

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)

Contents

1. Introduction

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

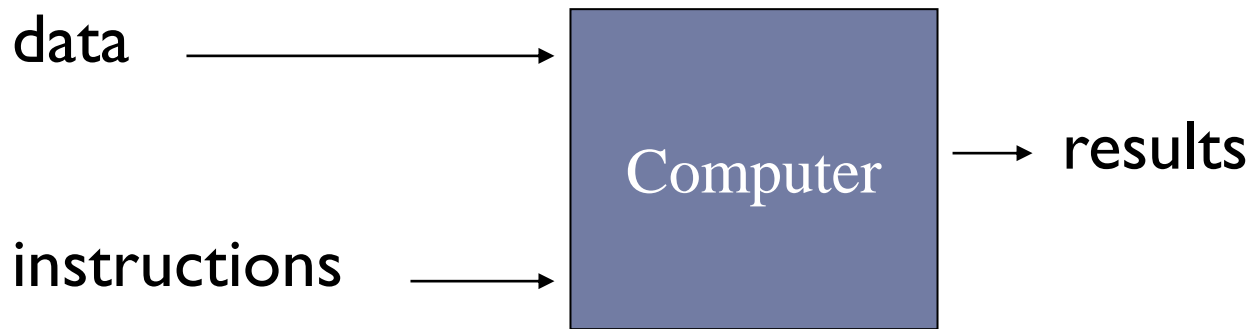
2. Representations

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.

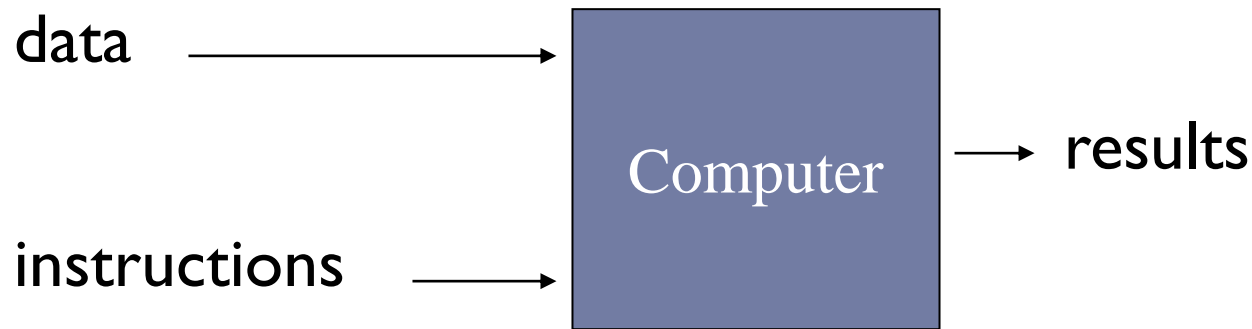


- ▶ Instructions are applied and results are obtained.

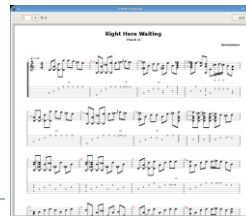
Introduction

Computer

- ▶ A computer is a machine designed to process data.



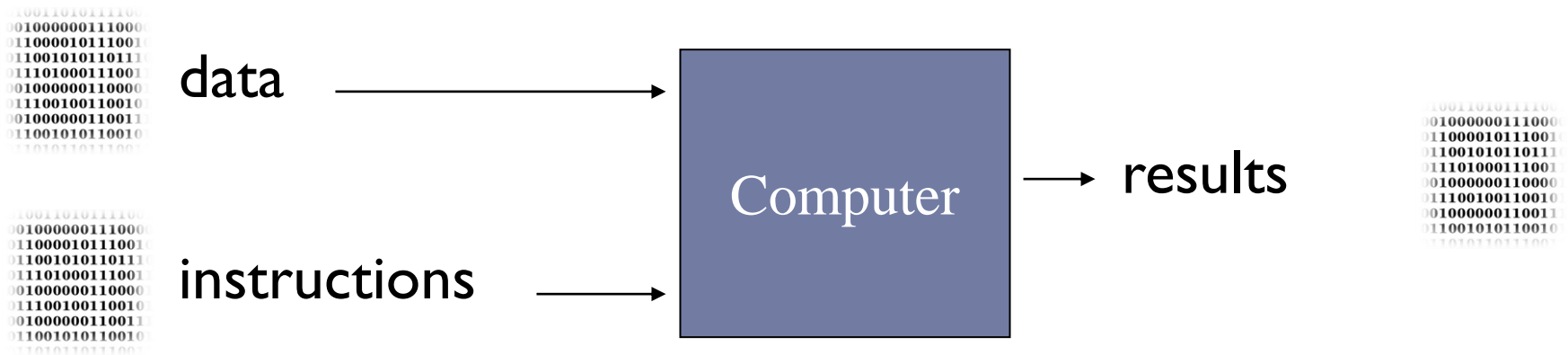
- ▶ Instructions are applied and results are obtained.
- ▶ The data/information can be of **different types**.



Introduction

Computer

- ▶ A computer is a machine designed to process data.

