16548 Notes10
Linear Block Codes IV
Cyclic Block Codes and CRC

Dr. Jay Weitzen
University of Massachusetts Lowell
ECE Department

Chapter 5: Cyclic Block Codes

Cyclic block codes are (arguably) the historically most import of all error correcting codes and perhaps the most important of all "simple" codes.

All cyclic codes are <u>linear</u> codes.

Why are cyclic codes so popular?

1) reasonably efficient

$$R = \frac{k}{n} \longrightarrow 1^{-}$$

2) efficient, low-cost hardware implementations

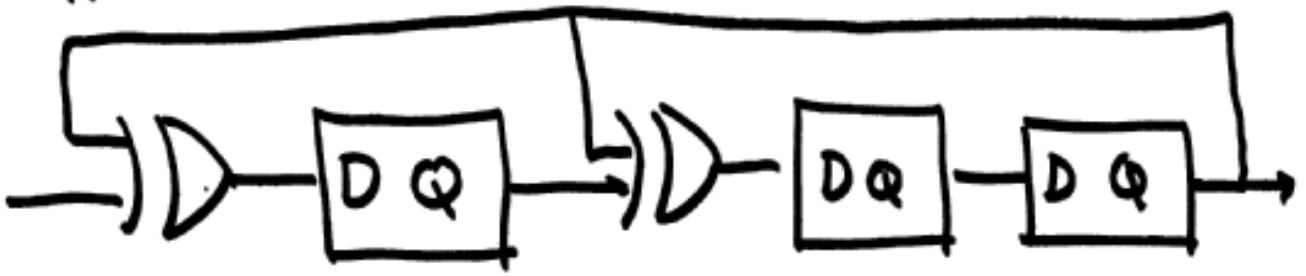
These codes are implemented using shift registers, EX-OR gates, and feedback.

University of Idaho

4

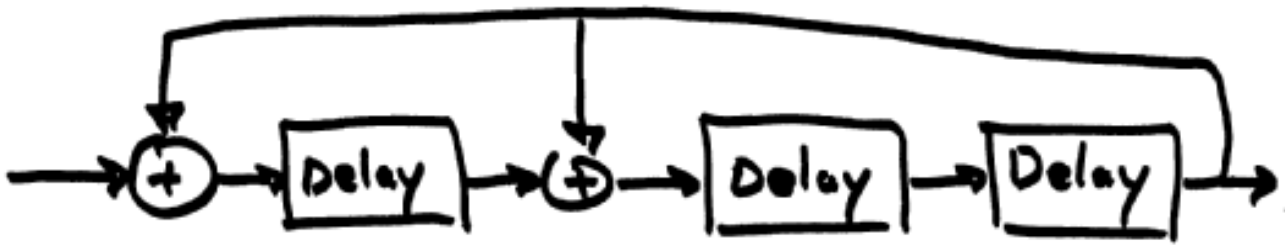
A typical circuit

D-flipflop = "delay flipflop"



These ckts are often called "digital filters."; EX-OR : $1+1=0$ $0+1=1+0=1$, $0+0=0$

addition modulo 2



A digital filter voilà!

if we agree that addition means addition modulo 2, then its not hard to show that the S.R./EX-OR circuit obeys convolution.

a convolution sum

Convolution is the <u>trademark</u> of the linear and time-invariant system.

∴ our feedback shift register circuit IS a LTI system provided 1+1=0

The next thing about this is that all our mathematical tools (e.g., transforms) that are useful in analyzing LTI systems can be used to analyze cyclic codes. (under the constraint that $|+|=0$)

What is a cyclic code?

Concept: cyclic shift

let $\bar{v} = (v_0\, v_1\, v_2\, v_3)$ ← ... first symbol Tx'd

The cyclic shift of $\bar{v}$ is defined to be

$$\bar{v}' = (v_3\, v_0\, v_1\, v_2)$$

Let C be a linear block code with codewords $\bar{c} = (c_0 c_1 \cdots c_{n-1})$

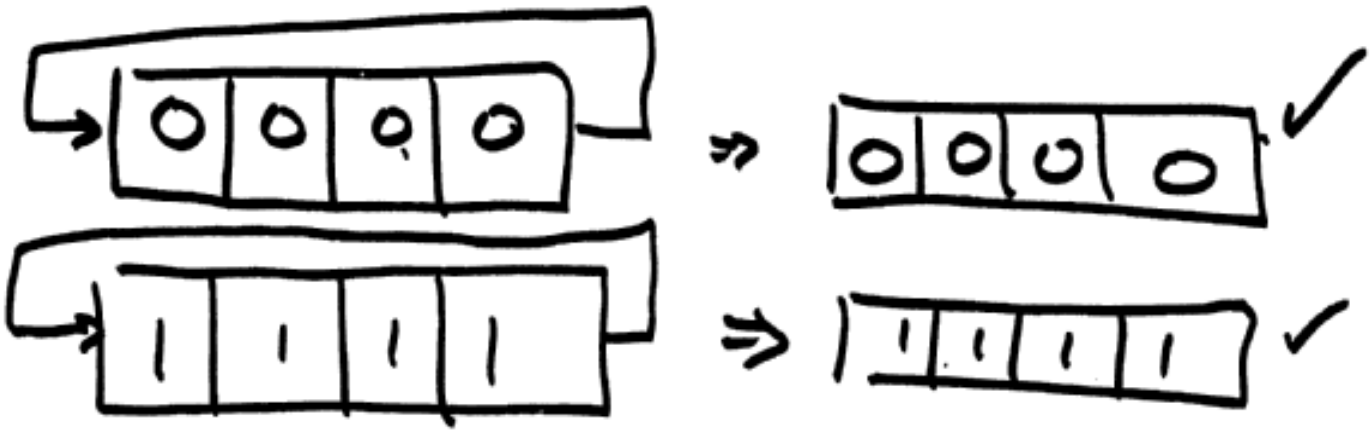
$$c_i \in \{0, 1\}$$

Example

$$C = \{(0000), (1111)\}$$

C is a <u>cyclic</u> code if and only if for every $\bar{c} \in C$, the cyclic shift of $\bar{c}$ is <u>also</u> a codevector.

Is our example a cyclic code?



every $\bar{c}'$ of a $\bar{c}$ is $\in C$

Our example <u>is</u> a cyclic code

another example

$$C = \{(000.0000), (010.1010), (101.0101), (111.1111)\}$$

This is a linear code

$$\bar{0} \in C, \quad \bar{c}_1 + \bar{c}_2 = \bar{c}_3 \in C$$

Is it also a cyclic code?

$(010.1010) \xrightarrow{\text{cyclic shift}} (001.0101) \notin C$

$\therefore$ not a cyclic code

Another example

$$C = \{(000.000), (010.101), (101.010), (111.111)\}$$

This is also linear.

Is it cyclic?

$(000.000) \xrightarrow{c.s.} (000.000)$ ✓

$(111.111) \xrightarrow{c.s.} (111.111)$ ✓

$(010.101) \xrightarrow{c.s.} (101.010)$ ✓

$(101.010) \xrightarrow{c.s.} (010.101)$ ✓

∴ cyclic

For all linear block codes, we have

$$\underset{1\times n}{\overline{c}} = \underset{1\times k}{\overline{m}} \cdot \underset{k\times n}{G}$$

All linear codes can be written either in systematic form $(c_0 c_1 \cdots c_{r-1} m_0 m_1 \cdots m_{k-1})$ or in non-systematic form.

For a non-systematic cyclic code, there always exists a G matrix in the following form.

