Homework Assignment 3 Answered

Signed Binary to Decimal

Convert each **signed binary number** into decimal. **DO NOT USE** two's complement. See task 3 of programming assignment 3 for instructions. Please show your work.

1. $0110 \Rightarrow -(0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

   $\Rightarrow 0 + 4 + 2 + 0$

   $\Rightarrow 6$

2. $1110 \Rightarrow -(1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

   $\Rightarrow -8 + 4 + 2 + 0$

   $\Rightarrow -2$

3. $001011 \Rightarrow -(0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

   $\Rightarrow 0 + 0 + 8 + 0 + 2 + 1$

   $\Rightarrow 11$

4. $111111 \Rightarrow -(1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

   $\Rightarrow -32 + 16 + 8 + 4 + 2 + 1$

   $\Rightarrow -1$

5. $01111111 \Rightarrow -(0 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4)$

   $+ (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

   $\Rightarrow 0 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$

   $\Rightarrow 255$

6. $10000000 \Rightarrow -(1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (0 \times 2^4)$

   $+ (0 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$

   $\Rightarrow -256 + 0 + 0 + 0 + 0 + 0 + 0 + 0$

   $\Rightarrow -256$

7. $000000000101 \Rightarrow -(0 \times 2^{11}) + (0 \times 2^{10}) + (0 \times 2^9) + (0 \times 2^8) + (0 \times 2^7) + (0 \times 2^6)$

$$+ (0 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$\Rightarrow 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 4 + 0 + 1$$

$$\Rightarrow 5$$

8. $111111111111 \Rightarrow -(1 \times 2^{11}) + (1 \times 2^{10}) + (1 \times 2^9) + (1 \times 2^8) + (1 \times 2^7) + (1 \times 2^6)$

$$+ (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$

$$\Rightarrow -2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$

$$\Rightarrow -1$$

9. $101010 \Rightarrow -(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$

$$\Rightarrow -32 + 0 + 8 + 0 + 2 + 0$$

$$\Rightarrow -22$$

10. $0101 \Rightarrow -(0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

$$\Rightarrow 0 + 4 + 0 + 1$$

$$\Rightarrow 5$$

Unsigned Hexadecimal to Binary

Convert each **unsigned hexadecimal number** to binary. No work needed.

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

1. A   => 1010
2. 2F   => 0010 1111
3. 0   => 0000
4. 7FF   => 0111 1111 1111
5. 1C   => 0001 1100
6. 9B   => 1001 1011
7. FF   => 1111 1111
8. 123   => 0001 0010 0011
9. 5   => 0101
10. ABCD => 1010 1011 1100 1101

Signed Binary Subtraction

Find the difference for each of the following signed binary subtractions. Subtraction is just addition with the subtrahend inverted. $10 - 5 = 10 + (-5)$. Please show your work.

1. $0101 - 111$

    One's Complement: $111 \Rightarrow 000$

Carry Bits		0 0	
		0 0 0	
	+	0 0 1	
Sum		0 0 1	

    Two's Complement: 001

Carry Bits		0 0 1
		0 1 0 1
	+	0 0 0 1
Sum		0 1 1 0

    Answer = 0110

    Both signed binary operands are positive. The final answer is also positive. There is no overflow.

2. $1110 - 100101$

    One's Complement: $100101 \Rightarrow 011010$

Carry Bits		0 0 0 0 0
		0 1 1 0 1 0
	+	0 0 0 0 0 1
Sum		0 1 1 0 1 1

    Two's Complement: 011011

Carry Bits		1 1 1 1 0
		1 1 1 1 1 0
	+	0 1 1 0 1 1
Sum		0 1 1 0 0 0

    Answer = 011000

    The signed binary operands have different sign bits. Therefore, we don't need to check for overflow.

3. $001011 - 101$

   One's Complement: $101 \Rightarrow 010$

Carry Bits	0 0
	0 1 0
+	0 0 1
Sum	0 1 1

   Two's Complement: 011

Carry Bits	0 0 0 1 1
	0 0 1 0 1 1
+	0 0 0 0 1 1
Sum	0 0 1 1 1 0

   Answer = 001110

   Both signed binary operands are positive. The final answer is also positive. There is no overflow.

