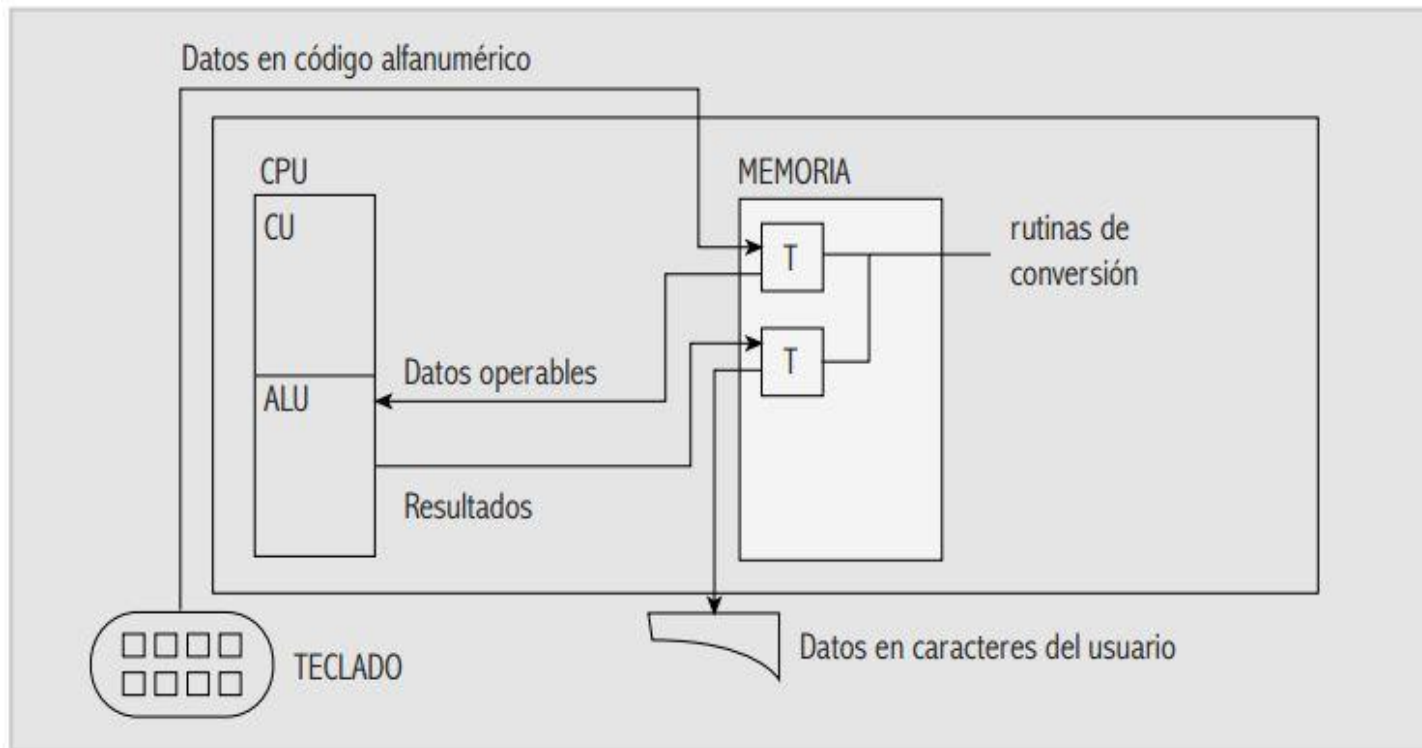


Representación de datos en la computadora. Aritmética.

Prof. Silvana Panizzo

Primera parte

Introducción



✓ CPU formada por:

- CU o Unidad de control: emite ordenes para llevar a cabo en forma secuencial y sincrónica para ejecutar una instrucción.
- ALU o Unidad aritmético lógica: es la unidad encargada de procesar datos.

