COMPUTER
ARCHITECTURE

CS 10 Computer Architecture and Organization
Number Systems and Data Formats

Foothill College
Computer Science Department

Number Systems

Module Objectives

- Determine the weighting factor of each digit position in the decimal, binary, octal, and hexadecimal numbering systems
- Convert any number among the four number systems, and its equivalent value in any of the remaining three numbering systems
- Describe binary coded decimal (BCD) numbers
- Translate alphanumeric data to and from ASCII using the ASCII code translation table

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	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

Counting 1/3

Decimal	Binary	Octal	Hexa-decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Counting 2/3

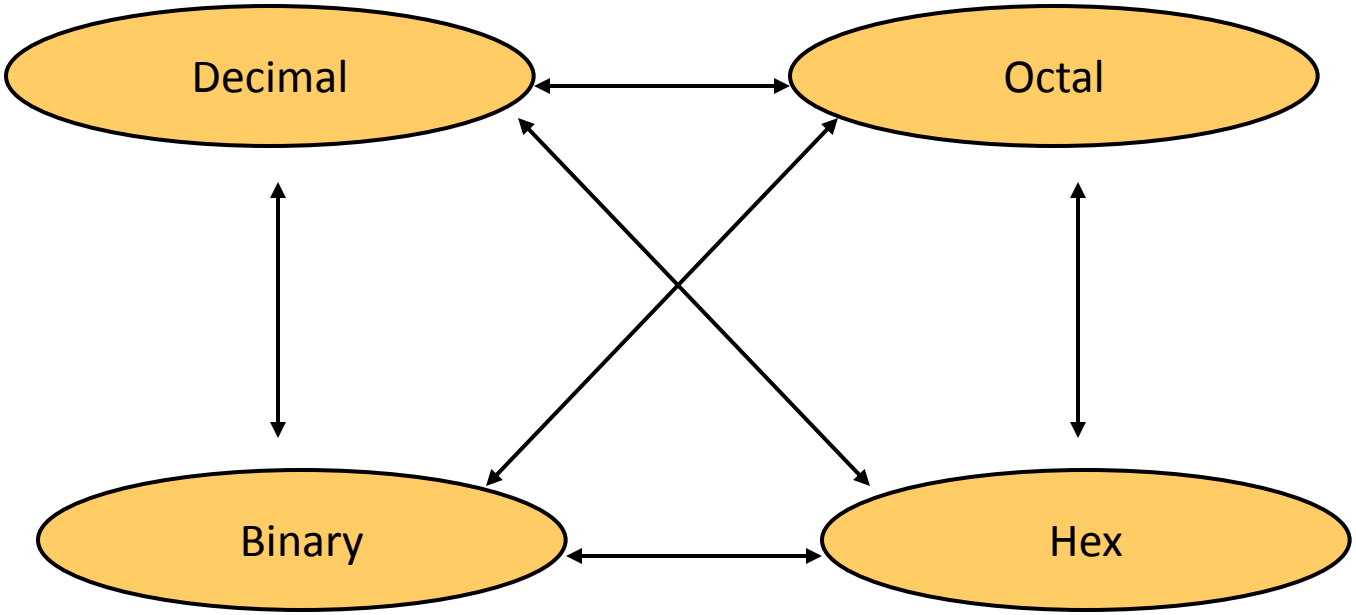
Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Counting 3/3

Decimal	Binary	Octal	Hexa-decimal
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

