MOTOROLA Semiconductors
BOX 20912 • PHOENIX, ARIZONA 85036

MC8503P

BIPOLAR LSI

UNIVERSAL POLYNOMIAL GENERATOR

UNIVERSAL POLYNOMIAL GENERATOR (UPG)

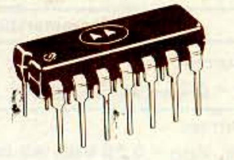
The MC8503 Universal Polynomial Generator (UPG) is used in serial digital data handing systems for error detection and correction. The serial data stream is divided by a selected polynomial and the division remainder is transmitted at the end of the data stream as a Cyclic Redundancy Check Character (CRCC). When the data is received the same calculation is performed. If there were no errors in transmission, the new remainder will be zero.

The MC8503 offers four of the more common polynominals for error detection techniques including a read forward and reverse on the CRCC-16 and CRCC-CCITT polynomial functions. These polynomials can be generated by changing the binary select codes as shown in Figure 1.

- Four Unique Polynomial Codes in One Package
- Compatible with TTL
  Maximum Fan-Out = 1 TTL Load
- Data Rate = 5 MHz Typical
- Total Power Dissipation = 400 mW Typical
- +5.0-Volt Operation

Typical Applications Include:

- Floppy Discs
- Cassettes
- Data Communications

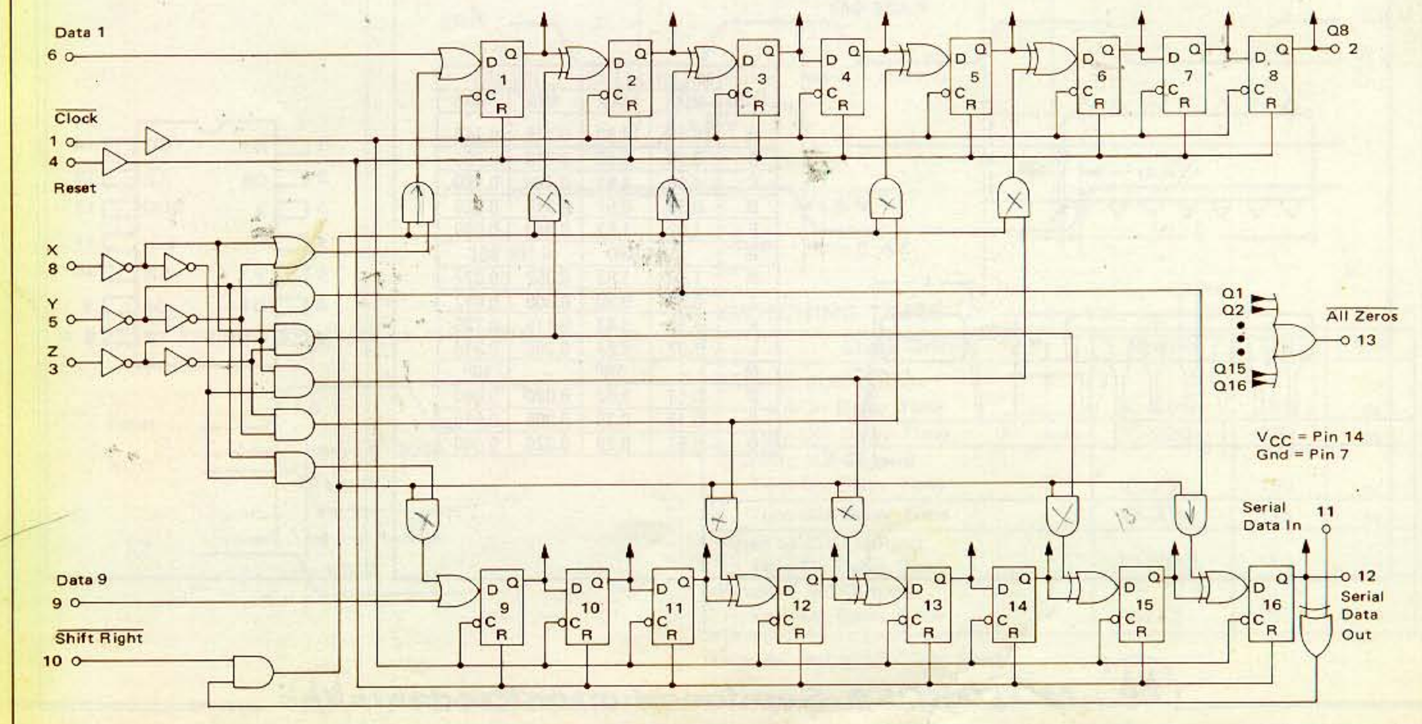


PLASTIC PACKAGE
CASE 646

FIGURE 1 – AVAILABLE POLYNOMIALS

CODE SELECT X	Y	Z		POLYNOMIAL
0	0	0	CRCC-16 (Fwd)	$X^{16} + X^{15} + X^2 + 1$
0	0	1	CRCC-16 (Bkwd)	$X^{16} + X^{14} + X + 1$
1	1	0	CRCC-CCITT (Fwd)	$X^{16} + X^{12} + X^5 + 1$
1	1	1	CRCC-CCITT (Bkwd)	$X^{16} + X^{11} + X^4 + 1$
0	1	0	LRCC-16	$X^{16} + 1$
1	0	1	LRCC-8	$X^8 + 1$

LOGIC DIAGRAM



MAXIMUM RATINGS

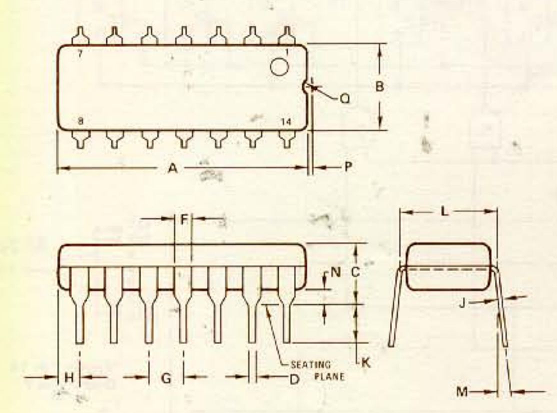
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.5 to +7.0	Vdc
Input Voltage	V_{in}	-1.0 to +5.5	Vdc
Output Voltage (Open collectors)	V_{OH}	-0.5 to +7.0	Vdc
Thermal Resistance	θ_{JA}	65	°C/W
Operating Temperature Range	T_A	-0 to +75	°C
Storage Temperature Range	T_{stg}	-55 to +165	°C

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, T_A = 0 to 75°C unless otherwise noted.)

