2) Multiply **D(x)** by what is called generator polynomial **g(x)** of order **r=n-k**. This **g(x)** is one of the multiplication of some factors of $\mathbf{x^n+1}$. **Note** that factorization of $\mathbf{x^n+1}$ is not an easy matter for any ***n***. tables are used to find the factors of $\mathbf{x^n+1}$.

n	$x^n + 1 =$
1	$(x + 1)$
2	$(x + 1)^2$
3	$(x + 1)(x^2 + x + 1)$
4	$(x + 1)^4$
5	$(x + 1)(x^4 + x^3 + x^2 + x + 1)$
6	$(x + 1)^2(x^2 + x + 1)^2$
7	$(x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)$
8	$(x + 1)^8$
9	$(x + 1)(x^2 + x + 1)(x^6 + x^3 + 1)$
10	$(x + 1)^2(x^4 + x^3 + x^2 + x + 1)^2$
11	$(x + 1)(x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$
12	$(x + 1)^4(x^2 + x + 1)^4$
13	$(x + 1)(x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$
14	$(x + 1)^2(x^3 + x + 1)^2(x^3 + x^2 + 1)^2$
15	$(x + 1)(x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + 1)(x^4 + x^3 + x^2 + x + 1)$
16	$(x + 1)^{16}$
17	$(x + 1)(x^8 + x^5 + x^4 + x^3 + 1)(x^8 + x^7 + x^6 + x^4 + x^2 + x + 1)$
18	$(x + 1)^2(x^2 + x + 1)^2(x^6 + x^3 + 1)^2$
19	$(x + 1)(x^{18} + x^{17} + x^{16} + \cdots + x + 1)$
20	$(x + 1)^4(x^4 + x^3 + x^2 + x + 1)^4$
21	$(x + 1)(x^2 + x + 1)(x^3 + x + 1)(x^3 + x^2 + 1)(x^6 + x^4 + x^2 + x + 1)(x^6 + x^5 + x^4 + x^2 + 1)$
22	$(x + 1)^2(x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)^2$
23	$(x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1)$
24	$(x + 1)^8(x^2 + x + 1)^8$
25	$(x + 1)(x^4 + x^3 + x^2 + x + 1)(x^{20} + x^{15} + x^{10} + x^5 + 1)$
26	$(x + 1)^2(x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)^2$
27	$(x + 1)(x^2 + x + 1)(x^6 + x^3 + 1)(x^{18} + x^9 + 1)$
28	$(x + 1)^4(x^3 + x + 1)^4(x^3 + x^2 + 1)^4$
29	$(x + 1)(x^{28} + x^{27} + \cdots + x + 1)$
30	$(x + 1)^2(x^2 + x + 1)^2(x^4 + x + 1)^2(x^4 + x^3 + 1)^2(x^4 + x^3 + x^2 + x + 1)^2$
31	$(x + 1)(x^5 + x^2 + 1)(x^5 + x^3 + 1)(x^5 + x^3 + x^2 + x + 1)$ $\cdot(x^5 + x^4 + x^2 + x + 1)(x^5 + x^4 + x^3 + x + 1)(x^5 + x^4 + x^3 + x^2 + 1)$

For example, if n=7, then $x^7+1=(x+1)(x^3+x^2+1)(x^3+x+1)$

We can extract the generator polynomial according to ***r***

(n,k)	$g(x)$
$(7,7)$	1
$(7,6)$	$x+1$
$(7,4)$	x^3+x+1
$(7,4)$	x^3+x^2+1
$(7,3)$	$(x+1)(x^3+x+1)$
$(7,3)$	$(x+1)(x^3+x^2+1)$
$(7,1)$	$(x^3+x+1)(x^3+x^2+1)$
$(7,0)$	x^7+1

For n=7, r=3, we can choose either $g_1(x)= x^3+x^2+1$ or $g_2(x)= x^3+x+1$

for n=7, r=4, we can choose either $g_1(x)=(x+1)(x^3+x^2+1)$ or $g_2(x) =(x+1)(x^3+x+1)$

(3) The output codeword polynomial will be:

C(x)=D(x) g(x)

Then **C(x)** is used to find the output code word **[C]**.

