ARCOS Group

uc3m | Universidad Carlos III de Madrid

Lesson 2 (I)
Representation of information

Computer Structure

Bachelor in Computer Science and Engineering

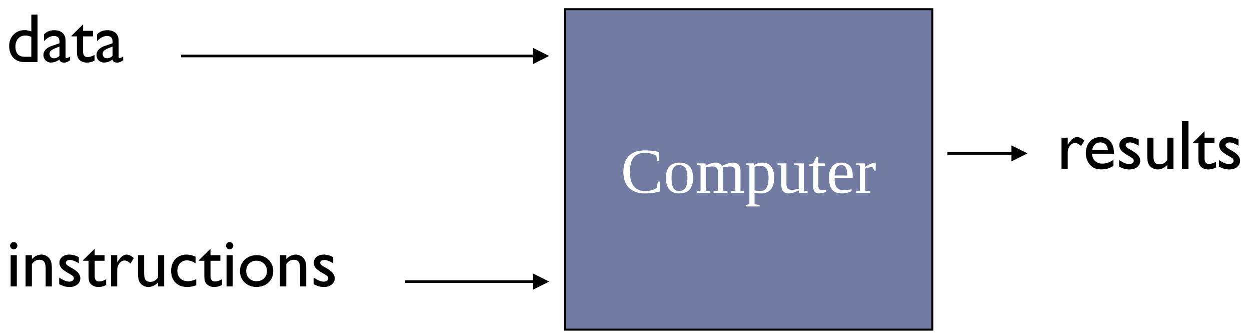
Contents

1. Introduction
   1. Motivation and goals
   2. Positional (numeral) systems

2. Representations
   1. Alphanumeric
      1. Characters
      2. Strings
   2. Numerical
      1. Natural and integer
      2. Fixed point
      3. Floating point (IEEE 754 standard)

Introduction
Computer

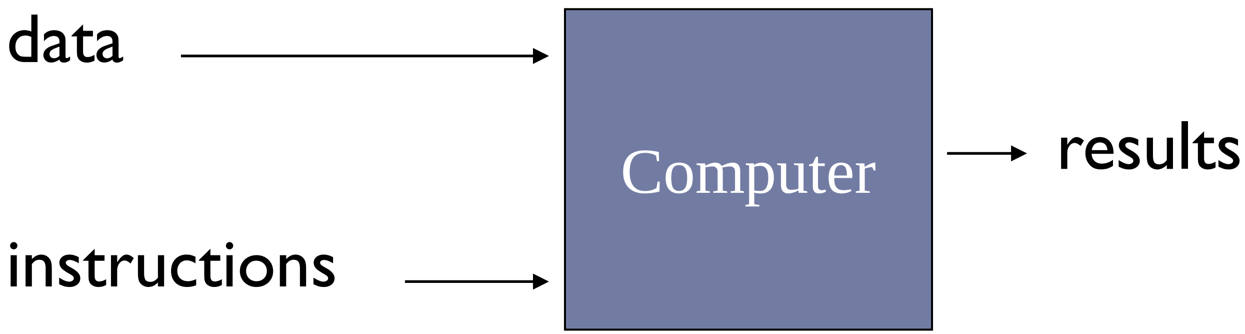
- A computer is a machine designed to process data.



- Instructions are applied and results are obtained.

Introduction
Computer

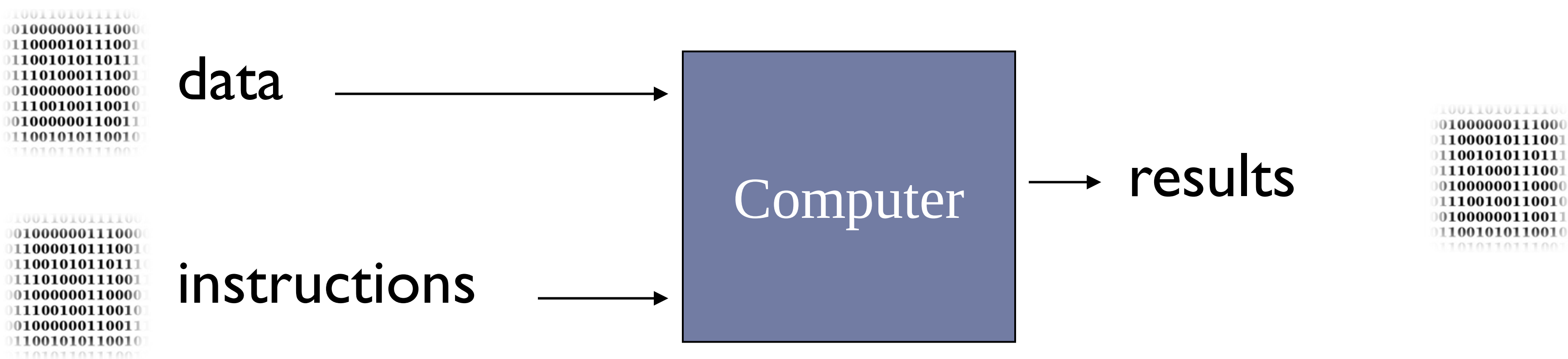
- A computer is a machine designed to process data.



- Instructions are applied and results are obtained.
- The data/information can be of different types.

Introduction
Computer

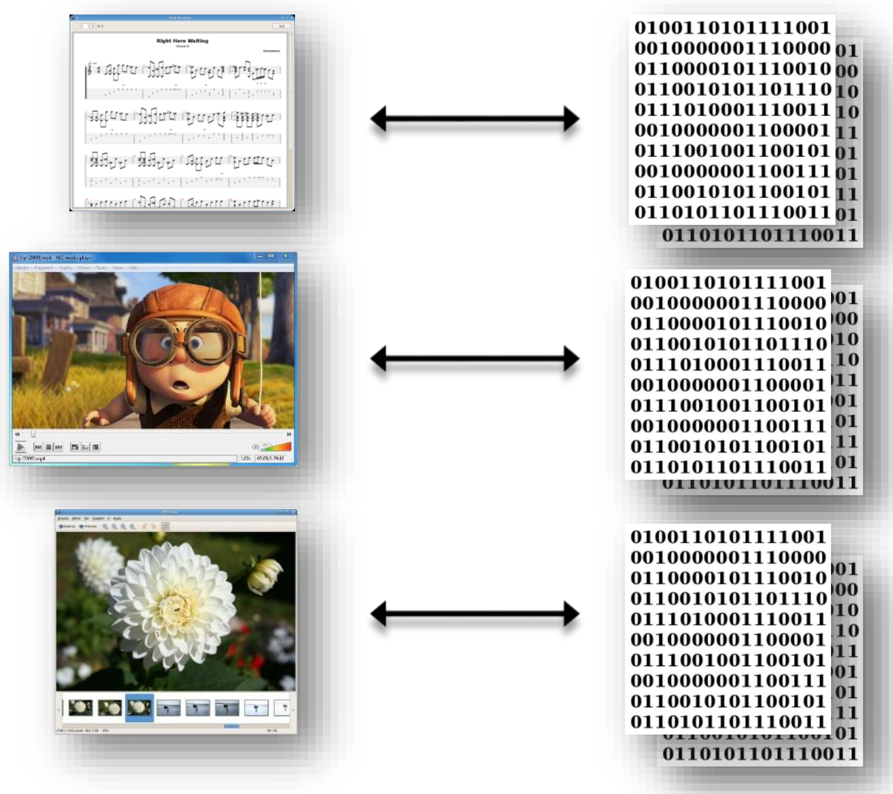
- A computer is a machine designed to process data.