Characteristic	Symbol	Min	Max	Unit
Input Forward Current (V_{IL} = 0, V_{CC} = 5.25 Vdc, All Inputs)	I_{IL}	–	-75	µAdc
Input Leakage Current (V_{IH} = 2.4 Vdc, V_{CC} = 5.25 Vdc, All Inputs)	I_{IH}	–	120	µAdc
Logic "0" Output Voltage (I_{OL} = 1.6 mAdc, V_{ILT} = 0.8 Vdc, V_{IHT} = 2.0 Vdc, V_{CC} = 4.75 Vdc)	V_{OL}	–	0.5	Vdc
Logic "1" Output Voltage (I_{OH} =-0.3 mAdc, V_{ILT} = 0.8 Vdc, V_{IHT} = 2.0 Vdc, V_{CC} = 4.75 Vdc)	V_{OH}	2.4	–	Vdc
Output Leakage Current (V_{CC} = V_{CEX} = 5.25 Vdc)	I_{CEX}	–	100	µAdc
Power Supply Drain Current (V_{CC} = 5.0 Vdc, All Inputs Open)	I_{CC}	–	117	mAdc

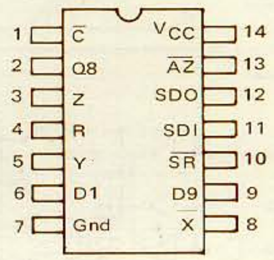
PACKAGE DIMENSIONS

PLASTIC PACKAGE
CASE 646

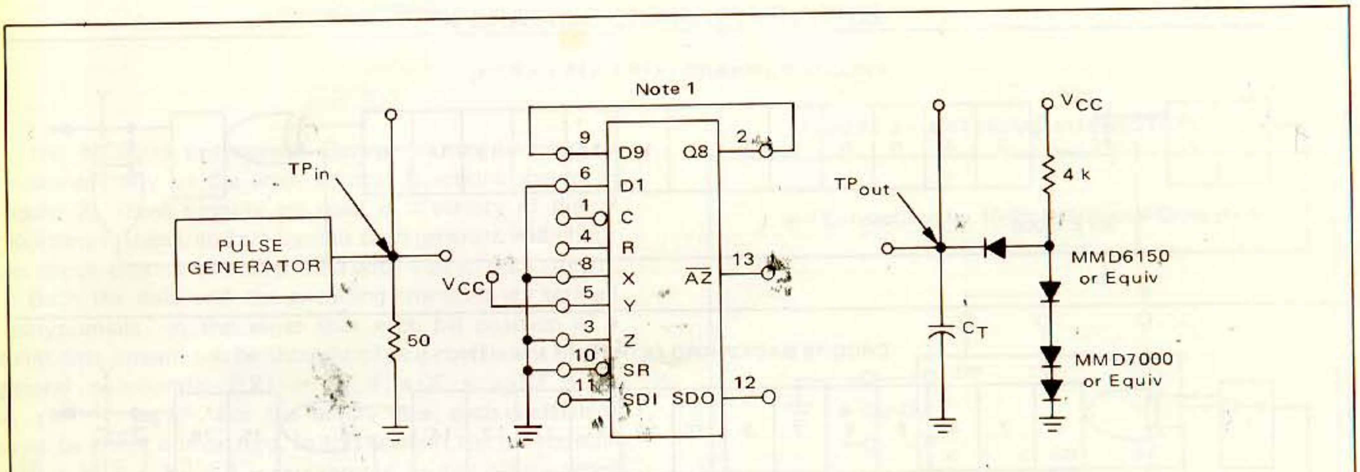


DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	18.16	18.80	0.715	0.740
B	6.10	6.60	0.240	0.260
C	4.06	4.57	0.160	0.180
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.32	1.83	0.052	0.072
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.37	7.87	0.290	0.310
M	–	10°	–	10°
N	0.51	1.02	0.020	0.040
P	0.13	0.38	0.005	0.015
Q	0.51	0.76	0.020	0.030