✓ T: Rutinas de conversión

Algunas definiciones

- ✓ Código alfanumérico: es una convención que determina una combinación binaria para cada símbolo que se quiera representar.
- ✓ Formato: de una entidad binaria es la estructura y la cantidad de bits de un determinado tipo de dato para su tratamiento dentro de la computadora.
- ✓ Datos primitivos en C:

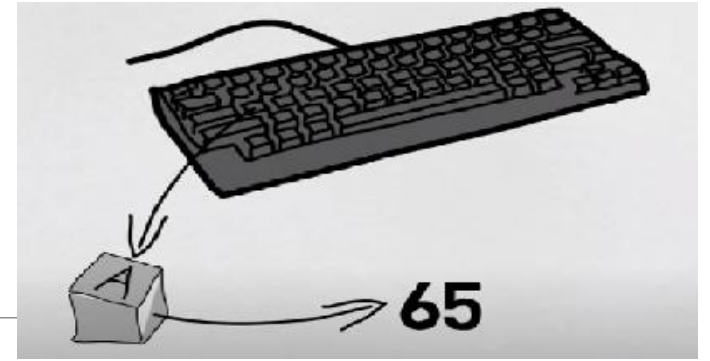
Tipo	Descripción	Bytes	Rango
char	Caracter	1	-128 a 127
int	Entero con signo	2	-32768 a 32767
float	Flotante simple	4	$-3,4 \cdot 10^{-38}$ a $3,4 \cdot 10^{38}$
double	Flotante doble	8	$-1,7 \cdot 10^{-308}$ a $1,7 \cdot 10^{308}$
long	Entero largo con signo	4	-2147483648 a 2147483647
short	Entero corto con signo	2	-32768 a 32767
long double	Flotante extendido	10	$-3,4 \cdot 10^{-4932}$ a $1,1 \cdot 10^{4932}$
unsigned	Entero sin signo	2	0 a 65535

Tipos de formato de datos

Representaciones alfanuméricas	ASCII ✖ UNICODE ✖	
Representaciones decimales (BCD)	BCD puro o natural (8421) ✖ BCD exceso tres BCD 2421 o AIKEN	
Representaciones binarias	Números enteros	Coma o punto fijo sin signo (enteros-positivos) ✖ Coma o punto fijo con signo (enteros) Coma o punto fijo con complemento a la base (enteros) ✖ Coma o punto fijo con complemento restringido (enteros)
	Números reales	Coma o punto flotante (entera y fraccionaria) ✖ Coma o punto flotante con mantisa normalizada ✖ Coma o punto flotante IEEE P754 ✖ Coma o punto flotante "exceso 64" ✖
Representaciones redundantes (detección de errores)	Códigos de paridad	Paridad vertical o a nivel carácter ✖ Paridad vertical o a nivel bloque ✖ Paridad entrelazada ✖ Código de Hamming

- ✓ Resumen de la representación de los distintos tipos de datos para poder ser procesados por el computador.
- ✓ Quedarán representados según el tipo de variables que un programador defina en las aplicaciones

Códigos de representación de caracteres alfanuméricos



- ✓ Uno de los códigos de representación de caracteres más usado es el “ASCII” (Código estándar americano para intercambio de información)
- ✓ Código ASCII de 7 bits permite determinar 128 combinaciones posibles que representan 128 símbolos distintos.
 - 1 carácter se representa con 7 bits
 - Permite $2^7 = 128$ combinaciones distintas
- ✓ Código ASCII de 8 bits ó ASCII extendido surge debido a la necesidad de agregar caracteres
 - 1 carácter se representa con 8 bit
 - Permite $2^8 = 256$ combinaciones distintas

Tabla Ascii

Tabla ASCII
128 caracteres

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0	1	2	3	4	5	6	7	8	9	<i>LF</i> 10	11	12	<i>CR</i> 13	14	15
0001	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0010	<i>SP</i> 32	!	“	#	\$	%	&	‘	()	*	+	^	-	.	/
0011	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	:	;	<	=	>	?
0100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
0101	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[\]	^	<u>95</u>
0110	` 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
0111	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{		}	~	127

Caracteres 0-31: no imprimibles (p.e. 10: LF, fin de línea; 13:CR, “retorno de carro”)

Caracter 32: “Espacio en blanco” (SP)

Caracter 127: no imprimible (DEL)

Bits de zona: Los primeros 4 bits de mayor significación en cada combinación se denominan bits de **zona** y en la tabla de doble entrada **numeran las filas 0 a 7**.

Bits de dígitos: Los últimos 4 bits (3, 2, 1 y 0) en cada combinación se denominan bits de **dígito** y en la **tabla numeran las columnas de 0 a 15, o de 0 a F en hexadecimal**.

