CCITT SGXV **Doc. # 476**
Working Party XV/4
Specialists Group on Coding for Visual Telephony

Source : NL, FRG, UK
Title : Comparison Bose-Chaudhuri-Hocquenghem BCH and Reed Solomon

Enclosed you will find a document which gives an description of the capabilities of two error correcting codes being the BCH and the RS.
First a brief description of the block codes is given after which the typical capabilities are discussed.
A discussion is given of the burst capabilities of the BCH code, and the tradeoff between interleaving and coding[1] delay. References of the available hardware for both systems are annexed to this report.

[1] Channel coding delay

2Kp

1 Summary

In this contribution alternant codes are examined specifically the BCH and the RS code. BCH and RS codes are becoming more frequently used due to the availability of VLSI components. Both codes have a powerful random and burst correcting ability. The codes also leave the data in its original form which allows video decoders to be specified with "optional" error correction.
The BCH code can be seen as a special case of the RS code. The RS code has burst correction capabilities due to its symbol based character whereas the BCH is more capable of correcting random errors. The BCH code can be adapted towards the burst error correcting capabilities and this will be shown in the contribution.
The complexity of the BCH and the RS will be discussed.

2 Conclusion

At the moment two candidates are proposed for the error correction. The BCH code and the RS code. Both BCH and RS error correction coding schemes are well suited for random error correction of compressed video data. In general the BCH is a binary code which is designed for the correction of random errors, and the RS code which is symbol based i.e. capable of correcting bursts. A disadvantage of the burst correcting capability is that if the burst is spread over several symbols the RS code is not capable of correcting this burst, whereas the BCH is still able to correct the burst because the position of the burst is not symbol dependant.

いしりまわす

Error correction should be optional which has some constraints where to position the parities, e.g. one should not tamper with the synchronisation capability.
Some constraints and requirements are itemized;

- The RS code is a subset of the BCH code.
- The coding process of the BCH is simple to implement.
- The decoding process of the BCH codes is relatively simple.
- The RS code is more suitable for concatenated codes.
- The decoding process of the RS is more complex.
- The burst capabilities of The RS and the BCH are similar but the occurrence of the burst in the bit stream could give the BCH to be more appropriate.
- The sensitivity to burst errors per bit in the case of VLC based coding, is smaller compared to the sensitivity to random errors per bit.

In the case of an fixed number of parity bits, the first priority should be to correct random errors.

The choice which code is going to be adopted should first be verified with various channel model. There is insufficient information whether the errors on the channels are bursty or random.

Using an error correcting code one has to bear in mind that the synchronization capabilities of the system could decreases.

The amount of delay should be as minimum as possible bearing in mind that the adopted error correcting code operate for videotelephony as well as videoconferencing.

25/

Contents

2f2

3 Introduction

The Reference Model simulations have been carried out in an error free environment. With the introduction of a service the influence of the network on the codec design becomes more and more important. The possible customer does not accept a degraded or a completely distorted image. He is used to a television quality. Due to all kinds of redundancy reduction techniques the decoding process is more susceptible to error. These error can occur inside the codec[2] and errors which occur on the transmission path from the encoder to the decoder see 1.
For the moment the second type of errors i.e. errors on the transmission path are discussed.

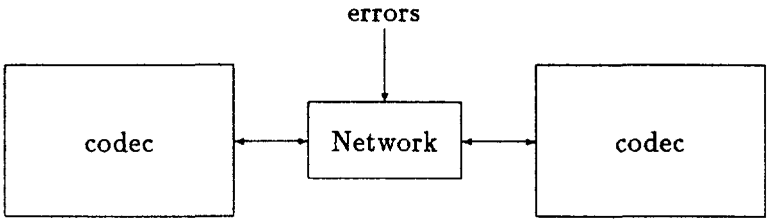


Figure 1: Communication path

Error resilience can be divided in three parts (in order of priority) :

- Synchronization

  Due to the use of variable length codes, uncorrectable errors may result in loss of synchronization. As a result of this two actions have to be taken in the decoder. The first action is to re-synchronize and continue decoding from the next detected synchronization word.