- Instructions are applied and results are obtained.
- The data/information can be of different types.
- A computer uses only one representation: binary.

Introduction
Information representation

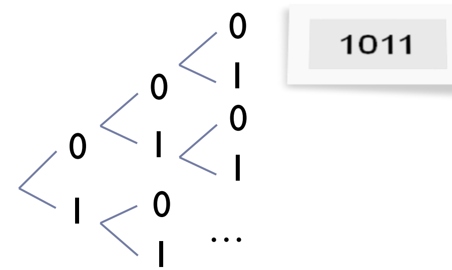
- The use of a representation allows the transformation of different types of information into binary (and vice versa).



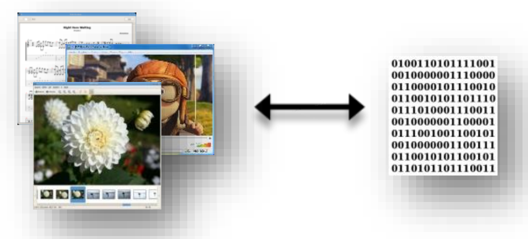
Introduction

Characteristics of the information representation

- A computer handles a finite set of values
  - Binary type (two states)
  - Finite (bounded representation)
    - Number of bits of the computer word (32/64)
      or bit (1), nibble (4), byte (8), half w., double w., …
    - With **n** bits, **2^n** different values can be encoded



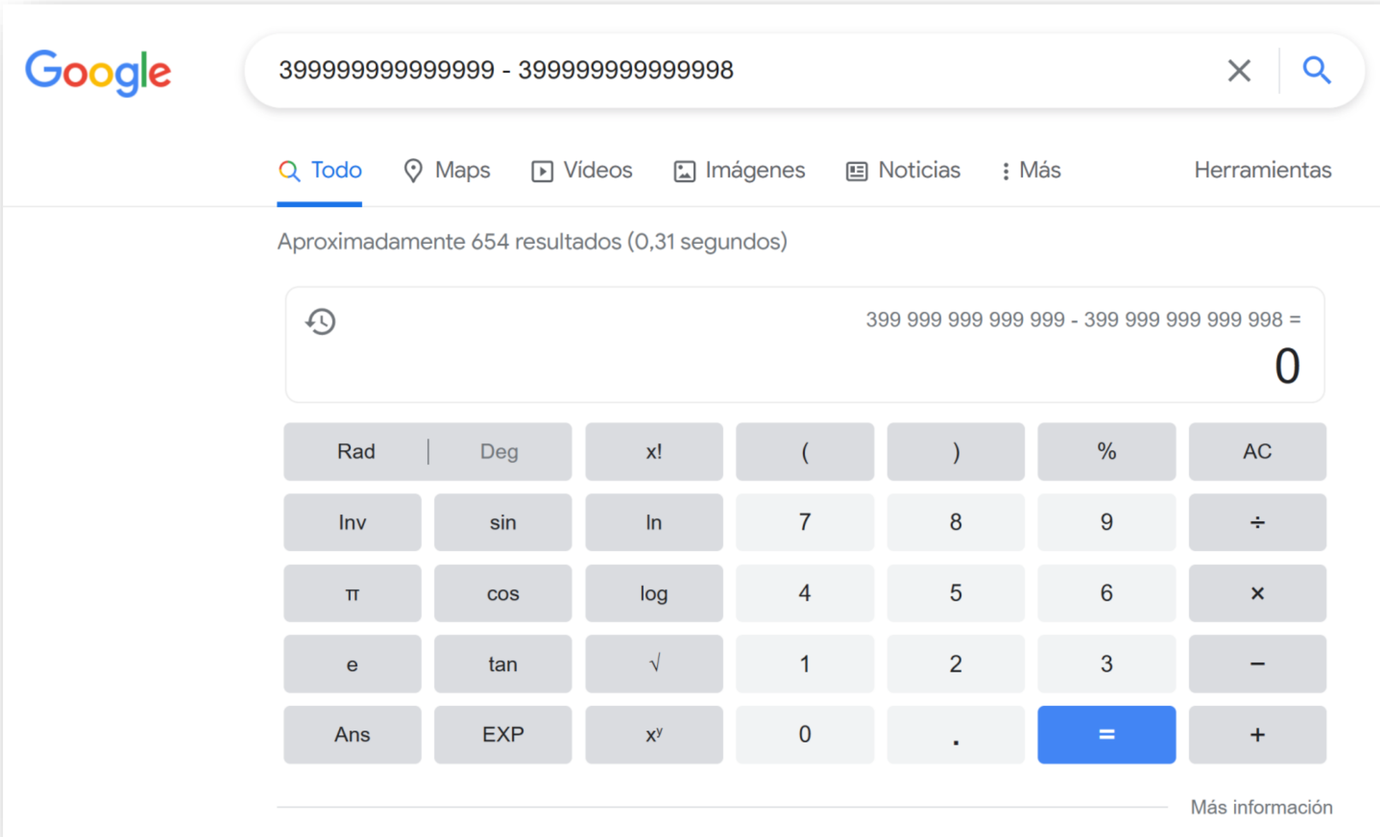
- There are some types of information that are infinite
  - Impossible to represent
    all values of natural numbers,
    real numbers, etc.



- The chosen representation has limitations.

