CRC 3

Generating CRC

1. Calculate the CRC, and the final message using: Message = 911_{16}, Divisor = $X^3 + X^2 + 1$

   Message:
   First, we need to convert the message (dividend) in hexadecimal to binary.
   So $911_{16} = 1001\ 0001\ 0001_2$

   Polynomial:
   Second, we need to convert the polynomial (divisor) into a binary form.
   So $X^3 + X^2 + 1$ is the same as $X^3 + X^2 + 0X^1 + 1$
   And so $X^3 + X^2 + 0X^1 + 1 = 1101$

   Now what?
   Now that we have the binary representations of both the message and the divisor, we can go ahead and generate the CRC! Because this is a 3-bit CRC, you will need to pad the message with 3 zeroes at the end of the message.

   You can do the full calculation bit by bit or do faster in the way shown on Question 2.

					1	1	1	1	0	1	0	1	0	0	1	0			
1	1	0	1		1	0	0	1	0	0	0	1	0	0	0	1	**0**	**0**	**0**
					1	1	0	1											
						1	0	0	0										
						1	1	0	1										
							1	0	1	0									
							1	1	0	1									
								1	1	1									
								1	1	0	1								
									0	1	1	1							
										1	1	0	1						
											0	1	1	0					
											0	0	0	0					
												1	1	0	0				
												1	1	0	1				
													0	0	1	1			
													0	0	0	0			
														0	1	1	0		
														0	0	0	0		
															1	1	0	0	
															1	1	0	1	
																0	0	1	0
																0	0	0	0
																	0	**1**	**0**

Remainder:

Your remainder is always 1 bit less than the polynomial, so in this case 3 bits = 010.

Appended Message:

1001 0001 0001 010

Now you should convert the remainder to a hexadecimal representation. First, start with padding it with a leading zero to make it 4 bits = 0010.

Now convert it to hexadecimal, and append it to the message.

$0010_2 = 2_{16}$

9112_{16} and that's it!

2. Calculate the CRC, and the final message using: Message = $A0E_{16}$, Divisor = $X^3 + X + 1$

   Message:
   First, we need to convert the message (dividend) in hexadecimal to binary.
   So $A0E_{16} = 1010\ 0000\ 1110_2$

   Polynomial:
   Second, we need to convert the polynomial (divisor) into a binary form.
   So $X^3 + X + 1$ is the same as $X^3 + 0X^2 + X^1 + 1$
   And so $X^3 + 0X^2 + X^1 + 1 = 1011$

   Now what?
   Now that we have the binary representations of both the message and the divisor, we can go ahead and generate the CRC! Because this is a 3-bit CRC, you will need to pad the message with 3 zeroes at the end of the message.

   As mentioned in Question 1, this method is quicker. Line up the leftmost '1' values, then apply XOR to subtract.

					1			1		1	1	1	1	1	1				
1	0	1	1		1	0	1	0	0	0	0	0	1	1	1	0	**0**	**0**	**0**
					1	0	1	1											
						0	0	1	0										
								1	0	1	1								
									0	1	1	0							
										1	0	1	1						
											1	1	0	1					
											1	0	1	1					
												1	1	0	1				
												1	0	1	1				
													1	1	0	0			
													1	0	1	1			
														1	1	1	0		
														1	0	1	1		
															1	0	1	0	
															1	0	1	1	
																	0	**1**	**0**

Remainder:

The remainder is 010

Appended Message:

1010 0000 1110 010

$0010_2 = 2_{16}$

$\mathbf{A0E2}_{16}$

Validating CRC

3. Validate the CRC using the result from Question 1.

   Appended message:
   1001 0001 0001 010

   But wait, you're thinking it should be 1001 0001 0001 0010. Remember however, that we padded the remainder 010 with a leading zero, so do not use the padded remainder, but the true remainder which should be 3 bits!

Polynomial:
Use the same polynomial = 1101

Now What?
Now that we have everything, do the modulo-2 division.

					1	1	1	1		1		1			1				
1	1	0	1		1	0	0	1	0	0	0	1	0	0	0	1	**0**	**1**	**0**
					1	1	0	1											
						1	0	0	0										
						1	1	0	1										
							1	0	1	0									
							1	1	0	1									
								1	1	1	0								
								1	1	0	1								
										1	1	1	0						
										1	1	0	1						
												1	1	0	0				
												1	1	0	1				
															1	1	0	1	
															1	1	0	1	
																	0	**0**	**0**

The 000 is the “syndrome” (again, fancy word for remainder in the phase of CRC validation). If your syndrome is all zeroes, that means no change has occurred in your original message. If your syndrome non-zero, that means there was a change in your original message.

Computers as you may already know, use 0’s and 1’s to communicate. The generated CRC remainder that is attached to the original message serves the purpose of checking the integrity of the original message. Because communication mediums are not perfect, there are cases where the message 1001 0001 0001 010 may change to something like 1001 0000 0000 010.

When you perform the validation on the 1001 0000 0000 010, you will know that there was a change to your original message, because your syndrome will NOT be all zeros. A computer will reject this corrupted message, and ask the sender to re-transmit.

4. Validate the CRC using the result from Question 2.

Appended message:
1010 0000 1110 010

Remember we want the 3 bit remainder. We only padded it with a leading zero for the sole purpose of converting it to a hexadecimal.

Polynomial:
Use the same polynomial = 1101

Now What?
Now that we have everything, do the modulo-2 division.

