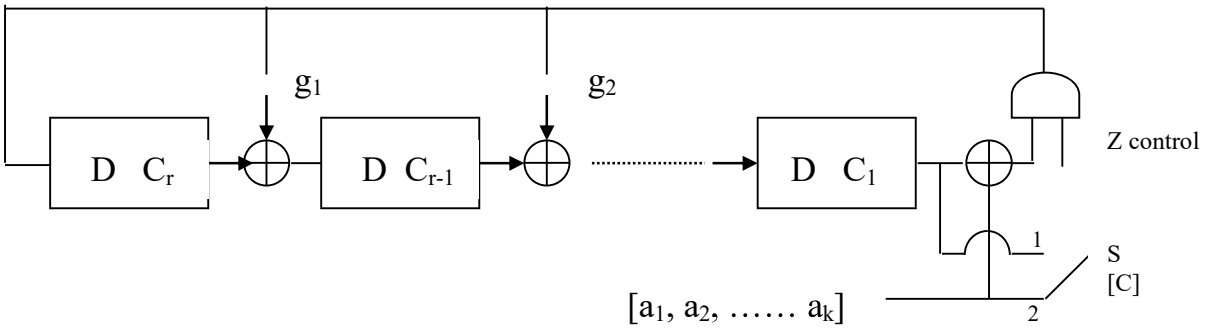
Implementation of Cyclic Encoder:

Practically, the long division required in encoding is done using logic circuit. That implements the division by g(X) polynomial. In general : $g(X)=g_0+ g_1X+ g_2X^2+\ldots\ldots\ldots+g_rX^r$, then $g_0=g_r=1$ always for any factorization of X^n+1. Hence only $g_1, g_2\ldots..,g_{r-1}$ is shown in the implementation circuit:



- This logic circuit is called Modular.
- Feedback shift register implemented using D-flip flop with synchronized clock.

Circuit Operation :-

Switch S is at position 1 giving the data bit to [C] output and at the same time for k clock pulses, the control Z is enabled "Z=1" to feedback the content to the registers to produce $c_1, c_2, \ldots.. c_r$ bits at the end of last clock.
Switch S is at position 2, the Z disabled to get these r parity bits to [C] and at the same time r 0's will be fed back to the register to initialize the register to the next data block.

NOTE

Previous encoding procedure for systematic cyclic code can be done faster without polynomial representation if instead of g(X) is converted into binary form called the "divisor" of the cyclic code.

Ex:
Using $g(X)=X^3+X^2+1$, find the output codeword for [D]=[0011] and [D]=[0010]

Sol:

4. Convert from polynomial into binary form
   $g(X)=X^3+X^2+1$ [G]=[1101]
5. Add r 0's as LSB to data=[a_1, a_2, …. a_k] to get [a_1, a_2, …. a_k 0 0 0]
   for [D]=[0011] we will have [0011000]
6. Divide it by [G]

$$
\begin{array}{r}
 \quad\quad\quad\quad 1 \\
1101\overline{)\,0011000} \\
\quad 001101 \\
\hline
\quad 000\underline{01}\quad \underline{0}
\end{array}
$$