Félix García-Carballeira, Alejandro Calderón Mateos

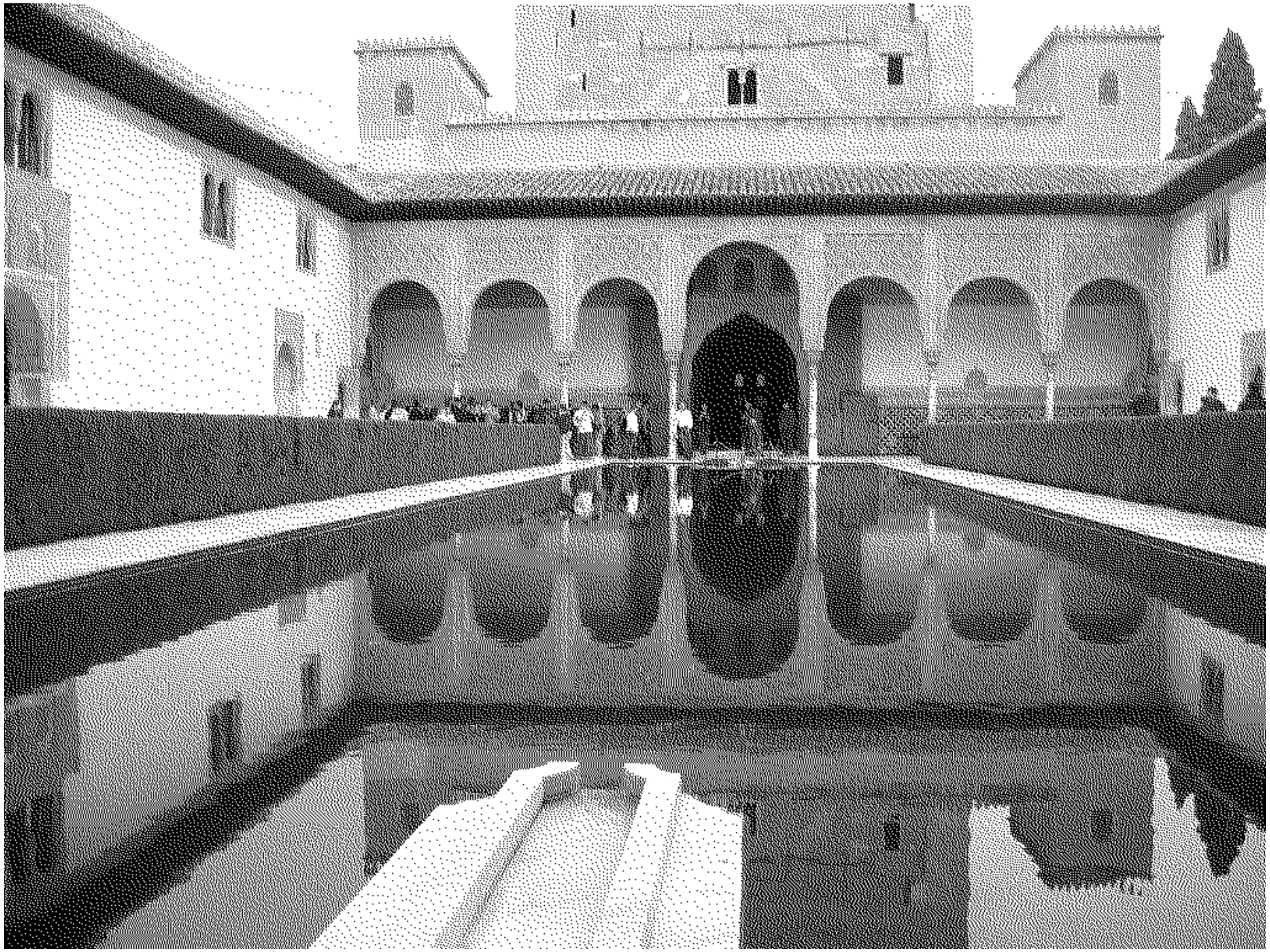
Example 1:
the Google calculator with 15 digits…



http://www.20minutos.es/noticia/415383/0/google/restar/error/

Example 2: color depth…

1 bit	2 colors
4 bits	16 colors
8 bits	256 colors



http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

Example 2:
color depth…

1 bit	2 colors
4 bits	16 colors
8 bits	256 colors

http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

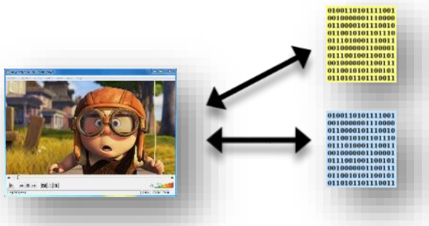
Example 2:
color depth…

1 bit	2 colors
4 bits	16 colors
8 bits	256 colors

http://platea.pntic.mec.es/~lgonzale/tic/imagen/conceptos.html

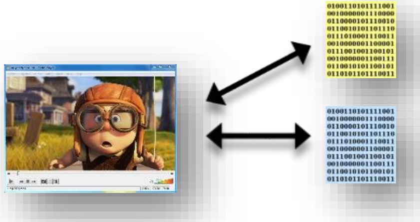
We need…

- To know possible representations:

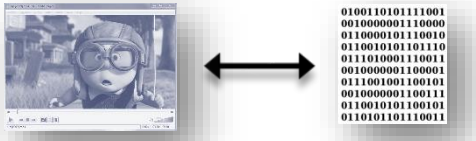


We need…

- To know possible representations:

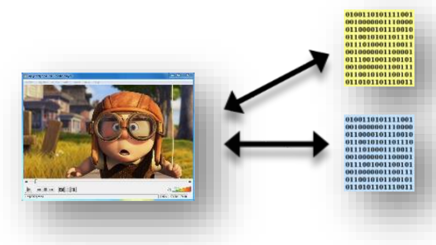


- To know the characteristics of theses representations:
  - Limitations

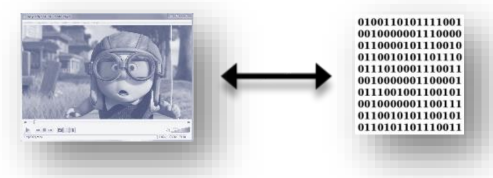


We need…

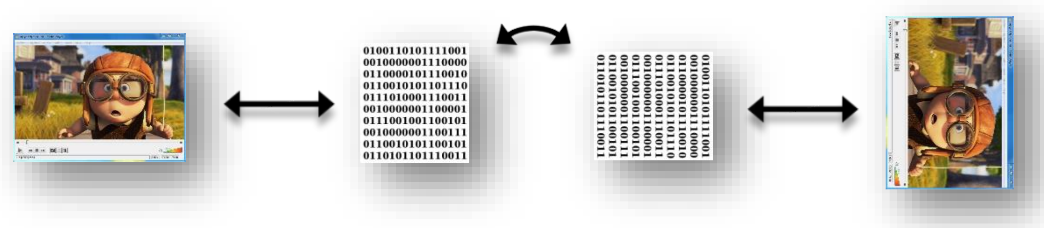
- To know possible representations:



- To know the characteristics of theses representations:
  - Limitations



- To know how work with the selected representation:



Contents

1. Introduction
   1. Motivation and goals
   2. **Positional (numeral) systems**

2. Representations
   1. Alphanumeric
      1. Characters
      2. Strings
   2. Numerical
      1. Natural and integer
      2. Fixed point
      3. Floating point (IEEE 754 standard)

Positional representation systems

- A number is defined by a ordered list of digits, each of which is affected by a scaling factor that depends on the position it occupies in the list.

- Given a numbering base b,
  a number X is defined as the list of digits:
  $X = (\ldots\ x_2\ \ x_1\ \ x_0\ ,\ \ x_{-1}\ \ x_{-2}\ \ \ldots)_b$ Con $0 \leq x_i < b$
  with a list of associated weights:
  $P = (\ldots\ b^2\ \ b^1\ \ b^0\ \ \ b^{-1}\ \ b^{-2}\ \ \ldots)_b$

- It value is:

$$V(X) = \sum_{i=-\infty}^{+\infty} b^i \cdot x_i = \cdots b^2 \cdot x_2 + b^1 \cdot x_1 + b^0 \cdot x_0 + b^{-1} \cdot x_{-1} + b^{-2} \cdot x_{-2} \cdots$$

