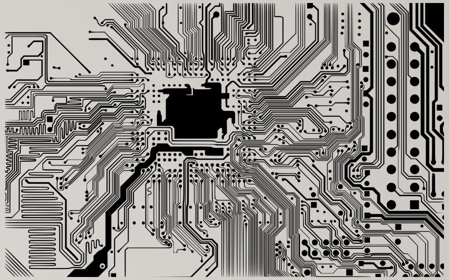
DATA REPRESENTATION
NUMBERING CONVERSION

DATA REPRESENTATION



Data Representation

Digital computers are **electronic devices** that contain a series of **circuits** and **voltage levels** that can store / represent data.

Binary numbers can represent those series of circuits with voltage levels. Those binary numbers (0's and 1's) are combined in a sequence to form a **byte**.

Bytes are used to represent **numbers** or **characters**.

It is the job of the computer program to understand if those bytes (series of o's and/or 1's) represent numbers or characters (eg. in **C programming**, declaring a variable with a **data type**)

Understanding how the computer stores numbers and characters can be useful when **administrating computer systems** and **creating programs** to be run on computer systems.

DEC.	BINARY								HEX.
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	2
3	0	0	0	0	0	0	1	1	3
4	0	0	0	0	0	1	0	0	4
5	0	0	0	0	0	1	0	1	5
6	0	0	0	0	0	1	1	0	6
7	0	0	0	0	0	1	1	1	7
8	0	0	0	0	1	0	0	0	8
9	0	0	0	0	1	0	0	1	9
10	0	0	0	0	1	0	1	0	A
11	0	0	0	0	1	0	1	1	B
12	0	0	0	0	1	1	0	0	C
13	0	0	0	0	1	1	0	1	D
14	0	0	0	0	1	1	1	0	E
15	0	0	0	0	1	1	1	1	F
16	0	0	0	1	0	0	0	0	10
17	0	0	0	1	0	0	0	1	11
				......					
				......					
				......					
253	1	1	1	1	1	1	0	1	FD
254	1	1	1	1	1	1	1	0	FE
255	1	1	1	1	1	1	1	1	FF

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

DATA REPRESENTATION

Numbering Conversion:

Computers have evolved over time. During that time, humans have interfaced with the computer by *binary* numbers, or by using **short-cuts** such as **<u>octal</u>** or **<u>hexadecimal</u>** numbers.

Computer Networking / Support Specialists and **Computer Programmers** occasionally need to convert between numbering systems:

- Converting **decimal** numbers to **binary** number for URLs (subnetting)
- Converting **decimal** numbers to **hexadecimal** numbers to format webpages (with web-safe colours)
- Converting **binary** numbers to **octal** numbers for setting file permissions in Unix/Linux

Before performing numbering conversions, we need to better understand the **decimal**, **binary**, **octal** and **hexadecimal** numbering systems.

DEC.	BINARY								HEX.
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1
2	0	0	0	0	0	0	1	0	2
3	0	0	0	0	0	0	1	1	3
4	0	0	0	0	0	1	0	0	4
5	0	0	0	0	0	1	0	1	5
6	0	0	0	0	0	1	1	0	6
7	0	0	0	0	0	1	1	1	7
8	0	0	0	0	1	0	0	0	8
9	0	0	0	0	1	0	0	1	9
10	0	0	0	0	1	0	1	0	A
11	0	0	0	0	1	0	1	1	B
12	0	0	0	0	1	1	0	0	C
13	0	0	0	0	1	1	0	1	D
14	0	0	0	0	1	1	1	0	E
15	0	0	0	0	1	1	1	1	F
16	0	0	0	1	0	0	0	0	10
17	0	0	0	1	0	0	0	1	11
	.....								
	.....								
	.....								
253	1	1	1	1	1	1	0	1	FD
254	1	1	1	1	1	1	1	0	FE
255	1	1	1	1	1	1	1	1	FF

