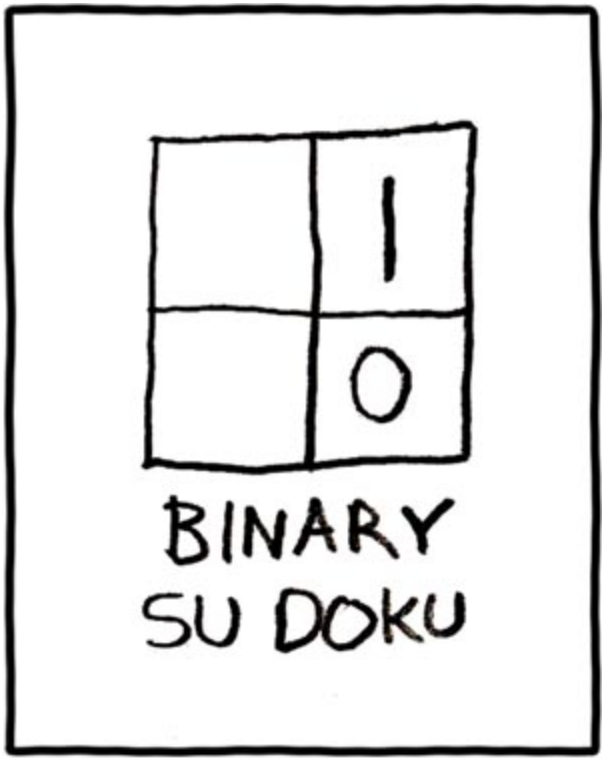
Embedded systems intro, Number systems



http://xkcd.com/74/

Overview

- What exactly is an *embedded* system?
- Why study embedded systems?
- Number systems
  - Base 2, 8, 10, 16
  - Converting between them all

General-purpose computer

- Device with a microprocessor
- Jack of all trades
- People willing to pay big $$$
- Rich input/output
- Crashing is irritating
- Complex operating system
- Lots of memory
- Powerful processor
- Program in high level languages ignoring hardware details

Embedded system

- Device with a microprocessor
- Specialize in 1 or a few things
- Must be very cheap
- Little to no input/output
- Crashing could kill somebody
- Often no operating system
- Little memory
- Simple processor
- Program "to the metal" often in low level languages (e.g. assembly)

General-purpose computer

1%

Embedded system

99%

Blurred lines?