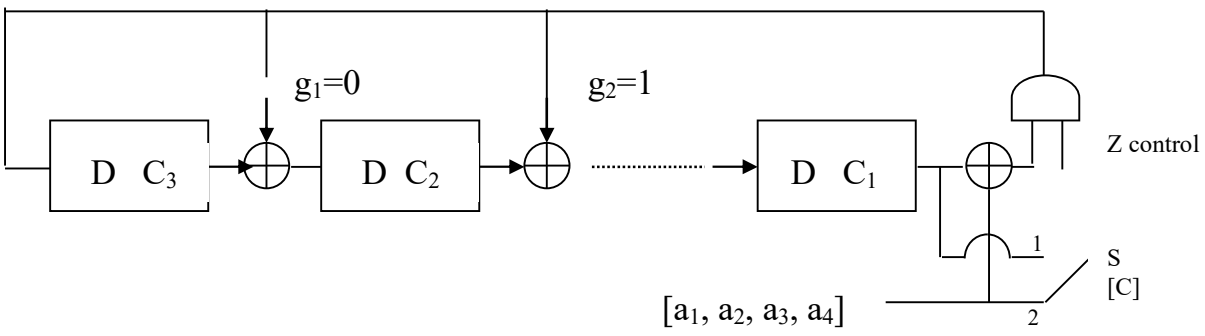
r-parity

Then [C]=[0011010]

For [D]=[0010]

The out put should be [C]=[0010111].. how??

Ex:
Using the encoder logic circuit to find the o/p codeword for systematic cyclic code with $g(x)= X^3+X^2+1$ and for [D]=[0101], [0010]



Sol:

First we write the transition equations for c_1, c_2, c_3 (we write the next state of them in terms of the present state and the input, this is done when Z=1 then:

$c_3^{n+1}=c_1^n+a_i$
$c_2^{n+1}=c_3^n$
$c_1^{n+1}=c_2^n+ c_1^n +a_i= c_2^n+ c_3^{n+1}$

For [D]=[0101]

a_i
0
1
0
1

c_3	c_2	c_1
0	0	0
0	0	0
1	0	1
1	1	1
0	1	1
0	0	1
0	0	0
0	0	0

Then $c_1\ c_2\ c_3$ =110 and [C]=[0101110]

For [D]=[0010]

a_i
0
0
1
0

c_3	c_2	c_1
0	0	0
0	0	0
0	0	0
1	0	1
1	1	1
0	1	1
0	0	1
0	0	0

Then $c_1\ c_2\ c_3$ =111 and [C]=[0010111]

Decoding of Systematic Cyclic Code:-

At the receiver
[R]=[C]+[E] where [E] is the error word
Or
R(X)=C(X)+E(X) , E(X) is the error polynomial

Now if we divide above equation by g(X) taking the reminder :-

$$\mathrm{Re}\,m\left[\frac{R(X)}{g(X)}\right]=\mathrm{Re}\,m\left[\frac{C(X)}{g(X)}\right]+\mathrm{Re}\,m\left[\frac{E(X)}{g(X)}\right]$$

And since $\mathrm{Re}\,m\left[\frac{C(X)}{g(X)}\right]=0$ as shown before, then :-

$\mathrm{Re}\,m\left[\frac{R(X)}{g(X)}\right]=\mathrm{Re}\,m\left[\frac{E(X)}{g(X)}\right]=S(X)$=syndrome polynomial of order (r-1).

1. If S(X) = 0 then no error occurs.
2. If S(X) # 0 then error occurs

To find the locations of these errors, the receiver may prepare a syndrome table, store it in its memory, use it to find [E] from [S] starting with less number of errors.

Ex: Prepare the syndrome table for (7,4) systematic cyclic code with $g(X)=X^3+X^2+1$ and for single error. Check the syndrome when double error at first and last positions occur.

Sol:-

Error Word [E]							S_1	S_2	S_3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	1	0	0
0	0	0	1	0	0	0	1	0	1
0	0	1	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	1	0	1	1
0	0	0	0	1	0	1	1	0	1

Each [S] is found from [E] by long division by g(X). for example if [E]=[0100000] error at second position from the left then:-

```
1 1 0 1 | 0 1 0 0 0 0 0
          1 1 0 1
          ---------
          0 1 0 1 0
            1 1 0 1
          ---------
            0 1 1 1 0
              1 1 0 1
            ---------
              0 0 1 1
                [S]
```

- Note that no repeated [S] for all possible single error. This is expected since $w_{i(min)}=3$ and the given (7,4) code is a single error correction. Note also for double error [E]=[0000011], then [S]=[011] which is single error at the second position (from the left), there by these errors cannot be corrected.