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

DATA REPRESENTATION

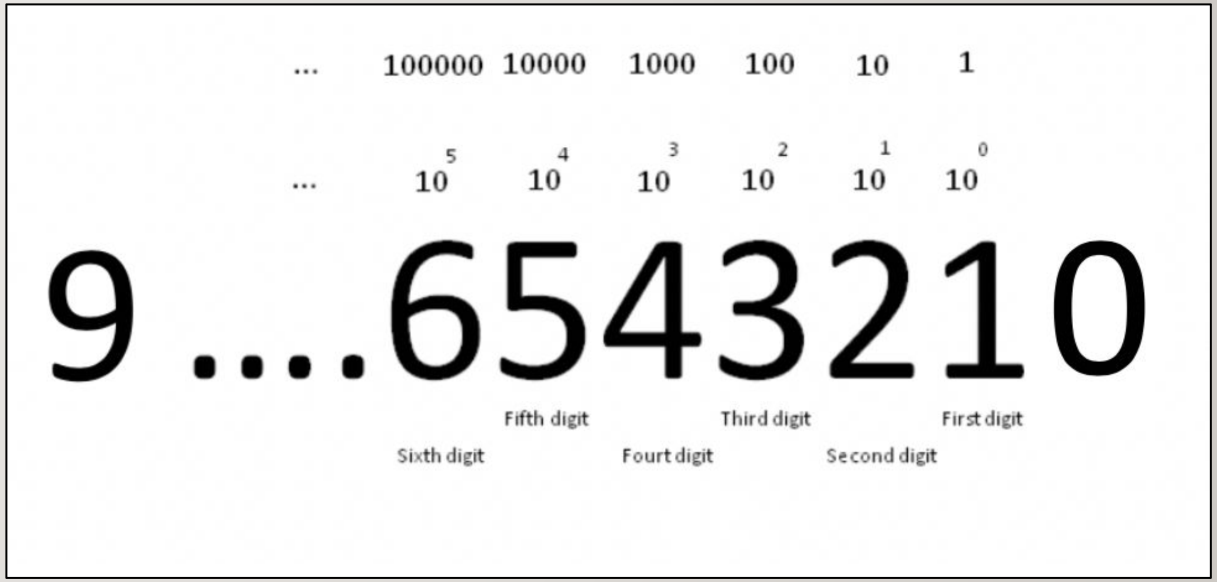
Decimal Numbering System (Humans)

The numbering system used by **humans**.

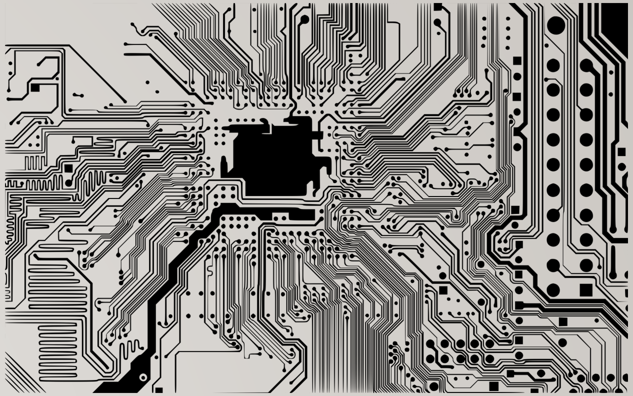
The **decimal** numbering system consists of **digits** ranging from **0** to **9**.

The fact that **humans** started counting on their **fingers** and **thumbs** most likely lead to the development of this numbering system.

The decimal numbering system is based on **sums of the power of 10** which provides a framework for mathematic calculations.



DATA REPRESENTATION

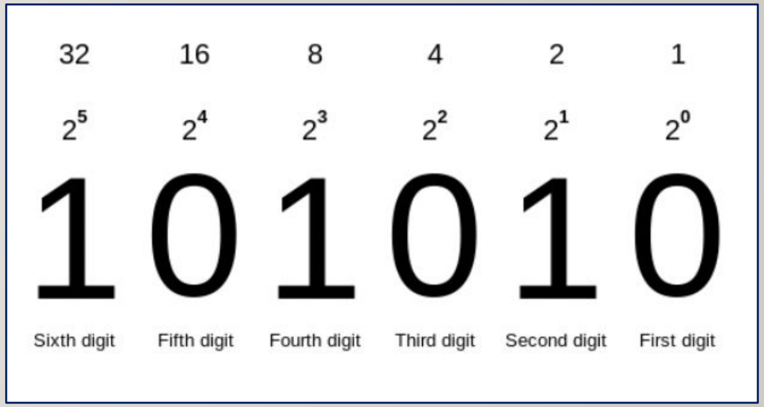


Binary Numbers (Computers)