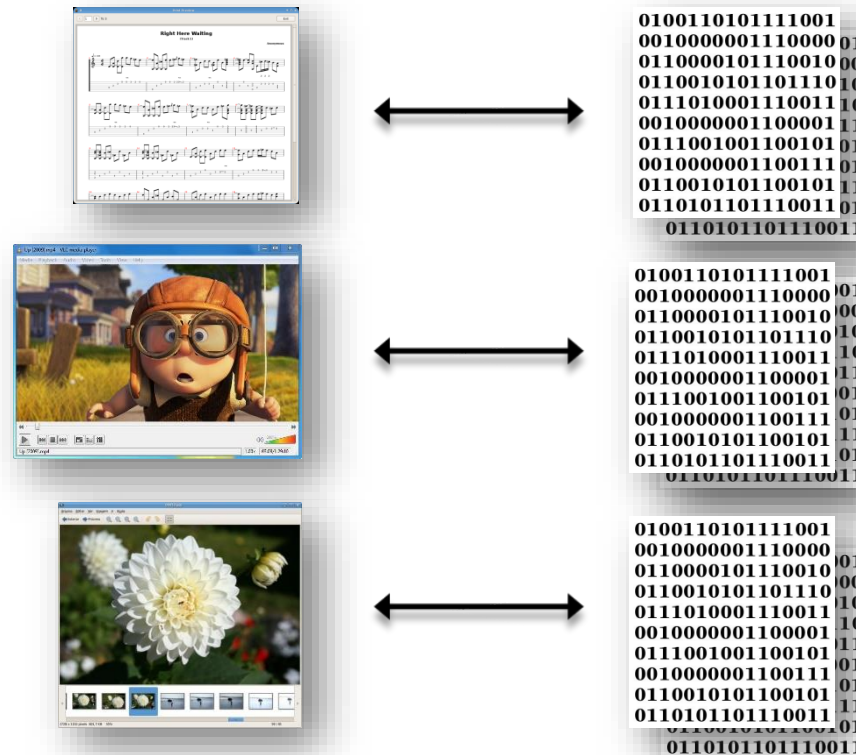


- ▶ Instructions are applied and results are obtained.
- ▶ The data/information can be of **different types**.
- ▶ A computer uses only one representation: **binary**.

Introduction

Information representation

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

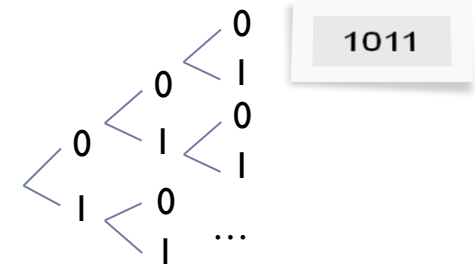


Introduction

Characteristics of the information representation

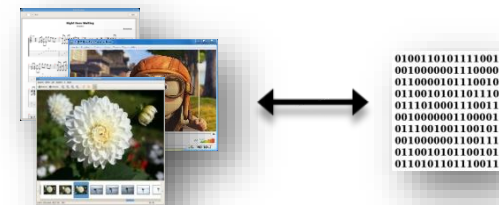
- ▶ A computer handles a finite set of values

- ▶ 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



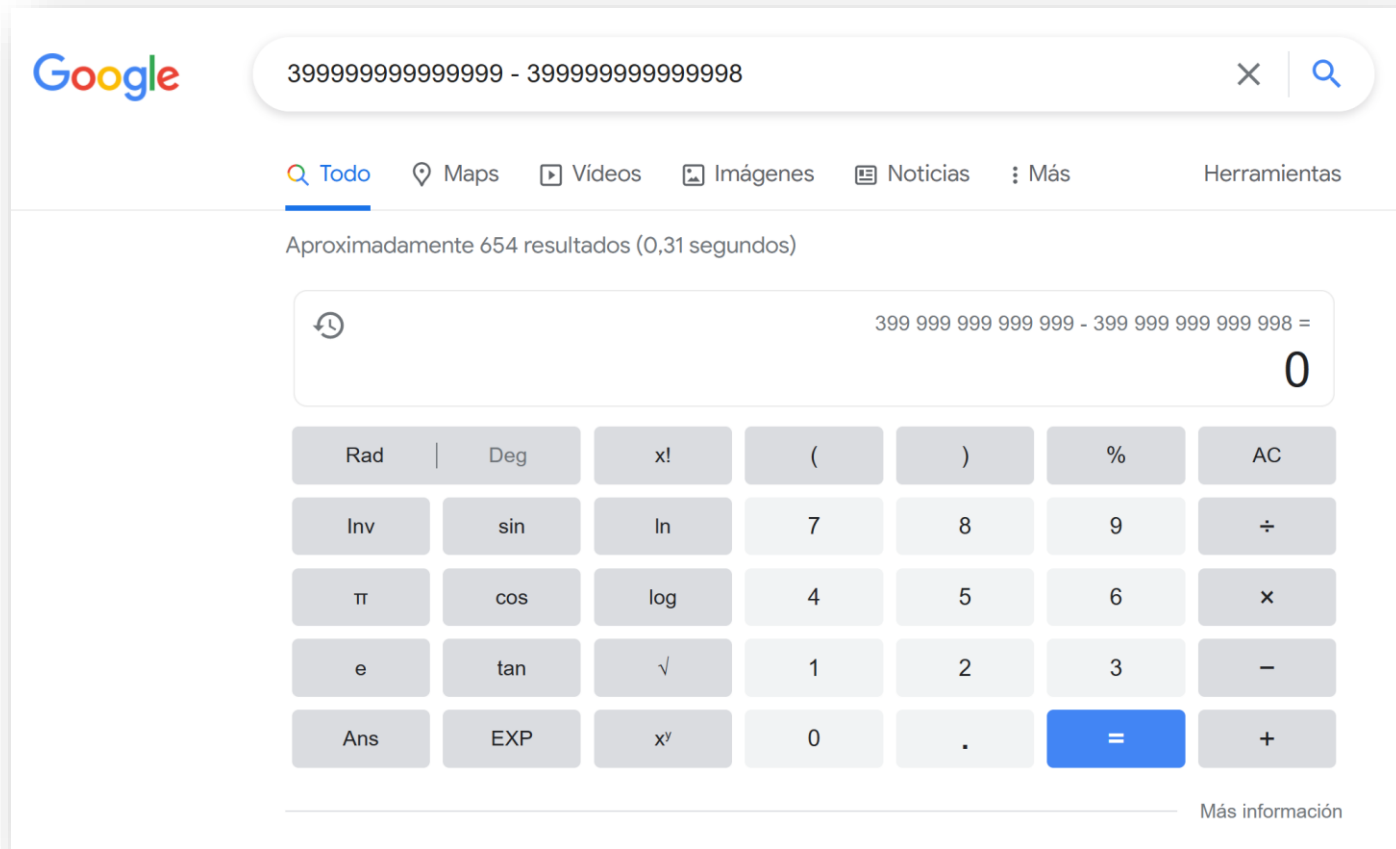
- ▶ There are some types of information that are infinite

- ▶ Impossible to represent all values of natural numbers, real numbers, etc.



