quiz 05.1 Solution

(1) Compute the CRC of the message 10001110101000 using generating polynomial with coefficients (from highest to lowest, left to right) 1001111.

That means to treat both binary strings as the list of coefficients of polynomials, from highest-order term to lowest, with coefficients lying in the finite field F_2, and divide-with-remainder. The bit string 10001110101000 of the message becomes the *dividend* $x^{13} + x^9 + x^8 + x^7 + x^5 + x^3$, and the CRC bit string 1001111 becomes the generating polynomial $x^6 + x^3 + x^2 + x + 1$ which will be the *divisor*. The remainder, (re-expressed, if desired, as a binary string rather than as a polynomial) will be the *value* of the cyclic redundancy code for this message with this generator. The polynomial division is

	$+x^7$	$+x^4$			$+x^1$	$+x^0$	R	$+x^3$			$+x^0$	
$+x^6$ $+x^3$ $+x^2$ $+x^1$ $+x^0$	$+x^{13}$		$+x^9$	$+x^8$	$+x^7$		$+x^5$		$+x^3$			
	$+x^{13}$	$+x^{10}$	$+x^9$	$+x^8$	$+x^7$							
		$+x^{10}$					$+x^5$		$+x^3$			
		$+x^{10}$			$+x^7$	$+x^6$	$+x^5$	$+x^4$				
					$+x^7$	$+x^6$		$+x^4$	$+x^3$			
					$+x^7$			$+x^4$	$+x^3$	$+x^2$	$+x^1$	
						$+x^6$				$+x^2$	$+x^1$	
						$+x^6$			$+x^3$	$+x^2$	$+x^1$	$+x^0$
									$+x^3$			$+x^0$

Thus, the CRC value is $x^3 + 1$, which gives the bit string 001001 (padding by 0's in front to make it of length equal to the degree of the generating polynomial).

(2) Find a 2-bit error that could occur in the message 01110101010100 that would be undetected by the CRC with generating polynomial coefficients (from highest to lowest) 10111.

The specific message 01110101010100 doesn't matter, except that it be long enough to have two bit-errors separated far enough so that the CRC fails to detect those two bit-errors. As observed both in class and in the notes, since the constant coefficient of the generator is non-zero, a failure to detect two bit-errors depends only upon their distance apart in the message, not their locations. And for the CRC with generator 10111 to fail to detect a pair of bit errors a distance n apart, it is necessary and sufficient that the generating polynomial $x^4 + x^2 + x + 1$ (made from 10111) divide $x^n - 1$ with remainder 0. Since the degree of the generator is very small, it is reasonable to test whether 10111 divides $x^t - 1$ starting with $t = 4, 4+1, 4+2$, and so on. The remainder upon dividing $x^4 + 1$ by $x^4 + x^2 + x + 1$ is $x^2 + x$ which is non-zero. So try again with x^5+1. The remainder upon dividing x^5+1 by x^4+x^2+x+1 is x^3+x^2+x+1 which is non-zero. So try again with x^6+1. The remainder upon dividing x^6+1 by x^4+x^2+x+1 is x^3+x which is non-zero. So try again with x^7+1. Finally, the remainder upon dividing x^7+1 by x^4+x^2+x+1 is 0. That is, any two single-bit errors a distance of 7 apart will *not* be detected by this CRC.

(3) Verify that the binary string 1011011 (interpreted as a polynomial, powers going from highest to lowest, coefficients in $\mathbf{F}_2$) is **primitive**. (In particular, this means that the CRC made using this as generating polynomial can detect *any* two bit-errors distance less than $2^6 - 1$ apart.)

