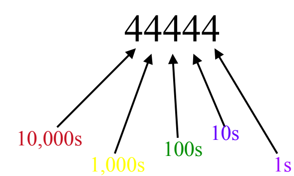
Data Representation

Numbers Are Represented in Bases



- Decimal System uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Base 10
- Place-value number system: position of a digit interpreted to give the value

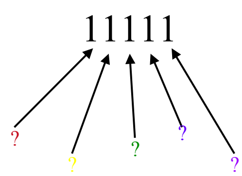
Decimal System

- 44,444 = 4 x 10,000 + 4 x 1,000 + 4 x 100 + 4 x 10 + 4 x 1
  $= 4 \times 10^4 + 4 \times 10^3 + 4 \times 10^2 + 4 \times 10^1 + 4 \times 10^0$
- 1 decimal digit produces ? distinct values
- 2 decimal digits produce ? distinct values
- 3 decimal digits produce ? distinct values
- *n* decimal digits produce ? distinct values

Decimal System

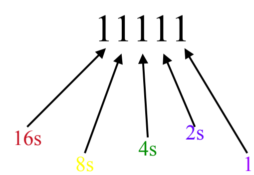
- 44,444 = 4 x 10,000 + 4 x 1,000 + 4 x 100 + 4 x 10 + 4 x 1
  $= 4 \times 10^4 + 4 \times 10^3 + 4 \times 10^2 + 4 \times 10^1 + 4 \times 10^0$
- 1 decimal digit produces 10 distinct values
- 2 decimal digits produce 100 distinct values
- 3 decimal digits produce 1000 distinct values
- *n* decimal digits produce 10^n distinct values

Binary System



- Decimal System uses 2 digits: 0, 1
- Base 2

Binary System



- Decimal System uses 2 digits: 0, 1
- Base 2

Binary System

- $11{,}111_2 = 1 \text{ x } 16 + 1 \text{ x } 8 + 1 \text{ x } 4 + 1 \text{ x } 2 + 1 \text{ x } 1$
  $= 1 \text{ x } 2^4 + 1 \text{ x } 2^3 + 1 \text{ x } 2^2 + 1 \text{ x } 2^1 + 1 \text{ x } 2^0$
- 1 decimal digit produces ? distinct values
- 2 decimal digits produce ? distinct values
- 3 decimal digits produce ? distinct values
- *n* decimal digits produce ? distinct values

Binary System

- $11{,}111_2 = 1 \text{ x } 16 + 1 \text{ x } 8 + 1 \text{ x } 4 + 1 \text{ x } 2 + 1 \text{ x } 1$
  $= 1 \text{ x } 2^4 + 1 \text{ x } 2^3 + 1 \text{ x } 2^2 + 1 \text{ x } 2^1 + 1 \text{ x } 2^0$
- 1 decimal digit produces 2 distinct values
- 2 decimal digits produce 4 distinct values
- 3 decimal digits produce 8 distinct values
- *n* decimal digits produce 2^n distinct values

Binary Numbers

- When counting in decimal, use the 9 “symbols” (0 – 9) for the first 10 numbers. After 9, use 10 (two symbols, the smallest non-zero symbol in the newly created second spot from the left)
- When counting in binary, use the 2 “symbols” (0, 1) for the first 2 numbers. After 1, use 10. So, 1 in the binary world is like 9 in the decimal world.

Binary Numbers

- Counting in Decimal:
- Counting in Binary

Binary Numbers

- Counting in Decimal:

0, 1, 2, …, 9
10, 11, 12, …, 19,
20, …, 29, …, 99,
100, …, 999,
1000, …, 9999,
10000, …, 99999,
…

- Counting in Binary

0, 1,
10, 11,
100, 101, 110, 111,
1000, 1001, 1010, 1011,
1100, 1101, 1110, 1111,
10000, …., 11111,
…

Decimal vs. Binary

Decimal	0	1	2	3	4	5	6	7
Binary	0	1	10	11	100	101	110	111

3 binary digits produce 8 distinct values: 0, …, 7 (8 including 0). So the largest number $7_{10} = 111_2 = 2^3-1 = 8-1$

Binary to Decimal Conversion

- $1011_2 = ?_{10}$

Binary to Decimal Conversion

- $1011_2 = ?_{10}$
- $1011_2 = 1 \times 2_{10}^3 + 0 \times 2_{10}^2 + 1 \times 2_{10}^1 + 1 \times 2_{10}^0$
  $= 8_{10} + 0_{10} + 2_{10} + 1_{10} = 11_{10}$

Decimal to Binary Conversion

- How to find the decimal digits that make up 53624_{10}?