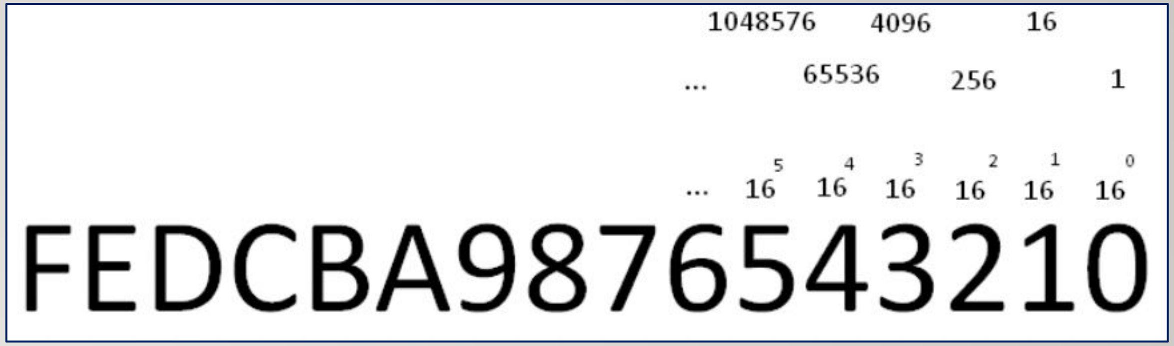
Digital computers have **circuits** which representing data in terms of voltage levels. Multiple circuits are used to represent data (in the form of *binary* numbers).

The **binary** numbering system consists of digits ranging from **0** to **1**. The numbering system is based on sums of the power of **2**.

Referring to the diagram to the right, the value of each decimal digit consists of the value (placeholder) multiplied by the corresponding power of 2. For example, $\mathbf{2^0}$, $\mathbf{2^1}$, $\mathbf{2^2}$, etc. which move in a **right-to-left** direction.



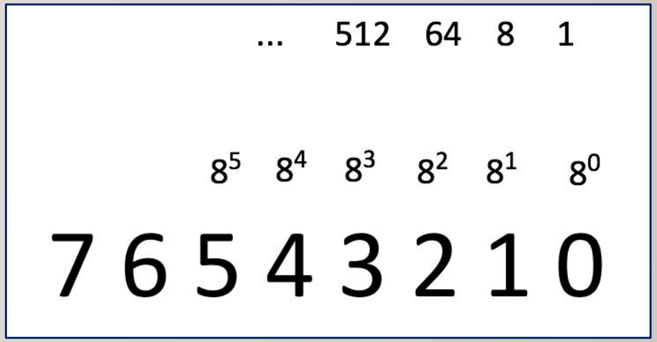
DATA REPRESENTATION



Octal / Hexadecimal Numbers (short-cuts)

The **octal** and **hexadecimal** numbering systems consist of digits ranging from **0 to 7** and ranging from **0 to F** respectively.

The **octal** and **hexadecimal** numbering system are based on sums of the power of **8** and **16** respectively. For *hexadecimal* numbers, values for **10 to 15** are represented by the characters **A to F** respectively.

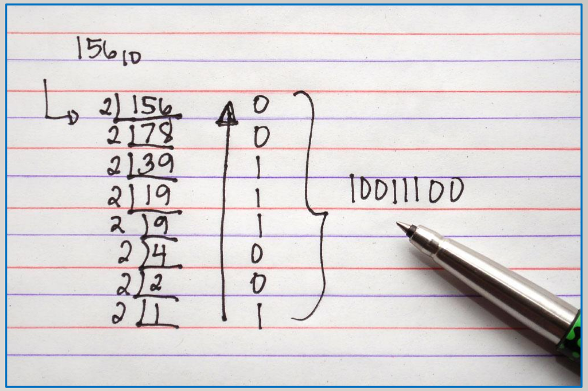


These numbering systems are useful since they are **both multiples of 2** (binary) and can be used as **short-cuts** to represent a series of binary numbers:

1 octal digit = 3 binary digits

1 hexadecimal digit = 4 binary digits).