- Concealment of non-correctable parts

  The second action is to reveal the erroneous part of the picture as much as possible. These parts are reconstructed with well received information in the decoder.

  This reconstruction results in a difference between the transmitted and decoded image. For this reason update has to be used for all interframe coding techniques i.e. otherwise the error will propagate in the temporal direction. The update coding method does not rely on previously transmitted information.

- Error correction

[2] caused by errors in the memory

> Applying error correcting codes (ECC) results in the correction of (generally speaking) most of the channel errors, leaving a small probability of uncorrectable errors.

In this contribution we will examine in which way the overall performance of the codec could be improved by using error correction. Important in making a choice which technique should be recommended is, what type of errors are to be expected. In the next section the used terminology is given. In section 5 and section 6 the capabilities of the BCH code and the RS code are explained. In section 7 the tradeoff between the two solutions are given.

3.1 Used Terminology

In general if C is a linear code, we say that a matrix H is a parity-check matrix for C if it has the property that $\underline{x} \in C$ when and only when $\underline{x}H^T = \underline{0}$

Suppose H is a parity check matrix for C, that $\underline{b}$ is the transmitted codeword and that $\underline{r} = \underline{b} + \underline{e}$ is received. Than the Syndrome of r, is defined as :

$$\underline{s} = \underline{r}H^T \tag{1}$$

The RS codes are sub codes of the BCH codes.
If C is an (n,k) linear code over $GF(q^m)$ with minimum distance d_{min}, then its $GF(q)$ subcode is an (n',k') linear code over $GF(q)$ with minimum distance d'_{min} where:

$$k' \leq k$$

and

$$d'_{min} \geq d_{min}$$

Where k denotes the number of information bits, n - k the number of parity bits. Therefor n is the number of bits in the block to be protected. The number of errors which can be corrected denoted with t is $t = \frac{d-1}{2}$.

25k

4 Models for various channels

The error correcting codes under consideration are belonging to the family of block codes. Given a number k of data bits and a number r of redundancy bits, together they form a coded block with the length $n = k + r$. Each error correction code can correct a specific amount of incorrect symbols. A binary BCH code uses bits as symbols, whereas the RS code eight-bit symbols.
As given in appendix A the performance of the error corrector highly depends on the characteristics of the channel. To describe the channel it is insufficient to give the bit error rate, some other characteristics are necessary in order to be able of making a choice for a proper error corrector.
The common used channel statistics are:

- The average error probability
- The gap distribution
- The burst distribution
- The burst interval distribution

In G.821 [7] the error performance is given in terms of; errored seconds, severely errored seconds and degraded minutes. These terms are used as a convenient and concise performance objective "identifier", it is not a measure for the acceptability. The measures is G.821 are not sufficient to make a decision which code should be chosen. Measuring the error types, should be carried out on real channels to obtain the mentioned channel statistics with which a proper choice could be formulated.

A two state Markov model like Gilbert proposed could be a tool for testing the various block codes. The channel statistics as mentioned before could be implemented and varied to check the capabilities.
The random error correcting capability of the BCH is one of the known properties, but the BCH is also capable to correct bursts. The burst correcting character is not obtained by interleaving but is also a capability of the BCH. The RS is on the otherhand more known on the burst correction. In appenndix E an explanation of the interleaving process is given.

The comparison of the BCH and the RS is this contribution is carried out only on a mathematical basis.

From earlier work [10] , the picture distortion caused by random errors without error correction was found to be:

Error Rate	Effect on picture quality
10^{-6}	Very few perceivable erors
10^{-5}	Noticeable distortion
10^{-4}	Bad distortion
10^{-3}	Total distortion
10^{-2}	Frame breakdown

5 The Bose-Chaudhuri-Hochquenghem (BCH) codes

The BCH code is a very powerful random error corrector. Modifying the number of parities give the BCH also a burst correcting ability. At the moment it is not possible to correct random as well as bursts.