An efficient way to check this is to check that $x^{2^6} \,\%\, x^6 + x^4 + x^3 + x + 1 = x$, and that for any prime p dividing $63 = 2^6 - 1$

$$x^{(2^6-1)/p} \,\%\, x^6 + x^4 + x^3 + x + 1 \neq 1$$

(These exponentiations are best done by the fast exponentiation algorithm.) The primes dividing 63 are 3 and 7 (by trial division, for example), so we must verify that $x^{63/3} \neq 1$ and $x^{63/7} \neq 1$. We find that indeed $x^{64} \,\%\, x^6 + x^4 + x^3 + x + 1 = x$ (not shown). Let's compute

$$x^{21} \,\%\, x^6 + x^4 + x^3 + x + 1 = x^3 + x^2 + x \neq 1$$

in detail: Initialize $(X, E, Y) = (x, 21, 1)$. E=21 is odd, so multiply Y by X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$, and subtract 1 from E, giving

$$(X, E, Y) = (x, 20, x)$$

E=20 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^2, 10, x)$$

E=10 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^4, 5, x)$$

E=5 is odd, so multiply Y by X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$, and subtract 1 from E, giving

$$(X, E, Y) = (x^4, 4, x^5)$$

E=4 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^5 + x^4 + x^2 + x + 1, 2, x^5)$$

E=2 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^4 + x + 1, 1, x^5)$$

E=1 is odd, so multiply Y by X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$, and subtract 1 from E, giving

$$(X, E, Y) = (x^4 + x + 1, 0, x^3 + x^2 + x)$$

Now E is 0, so $Y = x^3 + x^2 + x$ is the desired $x^{21} \,\%\, x^6 + x^4 + x^3 + x + 1$. . And also compute in detail

$$x^9 \,\%\, x^6 + x^4 + x^3 + x + 1 = x^5 + x^4 + x^2 + 1 \neq 1$$

in detail: Initialize $(X, E, Y) = (x, 9, 1)$. E=9 is odd, so multiply Y by X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$, and subtract 1 from E, giving

$$(X, E, Y) = (x, 8, x)$$

E=8 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^2, 4, x)$$

E=4 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^4, 2, x)$$

E=2 is even, so square X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$ and divide E by 2, giving

$$(X, E, Y) = (x^5 + x^4 + x^2 + x + 1, 1, x)$$

E=1 is odd, so multiply Y by X and reduce mod $m = x^6 + x^4 + x^3 + x + 1$, and subtract 1 from E, giving

$$(X, E, Y) = (x^5 + x^4 + x^2 + x + 1, 0, x^5 + x^4 + x^2 + 1)$$

Now E is 0, so $Y = x^5 + x^4 + x^2 + 1$ is the desired $x^9 \,\%\, x^6 + x^4 + x^3 + x + 1$. . Thus, $x^6 + x^4 + x^3 + x + 1$ is primitive.

(4) With codewords 01001, 10101, 11010 and channel bit-error probability 1/14, give a good estimate for the maximum (over words) expected number of uncorrectible bit errors.

Observing that the minimum distance (between any two of the codewords) is $3 = 2 \cdot 1 + 1$, we realize that any single-bit error can be corrected. So we are being conservative (in a good sense) if for simplicity we do not attempt to correct any received word that is more than 1 bit away from a codeword. Since the probabilities of such multi-bit errors are much smaller than the probability of a single-bit error, we are not sacrificing too much precision in our estimate. Thus, the (maximum) expected number of uncorrectible bit errors in a single transmitted word is *less than*

$$\sum_{1<x\leq 5} x \cdot P(\text{ exactly } x \text{ errors}) = 2 \cdot P(\text{2-bit error}) + 3 \cdot P(\text{3-bit error}) + 4 \cdot P(\text{4-bit error}) + 5 \cdot P(\text{5-bit error})$$

$$= 2 \cdot \binom{5}{2} (1/14)^2 (1-1/14)^3 + 3 \cdot \binom{5}{3} (1/14)^3 (1-1/14)^2 + 4 \cdot \binom{5}{4} (1/14)^4 (1-1/14)^1 + 5 \cdot \binom{5}{5} (1/14)^5 (1-1/14)^0$$

$$\approx 2 \cdot 10 \cdot 0.00408 + 3 \cdot 10 \cdot 0.000314 + 4 \cdot 5 \cdot 0.0000241 + 5 \cdot 1 \cdot 0.00000185 \approx 0.0915$$