SEE 1223: Digital Electronics

1 – Number Systems

Number Systems

- Standard number systems
  - Decimal
  - Binary
  - Hexadecimal
  - Octal
- Binary Codes
  - Binary Coded Decimal (BCD)
  - Gray Codes
  - ASCII
- Representation of negative numbers
  - Sign magnitude
  - 1's complement and 2's complement
- Arithmetic operations using 2's complement

Binary Numbers

- Counting in binary and decimal:

Binary		Decimal
0 0 0 0	=>	0
0 0 0 1	=>	1
0 0 1 0	=>	2
0 0 1 1	=>	3
0 1 0 0	=>	4
0 1 0 1	=>	5
0 1 1 0	=>	6
0 1 1 1	=>	7
1 0 0 0	=>	8
1 0 0 1	=>	9
1 0 1 0	=>	10
1 0 1 1	=>	11
1 1 0 0	=>	12
1 1 0 1	=>	13
1 1 1 0	=>	14
1 1 1 1	=>	15

How to represent 16 in binary?
=> 10000_2

How to represent decimal 33?
=> 100001_2

What is the value of 100101_2
=> 37

Binary Numbers (cont.)

- Binary number system uses “0” and “1”
- Example: find the decimal value of 00101

Bit Position:	4	3	2	1	0	
Binary:	0	0	1	0	1	
Decimal:	$0x2^4$	$0x2^3$	$1x2^2$	$0x2^1$	$1x2^0$	
	0 +	0 +	4 +	0 +	1	= 5

Therefore, $00101_2 = 5_{10}$

Binary Numbers (cont.)

- Convert these binary numbers to decimal:
  - 1010_2 => $2^3 + 2^1 = 10$
  - 10111_2 => $2^4 + 2^2 + 2^1 + 2^0 = 23$
- Convert these decimal numbers to binary:
  - 19 => $2^4 + 2^1 + 2^0 = 10011_2$
  - 58 => $2^5 + 2^4 + 2^3 + 2^1 = 111010_2$

Hexadecimal Numbers

- Counting in hexadecimal

Binary		Decimal		Hexadecimal
0 0 0 0	=>	0	=>	0
0 0 0 1	=>	1	=>	1
0 0 1 0	=>	2	=>	2
0 0 1 1	=>	3	=>	3
0 1 0 0	=>	4	=>	4
0 1 0 1	=>	5	=>	5
0 1 1 0	=>	6	=>	6
0 1 1 1	=>	7	=>	7
1 0 0 0	=>	8	=>	8
1 0 0 1	=>	9	=>	9
1 0 1 0	=>	10	=>	A
1 0 1 1	=>	11	=>	B
1 1 0 0	=>	12	=>	C
1 1 0 1	=>	13	=>	D
1 1 1 0	=>	14	=>	E
1 1 1 1	=>	15	=>	F

How to represent 16 in hexadecimal?

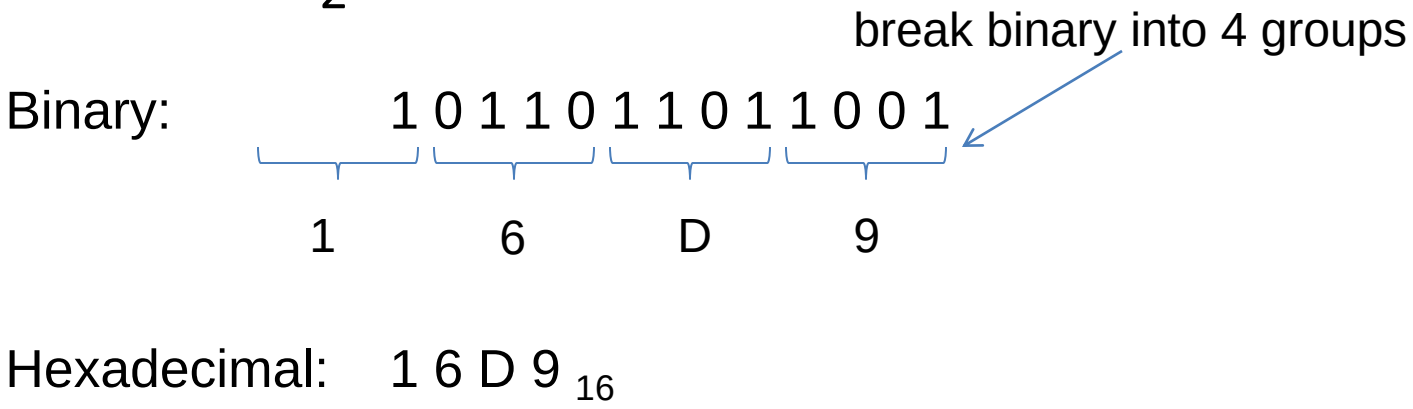
=> 10_{16}

Continue counting from 10_{16}..

