Introduction To Information And Communication Technologies

BSCS

Zahoor ul Hassan

Week 4

Fall, 2025

Number Systems

Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	Yes
Hexadecimal	16	0, 1, … 9, A, B, … F	No	Yes

Quantities/Counting (1 of 3)

Decimal	Binary	Octal	Hexadecimal
0	000	0	0
1	001	1	1
2	010	2	2
3	011	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Quantities/Counting (2 of 3)

Decimal	Binary	Octal	Hexadecimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

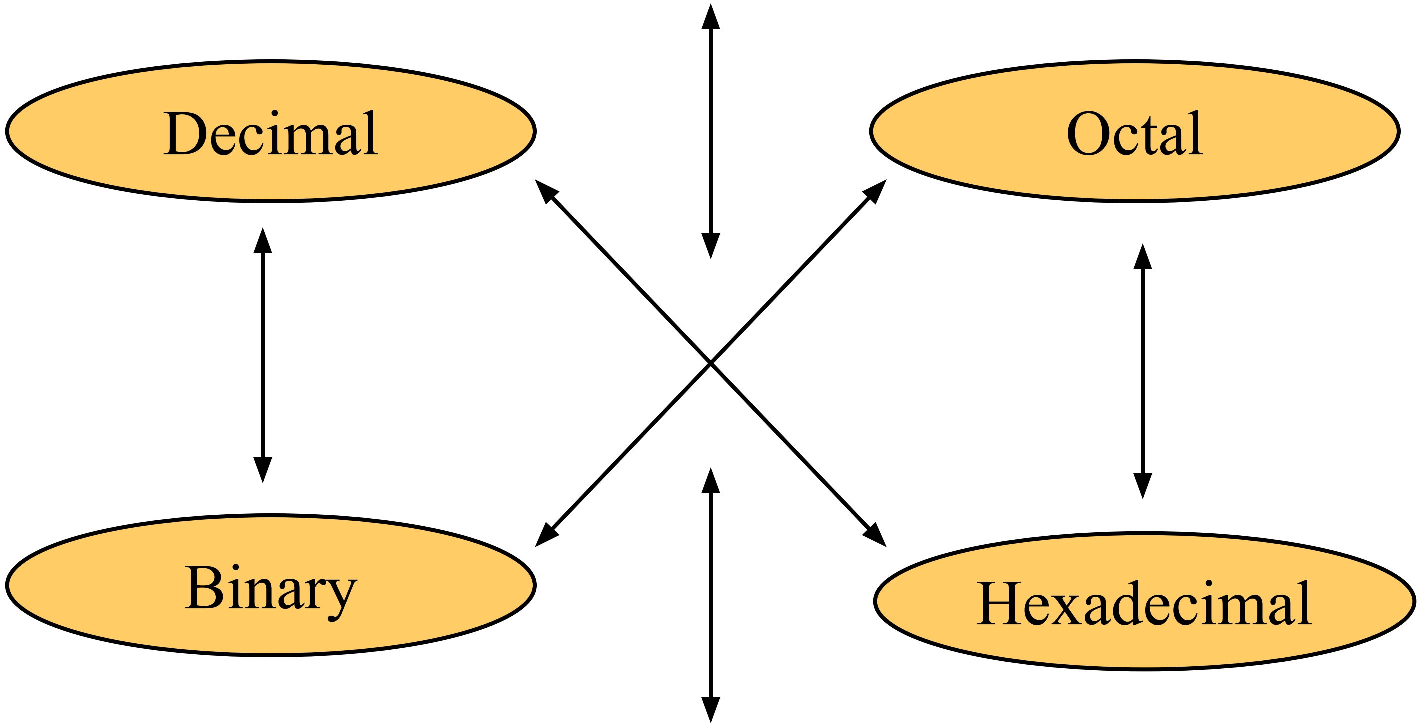
Quantities/Counting (3 of 3)

Decimal	Binary	Octal	Hexadecimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

Etc.

Conversion Among Bases

- The possibilities:



Base & Radix

- **Base**
  - 2
  - 8
  - 10
  - 16
- **Radix**

  Allowed number in a base
  - 2 (0, 1)
  - 8 (0, 1, 2, 3, 4, 5, 6, 7)
  - 10 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
  - 16 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)

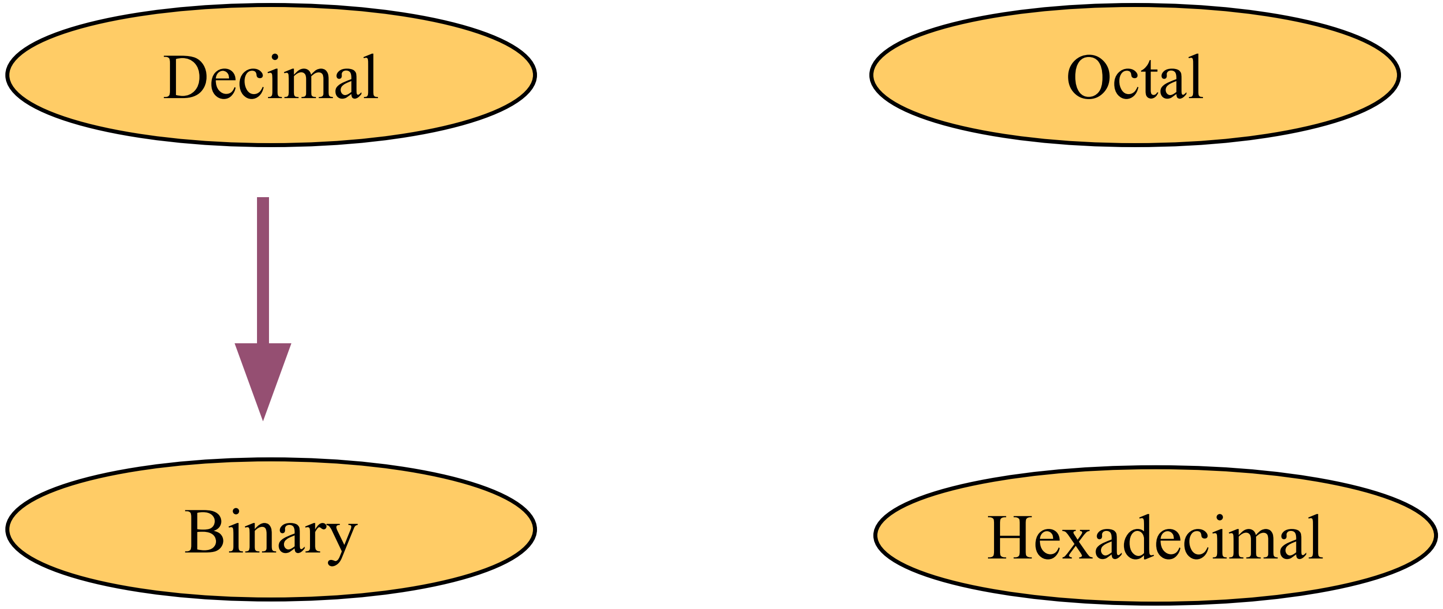
Quick Example

$$25_{10} = 11001_{2} = 31_{8} = 19_{16}$$

Base

From decimal to any base

Decimal to Binary

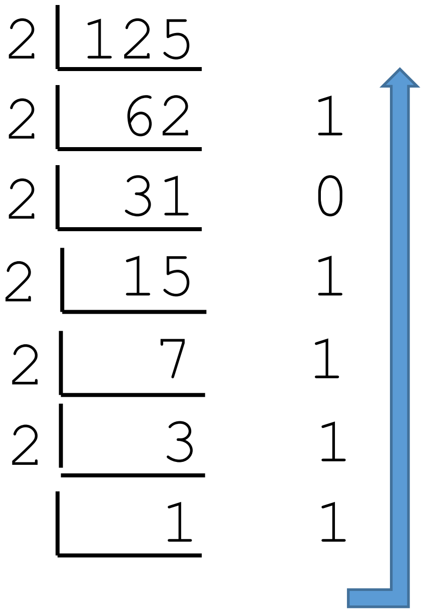


Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

Example

$125_{10} = ?_2$



2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
	1	1

$125_{10} = 1111101_2$

Decimal to Binary

Convert 100 from decimal to binary.

a) 1111100

b) 1000100

c) 1100101

d) 1100100

1100100

Exercise

Convert 555 from decimal to binary.