Example: Write down the code table for the (7,4) nonsystematic cyclic code with generator polynomial **$g(x)=x^3+x+1$.**

Solution: Here n=7, k=4, r=3, [D]=[a_1 a_2 a_3 a_4], so the table has 16 rows starts from 0000 up to 1111

x^3 x^2 x^1 x^0

--if [D]= [0 0 0 1], then D(x)=1 and C(x)=D(x)g(x)=1*(x^3+x+1) then [C]=[0001011]

x^3 x^2 x^1 x^0

--if [D]=[0 0 1 0], then D(x)=x and C(x)=D(x)g(x)=x(x^3+x+1)=x^4+x^2+x or [C]=[0010110].

--if [D]=[0011], then D(x)=1+x and C(x)=D(x)g(x)=(1+x)(x^3+x+1)=$x^3+x+1+x^4+x^2+x=x^4+x^3+x^2+1$ or [C]=[0011101]

--if [D]=[0100], then D(x)=x^2 and C(x)=D(x)g(x)=x^2 (x^3+x+1)=$x^5+x^3+x^2$ or [C]=[0101100].

Note that similar terms are canceled.

1+1=0

x+x=0

$x^2+x^2=0$ and so on

Also note that

o/p [C] are nonsystematic

i/p [D]				o/p [C]							
a_1	a_2	a_3	a_4	c_1	c_2	c_3	c_4	c_5	c_6	c_7	w_i
0	0	0	0	0	0	0	0	0	0	0	---
0	0	0	1	0	0	0	1	0	1	1	3
0	0	1	0	0	0	1	0	1	1	0	3
0	0	1	1	0	0	1	1	1	0	1	4
0	1	0	0	0	1	0	1	1	0	0	3
0	1	0	1	0	1	0	0	1	1	1	4
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

B) Systematic Cyclic Codes (Division):

The procedure for the generation of **(n,k)** systematic cyclic code is as follows:

1. Find D(x) from [D] as before.
2. As before, select a generator polynomial g(x) of order ***r*** from the factorization table of $\mathbf{x^n+1}$.
3. The output codeword will be:

$$C(x) = x^r\ D(x) + Rem\frac{x^r\ D(x)}{g(x)}$$

where ***Rem*** is the remainder of the long division of x^r D(x) by g(x).

4. Use C(x) to find [C].

Note that C(x) consists of two parts, the first is which is the same information data shifted to left by ***r*** position, the second is the reminder

$$Rem\frac{x^r\ D(x)}{g(x)}$$

of order (***r***-1) which is the ***r*** LSBs of the output code word or parity bits, hence [C] will have the form:-
[C]=[a_1, a_2, ... a_k c_1, c_2, ...c_r] systematic form.

Example: Write down the code table for the (7,4) systematic cyclic code generated by the generator polynomial $g(x)=x^3+x^2+1$.

Solution:

Here n=7, k=4, r=3: and $C(x) = x^r D(x) + Rem\frac{x^r D(x)}{g(x)}$

For [**D**]=[0001] , **D**(x)=1 then $x^r D(x)=x^3 . 1 == x^3$ -->

$$g(x)\overline{)x^r D(x)}$$

$$\begin{array}{r|l} & 1 \\ \hline x^3+x^2+1 & x^3 \\ & x^3+x^2+1 \\ \hline & x^2+1 \end{array}$$

(x^2+1) is the remainder and the long division stops since x^2+1 has an order less than r. Hence: $C(x)=x^3+x^2+1$, or [C]=[0 0 0 1 1 0 1].

Data parity

Note that the remainder directly gives the r parity bits if written in binary form.