LG VX5600

iPhone 4

Nokia 810

Android microwave

Android washing machine

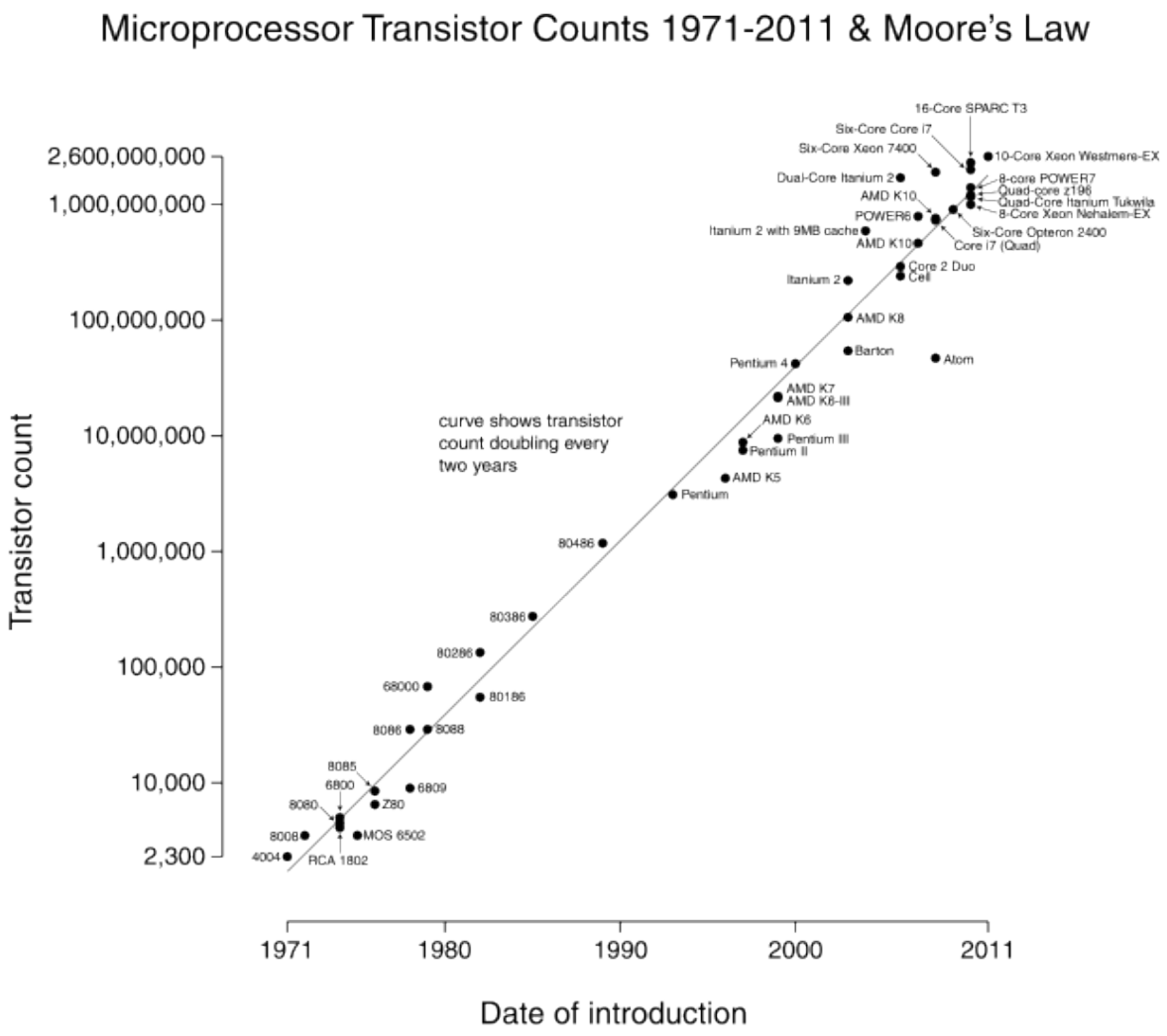
Why study embedded?

- Opens up the other 99% of microprocessors
- Learn how computers work from ground up
- Prepare you for more advanced computer architecture courses
  - Binary, octal, hexadecimal number systems
  - How computers store numbers
  - Bitwise operations
  - Boolean algebra
  - Combinational and sequential circuits
  - How computers fetch and execute instructions

Why study embedded?

- Gain assembly language experience
  - Makes you appreciate high-level languages
  - Understand underlying hardware realities
  - Programming simple processors with no memory/ cycles to waste
  - Extreme optimization of certain portions of a program (e.g. frequent calculations)
    - Modern optimizing compilers make gains questionable
  - Interacting directly with hardware (e.g. device drivers)
  - Real-time programs needing precise timing

Where have all the simple processors gone?



Real-time

- Soft real-time - respond to a click in 100 ms
- Hard real-time - respond to rudder correction in 100 ms
- Timing unpredictable in high-level languages & operating systems
  - Java garbage collection
  - Thread scheduling by OS
  - Windows update
- Real-time versions of languages & operating systems
  - Real-time Java, real-time Linux, Windows CE

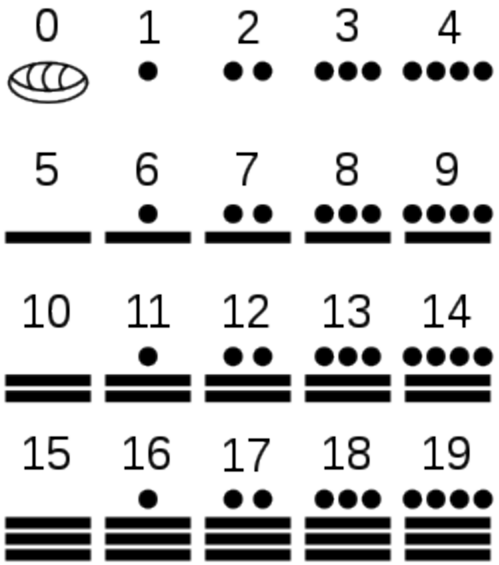
Embedded development challenges

- Reliability
  - Cannot crash, devices often run 24 x 7 x 365
  - Thorough testing, must handle errors and unexpected events
- Performance
  - Real time systems, rapid response
    - Antilock brakes → detect slippage and react within ms
  - Must not be bottleneck, good throughput
  - Multitasking, handle interrupts when data arrives