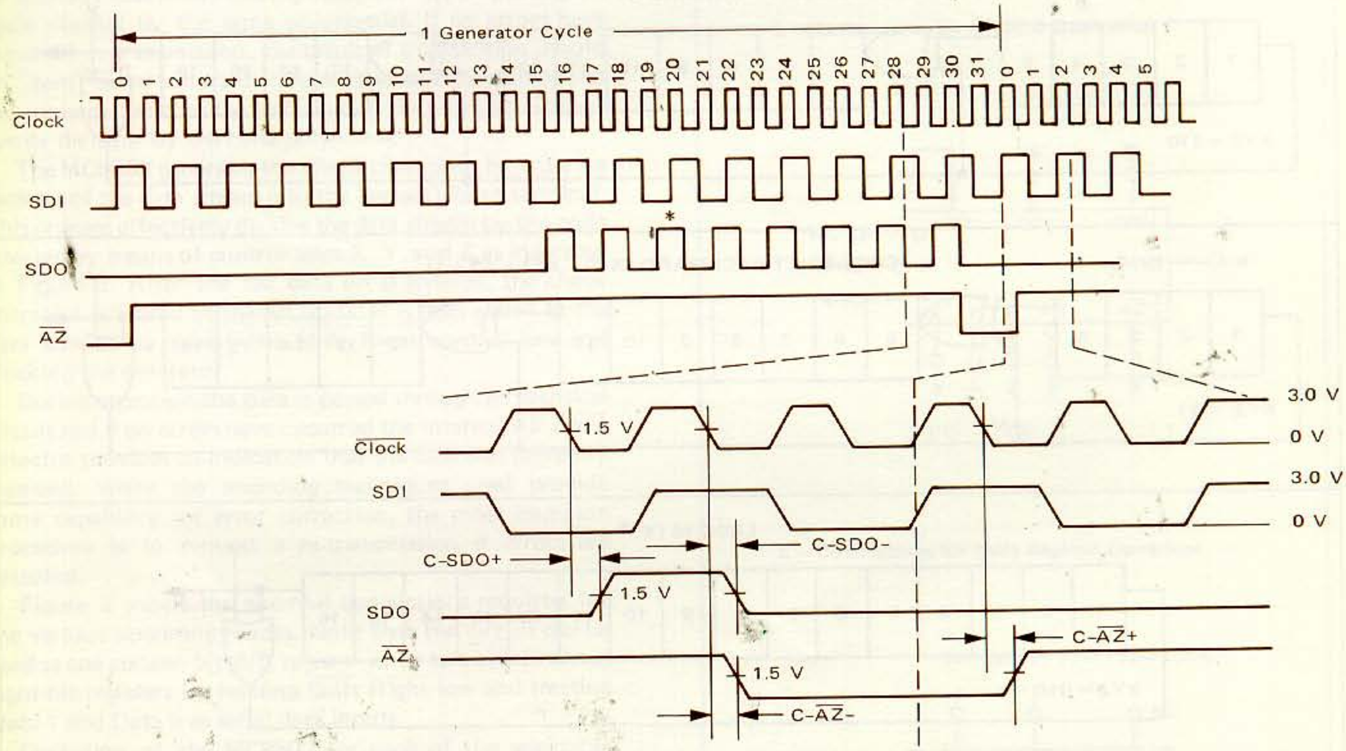
PIN ASSIGNMENT



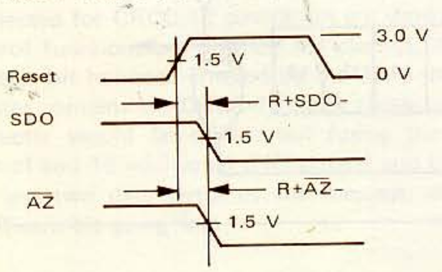
SWITCHING TIME TEST CIRCUIT AND WAVEFORMS



High-impedance probes (> 1.0 megohm) must be used.
C_T = 15 pF = total parasitic capacitance, which includes probe, wiring, and load capacitances.
Note 1: Use minimum wire length to connect Q8 to D9.



*RESET TEST TO BE MADE AT THIS POINT

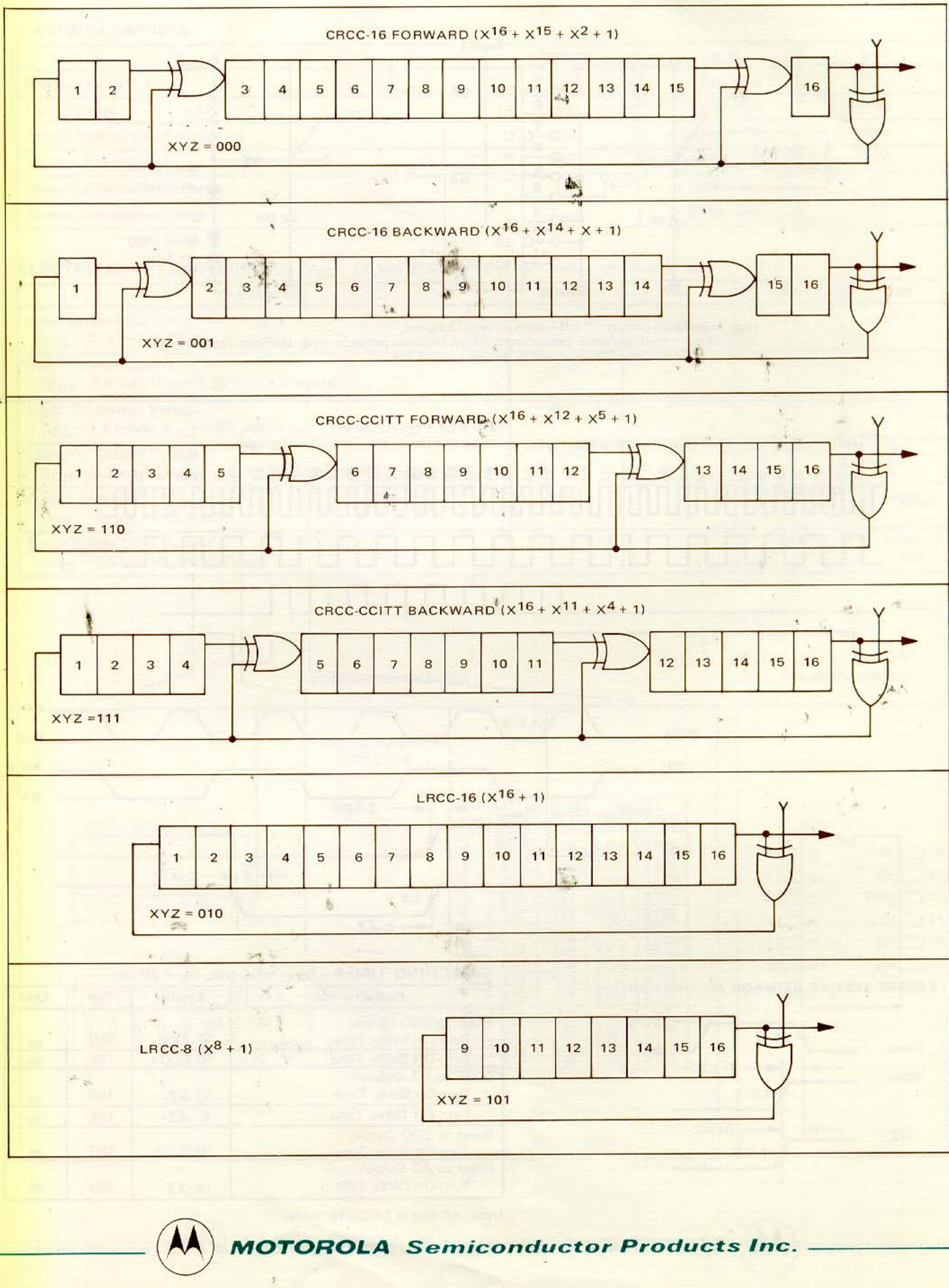


SWITCHING TIMES (V_{CC} = 5.0 Vdc, T_A = 25°C)

Characteristic	Symbol	Typ	Unit
Clock to SDO Output			
Turn-On Delay Time	$t_{\overline{C}\text{-SDO-}}$	150	ns
Turn-Off Delay Time	$t_{\overline{C}\text{-SDO+}}$	125	ns
Clock to $\overline{AZ}$ Output			
Turn-On Delay Time	$t_{\overline{C}\text{-}\overline{AZ}\text{-}}$	150	ns
Turn-Off Delay Time	$t_{\overline{C}\text{-}\overline{AZ}\text{+}}$	175	ns
Reset to SDO Output			
Turn-On Delay Time	$t_{R\text{+SDO-}}$	150	ns
Reset to $\overline{AZ}$ Output			
Turn-On Delay Time	$t_{R\text{+}\overline{AZ}\text{-}}$	150	ns

Note: AC test in LRCC-16 Mode.