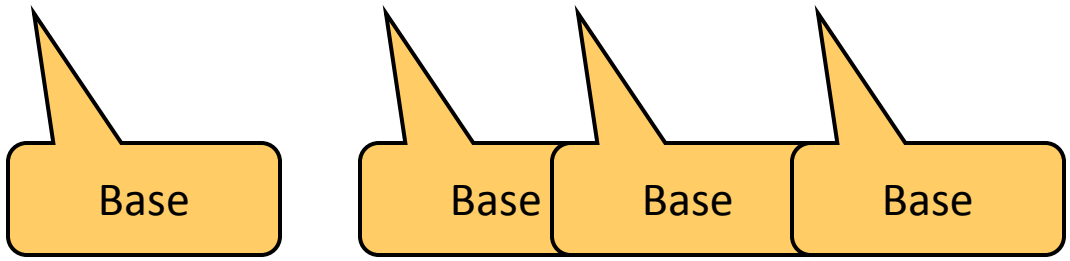
Conversion Among Bases

- The possibilities

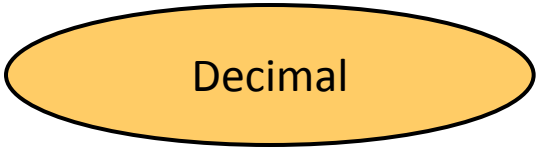


Base Conversion: Example

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

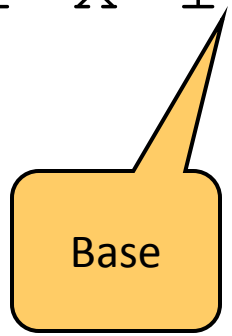


Decimal

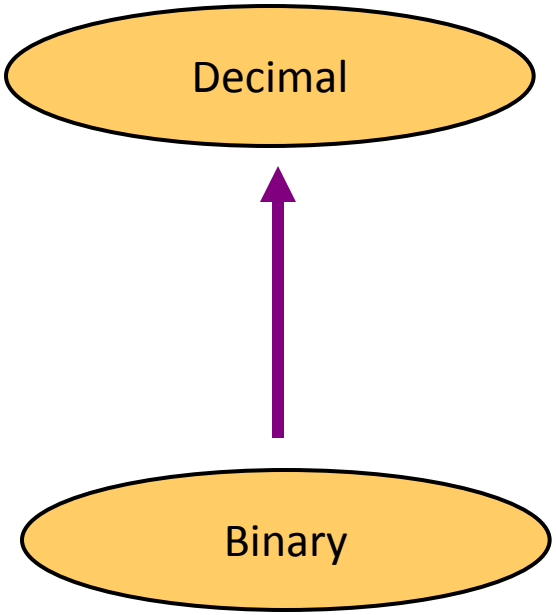


- 125_{10} =>

$$
\begin{aligned}
5 \times 10^{0} &= 5 \\
2 \times 10^{1} &= 20 \\
1 \times 10^{2} &= 100 \\
&\hline \\
&125
\end{aligned}
$$



Binary to Decimal Conversion



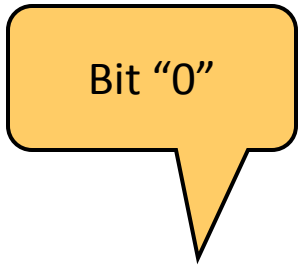
Technique:

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

Binary to Decimal Conversion: Example



$$101011_2 \Rightarrow \begin{array}{rcrcr} 1 & x & 2^0 & = & 1 \\ 1 & x & 2^1 & = & 2 \\ 0 & x & 2^2 & = & 0 \\ 1 & x & 2^3 & = & 8 \\ 0 & x & 2^4 & = & 0 \\ 1 & x & 2^5 & = & \underline{32} \\ & & & & 43_{10} \end{array}$$