Embedded development challenges

- Resource constraints
  - Cost of device
  - Amount of memory
  - Power consumption (huge for mobile devices)
- Debugging
  - Is this a software or hardware problem?
  - Easier than hardware circuits, harder than desktop
- User interface
  - One-off device with perhaps limit I/O capabilities
  - Users won't read manual

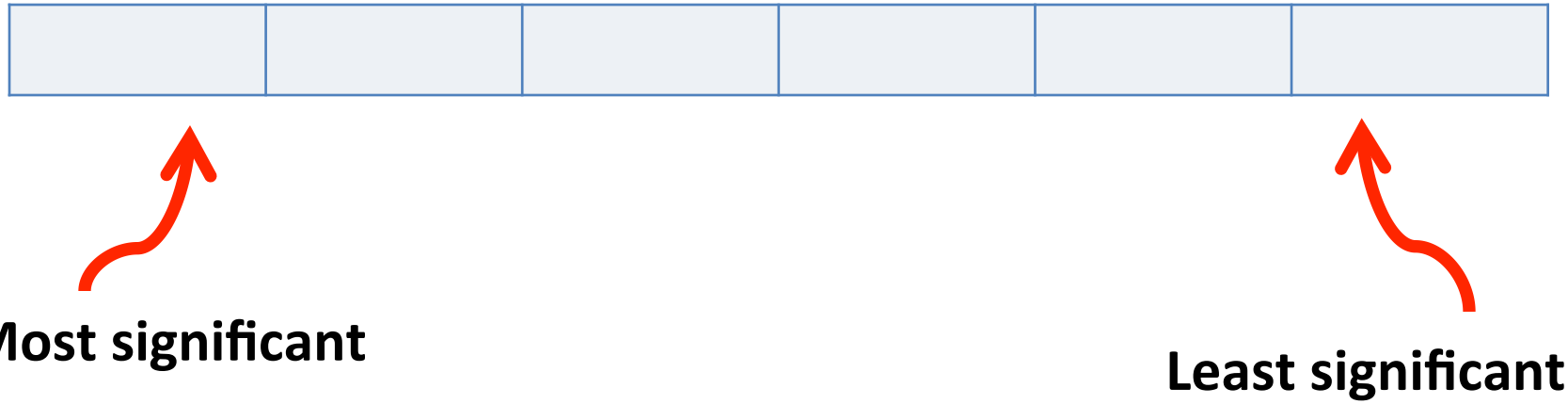
Number systems



Our friend: base 10

10

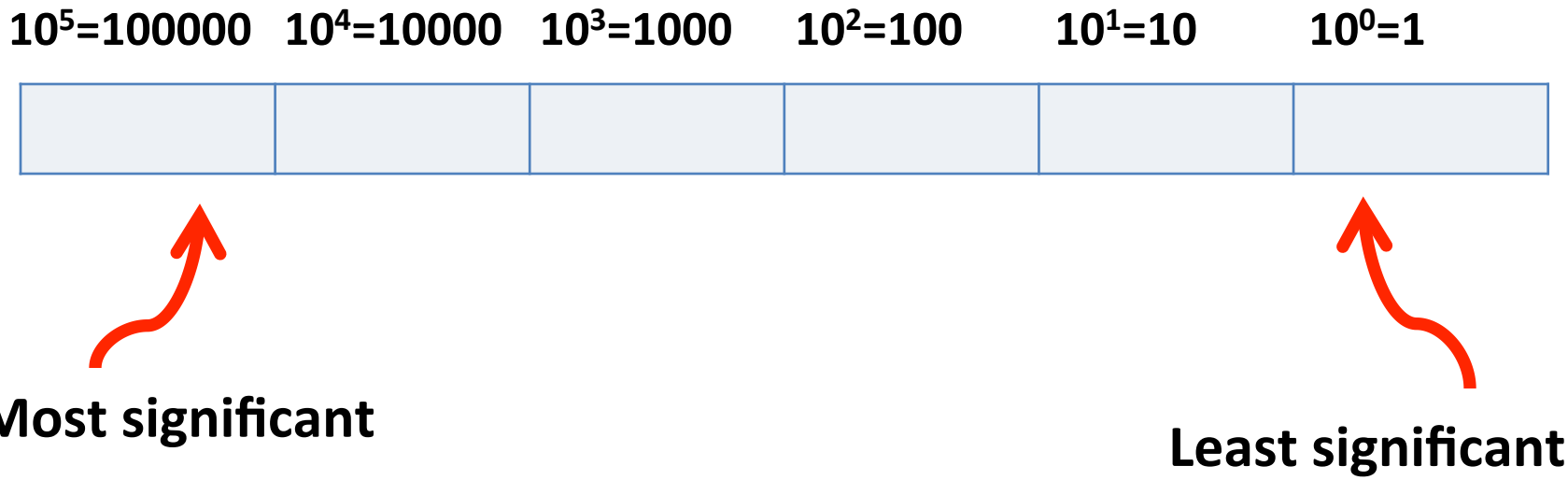
- “one thousand two hundred fifty two”
- Let’s convert this to 6-digit base 10:



Our friend: base 10

10

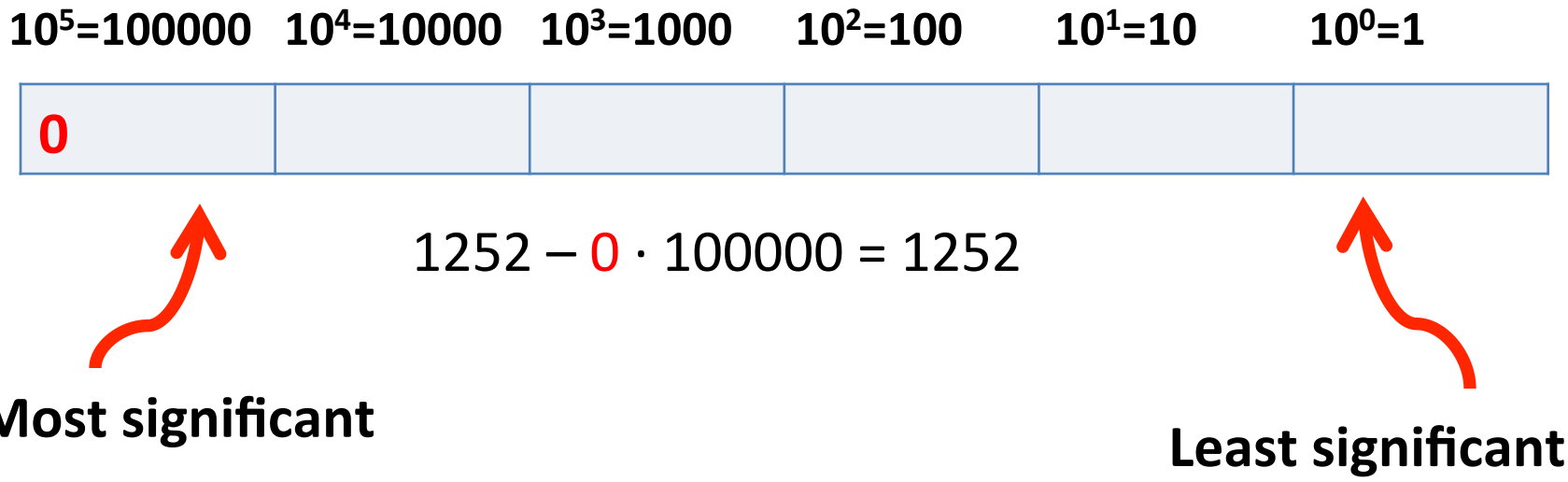
- “one thousand two hundred fifty two”
- Let’s convert this to 6-digit base 10:



Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:



Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:

$10^5=100000$	$10^4=10000$	$10^3=1000$	$10^2=100$	$10^1=10$	$10^0=1$
0	**0**				

$1252 - 0 \cdot 100000 = 1252$

$1252 - 0 \cdot 10000 = 1252$

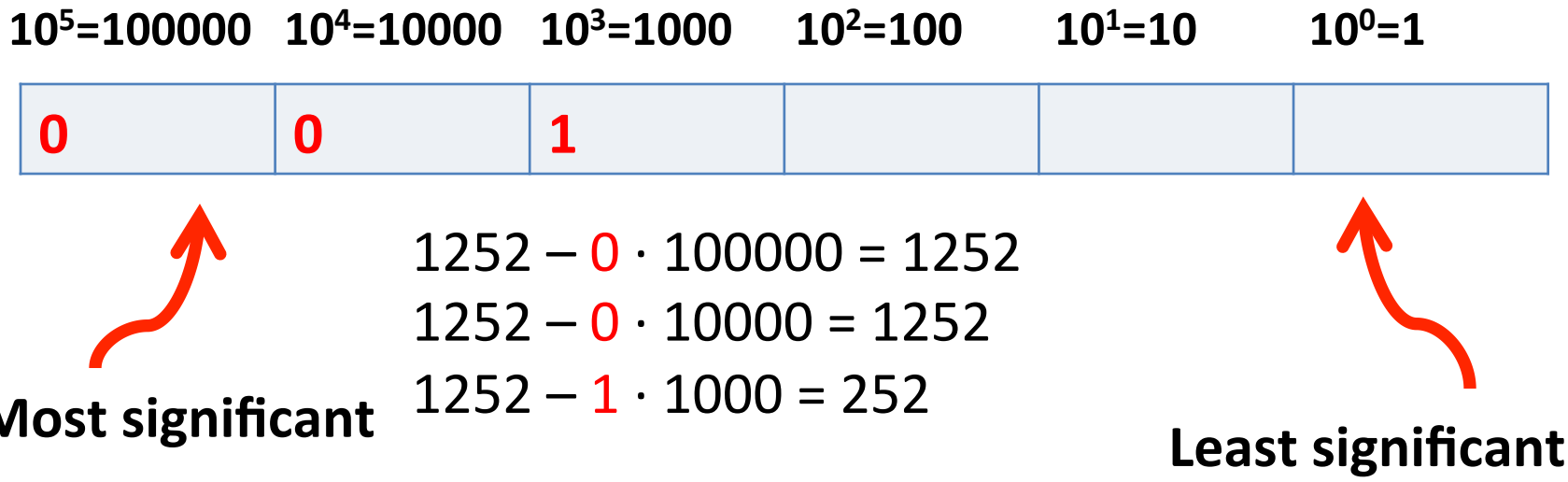
Most significant

Least significant

Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:



Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:

10^5=100000	10^4=10000	10^3=1000	10^2=100	10^1=10	10^0=1
0	**0**	**1**	**2**		

Most significant

$1252 - 0 \cdot 100000 = 1252$

$1252 - 0 \cdot 10000 = 1252$

$1252 - 1 \cdot 1000 = 252$

$252 - 2 \cdot 100 = 52$

Least significant

Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:

$10^5=100000$	$10^4=10000$	$10^3=1000$	$10^2=100$	$10^1=10$	$10^0=1$
0	0	1	2	5	

Most significant

$1252 - 0 \cdot 100000 = 1252$
$1252 - 0 \cdot 10000 = 1252$
$1252 - 1 \cdot 1000 = 252$
$252 - 2 \cdot 100 = 52$
$52 - 5 \cdot 10 = 2$

Least significant

Our friend: base 10

10

- "one thousand two hundred fifty two"
- Let's convert this to 6-digit base 10:

$10^5=100000$	$10^4=10000$	$10^3=1000$	$10^2=100$	$10^1=10$	$10^0=1$
0	**0**	**1**	**2**	**5**	**2**

Most significant

$1252 - 0 \cdot 100000 = 1252$
$1252 - 0 \cdot 10000 = 1252$
$1252 - 1 \cdot 1000 = 252$
$252 - 2 \cdot 100 = 52$
$52 - 5 \cdot 10 = 2$
$2 - 2 \cdot 1 = 0$

Least significant

Our geeky friend: base 2

- Binary = base 2
- 0s and 1s
- low voltage (0-1V) vs. high voltage (3-5V)

01100111 **0101**

Our geeky friend: base 2

- Binary = base 2
- 0s and 1s
- low voltage (0-1V) vs. high voltage (3-5V)