Characteristics:

$$n = 2^m - 1 \tag{2}$$

$$k \geq n - mt \tag{3}$$

$$d \geq 2t + 1 \tag{4}$$

$$\tag{5}$$

The encoding is simple due to its cyclic properties
The Decoding is carried out in four steps:

1. calculate syndromes s(j) = r(α^j) (j = 1,...,2t)
2. determine error location polynomial[3]
3. Determine[4] the error locations polynomial.
4. correct the received pattern by f(x) = r(x) - e(x)

[3] Berlekamp-Massey or Euclides
[4] using e.g. the Chien search

6 The Reed Solomon Codes

The Reed Solomon (RS) codes are BCH codes over GF(q) with the special property that the length n is $q - 1$. These codes are cyclic with symbols from $GF(2^M)$ generated by:

$$G(x) = (x - \alpha)(x - \alpha^2)\ldots(x - \alpha^{2t}) \tag{6}$$

Characteristics:

$$\begin{aligned} n = 2^m - 1 \;\; bytes \;\; &= m(2^m - 1) bits && (7) \\ n - k = 2t \;\; bytes \;\; &= m2t \;\; bits && (8) \\ d &= 2t + 1 && (9) \\ & && (10) \end{aligned}$$

RS codes are important for several reasons:

- They are the natural codes to use when a code is required of length less that the size of the field.
- They are convenient for building other codes.
- They are useful for correcting bursts of errors.

The decoding process of the Reed-Solomon code is more complex due to the determination of the weight of the found error. I.e. if an error is found than the error pattern in the symbol needs to be determined.
The decoding process can be thought to be calculated in six step:

1. calculate syndromes s(j) = r(α^j) (j = 1,...,2t)
2. determine error location polynomial[5]
3. Determine—footnote using e.g. the Chien search the error locations polynomial.
4. determine for each of the error locations the weight of the error.
5. correct the received pattern by f(x) = r(x) - e(x)

[5] Berlekamp-Massey or Euclides

7 Comparison of BCH and RS

A binary (n, k) BCH code can correct about twice as much incorrect symbols as an (n, k) RS code (with the same n and k). In table 1, table 2[6] and table 3[7] some examples are given.

(n, k) RS code	number of correctable symbols	% overhead	binary (n, k) BCH code	number of correctable symbols	% overhead
(504,488)	1	3.17	(511,493)	2	3.65
(1016,984)	2	3.15	(1023,983)	4	4.07
(2040,1976)	4	3.14			

Table 1: Comparison of some RS codes with some binary BCH codes

where:

RS (504,488)	RS (8*63, 8*61)
RS (1016,984)	RS (8*127, 8*123)
RS (2040,1976)	RS (8*255, 8*246)

	Bose-Chaudhuri-Hochquenghem	Reed-Solomon
Implementation		
Encoding	simple	simple
Decoding	simple	complex
Ability error	**correction behaviour**	
Burst $\geq m$ — ?	suitable	needs more parities
Burst $< m$	good	excellent
Random	see appendix B	see appendix B
Hardware complexity	reasonable	fairly complex
Software implementable		

[6] see appendix B

[7] see appendix B

References

[1] "A double Error Correction BCH Encoder/Decoder LSI CCITT SGXV/1, Document # 325, March 1988.

[2] Data Sheets Reed-Solomon Code Generator CXD1037G

[3] Data Sheets Reed-Solomon Decoder CXD1038G

[4] F.J.MacWilliams N.J.A.Sloan, " The Theory of error-correcting Codes, North-Holland Mathematical Library, 1988 ISBN 0 444 85193 3

[5] G.C.Clark, J.B.Cain, Error-Correction coding for Digital Communications, Plenum Press, 1982, ISBN 0 306 40615 2

[6] R.J.McEliece, The theory of information and coding, Addison-Wesley Publ. Company, Massachussets, 1977, 2nd ed.

[7] Error performance of an international digital connection forming part of an integrated services digital network. CCITT Recommendation G.821

[8] Ralf Hinz, Forward error correction for p * 64 kbps video codecs internal AEG FI 33-UL.

[9] Error Protection by Reed-Solomon Codes for Video Telephony. source: Philips Kommunications Industrie AG.