a) 1000101011

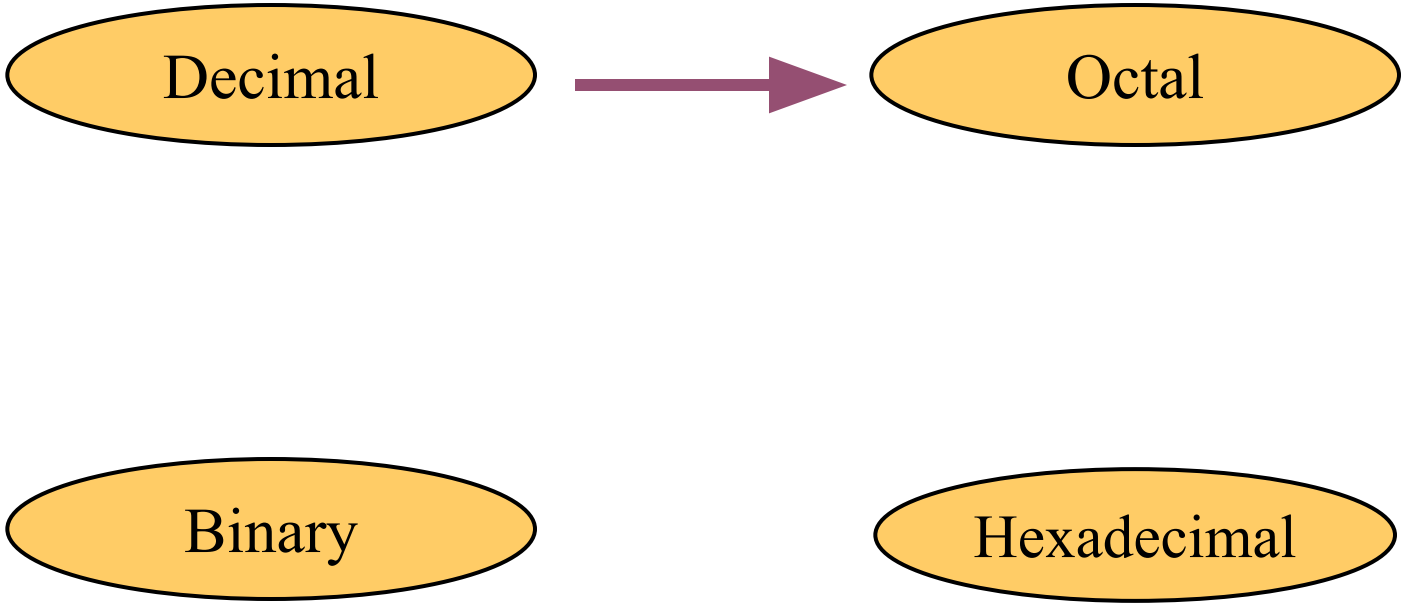
b) 1000111011

c) 1000101000

d) 1011101011

1000101011

Decimal to Octal

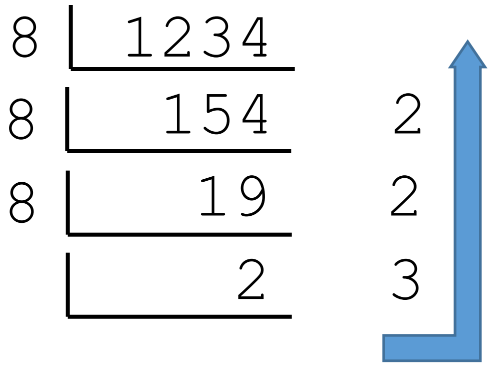


Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

Example

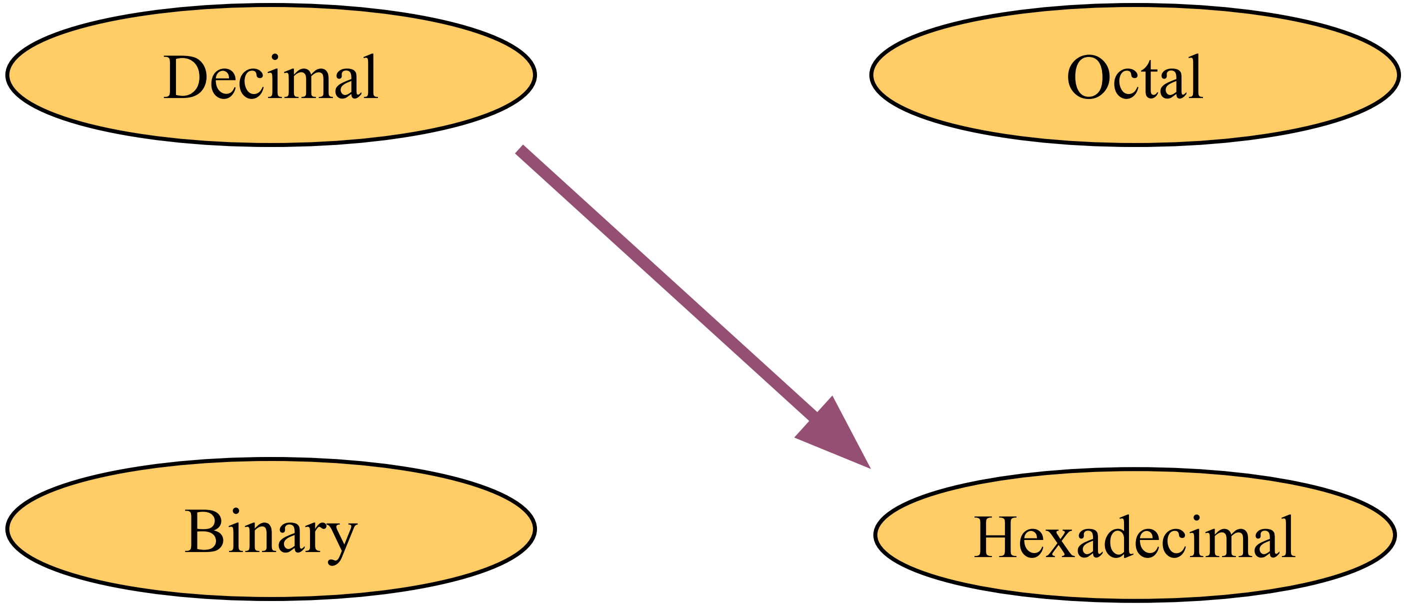
$1234_{10} = ?_8$



8	1234	
8	154	2
8	19	2
	2	3

$1234_{10} = 2322_8$

Decimal to Hexadecimal

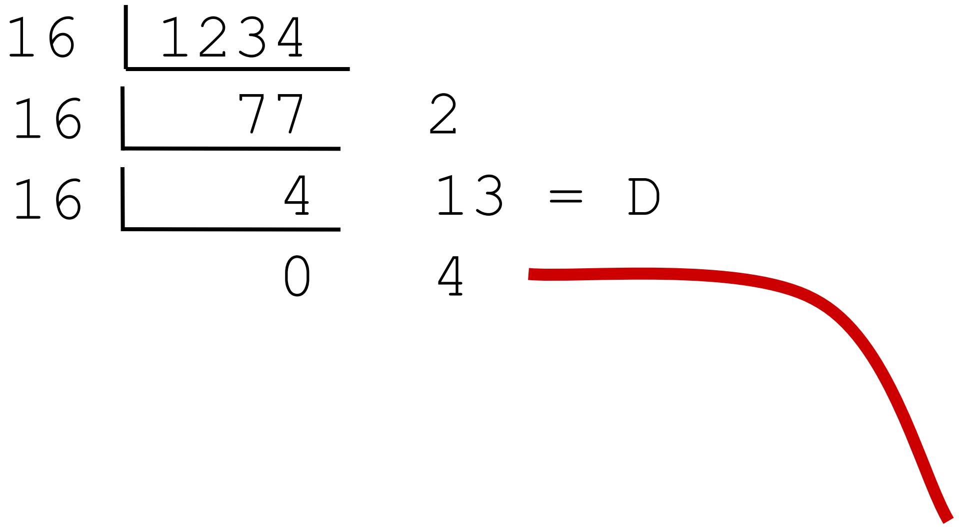


Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

Example

$1234_{10} = ?_{16}$

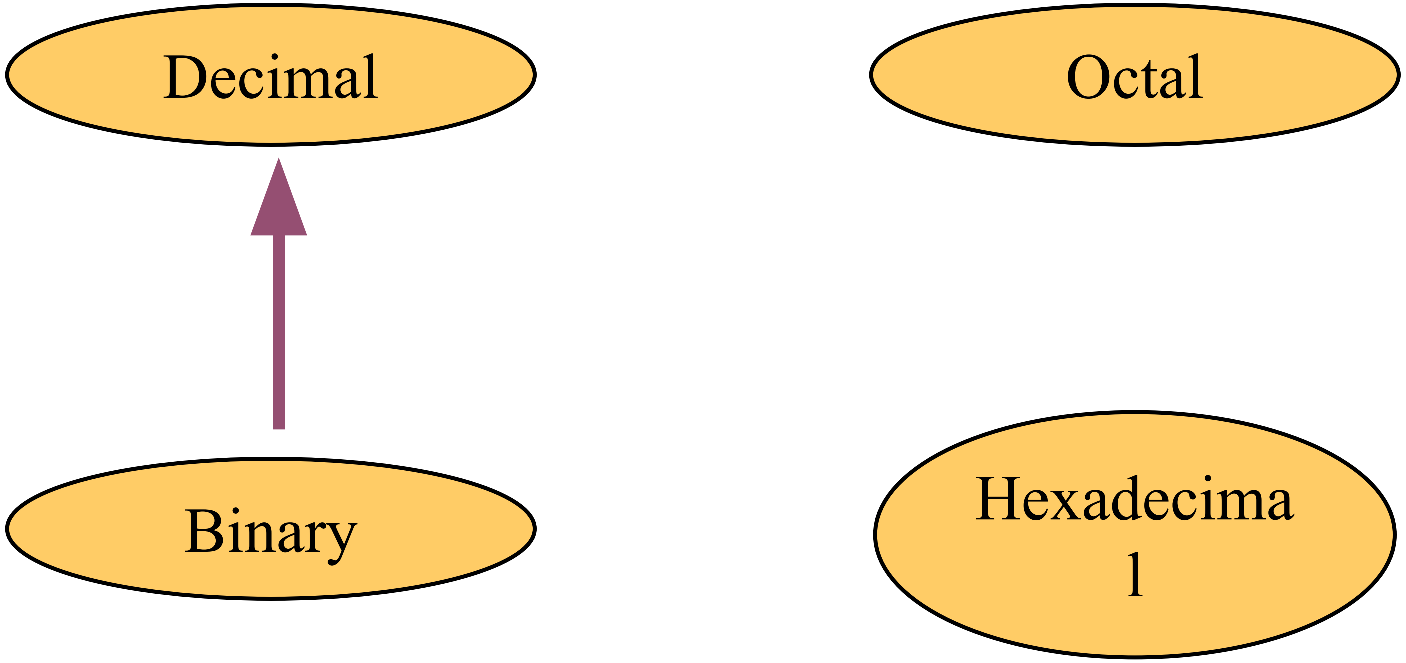


```
16 | 1234
16 |   77    2
16 |    4   13 = D
        0    4
```

$1234_{10} = 4D2_{16}$