Positional representation systems

- Decimal

  $X = \;\; 9 \;\; 7 \;\; 3 \;\; 1$

  $\ldots \; 10^3 \; 10^2 \; 10^1 \; 10^0$

- Binary

  $X = \;\; 0 \;\; 1 \;\; 0 \;\; 1$

  $\ldots \; 2^3 \; 2^2 \; 2^1 \; 2^0$

- Hexadecimal

  $X = \;\; 1 \;\; F \;\; A \;\; 8$

  $\ldots \; 16^3 \; 16^2 \; 16^1 \; 16^0$

Positional representation systems

- Decimal

  $X = 9\ 7\ 3\ 1$

  $\ldots\ 10^3\ 10^2\ 10^1\ 10^0$

- Binary

  $X = 0\ 1\ 0\ 1$

  $\ldots\ 2^3\ 2^2\ 2^1\ 2^0$

- Hexadecimal

  $X = 1\ F\ A\ 8$

  $\ldots\ 16^3\ 16^2\ 16^1\ 16^0$

From binary to hexadecimal:

- Group by 4 bits, right to left
- Each 4 bits is the value of a hexadecimal digit
- E.g.: $\underbrace{1\ 0\ 1\ 0}_{A}\ \underbrace{0\ 1\ 0\ 1}_{5}$

  0x A 5

Positional representation systems

- Decimal

  X = 9 7 3 1

  … 10^3 10^2 10^1 10^0

¿?

- Binary

  X = 0 1 0 1

  … 2^3 2^2 2^1 2^0

- Hexadecimal

  X = 1 F A 8

  … 16^3 16^2 16^1 16^0

Exercise

- To represent 342 in binary:

256	128	64	32	16	8	4	2	1
?	?	?	?	?	?	?	?	?

Exercise (solution)

- To represent 342 in binary:

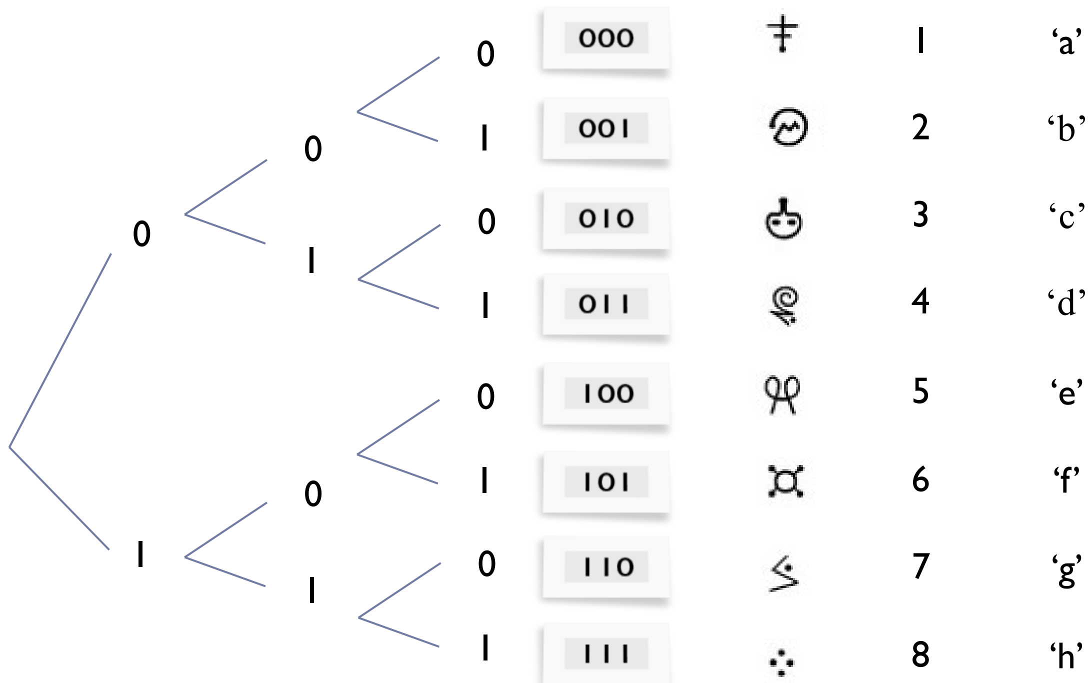
256	128	64	32	16	8	4	2	1
1	0	1	0	1	0	1	1	0

342-256=86 86-64=22 22-16=6 6-4=2 2-2=0

Félix García-Carballeira, Alejandro Calderón Mateos

Positional representation systems

- With 3 binary digits, up to 8 symbols can be represented:



Félix García-Carballeira, Alejandro Calderón Mateos

Positional representation systems

- How many values can be represented with n bits?

- How many bits are needed to represent m 'values'?

- With n bits, if the minimum representable value corresponds to the number 0,
  what is the maximum representable numerical value?

Félix García-Carballeira, Alejandro Calderón Mateos

Positional representation systems

- How many values can be represented with n bits?
  - **2^n**

  - E.g.: with 4 bits up to 16 values can be represented

- How many bits are needed to represent m 'values'?
  - $\lceil$**Log2(n)**$\rceil$ (Log2(n) round up)
  - E.g.: 6 bits are required to represent 35 values

- With n bits, if the minimum representable value corresponds to the number 0,
  what is the maximum representable numerical value?
  - **2^n-1**

Exercise

- To compute the value of (23 ones):

$$11111111111111111111111_2$$

Félix García-Carballeira, Alejandro Calderón Mateos

Exercise (solution)

- To compute the value of (23 ones):

$$11111111111111111111111_2$$

$$X = 2^{23} - 1$$

Tip:

$$
\begin{array}{rl}
 11111111111111111111_2 & = X \\
+ \; 00000000000000000001_2 & = 1 \\
\hline
 100000000000000000000_2 & = 2^{23}
\end{array}
$$

$$X = 2^{23} - 1$$

Example: operations

- Add in binary:

$$\begin{array}{ccccccc} & 1 & 1 & 1 & & & \\ & & 1 & 0 & 1 & 0 & 0 \\ + & & 1 & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 0 & 0 & 1 & 0 \end{array}$$

Example: operations

- Add in binary:

$$
\begin{array}{rcccccc}
 & & 1 & 1 & 1 & & \\
 & & 1 & 0 & 1 & 0 & 0 \\
+ & & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 0
\end{array}
$$

- Subtract in binary:

$$
\begin{array}{rccccc}
 & & & 1 \rightarrow & 1 \rightarrow & \\
 & 0 & 1 & 1 & 0 & 0 \\
- & 0 & 1 & 0 & 1 & 1 \\
\hline
 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

Contents

1. Introduction
   1. Motivation and goals
   2. Positional (numeral) systems
2. Representations
   1. **Alphanumeric**
      1. **Characters**
      2. **Strings**
   2. Numerical
      1. Natural and integer
      2. Fixed point
      3. Floating point (IEEE 754 standard)

Alphanumeric representation

- Each character is encoded as one byte.
- With n bits ⇒ up to 2^n characters can be encoded:

# bits	# characters	Includes…	Example
6	64	• 26 letter: a...z • 10 number: 0...9 • punctuation: . , ; : ... • specials: + - [...	***BCDIC***
7	128	• adds uppercases and control characters	***ASCII***
8	256	• adds accented letters, ñ, semigraphic characters	***EBCDIC*** ***ASCII extended***
16	34.168	• add support for Chinese, Arabic, ...	***UNICODE***

Example: ASCII table (7 bits)

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	¶	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