$n-k=r$ r terms

G is $k \times n$

$$G=\begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k-1} & g_{n-k} \end{bmatrix} \qquad 5.1.1.$$

The rows of this generator matrix are merely cyclic shifts of the $1 \times n$ basis vector

$$\bar{g}=[g_0\ g_1 \cdots g_{n-k-1}\ g_{n-k}\ 0\,0 \cdots 0] \qquad 5.1.2.$$

let $\bar{g} = (1\,1\,0\,1\,0\,0\,0)$

$r+1 = 4 \Rightarrow r = 3$

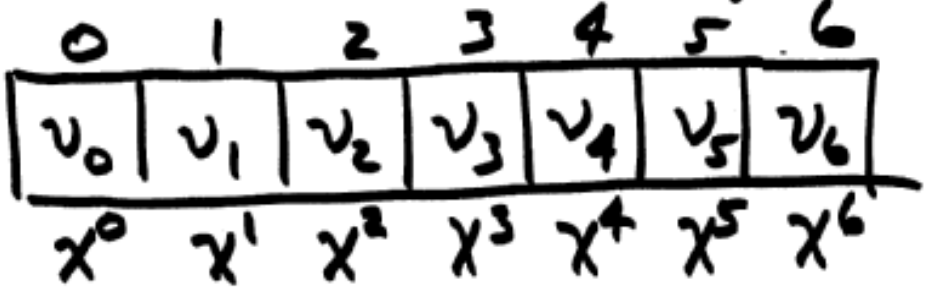
$n = 7 \Rightarrow k = 7-3 = 4$

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Towards the objective of understanding how to do a systematic cyclic code, we introduce the idea of representing our vectors using polynomials.

"Hardware model" of this idea:

Let $\bar{v} = (v_0\, v_1 \ldots v_6)$



← register

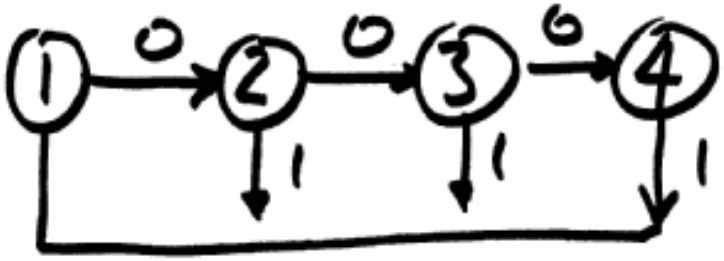
bit position operator: $x^j \Rightarrow j^{th}$ bit position

University of Idaho EE 455
Lec 29

Comment:

$(d, k) = (1, 3)$ $n = 3$

max and min # of possible 3-bit sequences available in this DTC



what is the connection matrix?

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$n = 3$ implies that the vector of possible paths is

$$N(3) = N(0)\,B^3 \qquad [n_1\ n_2\ n_3\ n_4]$$

$N(0) = [0\ 1\ 0\ 0]$

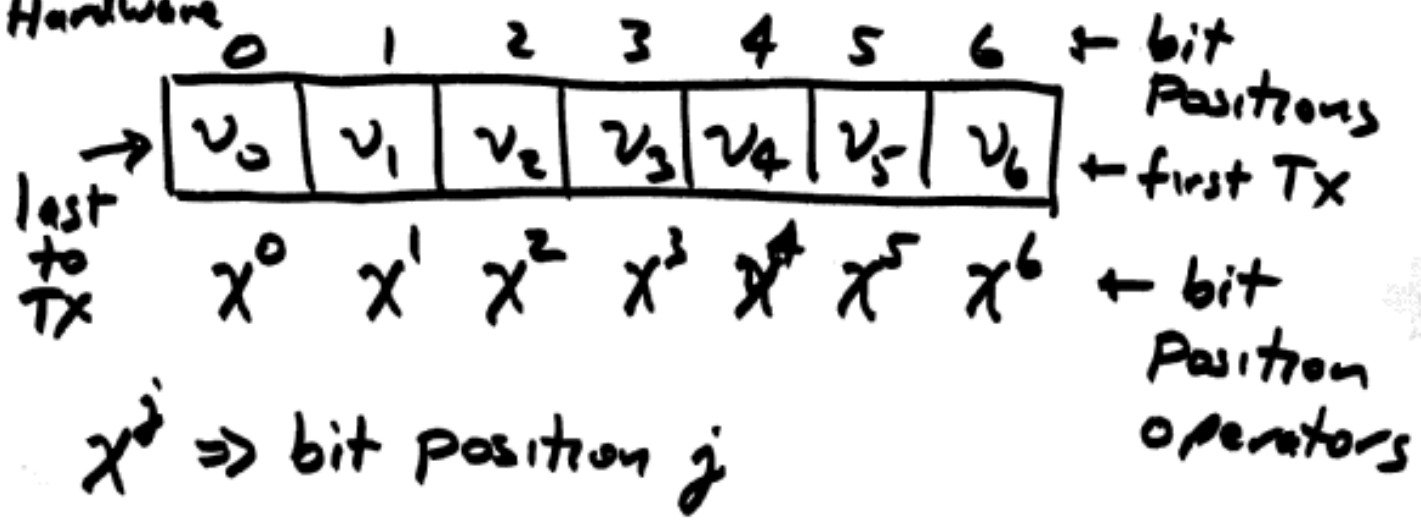
$$B^3 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} \rightarrow 3 \\ \rightarrow 4 \\ \rightarrow 3 \\ \rightarrow 2 \end{matrix}$$

$2 \le n_1 \le 4$

Back to cyclic codes

Last time we introduced the idea of a bit position operator X^j

$$\bar{v} = (v_0\, v_1 \cdots v_6), \quad v_j \in \{0, 1\}$$



$X^j \Rightarrow$ bit position j

Using the bit position operator, we can represent $\overline{v}$ as a polynomial $v(x)$

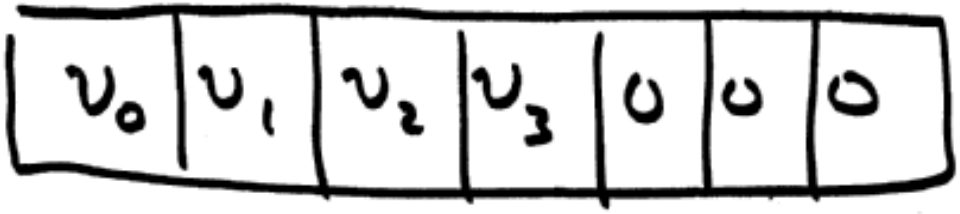
$$v(x) = v_0 x^0 + v_1 x^1 + v_2 x^2 + v_3 x^3 + \cdots + v_6 x^6$$

x in high-priced language is called an "indeterminant" by the math dept.

we can't just assume that $x=0$ or that $x=1$.

Polynomial arithmetic w/ This funny bit Position operator

Suppose we have $\vec{v}$ represented as

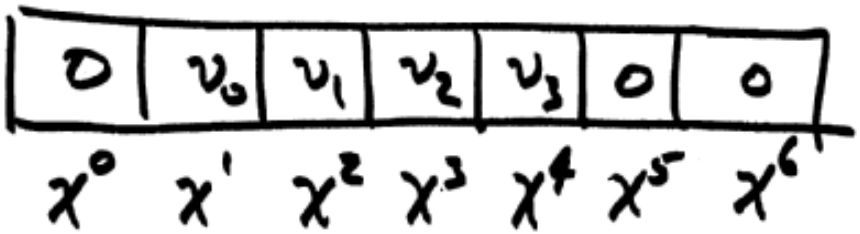


$$v(x) = v_0 x^0 + v_1 x^1 + v_2 x^2 + v_3 x^3$$

what do we get if we multiply $v(x)$ by x^1?

$$v(x) \cdot x^1 = v_0 x^{0+1} + v_1 x^{1+1} + v_2 x^{2+1} + v_3 x^{3+1}$$

$$= v_0 x^1 + v_1 x^2 + v_2 x^3 + v_3 x^4$$



$x^1 \cdot v(x)$ is $v(x)$ shifted 1 bit position with a "0" shifted in from the left into the x^0 position

$\therefore$ x^1 is <u>also</u> a shift operator

(z^{-1})?

other "funny polynomial" arithmetic rules.

multiplication:

$$(a x^i) \cdot (b x^j) \equiv (a \overset{b}{\cdot} b) x^{i+j}$$

for $i+j$, $1+1=2$

↳ regular addition

$a, b \in \{0, 1\}$ then $0 \cdot 0 = 0$

$1 \cdot 0 = 0 \cdot 1 = 0$

$1 \cdot 1 = 1$

The scalar multiplication is an "AND" function

Addition:

Suppose we have $ax^i + bx^j$

and $i \neq j$

bits in 2 different bit positions

add as a polynomial

$$ax^i + bx^j = ax^i + bx^j \text{ if } i \neq j$$

but, what <u>if</u> $i = j$?