- ▶ The chosen representation has limitations.

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...

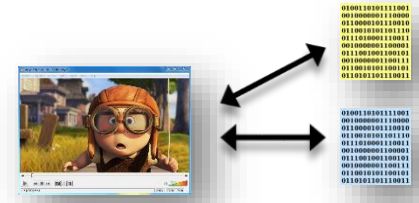
1 bit	2 colors
4 bits	16 colors
8 bits	256 colors



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

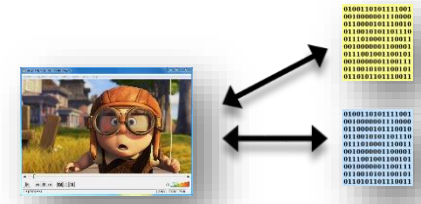
We need...

- ▶ To know possible representations:



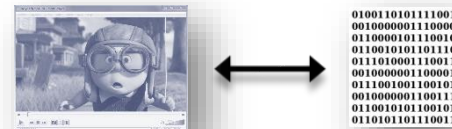
We need...

- ▶ To know **possible representations**:



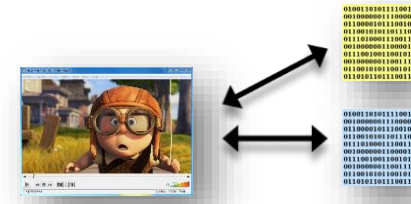
- ▶ To know the **characteristics** of these representations:

- ▶ Limitations



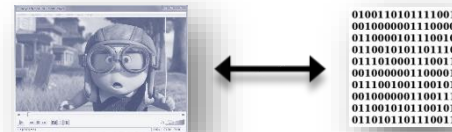
We need...

- ▶ To know **possible representations**:

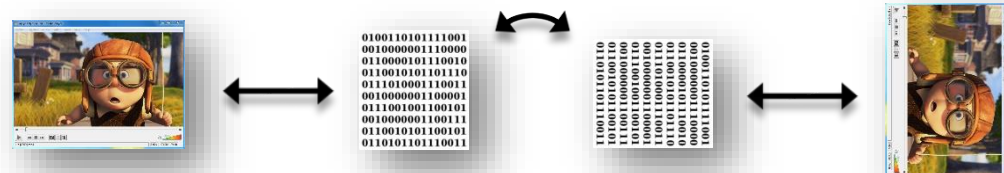


- ▶ To know the **characteristics** of these representations:

- ▶ Limitations



- ▶ To know **how work** with the selected representation:



Example of failure...

▶ **Ariane 5 explosion (first flight)**

- ▶ Sent by ESA in June 1996
- ▶ Cost of development:
10 years and 7 billion dollars
- ▶ Exploded 40 seconds after launch, at 3700 meters altitude.
- ▶ Failure due to total loss of altitude information:
 - ▶ The inertial reference system software performed the conversion of a 64-bit floating point real value to a 16-bit integer value.
 - ▶ The number to be stored was greater than 32767 (the largest 16-bit signed integer) and a conversion failure and exception occurred.



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:

$$X = (\dots x_2 x_1 x_0, x_{-1} x_{-2} \dots)_b \quad \text{Con } 0 \leq x_i < b$$

with a list of associated weights:

$$P = (\dots b^2 b^1 b^0 \quad b^{-1} b^{-2} \dots)_b$$

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:

$$X = (\dots x_2 x_1 x_0, x_{-1} x_{-2} \dots)_b \quad \text{Con } 0 \leq x_i < b$$

with a list of associated weights:

$$P = (\dots b^2 b^1 b^0 \quad b^{-1} b^{-2} \dots)_b$$

- ▶ Its value is:

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

Positional representation systems

- ▶ Decimal

$$X = \quad 9 \quad 7 \quad 3 \quad 1 \\ \quad \dots 10^3 10^2 10^1 10^0$$

- ▶ Binary

$$X = \quad 0 \quad 1 \quad 0 \quad 1 \\ \quad \dots 2^3 2^2 2^1 2^0$$

- ▶ Hexadecimal

$$X = \quad 1 \quad F \quad A \quad 8 \\ \quad \dots 16^3 16^2 16^1 16^0$$

Positional representation systems

▶ Decimal

$$X = \begin{array}{cccc} 9 & 7 & 3 & 1 \\ \dots & 10^3 & 10^2 & 10^1 & 10^0 \end{array}$$

▶ Binary

$$X = \begin{array}{cccc} 0 & 1 & 0 & 1 \\ \dots & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

▶ Hexadecimal

$$X = \begin{array}{cccc} 1 & F & A & 8 \\ \dots & 16^3 & 16^2 & 16^1 & 16^0 \end{array}$$


From binary to hexadecimal:

- ▶ Group by 4 bits, right to left
- ▶ Each 4 bits is the value of a hexadecimal digit

E.g.: $\begin{array}{ccccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ \hline & & & & & & & \\ 0x & A & & & & & 5 & \end{array}$

Positional representation systems

- ▶ Decimal

$$X = \begin{array}{cccc} 9 & 7 & 3 & 1 \\ \dots & 10^3 & 10^2 & 10^1 & 10^0 \end{array}$$


¿?