[10] Status field trials error correction hardware H.261 codecs, source: The Netherlands, BTRL Doc 421 CCITT SG XV/1/Q4 Dec. 1988

A List of requirements for Error-Control selection

To meet the requirements for a good error corrector one should bear in mind the listed items.

1. The required information throughput rate,
2. The allowed throughput delay for channel encoding and decoding,
3. The acceptable bandwidth expansion.
4. The modem modulation, symbol-rate, tracking-accuracy, phase-ambiguity and bit-slip characteristics.
5. The characteristics of the burst on bursty channels in terms of duration and attenuation of the bursts. Their periodic or random nature, the guard space between bursts or their duty cycle, and the availability and reliability of information on the occurrence of a burst that could be given to the decoder.
6. The burst rate and duty cycle of the received data and the suitability of using a decoder at the smoothed average rate, rather than the burst rate.
7. Decoder synchronization time and loss-of sync requirements.
8. The allowed rate of accepting incorrect messages.
9. The vulnerability of the destination data sink to bursts of errors from the decoder.
10. The costs, size, weight, and power restrictions.

B Error performance BCH and RS

BCH (n,k,t)-code	Improvement BER			overhead
	10**-3	10**-6	10**-9	
(1023,1013, 1)	2.73E-01	5.22E-07	5.23E-13	0.99 %
(1023,1003, 2)	8.45E-02	1.78E-10	1.78E-19	1.99 %
(1023, 993, 3)	2.04E-02	4.53E-14	4.54E-26	3.02 %
(1023, 983, 4)	4.00E-03	9.24E-18	4.53E-32	4.07 %
(1023, 973, 5)	6.61E-04	1.57E-21	-	5.14 %
(1023, 963, 6)	9.43E-05	2.28E-25	-	6.23 %
(511, 502, 1)	9.35E-02	1.30E-07	1.30E-13	1.79 %
(511, 493, 2)	1.52E-02	2.21E-11	2.21E-20	3.65 %
(511, 484, 3)	1.88E-03	2.81E-15	2.81E-27	5.58 %
(511, 475, 4)	1.87E-04	2.85E-19	1.81E-32	7.58 %
(255, 247, 1)	2.74E-02	3.24E-08	3.24E-14	3.24 %
(255, 239, 2)	2.26E-03	2.73E-12	2.73E-21	6.69 %
(255, 231, 3)	1.41E-04	1.72E-16	1.72E-28	10.39 %
(127, 120, 1)	7.36E-03	8.00E-09	8.00E-15	5.83 %
(127, 113, 2)	3.04E-04	3.33E-13	3.33E-22	12.39 %

Table 2: Improvement BER using BCH

Reed-Solomon (n,k,t)-code	Improvement BER			overhead
	10**-3	10**-6	10**-9	
(255, 251, 2)	2.26E-03	2.73E-12	2.73E-21	1.59 %
(255, 249, 3)	1.41E-04	1.72E-16	1.72E-28	2.41 %
(255, 247, 4)	7.02E-06	8.64E-21	-	3.24 %
(255, 245, 5)	2.91E-07	3.60E-25	-	4.08 %
(255, 243, 6)	1.03E-08	1.28E-29	-	4.94 %
(255, 241, 7)	3.19E-10	5.30E-33	-	5.81 %
(127, 123, 2)	3.04E-04	3.33E-13	3.33E-22	3.25 %
(127, 121, 3)	9.37E-06	1.03E-17	1.03E-29	4.96 %
(127, 119, 4)	2.30E-07	2.54E-22	-	6.72 %
(127, 117, 5)	4.66E-09	5.17E-27	-	8.55 %
(63, 59, 2)	3.80E-05	3.97E-14	3.97E-23	6.78 %
(63, 57, 3)	5.68E-07	5.96E-19	5.98E-31	10.53 %

Table 3: Improvement BER using RS

C Chip set for the BCH solution

Features Encoder/Decoder

- Four selectable polynomials
- Programable Shortened length
- High Speed operation (max 6.3 Mbps)
- I/O direct TTL compatible
- Uncorrectable Error indicator and Bit Error monitor Functions

