Not sure what this is, but it ends with my current drawing of one of the division how works. I guess there was some setup work to do before that drawing could be made. I do remember now that I had to decide exactly WHAT I was going to divide into what before doing the drawings.

Can I talk about remainder without my "proper" polys?

Let's go back to

$$O(z) = I(z) / H(z)$$

$$I(z) = i_0 + i_1 z^{-1} + i_2 z^{-2} + ...$$

$$O(z) = o_k z^{-k} + o_{k+1} z^{-k-1} + o_{k+2} z^{-k-2} + ...$$

Why is there no remainder showing in this equation? It is because O(z) is not truncated.

Suppose we write

$$O(z) = [o_k z^{-k} + o_{k+1} z^{-k-1} + o_{k+2} z^{-k-2} + ... o_{k+s} z^{-k-s}] + [o_{k+s+1} z^{-k-s-1} + ... \text{ to } \infty]$$

$$= O_{tr}(z) + \text{tail}(z)$$

Then we have

$$O(z) = I(z) / H(z) = O_{tr}(z) + \text{tail}(z) = O_{tr}(z) + \text{Rem}(z)/H(z)$$

The whole idea of remainder does not work well here because Rem(z) could have any degree you want, there is no shape or form to anything.

Remainder Concept #1

Let us now consider an input stream which has this form

.....0,0,0, i_0, i_1, ...i_r,0,0,0.....

The input polynomial in this case (the Z Transform of the above signal) is,

$$I(z) = i_0 + i_1 z^{-1} + i_2 z^{-2} + ... + i_r z^{-r} . \qquad (1.3.3)$$

We can rewrite this input sequence as:

$$I(z) = z^{-r} [i_0 z^r + i_1 z^{r-1} + i_2 z^{r-2} + ... + i_r] \equiv z^{-r} I_r(z). \qquad (1.3.4)$$

Here we have defined $I_r(z)$ to be the *proper* polynomial shown in [..], having degree r.

Since H(z) has degree k, and since I(z) has degree 0 (largest exponent in (1.3.3)), we know from (1.2.5) that O(z) must have degree 0-k = -k. This means that its first k symbols all vanish. This is easily interpreted in terms of Figure 1.1 since one has to wait k clocks before any non-zero symbols emerge from the circuit output, after i_0 hits the input. (This will also be true of our second divider circuit to be given below.) Thus we have, with a common time base, and assuming r > k,

i_j: .....0,0,0, i_0, i_1, ...i_r,0,0,0.....

o_j .....0,0,0, 0,0,0.....o_k,o_{k+1} .....

The output polynomial is then

$$O(z) = o_k z^{-k} + o_{k+1} z^{-k-1} + \ldots\ldots \qquad (1.3.5)$$

$$= z^{-r} [\ o_k z^{r-k} + o_{k+1}\ z^{r-k-1} + \ldots\ldots\ ] \quad = \quad z^{-r} O_{r-k}(z)$$

where $O_{r-k}(z)$ is a (generally) infinite and non-proper polynomial of degree r-n (since in general it will have negative powers of z).

Since a common factor z^{-r} has been extracted in the last two definitions, we can write

$$O(z) = I(z)/H(z) \qquad \Rightarrow$$

$$O_{r-k}(z) = I_r(z)/H(z)$$

where now the polynomial ratio $I_r(z)/H(z)$ is completely conventional since both $I_r(z)$ and H(z) are proper polynomials, of degree r and k respectively. We can then talk about a quotient Q and a remainder R as follows,

$$I_r(z)/H(z) = Q(z) + R(z)/H(z)$$

where

$$Q(z) = q_{r-k}z^{r-k} + q_{r-k=1}z^{r-k-1} + \ldots. q_1 z + q_0 \qquad \text{// r-k+1 terms, degree r-k}$$

$$R(z) = r_{k-1}z^{k-1} + r_{k-2}z^{k-2} + \ldots. + r_1 z + r_0 \qquad \text{// k terms, degree k-1}$$

But then

$$O(z) = I(z) / H(z) = z^{-r} I_r(z) / H(z) = z^{-r}[\ Q(z) + R(z)/H(z)]$$

$$= [z^{-r}Q(z)] \quad + \quad [z^{-r}R(z)] / H(z)$$

$$= [\ q_{r-k}z^{-k} + q_{r-k-1}z^{-k-1} + \ldots. q_1 z^{-r+1} + q_0 z^{-r}\ ] \ + \ [z^{-r} R(z)] / H(z)$$