Decimal to Binary Conversion

- How to find the decimal digits that make up 53624_{10}?

53624 / 10 = 5362 + 4/10

5362 / 10 = 526 + 2/10

526 / 10 = 52 + 6/10

52 / 10 = 5 + 2/10

5 / 10 = 0 + 5/10

Decimal to Binary Conversion

- How to find the binary digits that make up 24_{10}?

Decimal to Binary Conversion

- How to find the decimal digits that make up 24_{10}?

24 / 2 = 12 + 0/2

12 / 2 = 6 + 0/2

6 / 2 = 3 + 0/2

3 / 2 = 1 + 1/2

1 / 2 = 0 + 1/2

- $24_{10} = 11000_2$

Binary Addition

```
  • •        carry (in 2s)
  1 1 1 0
+ 1 0 1 1
---------------
1 1 0 0 1
```

Binary Numbers in Computing

- Easy to make fast, reliable, small devices that have only 2 states
- 1/0 represented by
  - hole/no hole in punched card
  - hi/low voltage (memory chips)
  - light bounces off/light doesn't bounce off (CDs/DVDs)
  - magnetic charge present/no magnetic charge (disks)

Measuring Data

We can group number of binary digits and refer to the group sizes by special names:

- 1 bit(b) = 2^1 = represents 2 different values
- 1 byte(B) = 8 bits = 2^8 = 256 values
- 1 kilobyte(KB) = 1024 bytes = 2^{10} bytes
- 1 megabyte(MB) = 1024 KB = 2^{20} bytes
- 1 gigabyte(GB) = 1024 MB = 2^{30} bytes
- 1 terabyte(TB) = 1024 GB = 2^{40} bytes

Other Number Systems

- Hexadecimal (base 16)
  - digits: 0, 1, 2, …, 9, A(10), B(11), …, F(15)
  - example: 5B7 = 5 x 16^2 + 11 x 16 + 7 = 1463_{10}
- Octal
  - digits: 0, 1, …, 7
  - example: 239 an invalid octal number

Hexadecimal Addition

```
 •   •      carry (in 16s)
 D F 6 D
+ 2 4 6 C
------------------
1 0 3 D 9
```

D + 2 + 1 = 13 + 3 = 16 = 16 + 0

F + 4 = 15 + 4 = 19 = 16 + 3

6 + 6 + 1 = 13 = D

D + C = 13 + 12 = 25 = 16 + 9

Representing Text

- Each letter is encoded using 1 byte
- ASCII (***A**merican **S**tandard **C**ode for **I**nformation **I**nterchange*) table

Space: “ ”

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00 NUL	01 SOH	02 STX	03 ETX	04 EOT	05 ENQ	06 ACK	07 BEL	08 BS	09 HT	10 LF	11 VT	12 FF	13 CR	14 SO	15 SI
1	16 DLE	17 DC1	18 DC2	19 DC3	20 DC4	21 NAK	22 SYN	23 ETB	24 CAN	25 EM	26 SUB	27 ESC	28 FS	29 GS	30 RS	31 US
2	32 SP	33 !	34 "	35 #	36 $	37 %	38 &	39 '	40 (	41)	42 *	43 +	44 ,	45 −	46 .	47 /
3	48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7	56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
4	64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G	72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
5	80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W	88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
6	96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g	104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
7	112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w	120 x	121 y	122 z	123 {	124 |	125 }	126 ~	127 DEL

Representing Text

“M	A	R	C”
1st byte	2nd byte	3rd byte	4th byte
77	97	114	99
01001101	01100001	01110010	01100011