From any base to decimal

Binary to Decimal



Binary to Decimal

- Technique
  - Multiply each bit by 2^{n}, where *n* is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

Bit “0”

101011_2 => $1 \times 2^0 = 1$

$1 \times 2^1 = 2$

$0 \times 2^2 = 0$

$1 \times 2^3 = 8$

$0 \times 2^4 = 0$

$1 \times 2^5 = 32$

Sum 32 + 0 + 8 + 0 + 2 + 1 = 43

101011_2 => 43_{10}

Exercise

Convert 10100011 from binary to decimal.

a) 121

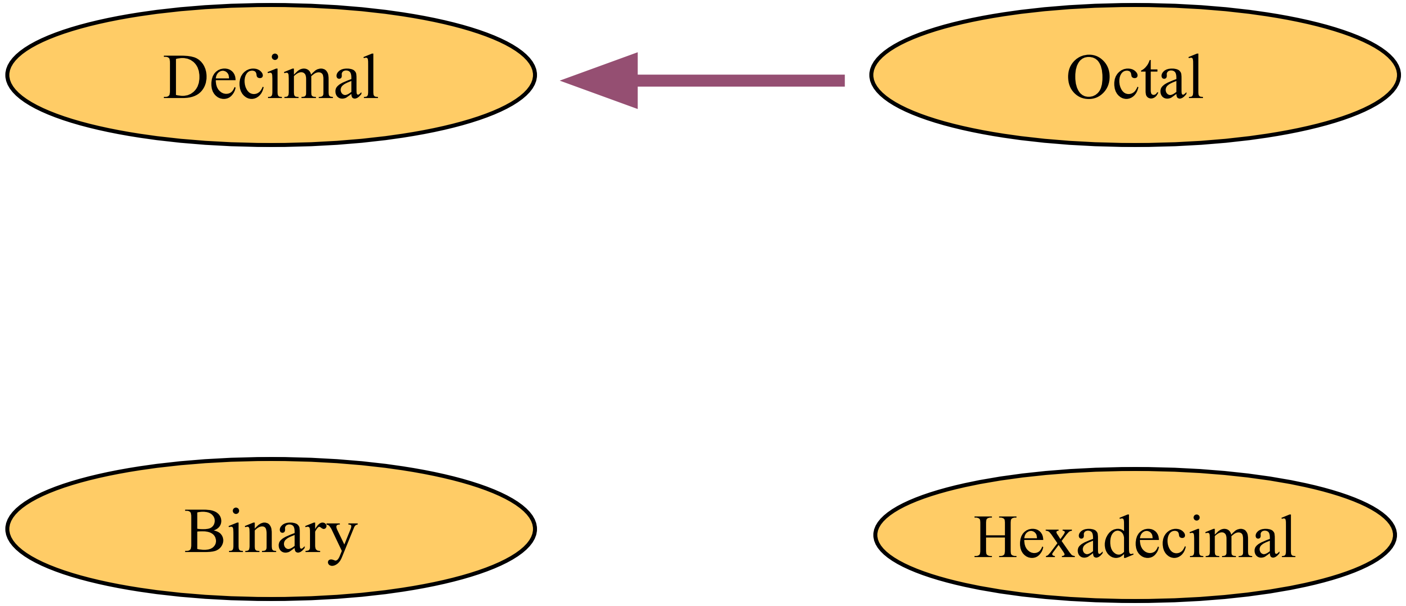
b) 163

c) 199

d) 212

163

Octal to Decimal



Octal to Decimal

- Technique
  - Multiply each bit by 8^n, where *n* is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