Comparing this to

$$O(z) = [\, o_k z^{-k} + o_{k+1} z^{-k-1} + o_{k+2} z^{-k-2} + \ldots + o_r z^{-r}] + [\, o_{r+1} z^{-r-1} + o_{r+2} z^{-r-2} + \ldots \,]$$

we can identify the first r-k+1 terms of O(z) with $[z^{-r}Q(z)]$ so that

$o_k = q_{r-k}$
$o_{k+1} = q_{r-k-1}$
.....
$o_{k+s} = q_{r-k-s}$ => $o_j = q_{r-j}$ for j = k to r
.....
$o_r = q_0$

The remaining infinite number of terms of O(z) are then associated with $[z^{-r} R(z)] / H(z)$

$$[z^{-r} R(z)] / H(z) = o_{r+1} z^{-r-1} + o_{r+2} z^{-r-2} + \ldots\ldots$$

which says

$$R(z) / H(z) = o_{r+1} z^{-1} + o_{r+2} z^{-2} + \ldots\ldots$$

In the above discussion point of view, there is only one remainder, it is R(z). There is no sequence of remainders.

The non-terminating Remainder in polynomial division.

In Chapter 1 we discussed the use of either Figure 1.1 or Figure 1.5 as a polynomial divider. Both circuits perform the same function. In Figure 1.1, the remainder is "hidden", but in Figure 1.5 it is exposed: the state vector always holds the "current remainder". If we let the divider continue for n clocks after the last non-zero input symbol i_r has arrived, we showed that

$$I_{r+n}(z) / H(z) = O_{r+n-k}(z) + R^{(n)}(z)/H(z) \qquad (1.3.10)$$

where

$$I(z) = z^{-r-n} I_{r+n}(z) = z^{-r-n} [\, i_0 z^{r+n} + i_1 z^{r+n-1} +$$

$$I_r(z) / H(z) = O_{r-k}(z) + R^{(0)}{}_{k-1}(z)/H(z) \qquad \textit{(1.3.10)}$$

$$I(z) = z^{-r} I_r(z) \qquad \textit{(1.3.4)}$$
$$O(z) = z^{-r} O_{r-k}(z) \qquad \textit{(1.3.5)}$$
$$O(z) = I(z) / H(z) \qquad \textit{(1.2.5)}$$

where i_r was last non-zero input symbol.

To review, $I_r(z)$ was an Input polynomial of degree r, H(z) was the divisor polynomial of degree k, and $O_{r-k}(z)$ was the Output quotient polynomial of degree r-k. The Remainder $R^{(0)}_{k-1}(z)$ is at most of degree k-1 . If some of its leading coefficients happen to vanish, it is of degree less than k-1.

In Chapter 1 we used upper-case polynomial names to stress the notion of the Z Transform, and for the same reason we used the variable z. We were also trying to show explicitly how the degrees of all the polynomials come out right.

Here we shall ignore all this baggage and rewrite the above in the simple form

$$i_0(x)/ h(x) = o_0(x) + r_0(x)/h(x) .$$

We have not changed anything, we are just simplifying the notation. The 0 subscript is a relative time index. We think now of time t=0 just after the last non-zero symbol of the input stream has been shifted in to the divider. If the input stream is always zero from this point on, we can think of the circuit as the *shift register generator* discussed above which has its register state initialized with the remainder appearing in the above equation, $r_0(x)$. That is, the coefficients of this polynomial are the starting contents of the k registers. After one clock, we have:

$$i_1(x) / h(x) = [\ x\ i_0(x)\ ] / h(x) = o_1(x) + r_1(x)/h(x)$$

The input polynomial $i_1(x) = x\ i_0(x)$, because we have added a trailing 0 to the input symbol stream. After s clocks we would have

$$i_s(x) / h(x) = [\ x^s\ i_0(x)\ ] / h(x) = o_s(x) + r_s(x)/h(x)$$

If at some point the remainder vanished, the subsequent quotient symbols would be zeros. Conversely, if the remainder were never to vanish, the quotient symbol stream would go on forever.

Fact 9: If h(x) of degree k is irreducible over GF(p), and if the first remainder

$$r_0(x) = \text{Rem}[\ i_0(x) / h(x)\ ]$$

is non-zero, then all subsequent remainders

$$r_s(x)= \text{Rem}[\ x^s\ i_0(x) / h(x)\ ]$$

are non-zero, and the quotient sequence never terminates. Moreover, the sequence of remainders has period N which is the state vector period discussed above. This period N divides evenly into $p^k - 1$. Furthermore, if h(x) is a primitive polynomial of $GF(p^k)$, then the state vector period is exactly $p^k - 1$.

Proof: Once we identify the remainder with the shift register state, everything follows directly from the Facts given above.

Comment: In Appendix A we show an example of polynomial division (coefficients in Mod(10)) where the quotient polynomial never terminates and has a repeating sequence of symbols. Fact 9 gives us many examples of the same thing, where now coefficients are in GF(p) and where h(x) is an irreducible polynomial of order k.