---for [D]=[0010], D(x)=x, $x^rD(x)=x^4$

$$\begin{array}{r|l} & x+1 \\ \hline x^3+x^2+1 & x^4 \\ & x^4+x^3+x \\ \hline & x^3+x \\ & x^3+x^2+1 \\ \hline & x^2+x+1 \end{array}$$

hence $C(x)=x^4+x^2+x+1$, or [C]=[0 0 1 0 1 1 1].

---for [D]=[0011], D(x)=1+x, $x^rD(x)=x^3(1+x)=x^4+x^3$

$$\begin{array}{r|l} & x \\ \hline x^3+x^2+1 & x^4+x^3 \\ & x^4+x^3+x \\ \hline & x \end{array}$$

hence $C(x)=x^4+x^3+x$, or [C]=[0 0 1 1 0 1 0].

and so on with the rest of the codes

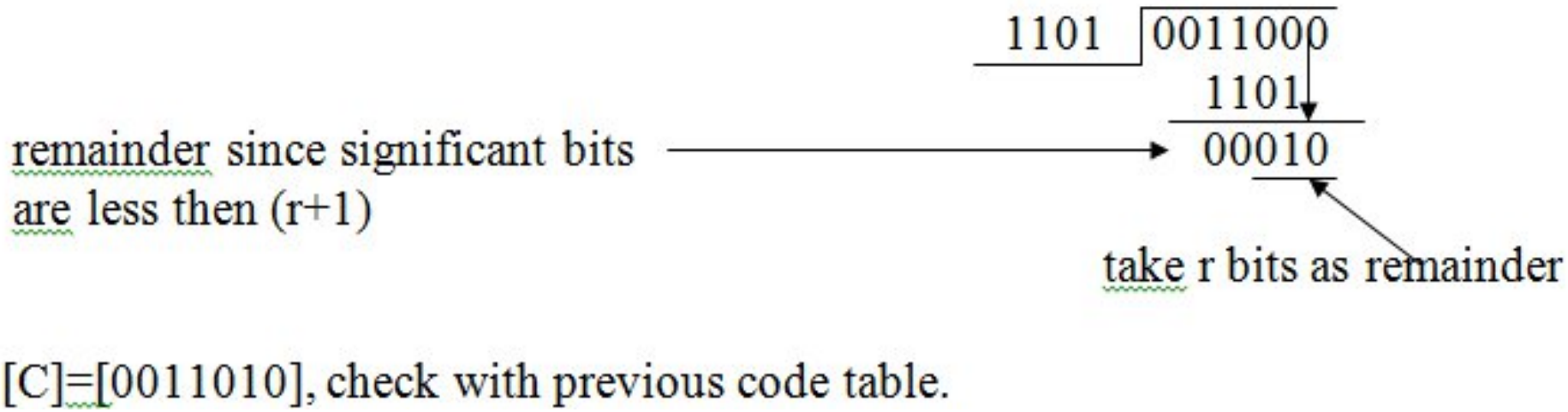
Note: Previous encoding procedure can also be done faster without polynomial representation if g(x) is converted to binary form called the **divisor** of the cyclic code.

For example if $g(x)=x^3+x^2+1$, then the divisor [G]=[1101] consisting of (r+1) bits. Next to find [C] for $[D]=[a_1\ a_2\ \ldots\ldots a_k]$, then put **r** 0's as LSB to get

$[a_1\ a_2\ \ldots\ldots a_k\ 0\ 0\ 0\ \ldots 0]$, then divide this by [G].