DATA REPRESENTATION



Performing Numbering Conversion

You will learn **several numbering conversion methods** in this course:

1. **Binary** to **Decimal**
2. **Decimal** to **Binary**
3. **Octal** to **Binary** / **Binary** to **Octal**
4. **Hexadecimal** to **Binary** / **Binary** to **Hexadecimal**
5. **Octal** to **Hexadecimal** / **Hexadecimal** to **Octal**

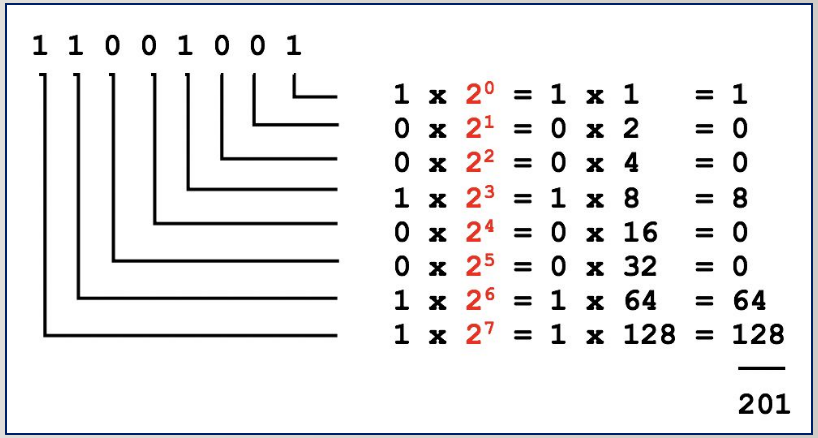
DATA REPRESENTATION

Numbering Conversion Method 1: Binary to Decimal

When converting **binary** numbers to **decimal** numbers, perform the following steps:

1. Write down the binary number.
2. Starting from the **right-side**, draw **L**'s below the binary number moving to the left (refer to diagram on right).
3. Starting on the *rightmost* "**L**", multiply the value (placeholder) by **2** to the power of zero.
4. Continually repeat **step #3** moving leftwards, increasing the power of *2* by **1** (refer to diagram on right).
5. Add up the results to obtain the decimal value equivalent.

NOTE: To convert *octal* and *hexadecimal* numbers to **decimal**, replace the number **2** (in red in the diagram to the right) with **8** (for *octal*) or **16** (for *hexadecimal*).

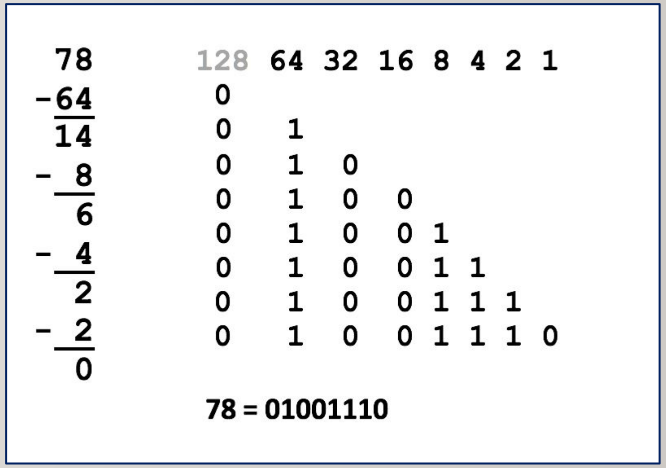


DATA REPRESENTATION

Numbering Conversion Method 2:
Decimal to Binary

When converting **decimal** numbers to **binary** numbers, perform the following steps:

1. Write down the **decimal number** to be converted.
2. On the *right-side*, write the number **1** and moving **leftwards**, keep doubling the numbers until that number is **greater than** the decimal number to be converted (refer to the diagram on the right).
3. Starting on the left-side of those doubled numbers, compare that number with the decimal number. If that number if less than or equal to the decimal number, then write a **1** below and subtract that number from the decimal number to get a remainder. If the number is greater than decimal number (or remainder), then write a **0** below.
4. Repeat **step #3** (moving rightwards and comparing the number with the decimal's remainder)