Ejemplo:

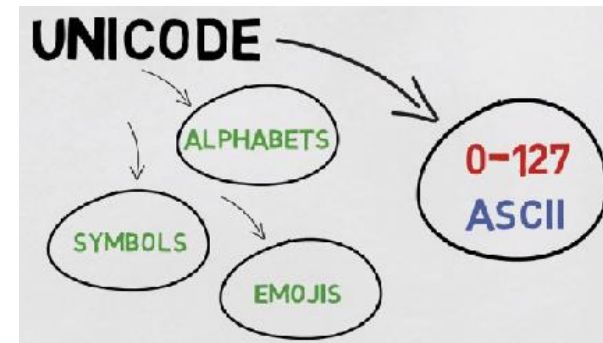
“A” = 0100 0001 = $41_{(h)} = 65_{(d)}$

Tabla Ascii extendido

Caracteres ASCII de control			Caracteres ASCII imprimibles				ASCII extendido (Página de código 437)									
00	NULL	(carácter nulo)	32	espacio	64	@	96	'	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(inicio encabezado)	33	!	65	A	97	a	129	ú	161	í	193	ł	225	ô
02	STX	(inicio texto)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
03	ETX	(fin de texto)	35	#	67	C	99	c	131	â	163	ú	195	ł	227	Ó
04	EOT	(fin transmisión)	36	\$	68	D	100	d	132	ä	164	ñ	196	Ł	228	ö
05	ENQ	(consulta)	37	%	69	E	101	e	133	å	165	Ñ	197	ł	229	Õ
06	ACK	(reconocimiento)	38	&	70	F	102	f	134	ä	166	ª	198	Ł	230	µ
07	BEL	(timbre)	39	'	71	G	103	g	135	ç	167	º	199	Ł	231	þ
08	BS	(retroceso)	40	(72	H	104	h	136	ë	168	¿	200	Ł	232	ð
09	HT	(tab horizontal)	41)	73	I	105	i	137	ë	169	©	201	Ł	233	Ú
10	LF	(nueva línea)	42	*	74	J	106	j	138	è	170	¬	202	Ł	234	Û
11	VT	(tab vertical)	43	+	75	K	107	k	139	ï	171	½	203	Ł	235	Ü
12	FF	(nueva página)	44	,	76	L	108	l	140	î	172	¾	204	Ł	236	Ý
13	CR	(retorno de carro)	45	-	77	M	109	m	141	ï	173	¿	205	Ł	237	Ÿ
14	SO	(desplaza afuera)	46	.	78	N	110	n	142	Ä	174	«	206	Ł	238	—
15	SI	(desplaza adentro)	47	/	79	O	111	o	143	Å	175	»	207	Ł	239	ˆ
16	DLE	(esc.vínculo datos)	48	0	80	P	112	p	144	É	176	⋮	208	Ł	240	≡
17	DC1	(control disp. 1)	49	1	81	Q	113	q	145	æ	177	⋮	209	Ł	241	±
18	DC2	(control disp. 2)	50	2	82	R	114	r	146	Æ	178	⋮	210	Ł	242	—
19	DC3	(control disp. 3)	51	3	83	S	115	s	147	ø	179		211	Ł	243	¼
20	DC4	(control disp. 4)	52	4	84	T	116	t	148	ö	180	¡	212	Ł	244	¶
21	NAK	(conf. negativa)	53	5	85	U	117	u	149	õ	181	À	213	Ł	245	§
22	SYN	(inactividad sinc)	54	6	86	V	118	v	150	ù	182	Á	214	Ł	246	÷
23	ETB	(fin bloque trans)	55	7	87	W	119	w	151	û	183	Â	215	Ł	247	ˆ
24	CAN	(cancelar)	56	8	88	X	120	x	152	ÿ	184	Ã	216	Ł	248	ˆ
25	EM	(fin del medio)	57	9	89	Y	121	y	153	Û	185	Ä	217	Ł	249	ˆ
26	SUB	(sustitución)	58	:	90	Z	122	z	154	Ü	186	Å	218	Ł	250	ˆ
27	ESC	(escape)	59	;	91	[123	{	155	ø	187	Æ	219	Ł	251	ˆ
28	FS	(sep. archivos)	60	<	92	\	124		156	£	188	Ç	220	Ł	252	ˆ
29	GS	(sep. grupos)	61	=	93]	125	}	157	Ø	189	Ð	221	Ł	253	ˆ
30	RS	(sep. registros)	62	>	94	^	126	~	158	×	190	¥	222	Ł	254	■
31	US	(sep. unidades)	63	?	95	_			159	f	191	γ	223	Ł	255	nbsp
127	DEL	(suprimir)														

Actividad

- ✓ Escriba su nombre en binario según la tabla ascii.
- ✓ Leer apunte subido al campus sobre Unicode.