					1			1		1	1	1	1	1	1				
1	0	1	1		1	0	1	0	0	0	0	0	1	1	1	0	**0**	**1**	**0**
					1	0	1	1											
								1	0	0	0								
								1	0	1	1								
										1	1	0	1						
										1	0	1	1						
											1	1	0	1					
											1	0	1	1					
												1	1	0	1				
												1	0	1	1				
													1	1	0	0			
													1	0	1	1			
														1	1	1	0		
														1	0	1	1		
															1	0	1	1	
															1	0	1	1	
																	0	**0**	**0**

The 000 is the "syndrome" (again, fancy word for remainder in the phase of CRC validation). If your syndrome is all zeroes, that means no change has occurred in your original message. If your syndrome non-zero, that means there was a change in your original message.

CRC 4

Generating a CRC

5. Calculate the CRC, and the final message using: Message = $A0E_{16}$, Divisor = $X^4 + X^3 + 1$

Message:

First, we need to convert the message (dividend) in hexadecimal to binary.
So $A0E_{16} = 1010\ 0000\ 1110_2$

Polynomial:
Second, we need to convert the polynomial (divisor) into a binary form.
So $X^4 + X^3 + 1$ is the same as $X^4 + X^3 + 0X^2 + 0X^1 + 1$
And so $X^4 + X^3 + 0X^2 + 0X^1 + 1 = 11001$

Now what?
Now that we have the binary representations of both the message and the divisor, we can go ahead and generate the CRC! Because this is a 4-bit CRC, you will need to pad the message with 4 zeroes at the end of the message.

						1	1			1					1						
1	1	0	0	1		1	0	1	0	0	0	0	0	1	1	1	0	**0**	**0**	**0**	**0**
						1	1	0	0	1											
							1	1	0	1	0										
							1	1	0	0	1										
										1	1	0	0	1							
										1	1	0	0	1							
															1	1	0	0	0		
															1	1	0	0	1		
																		0	**1**	**0**	**0**

Remainder:

Your remainder will be 0100.

Appended Message:

1010 0000 1110 0100

Now convert it to hexadecimal, and append it to the message.

$0100_2 = 4_{16}$

$A0E4_{16}$ and that's it!

6. Calculate the CRC, and the final message using: Message = ABC_{16}, Divisor = $X^4 + X^3 + X + 1$

Message:
First, we need to convert the message (dividend) in hexadecimal to binary.
So $ABC_{16} = 1010\ 1011\ 1100_2$

Polynomial:

Second, we need to convert the polynomial (divisor) into a binary form.
So $X^4 + X^3 + X + 1$ is the same as $X^4 + X^3 + 0X^2 + X + 1$
And so $X^4 + X^3 + 0X^2 + X + 1 = 11011$

Now what?
Now that we have the binary representations of both the message and the divisor, we can go ahead and generate the CRC! Because this is a 4-bit CRC, you will need to pad the message with 4 zeroes at the end of the message.

						1	1		1		1	1	1	1							
1	1	0	1	1		1	0	1	0	1	0	1	1	1	1	0	0	**0**	**0**	**0**	**0**
						1	1	0	1	1											
							1	1	1	0	0										
							1	1	0	1	1										
									1	1	1	1	1								
									1	1	0	1	1								
											1	0	0	1	1						
											1	1	0	1	1						
												1	0	0	0	0					
												1	1	0	1	1					
													1	0	1	1	0				
													1	1	0	1	1				
														1	1	0	1	0			
														1	1	0	1	1			
																		1	**0**	**0**	**0**

Remainder:

Your remainder will be 1000.

Appended Message:

1010 1011 1100 1000

Now convert it to hexadecimal, and append it to the message.

$1000_2 = 8_{16}$

$ABC8_{16}$ and that's it!

Validating a CRC

7. Validate the CRC using the result from Question 5.

Appended message:

1010 0000 1110 0100

The CRC remainder is 4 bits = 0100.

Polynomial:
Use the same polynomial = 11001

Now What?
Now that we have everything, do the modulo-2 division.

						1	1			1					1						
1	1	0	0	1		1	0	1	0	0	0	0	0	1	1	1	0	**0**	**1**	**0**	**0**
						1	1	0	0	1											
							1	1	0	1	0										
							1	1	0	0	1										
										1	1	0	0	1							
										1	1	0	0	1							
															1	1	0	0	1		
															1	1	0	0	1		
																		0	**0**	**0**	**0**

The 0000 is the "syndrome" (again, fancy word for remainder in the phase of CRC validation). If your syndrome is all zeroes, that means no change has occurred in your original message. If your syndrome non-zero, that means there was a change in your original message.

8. Validate the CRC using the result from Question 6.

   Appended message:
   1010 1011 1100 1000

   The CRC remainder is 4 bits = 1000

   Polynomial:
   Use the same polynomial = 11011

   Now What?
   Now that we have everything, do the modulo-2 division.

						1	1		1		1	1	1	1							
1	1	0	1	1		1	0	1	0	1	0	1	1	1	1	0	0	**1**	**0**	**0**	**0**
						1	1	0	1	1											
							1	1	1	0	0										
							1	1	0	1	1										
									1	1	1	1	1								
									1	1	0	1	1								
											1	0	0	1	1						
											1	1	0	1	1						
												1	0	0	0	0					
												1	1	0	1	1					
													1	0	1	1	0				
													1	1	0	1	1				
														1	1	0	1	1			
														1	1	0	1	1			
																		0	**0**	**0**	**0**

The 0000 is the “syndrome” (again, fancy word for remainder in the phase of CRC validation). If your syndrome is all zeroes, that means no change has occurred in your original message. If your syndrome is non-zero, that means there was a change in your original message.