Octal to Decimal Conversion

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

Octal to Decimal Conversion: Example

724_8 =>

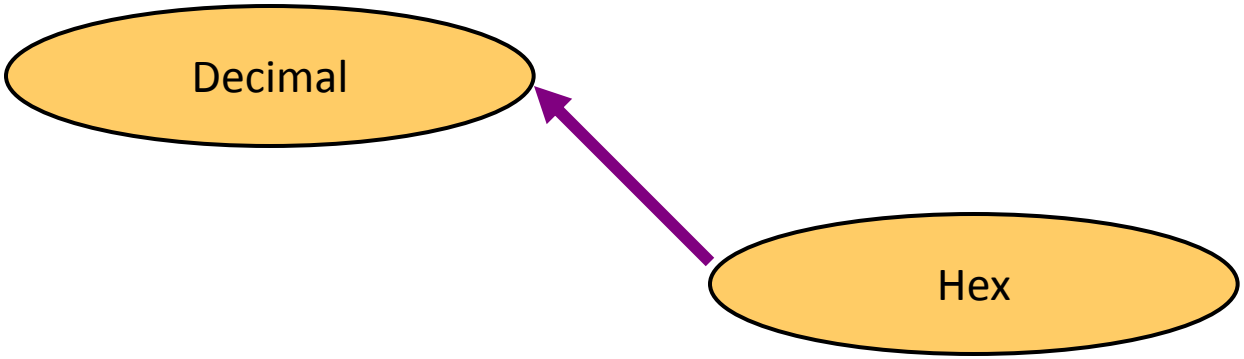
$4 \times 8^0 = 4$

$2 \times 8^1 = 16$

$7 \times 8^2 = 448$

468_{10}

Hex to Decimal Conversion



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

Hex to Decimal Conversion: Example

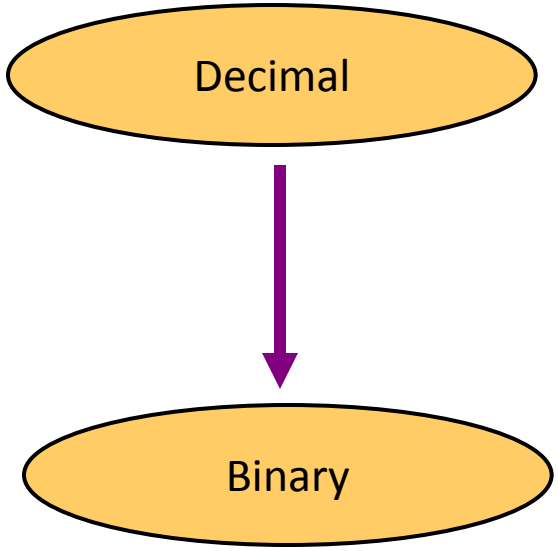
ABC_{16} =>C x 16^0 = 12 x 1 = 12

B x 16^1 = 11 x 16 = 176

A x 16^2 = 10 x 256 = 2560

2748_{10}

Decimal to Binary Conversion



Technique:

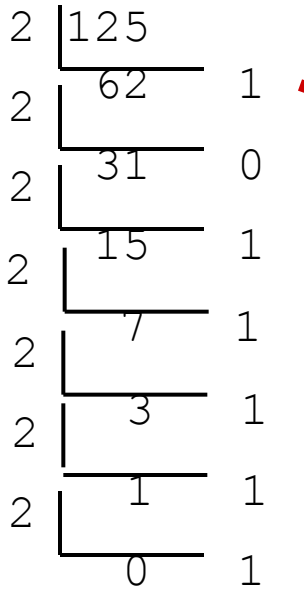
Divide by two, keep track of the remainder

First remainder is bit 0 (LSB, least-significant bit)