- ▶ Binary

$$X = \begin{array}{cccc} 0 & 1 & 0 & 1 \\ \dots & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

- ▶ Hexadecimal

$$X = \begin{array}{cccc} 1 & F & A & 8 \\ \dots & 16^3 & 16^2 & 16^1 & 16^0 \end{array}$$

Exercise

1 minute máx.



- To represent 342 in binary:

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

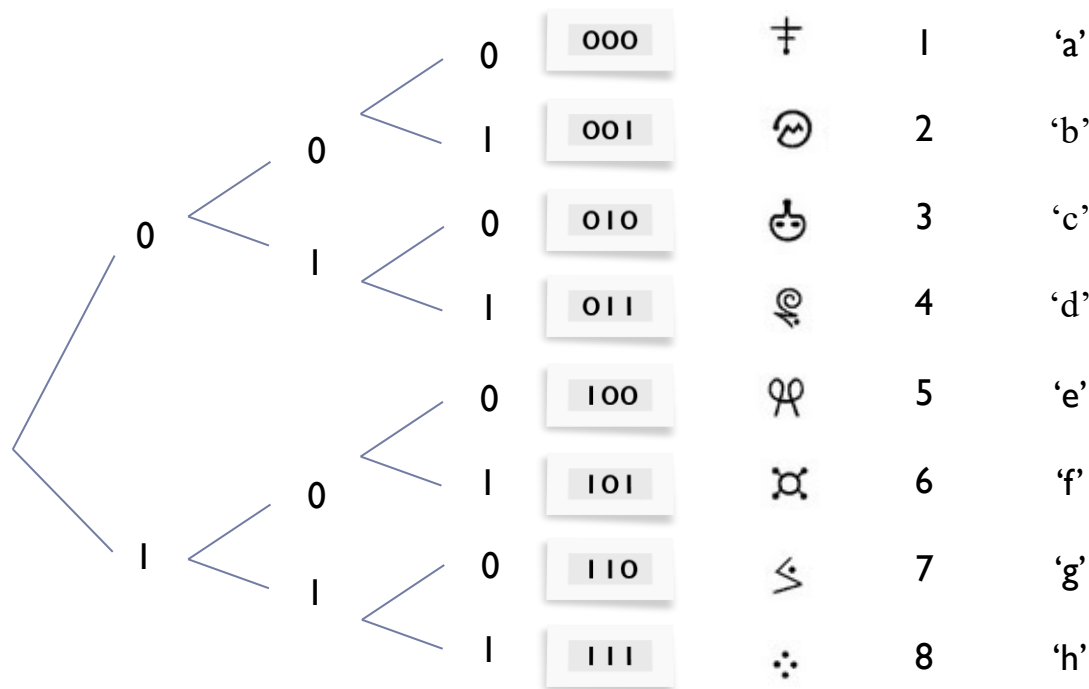
Exercise (solution)

- To represent 342 in binary:

256	128	64	32	16	8	4	2	1
	0		0		0			0
342-256=86	86-64=22	22-16=6	6-4=2	2-2=0				

Positional representation systems


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



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?

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'?
 - ▶ $\lceil \text{Log}_2(n) \rceil$ ($\text{Log}_2(n)$ 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

10 seconds máx.



- To compute the value of (23 ones):

Exercise (solution)

- To compute the value of (23 ones):

$$\text{|||||} \dots \text{|||||}_2$$

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

Tip:

$$\begin{array}{r} \text{|||||} \dots \text{|||||}_2 = X \\ + \text{000000000000000000000000}_2 = 1 \\ \hline \text{100000000000000000000000}_2 = 2^{23} \end{array}$$

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

Example: operations

- Add in binary:

$$\begin{array}{r} 1 \quad 1 \quad 1 \\ 10100 \\ + 11110 \\ \hline 110010 \end{array}$$

Example: operations

- Add in binary:

$$\begin{array}{r} 1 \quad 1 \quad 1 \\ 10100 \\ + 11110 \\ \hline 110010 \end{array}$$

- Subtract in binary:

$$\begin{array}{r} 1 \rightarrow 1 \rightarrow \\ 01100 \\ - 01011 \\ \hline 00001 \end{array}$$

Exercise

2 minutes máx.



You have a 5 liter bottle
and a 3 liter bottle.
How can you get 4 liters
just right?



Exercise (solution)

2 minutes máx.



You have a 5 liter bottle and a 3 liter bottle.
How can you get 4 liters just right?



- ▶ Fill the 5-liter jar
- ▶ Empty it into the 3-liter jar
 - ▶ There are 2 left in the 5-liter jar.
- ▶ Throw away what is in the 3-liter jar
- ▶ Transfer the 2 from the 5-liter jar to the 3-liter jar
 - ▶ There are 1 left in the 3-liter jar (-1 to 3).
- ▶ Refill the 5-liter jar
- ▶ Fill the 3-liter jar to the top, what is left in the 5-liter jar is 4 liters

Exercise

2 minutes máx.



- For the numbers 112 and -71 in decimal base, do the sum in complement to the base (base 10).

Exercise (solution)

2 minutes máx.



