COMPUTER SYSTEMS & PROGRAMMING

Number Systems & Conversion

Zuhaib A. Shaikh,
Asst. Prof., CSE Deptt.,QUEST
Web: zuhaib-shaikh.neocities.org

Numbering Systems and Conversion

- Binary Numbers
- Hexadecimal Numbers
- Conversion among Binary, Decimal, and Hexadecimal
- Two's Complement Notation
- Two's Complement Arithmetic
- Alphanumeric Codes
- Floating Point Number representation

Binary Numbers

- Binary digITs or Bits are used by the Digital Computers
- Also called Base 2 Numbers
- 1 means a High voltage Level and 0 means Low voltage level (may be 0)
- Similarity with the Decimal Number System
- Decimal 1327 is 1 thousands, 3 hundreds, 2 twenties, and 7 ones
- Place value characteristics in Binary (…64, 32, 16, 8, 4, 2,1)
- … 2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0
- LSB and MSB in Binary Numbers

Binary – Decimal Conversion

- $(0001)_2 = ?_{10}$
- $(0101)_2 = ?_{10}$
- $(1000)_2 = ?_{10}$
- $(1011)_2 = ?_{10}$
- $(1111)_2 = ?_{10}$
- $(0111)_2 = ?_{10}$

Decimal - Binary Conversion

- Number is repeatedly divided by 2 (keeping remainder) until it result is 0 or 1 (End)
- Try to convert $(5)_{10}$ into binary
  - 5 % 2 ➔ Ans. 2 and remainder 1
  - 2 % 2 ➔ Ans. 1 and remainder 0
  - Answer is: 101

Decimal - Binary Conversion

- $(155)_{10} = (??)_2$
- $(77)_{10} = (??)_2$
- $(38)_{10} = (??)_2$
- $(48)_{10} = (??)_2$
- $(215)_{10} = (??)_2$
- $(455)_{10} = (??)_2$

Hexadecimal Numbers

- Remembering and typing Long strings of bits is tedious and error prone
- Using Decimal Conversion is a good idea but the conversion process takes a long time
- In most of systems, hexadecimal numbers are used as shorthand representation of binary numbers
- 16 digits. First 10 digits are same as decimal then A-F
- First 16 Hexadecimal numbers and their binary equivalents
- 10011110 in Hexadecimal is 9Eh
- Letter h differentiate it from the other numbers

Binary - Hexadecimal Conversion

- Starting from LSB, make group of four bits and write their Hexadecimal equivalent
- $(111010)_2 = (??)_{16}$
- $(1111111)_2 = (??)_{16}$
- $(101000)_2 = (??)_{16}$
- $(1011110)_2 = (??)_{16}$
- $(1001100)_2 = (??)_{16}$
- $(1110011)_2 = (??)_{16}$

Hexadecimal – Binary Conversion

- Replace each hexadecimal digit with its four bit binary equivalent
- $(7F)_{16} = (??)_2$
- $(2C)_{16} = (??)_2$
- $(5A)_{16} = (??)_2$
- $(97)_{16} = (??)_2$
- $(100)_{16} = (??)_2$
- $(255)_{16} = (??)_2$

Decimal-Hexadecimal Conversion

- Similar to Decimal – Binary (Now divide by 16)
- Try to convert $(280)_{10}$ into Hexadecimal

  - 280 % 16 ➔ Ans. 17 and remainder 8
  - 17 % 16 ➔ Ans. 1 and remainder 1

  - Answer is: 118

Hexadecimal-Decimal Conversion

- Same as Binary to Decimal. (Now powers of 16)
- $(7F)_{16} = (??)_{10}$
- $(2C)_{16} = (??)_{10}$
- $(5A)_{16} = (??)_{10}$
- $(97)_{16} = (??)_{10}$
- $(100)_{16} = (??)_{10}$
- $(255)_{16} = (??)_{10}$

Binary Arithmetic

- Addition
- Subtraction
- Multiplication
- Division

- Solve the following:
  - (i) 1010 + 0101 (ii) 1101 + 0101 (iii) 01011011 + 00001111
  - (iv) 1110 – 1000 (v) 1010 + 0101 (vi) 01100110 – 00011010
  - (vii) 1001 X 11 (viii) 1101 X 1001 (ix) 1110 X 1110

Twos Complement Notation

- To simplify the computer circuitry, twos complement codes are used
- Negative numbers are represented using Twos complement
- If sign of numbers is being taken care of then the representation is called Signed
- In signed representation negative numbers are represented using twos complement
- If signed representation is being used, then the MSB of the register tells the sign of the number
- Table of 8-bit Signed numbers

Twos Complement Notation

- If MSB of the number is 0, the number is positive and is in normal form
- If MSB is 1 then number is negative and is in Twos Complement form
- Remaining 7 bits tell the magnitude if the number
- If register = 0100 0001 then it has value decimal +65
- If register = 0111 1111 then it has value decimal +127
- If register = 1111 1111 then it has value decimal -1
- What will be twos complement for -9?
- What will be twos complement for -16?
- What will be twos complement for -100?
- What will be twos complement for -95?

Twos Complement Arithmetic

- Using Twos complement, subtraction can be performed by means of addition
- This saves Subtraction circuitry
- Following are to be noted for Signed Arithmetic
  - If (+A)+(+B) Then its like normal binary addition
  - If (-A)+(+B) Then first take 2's complement of A then add with B
  - If (+A)+(-B) Then first take 2's complement of B then add with A
  - If (-A)+(-B) Then first take 2's complement of A and B and then add both
  - Result is also in 2's complement. (Check its MSB)

Twos Complement Arithmetic

- Perform the following signed arithmetic
  - (i) (+7) + (+1)
  - (ii) (+31) + (26)
  - (iii) (+8) + (-5)
  - (iv) (+89) + (-46)
  - (v) (-3) + (-4)

Alphanumeric Codes

- Both characters and numbers are coded
- Used when display data on CRT, LCD or other displays
- The most common code is ASCII
- American Standard Code for Information Interchange
- Basic ASCII codes are 7-bit wide
- Extended ASCII are 8-bit wide
- Partial ASCII chart

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

Extended ASCII Table

128	Ç	144	É	160	á	176	░	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	^{n}
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	^{2}
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

Floating Point Representation

- Fixed point notations
  - Bits are divided into two equal groups
  - 1st group for number which is before decimal and 2nd group for number that is after decimal
  - Faster and less precise
- IEEE 754 Standard (32bit or 64 bit)
  - 32bit standard
    - 1 bit for sign
    - 8 bits for exponent
    - 23 bits for mantissa
  - Complex, lengthy and more precise

Questions