Código de representación decimal (BCD)

- ✓ Es una convención que permite la representación de números entre 0 y 9 en bloques de binarios de 4 bits
- ✓ BCD puro o natural o 8421

<i>Valor decimal</i>	<i>BCD puro 8421</i>	<i>Valor decimal</i>	<i>BCD puro 8421</i>
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- ✓ Ejemplos

<i>Valor decimal</i>	<i>Binario puro</i>	<i>BCD puro</i>
$15_{(10)}$	$1111_{(2)}$	$0001\ 0101_{(BCD)}$
$256_{(10)}$	$2^8 = 100000000_{(2)}$	$0010\ 0101\ 0110_{(BCD)}$
$1_{(10)}$	$1_{(2)}$	$0001_{(BCD)}$

Suma en BCD 8421

Ejemplo: $99_{(10)} + 1_{(10)} = 100_{(10)}$

1	1	
	1001	1001
	0000	0001
	1010	1010
	0110	0110
0001	0000	0000
0001 = $1_{(10)}$	0000 = $0_{(10)}$	0000 = $0_{(10)}$

Si el resultado es mayor a $9_{(10)}$ entonces se suma $6_{(10)}$ en binario 0110

Sumar $6_{(10)}$ en binario me permite volver al rango de representación del código

Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Resta en BCD 8421

Ejemplo: $99_{(10)} - 1_{(10)} = 98_{(10)}$

Se C1 el valor negativo para realizar una suma $99_{(10)} + (-1_{(10)}) = 98_{(10)}$

0 = $0000 + 0110 = 0110 \rightarrow 1001$

1 = $0001 + 0110 = 0111 \rightarrow 1000$

Se suma 0110 y se invierten los bit para obtener C1

Si el resultado es mayor a $9_{(10)}$ entonces se suma $6_{(10)}$ en binario 0110

Sumar $6_{(10)}$ en binario me permite volver al rango de representación del código

Se desprecia por exceso de 10^n

Sumo 0001 por uso de C1

Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Actividad

Ejercicios:

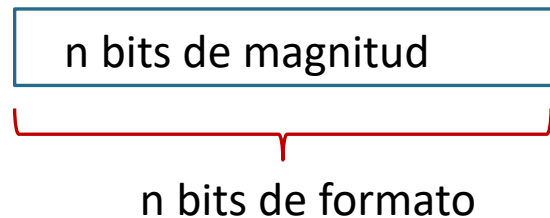
Realizar las siguientes operaciones por BCD:

- a) $25 + 2$
- b) $662 + 1045$
- c) $234 - 152$
- d) $348 - 216$
- e) $483 + 257$
- f) $599 + 345$
- g) $3482 - 638$

Código de representación binaria de números enteros

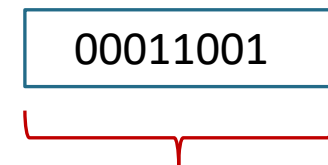
Coma fija o punto fijo sin signo para representar números enteros positivos

- ✓ Se representan como número binario.
- ✓ Como este formato sólo permite números sin signo, para la representación de un número se utiliza la totalidad de bits del formato.



Rango de representación:
 $(0, 2^n - 1)$

- ✓ Ejemplo: si quiero representar en un formato de $n = 8$ el valor $25_{(10)}$



Rango de representación:
 $(0, 2^n - 1) = (0, 255)$

Es decir, 256 combinaciones de 0 y 1 que me permiten almacenar valores entre 0 y 255 en decimal.