- ▶ Base complement of -71 is:

$$\begin{array}{r} 1000 \\ - 071 \\ \hline 929 \end{array}$$

- ▶ The sum is:

$$\begin{array}{r} 112 \\ 929 \\ \hline \text{X } 041 \end{array}$$

$$\begin{array}{r} 112 \\ -071 \\ \hline 041 \end{array}$$

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 \Rightarrow up to 2^n characters can be encoded:

# bits	# characters	Includes...	Example
6	64	<ul style="list-style-type: none">• 26 letter: a...z• 10 number: 0...9• punctuation: .,;: ...• specials: + - [...	<i>BCDIC</i>
7	128	<ul style="list-style-type: none">• adds uppercases and control characters	<i>ASCII</i>
8	256	<ul style="list-style-type: none">• adds accented letters, ñ, semigraphic characters	<i>EBCDIC</i> <i>ASCII extended</i>
16	34.168	<ul style="list-style-type: none">• add support for Chinese, Arabic, ...	<i>UNICODE</i>

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	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	☐

Copyright 1998, JimPrice.Com Copyright 1992, Loading Edge Computer Products, Inc.

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	☐

Copyright 1998, JimPrice.Com Copyright 1992, Loading Edge Computer Products, Inc.

< 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	!	<u>065</u>	<u>A</u>	<u>097</u>	<u>a</u>
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	☐

$$97-65=32$$

Copyright 1998, JimPrice.Com Copyright 1992, Loading Edge Computer Products, Inc.

Example: ASCII table (7 bits)

conversion of a number to a character

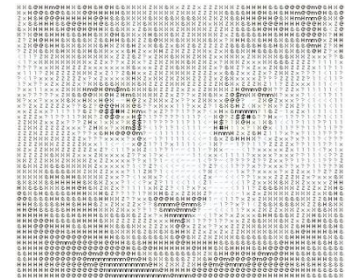
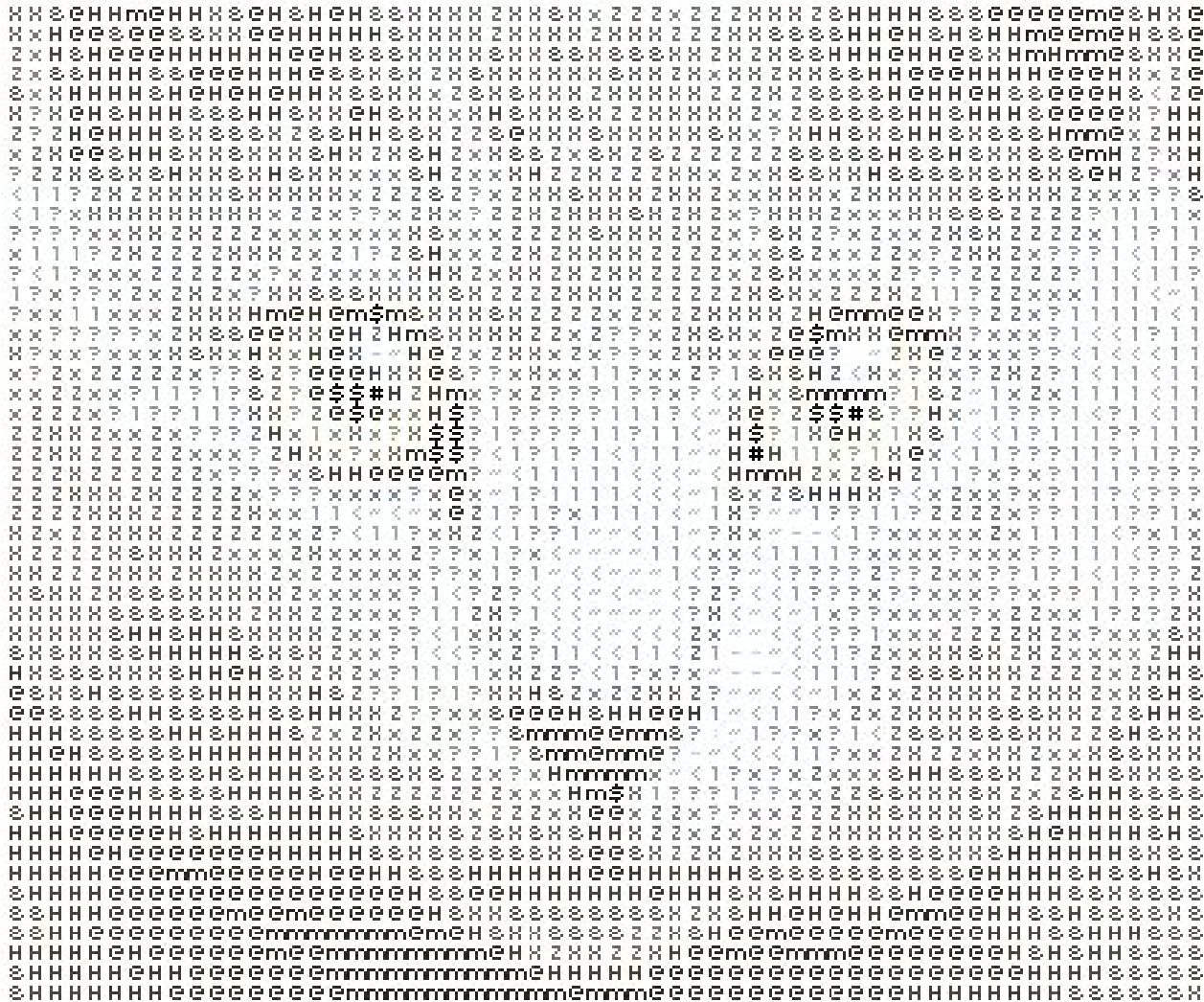
ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☹	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♣	ENQ	037	%	069	E	101	e
006	♠	ACK	038	&	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	☐

Copyright 1998, JimPrice.Com Copyright 1992, Loading Edge Computer Products, Inc.