FIGURE 2 – ERROR CONTROL FUNCTIONS



OPERATING CHARACTERISTICS

The MC8503 Polynomial Generator may be used to implement any of the error control functions shown in Figure 2. These circuits are used in a variety of digital recording systems, and are used to both generate and check the check characters transmitted with a serial data stream.

Both the data and the encoding character are termed "polynomials" in the sense that each bit position of a serial data stream can be thought of as a coefficient of the general polynomial $f(X) = a_0 + a_1X + a_2X^2 + \ldots a_{k-1}X^{k-1} + a_kX^k$. For the binary case, each coefficient must be either one or zero. In this context the polynomial $X^{16} + X^{15} + X^2 + 1$ is represented by the binary word 11000000000000101.

The check characters are generated by "dividing" the data by the encoding polynomial. The remainder resulting from the division is then appended to the message stream as a check character. During reception of the data it is again divided by the same polynomial. If no errors have occurred in transmission, the result of this division should be "zero" since adding the check character (remainder) to the message has the effect of making the received message evenly divisible by the code polynomial.

The MC8503 generates the check characters by entering each bit of the data stream into the Serial Data In terminal. This process effectively divides the data stream by the code selected by means of control lines X, Y, and Z as indicated in Figure 2. After the last data bit is entered, the check character is stored in the MC8503; it is then added to the data stream by taking the Shift Right control low and clocking the generator.

During reception the data is passed through an identical circuit and if no errors have occurred the internal All Zeros detector provides an indication that the data was correctly received. While the encoding techniques used provide some capability for error correction, the most common procedure is to request a re-transmission if errors are detected.

Figure 3 shows the external connections required for the various operating modes. Note that the circuit can be used as one sixteen-bit shift register, or as two synchronous eight-bit registers by holding Shift Right low and treating Data 1 and Data 9 as serial data inputs.

Operation of the MC8503 for each of the selectable codes is shown in Figures 4 thru 11. In Figure 4, the results of shifting a four byte data message into an MC8503 connected for CRCC-16 generation are shown (clock and control functions are omitted for clarity). After the last message bit has been entered by the 32nd shift pulse, the register contains the CRCC-16 check character. The check character would be shifted out (using the Shift Right control and 16 additional shift pulses) and transmitted as the last two data bytes of the message, with the least significant bit going first.

FIGURE 3 – EXTERNAL CONNECTIONS

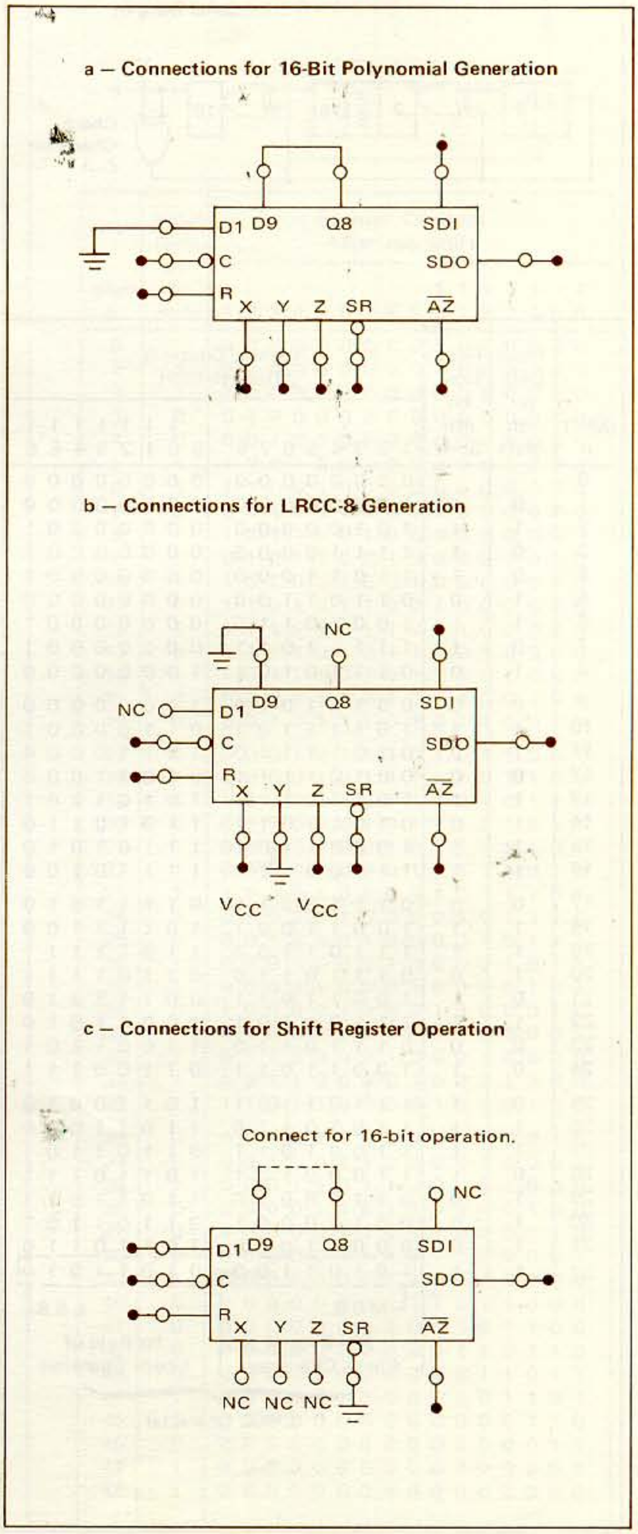
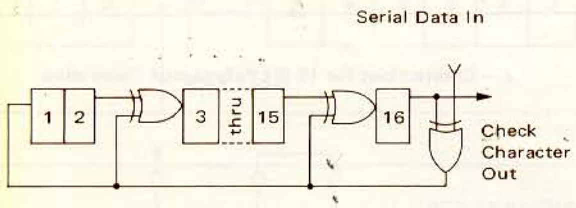


