CS/COE 0447 Fall 2009
Homework 2
Solution

1. (5 pts) Add the following unsigned binary numbers (show the carry and overflow bits)

```
  0111 1000 0100
+ 1001 1100 1101
----------------
1 1111 0001 1000   (Carry)
1 0001 0101 0001   (Result: overflowed 12 bits)
```

2. (5 pts) Subtract the following unsigned binary numbers (show the borrow and underflow bits)

```
  0101 0100 0101
- 1101 1011 1011
----------------
1 1111 0111 0100   (Borrow)
1 0111 1000 1010   (Result: underflowed 12 bits)
```

3. (5 pts) Convert the following decimal numbers to binary numbers:

2640 / 2 = 1320 + **0** / 2
1320 / 2 = 660 + **0** / 2
660 / 2 = 330 + **0** / 2
330 / 2 = 165 + **0** / 2
165 / 2 = 82 + **1** / 2
82 / 2 = 41 + **0** / 2
41 / 2 = 20 + **1** / 2
20 / 2 = 10 + **0** / 2
10 / 2 = 5 + **0** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
2640 = 101001010000b

4021 / 2 = 2010 + **1** / 2
2010 / 2 = 1005 + **0** / 2
1005 / 2 = 502 + **1** / 2
502 / 2 = 251 + **0** / 2
251 / 2 = 125 + **1** / 2
125 / 2 = 62 + **1** / 2
62 / 2 = 31 + **0** / 2
31 / 2 = 15 + **1** / 2

15 / 2 = 7 + **1** / 2
7 / 2 = 3 + **1** / 2
3 / 2 = 1 + **1** / 2
1 / 2 = 0 + **1** / 2
4021 = 111110110101b

1362 / 2 = 681 + **0** / 2
681 / 2 = 340 + **1** / 2
340 / 2 = 170 + **0** / 2
170 / 2 = 85 + **0** / 2
85 / 2 = 42 + **1** / 2
42 / 2 = 21 + **0** / 2
21 / 2 = 10 + **1** / 2
10 / 2 = 5 + **0** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
1362 = 10101010010b

4. (5 pts) Convert the following unsigned binary numbers to decimal numbers:

100010001000:
2^3 + 2^7 + 2^11 = 8 + 128 + 2048 = 2184

111010111100:
2^2 + 2^3 + 2^4 + 2^5 + 2^7 + 2^9 + 2^10 + 2^11 = 4 + 8 + 16 + 32 + 128 + 512 + 1024 + 2048 = 3772

111100001010:
2^1 + 2^3 + 2^8 + 2^9 + 2^10 + 2^11 = 2 + 8 + 256 + 512 + 1024 + 2048 = 3850

5. (5 pts) Convert the following decimal numbers into 9-bit binary numbers (with sign-magnitude):

11:
11 / 2 = 5 + **1** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
Sign bit: **0**
11 = 0 0000 1011b (sign magnitude)

-175:
175 / 2 = 87 + **1** / 2
87 / 2 = 43 + **1** / 2
43 / 2 = 21 + **1** / 2
21 / 2 = 10 + **1** / 2
10 / 2 = 5 + **0** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2

1 / 2 = 0 + **1** / 2
Sign bit: **1**
-175 = 1 1010 1111b (sign magnitude)

-47:
47 / 2 = 23 + **1** / 2
23 / 2 = 11 + **1** / 2
11 / 2 = 5 + **1** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
Sign bit: **1**
-47 = 1 0010 1111b (sign magnitude)

6. (5 pts) Convert the following 8-bit binary numbers (with sign-magnitude) to decimal numbers:

00110010:
2^1 + 2^4 + 2^5 = 2 + 16 + 32 = 50

10001110:
- (2^1 + 2^2 + 2^3) = - (2 + 4 + 8) = -14

11010111:
- (2^0 + 2^1 + 2^2 + 2^4 + 2^6) = - (1 + 2 + 4 + 16 + 64) = -87

7. (5 pts) Convert the following decimal numbers into 9-bit binary numbers in 1's complement form:

11:
11 = 0 0000 1011b (unsigned) (from point 5)
11 = 0 0000 1011b (one's complement)

-175:
175 = 0 1010 1111b (unsigned) (from point 5)
-175 = 1 0101 0000b (one's complement)

-47:
47 = 0 0010 1111b (unsigned) (from point 5)
-47 = 1 1101 0000b (one's complement)

8. (5 pts) Convert the following 8-bit binary numbers in 1's complement to decimal numbers:

00110010: (positive, because MSB is 0)
2^1 + 2^4 + 2^5 = 2 + 16 + 32 = 50

10001110: (negative, because MSB is 1)
Bitwise NOT: 01110001
- (2^0 + 2^4 + 2^5 + 2^6) = -(1 + 16 + 32 + 64) = -113