2 bits in The <u>same</u> bit position

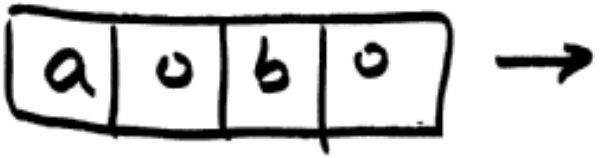
add distributively: $ax^j + bx^j = (a+b)x^j$

$1+1=0$

This means we can do all our normal "factoring" types of tricks

suppose $i < j$

$$ax^i + bx^j = \underbrace{(ax^0 + bx^{j-i})}x^i$$



look at

$$ax^i + bx^i = (ax^0 + bx^0)x^i$$

$$= (a+b)x^0x^i = (a+b)x^i$$

This means x^0 is an operator multiplicative identity and it acts just like "1"

$$v(x) = v_0 x^0 + v_1 x^1 + v_2 x^2$$

$$= v_0 + v_1 x + v_2 x^2$$

abbreviate as

"implied" x^1

implied x^0

$f_i \in \{0,1\}$

$g_i \in \{0,1\}$

Generalizing this:

let $f(x) = f_0 + f_1 x + f_2 x^2$

$g(x) = g_0 + g_1 x + g_2 x^2$

Then

$$f(x)+g(x) = (f_0+g_0)x^0 + (f_1+g_1)x + (f_2+g_2)x^2$$

$1+1=0$

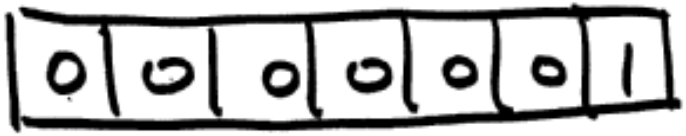
in general

$$f(x)+g(x) = \sum_{j=0}^{n-1} (f_j+g_j)x^j$$

x is a shift operator **but** it is **not** a cyclic shift operator.

Suppose $n = 7$ and $f(x) = x^6$

$f(x)$:



$$x f(x) = x \cdot x^6 = x^{1+6} = x^7$$

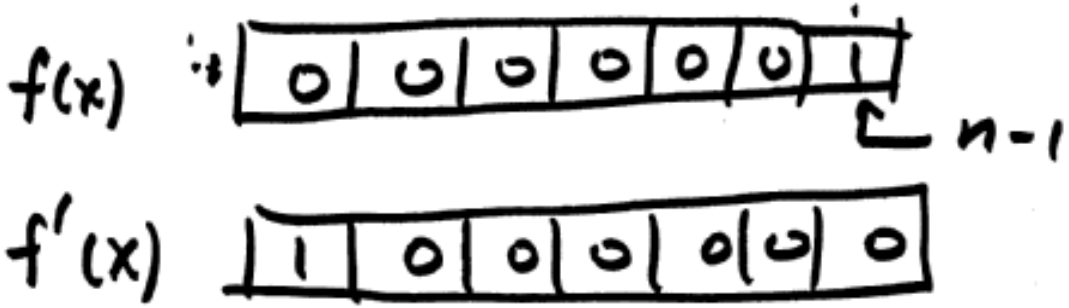
not a cyclic shift!

Here's the problem:

Suppose $f(x)$ is the polynomial representation of a codeword in a cyclic code. Then the cyclic shift of $f(x)$ – call it $f'(x)$ – must also be a legal codeword.

How can we represent a cyclic shift using polynomials?

for $n=7$ and $f(x)=x^6$, we have to have $f'(x)=x^0$



$f'(x)=x^0=1 \qquad x f(x)=x^7$

$n = 7$

I claim that

$$1 = x^7 - (x^7 - 1)$$

but what's this "minus" gaze?

$$1 + 1 = 0 \Rightarrow 1 = -1, \quad \underline{\underline{-1 = 1}}$$

$$\therefore -1 = 1 \Rightarrow x^7 - 1 = x^7 + 1$$

$$-(x^7 - 1) = -x^7 + 1 = x^7 + 1$$

$$\therefore f'(x) = x \cdot f(x) - (x^n - 1) = xf(x) + x^n + 1$$

when $f(x) \neq x^{n-1}$, if we look at the hardware representation, we find that what we just did on pg 15 ends up being equivalent to

$$f'(x) = (x \cdot f(x)) \bmod (x^n - 1)$$

$$= x^n + 1$$

$f(x) = 1$

$$(x f(x)) \bmod (x^7 + 1) = x \bmod x^7 + 1 = x$$

what would

$p(x) = (x^4 + x^2 + x) \bmod (x^4 + 1)$ be ?

long division

$$
\begin{array}{r|l}
 & 1 \quad \leftarrow \text{quotient} \\
\hline
x^4 + 1 & x^4 + x^2 + x \\
 & x^4 \qquad\qquad + 1 \\
\hline
 & 0 + x^2 + x + 1
\end{array}
$$

$p(x) = x^2 + x + 1$ remainder

Polynomial arithmetic: (more)

- objective: get comfortable (and good) at the mechanics of modulo polynomial arithmetic
- why? This arithmetic tells us how to build hardware for encoders and decoders.

Last time we intro'd the cyclic shift and in polynomial form we have

if $v'(x)$ = cyclic shift of $(v(x))$

and $v(x) = v_0 + v_1 x + v_2 x^2 + \cdots + v_{n-1} x^{n-1}$

$$v'(x) = x\,v(x) - v_{n-1}\left(x^n - 1\right)$$

since $v_i \in \{0, 1\}$ and $1+1=0 \Rightarrow -1 = +1$

$$v'(x) = x v(x) + v_{n-1}\left(x^n + 1\right)$$

and This is equivalent to saying

$$v'(x) = x \cdot v(x) \bmod (x^n + 1)$$

Now as it happens, x^n+1 is <u>not</u> the <u>only</u> polynomial we're interested in as the "modulo" term

Let's consider The more general operation of $f(x) \bmod g(x)$

a modulo is basically a remainder

$5 \bmod 2 = 1$

0	1	2	3	4	5
0	1	0	1	0	1

$f(x) \bmod g(x) \equiv f(x) / g(x) = p(x)$

$\equiv$: define notation

$p(x)$: remainder

Polynomial remainders can be found by the algorithm known as long division

Example: $f(x) = x^3 + x + 1$

$g(x) = x^2 + 1$

find $f(x) / g(x)$

$$
\begin{array}{r|l}
 & x \leftarrow \text{Quotient } Q(x) \\
\hline
x^2+1 & x^3 + x + 1 \\
 & x^3 + x \\
\hline
 & 0 + 0 + 1
\end{array}
$$

$\uparrow$ Remainder $r(x) = x^0$

Division "in general" for polynomials has to be carefully defined. Let's call the notation for "regular polynomial division" $f(x) \div g(x)$

$f(x) \div g(x)$ is <u>defined</u> as

$$f(x) = Q(x) \cdot g(x) + \rho(x)$$

$Q(x)$: quotient; $\rho(x)$: remainder

$$\rho(x) = f(x) / g(x)$$

$$Q(x) \cdot g(x) = f(x) - \rho(x) = f(x) + \rho(x)$$

more terminology

let $f(x) = f_0 + f_1 x + f_2 x^2 + \cdots + f_{n-1} x^{n-1}$

$f_i \in \{0, 1\}$ and that $f_{n-1} \neq 0$

Then the <u>degree</u> of polynomial $f(x)$ is $\deg(f(x)) = n-1$

now suppose

$g(x) = g_0 + g_1 x + \cdots + g_r x^r$

w/ $g_r \neq 0$; Then $\deg(g(x)) = r < n$

in general, if we have

$$p(x) = f(x) / g(x)$$

Then $\deg(p(x)) < \deg(g(x))$

$$\begin{array}{r|l} & x+1 \\ \hline x^2+1 & x^3 + x^2 + 1 \\ g(x) & x^3 + x \\ \hline & x^2 + x + 1 \\ & x^2 + 1 \\ \hline & x = p(x) \end{array}$$

some notation convention for what is about to follow: Given $g(x)$,

$$f_1(x) = Q_1(x)\,g(x) + P_1(x)\,;\quad P_1(x) = f_1(x)/g(x)$$

$$f_2(x) = Q_2(x)\,g(x) + P_2(x),\quad P_2(x) = f_2(x)/g(x)$$

3 handy identities:

1) $\left[Q(x)\cdot g(x)\right]/g(x) \equiv 0$

2) $\left[f_1(x) + f_2(x) \right] / g(x) = P_1(x) + P_2(x)$

Proof :

$$f_1(x) + f_2(x) = Q_1(x)\, g(x) + P_1(x)$$
$$+ Q_2(x)\, g(x) + P_2(x)$$
$$= \left(Q_1(x) + Q_2(x) \right) g(x) + P_1(x) + P_2(x)$$

now

$$\left[Q(x) + Q_2(x) \right] \cdot g(x) / g(x) = 0$$

in addition

$$\deg(P_1(x)) < \deg(g(x))$$

$$\deg(P_2(x)) < \deg(g(x))$$

$$\therefore \deg(P_1(x) + P_2(x)) < \deg(g(x))$$

$$\therefore (f_1(x) + f_2(x))/g(x) = P_1(x) + P_2(x)$$

QED

The "theorem" we just saw tells us that the remainder wrt $g(x)$ of a sum of polynomials is the sum of the individual remainders $r_i(x)$

Suppose

$$v(x) = v_0 + v_1 x + v_2 x^2 + \cdots + v_{n-1} x^{n-1}$$

where $n = 7$

and $g(x) = x^3 + x + 1$

$v(x) / g(x) = ?$

University of Idaho

(14)

w/ $n = 7$ let $v(x) = x^5 + x^4 + x^2 + 1$

$v(x)/g(x)$

$x^6/g(x) = x^2+1$

$x^5/g(x) = x^2+x+1$ ←

$x^4/g(x) = x^2+x$ ←

$x^3/g(x) = x+1$

$x^2/g(x) = x^2$

$x/g(x) = x$

$1/g(x) = 1$

$$x^2+x+1$$
$$x^2+x$$
$$x^2$$
$$1$$
$$x^2$$

$v(x)/g(x) = x^2$

is $x^3/g(x)$ really $= x+1$?

$$x^3+x+1 \;\overline{)\; x^3 \;}\; 1$$

$$x^3+x+1$$

$$r = 3$$

$$x+1$$

✓

Handy identity #3:

what is

$$\left[f_1(x) \cdot f_2(x) \right] / g(x) = \left[P_1(x) \cdot P_2(x) \right] / g(x)$$

Proof:

$$f_1(x) \cdot f_2(x) = \left[Q_1(x) g(x) + P_1(x) \right] \cdot \left[Q_2(x) g(x) + P_2(x) \right]$$

$$= Q_1(x) Q_2(x) g(x) g(x)$$

$$+ Q_1(x) P_2(x) g(x) + Q_2(x) P_1(x) g(x)$$

$$+ P_1(x) \cdot P_2(x)$$

$$\therefore [f_1(x) \cdot f_2(x)] / g(x) = 0 + 0 + 0$$
$$+ \left(P_1(x) \cdot P_2(x)\right) / g(x)$$

QED

Now the biggest little theorem you ever saw: Meggitt's Theorem

Theorem (ex 5.3.4)

if $\deg(g(x)) = r$ and if

$(x^n + 1) / g(x) = 0$

and if $v(x) / g(x) = p(x)$

Then $v'(x) / g(x) = [x \cdot p(x)] / g(x)$

$v'(x) = [x\, v(x) \bmod (x^n + 1)]$ cyclic shift of $v(x)$

Proof:

since $(X^n+1)/g(x) = 0$

by the definition of poly. division

$$X^n+1 = Q(x)\cdot g(x)$$

Let $Q(x) = h(x)$

$$X^n+1 = h(x)\, g(x)$$

now

$$v'(x) = x\,v(x) + v_{n-1}(x^n+1) \qquad \text{cyclic shift}$$

but $x^n+1 = h(x)\cdot g(x)$

$$\therefore\ v'(x) = x\,v(x) + v_{n-1}\,h(x)\,g(x)$$

Also since $p(x) = v(x)/g(x)$, then

$$v'(x)/g(x) = [x\,v(x)]/g(x) + \underbrace{[v_{n-1}\,h(x)\,g(x)]/g(x)}_{0}$$

$$= [(x/g(x))\cdot(v(x)/g(x))]/g(x)$$

$$\therefore \quad v'(x)/g(x) = [x \cdot P(x)]/g(x)$$

University of Idaho EE 455 Lec 31

Basic strategy:

1) come up w/ systematic cyclic codes (modulo polynomial arithmetic)

2) show how we can do mod. poly. arithmetic with a state machine
   = to showing an algorithm in state variable format

3) write down the circuit

In chap. 4, we viewed encoding as

$$\underset{1\times n}{\bar{c}} = \underset{1\times k}{\bar{m}}\ \underset{k\times n}{G}$$

let's do an example of a <u>cyclic</u> generator matrix.

let $k=3$ $n=7$ $\Rightarrow r=4$

for a cyclic code, a generator G can always be found in the form

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & g_3 & g_4 & 0 & 0 \\ 0 & g_0 & g_1 & g_2 & g_3 & g_4 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & g_3 & g_4 \end{bmatrix}$$

$$\bar{c} = (m_0 m_1 m_2) \cdot G$$

$$\bar{g} = (g_0\ g_1\ g_2\ g_3\ g_4\ 0\ 0)$$

let $g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + g_4 x^4$

Then we can re-express G as

$$G = \begin{bmatrix} g(x) \\ x\,g(x) \\ x^2 g(x) \end{bmatrix} \cdots$$

Now we can write

$$\bar{C} = (m_0\ m_1\ m_2) \begin{bmatrix} g(x) \\ x\,g(x) \\ x^2 g(x) \end{bmatrix}$$

$$= m_0 g(x) + m_1 x g(x) + m_2 x^2 g(x)$$

$$= (m_0 + m_1 x + m_2 x^2)\, g(x)$$

$$\boxed{C(x) = m(x) \cdot g(x)}$$

one issue: This code is non-systematic

In other words

$$\overline{C} = (c_0\, c_1\, c_2\, c_3\, c_4\, c_5\, c_6)$$

$$\neq (c_0\, c_1\, c_2\, c_3\, m_0\, m_1\, m_2)$$

systematic form

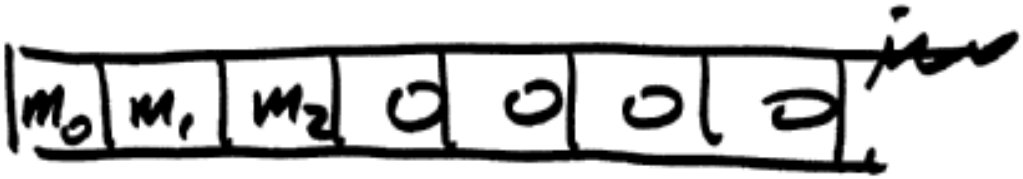
$C(x) = m(x)\, g(x)$ as we have done it here does <u>not</u> give us a $c(x)$ corresponding to a <u>systematic</u> $\overline{C}$

Is There an easy way to find a systematic form for our code?

Answer: yes.

