University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

EECS150 J. Wawrzynek
Spring 2000

Homework #10

This homework is due on **Friday April 21th by 11am**. Homework will be accepted in the EECS150 box on the door to room 218 Cory Hall. Late homework will be penalized by 50%. No late homework will be accepted after the solution is handed out.

From the **Katz book**:

Simple-circuits with Feedback 6.1, 6.2

Flip-flops 6.14, 6.15, 6.16, 6.17, 6.18, 6.19

Error corrections circuits

This exercise is designed to convince you the hardware for CRC Checksums presented in slide 21 (lecture 4/13/00) really works. We'll be using small numbers in our example.

a) Following the example on slide 19, do this long division by hand in modulo 2 arithmetic:

Dividend: Data 1010001101 concatenated with 00000
: 101000110100000

Divisor: 110101

The remainder is 01110. We're really sure this is correct, so if you can't get the right answer, you're doing something wrong! Look at the long-division example on slide 19 for guidance.

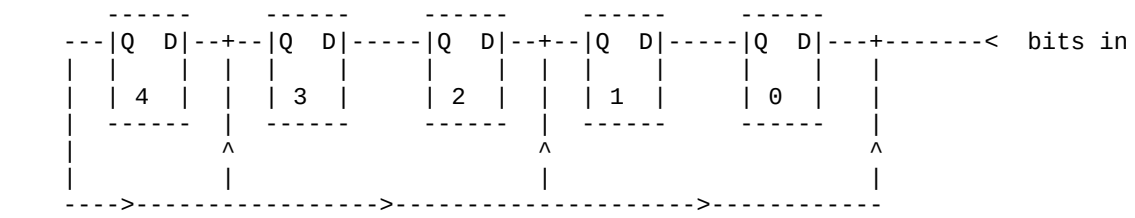
Then do this long division by hand in modulo 2 arithmetic:

Dividend: Data 1010001101 concatenated with checksum 01110
: 101000110101110

Divisor: 110101

Notice that the dividend is identical to the first dividend, but the last five zeros have been replaced with the calculated remained 01110. What should the result of this long division be? Do the long division and confirm your guess.

b) This Linear Feedback Shift Register implements the CRC code that uses 110101 as the dividend (which we used in a) above). This divisor has the special "cyclic" property needed for the shift register implementation in slide 21 to work.



Where the + is XOR, the data flows from right to left, all flip-flops start at 0 and share the same clock, etc.

Using this divisor:

Divisor: 101000110100000
^
first bit into the shift register

Here is a trace of the state of FF0 – FF4, for each clock of the flip-flops. The last clock cycle leaves 01110 in the flip-flops, as predicted by the long division we did.

T	Bit	FFFFF 43210
0		00000
1	1	00001
2	0	00010
3	1	00101
4	0	01010
5	0	10100
6	0	11101
7	1	01110
8	1	11101
9	0	01111
10	1	11111
11	0	01011
12	0	10110
13	0	11001
14	0	00111
15	0	01110

Using this divisor:

Divisor: 101000110101110

Show a trace of the state of FF0 – FF4, for each clock in to the divisor. The last clock cycle should leave 00000 in the register. Hint: you can reuse most of the trace above!