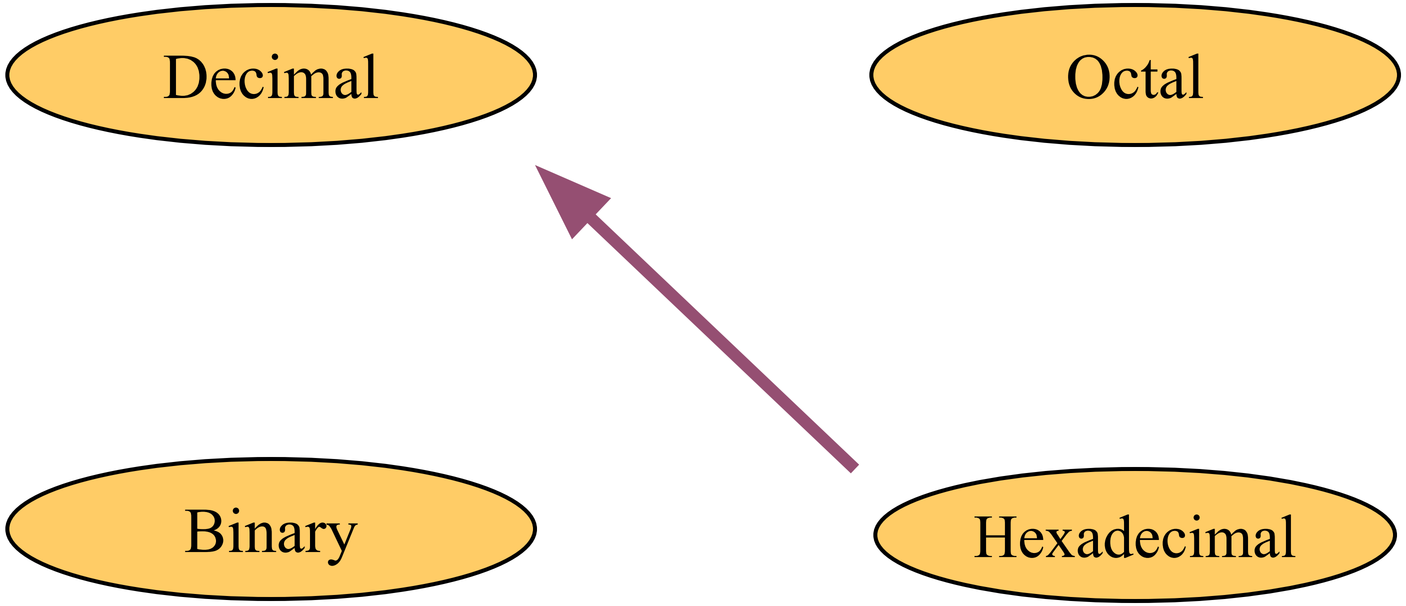
724_8 => $4 \text{ x } 8^0 = 4$

$2 \text{ x } 8^1 = 16$

$7 \text{ x } 8^2 = 448$

468_{10}

Hexadecimal to Decimal



Hexadecimal to Decimal

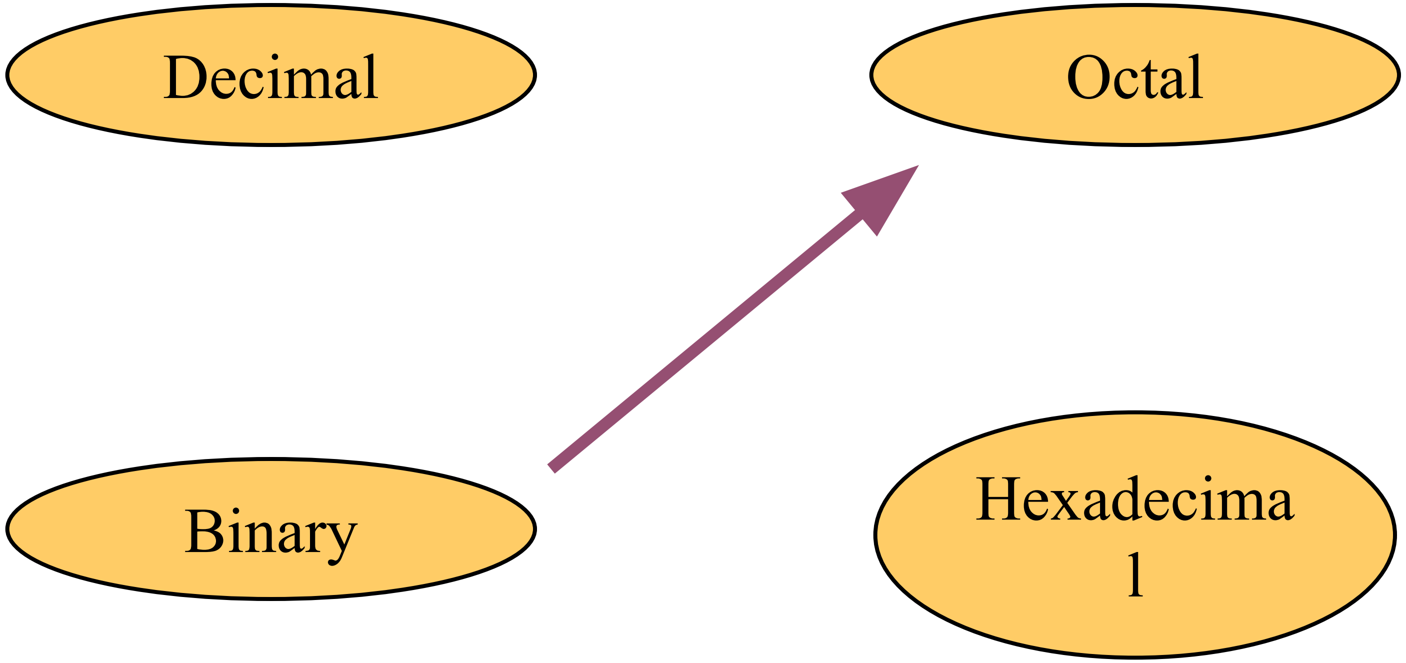
- Technique
  - Multiply each bit by 16^n, where *n* is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

$$\begin{array}{rr}
ABC_{16} \Rightarrow C \text{ x } 16^{0} = 12 \text{ x } 1 = & 12 \\
B \text{ x } 16^{1} = 11 \text{ x } 16 = & 176 \\
A \text{ x } 16^{2} = 10 \text{ x } 256 = & 2560 \\ \hline
& 2748_{10}
\end{array}$$