01100111

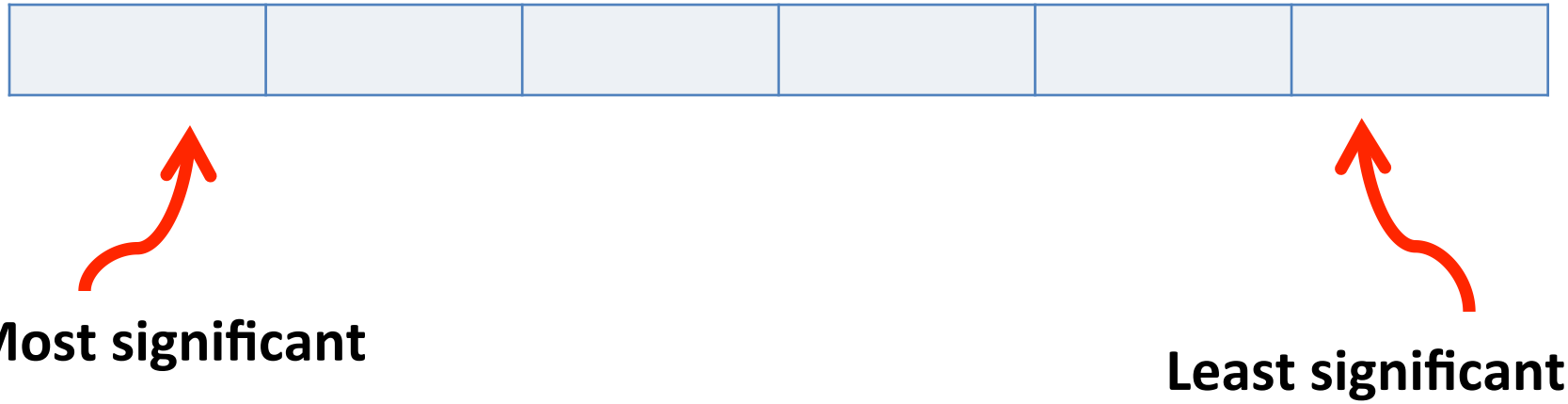
8 bits = byte

0101

4 bits = nibble

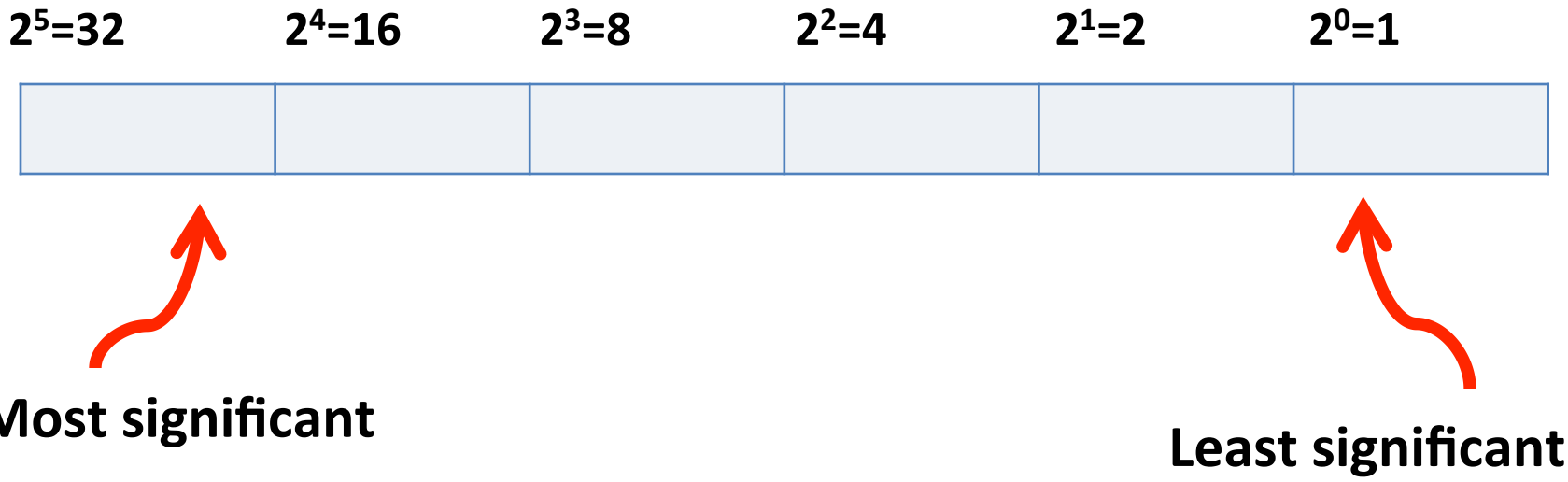
Our geeky friend: base 2

- “forty two”
- Let’s convert this to 6-digit base 2:



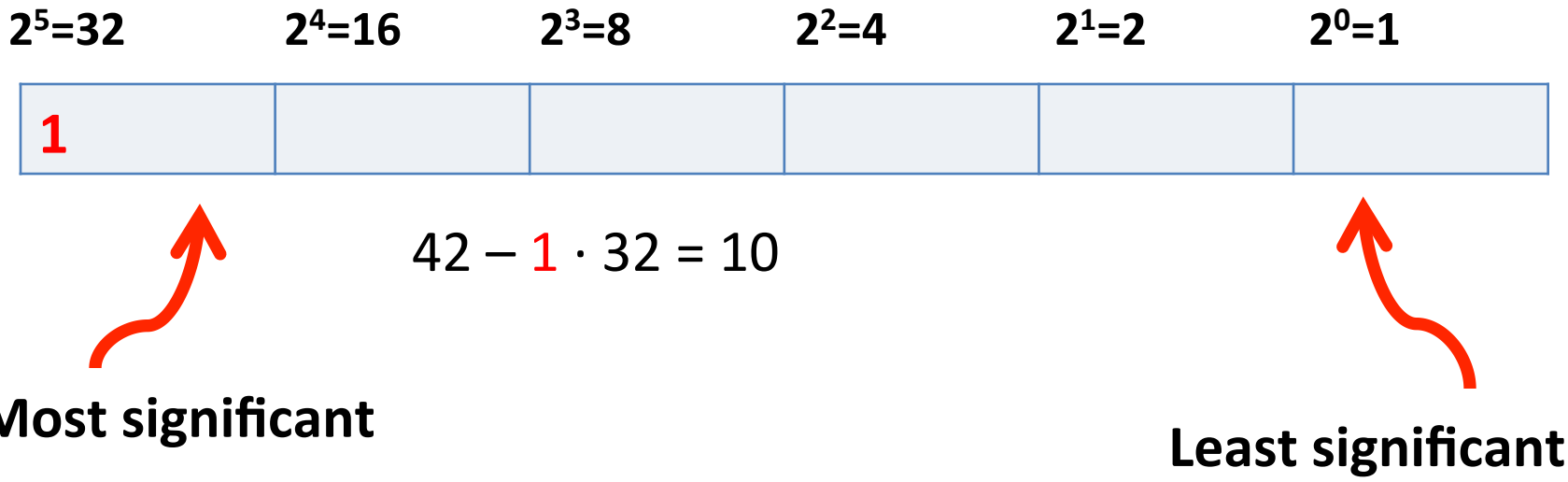
Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:



Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:



Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0				

$42 - 1 \cdot 32 = 10$

$10 - 0 \cdot 16 = 10$

Most significant

Least significant

Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0	1			

$42 - 1 \cdot 32 = 10$

$10 - 0 \cdot 16 = 10$

$10 - 1 \cdot 8 = 2$

Most significant

Least significant

Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0	1	0		

Most significant

Least significant

$42 - 1 \cdot 32 = 10$

$10 - 0 \cdot 16 = 10$

$10 - 1 \cdot 8 = 2$

$2 - 0 \cdot 4 = 2$

Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0	1	0	1	

Most significant

Least significant

$42 - 1 \cdot 32 = 10$

$10 - 0 \cdot 16 = 10$

$10 - 1 \cdot 8 = 2$

$2 - 0 \cdot 4 = 2$

$2 - 1 \cdot 2 = 0$

Our geeky friend: base 2

- "forty two"
- Let's convert this to 6-digit base 2:

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
1	0	1	0	1	0

Most significant

$42 - 1 \cdot 32 = 10$
$10 - 0 \cdot 16 = 10$
$10 - 1 \cdot 8 = 2$
$2 - 0 \cdot 4 = 2$
$2 - 1 \cdot 2 = 0$
$0 - 0 \cdot 1 = 0$

Least significant

Our geeky friend: base 2

- Given a binary number: 011100
- What is this number in base 10?

0	1	1	1	0	0

Most significant

Least significant

Our geeky friend: base 2

- Given a binary number: 011100
- What is this number in base 10?