11, 12, 13, 14, 15, 16, 17, 18, 19, 1A
1B, 1C, 1D, 1E, 1F, 20…

Hexadecimal Numbers (cont.)

- Hexadecimal number conversion: Convert 1011011011001_2 to hexadecimal



Hexadecimal: $1\ 6\ D\ 9_{16}$

Can you convert this hex number to decimal?

$=>1x16^3+6x16^2+13x16^1+9x16^0 = 5849_{10}$

Hexadecimal Numbers (cont.)

- Convert the following to binary:
  - $CF8E_{16}$ => $1100\ 1111\ 1000\ 1110_2$
  - 974_{16} => $1001\ 0111\ 0100_2$
- Convert the following to hexadecimal
  - $1111\ 0000\ 1010_2$ => $F0A_{16}$
  - $10\ 0001\ 1101\ 1001_2$ => $21D9_{16}$

Octal Numbers

- Counting in Octal

Binary		Decimal		Hexadecimal		Octal
0 0 0 0	=>	0	=>	0	=>	0
0 0 0 1	=>	1	=>	1	=>	1
0 0 1 0	=>	2	=>	2	=>	2
0 0 1 1	=>	3	=>	3	=>	3
0 1 0 0	=>	4	=>	4	=>	4
0 1 0 1	=>	5	=>	5	=>	5
0 1 1 0	=>	6	=>	6	=>	6
0 1 1 1	=>	7	=>	7	=>	7
1 0 0 0	=>	8	=>	8	=>	10
1 0 0 1	=>	9	=>	9	=>	11
1 0 1 0	=>	10	=>	A	=>	12
1 0 1 1	=>	11	=>	B	=>	13
1 1 0 0	=>	12	=>	C	=>	14
1 1 0 1	=>	13	=>	D	=>	15
1 1 1 0	=>	14	=>	E	=>	16
1 1 1 1	=>	15	=>	F	=>	17

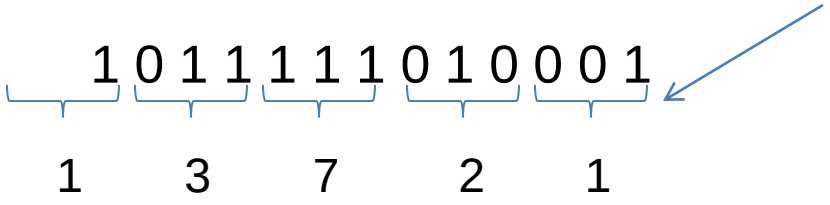
After 17_8?

=> 20_8

Octal Numbers (cont.)

- Octal numbers conversion: Convert 1011111010001 to octal

break binary into 3 groups



Octal: $1\ 3\ 7\ 2\ 1_8$

Can you convert this octal number to decimal?

$=>1x8^4+3x8^3+7x8^2+2x8^1+1x8^0= 6097_{10}$

Octal Numbers (cont.)

- Convert the following to binary
  - 25_8 => $10\ 101_2$
  - 140_8 => $001\ 100\ 000_2$
- Convert the following to octal
  - $110\ 101_2$ => 65_8
  - $1\ 101\ 111\ 001_2$ => 1571_8

More Number Conversions

- Convert $A7B_{16}$ to binary and decimal – easy
- Convert 650_{10} to hexadecimal – 2 ways
  - Convert to binary first, then to hex
  - Convert directly to hex

More number conversions (cont.)

650_{10} to binary using repeated division method:

650/2 = 325, remainder 0 ← Least significant bit (MSB)
325/2 = 162, remainder 1
162/2 = 81, remainder 0
81/2 = 40, remainder 1
40/2 = 20, remainder 0
20/2 = 10, remainder 0
10/2 = 5, remainder 0
5/2 = 2, remainder 1
2/2 = 1, remainder 0
1/2 = 0, remainder 1 ← Most significant bit (MSB)

Therefore, $650_{10} = 1010001010_2$

What is 650_{10} in hexadecimal? $650_{10} = 28A_{16}$

More number conversions (cont.)