Example: ASCII table (7 bits)

control characters

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

< 32

Example: ASCII table (7 bits)

distance between uppercase and lowercase letters

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	`
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	|
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

97-65=32

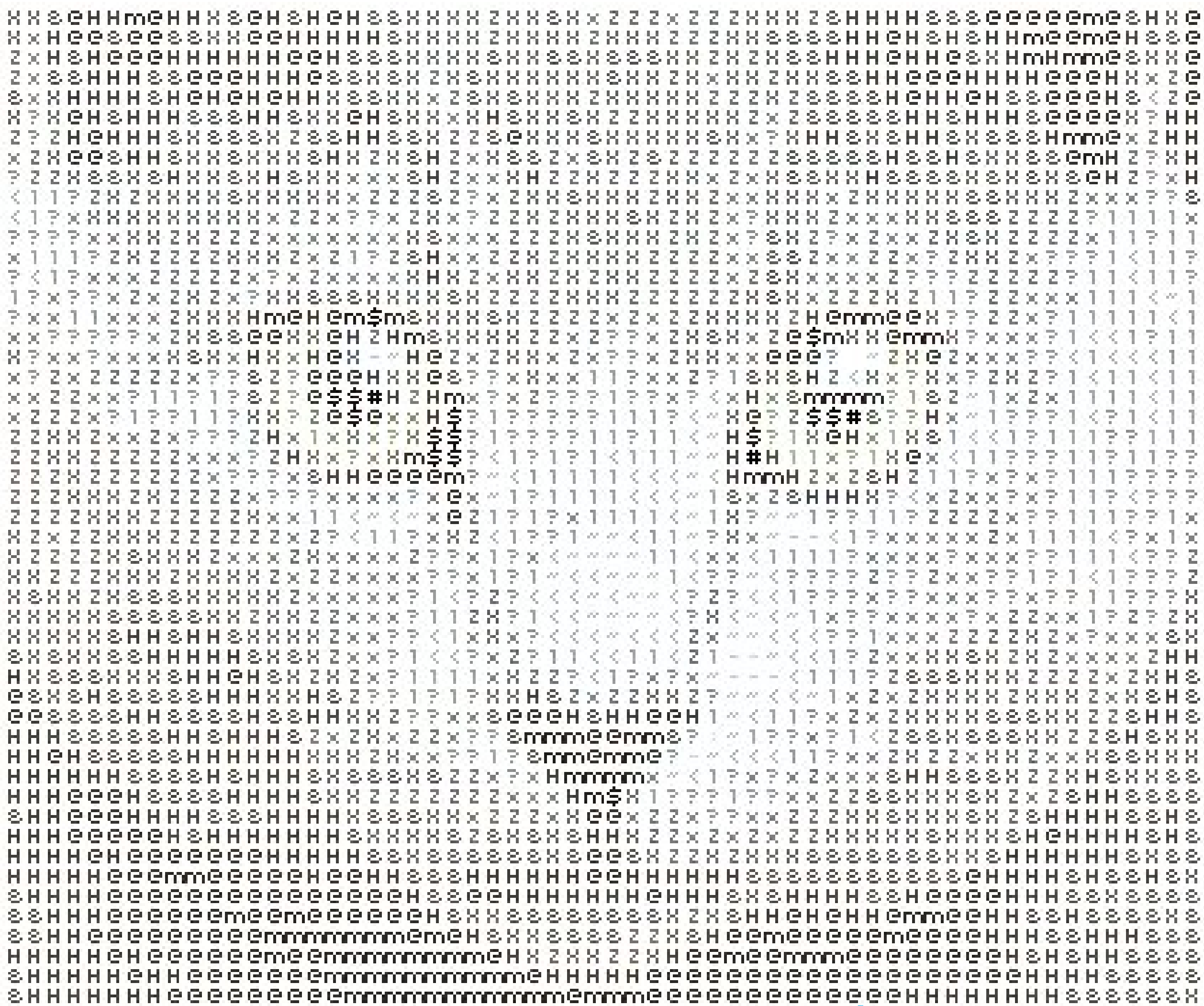
Example: ASCII table (7 bits)

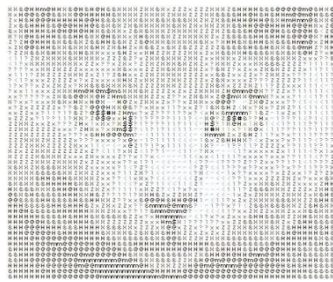
conversion of a number to a character

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☻	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	070	F	102	f
007	(beep)	BEL	039	'	071	G	103	g
008	◘	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♫	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	►	DLE	048	0	080	P	112	p
017	◄	DC1	049	1	081	Q	113	q
018	↕	DC2	050	2	082	R	114	r
019	‼	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↨	ETB	055	7	087	W	119	w
024	↑	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	¦
029	(cursor left)	GS	061	=	093	]	125	}
030	(cursor up)	RS	062	>	094	^	126	~
031	(cursor down)	US	063	?	095	_	127	⌂

6+48=54

Curiosity:
Display “image” with characters





http://www.typorganism.com/asciiomatic/

Félix García-Carballeira, Alejandro Calderón Mateos

Character strings

1000	00110011
1001	01101100
	…
1008	10100011

1. Fixed-length string:

h	e	l	l	o			
1	2	3	4	5	6	7	8

2. Variable-length string with delimiter:

h	e	l	l	o	\0		
1	2	3	4	5	6	7	8

3. Variable-length strings with length in header:

5	h	e	l	l	o		
1	2	3	4	5	6	7	8

Contents

1. Introduction
   1. Motivation and goals
   2. Positional (numeral) systems

2. Representations
   1. Alphanumeric
      1. Characters
      2. Strings
   2. **Numerical**
      1. **Natural and integer**
      2. Fixed point
      3. Floating point (IEEE 754 standard)

Numerical representation

- Classification of real numbers:
  - Naturals: 0, 1, 2, 3, ...
  - Integers: ... -3, -2, -1, 0, 1, 2, 3, ...
  - Rational: fractions (5/2 = 2,5)
  - Irrational: $2^{1/2}$, π, e, ...

- Infinite sets but finite representation space:
  - Impossible to represent all

- Characteristics of the representation used:
  - Represented element:
    Natural, integer, …
  - Representation range:
    Interval between minor and major not representable
  - Resolution of representation:
    Difference between a representable number and the following one.
    It represents the maximum error committed. It can be cte. or variable.

Most used binary representation systems

A. (Pure) binary — natural

B. Sign-Magnitude

C. One's complement (Ca 1)

D. Two's complement (Ca 2) — integer

E. Biased $2^{n-1}-1$

F. Floating point: IEEE 754 standard — real

(Pure) binary or unsigned binary [natural numbers]

- Positioning system with base 2 and without fractional part.

n-1 0

n bits

$$V(X) = \sum_{i=0}^{n-1} 2^i \cdot x_i$$

- Representation range: $[0, 2^n - 1]$
- Resolution: 1 unit

Comparative example (3 bits)

Decimal	Pure Binary
+7	111
+6	110
+5	101
+4	100
+3	011
+2	010
+1	001
+0	000
-0	N.D.
-1	N.D.
-2	N.D.
-3	N.D.
-4	N.D.
-5	N.D.
-6	N.D.
-7	N.D.

