V 1.0 9th – May – 2020

Universidad de León
Bachelor Degree on Computer Science and Engineering
Course on Computer Networks

Reference Solution to WH_2

Exercises

1. **Assume the integer subset {0, 1}. Demonstrate that the one-bit addition** and subtraction operations are the same.

   The truth tables to each operation follows:

X	Y	+	Carry	X	Y	-	Borrow
0	0	**0**	0	0	0	**0**	**0**
0	1	**1**	0	0	1	**1**	**0**
1	0	**1**	0	1	0	**1**	**1**
1	1	**0**	1	1	1	**0**	**0**

   Consequently, as far as the bit resulting from the operation is concerned, both operations are the same; otherwise, the 1-bit precision addition and the 1-bit precision subtraction, are the same. The difference between them comes when the carry and the borrow are considered, which are obviously distinct, as one should expect.

2. **Assume a generator polynomial C(x) which order is k. C(x) is used by** the CRC circuit of a NIC. Explain the different *numbers* of errors that can be detected by the circuit.

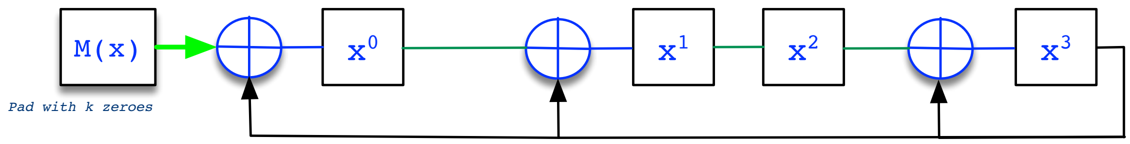
   A convenient choice of the terms present in C(x) can enable it to detect:
   · All single-bit errors
   · All double-bit errors
   · Any odd number of errors
   · Any burst error with length less than k

3. **Consider the preceding CRC circuit, again. How many errors can the** circuit correct?

   None. The CRC is a cyclic error-detection code, only. It has no error-correction capability.

4. **Consider the generator polynomial C(x) given by the bit-vector** denoted in decreasing order, from left-to-right: (1,1,0,1,1). Check the resulting CRC circuit by feeding the following data bit-vector, again in decreasing order, from left-to-right: (1, 1, 0, 0, 1, 0).

Generator polynomial: $C(x) = 1\cdot x^0 + 1\cdot x^1 + 0\cdot x^2 + 1\cdot x^3 + 1\cdot x^4$



M(x)	x^0	x^1	x^2	x^3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
0	0	1	1	0
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	0	1
0	1	0	1	1
–	**1**	**0**	**0**	**0**

5. **Continuing from the preceding exercise, assume that the data bit-**vector is sent to the receiver along with the calculated CRC (Data + Redundancy). The receiver must check the CRC for errors. The CRC-checking procedure applied by the receiver consists of:

    a. Receive M(x), in our case the data bit vector which size is 6 bits. Store M(x).

    We arrange the bits from M(x) in a column each of which bits will be feed in to the circuit as it evolves from each row to the next one.

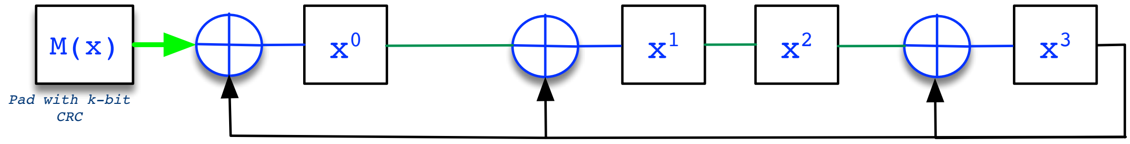
    b. Receive C(x), the CRC computed by the sender and store it.

    As we did in the preceding question, C(x) is represented by having a xor gate's output be (Coefficient 1 of term) connected to the respective term's one-bit register or having the term's register directly connected to its preceding one-bit register.

    c. Calculate the CRC by applying the procedure explained in the WebConference-lecture of 26th-March, taking into account, though, that instead of padding M(x) (The data polynomial) with as many zeroes as the order of C(x), you will have to pad the M(x) column with the bits from the received CRC. Pay attention not to invert the order of the CRC bits as you use each of them to pad M(x). If no error took place, then the new CRC that you are computing should yield all zeroes, *i.e.*, your CRC should be equal to (0,0,0,0) in this case.[1]

Search the documents in paloalto.unileon.es/cn for an exercise similar to this.

Generator polynomial: $C(x) = 1{\cdot}x^0 + 1{\cdot}x^1 + 0{\cdot}x^2 + 1{\cdot}x^3 + 1{\cdot}x^4$



M(x)	x^0	x^1	x^2	x^3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
0	0	1	1	0
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	1	0	1
1	1	0	1	1
–	**0**	**0**	**0**	**0**

In this case, the CRC = (0,0,0,0) which means that no error was *detected*.

[1] Since the order of M(x) = 4, the order of the resulting CRC polynomial (The remainder of the integer division) must be = 4 -1 = 3; consequently, the resulting CRC should be equal to (0,0,0,0).