FIGURE 4 – GENERATION OF CRCC-16 ($X^{16} + X^{15} + X^2 + 1$) CHECK CHARACTER



Shift n	Input Data on nth Shift	Feedback on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	Register Contents After nth Shift: 9 10 11 12 13 14 15 16
0			0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
2	1	1	1 0 1 0 0 0 0 0	0 0 0 0 0 0 0 1
3	0	1	1 1 1 1 0 0 0 0	0 0 0 0 0 0 0 1
4	0	1	1 1 0 1 1 0 0 0	0 0 0 0 0 0 0 1
5	1	0	0 1 1 0 1 1 0 0	0 0 0 0 0 0 0 0
6	1	1	1 0 0 1 0 1 1 0	0 0 0 0 0 0 0 1
7	0	1	1 1 1 0 1 0 1 1	0 0 0 0 0 0 0 1
8	1	0	0 1 1 1 0 1 0 1	1 0 0 0 0 0 0 0
9	0	0	0 0 1 1 1 0 1 0	1 1 0 0 0 0 0 0
10	1	1	1 0 1 1 1 1 0 1	0 1 1 0 0 0 0 1
11	1	0	0 1 0 1 1 1 1 0	1 0 1 1 0 0 0 0
12	0	0	0 0 1 0 1 1 1 1	0 1 0 1 1 0 0 0
13	1	1	1 0 1 1 0 1 1 1	1 0 1 0 1 1 0 1
14	1	0	0 1 0 1 1 0 1 1	1 1 0 1 0 1 1 0
15	1	1	1 0 0 0 1 1 0 1	1 1 1 0 1 0 1 0
16	1	1	1 1 1 0 0 1 1 0	1 1 1 1 0 1 0 0
17	0	0	0 1 1 1 0 0 1 1	0 1 1 1 1 0 1 0
18	1	1	1 0 0 1 1 0 0 1	1 0 1 1 1 1 0 0
19	1	1	1 1 1 0 1 1 0 0	1 1 0 1 1 1 1 1
20	1	0	0 1 1 1 0 1 1 0	0 1 1 0 1 1 1 1
21	0	1	1 0 0 1 1 0 1 1	0 0 1 1 1 0 1 0
22	1	1	1 1 1 0 1 1 0 1	1 0 0 1 1 0 1 0
23	0	0	0 1 1 1 0 1 1 0	1 1 0 0 1 1 0 1
24	0	1	1 0 0 1 1 0 1 1	0 1 1 0 0 1 1 1
25	0	1	1 1 1 0 1 1 0 1	1 0 1 1 0 0 1 0
26	1	1	1 1 0 1 0 1 1 0	1 1 0 1 1 0 0 0
27	1	1	1 1 0 0 1 0 1 1	0 1 1 0 1 1 0 1
28	0	1	1 1 0 0 0 1 0 1	1 0 1 1 0 1 1 1
29	1	0	0 1 1 0 0 0 1 0	1 1 0 1 1 0 1 1
30	1	0	0 0 1 1 0 0 0 1	0 1 1 0 1 1 0 1
31	1	0	0 0 0 1 1 0 0 0	1 0 1 1 0 1 1 0
32	1	1	1 0 1 0 1 1 0 0	0 1 0 1 1 0 1 0
			M.S.B. — 2nd Byte of Check Character	L.S.B. — 1st Byte of Check Character
			CRCC Character	

During reception, error checking is done by shifting the data, including the check character, through a similar circuit. This is illustrated in Figure 5. If no errors have occurred during transmission, the results of the first 32 shifts would be identical to those shown in Figure 4. Shifts 33 through 48 cause the check character to be entered, and for correct data will activate the All Zeros detector on shift 48.

Detection of errors is normally followed by a request for re-transmission of the data block. In cassette tape systems using this method of error detection, re-reading the data would require rewinding the tape to return to the beginning of the data block. The MC8503 eliminates this problem by providing the read backward mode shown in Figure 6. Note that the data is shown entering the MC8503 in reverse order, i.e., the most significant bit of the recorded check character is the first bit entered. The operation shown assumes a correct read is obtained as indicated by the all zeros condition of the register after the 48th shift.

Generation and checking examples similar to those just described, but for the CCITT check character, are shown in Figures 7, 8 and 9.

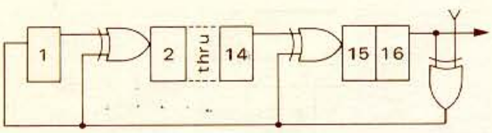
Longitudinal Redundancy Check Character (LRCC) operation, while related to CRCC operation, has significant differences. In Figures 4 thru 9, the CRCC circuits are shown operating on eight bit bytes since this is a common transmission or recording mode. However, the operation is equally valid for serial message streams of any length, i.e., the message length does not have to be a multiple of the byte length.

Longitudinal redundancy checking is used only with data bytes of specified length, i.e., LRCC-8 is used with eight-bit bytes and LRCC-16 is used with sixteen-bit characters. The value (1 or 0) of each bit position in the

FIGURE 5 – CRCC-16 READ OPERATION

Shift n		Input Data on nth Shift	Feedback on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	Register Contents After nth Shift: 9 10 11 12 13 14 15 16
•		•	•	• • • • • • • •	• • • • • • • •
31		1	0	0 0 0 1 1 0 0 0	1 0 1 1 0 1 1 0
32		1	1	1 0 1 0 1 1 0 0	0 1 0 1 1 0 1 0
33	Check Character	0	0	0 1 0 1 0 1 1 0	0 0 1 0 1 1 0 1
34		1	0	0 0 1 0 1 0 1 1	0 0 0 1 0 1 1 0
35		0	0	0 0 0 1 0 1 0 1	1 0 0 0 1 0 1 1
36		1	0	0 0 0 0 1 0 1 0	1 1 0 0 0 1 0 1
37		1	0	0 0 0 0 0 1 0 1	0 1 1 0 0 0 1 0
38		0	0	0 0 0 0 0 0 1 0	1 0 1 1 0 0 0 1
39		1	0	0 0 0 0 0 0 0 1	0 1 0 1 1 0 0 0
40		0	0	0 0 0 0 0 0 0 0	1 0 1 0 1 1 0 0
41		0	0	0 0 0 0 0 0 0 0	0 1 0 1 0 1 1 0
42		0	0	0 0 0 0 0 0 0 0	0 0 1 0 1 0 1 1
43		1	0	0 0 0 0 0 0 0 0	0 0 0 1 0 1 0 1
44		1	0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0
45		0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1
46		1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0
47		0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
48		1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

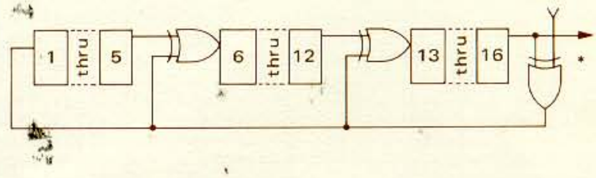
FIGURE 6 – CRCC-16 BACKWARD $(X^{16} + X^{14} + X + 1)$ READ OPERATION



When reading backward, Most Significant Bit in each character appears first.