Código de representación binaria de números enteros

Coma fija o punto fijo con signo para representar números enteros positivos y negativos en C2

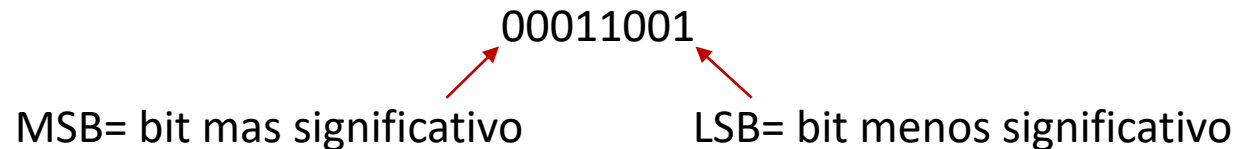
Convenio de representación:

Para poder representar magnitudes enteras positivas y negativas, lo primero que se debe definir es como se representa el signo. Este convenio adopta que el bit mas significativo (MSB) se lo utilice para identificar el signo, siendo 0 para los positivos y 1 para los negativos.

Ejemplo:

00011001

MSB= bit mas significativo LSB= bit menos significativo



Por lo tanto, si una variable está definida como entera todo número positivo comenzará de izquierda a derecha con 0 y todo número negativo comenzará con 1.

Código de representación binaria de números enteros

Coma fija o punto fijo con signo para representar números enteros positivos y negativos en C2

Convenio de representación continuación:

Es de pensar que, si tenemos formato $n = 3$ las posibles combinaciones serán 8 desde 000 hasta 111, por lo tanto las siguientes combinaciones 000, 001, 010, 011 representarán números positivos y las combinaciones 100, 101, 110 y 111 representarán números negativos.

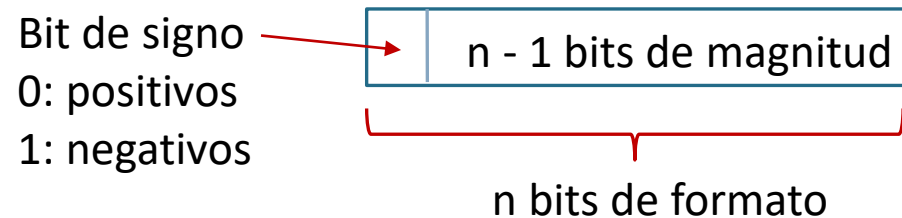
Existen diferentes formas de poder representar los valores y esto dependerá del convenio elegido, pudiendo ser Signo y Magnitud, Complemento a 1 ó **Complemento a 2**.

Para el convenio coma fija o punto fijo con signo, vamos a definir la utilización del **Complemento a 2**, que **enuncia que los números positivos estarán representados por su magnitud binaria en los $n-1$ bits restantes del formato (excluye el bit de signo) y los números negativos estarán representados por el complemento a 2 de la magnitud binaria en los $n-1$ bits restantes del formato.**

Código de representación binaria de números enteros

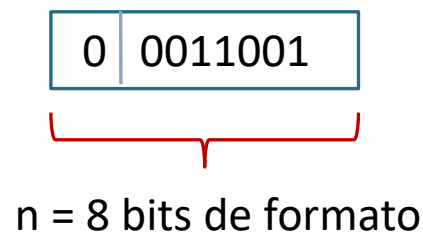
Coma fija o punto fijo con signo para representar números enteros positivos y negativos en C2

- ✓ Se representan como número binario.
- ✓ Como este formato permite números con signo, para la representación de un número se utilizan los $n - 1$ bits del formato.



Rango de representación:
 $(- 2^{n-1}, 2^{n-1} - 1)$

- ✓ Ejemplo: si quiero representar en un formato de $n = 8$ el valor $+25_{(10)}$



Rango de representación:
 $(- 2^{n-1}, 2^{n-1} - 1)$
 $(- 128, +127)$

Es decir, 128 combinaciones de 0 y 1 que me permiten almacenar valores negativos y 127 combinaciones que permiten almacenar valores positivos.

Actividad

Ejercicios:

- ✓ Leemos de la Memoria el siguiente dato: 10010100
- Calcular el valor del mismo suponiendo que se trabaja con un micro de 8 bits ($n=8$) para los siguientes casos:
 - a) Suponiendo que dicha variable fue definida como Entera sin signo
 - b) Suponiendo que dicha variable fue definida como Entera bajo el convenio de Complemento a 2
- Respuestas
 - a) 148
 - b) Hacemos C2 $-(01101100)$ resultando -108
- Ahora realizar el mismo ejercicio para lo que leemos de la memoria es 00010011