From binary to octal

Binary to Octal

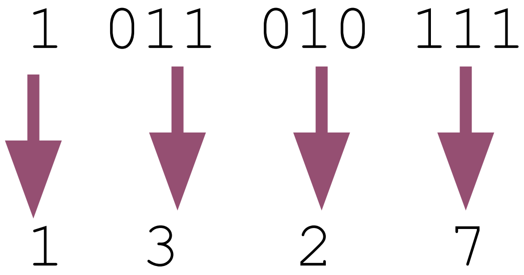


Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

Example

$1011010111_2 = ?_8$



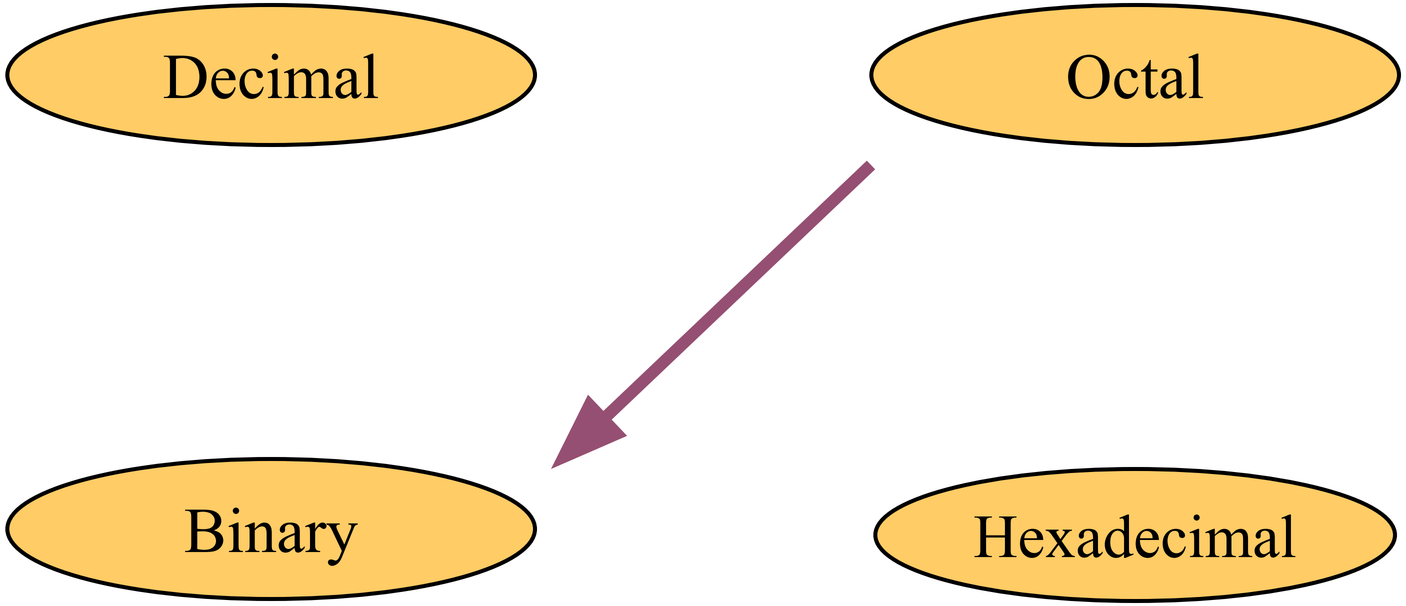
1 011 010 111

1 3 2 7

$1011010111_2 = 1327_8$

From octal to binary

Octal to Binary

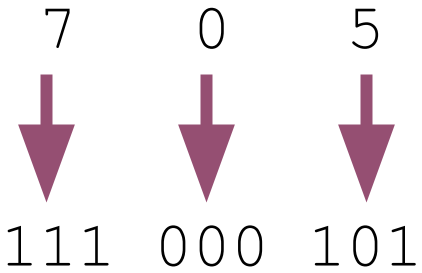


Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

Example

$705_8 = ?_2$



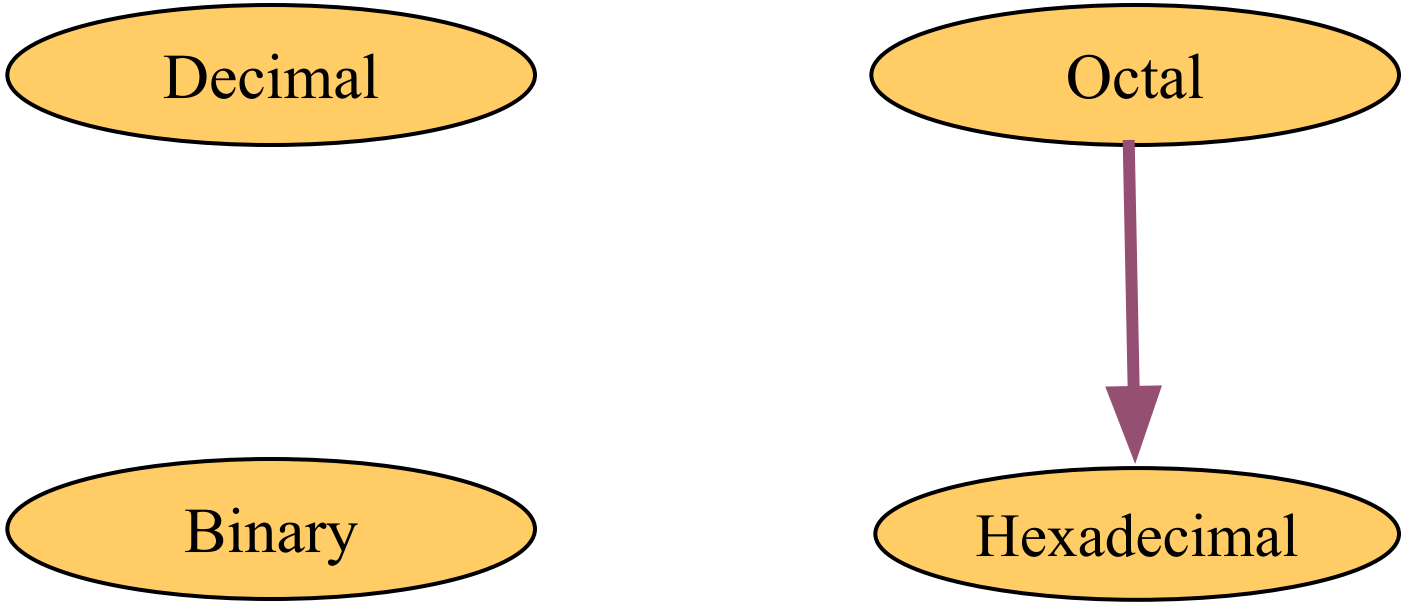
7 0 5

111 000 101

$705_8 = 111000101_2$

From octal to hexadecimal

Octal to Hexadecimal

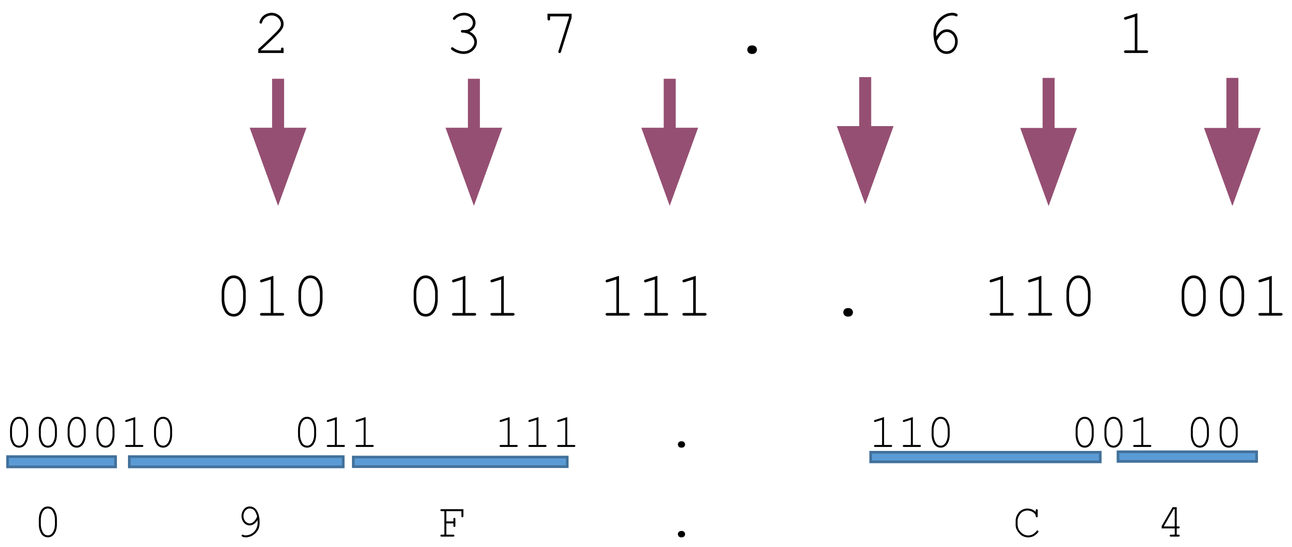


Octal to Hexadecimal

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation
  - Make pair of 4-bit and represent it in equivalent hexadecimal representation

Example

$237.61_8 = ?_{16}$



2 3 7 . 6 1

010 011 111 . 110 001

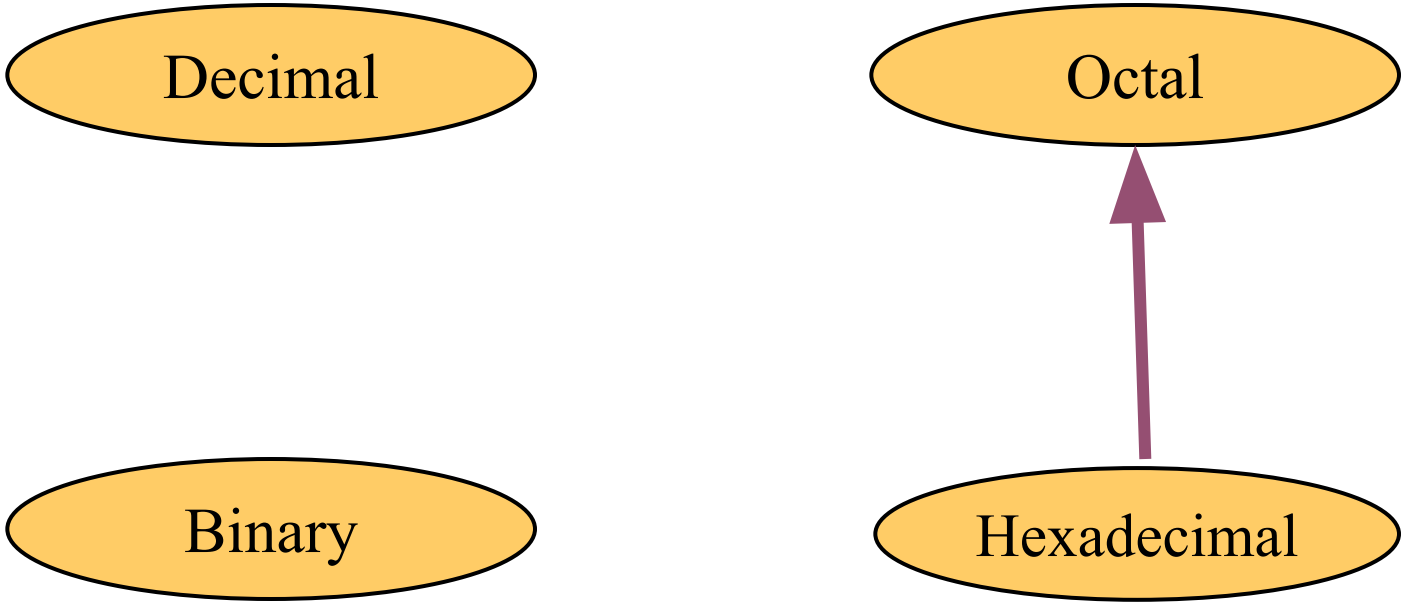
000010 011 111 . 110 001 00

0 9 F . C 4

$237.61_8 = 9F.C4_{16}$

From hexadecimal to octal

hexadecimal to octal



Hexadecimal to octal

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation
  - Make pair of 3-bit and represent it in equivalent octal representation