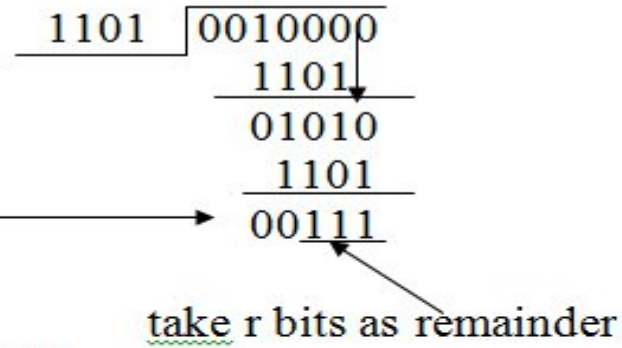
r 0's

Example: Using the generator polynomial of previous example, then $g(x)=x^3+x^2+1$ and [G]=[1101]. For [D]=[0011], then divide [0011000] by [1101]:



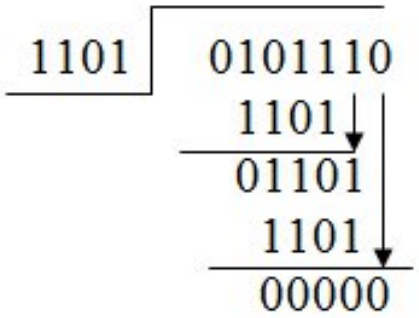
[C]=[0011010], check with previous code table.

for [D]=[0010], then divide [0010000] by [1101]



[C]=[0010111] as before(check with the code table).

Note:
Since the remainder is put as LSB of [C] then we expect that if [C] is divided by g(x) or [G], then the result is always [0].
Check the note by selecting any [C] from precious table and divide by [G]:



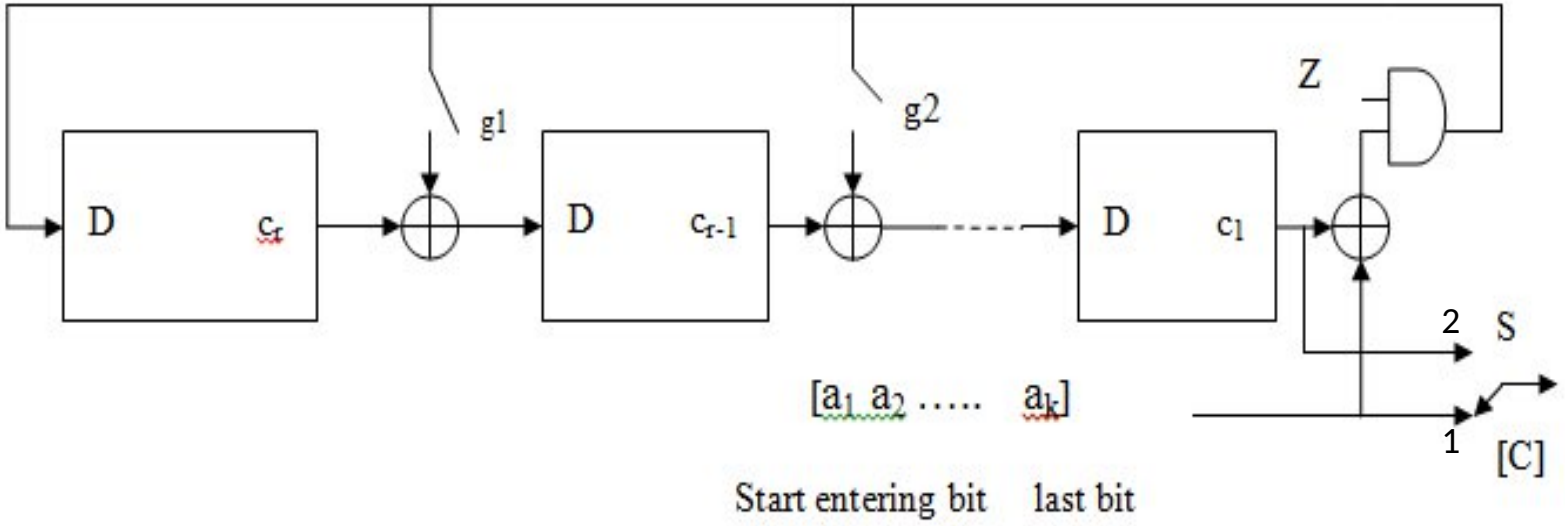
Implementation of systematic cyclic encoder:

In this section we present circuits for performing the encoding operation by presenting circuits for computing polynomial multiplication and division. Hence, we shall show that every cyclic code can be encoded with a simple finite-state machine called a shift-register encoder.