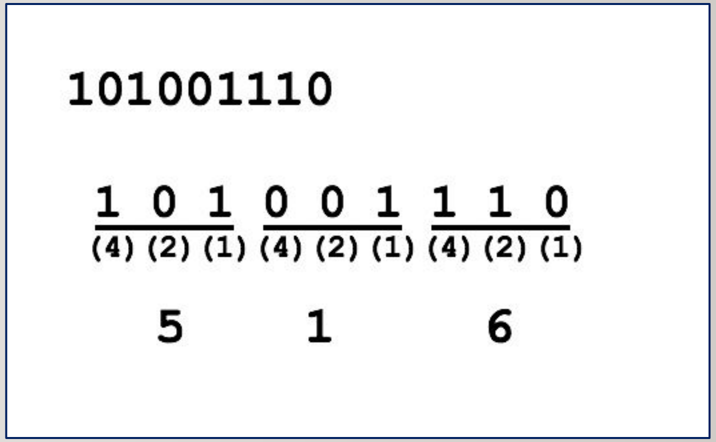
NOTE: If you are converting to **8-bit**, **32-bit**, etc., add **leading zeros** if necessary.

DATA REPRESENTATION

Numbering Conversion Method 3: Octal to Binary / Binary to Octal

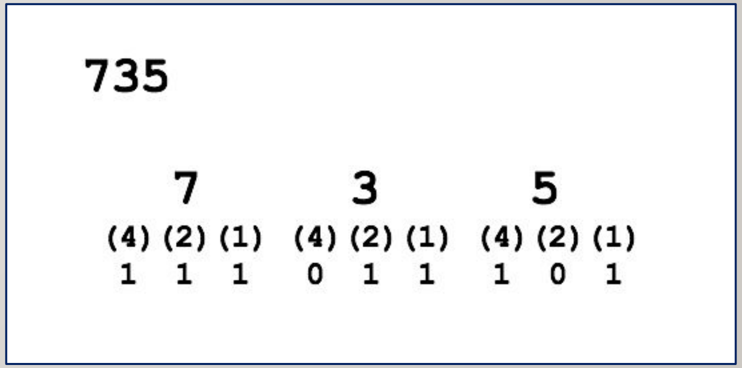
Binary to Octal

1. One octal number represents 3 binary numbers, so starting from right-side, group binary digits into groups of 3 (add leading zeros if necessary).
2. Write (4)(2)(1) under each group of 3 binary numbers.
3. Multiply the value or "placeholder" (i.e. 0's and 1's) by the corresponding (4)(2)(1) for each group to obtain the octal number (refer to diagram of binary to octal conversion).



Octal to Binary

4. One octal number represents 3 binary numbers, so space-out the octal numbers to make space for a binary number.
5. Write (4)(2)(1) under each octal number.
6. Write 0's or 1's for each group of binary numbers to add up to the corresponding octal number (refer to diagram of octal to binary conversion).