Shift n	Input Data on nth Shift	Feedback on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16
0			0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1	1	1	1 1 0 0 0 0 0 0	0 0 0 0 0 0 1 0
2	0	0	0 1 1 0 0 0 0 0	0 0 0 0 0 0 1
3	1	0	0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0
4	0	0	0 0 0 1 1 0 0 0	0 0 0 0 0 0 0 0
5	1	1	1 1 0 0 1 1 0 0	0 0 0 0 0 0 1 0
6	1	1	1 0 1 0 0 1 1 0	0 0 0 0 0 0 1 1
7	0	1	1 0 0 1 0 0 1 1	0 0 0 0 0 0 1 1
8	0	1	1 0 0 0 1 0 0 1	1 0 0 0 0 0 1 1
9	0	1	1 0 0 0 0 1 0 0	1 1 0 0 0 0 1 1
10	1	0	0 1 0 0 0 0 1 0	0 1 1 0 0 0 0 1
11	0	1	1 1 1 0 0 0 0 1	0 0 1 1 0 0 1 0
12	1	1	1 0 1 1 0 0 0 0	1 0 0 1 1 0 1 1
13	1	0	0 1 0 1 1 0 0 0	0 1 0 0 1 1 0 1
14	0	1	1 1 1 0 1 1 0 0	0 0 1 0 0 1 0 0
15	1	1	1 0 1 1 0 1 1 0	0 0 0 1 0 0 0 0
16	0	0	0 1 0 1 1 0 1 1	0 0 0 0 1 0 0 0
17	1	1	1 1 1 0 1 1 0 1	1 0 0 0 0 1 1 0
18	1	1	1 0 1 1 0 1 1 0	1 1 0 0 0 0 0 1
19	1	0	0 1 0 1 1 0 1 1	0 1 1 0 0 0 0 0
20	1	1	1 1 1 0 1 1 0 1	1 0 1 1 0 0 1 0
21	0	0	0 1 1 1 0 1 1 0	1 1 0 1 1 0 0 1
22	1	0	0 0 1 1 1 0 1 1	0 1 1 0 1 1 0 0
23	1	1	1 1 0 1 1 1 0 1	1 0 1 1 0 1 0 0
24	0	0	0 1 1 0 1 1 1 0	1 1 0 1 1 0 1 0
25	0	0	0 0 1 1 0 1 1 1	0 1 1 0 1 1 0 1
26	0	1	1 1 0 1 1 0 1 1	1 0 1 1 0 1 0 0
27	1	1	1 0 1 0 1 1 0 1	1 1 0 1 1 0 0 0
28	0	0	0 1 0 1 0 1 1 0	1 1 1 0 1 1 0 0
29	1	1	1 1 1 0 1 0 1 1	0 1 1 1 0 1 0 0
30	1	1	1 0 1 1 0 1 0 1	1 0 1 1 1 0 0 0
31	1	1	1 0 0 1 1 0 1 0	1 1 0 1 1 1 1 0
32	0	0	0 1 0 0 1 1 0 1	0 1 1 0 1 1 1 1
33	1	0	0 0 1 0 0 1 1 0	1 0 1 1 0 1 1 1
34	1	0	0 0 0 1 0 0 1 1	0 1 0 1 1 0 1 1
35	1	0	0 0 0 0 1 0 0 1	1 0 1 0 1 1 0 1
36	1	0	0 0 0 0 0 1 0 0	1 1 0 1 0 1 1 0
37	0	0	0 0 0 0 0 0 1 0	0 1 1 0 1 0 1 1
38	1	0	0 0 0 0 0 0 0 1	0 0 1 1 0 1 0 1
39	1	0	0 0 0 0 0 0 0 0	1 0 0 1 1 0 1 0
40	0	0	0 0 0 0 0 0 0 0	0 1 0 0 1 1 0 1
41	1	0	0 0 0 0 0 0 0 0	0 0 1 0 0 1 1 0
42	0	0	0 0 0 0 0 0 0 0	0 0 0 1 0 0 1 1
43	1	0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 1
44	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0
45	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0
46	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
47	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
48	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

FIGURE 7 – GENERATION OF CCITT $(X^{16} + X^{12} + X^5 + 1)$ CHECK CHARACTER



Shift n	Input Data on nth Shift	Feedback on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16
0			0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
2	1	1	1 0 0 0 0 1 0 0	0 0 0 0 1 0 0 0
3	0	0	0 1 0 0 0 0 1 0	0 0 0 0 0 1 0 0
4	0	0	0 0 1 0 0 0 0 1	0 0 0 0 0 0 1 0
5	1	1	1 0 0 1 0 1 0 0	1 0 0 0 1 0 0 1
6	1	0	0 1 0 0 1 0 1 0	0 1 0 0 0 1 0 0
7	0	0	0 0 1 0 0 1 0 1	0 0 1 0 0 0 1 0
8	1	1	1 0 0 1 0 1 1 0	1 0 0 1 1 0 0 1
9	0	1	1 1 0 0 1 1 1 1	0 1 0 0 0 1 0 0
10	1	1	1 1 1 0 0 0 1 1	1 0 1 0 1 0 1 0
11	1	1	1 1 1 1 0 1 0 1	1 1 0 1 1 1 0 1
12	0	1	1 1 1 1 1 1 1 0	1 1 1 0 0 1 1 0
13	1	1	1 1 1 1 1 0 1 1	0 1 1 1 1 0 1 1
14	1	0	0 1 1 1 1 1 0 1	1 0 1 1 1 1 0 1
15	1	0	0 0 1 1 1 1 1 0	1 1 0 1 1 1 1 0
16	1	1	1 0 0 1 1 0 1 1	0 1 1 0 0 1 1 1
17	0	1	1 1 0 0 1 0 0 1	1 0 1 1 1 0 1 1
18	1	0	0 1 1 0 0 1 0 0	1 1 0 1 1 1 0 1
19	1	0	0 0 1 1 0 0 1 0	0 1 1 0 1 1 1 0
20	1	1	1 0 0 1 1 1 0 1	0 0 1 1 1 1 1 1
21	0	1	1 1 0 0 1 0 1 0	1 0 0 1 0 1 1 1
22	1	0	0 1 1 0 0 1 0 1	0 1 0 0 1 0 1 1
23	0	1	1 0 1 1 0 1 1 0	1 0 1 0 1 1 0 1
24	0	1	1 1 0 1 1 1 1 1	0 1 0 1 1 1 1 0
25	0	0	0 1 1 0 1 1 1 1	1 0 1 0 1 1 1 1
26	1	0	0 0 1 1 0 1 1 1	1 1 0 1 0 1 1 1
27	1	0	0 0 0 1 1 0 1 1	1 1 1 0 1 0 1 1
28	0	1	1 0 0 0 1 0 0 1	1 1 1 1 1 1 0 1
29	1	0	0 1 0 0 0 1 0 0	1 1 1 1 1 1 1 0
30	1	1	1 0 1 0 0 1 1 0	0 1 1 1 0 1 1 1
31	1	0	0 1 0 1 0 0 1 1	0 0 1 1 1 0 1 1
32	1	0	0 0 1 0 1 0 0 1	1 0 0 1 1 1 0 1
			M.S.B. — 2nd Byte of Check Character	1st Byte of Check Character — L.S.B.

