Problem Set#1
Multiple Choice Test
Chapter 01.04 Binary Representation
COMPLETE SOLUTION SET

1. $(25)_{10} = (?)_2$
   (A) 100110
   (B) 10011
   (C) 11001
   (D) 110010

Solution
The correct answer is (C).

	Quotient	Remainder
25/2	12	1
12/2	6	0
6/2	3	0
3/2	1	1
1/2	0	1

The binary representation is therefore

$$(25)_{10} = (11001)_2$$

2. $(1101)_2 = (?)_{10}$

(A) 3
(B) 13
(C) 15
(D) 26

Solution

The correct answer is (B).

To convert from base 2 to base 10,

$$\begin{aligned}(1101)_2 &= 1\times 2^0 + 0\times 2^1 + 1\times 2^2 + 1\times 2^3 \\ &= 1 + 0 + 4 + 8 \\ &= 13\end{aligned}$$

3. $(25.375)_{10} = (?.?)_2$

(A) 100110.011
(B) 11001.011
(C) 10011.0011
(D) 10011.110

Solution

The correct answer is (B).

Integer portion:

	Quotient	Remainder
25/2	12	1
12/2	6	0
6/2	3	0
3/2	1	1
1/2	0	1

Decimal portion:

	Number	Number after decimal	Number before decimal
0.375×2	0.750	0.750	0
0.750×2	1.500	0.500	1
0.500×2	1.000	0	1

$$(25.375)_{10} = (11001.011)_2$$

4. Representing $\sqrt{2}$ in a fixed point register with 2 bits for the integer part and 3 bits for the fractional part gives a round off error of most nearly

(A) -0.085709
(B) 0.0392
(C) 0.1642
(D) 0.2892

Solution

The correct answer is (B).

$$\sqrt{2} = 1.4142$$

If two bits for the integer portion are used,

$$(1)_{10} = (01)_2 = (1)_2$$

Decimal portion:

	Number	Number after decimal	Number before decimal
0.4142×2	0.8284	0.8284	0
0.8284×2	1.6568	0.6568	1
0.6568×2	1.3136	0.3136	1

Since only three bits are allowed,

$$(0.4142)_{10} \approx (0.011)_2$$

Total approximation

$$(1.4142)_{10} \approx (1.011)_2$$

Converting $(1.011)_2$ back to base 10 yields

$$(1.011)_2 = 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$
$$= (1.375)_{10}$$

Subtracting gives the round-off error as

$$1.4142 - 1.3750 = 0.0392$$

5. An engineer working for the Department of Defense is writing a program that transfers non-negative real numbers to integer format. To avoid overflow problems, the maximum non-negative integer that can be represented in a 5-bit integer word is

(A) 16
(B) 31
(C) 63
(D) 64

Solution
The correct answer is (B).

The maximum non-negative integer that can be represented using 5 bits is given by

$$\begin{aligned}(11111)_2 &= \left(1\times 2^0 + 1\times 2^1 + 1\times 2^2 + 1\times 2^3 + 1\times 2^4\right)\\ &= 1+2+4+8+16\\ &= 31\end{aligned}$$

6. For a numerically controlled machine, integers need to be stored in a memory location. The minimum number of bits needed for an integer word to represent all integers between 0 and 1024 is

(A) 8
(B) 9
(C) 10
(D) 11

Solution

The correct answer is (D).

If you have

- 1 bit for representing integers, the maximum integer that can be represented is 1;
- 2 bits for representing integers, the maximum integer that can be represented is 3;
- 3 bits for representing integers, the maximum integer that can be represented is 7;
- 4 bits for representing integers, the maximum integer that can be represented is 15;
- 5 bits for representing integers, the maximum integer that can be represented is 31;

and so on.

If you notice the trend, if *n* bits are available to represent integers, the maximum integer that can be represented is $2^n - 1$.

For example, if you have 4 bits for representing integers, the maximum integer that can be represented is $2^4 - 1 = 15$ as can be witnessed by

$$(1111)_2 = \left(1\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0\right) = 15$$

So what would it take to represent integers up to 1024.

$$2^n - 1 = 1024$$
$$2^n = 1025$$
$$\ln(2^n) = \ln(1025)$$
$$n\ln(2) = \ln(1025)$$
$$n\times 0.6931 = 6.932$$
$$n = 10.000762$$

Since *n* is an integer larger than 10.000762,

n=11.