To see how to set There, let's look at The problem in "hardware form"

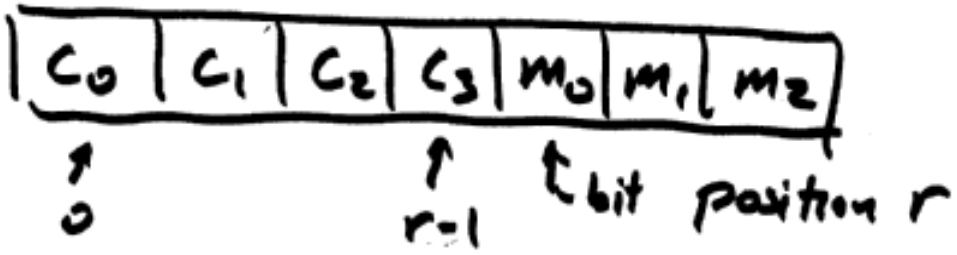
$$m(x) = m_0 + m_1 x + m_2 x^2$$



for a systematic codeword $C(X)$,

we want is

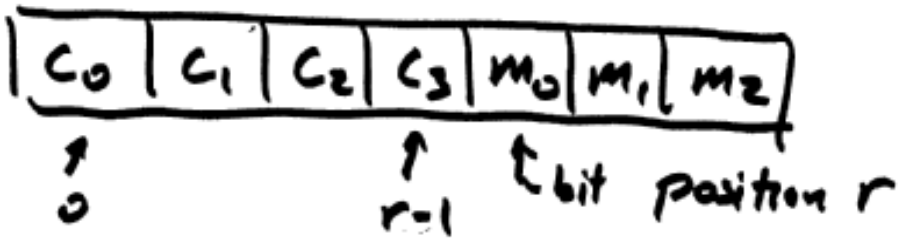
$$C(X) = \underbrace{C_0 + C_1 X + C_2 X^2 + C_3 X^3}_{r \text{ check bits}} + \underbrace{M_0 X^4 + M_1 X^5 + M_2 X^6}_{k \text{ message bits}}$$



for a systematic codeword $C(x)$,

we want is

$$C(x) = \underbrace{c_0 + c_1 x + c_2 x^2 + c_3 x^3}_{r \text{ check bits}} + \underbrace{m_0 x^4 + m_1 x^5 + m_2 x^6}_{k \text{ message bits}}$$



How do we shift $m(x)$ up into the r "top" positions in the register?

answer: multiply by x^r

Then we can say

$$c(x) = x^r \cdot m(x) + d(x)$$

$d(x)$ ← check bit Polynomial

with $\deg(d(x)) \leq r-1$

now, $\deg(g(x)) = r$ provided that $g_r \neq 0$

But if $g_r = 0$, then we'd really have a bigger k and a smaller r

it is always true for an (n, k) cyclic that $g_r = 1$, $r = n - k$

If $\deg(g(x)) = r$ what is the degree of $f(x)/g(x)$? $\deg[f(x)/g(x)] < r$

What if we make "$f(x)$" equal to $x^r m(x)$?

Then we'd be saying that

$$c(x) = x^r m(x) + \underbrace{\left[x^r m(x) / g(x) \right]}_{d(x)}$$

This satisfies our formal requirement for a systematic code.

The only question is: does this actually give us a cyclic code?

As it happens, if we pick any old $g(x)$ at random, the resulting set of $c(x)$ "codewords" will generally <u>not</u> form a cyclic code.

But, we <u>will</u> have a cyclic code <u>if</u> $g(x)$ satisfies one little property, namely

$$(x^n - 1)/g(x) = 0$$

remember the definition of polynomial division, e.g. $f(x) \div g(x)$ is defined

$$f(x) = Q(x)g(x) + r(x)$$

if $f(x) = x^n + 1$ (in $GF(2)[x]$)

and if $(x^n+1)/g(x) = 0 = r(x)$

Then we can say

$$x^n + 1 = h(x) \cdot g(x)$$

degree k ↗ ↖ degree $= r$

Summarize: systematic (n, k) cyclic code has

- $g(x)$ such that $\deg(g(x)) = r = n - k$
- $h(x)g(x) = x^n + 1$

(which by the way means $g_0 = 1, h_0 = 1$)

- $c(x) = x^r m(x) + [x^r m(x) / g(x)]$

$h(x)$ is also the generator poly for the dual code

Remember "syndromes"?

Let me propose the following method for doing syndrome calculation in a cyclic code:

$$s(x) \triangleq v(x)/g(x)$$

$$v(x) = c(x) + e(x)$$

If $e(x) = 0$ so that $v(x) = c(x)$, This gives us

$$s(x) = c(x)/g(x) = \left[x^r m(x) + d(x)\right]/g(x)$$

Using our ~~2nd~~ 2nd handy identity

$$[x^r m(x) + d(x)]/g(x) = [x^r m(x)]/g(x) + d(x)/g(x)$$

$$= d(x) + d(x) = 0$$

$$\therefore \quad 1(x) = c(x)/g(x) = 0$$

which is what we want.

If $e(x) \neq 0$, Then $1(x) = v(x)/g(x) = e(x)/g(x)$

EE 455
Lec 32

Last time, we said we could a systematic cyclic code as follows:

$$C(X) = X^r m(X) + [X^r m(X)]/g(X)$$

with $g(X)$ such that

$$(X^n - 1)/g(X) = 0$$

$$\Rightarrow \quad X^n - 1 = X^n + 1 = h(X)\, g(X) + 0$$

$$\deg(g(X)) = r = n - k$$

How do we generate this?

First we need $g(x)$.

One way to set $g(x)$ is to factor

x^n+1

Example: $n = 7$

$$x^7+1 = (x+1)(x^3+x+1)(x^3+x^2+1)$$

Suppose $r=3$. Then pick either

Suppose $r=4$: then pick $(x+1)\cdot g(x)$; $g(x)$

Suppose $r=5$: Too BAD

Once we've got $g(x)$, we could generate our $d(x)$ by table lookup.

What we do is build a remainder table

Example: $n = 7$, $r = 3$ $\quad (n, k) = (7, 4)$

$$c(x) = x^3 m(x) + [x^3 m(x)] / g(x)$$

$m_0 \Rightarrow m_0 x^3 \Rightarrow x^3 / g(x)$

$m_1 \Rightarrow m_1 x^4 \Rightarrow x^4 / g(x)$

$m_2 \Rightarrow m_2 x^5 \Rightarrow x^5 / g(x)$

$m_3 \Rightarrow m_3 x^6 \Rightarrow x^6 / g(x)$

} remainder table

Example 5.4.1: Construct a systematic (7, 4) cyclic code.

Solution: We previously found the factorization $x^7+1=(x+1)(x^3+x+1)(x^3+x^2+1)$. The generator polynomial must be of degree $r = n - k = 7 - 4 = 3$. Let our generator polynomial be

$$g(x)=x^3+x+1.$$

The codewords are the 16 polynomials defined by

$$c(x)=x^3(m_0+m_1x+m_2x^2+m_3x^3)/g(x)+x^3m(x)=d(x)+x^3m(x).$$

In example 5.3.2, we found the remainders for this $g(x)$ for the terms x^3, x^4, x^5, and x^6. Using these results and equation (5.3.2), we get the following code table.

$m(x)$	$c(x)$	$m(x)$	$c(x)$
0	0	x^3	$1+x^2+x^6$
1	$1+x+x^3$	$1+x^3$	$x+x^2+x^3+x^6$
x	$x+x^2+x^4$	$x+x^3$	$1+x+x^4+x^6$
$1+x$	$1+x^2+x^3+x^4$	$1+x+x^3$	$x^3+x^4+x^6$
x^2	$1+x+x^2+x^5$	x^2+x^3	$x+x^5+x^6$
$1+x^2$	$x^2+x^3+x^5$	$1+x^2+x^3$	$1+x^3+x^5+x^6$
$x+x^2$	$1+x^4+x^5$	$x+x^2+x^3$	$x^2+x^4+x^5+x^6$
$1+x+x^2$	$x+x^3+x^4+x^5$	$1+x+x^2+x^3$	$1+x+x^2+x^3+x^4+x^5+x^6$

We could do it this way, but there's a better way. To find this better way, we need to look at the mechanics of long division. What we will find is that calculating the remainder $P(X)$ can be expressed recursively using state variables and $\therefore$ $X^r m(X)/g(X)$ can be implemented as a state machine.