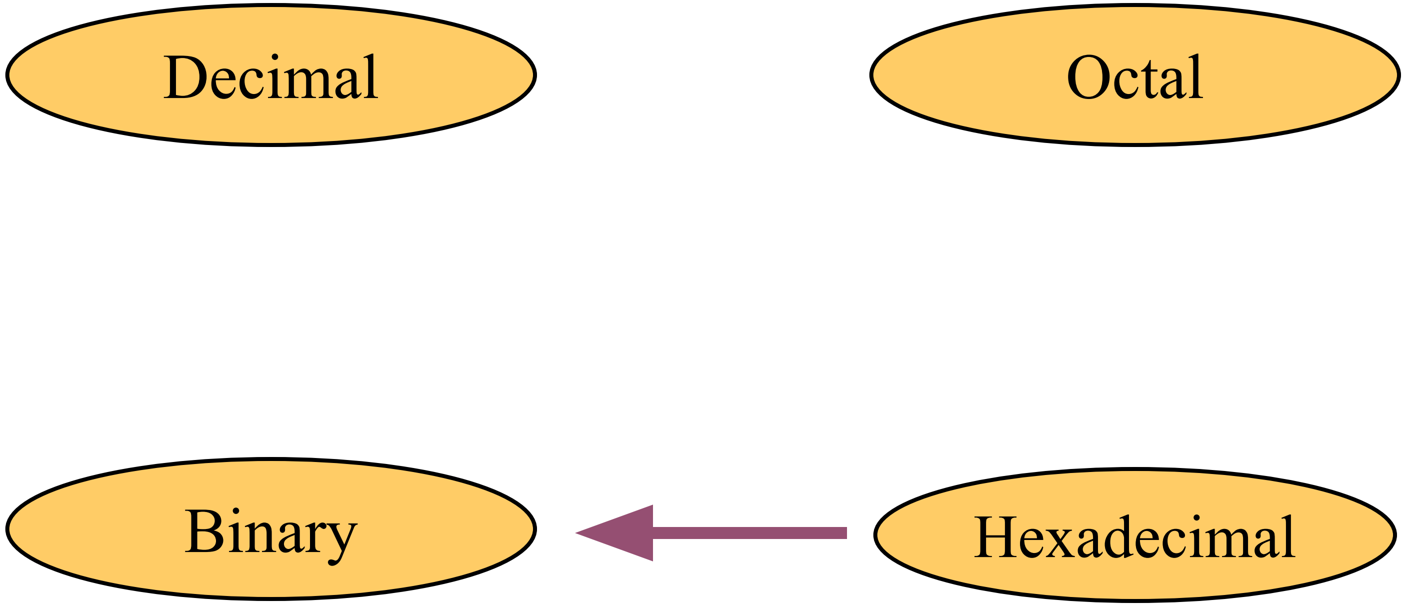
Example

$$9F.C4_{16} = (?)_{8}$$

????

From Hexadecimal to Binary

Hexadecimal to Binary

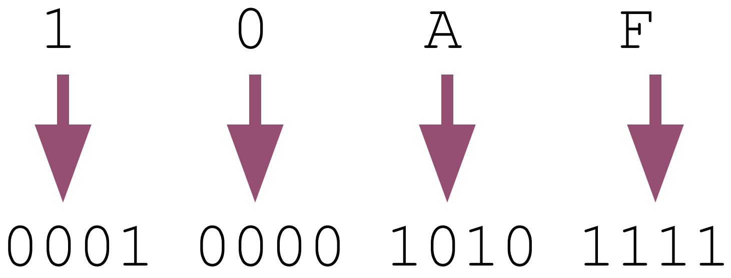


Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

Example

$10AF_{16} = ?_2$



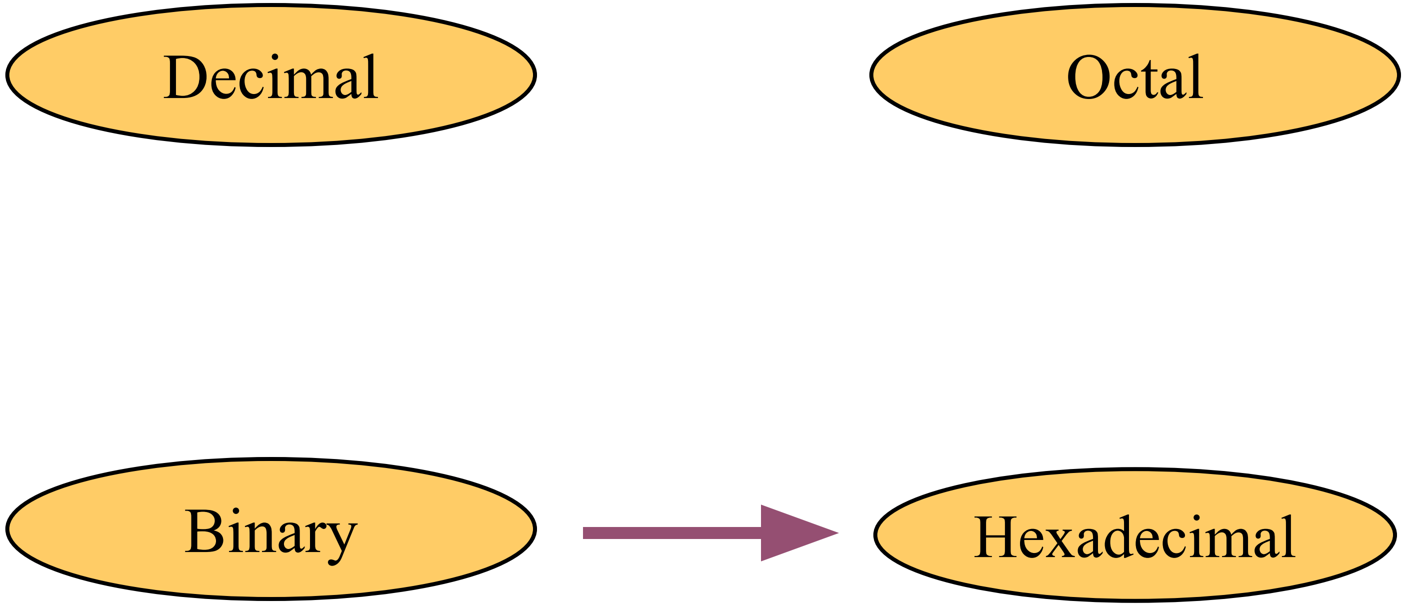
1 0 A F

0001 0000 1010 1111

$10AF_{16} = 0001000010101111_2$

From binary to hexadecimal

Binary to Hexadecimal

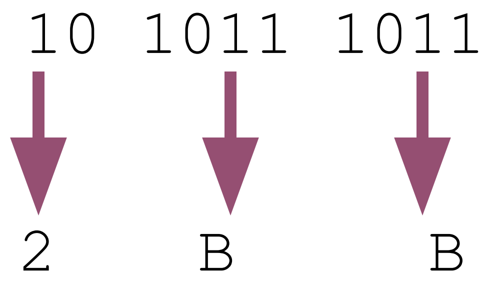


Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

Example

$1010111011_2 = ?_{16}$



10 1011 1011

2 B B

$1010111011_2 = 2BB_{16}$

Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Common Powers (1 of 2)

- Base 10

Power	Prefix	Symbol	Value
10^{-12}	pico	p	.000000000001
10^{-9}	nano	n	.000000001
10^{-6}	micro	µ	.000001
10^{-3}	milli	m	.001
10^{3}	kilo	k	1000
10^{6}	mega	M	1000000
10^{9}	giga	G	1000000000
10^{12}	tera	T	1000000000000

Common Powers (2 of 2)

- Base 2

Power	Prefix	Symbol	Value
2^{10}	kilo	k	1024
2^{20}	mega	M	1048576
2^{30}	Giga	G	1073741824

- In computing, particularly w.r.t. memory, the base-2 interpretation generally applies

Binary Addition (1 of 2)

- Two 1-bit values

A	B	A + B
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**10**

“two”

Binary Addition (2 of 2)

- Two *n*-bit values
  - Add individual bits
  - Propagate carries
  - E.g.,

```
 1   1
  10101        21
+ 11001      + 25
-------      ----
 101110        46
```

Overflow bit

Multiplication (1 of 3)

- Decimal (just for fun)

```
   35
x 105
-----
  175
 000
 35
-----
 3675
```

Multiplication (2 of 3)

- Binary, two 1-bit values

A	B	A × B
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Multiplication (3 of 3)

- Binary, two *n*-bit values
  - As with decimal values
  - E.g.,

```
    1110
  x 1011
--------
    1110
   1110
  0000
 1110
--------
10011010
```

Fractions

- Decimal to decimal (just for fun)

(10^{-1}= 1/10, 10^{-2}=1/100)

$$\begin{array}{lr}
3.14 \Rightarrow & 4 \text{ x } 10^{-2} = 0.04 \\
 & 1 \text{ x } 10^{-1} = 0.1 \\
 & 3 \text{ x } 10^{0} = 3 \\
\hline
 & 3.14
\end{array}$$