4. $111111 - 11010$

   One's Complement: $11010 \Rightarrow 00101$

Carry Bits	0 0 0 0 1
	0 0 0 1 0 1
+	0 0 0 0 0 1
Sum	0 0 0 1 1 0

   Two's Complement: 00110

Carry Bits	1 1 1 1 0
	1 1 1 1 1 1
+	0 0 0 1 1 0
Sum	0 0 0 1 0 1

   Answer = 000101

   The signed binary operands have different sign bits. Therefore, we don't need to check for overflow.

5. 01111111 – 1010010

One's Complement: 1010010 => 0101101

```
Carry Bits        0 0 0 0 1
                  0 1 0 1 1 0 1
              +   0 0 0 0 0 0 1
Sum               0 1 0 1 1 1 0
```

Two's Complement: 0101110

```
Carry Bits        0 1 1 1 1 1 0
                  0 1 1 1 1 1 1
              +     0 1 0 1 1 1 0
Sum               0 1 1 0 1 1 0 1
```

Answer = 01101101

Both signed binary operands are positive. However, the answer is negative. There is overflow. A positive sign bit was prepended to the final answer to indicate that is positive.

6. 10000000 – 111111111

One's Complement: 111111111 => 000000000

```
Carry Bits        0 0 0 0 0 0 0 0
                  0 0 0 0 0 0 0 0 0
              +   0 0 0 0 0 0 0 0 1
Sum               0 0 0 0 0 0 0 0 1
```

Two's Complement: 000000001

```
Carry Bits        0 0 0 0 0 0 0 0
                  1 1 0 0 0 0 0 0 0
              +   0 0 0 0 0 0 0 0 1
Sum               1 1 0 0 0 0 0 0 1
```

Answer = 110000001

The signed binary operands have different sign bits. Therefore, we don't need to check for overflow.

7. 000000000101 – 110

   One's Complement: 110 => 001

Carry Bits	0 1
	0 0 1
+	0 0 1
Sum	0 1 0

   Two's Complement: 010

Carry Bits	0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 0 0 0 0 0 1 0 1
+	0 0 0 0 0 0 0 0 0 0 1 0
Sum	0 0 0 0 0 0 0 0 0 1 1 1

   Answer = 000000000111

   Both signed binary operands are positive. The final answer is also positive. There is no overflow.

8. 111111111111 – 10101

   One's Complement: 10101 => 01010

Carry Bits	0 0 0 0
	0 1 0 1 0
+	0 0 0 0 1
Sum	0 1 0 1 1

   Two's Complement: 01011

Carry Bits	1 1 1 1 1 1 1 1 1 1 1
	1 1 1 1 1 1 1 1 1 1 1 1
+	0 0 0 0 0 0 0 0 1 0 1 1
Sum	0 0 0 0 0 0 0 0 1 0 1 0

   The signed binary operands have different sign bits. Therefore, we don't need to check for overflow.

9. 101010 – 111

One's Complement: 111 => 000

Carry Bits	0 0
	0 0 0
+	0 0 1
Sum	0 0 1

Two's Complement: 001

Carry Bits	0 0 0 0 0
	1 0 1 0 1 0
+	0 0 0 0 0 1
Sum	1 0 1 0 1 1

Answer = 101011

The signed binary operands have different sign bits. Therefore, we don't need to check for overflow.

10. 0101 – 1111001

One's Complement: 1111001 => 0000110

Carry Bits	0 0 0 0 0 0
	0 0 0 0 1 1 0
+	0 0 0 0 0 0 1
Sum	0 0 0 0 1 1 1

Two's Complement: 0000111

Carry Bits	0 0 0 1 1 1
	0 0 0 0 1 0 1
+	0 0 0 0 1 1 1
Sum	0 0 0 1 1 0 0

Answer = 0001100

Both signed binary operands are positive. The final answer is also positive. There is no overflow.