Now going back to the example for double error [E]=[1000001]

```
1 1 0 1 | 1 0 0 0 0 0 1
          1 1 0 1
          ---------
          0 1 0 1 0
            1 1 0 1
          ---------
            0 1 1 1 0
              1 1 0 1
            ---------
              0 0 1 1 1
                [S]
```

[S]=[111] which is the same [S] as if single error occurs at third position from the left.

Ex: using previous syndrome table, find the corrected word for the received word [R]=[1010011].

Sol:-
First, the receiver will find [S] from the reminder of R(X)/g(X)

```
1 1 0 1 | 1 0 1 0 0 1 1
          1 1 0 1
          -------------
          0 1 1 1 0
            1 1 0 1
            -----------
            0 0 1 1 1 1
                1 1 0 1
            -----------
                0 0 0 1 0
                    -----
                     [S]
```

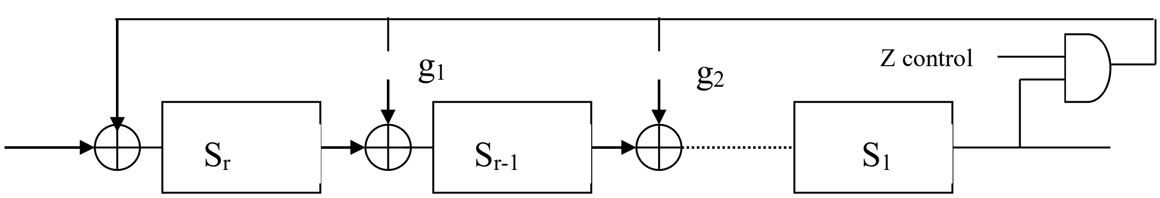
Hence, [S]=[010], using syndrome table and for this syndrome, [E]=[0000010]
Then the corrected codeword

[C]=[R]+[E]

$$\begin{array}{r} 1010011 \\ \underline{0000010} \\ [C] = 1010001 \end{array}$$

Implementation of Cyclic Decoder:

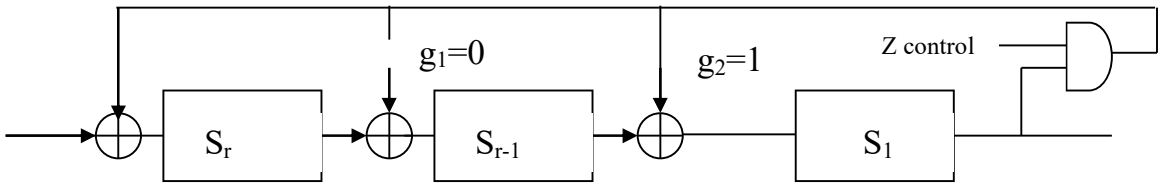
The long division of [R] by [G] to obtain the reminder is implemented is using logic circuit as shown for the generator g(X). The control Z is enabled for "n" clock pulses and then disabled for "r" clock pulses.



$[R]= [r_1, r_2, \ldots, r_n]$ $S=[S_1\ S_2 \ldots\ldots S_r]$

Ex: use the decoder circuit to find the syndrome and the corrected word for the received word [R]=[1011010] , if g(X)= X^3+X^2+1

Sol:-



[R]= [1011010]

S=[S_1 S_2 S_3]

First we write the transition equations for S_1 S_2 S_3 when Z=1 :-

$S_3^{n+1}=S_1^n+r_i$
$S_2^{n+1}=S_3^n$
$S_1^{n+1}=S_2^n+S_1^n$

r_i

1
0
1
1
0
1
0

S_3	S_2	S_1
0	0	0
1	0	0
0	1	0
1	0	1
0	1	1
1	0	0
1	1	0
0	1	1

Then [S]=[110], using the syndrome table then [E]=[1000000].
The corrected code word

$$\begin{array}{r} 1011010 \\ \underline{1000000} \\ [C] = 0011010 \end{array}$$