Practically, the previous long division required in long division is done using logic circuit that implements the division by g(x). In general, if:

$g(x)=g_o+g_1 x+g_2 x^2+\ldots\ldots+g_r x^r$, then we must note that for any factorization of x^n+1, **$g_o=g_r=1$** always, hence **only $g_1, g_2, \ldots g_{r-1}$ is shown in the implementation**

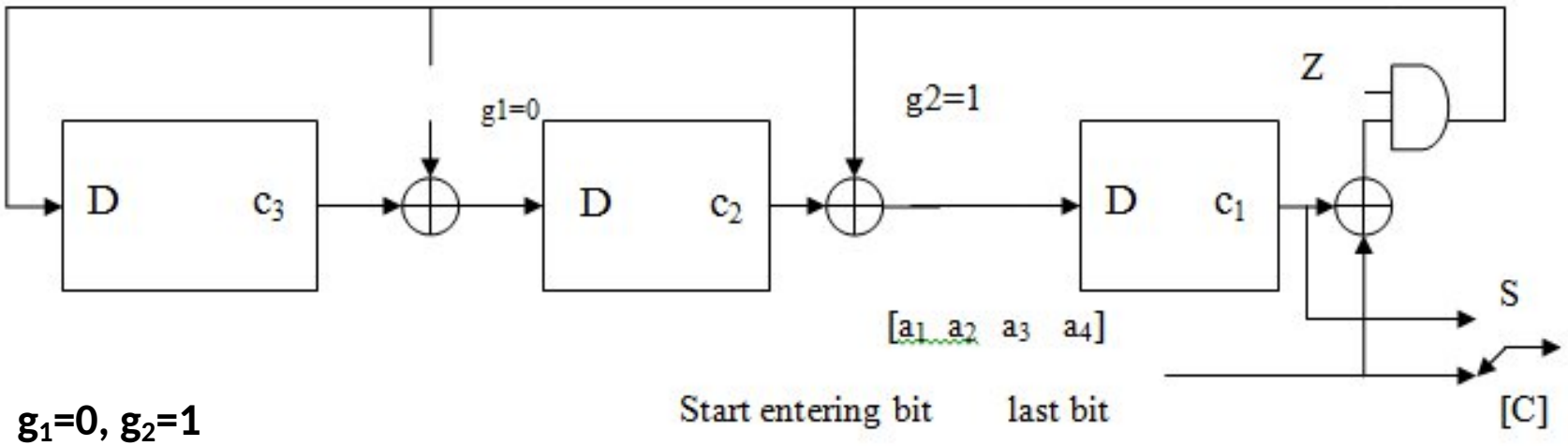
The circuit in Figure below is build by connecting together three types of components: D-flip-flops, adders, and constant multipliers.



Switch S at position (1) and at the same time the control Z is enabled (Z=1) for **k** clock pulses. Then Z is disabled (Z=0) and switch S is changed to position (2) for **r** clock pulses .

Example: Using the encoder circuit, find the output codeword for systematic cyclic code with $g(x)=x^3+x^2+1$ for data words [D]=[0101] and [0010].

Solution: 1) r=3, we need 3 flip flops



$g_1=0$, $g_2=1$

2) we write the transition eqs for c_3, c_2, and c_1:

$$c_3^+ = a_i + c_1^- \text{ where } c_3^+ \text{ is the next state of } c_3. \text{ and } c_1^- \text{ is the present state of } c_1$$

$$c_2^+ = c_3^- \text{ and}$$

$$c_1^+ = c_2^- + c_1^- + a_i = c_2^- + c_3^+$$