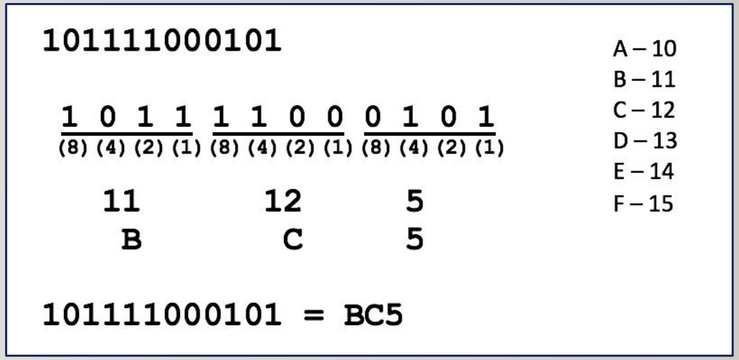


DATA REPRESENTATION

Numbering Conversion
Method 4: Hexadecimal to Binary / Binary to Hexadecimal

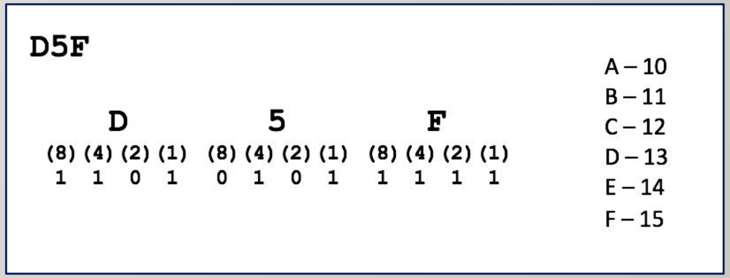
Binary to Hexadecimal

- **One hexadecimal number** represents **4 binary numbers**, so starting from right-side, group binary digits into **groups of 4** (add leading zeros if necessary).
- Write **(8)(4)(2)(1)** under <u>each</u> group of 4 binary numbers.
- Multiply the placeholders (i.e. **0'**s and **1'**s) by the corresponding (8)(4)(2)(1) for each group to obtain the octal number.
- Convert values from **10** to **15** to **A** to **F**
  (refer to diagram of *binary to hexadecimal* conversion)



Hexadecimal to Binary

- **One hexadecimal number** represents **4 binary numbers**,
  so space-out the hexadecimal numbers to make space for a binary number.
- Convert letters **A** to **F** to **10** to 15 (refer to diagram of *binary to hexadecimal* conversion)
- Write **(8)(4)(2)(1)** under <u>each</u> hexadecimal number.
- Write **0'**s or **1'**s for each group of binary numbers to add up to the corresponding hexadecimal number (refer to diagram of *hexadecimal to binary* conversion).



DATA REPRESENTATION

Numbering Conversion Method 5:
Octal to Hexadecimal / Hexadecimal to Octal

To convert using the method, simply use binary as a "**bridge**".

```
Octal -> binary -> Hexadecimal

Hexadecimal -> binary -> Octal
```

Example:

- To convert octal to hexadecimal, convert octal to binary, then convert binary to hexadecimal.
- To convert hexadecimal to octal, convert hexadecimal to binary, then convert binary to octal.

REPRESENTING NEGATIVE NUMBERS

Two Methods: 1. Sign Magnitude 2. Twos Complement

Example:

- convert -37 decimal to 8 bit signed magnitude
- 8 bits → bit values of s | 64 | 32 | 16 | 8 | 4 | 2 | 1
- 1 0 1 0 0 1 0 1
- convert 10010110 signed magnitude to decimal
- 8 bits → bit values of s | 64 | 32 | 16 | 8 | 4 | 2 | 1
- 10010110

• Signed Magnitude
• Maximum values: (non fractional) • 4 bits (s111) = +-7
• 8 bits (s111 1111) = +-127
• 16 bits (s111 1111 1111 1111) = +-32,767

TWOS COMPLEMENT

Twos Complement

Example:

- Find the two’s complement of 17
- Step 1: 1710 = 0001 00012
- Step 2: Take the complement: 1110 1110
- Step 3: Add 1: 1110 1110 + 1 = 1110 1111.

Subtract Binary Using Two's Complement

8 Bit Twos Complement

Example: 23-17

- 23=0001 0111
- -17=0001 0001→complement→1110 1110→add 1→1110 1111
- Add both binary numbers
- 0001 0111
- <u>1110 1111</u>
- 1 **000 0110 →6**

HOMEWORK

1. List the number of digits for the following numbering systems:
   - **Decimal: 10 digits (0–9)**
   - **Binary: 2 digits (0, 1)**
   - **Octal: 8 digits (0–7)**
   - **Hexadecimal: 16 digits (0–9, A–F)**

2. Write a simple chart to show which values are represented for letter **A - F** for a hexadecimal number.

Hex	Decimal
A	10
B	11
C	12
D	13
E	14
F	15

3. How many **binary** digits does 1 octal digit represent?

3 binary digits

4. How many **binary** digits does 1 hexadecimal digit represent?

4 binary digits

5. Use **manual numbering conversion** to complete the table displayed to the right.

Decimal	Binary	Octal	Hexadecimal
101	01100101	145	65
243	11110011	363	F3
56	00111000	56	38
172	10101100	254	AC