650_{10} to hexadecimal using repeated division method:

650/16 = 40.625 → 0.625 x 16 = 10 ➔ A ← Least significant bit (MSB)

40/16 = 2.5 → 0.5 x 16 = 8 ➔ 8

2/16 = 0.125 → 0.125 x 16 = 2 ➔ 2 ← Most significant bit (MSB)

Therefore, $650_{10} = 28A_{16}$

Binary Coded Decimal (BCD)

- Each decimal digit (0 to 9) is represented by 4 bit binary

Binary		Decimal
0 0 0 0	=>	0
0 0 0 1	=>	1
0 0 1 0	=>	2
0 0 1 1	=>	3
0 1 0 0	=>	4
0 1 0 1	=>	5
0 1 1 0	=>	6
0 1 1 1	=>	7
1 0 0 0	=>	8
1 0 0 1	=>	9

How to represent 28 in BCD?
=> $0010\ 1000_2$

What is 0011 0010 in BCD?
=> 32

What is 32 in binary?
=> 100000_2

Gray Code

Binary		Decimal		Gray Code
0 0 0 0	=>	0	=>	0 0 0 0
0 0 0 1	=>	1	=>	0 0 0 1
0 0 1 0	=>	2	=>	0 0 1 1
0 0 1 1	=>	3	=>	0 0 1 0
0 1 0 0	=>	4	=>	0 1 1 0
0 1 0 1	=>	5	=>	0 1 1 1
0 1 1 0	=>	6	=>	0 1 0 1
0 1 1 1	=>	7	=>	0 1 0 0
1 0 0 0	=>	8	=>	1 1 0 0
1 0 0 1	=>	9	=>	1 1 0 1
1 0 1 0	=>	10	=>	1 1 1 1
1 0 1 1	=>	11	=>	1 1 1 0
1 1 0 0	=>	12	=>	1 0 1 0
1 1 0 1	=>	13	=>	1 0 1 1
1 1 1 0	=>	14	=>	1 0 0 1
1 1 1 1	=>	15	=>	1 0 0 0

- Exhibits a single bit change from one code word to another

Binary-Gray Code Conversions

- MSB of Gray Code is the same MSB in binary
- From left to right, add each adjacent pair of binary code, discard carry

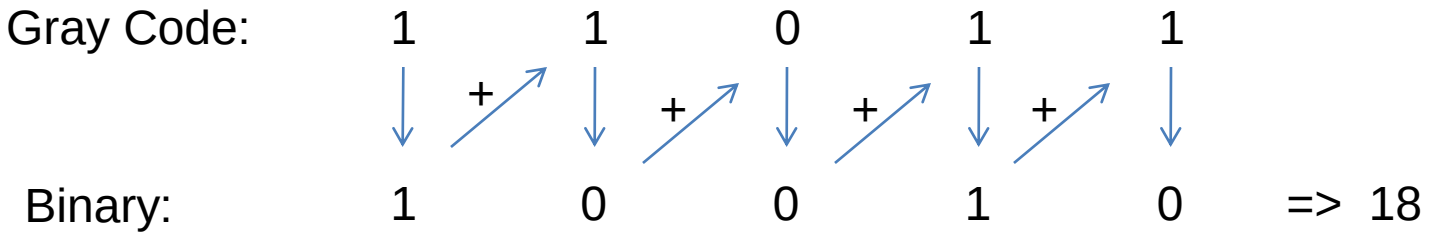
Binary: 1 + 0 + 1 + 1 + 0 => 22

Gray Code: 1 1 1 0 1

Therefore, binary 10110 is equivalent to gray code 11101

Gray Code-Binary Conversions

- MSB of binary is the same MSB in Gray Code
- From left to right, add each generated binary code with adjacent Gray Code, discard carry



Therefore, gray code11011 is equivalent to binary 10010

ASCII

- American Standard Code for Information Interchange
- 128 characters, represented by 8-bit binary code with MSB '0'
- The 8-bit code runs from 00_{16} to $7F_{16}$
- The first 32 ASCII characters used for controls such as ESC, new line, space, start of text, etc
- Other characters include letters (upper and lower case), decimal digits, and symbols

ASCII Table

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

ASCII Example

- Find the ASCII equivalent “ab.12” in binary

  => 0110 0001 0110 0010 0010 1110 0011 0001 0011 0010

- A receiver receives the bit sequence: $504D544B23_{16}$
  - Find the ASCII characters corresponding the transmitted data

    => PMTK#