Second remainder is bit 1

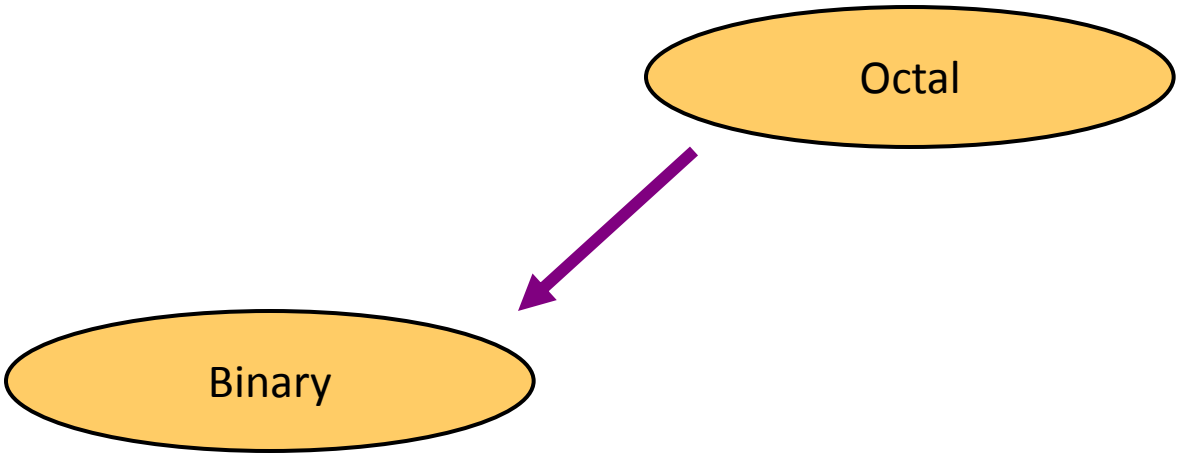
Decimal to Binary Conversion: Example

$125_{10} = ?_2$



$125_{10} = 1111101_2$

Octal to Binary Conversion

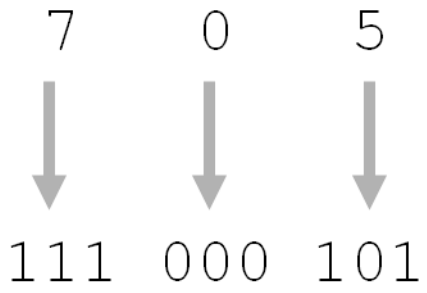


Technique:

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

Octal to Binary Conversion: Example

$705_8 = ?_2$



$705_8 = 111000101_2$

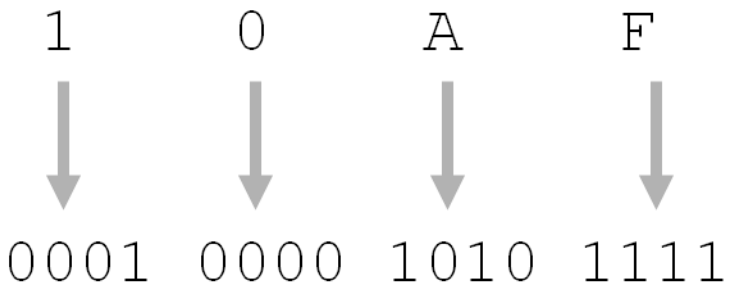
Hex to Binary Conversion

Technique:

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

Hex to Binary Conversion: Example

$10AF_{16} = ?_2$



$10AF_{16} = 0001000010101111_2$

Decimal to Octal Conversion

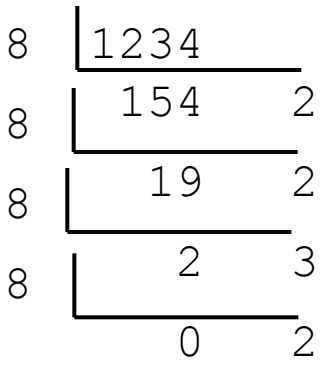
Technique:

Divide by 8

Keep track of the remainder

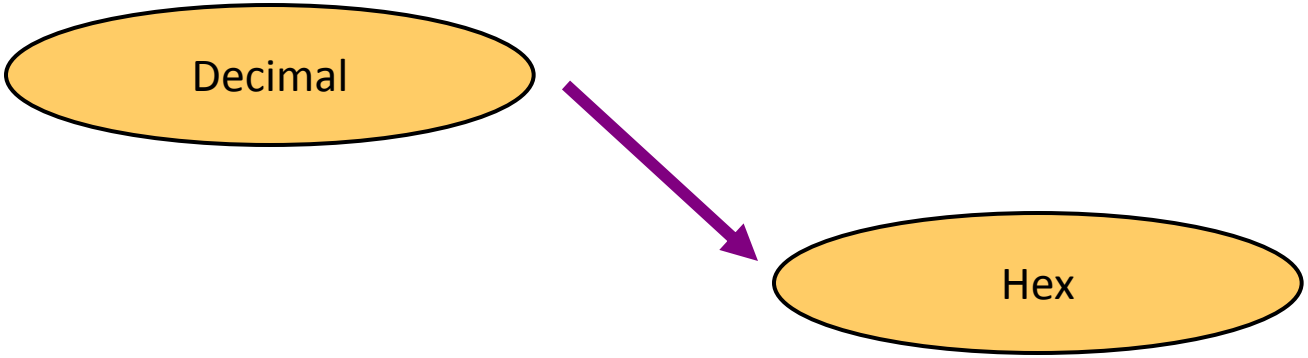
Decimal to Octal Conversion: Example

$1234_{10} = ?_{8}$



$1234_{10} = 2322_{8}$

Decimal to Hex Conversion



Technique:

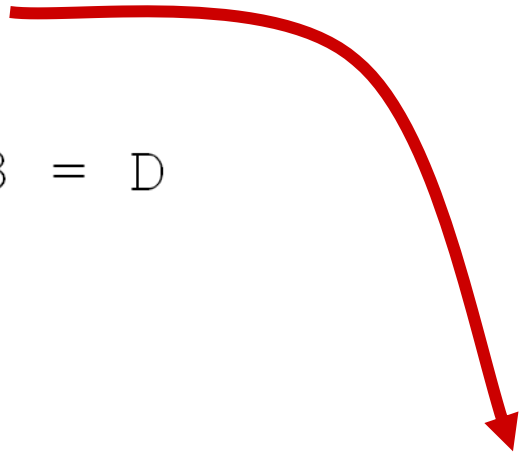
Divide by 16

Keep track of the remainder

Decimal to Hex Conversion: Example

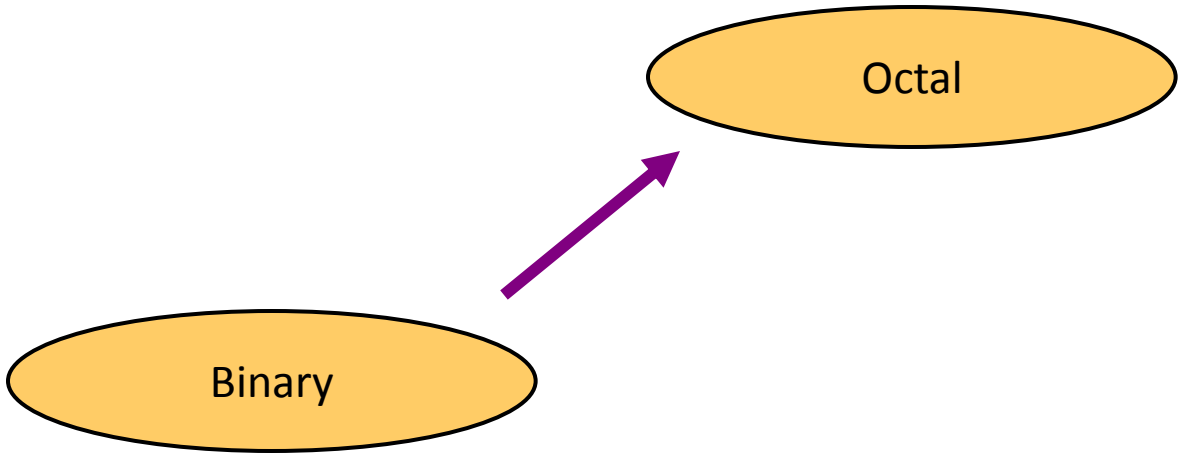
$1234_{10} = ?_{16}$

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



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

Binary to Octal Conversion



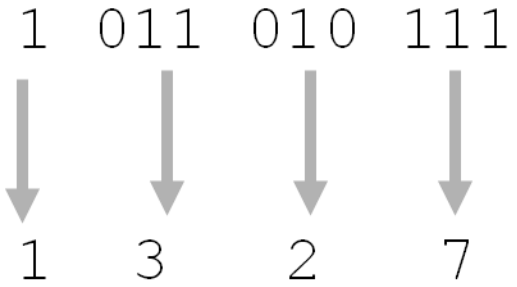
Technique:

Group bits in threes, starting on right

Convert to octal digits

Binary to Octal Conversion: Example

$1011010111_2 = ?_8$



$1011010111_2 = 1327_8$

Binary to Hex Conversion

Technique:

Group bits in fours, starting on right

Convert to hexadecimal digits

Binary to Hex Conversion: Example

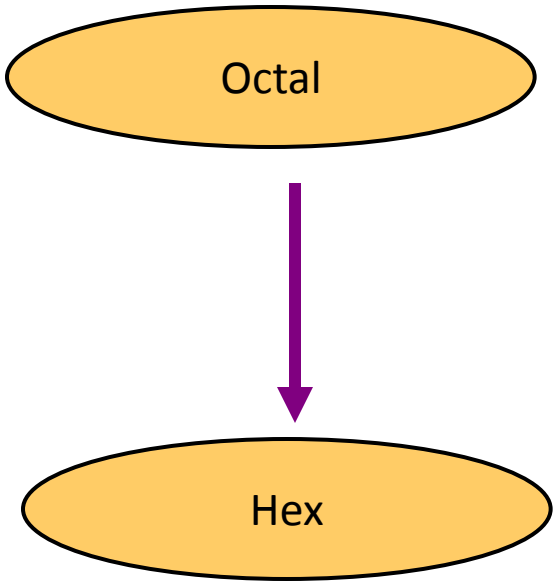
$1010111011_2 = ?_{16}$

10 1011 1011

2 B B

$1010111011_2 = 2BB_{16}$

Octal to Hex Conversion

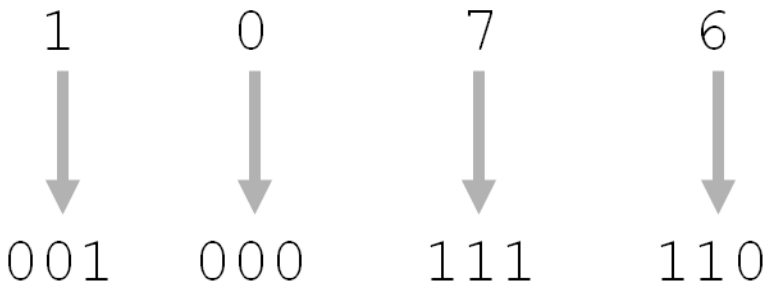


Technique:

Use binary as an intermediary

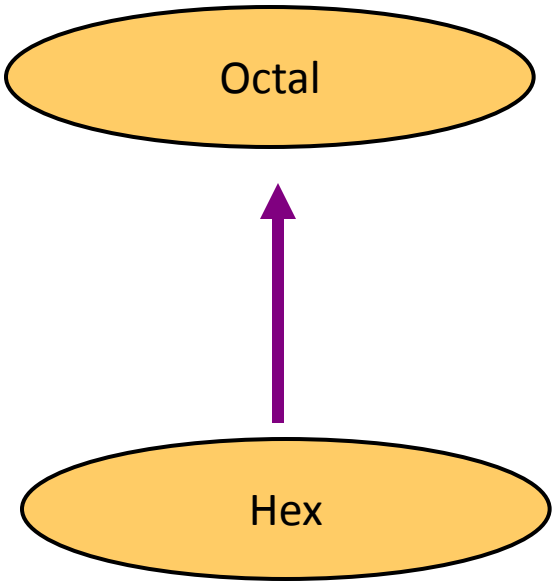
Octal to Hex Conversion: Example

$1076_8 = ?_{16}$



$1076_8 = 23E_{16}$

Hex to Octal Conversion

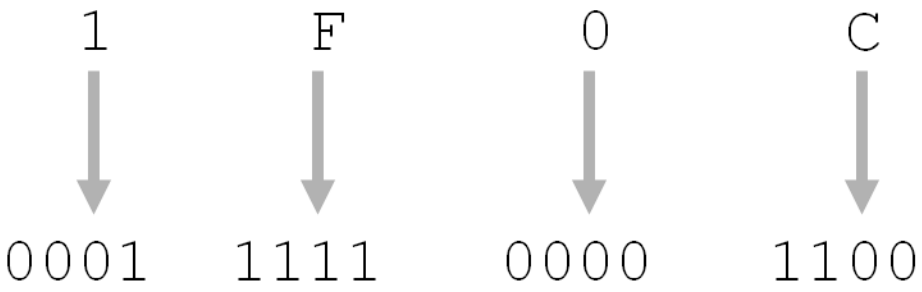


Technique:

Use binary as an intermediary

Hex to Octal Conversion: Example

$1F0C_{16} = ?_{8}$



$1F0C_{16} = 17414_{8}$

Data Formats

Alpha Numeric Data

- Four standards for representing letters (alpha) and numbers
  - BCD – Binary-coded decimal
  - ASCII – American standard code for information interchange
  - EBCDIC – Extended binary-coded decimal interchange code
  - Unicode

Binary-Coded-Decimal System (BCD)

- The BCD is simply the 4 bit representation of the decimal digit
- For multiple digit base 10 numbers, each symbol is represented by its BCD digit

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Binary-Coded-Decimal System (BCD)

- Technique:
  - Decimal-to-BCD conversion
    - Convert each decimal digit to its 4-bit binary code
  - BCD-to-Decimal conversion
    - Reverse the process

Decimal to BCD Conversion

Decimal	BCD
0	0000
1	0001
..	…
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
14	0001 0100
15	0001 0101
…	…
20	0010 0000

Binary-Coded Decimal (BCD)

Four bits per digit

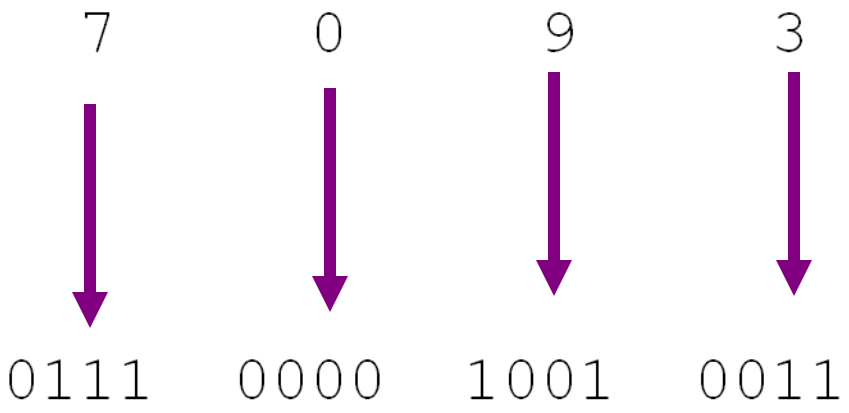
Digit	Bit pattern
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Note: the following bit patterns are not used:

1010
1011
1100
1101
1110
1111

Decimal to BCD Conversion: Example

7093_{10} = ? (in BCD)



The ASCII Code

- American National Standard Code for Information Interchange (ASCII)
  - (Pronounced *ass-key)*
  - Each character is coded as a byte