11010111: (negative, because MSB is 1)
Bitwise NOT: 00101000
-(2^3 + 2^5) = -(8 + 32) = -40

9. (5 pts) Convert the following decimal numbers into 9-bit binary numbers in 2's complement form:

11:
11 = 0 0000 1011b (unsigned) (from point 5)
11 = 0 0000 1011b (two's complement)

-175:
175 = 0 1010 1111b (unsigned) (from point 5)
-175 = 1 0101 0000b + 1 = 1 0101 0001b (two's complement)

-47:
47 = 0 0010 1111b (unsigned) (from point 5)
-47 = 1 1101 0000b + 1 = 1 1101 0001 (two's complement)

10. (5 pts) Convert the following 8-bit binary numbers in 2's complement to decimal numbers:

00110010:
2^1 + 2^4 + 2^5 = 2 + 16 + 32 = 50

10001110:
2^1 + 2^2 + 2^3 - 2^8 = 2 + 4 + 8 – 128 = - 114

11010111:
2^0 + 2^1 + 2^2 + 2^4 + 2^6 – 2^7 = 1 + 2 + 4 + 16 + 64 - 128 = - 41

11. (10 pts) For each of the following decimal expressions, show the converted binary numbers in 2's complement form and calculate the result using the binary numbers. Convert the result back to decimal.

a. 58 - 97 = ?
58 / 2 = 29 + **0** / 2
29 / 2 = 14 + **1** / 2
14 / 2 = 7 + **0** / 2
7 / 2 = 3 + **1** / 2
3 / 2 = 1 + **1** / 2
1 / 2 = 0 + **1** / 2
58 = 0011 1010b

97 / 2 = 48 + **1** / 2
48 / 2 = 24 + **0** / 2
24 / 2 = 12 + **0** / 2
12 / 2 = 6 + **0** / 2
6 / 2 = 3 + **0** / 2

3 / 2 = 1 + **1** / 2
1 / 2 = 0 + **1** / 2
-97 = -(0110 0001b) = 1001 1110b + 1 = 1001 1111b

```
  0011 1010
+ 1001 1111
-----------
0 0111 1100   (Carry)
0 1101 1001   (Result)
```

1101 1001:
2^0 + 2^3 + 2^4 + 2^6 – 2^7 = 1 + 8 + 16 + 64 – 128 = -39

b. 86 + 57 = ?
86 / 2 = 43 + **0** / 2
43 / 2 = 21 + **1** / 2
21 / 2 = 10 + **1** / 2
10 / 2 = 5 + **0** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
86 = 0101 0110b

57 / 2 = 28 + **1** / 2
28 / 2 = 14 + **0** / 2
14 / 2 = 7 + **0** / 2
7 / 2 = 3 + **1** / 2
3 / 2 = 1 + **1** / 2
1 / 2 = 0 + **1** / 2
57 = 0011 1001b

```
  0101 0110
+ 0011 1001
-----------
0 1110 0000   (Carry)
0 1000 1111   (Result: overflowed 8 bits)
```

0 1000 1111 (9 bits)
2^0 + 2^1 + 2^2 + 2^3 + 2^7 = 1 + 2 + 4 + 8 +128 = 143

c. -93 + (-55) = ?
93 / 2 = 46 + **1** / 2
46 / 2 = 23 + **0** / 2
23 / 2 = 11 + **1** / 2
11 / 2 = 5 + **1** / 2
5 / 2 = 2 + **1** / 2
2 / 2 = 1 + **0** / 2
1 / 2 = 0 + **1** / 2
-93 = -(01011101b) = 10100010b + 1 = 1010 0011b

55 / 2 = 27 + **1** / 2
27 / 2 = 13 + **1** / 2
13 / 2 = 6 + **1** / 2
6 / 2 = 3 + **0** / 2
3 / 2 = 1 + **1** / 2
1 / 2 = 0 + **1** / 2
-55 = =(00110111b) = 11001000b + 1 = 1100 1001b

```
  1010 0011
+ 1100 1001
-----------
1 0000 0110   (Carry)
1 0110 1100   (Result: underflowed 8 bits)
```

1 0110 1100 (9 bits):
2^2 + 2^3 + 2^5 + 2^6 - 2^8 = 4 + 8 + 32 + 64 - 256 = -148

12. (10 pts) Show the steps for the multiplication of 01101101b and 11011010b (unsigned) using Hardware Design 2 (available here: http://www.cs.pitt.edu/~childers/CS0447/lectures/numbers3.pdf). Draw a table similar to the following one and fill up the columns:

Iteration	Step	Multiplier (8 bits)	Multiplicand (8 bits)	Product (16 bits)
0	Initial Values	0110 1101	1101 1010	0000 0000 0000 0000
1	p = p + m	0110 1101	1101 1010	1101 1010 0000 0000
	shift p right, shift m right	0011 0110		0110 1101 0000 0000
2	no op	0011 0110	1101 1010	0110 1101 0000 0000
	shift p right, shift m right	0001 1011		0011 0110 1000 0000
3	p = p + m	0001 1011	1101 1010	1 0001 0000 1000 0000
	shift p right, shift m right	0000 1101		1000 1000 0100 0000
4	p = p + m	0000 1101	1101 1010	1 0110 0010 0100 0000
	shift p right, shift m right	0000 0110		1011 0001 0010 0000
5	no op	0000 0110	1101 1010	1011 0001 0010 0000
	shift p right, shift m right	0000 0011		0101 1000 1001 0000
6	p = p + m	0000 0011	1101 1010	1 0011 0010 1001 0000
	shift p right, shift m right	0000 0001		1001 1001 0100 1000
7	p = p + m	0000 0001	1101 1010	1 0111 0011 0100 1000
	shift p right, shift m right	0000 0000		1011 1001 1010 0100
8	no op	0000 0000	1101 1010	1011 1001 1010 0100
	shift p right, shift m right	0000 0000		0101 1100 1101 0010

13. (10 pts) Show the steps for the multiplication of 01101101b and 11011010b (unsigned) using Hardware Design 3 (available here: http://www.cs.pitt.edu/~childers/CS0447/lectures/numbers3.pdf). Draw a table similar to the following one and fill up the columns:

Iteration	Step	Multiplicand (8 bits)	Product (16 bits)
0	Initial Values	1101 1010	0000 0000 0110 1101
1	p = p + m	1101 1010	1101 1010 0110 1101
	shift p right		0110 1101 0011 0110
2	no op	1101 1010	0110 1101 0011 0110
	shift p right		0011 0110 1001 1011
3	p = p + m	1101 1010	1 0001 0000 1001 1011
	shift p right		1000 1000 0100 1101
4	p = p + m	1101 1010	10110001001001101
	shift p right		1011000100100110
5	no op	1101 1010	1011000100100110
	shift p right		0101100010010011
6	p = p + m	1101 1010	10011001010010011
	shift p right		1001 1001 0100 1001
7	p = p + m	1101 1010	1 0111 0011 0100 1001
	shift p right		1011 1001 1010 0100
8	no op	1101 1010	1011 1001 1010 0100
	shift p right		0101 1100 1101 0010

14. (5 pts) Convert the following decimal numbers into 9-bit binary numbers in Booth's encoding form:

11 = 0 0000 1011b (two's complement) (from point 9)
Add 0 at the right: 0000010110
Booth's encoding: 00001-110-1

-175 = 1 0101 0000b + 1 = 1 0101 0001b (two's complement) (from point 9)
Add 0 at the right: 1010100010
Booth's encoding: -11-11-1001-1

-47 = 1 1101 0000b + 1 = 1 1101 0001 (two's complement) (from point 9)
Add 0 at the right: 1110100010
Booth's encoding: 00-11-1001-1

15. (5 pts) Convert the following 8-bit binary numbers into Booth's encoding form:

00110010:
Add 0 at the right: 001100100
Booth's encoding: 010-101-10

10001110:
Add 0 at the right: 100011100
Booth's encoding: -100100-10

11010111:
Add 0 at the right: 110101110
Booth's encoding: 0-11-1100-1

16. (10 pts) Show the steps for the multiplication of 01101101b and 11011010b (signed) using Booth's algorithm (available here: http://www.cs.pitt.edu/~childers/CS0447/lectures/numbers3.pdf). Draw a table similar to the following one and fill up the columns:

Iteration	Step	Multiplicand (8 bits)	Product (17 bits)
0	Initial Values	1101 1010	0000 0000 0110 1101 0
1	p = p - m	1101 1010	0010 0110 0110 1101 0
	shift p right		0001 0011 0011 0110 1
2	p = p + m	1101 1010	1110 1101 0011 0110 1
	shift p right		1111 0110 1001 1011 0
3	p = p - m	1101 1010	0001 1100 1001 1011 0
	shift p right		0000 1110 0100 1101 1
4	p = p	1101 1010	0000 1110 0100 1101 1
	shift p right		0000 0111 0010 0110 1
5	p = p + m	1101 1010	1110 0001 0010 0110 1
	shift p right		1111 0000 1001 0011 0
6	p = p - m	1101 1010	0001 0110 1001 0011 0
	shift p right		0000 1011 0100 1001 1
7	no op	1101 1010	0000 1011 0100 1001 1
	shift p right		0000 0101 1010 0100 1
8	p = p + m	1101 1010	1101 1111 1010 0100 1
	shift p right		1110 1111 1101 0010 0
	Drop least significant bit		1110 1111 1101 0010