6+48=54

Curiosity:

Display “image” with characters

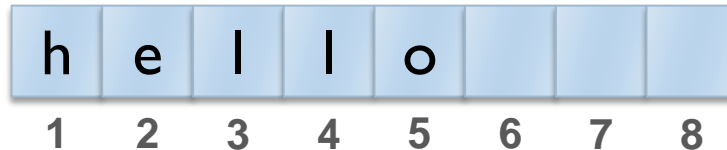


<http://www.typorganism.com/asciomatic/>

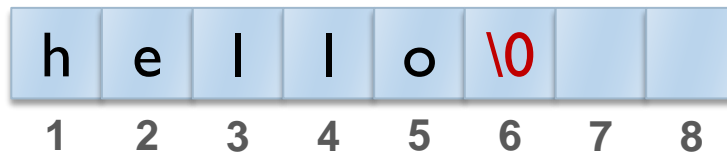
Character strings

1000	00110011
1001	01101100
...	
1008	10100011

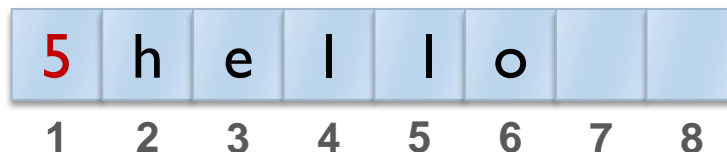
1. Fixed-length string:



2. Variable-length string with delimiter:



3. Variable-length strings with length in header:



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

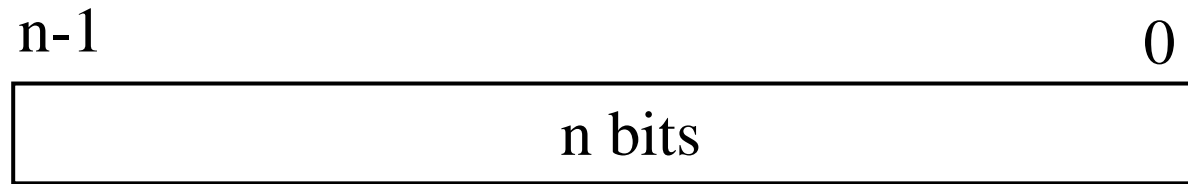
- A. (Pure) binary natural

- 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.



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

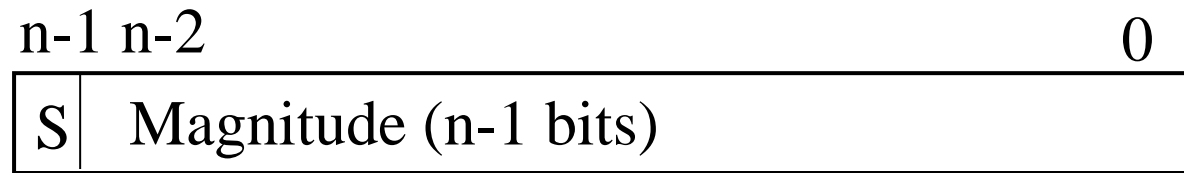
- Representation range: $[0, 2^n - 1]$
- Resolution: 1 unit

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 ($0 \Rightarrow +$; $1 \Rightarrow -$)



$$\begin{array}{l}
 \text{Si } x_{n-1} = 0 \quad v(X) = \sum_{i=0}^{n-2} 2^i \cdot x_i \\
 \text{Si } x_{n-1} = 1 \quad v(X) = - \sum_{i=0}^{n-2} 2^i \cdot x_i
 \end{array}
 \left| \Rightarrow V(X) = (1 - 2 \cdot x_{n-1}) \cdot \sum_{i=0}^{n-2} 2^i \cdot x_i \right.$$

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

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

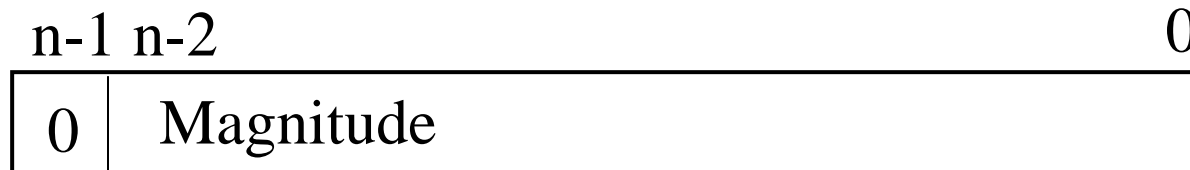
- ▶ Can we represent 745_{10} in sign-magnitude with 10 bits?

Example (solution)

- ▶ Can we represent 745_{10} in sign-magnitude with 10 bits?
- ▶ With 10 bits the range in sign-magnitude is:
 $[-2^9+1, \dots, -0, +0, \dots, 2^9-1] \Rightarrow [-511, 511]$
then, **we cannot represent 745**

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

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



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

- 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



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

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

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

Tip: $C a l (X) = X$

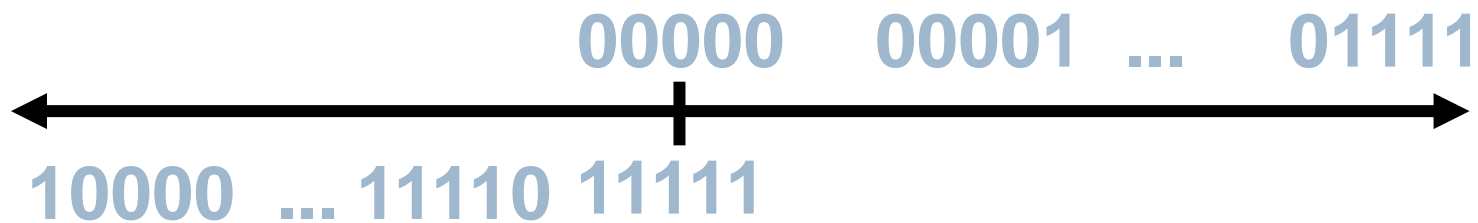
$C a l (-X)$ = change the 1's to 0's and the 0's to 1's

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

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

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?