Fractions

- Binary to decimal

10.1011 => 1 x 2^{-4} = 0.0625

1 x 2^{-3} = 0.125

0 x 2^{-2} = 0.0

1 x 2^{-1} = 0.5

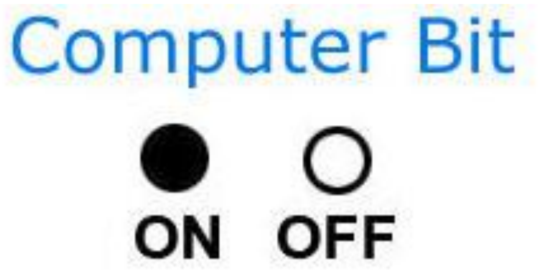
0 x 2^{0} = 0.0

1 x 2^{1} = 2.0

2.6875

Data Representation

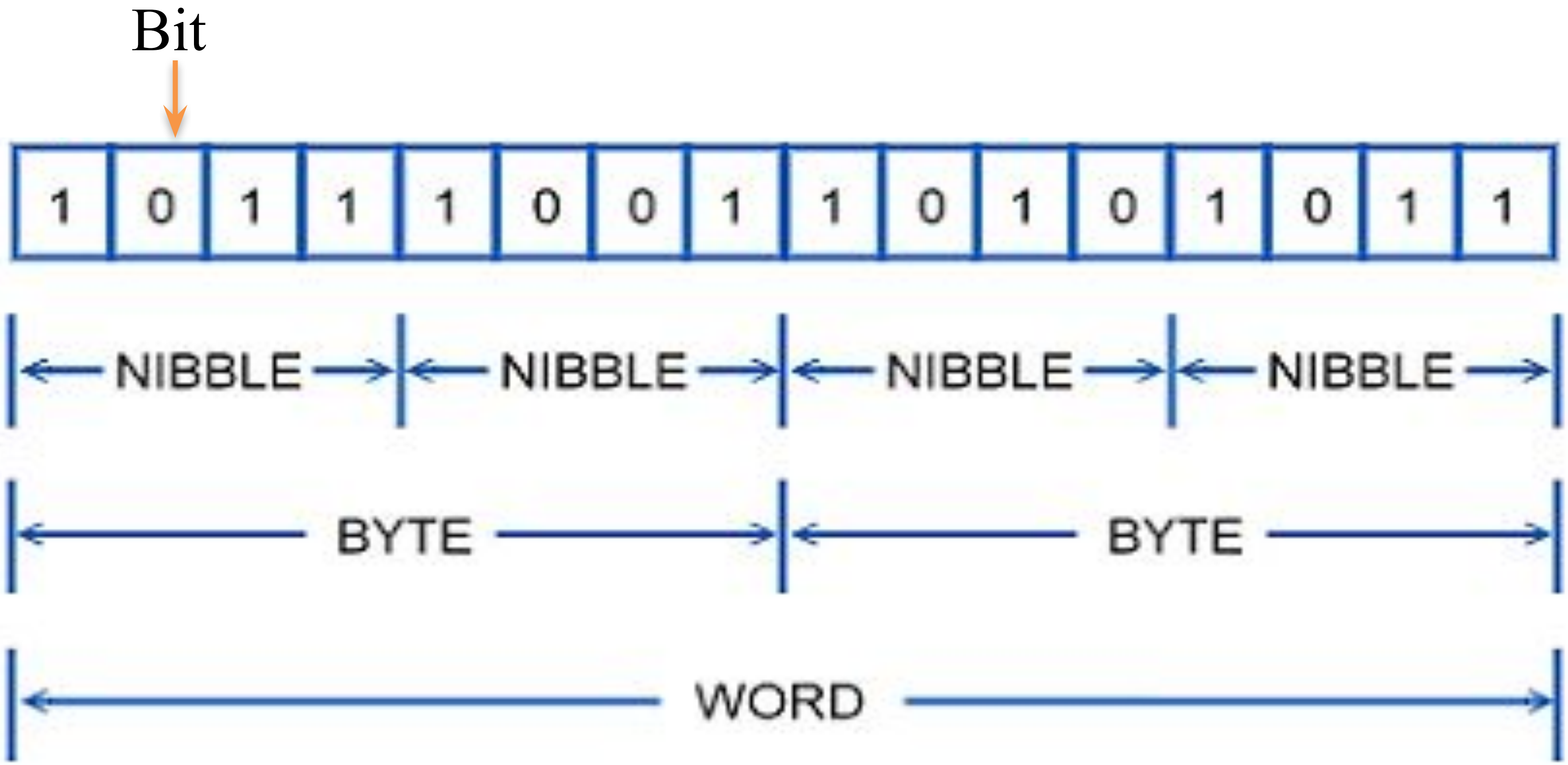
- Bits
  - The smallest unit of storage
  - 0 and 1 are called **bits** (short for binary digits)



1 0

- Bytes
  - Eight bits grouped together as a unit

The Whole Picture



Representation of Negative Numbers in Binary

- The **leftmost** digit of the number is used as a *special value* to represent the sign of the number:

  0 = positive, 1 = negative

- For example,

  Positive 12 (Decimal) = **0**1100

  Negative 12(Decimal) = **1**1100

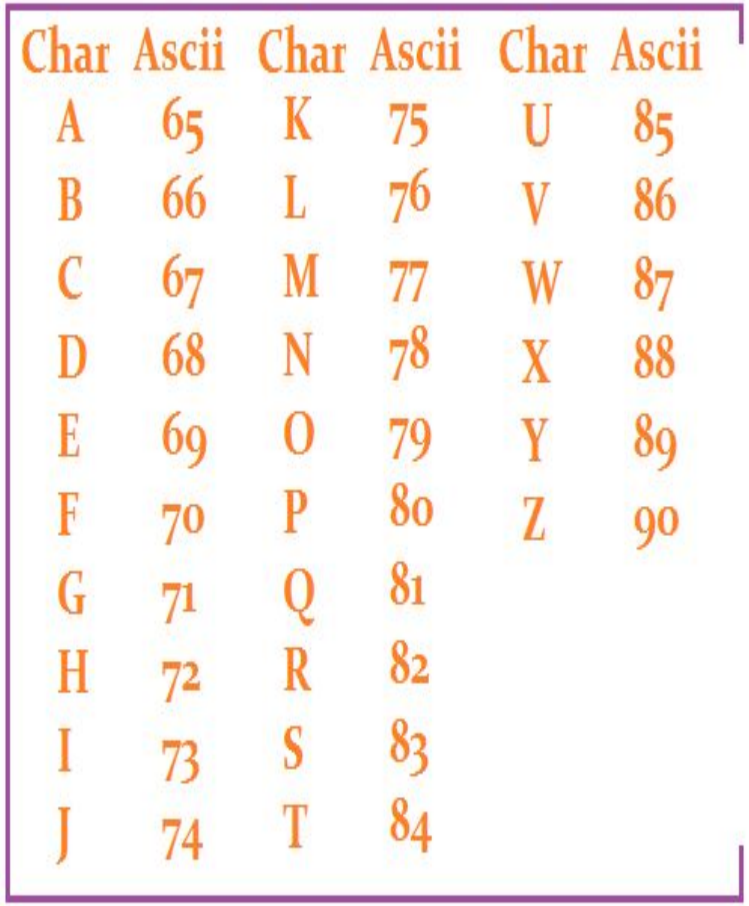
Computer Codes

Number of characters encoded in binary

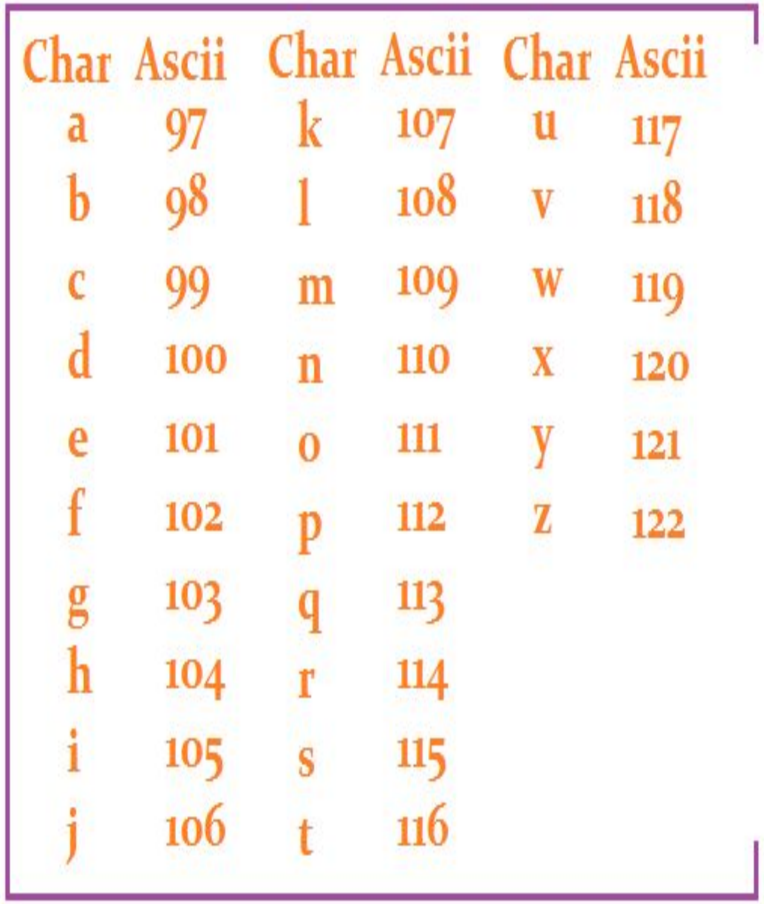
Computer Codes

- ASCII (7 bit coding system)
  - American Standard Code for Information Interchange
  - 7-bit system containing 128 characters
  - A **Character Encoding** standard for electronic communication
  - 26 for lower case
  - 26 for upper case and etc..