C.1 Basics of BCH chip set

The user can choose among 4 basic code formats:

Code (n,k)	ECC	t	overhead
(63,51)	12	2	19 %
(127,113)	14	2	11 %
(255,239)	16	2	6 %
(511,493)	18	2	3.5 %

Each of these code formats (n,k) can be shortened to $(n - shl, k - shl)$ with $shl < k$. To make the word length equal to a power of 2 a frame bit can be added to the code word.

$$g_{(63,51)}(x) = (x^6 + x^5 + 1)(x^6 + x^5 + x^4 + x^2 + 1) \tag{11}$$

$$g_{(127,113)}(x) = (x^7 + x^3 + 1)(x^7 + x^3 + x^2 + x + 1) \tag{12}$$

$$g_{(255,239)}(x) = (x^8 + x^4 + x^3 + x^2 + 1)(x^8 + x^6 + x^5 + x^4 + x^2 + x + 1) \tag{13}$$

$$g_{(511,493)}(x) = (x^9 + x^4 + 1)(x^9 + x^6 + x^4 + x^3 + 1) \tag{14}$$

D Chip set for the RS solution

Features Encoder

- The number of check words is selectable from 2,3,4,6 and 8
- Possible to code and interleave data up to 4 words.
- 72 pins

Features Decoder

- Can select various decoding algorithms
- Can cope with the desired code length 255 (Max)
- High speed operation 16MHz
- 132 pins

D.1 Basics of Reed-Solomon chip set

The field generator polynomial is expressed as:

$$P(x) = x^5 + x^4 + x^3 + x^2 + 1 \tag{15}$$

The number of n - k check words can be selected in five ways which yields in the following possible generator polynomials:

$$\begin{aligned}
G_8 &= (x-\alpha^0)(x-\alpha^1)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^6)(x-\alpha^7) && (16)\\
G_6 &= (x-\alpha^0)(x-\alpha^1)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5) && (17)\\
G_4 &= (x-\alpha^0)(x-\alpha^1)(x-\alpha^2)(x-\alpha^3) && (18)\\
G_3 &= (x-\alpha^0)(x-\alpha^1)(x-\alpha^2) && (19)\\
G_2 &= (x-\alpha^0)(x-\alpha^1) && (20)
\end{aligned}$$

$$\begin{aligned}
G_8 &= x^8 + \alpha x^7 + \alpha x^6 + \alpha x^5 + \alpha x^4 + \alpha x^3 + \alpha x^2 + \alpha x^1 + \alpha^{28} && (21)\\
G_6 &= x^6 + \alpha^{166} x^5 + \alpha^0 x^4 + \alpha^{134} x^3 + \alpha^5 x^2 + \alpha^{176} x^1 + \alpha^{15} && (22)\\
G_4 &= x^4 + \alpha^{75} x^3 + \alpha^{249} x^2 + \alpha^{78} x^1 + \alpha^6 && (23)\\
G_3 &= x^3 + \alpha^{198} x^2 + \alpha^{199} x^1 + \alpha^3 && (24)\\
G_2 &= x^2 + \alpha^{25} x^1 + \alpha^1 && (25)
\end{aligned}$$

E Interleaving

Interleaving is an technique which rearrange the input bit sequence in such a way that bursts are spread over an length of N bits to create random errors. An *interleaver* is an device that rearranges (or permutates) the ordering of a sequence of symbols in a deterministic manner. The *deinterleaver* applies the inverse operations to restore the sequence in its originals ordering.

The interleaving process is carried out by reading in the coded symbols into a matrix on N rows and B columns. The permutation consists of reading out the symbols by row s prior to transmission. The interleaver is also referred to as a (B,N) interleaver.

The characteristics of the interleaving process is:

- Any burst of length $b \leq B$ results in a single error, separated by at least N symbols.
- A periodic sequence of single errors spaced by B symbols results in a single burst of length N at the deinterleaver output.
- The end to end delay is 2NB symbols exclusive the channel delay.

The choice of N depends on the realization of the encoder but for the proposed block codes N should be larger than the code block length.