Signed binary number or Sign-Magnitude [integer numbers]

- One bit (S) is reserved for the sign ($0 \Rightarrow +$; $1 \Rightarrow -$)

n-1	n-2 ... 0
S	Magnitude (n-1 bits)

$$\text{Si } x_{n-1} = 0 \quad \mathbf{v}(X) = \sum_{i=0}^{n-2} 2^i \cdot \mathbf{x}_{\mathbf{i}}$$

$$\text{Si } x_{n-1} = 1 \quad \mathbf{v}(X) = -\sum_{i=0}^{n-2} 2^i \cdot \mathbf{x}_{\mathbf{i}}$$

$$\Rightarrow V(X) = (1 - 2 \cdot x_{n-1}) \cdot \sum_{i=0}^{n-2} 2^i \cdot x_i$$

- Representation range: $[-2^{n-1}+1,\ 2^{n-1}-1]$
- Resolution: 1 unit
- Ambiguity of zero + complex hw. for subtraction

Félix García-Carballeira, Alejandro Calderón Mateos

Comparative example (3 bits)

Decimal	Pure Binary	Sign-Magnitude
+7	111	N.D.
+6	110	N.D.
+5	101	N.D.
+4	100	N.D.
+3	011	011
+2	010	010
+1	001	001
+0	000	000
-0	N.D.	100
-1	N.D.	101
-2	N.D.	110
-3	N.D.	111
-4	N.D.	N.D.
-5	N.D.	N.D.
-6	N.D.	N.D.
-7	N.D.	N.D.

Example

- Can we represent 745_{10} in sign-magnitude with 10 bits?

Example (solution)

- Can we represent 745_{10} in sign-magnitude with 10 bits?

- With 10 bits the range in sign-magnitude is:
  $[-2^9+1,\ldots,-0,+0,\ldots 2^9-1] \Rightarrow [-511, 511]$
  then, **we cannot represent** 745

Félix García-Carballeira, Alejandro Calderón Mateos

One's complement (to the base minus one) [integer] (1/3)

- **Positive number**:
  is represented in pure binary with n-1 bits

n-1	n-2 ... 0
0	Magnitude

$$V(X) = \sum_{i=0}^{n-1} 2^i \cdot x_i = \sum_{i=0}^{n-2} 2^i \cdot x_i$$

- Representation range (+): $[0, 2^{n-1} - 1]$
- Resolution: 1 unit

One's complement (to the base minus one) [integer] (2/3)

- **Negative number**:
  - Complemented to the base minus one.
  - The number $X < 0$ is represented as $2^n - X - 1$ with n bits

n-1	n-2 ... 0
1	1'C of magnitude

$$V(X) = -2^n + \sum_{i=0}^{n-1} 2^i \cdot y_i + 1$$

- Representation range (-): $[-(2^{n-1}-1), -0]$
- Resolution: 1 unit

One's complement (to the base minus one) [integer] (3/3)

Tip: $C_a1(X) = X$
$C_a1(-X)$ = change the 1's to 0's and the 0's to 1's

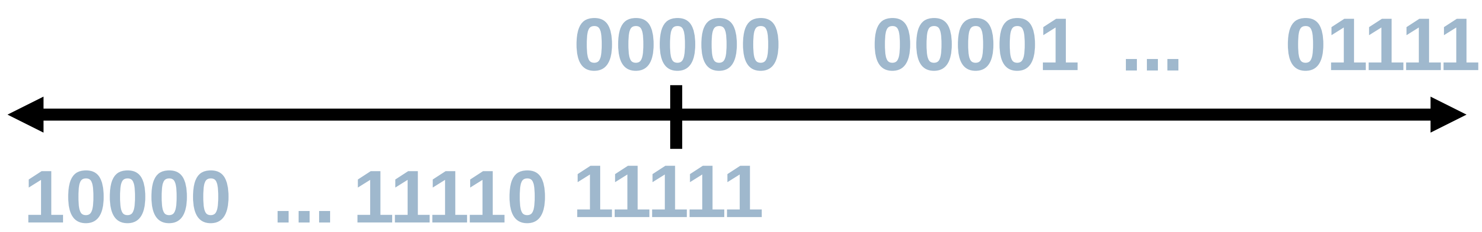
- Example: For n=4 ⇒ the value $+3_{10} = 0011_2$
- Example: For n=4 ⇒ the value $-3_{10} = 1100_2$
  - - ⇒ 1 (sign bit and also part of magnitude)
  - $C_a1(3) \Rightarrow 2^4 - 0011_2 - 1 = 2^4 - 3 - 1 = 12 \Rightarrow 1100_2$

- Representation range: $[-2^{n-1}+1, 2^{n-1}-1]$
- Resolution: 1 unit
- Zero has a double representation (+0 y -0)
- Symmetrical range

Félix García-Carballeira, Alejandro Calderón Mateos

Ones' complement

- Positive numbers have a 0 in the most significant bit.



- Negative numbers have a 1 in the most significant bit.

Comparative example (3 bits)

Decimal	Pure Binary	Sign-Magnitude	One's complement
+7	111	N.D.	N.D.
+6	110	N.D.	N.D.
+5	101	N.D.	N.D.
+4	100	N.D.	N.D.
+3	011	011	011
+2	010	010	010
+1	001	001	001
+0	000	000	000
-0	N.D.	100	111
-1	N.D.	101	110
-2	N.D.	110	101
-3	N.D.	111	100
-4	N.D.	N.D.	N.D.
-5	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.

Example

With n = 5 bits and using one's complement:

- How is represented X = 5?

- How is represented X = -5?

- What is the value of 00111 in 1's complement?

- What is the value of 11000 in 1's complement?

Example (solution)

With n = 5 bits and using one's complement:

- How is represented X = 5?
  - Because is positive then is like (pure) binary
    - 00101
- How is represented X = -5?
  - Because is negative, then 5 is complemented to one (00101)
    - 11010
- What is the value of 00111 in 1's complement?
  - Because is positive then its value is 7
- What is the value of 11000 in 1's complement?
  - Because is negative, then is complemented and is 00111 (7)
    - The value is -7

Two's complement (complement to the base) [integer] (1/3)

- **Positive number**:
  is represented in pure binary with n-1 bits

n-1	n-2 … 0
0	Magnitude

$$V(X) = \sum_{i=0}^{n-1} 2^i \cdot x_i = \sum_{i=0}^{n-2} 2^i \cdot x_i$$

- Representation range (+): $[0, 2^{n-1} - 1]$
- Resolution: 1 unit

Félix García-Carballeira, Alejandro Calderón Mateos

Two's complement (complement to the base) [integer] (2/3)

- **Negative number**:
  - Complemented to the base.
  - The number $X < 0$ is represented as $2^n - X$ with n bits

n-1	n-2 … 0
1	2'C of the magnitude

$$V(X) = -2^n + \sum_{i=0}^{n-1} 2^i \cdot y_i$$

- Representation range (-): $[-2^{n-1}, -1]$
- Resolution: 1 unit

Félix García-Carballeira, Alejandro Calderón Mateos

Two's complement (complement to the base) [integer] (3/3)