CRCC Character

FIGURE 8 – CCITT READ OPERATION

Shift n		Input Data on nth Shift	Feedback on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16
•		•	•	• • • • • • • •	• • • • • • • •
31		1	0	0 1 0 1 0 0 1 1	0 0 1 1 1 0 1 1
32		1	0	0 0 1 0 1 0 0 1	1 0 0 1 1 1 0 1
33	Character	1	0	0 0 0 1 0 1 0 0	1 1 0 0 1 1 1 0
34	Character	0	0	0 0 0 0 1 0 1 0	0 1 1 0 0 1 1 1
35	Character	1	0	0 0 0 0 0 1 0 1	0 0 1 1 0 0 1 1
36	Character	1	0	0 0 0 0 0 0 1 0	1 0 0 1 1 0 0 1
37	Character	1	0	0 0 0 0 0 0 0 1	0 1 0 0 1 1 0 0
38	Character	0	0	0 0 0 0 0 0 0 0	1 0 1 0 0 1 1 0
39	Character	0	0	0 0 0 0 0 0 0 0	0 1 0 1 0 0 1 1
40	Character	1	0	0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 1
41	Check	1	0	0 0 0 0 0 0 0 0	0 0 0 1 0 1 0 0
42	Check	0	0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0
43	Check	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1
44	Check	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0
45	Check	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
46	Check	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
47	Check	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
48	Check	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

LRCC is determined by the data content of the corresponding bit position in each of the data bytes preceeding it. As an example, in Figure 10 four data characters and an appropriate LRCC are shown as the data being entered into an LRCC-8 circuit. Note that each vertical column (bit position) displays even parity. Generation of the LRCC is illustrated in the first 32 rows of the table in Figure 10, showing the results of serially entering the four character data block into an MC8503 connected for LRCC-8 operation. Note that after the 32nd shift the register contains the required LRCC. This character would be shifted out and transmitted with the data.

The LRCC error check is illustrated by the remaining rows of the table. During reception of correct data, the results of serially entering the data into an MC8503 would be identical to those obtained while generating the LRCC (first 32 rows of the table). Entry of the transmitted LRCC (last eight rows of the table) leads to an All Zeros indication after the 40th shift.

Similar results for LRCC-16 operation are shown in Figure 11. For simplicity, only two data bytes are assumed. Note that the required LRCC is present in the register after the last data bit (shift 32) is entered. As before, shifts 33-48 show the results of the error check during reception.

REFERENCES

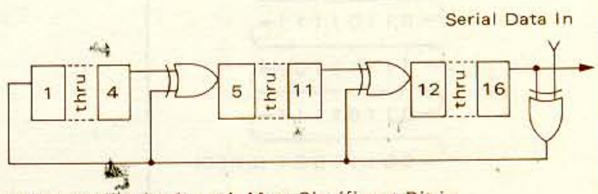
Additional theoretical background concerning these error correction techniques can be found in:

1. Peterson, W. Wesley: "Error-Correcting Codes", MIT Press, Cambridge, Mass., 1965.

Procedures for using the check characters in data transmission and digital magnetic tape recording, respectively, are described in:

2. "General Information – Binary Synchronous Communications", IBM Systems Reference Library, File No. TP-09, Form GA27-3004-1.
3. "Specifications for Magnetic Tape Cassette for Information Exchange", Secretariat Business Equipment Manufacturers Association, ANSI X3B1/579, American National Standards Institute, Inc., 1430 Broadway, New York, N.Y., September 14, 1972.

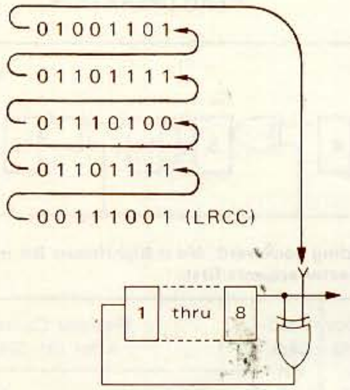
FIGURE 9 – CCITT BACKWARD ($X^{16} + X^{11} + X^4 + 1$) READ OPERATION



* When reading backward, Most Significant Bit in each character appears first.