The interleavers can cause the same synchronization problems as with the block codes.)

The burst correcting capabilities of a code can be greatly improved by interleaving the code. From the two methods where there are pseudorandom and periodic the last one is commonly used.

F Realisation aspects BCH and RS

Burst Error Correction

There is a tendency for errors in digital transmission media to occur in bursts, however, the burstiness of real transmission systems is not well defined. The effectiveness of an error correction scheme employing random correction only would be much reduced if a significant proportion to errors occurred in bursts.

BCH codes are basically designed to correct random errors. They do however have an ability to allow correction of burst errors. The methods of correction are, however, totally different and more significantly separate. A problem occurs if both random and burst errors are present, as there is no simple way of arbitrating between the two methods.

For example, consider a single burst of 3 bits. The burst corrector will correct the incoming data fully. The random error corrector is unable to correct 3 errors. However, as the received vector is an invalid code it may try to correct it the best way it can (ie correct 2 bits). In doing so the random corrector wrongly changes 2 further bits, thus the data block now has 5 errors. Both burst and random correctors think that they have corrected the code properly (only one is right).

Similarly the burst corrector can correct burst errors, but when faced with double random errors may miscorrect them.

Whenever a corrector makes a correction it produces a valid code. If both correctors correct but disagree as to the position of of the errors, there is no simple way of determining which is right.

Using both simultaneously in parallel is unacceptable as one corrects properly and the other miscorrects still leaving the data with errors.

However, it is possible to gain information about the nature of errors that have been occurring, which can be used to make an intelligent guess as to which of the two correctors is giving the right results. When just the random error corrector corrects, then the error is a double random, and when just the burst corrects, then errors are occurring in bursts. This allows a measure of the occurrance of each to be made over a suitable time period. This can then be used to switch out one of the correctors, thus producing a decoder with the best properties of both burst and random correctors.

RS codes are decoded using a single algorithm for burst and random correction (no distinction is made between the two) so the above problem for BCH codes does not occur. In this respect RS codes offer a more powerful burst correction ability than BCH. However, as we will see in section 4.0 this is at the expense of a more complex algorithm and hardware.

4.0 Hardware Requirements

A comparison of the hardware required for each of the coding schemes can be made by estimating the hardware necessary for each of the above stages:-

	BCH	RS
(1) Encoding	18-stage LFSR	4-stage 7-bit wide SR 7-bit SiPo (7 * 512 bit) LUT 4 * (7-bit) XOR 7-bit PiSo
(2) Block Alignment	3-bit frame count 8-bit word sequencer	2-bit frame count 8-bit word sequencer
Decoding		
(3) Block Alignment	Frame store Control PROM(4K*8)	Frame Store Control PROM(4K*4)
(4) Syndrome Calculation	3* (9-stage LFSR)	4* (7-stage LFSR)
(5) Calculate error location equation coefficients	2* (9*512 it LUT) 9-bit Mod 511 Adder 9-bit Mod 511 Subtractor	DSP
(6) Calculate Error magnitude-		DSP
(7) Solve Equation	2* (9-stage LFSR) 9-bit Mod 2 Adder 9-bit Comparator	DSP
(8) Correct Data	XOR gate	XOR gate 10-bit count 2*7-bit Piso 2*7-bit comparator 2-1 mux
(9) Burst correction	2* (18-stage LFSR)	-
(10) Other's	511-stage shift reg.	2*889-stage shift reg.

Note:

1. LFSR	- Linear feedback shift register
2. SR	- Shift register.
3. SiPo	- Serial in parallel out, (serial to parallel conversion).
4. PiSo	- Parallel in serial out, (parallel to serial conversion).
5. LUT	- Look up table.
6. MUX	- Multiplexer.

4.1 BCH(511,493) Encoder

A block diagram for the BCH encoder showing the main processing stages is shown in Fig 1(a). This may be implemented using a single ASIC or programmable gate array (2000 gates).

4.2 RS(127,123) Encoder

The RS(127,123) encoder is shown in Fig 1(a). The encoder is more complex than for the BCH code as it operates using galois field arithmetic on 7-bit words. The data must first be blocked using a SiPo, before being presented to the shift register. Galois arithmetic is implemented using a single look-up-table. The complete implementation may consist of a programmable gate array (3000 gates) and 8*512bit PROM.