Fact 10: The *state vector period* of a Figure 1.5 shift register generator is the same as the *period* of h(x) .

Proof: The *period* of a monic irreducible h(x) was defined rather mysteriously in Ref [GA] Chapter 5 as being the smallest value of n such that h(x) divides evenly into $x^n - 1$. Let us assume that h(x) has *period* n, so we can then write $(x^n - 1)/h(x) = g(x)$, where g(x) is whatever quotient polynomial we get by this division. Then consider the remainder sequence noted above, for time s clocks after the input polynomial coefficients resume being 0,

$$r_s(x) = \text{Rem}[\ x^s\, i_0(x) / h(x)\ ]$$

Then we can evaluate

$$r_{s+n}(x) - r_s(x) = \text{Rem}[\ (x^n - 1)x^s\ i_0(x) / h(x)\ ] = \text{Rem}[\ x^s\ g(x)i_0(x)] = 0.$$

This shows that $r_{s+n}(x) = r_s(x)$, which says that the state vector period is n. Recall that the *period* n refers to the smallest power n such that $(x^n - 1)/h(x) = g(x)$. Similarly, the *state vector period* is the smallest integer n that makes the register state (= remainder) repeat. Thus, we see that these two entities are one in the same.

Summary: This section has focused on the state vector as being an element of a Galois Field $GF(p^k)$. For the Figure 1.5 circuit, this state vector can be identified with the remainder of polynomial division. We learned that the state vector period N of a Figure 1.5 style shift register generator is determined by the nature of h(x).

We did not in this section have much to say about any particular register. For example, the output symbol sequence is basically the contents of the rightmost register in Figure 1.5. One wonders what the periodicity of the contents of a particular register might be, and thus, one wonders about the periodicity of the output data stream of a shift register generator.

For the moment, we settle for the following:

Fact 11: The period of any particular register of the Figure 1.5 circuit must integrally divide the state vector period N.

Proof: Certainly the period of one register cannot exceed the period of the register set as a whole. If the whole register state repeats with period N, then so must any one register. However, it is easy to imagine that a particular register might have a smaller period. To be compatible with the state vector period, the state vector period would have to consist of an integral multiple of these smaller periods.

Corollary: The *output period* of a Figure 1.5 shift register generator must be a divisor of the state vector period N.

In the next section, we shall find a much stronger statement about the output period.

****

If we trivially rewrite the above division as

$$[z^5O(z)] = [z^5I(z)] / H(z)$$

and then rename things so $z^5O(z) \rightarrow O(z)$ and $z^5I(z) \rightarrow I(z)$, *and* if we have the input stream terminate with i_5, then the above "layout" for long division is exactly the same but the powers are all shifted by +5 :

$$\begin{array}{rl} & o_3z^2 + o_4z^1 + o_5 \\ \hline h_3z^3 + h_2z^2 + h_1z^1 + h_0z^0 \;|\; & i_0z^5 + i_1z^4 + i_2z^3 + i_3z^2 + i_4z^1 + i_5 \\ & -o_3h_3z^5 - o_3h_2z^4 - o_3h_1z^3 - o_3h_0z^2 \\ & \text{---} \end{array}$$

Current Dividend #1

$$\begin{array}{rl} & a_1z^4 + a_2z^3 + a_3z^2 + i_4z^1 + i_5 \\ & -o_4h_3z^4 - o_4h_2z^3 - o_4h_1z^2 - o_4h_0z^1 \\ & \text{---} \end{array}$$

Current Dividend #2

$$\begin{array}{rl} & b_2z^3 + b_3z^2 + b_4z^1 + i_5 \\ & -o_5h_3z^3 - o_5h_2z^2 - o_5h_1z^1 - o_5h_0 \\ & \text{---} \end{array}$$

Current Dividend #3

$$c_3z^2 + c_4z^1 + c_5$$

We now have a the division of two "proper" polynomials. In this case our timing diagram above looks like this:

		L1	L2	L2	C1	C2	C3	
	↑	↑	↑	↑	↑	↑	↑	
i:	0	i_0	i_1	i_2	i_3	i_4	i_5	
o:	0	0	0	0	o_3	o_4	o_5	o_5
current dividend:					#0	#1	#2	#3

After clock C3, the output bus has symbol o_5 and the registers hold Current Dividend #3 which is in fact **the remainder polynomial**, since this is the first current dividend of degree less than that of the divisor H(z). The remainder is arranged such that $(q_3, q_2, q_1) = (c_3, c_4, c_5)$ so the highest power coefficient c_3 is sitting in the rightmost register q_3.