Tip: C a 2 (X) = X
C a 2 (-X) = C a 1 (X) + 1

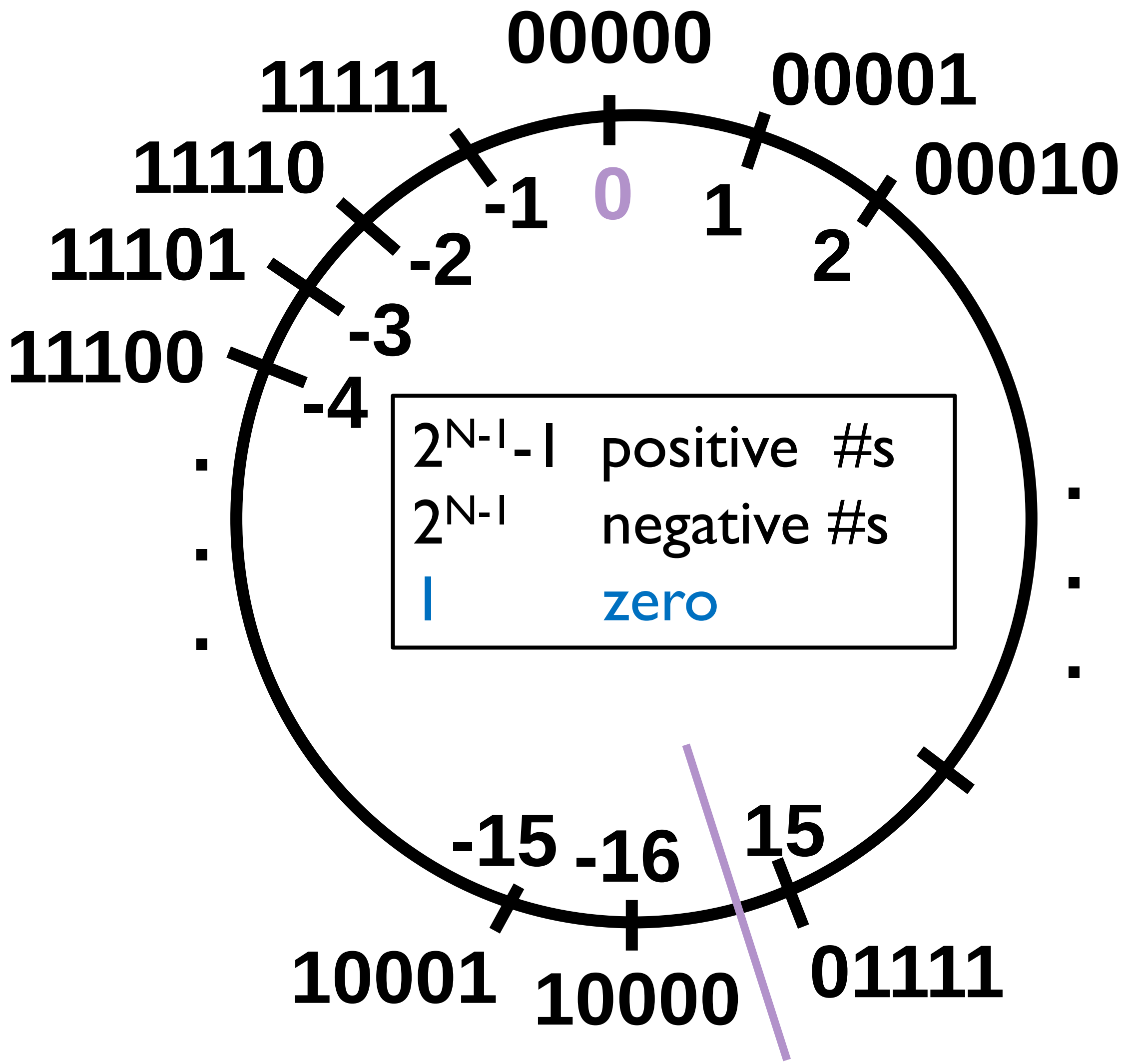
- Example: For n=4 $\Rightarrow$ + 3 = 0011_2
- Example: For n=4 $\Rightarrow$ -3 = 1101_2
  - 1 $\Rightarrow$ - (sign bit and also part of magnitude)
  - C a 2 (3) = C a 2(0011_2) = 2^4 - 3 = 13 $\Rightarrow$ 1101_2

- Representation range: $[-2^{n-1}, 2^{n-1}-1]$
- Resolution: 1 unit
- 0 has only one representation ($\nexists$ -0)
- Asymmetric range

Comparative example (3 bits)

Decimal	Pure Binary	Sign-Magnitude	One's complement	Two's complement
+7	111	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.
+3	011	011	011	011
+2	010	010	010	010
+1	001	001	001	001
+0	000	000	000	000
-0	N.D.	100	111	N.D.
-1	N.D.	101	110	111
-2	N.D.	110	101	110
-3	N.D.	111	100	101
-4	N.D.	N.D.	N.D.	100
-5	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.

Two's complement



Félix García-Carballeira, Alejandro Calderón Mateos

Biased 2^{n-1}-1 representation [integer]

- El valor X con n bits se reprsenta como X + 2^{n-1}-1
- Bias refers to the value 2^{n-1}-1

n-1	0
n bits	

$$V(X) = \sum_{i=0}^{n-1} 2^{i} \cdot x_{i} - (2^{n-1} - 1)$$

- Representation range: $[-(2^{n-1}-1), 2^{n-1}]$
- Resolution: 1 unit
- No existe ambigüedad con el 0

Comparative example (3 bits)

Decimal	Pure Binary	Sign-Magnitude	One's complement	Two's complement	Biased-3
+7	111	N.D.	N.D.	N.D.	N.D.
+6	110	N.D.	N.D.	N.D.	N.D.
+5	101	N.D.	N.D.	N.D.	N.D.
+4	100	N.D.	N.D.	N.D.	111
+3	011	011	011	011	110
+2	010	010	010	010	101
+1	001	001	001	001	100
+0	000	000	000	000	011
-0	N.D.	100	111	N.D.	N.D.
-1	N.D.	101	110	111	010
-2	N.D.	110	101	110	001
-3	N.D.	111	100	101	000
-4	N.D.	N.D.	N.D.	100	N.D.
-5	N.D.	N.D.	N.D.	N.D.	N.D.
-6	N.D.	N.D.	N.D.	N.D.	N.D.
-7	N.D.	N.D.	N.D.	N.D.	N.D.