Example: $n = 7$, $r = 3$

$$g(x) = x^3 + g_2 x^2 + g_1 x + 1$$

$m(x)$ has deg. $(m(x)) \leq n-1 = 4-1 = 3$

$x^3 m(x)$ has degree $\leq n-1$

let's look at $x^{n-1} / g(x) = x^6 / g(x)$

by long division

$$x^3 + g_2 x^2 + g_1 x + 1 \;\overline{)\; x^6}$$

$$x^3$$

$$x^6 + g_2 x^5 + g_1 x^4 + x^3$$

$$g_2 x^5 + g_1 x^4 + x^3$$

← Partial remainder

define a vector $S_1 = \begin{bmatrix} g_2 \\ g_1 \\ 1 \end{bmatrix}$

1 cycle of the long division

$g_2 \cdot g_2 = g_2$ in GF(2)

next cycle:

$$
\begin{array}{r}
g_2 x^2 \\
\hline
x^3+g_2x^2+g_1x+1 \;\big|\; g_2x^5+g_1x^4+x^3 \\
\qquad\qquad g_2x^5+g_2x^4+g_1g_2x^3+g_2x^2 \\
\hline
(g_1+g_2)x^4+(1+g_1g_2)x^3+g_2x^2
\end{array}
$$

$$
S_2 = \begin{bmatrix} g_1+g_2 \\ 1+g_1g_2 \\ g_2 \end{bmatrix} \equiv \underbrace{\begin{bmatrix} g_2 & 1 & 0 \\ g_1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} g_2 \\ g_1 \\ 1 \end{bmatrix}}_{S_1}
$$

What do you suppose we'll get from the 3rd cycle of long division?

$$x^3+g_2x^2+g_1x+1 \;\overline{)\;(g_1+g_2)x^4+(1+g_1g_2)x^3+g_2x^2}\;\;(g_1+g_2)x$$

what do you think the partial remainder will be?

$$S_3 = \Gamma S_2$$

and in general, the t^{th} cycle will give $S_t = \Gamma S_{t-1}$

To calculate $x^6 / g(x)$

Ex: let $g(x) = x^3 + x + 1$ $\quad g_2 = 0$, $g_1 = 1$

$$\Gamma = \left[\begin{array}{c|cc} g_2 & 1 & 0 \\ g_1 & 0 & 1 \\ 1 & 0 & 0 \end{array}\right] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} g_2 \\ g_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$$S_3 = \Gamma S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{matrix} \leftarrow x^2 \\ \leftarrow x^1 \\ \leftarrow x^0 \end{matrix}$$

4 shifts and $k = 4$; this means that the poly. represented by S_k has deg of $r - 1 = 2$

$$x^6 / (x^3 + x + 1) = x^3 + 1$$

Does This trick generalize?

Does it work for

$$g(x) = x^r + g_{r-1}x^{r-1} + g_{r-2}x^{r-2} + \cdots + g_1 x + 1$$

?

Yep.

$$\Gamma = \begin{bmatrix} g_{r-1} & & \\ g_{r-2} & I_{(r-1)\times(r-1)} & \\ \vdots & & \\ g_1 & & \\ 1 & 0\ 0\ \cdots\ 0 & \end{bmatrix} \leftarrow \text{State Matrix}$$

Now, what if $m(x)$ is general

$$m(x) = m_{n-1}x^{n-1} + m_{n-2}x^{n-2} + \cdots + m_1 x + m_0$$

$$x^r m(x) = m_{n-1}x^{n-1} + m_{n-2}x^{n-2} + \cdots + m_1 x^{r+1} + m_0 x^r$$

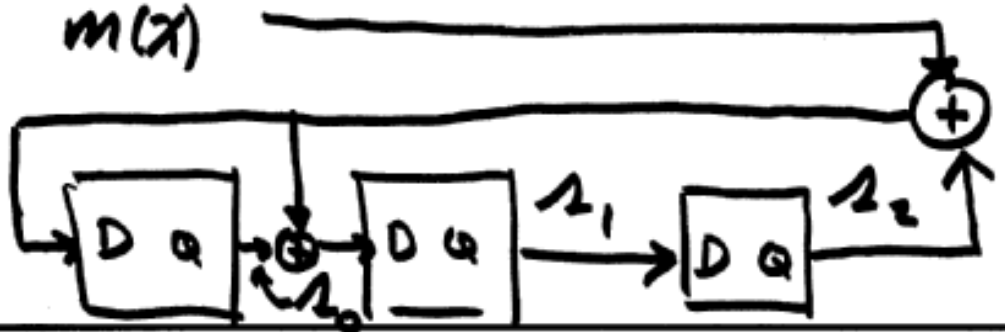
Our "state vector" containing the partial remainders (shifting in $m(x)$ one bit at a time) generalizes to

$$S_t = \Gamma S_{t-1} + \begin{bmatrix} g_{r-1} \\ g_{r-2} \\ \vdots \\ 1 \end{bmatrix} \cdot m_{n-t} \qquad S_0 = \bar{0}$$

This implies we can build our $x^r m(x)/g(x)$ calculator as follows:

EX. $g(x) = x^3 + x + 1$: $S_0 = \overline{0}$

$$S_t = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} S_{t-1} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} m_{n-t} \qquad \left\{ S_t = \begin{bmatrix} s_2 \\ s_1 \\ s_0 \end{bmatrix}_t \right.$$



$1+1=0$

$n = 7$
$r = 3$
$k = 4$

Enduring Big Ideas:

1) $g(x) = x^r + g_{r-1}x^{r-1} + \cdots + g_1 x + 1$

$$\Gamma = \begin{bmatrix} g_{r-1} & & \\ g_{r-2} & & \\ \vdots & & I_{(r-1)\times(r-1)} \\ g_1 & & \\ 1 & 0 \cdots & 0 \cdots 0 \end{bmatrix}$$

$$S_t = \Gamma S_{t-1} + \begin{bmatrix} g_{r-1} \\ g_{r-2} \\ \vdots \\ 1 \end{bmatrix} m_{k-t} \qquad S_0 = \bar{0}$$

shift k times

2)

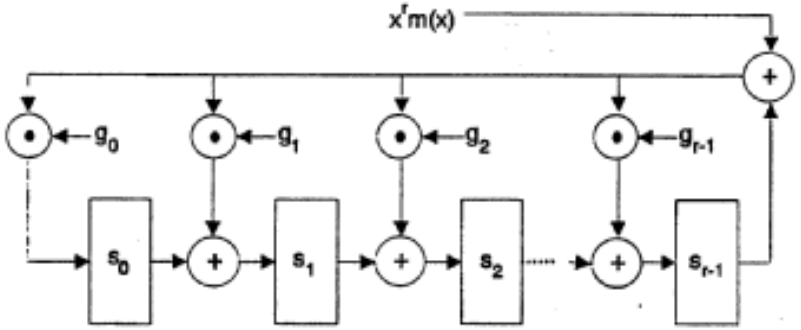
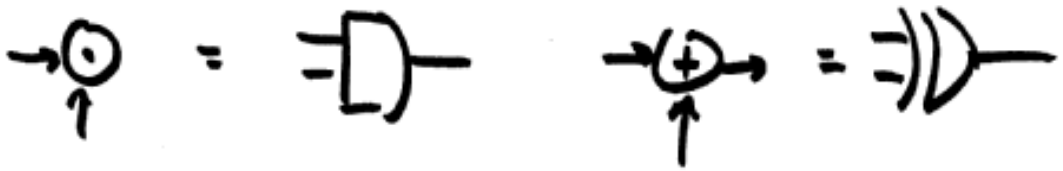