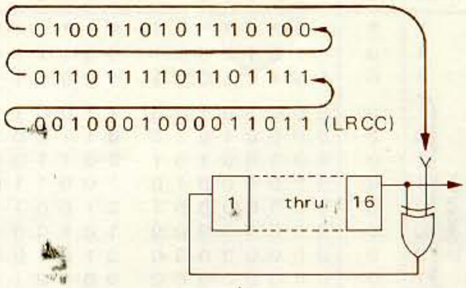
Shift n	Input Data on nth Shift	Feed-Back on nth Shift	Register Contents After nth Shift: 1 2 3 4 5 6 7 8	9 10 11 12 13 14 15 16
0			0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
1	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
2	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
3	1	1	1 0 0 0 1 0 0 0	0 0 0 1 0 0 0 0
4	0	0	0 1 0 0 0 1 0 0	0 0 0 0 1 0 0 0
5	1	1	1 0 1 0 1 0 1 0	0 0 0 1 0 1 0 0
6	0	0	0 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0
7	0	0	0 0 1 0 1 0 1 0	1 0 0 0 0 1 0 1
8	1	0	0 0 0 1 0 1 0 1	0 1 0 0 0 0 1 0
9	1	1	1 0 0 0 0 0 1 0	1 0 1 1 0 0 0 1
10	0	1	1 1 0 0 1 0 0 1	0 1 0 0 1 0 0 0
11	0	0	0 1 1 0 0 1 0 0	1 0 1 0 0 1 0 0
12	1	1	1 0 1 1 1 0 1 0	0 1 0 0 0 0 1 0
13	1	1	1 1 0 1 0 1 0 1	0 0 1 1 0 0 0 1
14	1	0	0 1 1 0 1 0 1 0	1 0 0 1 1 0 0 0
15	0	0	0 0 1 1 0 1 0 1	0 1 0 0 1 1 0 0
16	1	1	1 0 0 1 0 0 1 0	1 0 1 1 0 1 1 0
17	1	1	1 1 0 0 0 0 0 1	0 1 0 0 1 0 1 1
18	1	0	0 1 1 0 0 0 0 0	1 0 1 0 0 1 0 1
19	1	0	0 0 1 1 0 0 0 0	0 1 0 1 0 0 1 0
20	1	1	1 0 0 1 0 0 0 0	0 0 1 1 1 0 0 1
21	0	1	1 1 0 0 0 0 0 0	0 0 0 0 1 1 0 0
22	1	1	1 1 1 0 1 0 0 0	0 0 0 1 0 1 1 0
23	1	1	1 1 1 1 1 1 0 0	0 0 0 1 1 0 1 1
24	0	1	1 1 1 1 0 1 1 0	0 0 0 1 1 1 0 1
25	0	1	1 1 1 1 0 0 1 1	0 0 0 1 1 1 1 0
26	0	0	0 1 1 1 1 0 0 1	1 0 0 0 1 1 1 1
27	1	0	0 0 1 1 1 1 0 0	1 1 0 0 0 1 1 1
28	0	1	1 0 0 1 0 1 1 0	0 1 1 1 0 0 1 1
29	1	0	0 1 0 0 1 0 1 1	0 0 1 1 1 0 0 1
30	1	0	0 0 1 0 0 1 0 1	1 0 0 1 1 1 0 0
31	1	1	1 0 0 1 1 0 1 0	1 1 0 1 1 1 1 0
32	0	0	0 1 0 0 1 1 0 1	0 1 1 0 1 1 1 1
33	1	0	0 0 1 0 0 1 1 0	1 0 1 1 0 1 1 1
34	1	0	0 0 0 1 0 0 1 1	0 1 0 1 1 0 1 1
35	1	0	0 0 0 0 1 0 0 1	1 0 1 0 1 1 0 1
36	1	0	0 0 0 0 0 1 0 0	1 1 0 1 0 1 1 0
37	0	0	0 0 0 0 0 0 1 0	0 1 1 0 1 0 1 1
38	1	0	0 0 0 0 0 0 0 1	0 0 1 1 0 1 0 1
39	1	0	0 0 0 0 0 0 0 0	1 0 0 1 1 0 1 0
40	0	0	0 0 0 0 0 0 0 0	0 1 0 0 1 1 0 1
41	1	0	0 0 0 0 0 0 0 0	0 0 1 0 0 1 1 0
42	0	0	0 0 0 0 0 0 0 0	0 0 0 1 0 0 1 1
43	1	0	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 1
44	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0
45	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0
46	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1
47	1	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
48	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

FIGURE 10 – LRCC-8 OPERATION



Shift n	Input Data on nth Shift	Register Contents After nth Shift 9	10	11	12	13	14	15	16
0		0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	1	1	0	0	1	0	0	0	0
6	1	1	1	0	0	1	0	0	0
7	0	0	1	1	0	0	1	0	0
8	1	1	0	1	1	0	0	1	0
9	0	0	1	0	1	1	0	0	1
10	1	0	0	1	0	1	1	0	0
11	1	1	0	0	1	0	1	1	0
12	0	0	1	0	0	1	0	1	1
13	1	0	0	1	0	0	1	0	1
14	1	0	0	0	1	0	0	1	0
15	1	1	0	0	0	1	0	0	1
16	1	0	1	0	0	0	1	0	0
17	0	0	0	1	0	0	0	1	0
18	1	1	0	0	1	0	0	0	1
19	1	0	1	0	0	1	0	0	0
20	1	1	0	1	0	0	1	0	0
21	0	0	1	0	1	0	0	1	0
22	1	1	0	1	0	1	0	0	1
23	0	1	1	0	1	0	1	0	0
24	0	0	1	1	0	1	0	1	0
25	0	0	0	1	1	0	1	0	1
26	1	0	0	0	1	1	0	1	0
27	1	1	0	0	0	1	1	0	1
28	0	1	1	0	0	0	1	1	0
29	1	1	1	1	0	0	0	1	1
30	1	0	1	1	1	0	0	0	1
31	1	0	0	1	1	1	0	0	0
32	1	1	0	0	1	1	1	0	0
33	0	0	1	0	0	1	1	1	0
34	0	0	0	1	0	0	1	1	1
35	1	0	0	0	1	0	0	1	1
36	1	0	0	0	0	1	0	0	1
37	1	0	0	0	0	0	1	0	0
38	0	0	0	0	0	0	0	1	0
39	0	0	0	0	0	0	0	0	1
40	1	0	0	0	0	0	0	0	0

FIGURE 11 – LRCC-16 OPERATION



Shift n	Input Data on nth Shift	Register Contents After nth Shift 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
8	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
9	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
10	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0
11	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0
12	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0	0
13	0	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
14	1	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0
15	0	0	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0
16	0	0	0	1	0	1	1	1	0	1	0	1	1	0	0	1	0
17	0	0	0	0	1	0	1	1	1	0	1	0	1	1	0	0	1
18	1	0	0	0	0	1	0	1	1	1	0	1	0	1	1	0	0
19	1	1	0	0	0	0	1	0	1	1	1	0	1	0	1	1	0
20	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	1	1
21	1	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0	1
22	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0
23	1	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1
24	1	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0
25	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1
26	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1
27	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1
28	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0
29	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1
30	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0
31	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0
32	1	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0
33	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0
34	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1
35	1	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0
36	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0	0
37	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1	0
38	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0	1
39	1	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0
40	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0
41	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0
42	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
43	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
44	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
45	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
47	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
48	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MOTOROLA Semiconductor Products Inc.

BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.

8722 PRINTED IN USA 9-73 IMPERIAL LITHO 839112 13M DS 9285