Binary Data, Integer Addition and Subtraction

Tom Kelliher, CS 220

1 Administrivia

Today's Objectives

1. Understand that everything within a computer system is represented by binary values.
2. Understand the properties of, write, read, and convert ASCII data and unsigned and two's complement binary integers.
3. Convert between decimal, binary, octal, and hexadecimal representations of binary data.
4. Perform binary integer addition and subtraction, recognizing when overflow has occurred.

Next Up

Read 2.5. Figure 2.5 on pg. 70 note: The edge labeled "No" leaving the lower decision diamond should not lead back to the "Subtract divisor from dividend" step. Instead, it should lead back to the next step — "Shift divisor one place right." In other words, the "Shift divisor one place right" step is *not* part of the loop.

Read mult.c and div.c on the class web site. Prior to class, make sure you can log-in to your phoenix account.

1. Perform left and right arithmetic shifts on binary numbers, recognizing that this is equivalent to multiplying by two and dividing by two, respectively, and also recognizing when the result will be incorrect.

2. Perform unsigned binary multiplication, using both the traditional “paper and pencil” method and Booth’s algorithm.

3. Perform unsigned restoring and non-restoring division.

4. Implement Booth’s algorithm and non-restoring division by modifying C programs implementing “paper and pencil” multiplication and restoring division.

2 Warm-Up

1. ASCII is a seven bit code. A single location in RAM is ordinarily some integral number of bytes long (64 bits or 8 bytes, 32 bits or 4 bytes, etc.). How is ASCII data stored in RAM?

   (a) The seven bit ASCII characters are placed together, with a few extra 0 bits at the end of the location.

   (b) Each memory location holds a single ASCII character.

   (c) Each seven bit ASCII character is padded with a 0 bit to be one byte in size.

   (d) ASCII is really an eight bit code.

2. A four bit binary number (1011, for example) can represent how many different values?

   (a) 1

   (b) 4

   (c) 8

   (d) 16

3. The hexadecimal value 0x6a corresponds to which of these binary values?

   (a) 6a

   (b) 610

   (c) 1101010

   (d) 01101010

4. The decimal value of the unsigned binary integer 1111 is

   (a) -1

   (b) 1

   (c) 4

   (d) 15

   (e) 16

5. To negate a two's complement binary number we

   (a) Invert the sign bit of the number

   (b) Invert all the bits of the number except for the sign bit

   (c) Invert all the bits of the number

   (d) Invert all the bits of the number and then add 1

6. The decimal value of the two's complement binary integer 1111 is

   (a) -1

   (b) 1

   (c) 4

   (d) 15

   (e) 16

3 Problems

1. Convert the following decimal values to eight bit two's complement binary values: 120 and -63. Express each value in both binary and hexadecimal representations.

2. The following hexadecimal numbers represent two's complement binary numbers. Convert each to decimal. 0xAB and 0x42.

3. Using the following ASCII code table, interpret the binary value 0101 0100 0110 1111 0110 1101 0010 0001 0000 0000 as an ASCII character string.

TABLE 2.1 The ASCII Code

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

4. Write the entire set of three bit unsigned binary numbers in order, from least to greatest. Include each number's decimal value.

5. Write the entire set of three bit two's complement numbers in order, from least to greatest. Include each number's decimal value.

6. Add the following two binary numbers, interpreting them as unsigned numbers (a) and as two's complement numbers (b). For each of the two cases, indicate the eight-bit sum and whether or not overflow has occurred. 00110111 and 01011011.

7. Add the following two binary numbers, interpreting them as unsigned numbers (a) and as two's complement numbers (b). For each of the two cases, indicate the eight-bit sum and whether or not overflow has occurred. 01001000 and 11100100.