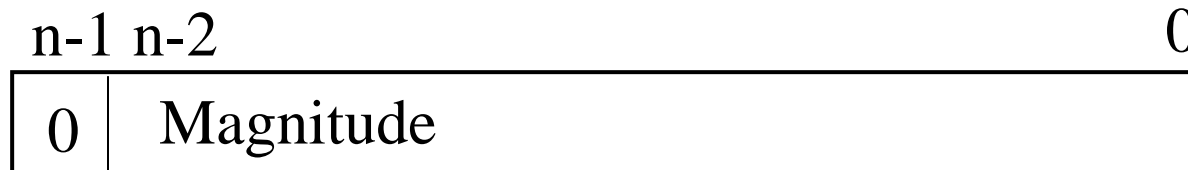
Example (solution)

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



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

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

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



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

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

Two's complement (complement to the base)

[integer] (3/3)

$$\begin{aligned}\text{Tip: } C a 2 (X) &= X \\ C a 2 (-X) &= C a 1 (X) + 1\end{aligned}$$

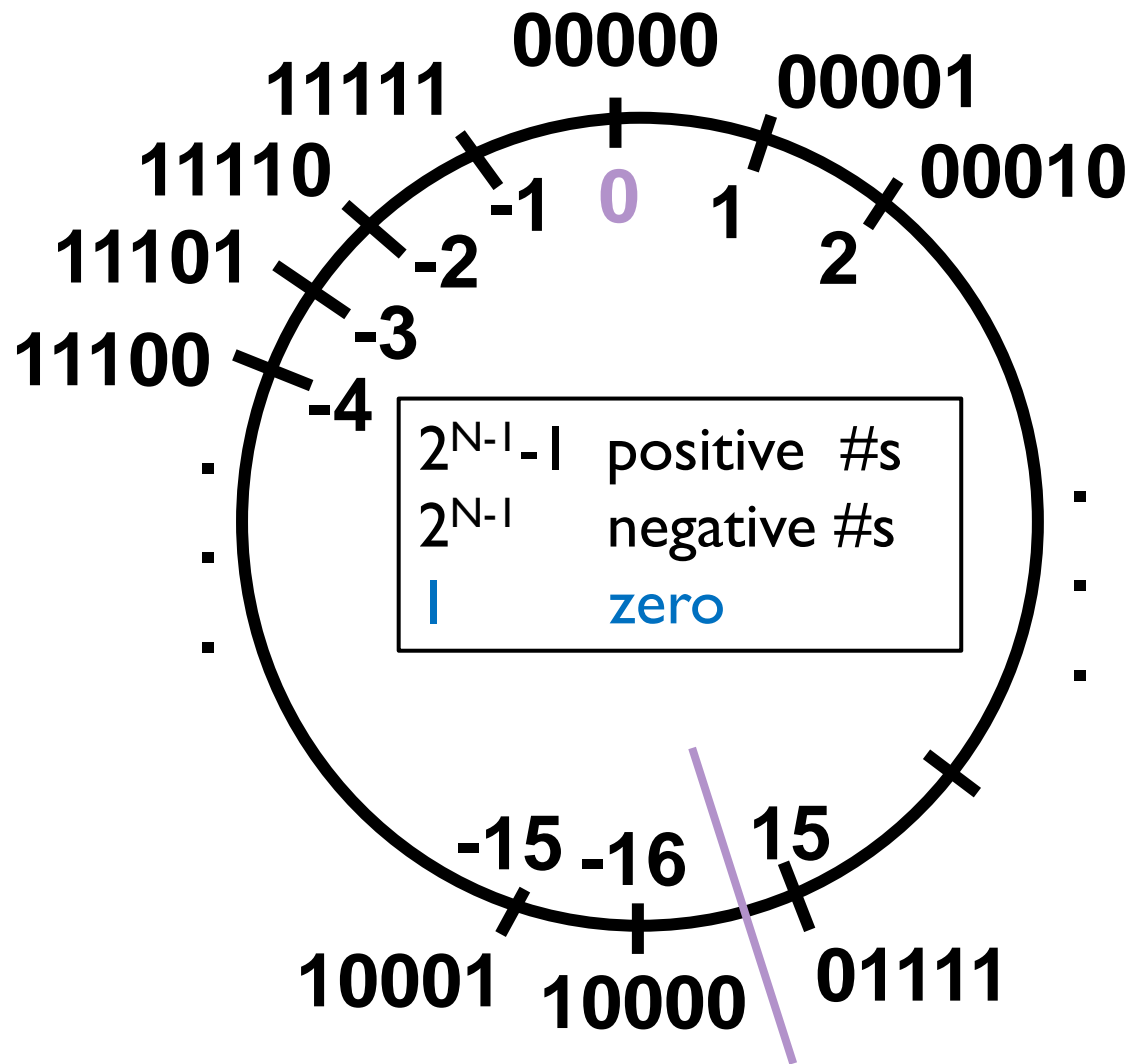
- ▶ Example: For $n=4 \Rightarrow +3 = 0011_2$
- ▶ Example: For $n=4 \Rightarrow -3 = 1101_2$
 - ▶ $1 \Rightarrow -$ (sign bit and also part of magnitude)
 - ▶ $C a 2 (3) = C a 2 (0011_2) = 2^4 - 3 = 13 \Rightarrow 1101_2$