Complete the following (answers on the following page)

$\begin{array}{r} 1100011_2 \\ -\ 100100_2 \\ \hline \end{array}$	$\begin{array}{r} 101101_2 \\ +\ 110010_2 \\ \hline \end{array}$	$\begin{array}{r} 101111_2 \\ +\ 101011_2 \\ \hline \end{array}$
111111	1011111	1011010
$\begin{array}{r} 1011001_2 \\ -\ 101011_2 \\ \hline \end{array}$	$\begin{array}{r} 1100110_2 \\ -\ 110110_2 \\ \hline \end{array}$	$\begin{array}{r} 101010_2 \\ +\ 100001_2 \\ \hline \end{array}$
101110	110000	1001011

Solutions

$$\begin{array}{r} 1100011_2 \\ -\ 100100_2 \\ \hline 111111_2 \end{array} \qquad \begin{array}{r} 101101_2 \\ +\ 110010_2 \\ \hline 1011111_2 \end{array} \qquad \begin{array}{r} 101111_2 \\ +\ 101011_2 \\ \hline 1011010_2 \end{array}$$

$$\begin{array}{r} 1011001_2 \\ -\ 101011_2 \\ \hline 101110_2 \end{array} \qquad \begin{array}{r} 1100110_2 \\ -\ 110110_2 \\ \hline 110000_2 \end{array} \qquad \begin{array}{r} 101010_2 \\ +\ 100001_2 \\ \hline 1001011_2 \end{array}$$

Hexidecimal Addition and Subtraction

Complete the following (carry overs are shown as hints)

```
 ¹6 ¹B E
+ E  5 7
```

1515

```
   1
  5A
+ BF
```

119

```
  1  1
  A  B C
+2  A 9
```

D 6 5

```
  1  1
  A  B C
+2  A 9
  D  6 5
```

```
 A 5 7
-1 2 5
```

9 3 2

```
    27
 A ~~B~~ 21
 ~~B~~ ~~C~~ ~~5~~
-1  D  A
```

9 E B

Extension:

```
6 1 6
  A 1 9
  x   B
```

6 F 1 3

```
 1 1
 6 6
A B C
x  2 9
```

1 B 8 1 C

Solutions

$$\begin{array}{rrrr} & {}^{1}6 & {}^{1}B & E \\ + & E & 5 & 7 \\ \hline = 1 & 5 & 1 & 5 \end{array}$$

$$\begin{array}{rrr} & 1 & \\ & 5 & A \\ + & B & F \\ \hline 1 & 1 & 9 \end{array}$$

$$\begin{array}{rrrr} & 1 & 1 & \\ & A & B & C \\ + & 2 & A & 9 \\ \hline & D & 6 & 5 \end{array}$$

$$\begin{array}{rrrr} & A & 5 & 7 \\ - & 1 & 2 & 5 \\ \hline & 9 & 3 & 2 \end{array}$$

$$\begin{array}{rrrr} & & 27 & \\ & A & \cancel{B} & 21 \\ & \cancel{B} & \cancel{C} & \cancel{5} \\ - & 1 & D & A \\ \hline & 9 & E & B \end{array}$$

$$\begin{array}{rrrr} 6 & 1 & 6 & \\ & A & 1 & 9 \\ & \times & & B \\ \hline 6 & F & 1 & 3 \end{array}$$

$$\begin{array}{rrrrr} & & 1 & 1 & \\ & & 6 & 6 & \\ & & A & B & C \\ & & \times & 2 & 9 \\ \hline & 6 & 0 & 9 & C \\ 1 & 5 & 7 & 8 & \\ \hline 1 & B & 8 & 1 & C \end{array}$$

8 Bit Negative Numbers

Convert the following numbers to binary using the sign magnitude method

-7 → 1 0000111
-15 → 1 0001111
-36 → 1 0100100
-67 → 1 1000011

Perform the following math using the Twos Complement method

6-5 00000110 + 11111011 = 00000001 → 1
18-6 00010010 + 11111010 = 00001100 → 12
26-17 00011010 + 11101111 = 00001001 → 9
42-33 00101010 + 11011111 = 00001001 → 9