Figure 5.4.1: Divide by g(x) Circuit



University of Idaho

17

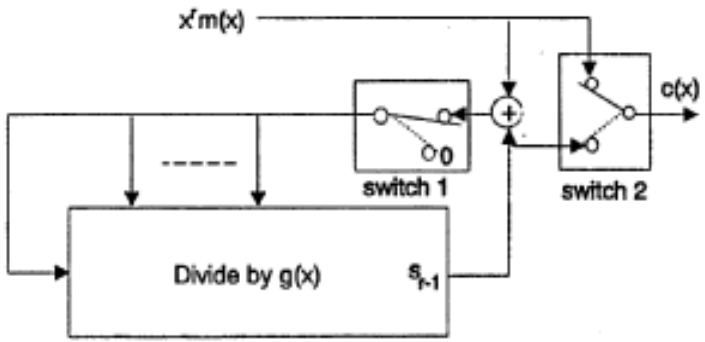


Figure 5.4.2: Systematic Encoder

after k shifts, change the switches

Decoding systematic cyclic block codes

Codeword:

$$c(X) = X^r m(X) + d(X)$$

where $d(X) = [X^r m(X)] / g(X)$

$\deg(g(X)) = r$; $(X^n + 1)/g(X) = 0$

We can write the received block as

$$v(X) = c(X) + e(X)$$

where

$$e(x) = e_0 + e_1 x + e_2 x^2 + \cdots + e_{n-1} x^{n-1}$$

$e_i = 0 \Rightarrow$ no error

$e_i = 1 \Rightarrow$ error

For decoding, we will use the syndrome decoding method

gen. linear

$\bar{s} = \bar{v} H^T$

cyclic codes

let

$s(x) = [x^r v(x)] / g(x)$

$$a(x) = [x^r v(x)]/g(x)$$

$$= [x^r c(x) + x^r e(x)]/g(x)$$

$$= [x^r c(x)]/g(x) + [x^r e(x)]/g(x)$$

$$= [(x^r/g(x)) \cdot (c(x)/g(x))]/g(x)$$

$$+ [x^r e(x)]/g(x)$$

Now

$$c(x)/g(x) = \left[x^r m(x) + d(x)\right] / g(x)$$

$$= \left[x^r m(x)\right]/g(x) + d(x)/g(x)$$

$$= d(x) + d(x) = 0$$

$\therefore$ $\boxed{1(x) = \left[x^r e(x)\right]/g(x)}$

if $e(x) \in C$ Then $1(x) = 0$ undetectable errors

$$s(x) = [x^r e(x)] / g(x)$$

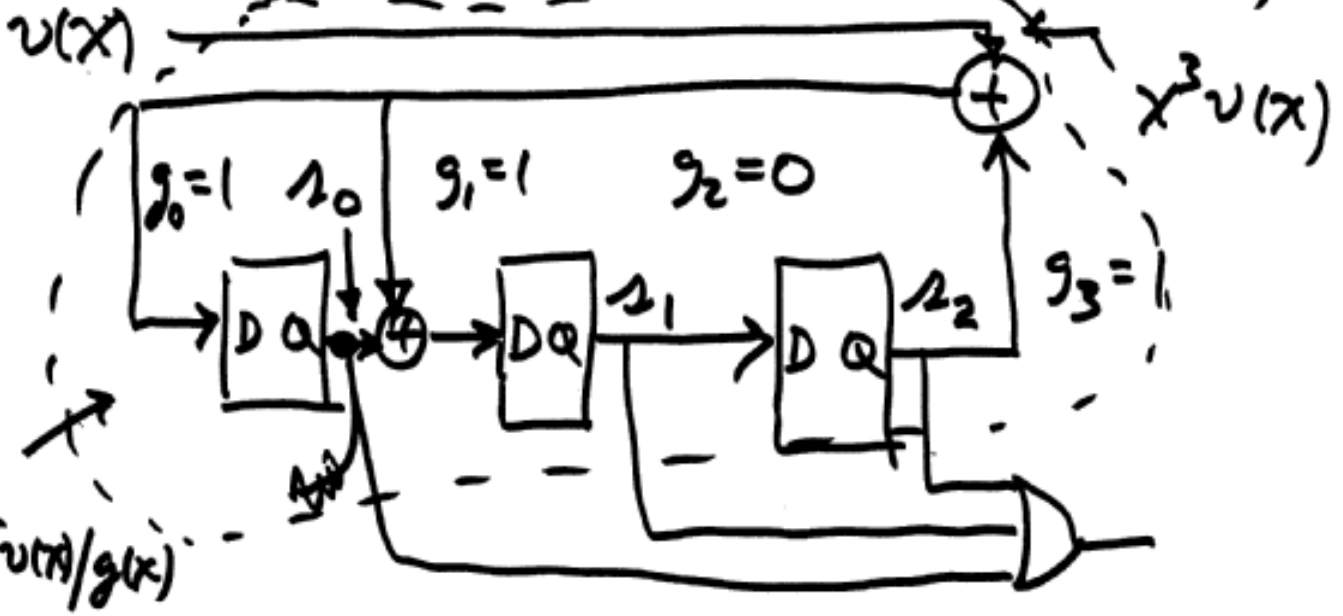
$$= s_0 + s_1 x + s_2 x^2 + \cdots + s_{r-1} x^{r-1}$$

If all we want is error <u>detection</u> then

$s(x) \neq 0$ tells us we have an error

Error detect circuit

(suppose $n=7$, $r=3$, $g(x) = x^3+x+1$)



after $n=7$ shift cycles, OR-output $=1$ if we detect an error

State variable equation for the circuit

$$S_t = \begin{bmatrix} z_2 \\ z_1 \\ z_0 \end{bmatrix}$$

$$S_t = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} S_{t-1} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} v_{n-t}$$

for $t = 1$ to n

with $S_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

$$S_t = \Gamma S_{t-1} + \begin{bmatrix} g_{n-1} \\ g_{n-2} \\ \vdots \\ g_0 \end{bmatrix} v_{n-t}$$

How about error correction?

Codes are designed to correct up to some maximum number, t_c, of errors

One way to do it could be to build a syndrome table that maps

$$s(x) \Longrightarrow e(x)$$

How big is the lookup table in this method?

One entry per correctable error

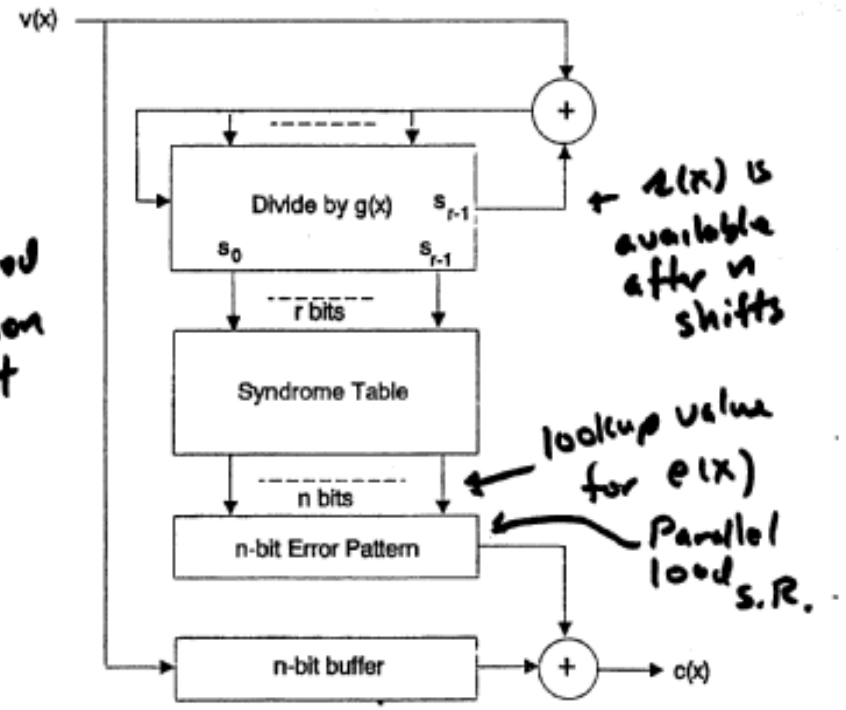
Suppose the code corrects t_c errors

$$d_{min} \geq 2t_c + 1$$

$w_H(\bar{e}) = 1$: $n_1 = n = \binom{n}{1}$

$w_H(\bar{e}) = 2$: $n_2 = n \cdot (n-1)$

The total table size $\sim n^{t_c}$



Pipelined correction circuit

← s(x) is available after n shifts

lookup value for e(x)

Parallel load S.R.