Contents

1. Introduction
   1. Motivation and goals
   2. Positional (numeral) systems

2. Representations
   1. Alphanumeric
      1. Characters
      2. Strings
   2. **Numerical**
      1. **Natural and integer**
         1. **Arithmetic operations**
      2. Fixed point
      3. Floating point (IEEE 754 standard)

Comparison of arithmetic in B, 1C and 2C

	Binary	One's complement	Two' complement
Add	10110 01100 ---------- 100010	same as binary	same as binary
Subtract	10110 01100 ---------- 01010	add and if there is Cn-1 then add Cn-1 to total	add and if there is Cn-1 then discard it

In hardware, it is easier to operate with complement

Comparison of arithmetic in B, 1C and 2C

why add the carry to the result in 1C

	Bi…		…ent
Add			
Subtract	10110 01100 ---------- 01010	add and if there is Cn-1 then add Cn-1 to total	add and if there is Cn-1 then discard it

- -X is represented as $2^n - X - 1$
- -Y is represented as $2^n - Y - 1$
- -(X + Y) is represented as $2^n - (X+Y) - 1$

- -(X + Y) the operation gives $2^n + 2^n - (X + Y) - 2$ $\boxed{+1}$

Correction of the result by adding the carry…

Félix García-Carballeira, Alejandro Calderón Mateos

Comparison of arithmetic in B, 1C and 2C

why discard the carry in 2C

	Bin		ment
Add			
Subtract	10110 01100 ---------- 01010	add and if there is Cn-1 then add Cn-1 to total	add and if there is Cn-1 then discard it

- -X is represented as $2^n - X$
- -Y is represented as $2^n - Y$
- -(X + Y) is represented as $2^n - (X+Y)$

- -(X + Y) the operation gives $2^n + 2^n - (X + Y)$

Correction of the result by discarding the carry…

Comparison of arithmetic in B, 1C and 2C

	Binary	One's complement	Two' complement
Detect overflow	The result needs 1 bit more There are Cn	Adding ++ is –, Adding – – is + Cn <> Cn-1	Adding ++ is –, Adding – – is + Cn <> Cn-1
Sign extension	0…0 10110	1…1 10110 0…0 00110	1…1 10110 0…0 00110
…	…	…	…

Félix García-Carballeira, Alejandro Calderón Mateos

ARCOS Group

uc3m | Universidad Carlos III de Madrid

Lesson 2 (I)
Representation of information

Computer Structure

Bachelor in Computer Science and Engineering

Example

Indicate the representation of the following numbers, giving a brief justification of your answer:

1. **-32** in one's complement with **6 bits**
2. **-32** in two's complement with **6 bits**
3. **-10** in sign-magnitude with **5 bits**
4. **+14** in two's complement with **5 bits**

Example (solution)

1. With 6 bits **is not representable** in 1C: $[-2^{6-1}+1,\ldots,-0,+0,\ldots 2^{6-1}-1]$

2. 1C + 1 -> **100000**

3. Sign=1, magnitude=1010 -> **11010**

4. Positive -> 1C=2C=SM -> **01110**

Example
Arithmetic in 1's complement

- With n = 5 bits
- X = 5
  - In one's complement = 00101
- Y = 7
  - In one's complement = 00111
- ¿X + Y?

$$\begin{array}{lcl} X & = & 00101 \\ Y & = & \underline{00111+} \\ X+Y & = & 01100 \end{array}$$

- The value 01100 in one's complement is 12

Example
Arithmetic in 1's complement

- With n = 5 bits
- X = -5
  - In one's complement = complement of 00101: 11010
- Y = -7
  - In one's complement = complement of 00111: 11000
- ¿X + Y?

-X	=	11010	
-Y	=	11000+	
-(X+Y)	=	110010	A carry is generated and is added
		1	
		10011	

- The value of 10011 in one's complement is negative and the complement is -01100 = - 12

Example
Arithmetic in 2's complement

- With n = 5 bits
- X = 5
  - In two's complement = 00101
- Y = 7
  - Is two's complement = 00111
- X + Y?

X	=	00101
Y	=	00111+
X+Y	=	01100

- The value of 01100 in two's complement is 12

Example
Arithmetic in 2's complement

- With n = 5 bits
- X = -5
  - In two's complement = 11010 + 1 = 11011
- Y = -7
  - In two's complement = 11000 +1 = 11001
- X + Y?

-X = 11011
-Y = 11001+
-(X+Y) = 110100 discard the carry

- The result is 10100. The value is 01011 + 1 = 01100 = >- 12

Ejemplo
Aritmética en complemento a dos

- With n = 5 bits
- X = 8
  - In two's complement = 01000
- Y = 9
  - In two's complement = 01001
- ¿X + Y?

X	=	01000
Y	=	01001+
X+Y	=	10001

- A negative value is obtained ⇒ overflow

Ejemplo
Aritmética en complemento a dos

- With n = 5 bits
- X = -8
  - In two's complement = 10111 + 1 = 11000
- Y = -9
  - In two's complement = 10110 +1 = 10111
- X + Y?

-X	=	11000
-Y	=	10111+
-(X+Y)	=	101111

The carry is discarded

- The result 01111, is positive $\Rightarrow$ overflow

Extensión de signo en complemento a dos

- How to represent the same number of *n* bits but with *m* bits, being $n < m$?

- Example:
  - n = 4, m = 8
  - X = 0110 with 4 bits ⇒ X = 00000110 with 8 bits
  - X = 1011 with 4 bits ⇒ X = 11111011 with 8 bits

Example

- Using 5 bits, compute the followingg additions in 1's complement:
  a) 4 +12
  b) 4 -12
  c) -4 -12

Félix García-Carballeira, Alejandro Calderón Mateos

Example (solution)

- By using 5 bits in 1's complement the result is:
  a) 4 +12

00100
01100

10000 $\Rightarrow$ -15 $\Rightarrow$ negative! $\Rightarrow$ overflow

Example (solution)

- By using 5 bits in 1's complement the result is:
  b) 4 - 12

```
00100
10011
--------
10111 ⇒ -8
```

Example (solution)

- By using 5 bits in 1's complement the result is:
  c) -4 - 12

  11011
  10011

  101110 $\Rightarrow$ 6 bits are needed $\Rightarrow$ overflow

Two's complement with 32-bits

$$
\begin{aligned}
0000 \ldots 0000\ 0000\ 0000\ 0000_{2c} &= 0_{(10} \\
0000 \ldots 0000\ 0000\ 0000\ 0001_{2c} &= 1_{(10} \\
0000 \ldots 0000\ 0000\ 0000\ 0010_{2c} &= 2_{(10} \\
&\ldots \\
0111 \ldots 1111\ 1111\ 1111\ 1101_{2c} &= 2{,}147{,}483{,}645_{(10} \\
0111 \ldots 1111\ 1111\ 1111\ 1110_{2c} &= 2{,}147{,}483{,}646_{(10} \\
0111 \ldots 1111\ 1111\ 1111\ 1111_{2c} &= 2{,}147{,}483{,}647_{(10}
\end{aligned}
$$

$$
\begin{aligned}
1000 \ldots 0000\ 0000\ 0000\ 0000_{2c} &= -2{,}147{,}483{,}648_{(10} \\
1000 \ldots 0000\ 0000\ 0000\ 0001_{2c} &= -2{,}147{,}483{,}647_{(10} \\
1000 \ldots 0000\ 0000\ 0000\ 0010_{2c} &= -2{,}147{,}483{,}646_{(10} \\
&\ldots \\
1111 \ldots 1111\ 1111\ 1111\ 1101_{2c} &= -3_{(10} \\
1111 \ldots 1111\ 1111\ 1111\ 1110_{2c} &= -2_{(10} \\
1111 \ldots 1111\ 1111\ 1111\ 1111_{2c} &= -1_{(10}
\end{aligned}
$$