- 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



Two's complement with 32-bits

$$0000 \dots 0000 \ 0000 \ 0000 \ 0000_{2c} = 0_{(10)}$$

$$0000 \dots 0000 \ 0000 \ 0000 \ 0001_{2c} = 1_{(10)}$$

$$0000 \dots 0000 \ 0000 \ 0000 \ 0010_{2c} = 2_{(10)}$$

...

$$0111 \dots 1111 \ 1111 \ 1111 \ 1101_{2c} = 2,147,483,645_{(10)}$$

$$0111 \dots 1111 \ 1111 \ 1111 \ 1110_{2c} = 2,147,483,646_{(10)}$$

$$0111 \dots 1111 \ 1111 \ 1111 \ 1111_{2c} = 2,147,483,647_{(10)}$$

$$1000 \dots 0000 \ 0000 \ 0000 \ 0000_{2c} = -2,147,483,648_{(10)}$$

$$1000 \dots 0000 \ 0000 \ 0000 \ 0001_{2c} = -2,147,483,647_{(10)}$$

$$1000 \dots 0000 \ 0000 \ 0000 \ 0010_{2c} = -2,147,483,646_{(10)}$$

...

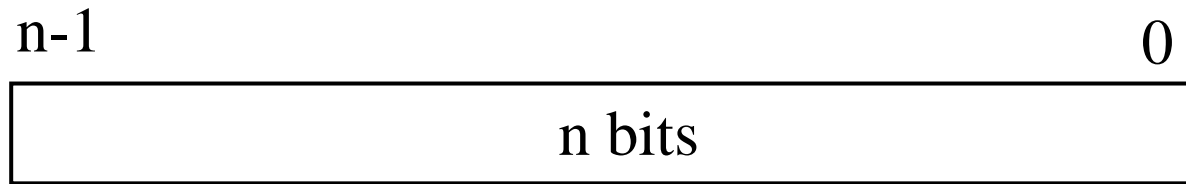
$$1111 \dots 1111 \ 1111 \ 1111 \ 1101_{2c} = -3_{(10)}$$

$$1111 \dots 1111 \ 1111 \ 1111 \ 1110_{2c} = -2_{(10)}$$

$$1111 \dots 1111 \ 1111 \ 1111 \ 1111_{2c} = -1_{(10)}$$

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

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



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

- Representation range: $[-(2^{n-1} - 1), 2^{n-1} - 1]$
- 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	$\begin{array}{r} 10110 \\ 01100 \\ \hline 100010 \end{array}$	same as binary	same as binary
Subtract	$\begin{array}{r} 10110 \\ 01100 \\ \hline 01010 \end{array}$	add and if there is C_{n-1} then add C_{n-1} to total	add and if there is C_{n-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

	Bin	ment	
Add	<div><ul style="list-style-type: none">• $-X$ 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 $2^n + 2^n - (X + Y) - 2$ <div>+ 1</div></div>		
Subtract	<div>10110 01100 ----- 01010</div>	<div>add and if there is C_{n-1} then add C_{n-1} to total</div>	<div>add and if there is C_{n-1} then discard it</div>

Correction of the result by adding the carry...

Comparison of arithmetic in B, 1C and 2C

why discard the carry in 2C

	Binary	Comment
Add	<ul style="list-style-type: none"> • $-X$ 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 $2^n + 2^n - (X + Y)$ 	
Subtract	<pre> 10110 01100 ----- 01010 </pre>	<div>add and if there is C_{n-1} then add C_{n-1} to total</div> <div>add and if there is C_{n-1} then discard it</div>

Correction of the result by discarding the carry...

Comparison of arithmetic in B, 1C and 2C

	Binary	One's complement	Two' complement
Detect overflow	<p>The result needs 1 bit more</p> <p>There are C_n</p>	<p>Adding ++ is −, Adding − − is +</p> <p>$C_n \neq C_{n-1}$</p>	<p>Adding ++ is −, Adding − − is +</p> <p>$C_n \neq C_{n-1}$</p>
Sign extension	<p>0...0 10110</p>	<p>1...1[↙]10110 0...0[↙]00110</p>	<p>1...1[↙]10110 0...0[↙]00110</p>
...

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:
 $[-2^{6-1}+1, \dots, -0, +0, \dots, 2^{6-1}-1]$
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
- ▶ ¿ $X + Y$?

$$X = 00101$$

$$Y = \underline{00111} +$$

$$X+Y = 01100$$

- ▶ 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
- ▶ $X + Y$?

$$-X = 11010$$

$$-Y = \underline{11000+}$$

$$-(X+Y) = 110010 \quad \text{A carry is generated and is added}$$

$$\begin{array}{r} \underline{1} \\ 10011 \end{array}$$

- ▶ 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
- ▶ $Y = 7$
 - ▶ In two's complement = 00111
- ▶ $X + Y?$

$$X = 00101$$

$$Y = \underline{00111} +$$

$$X+Y = 01100$$

- ▶ 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 $= 11000 + 1 = 11001$
- ▶ $X + Y?$
$$\begin{array}{rcl} -X & = & 11011 \\ -Y & = & \underline{11001} + \\ -(X+Y) & = & 110100 \end{array}$$
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
- ▶ ¿ $X + Y$?

$$X = 01000$$

$$Y = \underline{01001} +$$

$$X+Y = 10001$$

- ▶ A negative value is obtained \Rightarrow 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 $= 10110 + 1 = 10111$
- ▶ $X + Y?$

$$-X = 11000$$

$$-Y = \underline{10111} +$$

$$-(X+Y) = 10111 \quad \text{The carry is discarded}$$

- ▶ 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 $\Rightarrow X = 00000110$ with 8 bits
 - ▶ $X = 1011$ with 4 bits $\Rightarrow X = 11111011$ with 8 bits

Example

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

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 \Rightarrow -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