Homework 2 Solution

1. a) $X10 = (CFFF\_FFFF)_{16} = (1100\_1111\_1111\_1111\_1111\_1111\_1111\_1111)_2$

$X11 = (C000\_0000)_{16} = + (1100\_0000\_0000\_0000\_0000\_0000\_0000\_0000)_2$

$X9 = (8FFF\_FFFF)_{16} = (1000\_1111\_1111\_1111\_1111\_1111\_1111\_1111)_2$

We are using ADD instruction, so operands are signed numbers. As we are adding two negative numbers and result is also negative (sign bit not changed), this means **no overflow** occurred.

b) $X10 = (CFFF\_FFFF)_{16} = (1100\_1111\_1111\_1111\_1111\_1111\_1111\_1111)_2$

$X11 = (A000\_0000)_{16} = + (1010\_0000\_0000\_0000\_0000\_0000\_0000\_0000)_2$

$X9 = (6FFF\_FFFF)_{16} = (0110\_1111\_1111\_1111\_1111\_1111\_1111\_1111)_2$

As we are adding two negative numbers and result is positive (sign bit changed), this means **overflow** occurred.

2. $s0 = (0000\_0030)_{16} = (0000\_0000\_0000\_0000\_0000\_0000\_0011\_0000)_2$

$s1 = (0000\_0005)_{16} = (0000\_0000\_0000\_0000\_0000\_0000\_0000\_0101)_2$

We will ignore the 16 least significant 0 bits in $s1, and later shift the product accordingly.

```
    0000_0000_0000_0000_0000_0000_0011_0000
                                      X 0101
-----------------------------------------------
      00000000000000000000000000110000
     00000000000000000000000000000000
  + 00000000000000000000000000110000
-----------------------------------------------
    00000000000000000000000011110000
```

Adding 30 0s at the left gives us the result:

$(0000\_0000\_0000\_0000\_0000\_0000\_0000\_0000\_0000\_0000\_0000\_1111\_0000)_2$

$= (0000\_0000\_0000\_000F0)_{16}$

3. $\$s0 = (0000\_0008)_{16} = (0000\_0000\_0000\_0000\_0000\_0000\_0000\_1000)_2$

$\$s1 = (0000\_0005)_{16} = (0000\_0000\_0000\_0000\_0000\_0000\_0000\_0101)_2$

$0000\ \ 0101)\ 0000\ 1000_2$ Quotient = 1

$-0000\ 0101_2$

$0000\ 0011_2$ <---- Remainder

$X9 = \text{Quotient} = (0000\_0001)_{16}$

4.

```
        MOV   X10, #0           #reset count to zero
Loop:
        ANDI  X11, X9, 1        #check if LSB is 1
        ADD   X10, X10, X11     #add LSB to count, so count is increased when LSB 1
        LSR   X9, X9, #1        #shift  number to right by 1, thus next digit becomes new LSB
        CMP X9, #0
        BGT Loop                #loop until given number becomes zero
```

5. Computes the volume of a cylinder.

Pseudocode:

X9 = height

X10 = radius

X11 = 3.14

X11 = 3.14 * X10 * X10 * X9 //volume = pi * radius^2 * height