Figure 5.4.3: Error Correction

Secret to simplifying even more is:

Meggitt's Theorem (Th. 5.3.1)

$$g(x)h(x) = x^n + 1 \quad = x^n - 1$$

Suppose that $f(x)/g(x) = p(x)$

Then $$[xf(x) \bmod (x^n-1)]/g(x) = [xp(x)]/g(x)$$

a cyclic property to syndromes

Example:

Suppose we have $n = 7$, $k = 4$,

$g(x) = x^3 + x + 1$

also suppose $e(x) = x^5$

oops! where's x^r? wells!?

0	0	0	0	0	1	0

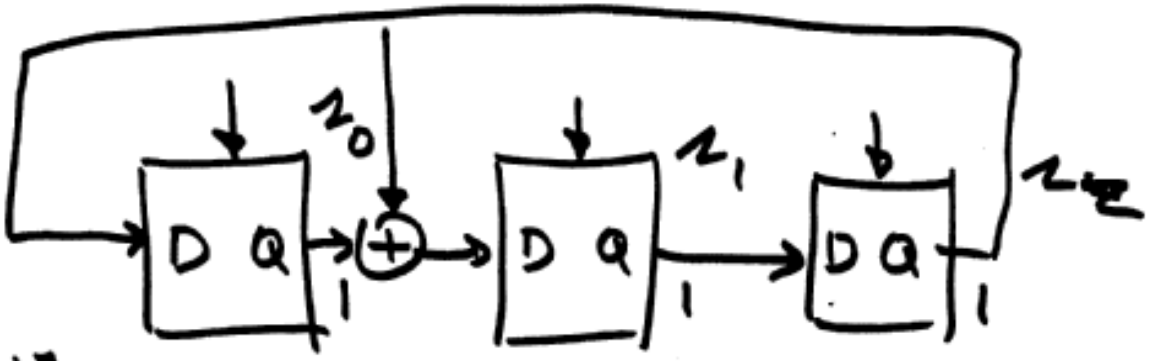
$$
\begin{array}{r|l}
 & x^2 + 1 \\
\hline
x^3 + x + 1 & x^5 \\
 & x^5 + x^3 + x^2 \\
\hline
 & x^3 + x^2 \\
 & x^3 + x + 1
\end{array}
$$

$\Rightarrow P(x) = x^2 + x + 1$

what if we pre-load this $z(x)$

$$z(x) = x^2 + x + 1$$

into another $\div$ by $g(x)$ circuit



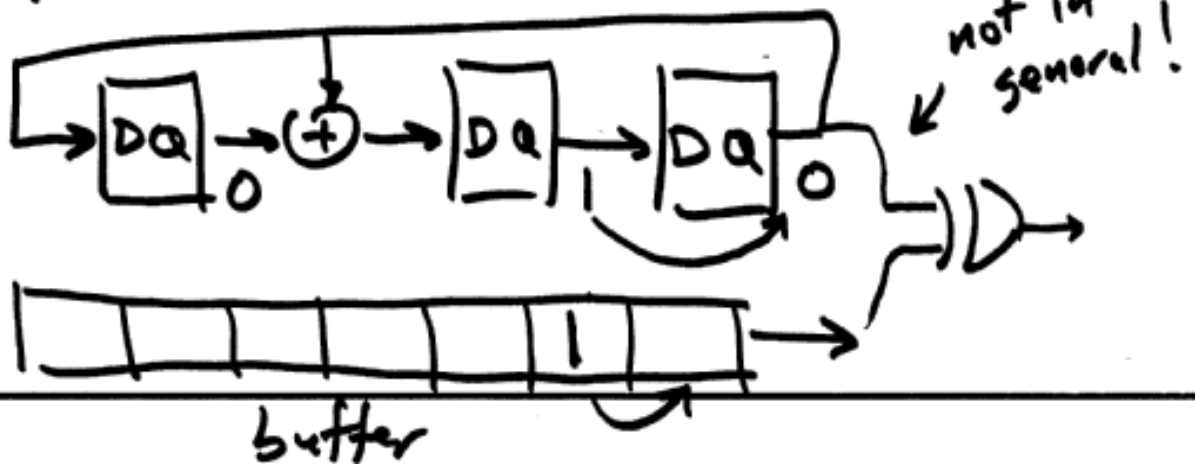
$$S_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad S_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Fix Rick's glitch !

$e(x) = x^5$

$x^r e(x) = x^3 \cdot x^5 = x^8$

$x^8 / (x^3 + x + 1) = z(x) = x$



To build up a general solution, we do this trick:

let $\mathcal{E} = \{ e(x) \mid 0 < w_H(\bar{e}) \leq t_c \}$

define 2 <u>subsets</u> of $\mathcal{E}$

$$\mathcal{E}_{meg} \triangleq \{ e(x) \in \mathcal{E} \mid e_{n-1} = 1 \}$$

$$\mathcal{E}_{shift} \triangleq \{ e(x) \in \mathcal{E} \mid e_{n-1} = 0 \}$$

if $e(x) = x^{n-1}$

then $\left[x^r e(x)\right] / g(x) \equiv x^{r-1}$

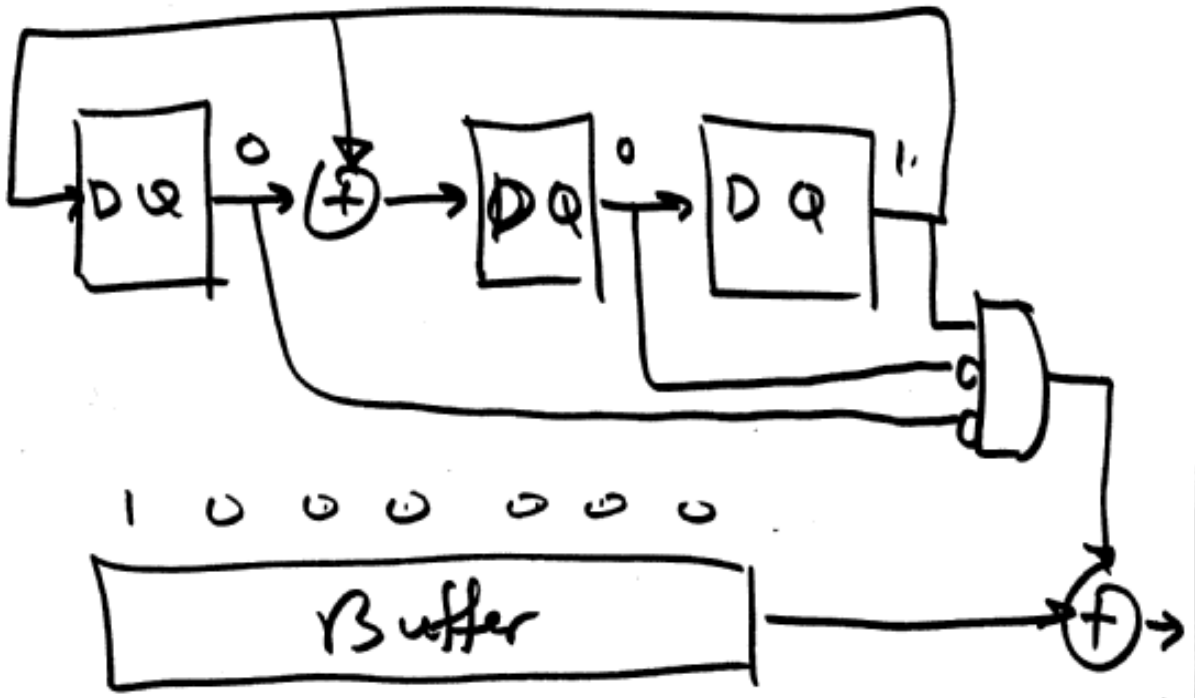
Consider a Hamming code:

$t_c = 1$ $\quad \mathcal{E}_{meg} = \{ x^{n-1} \}$

syndrome for x^{n-1} is $z(x) = x^{r-1}$

(7,4) H.C. has $r = 3 \Rightarrow z(x) = x^2$

University of Idaho



all the $e(x) \in \mathcal{E}_{shift}$ have syndromes that "shift" to $z(x) = x^2$ when they come out