Contd..



Char	Ascii	Char	Ascii	Char	Ascii
A	65	K	75	U	85
B	66	L	76	V	86
C	67	M	77	W	87
D	68	N	78	X	88
E	69	O	79	Y	89
F	70	P	80	Z	90
G	71	Q	81		
H	72	R	82		
I	73	S	83		
J	74	T	84		

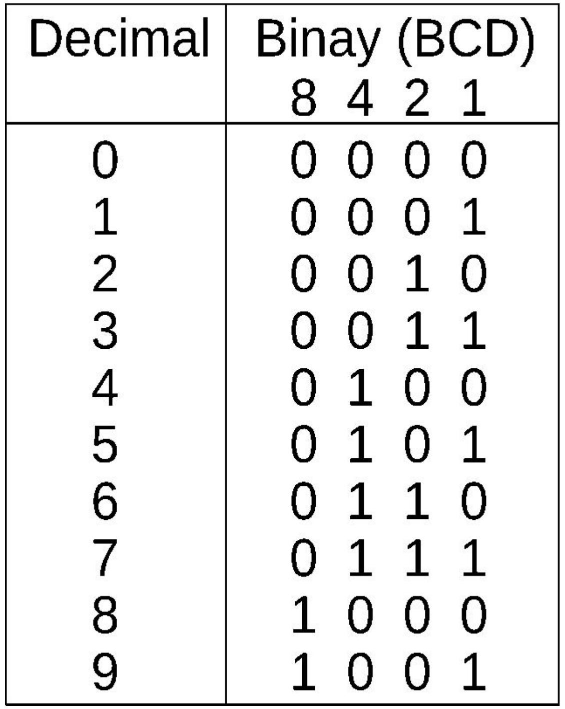


Char	Ascii	Char	Ascii	Char	Ascii
a	97	k	107	u	117
b	98	l	108	v	118
c	99	m	109	w	119
d	100	n	110	x	120
e	101	o	111	y	121
f	102	p	112	z	122
g	103	q	113		
h	104	r	114		
i	105	s	115		
j	106	t	116		

Control Characters				Graphic Symbols											
Name	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(	40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09	)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	−	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[	91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C	|	124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D	]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F

Computer Codes

- BCD
  - Binary Coded Decimal
  - Assigns a four-digit binary code to each digit



Decimal	Binay (BCD)
	8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

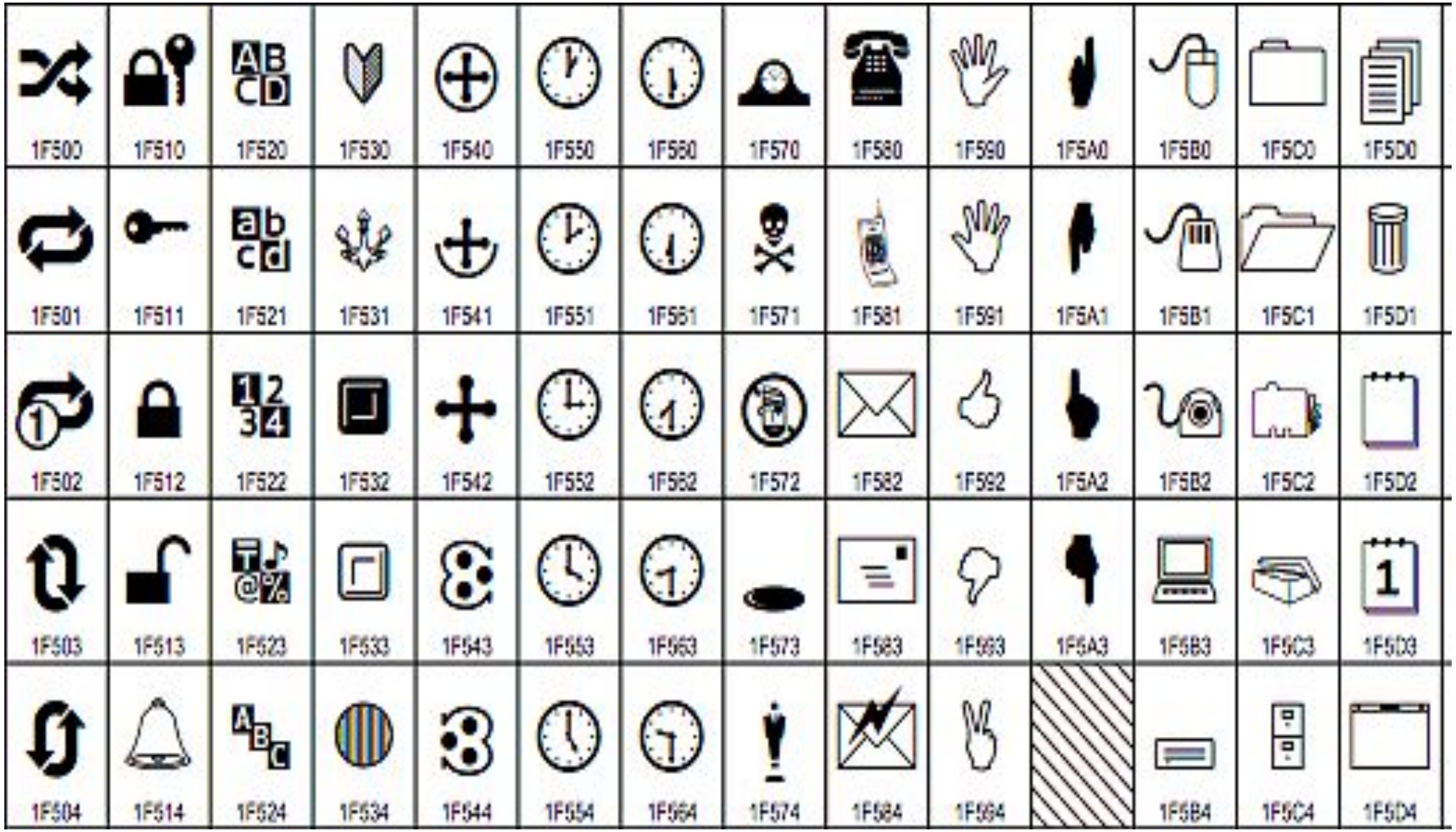
BCD Addition (not included in syllabus)

- 0-9 range is okay
- For greater than 9, then we have to add 6
- 2+6 = 8 what is in BCD?
- 8+3 = 11 what is in BCD? We have to add 6 (0110)

Computer Codes

- UNICODE (8-bit, 16-bit, 32-bit)
  - An international encoding standard for use with different languages and scripts
  - Each letter, digit, or symbol is assigned a *unique numeric value* that applies across different platforms and programs
  - Often, while reading about Unicode you will encounter acronyms such as **UCS-***, **UTF-***, and **BOM**
  - Java, Python

Contd..



🔀 1F500	🔐 1F510	🔠 1F520	🔰 1F530	🕀 1F540	🕐 1F550	🕠 1F560	🕰 1F570	🖀 1F580	🖐 1F590	🖠 1F5A0	🖰 1F5B0	🗀 1F5C0	🗐 1F5D0
🔁 1F501	🔑 1F511	🔡 1F521	🔱 1F531	🕁 1F541	🕑 1F551	🕡 1F561	🕱 1F571	🖁 1F581	🖑 1F591	🖡 1F5A1	🖱 1F5B1	🗁 1F5C1	🗑 1F5D1
🔂 1F502	🔒 1F512	🔢 1F522	🔲 1F532	🕂 1F542	🕒 1F552	🕢 1F562	🕲 1F572	🖂 1F582	🖒 1F592	🖢 1F5A2	🖲 1F5B2	🗂 1F5C2	🗒 1F5D2
🔃 1F503	🔓 1F513	🔣 1F523	🔳 1F533	🕃 1F543	🕓 1F553	🕣 1F563	🕳 1F573	🖃 1F583	🖓 1F593	🖣 1F5A3	🖳 1F5B3	🗃 1F5C3	🗓 1F5D3
🔄 1F504	🔔 1F514	🔤 1F524	🔴 1F534	🕄 1F544	🕔 1F554	🕤 1F564	🕴 1F574	🖄 1F584	🖔 1F594	▨	🖴 1F5B4	🗄 1F5C4	🗔 1F5D4

Thank you