Application of another clock C4 will destroy this remainder. However, the registers after clock C4 will hold the remainder of a *new* division problem -- one in which a new symbol i_6 is added to the input stream. In this case we write things using

$$[z^6O(z)] = [z^6I(z)] / H(z)$$

and here is the new diagram

$$\begin{array}{l} \\ h_3z^3 + h_2z^2 + h_1z^1 + h_0z^0 \end{array} \begin{array}{l} o_3z^3 + o_4z^2 + o_5z^1 + o_6 \\ \hline | \; i_0z^6 + i_1z^5 + i_2z^4 + i_3z^3 + i_4z^2 + i_5z^1 + i_6 \\ -o_3h_3z^6 - o_3h_2z^5 - o_3h_1z^4 - o_3h_0z^3 \\ \text{---} \end{array}$$

Current Dividend #1

$$\begin{array}{l} a_1z^5 + a_2z^4 + a_3z^3 + i_4z^2 + i_5z^1 + i_6 \\ -o_4h_3z^5 - o_4h_2z^4 - o_4h_1z^3 - o_4h_0z^2 \\ \text{---} \end{array}$$

Current Dividend #2

$$\begin{array}{l} b_2z^4 + b_3z^3 + b_4z^2 + i_5z^1 + i_6 \\ -o_5h_3z^4 - o_5h_2z^3 - o_5h_1z^2 - o_5h_0z^1 \\ \text{---} \end{array}$$

Current Dividend #3

$$\begin{array}{l} c_3z^3 + c_4z^2 + c_5z^1 + i_6 \\ -o_6h_3z^3 - o_6h_2z^2 - o_6h_1z^1 - o_6h_0z^0 \\ \text{-----------------------------------} \end{array}$$

Current Dividend #4

$$d_4z^2 + d_5z^1 + d_6z^0$$

and the new timing diagram

		L1	L2	L3	C1	C2	C3	C4	
	↑	↑	↑	↑	↑	↑	↑	↑	
i:	0	i_0	i_1	i_2	i_3	i_4	i_5	i_6	
o:	0	0	0	0	o_3	o_4	o_5	o_5	o_6
current dividend:					#0	#1	#2	#3	#4

Now after clock C4 the registers contain the Current Dividend #4 which is in fact the remainder of this new division problem, and now we have $(q_3, q_2, q_1) = (d_4, d_5, d_6)$.

Even if it happens that $i_6 = 0$, we still have a new division problem with a new remainder when we clock in $i_6 = 0$. For example, suppose the first proper division problem was this (as shown above)

$(i_0z^5 + i_1z^4 + i_2z^3 + i_3z^2 + i_4z^1 + i_5)/h(z)$ remainder = $c_3z^2 + c_4z + c_5$

If we then clock in i_6 and it happens to be zero, the new proper division problem is this:

$(i_0z^6 + i_1z^5 + i_2z^4 + i_3z^3 + i_4z^2 + i_5z + 0)/h(z)$

$= \{ z\,(i_0z^5 + i_1z^4 + i_2z^3 + i_3z^2 + i_4z^1 + i_5) \} / h(z)$ remainder = $d_4z^2 + d_5z + d_6$

where $d_6 = -\,o_6h_0$. The remainder is different because the dividend has one higher degree and so the division process has to continue through one more current dividend to get a current dividend of degree less than the original dividend.

Conclusion: If we have an endless input stream i_0, i_1..... , then we can regard the contents of the registers of a Type B divider at any clock as holding the remainder of a certain proper polynomial division problem, while the output bus o will have delivered the quotient for that problem. Here is a list of these division problems for the case k = 3, where the first three rows correspond to 0 quotient and involve only the loading clocks:

I(z)	Remainder	Quotient	Clk
i_0	i_0	0	L1
$i_0z + i_1$	$i_0z + i_1$	0	L2
$i_0z^2 + i_1z + i_2$	$i_0z^2 + i_1z + i_2$	0	L3
$i_0z^3 + i_1z^2 + i_2z + i_3$	$a_1z^2 + a_2z + a_3$	o_3	C1
$i_0z^4 + i_1z^3 + i_2z^2 + i_3z + i_4$	$b_2z^2 + b_3z + b_4$	$o_3z + o_4$	C2
$i_0z^5 + i_1z^4 + i_2z^3 + i_3z^2 + i_4z + i_5$	$c_3z^2 + c_4z + c_5$	$o_3z^2 + o_4z + o_5$	C3
$i_0z^6 + i_1z^5 + i_2z^4 + i_3z^3 + i_4z^2 + i_5z + i_6$	$d_4z^2 + d_5z + d_6$	$o_3z^3 + o_4z^2 + o_5z + o_6$	C4
etc.			