 1 hex digit

    example: 0b 10 0111 1101

    2 7 d → 0x27d

Achievement Unlocked!

There are 10 types of people in the world:

- Those who understand binary
- And those who do not
- *And* those who know this joke is in base 3

Goals for Today

- Number representations
  - How to translate between different **bases**

  (Next Monday we'll see how computers use binary representation)
- Addition
  - How to construct simple **binary** adders

Binary Addition

Addition works the same for all bases

- Add the digits in each position
- Propagate the carry

Binary addition is pretty easy

- Combine two bits at a time
- Along with a carry

```
    1
  183
+ 254
-----
  437
```

Carry-in (of 4th bit)

Carry-out (of 3rd bit)

```
  111
  001110
+ 011100
--------
  101010
```

Goals for Today

- Number representations
  - How to translate between different **bases**

  (Next Monday we'll see how computers use binary representation)
- Addition
  - How to construct simple **binary** adders