$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
0	1	1	1	0	0

16 + 8 + 4 = 28

Most significant

Least significant

1110 0101 1010 0111

Our super geeky friend: base 16

1110 0101 1010 0111

E 5 A 7

- Hexadecimal (or hex)
- 16 digits: 0 1 2 3 4 5 6 7 8 9 **A B C D E F**
- Often written: 0xE5A7

Our super geeky friend: base 16

16

Binary : 1110 0101 1010 0111

Decimal : 58791

Hex : ?

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$

Our super geeky friend: base 16

16

Binary : 1110 0101 1010 0111

Decimal : 58791

Hex : E

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
E			

$$58791 - 14 \cdot 4096 = 1447$$

Our super geeky friend: base 16

16

Binary : 1110 0101 1010 0111

Decimal : 58791

Hex : E5

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
E	5		

$58791 - 14 \cdot 4096 = 1447$

$1447 - 5 \cdot 256 = 167$

Our super geeky friend: base 16

16

Binary : 1110 0101 1010 0111

Decimal : 58791

Hex : E5A

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
E	**5**	**A**	

$58791 - 14 \cdot 4096 = 1447$

$1447 - 5 \cdot 256 = 167$

$167 - 10 \cdot 16 = 7$

Our super geeky friend: base 16

Binary : 1110 0101 1010 0111

Decimal : 58791

Hex : E5A7

$16^3=4096$	$16^2=256$	$16^1=16$	$16^0=1$
E	**5**	**A**	**7**

$58791 - 14 \cdot 4096 = 1447$

$1447 - 5 \cdot 256 = 167$

$167 - 10 \cdot 16 = 7$

$7 - 7 \cdot 1 = 0$

Binary	Hex	Decimal
0000	0x0	0
0001	0x1	1
0010	0x2	2
0011	0x3	3
0100	0x4	4
0101	0x5	5
0110	0x6	6
0111	0x7	7
1000	0x8	8
1001	0x9	9
1010	0xA	10
1011	0xB	11
1100	0xC	12
1101	0xD	13
1110	0xE	14
1111	0xF	15

0100111101100101

"nibblize" ↓

0100 1111 0110 0101

Lookup in table ↓

4 F 6 5

(also works the other way: hex to binary)

Our fringe friend: base 8

8

Binary : 110100111

Decimal : 423

Octal : ???

- Octal has 8 digits:

  0 1 2 3 4 5 6 7

Binary	Octal	Hex	Decimal
000	0	0x0	0
001	1	0x1	1
010	2	0x2	2
011	3	0x3	3
100	4	0x4	4
101	5	0x5	5
110	6	0x6	6
111	7	0x7	7

Our fringe friend: base 8

8

Binary : 110 100 111

Decimal : 423

Octal : 647

- Octal has 8 digits:

  0 1 2 3 4 5 6 7

Binary	Octal	Hex	Decimal
000	0	0x0	0
001	1	0x1	1
010	2	0x2	2
011	3	0x3	3
100	4	0x4	4
101	5	0x5	5
110	6	0x6	6
111	7	0x7	7

Our fringe friend: base 8

8

- Ancient 18-bit computers
- Some natural languages count using spaces between fingers
- Used in **`chmod`** unix command

	User			Group			All		
	-	w	x	r	-	x	-	-	x
	$0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0$			$1\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$			$0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$		
0	3			5			1		

chmod 0351 ↔ chmod u=wx,g=rx,o=x

Different bases in C

```
void main(void)
{
    int a = 10;
    int b = 010;
    int c = 0x10;

    printf("a = %d\n", a);
    printf("b = %d\n", b);
    printf("c = %d\n", c);
}
```

Different bases in C

```
void main(void)
{
    int a = 10;
    int b = 010;
    int c = 0x10;

    printf("a = %d\n", a);
    printf("b = %d\n", b);
    printf("c = %d\n", c);
}
```

```
a = 10
b = 8
c = 16
```

Summary

- Embedded systems
  - Specialized computing devices
  - Limited: I/O, memory, power, cost
  - May be mission critical, real-time systems
- Number systems
  - Base 2 (binary), 8 (octal), 10 (decimal), 16 (hex)
  - Conversion using simple math
    - Add, subtract, division with remainder, powers
  - Conversion using a lookup table