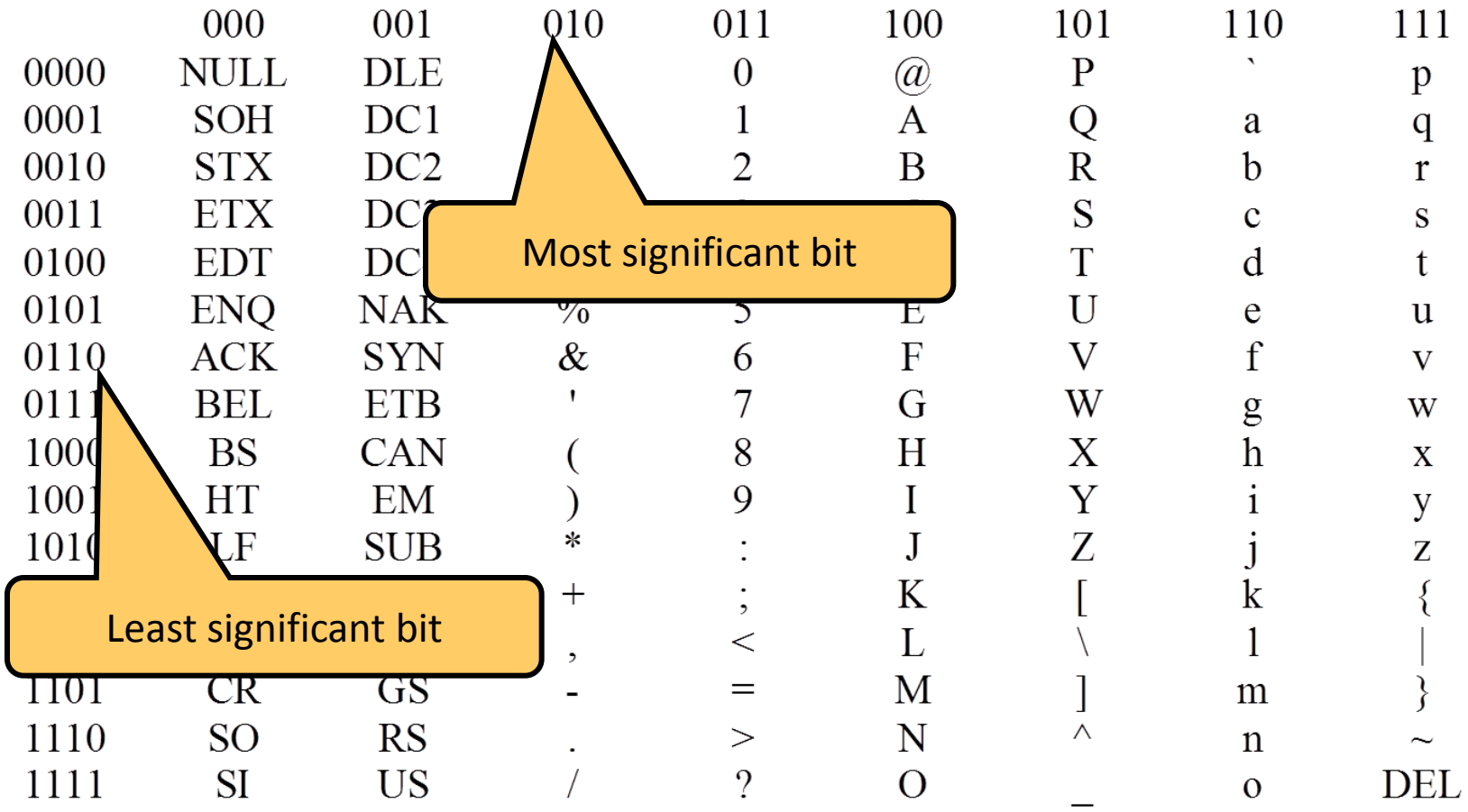
ASCII Features

- 7-bit code
- 8^{th} bit is unused (or used for a parity bit)
- $2^7 = 128$ codes
- Two general types of codes:
  - 95 are “Graphic” codes (displayable on a console)
  - 33 are “Control” codes (control features of the console or communications channel)

ASCII Chart

	000	001	010	011	100	101	110	111
0000	NULL	DLE		0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EDT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Working with ASCII Chart



	000	001	010	011	100	101	110	111
0000	NULL	DLE		0	@	P	`	p
0001	SOH	DC1		1	A	Q	a	q
0010	STX	DC2		2	B	R	b	r
0011	ETX	DC				S	c	s
0100	EDT	DC				T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
			+	;	K	[	k	{
			,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Working with ASCII Chart: Example

e.g., ‘a’ = 1100001

	000	001	010	011	100	101	110	111
0000	NULL	DLE		0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EDT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

ASCII Code: Example

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

EBCDIC

- Extended BCD Interchange Code
  - another code for characters called EBCDIC
  - (pronounced *ebb'-se-dick*)
  - 8-bit code
  - Developed by IBM

Unicode

- 16-bit standard
- Developed by a consortia
  - http://www.unicode.org/
- Intended to supercede older 7- and 8-bit codes

Summary 1/2

- Any number system can be converted to decimal by multiplying each digit by its weighting factor
- The weighting factor for the least significant digit in any number system is always 1
- Binary numbers can be converted to octal by forming groups of 3 bits and to hexadecimal by forming groups of 4 bits

Summary 2/2

- The successive division procedure can be used to convert from decimal to binary, octal, or hexadecimal
- The binary-coded-decimal system assigns a four-bit binary code groups of 4 bits to each digit 0 through 9 in a **decimal** (base-10) number system
- ASCII is used by computers to represent all letters, numbers and symbols in digital form