4.3 BCH (511,493) Decoder

The BCH decoder is shown in Fig 2,3,4. A complete description of its operation and timing is beyond the scope of this document. However, a full implementation is possible using a programmable gate array (4200 gates), 512 element shift register.

4.4 RS(127,123) Decoder

The RS(127,123) decoder is shown in Fig 5. The syndromes are calculated in hardware as before using a gate array device. However, the translation of the syndromes into the error locations and magnitudes requires a DSP or microcontroller, due to the complexity of the Reed-Solomon algorithm. It has be estimated that approximately 200 instruction are required for this calculation. In order for the decoder to operate as a pipeline the error locations and magnitudes must be calculated within a single block length (889 clock cycles). Therefore it may be necessary to clock the DSP at 2-4 times the line clock rate (max 4-8MHz) . The locations and magnitudes are then passed back to the gate array to correct the data.

The implementation is thus more hardware intensive than for BCH, requiring a programmable gate array (6400 gates), DSP and Programme ROM.

BCH (511,493) ENCODER

Requirements:-

1 Xilinx 3020

1 PROM to load Xilinx

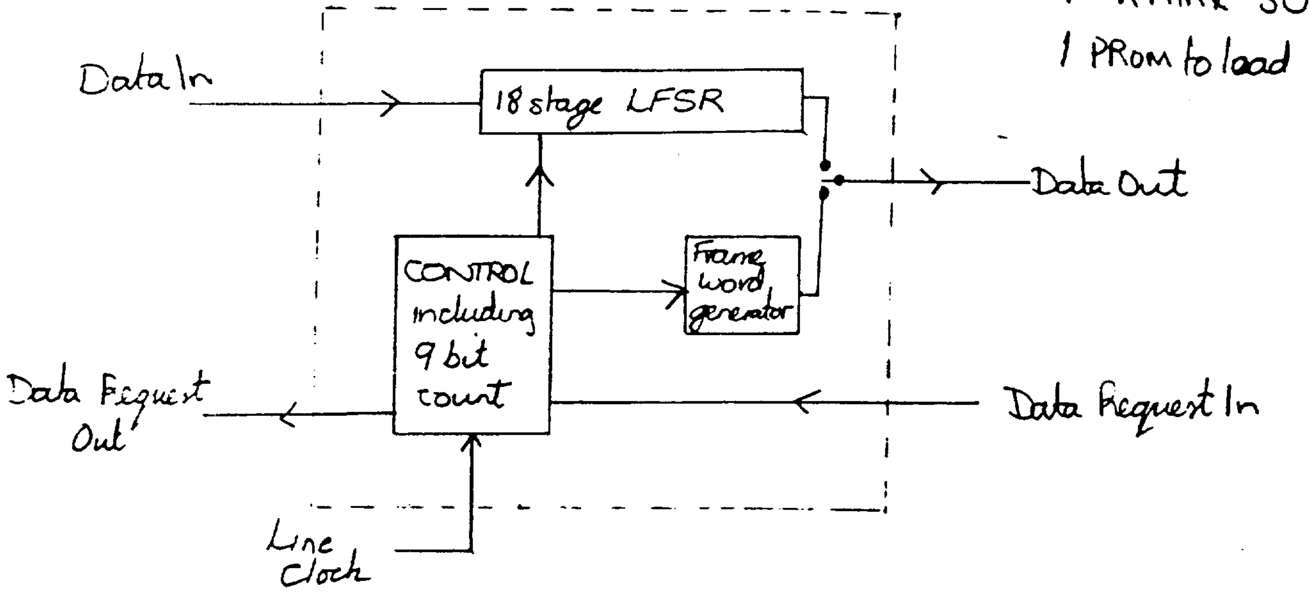


Fig 1(a)

REED SOLOMON (127,123) ENCODER

Requirements

1 Xilinx 3030

1 PROM to load Xilinx

1 FROM look up table

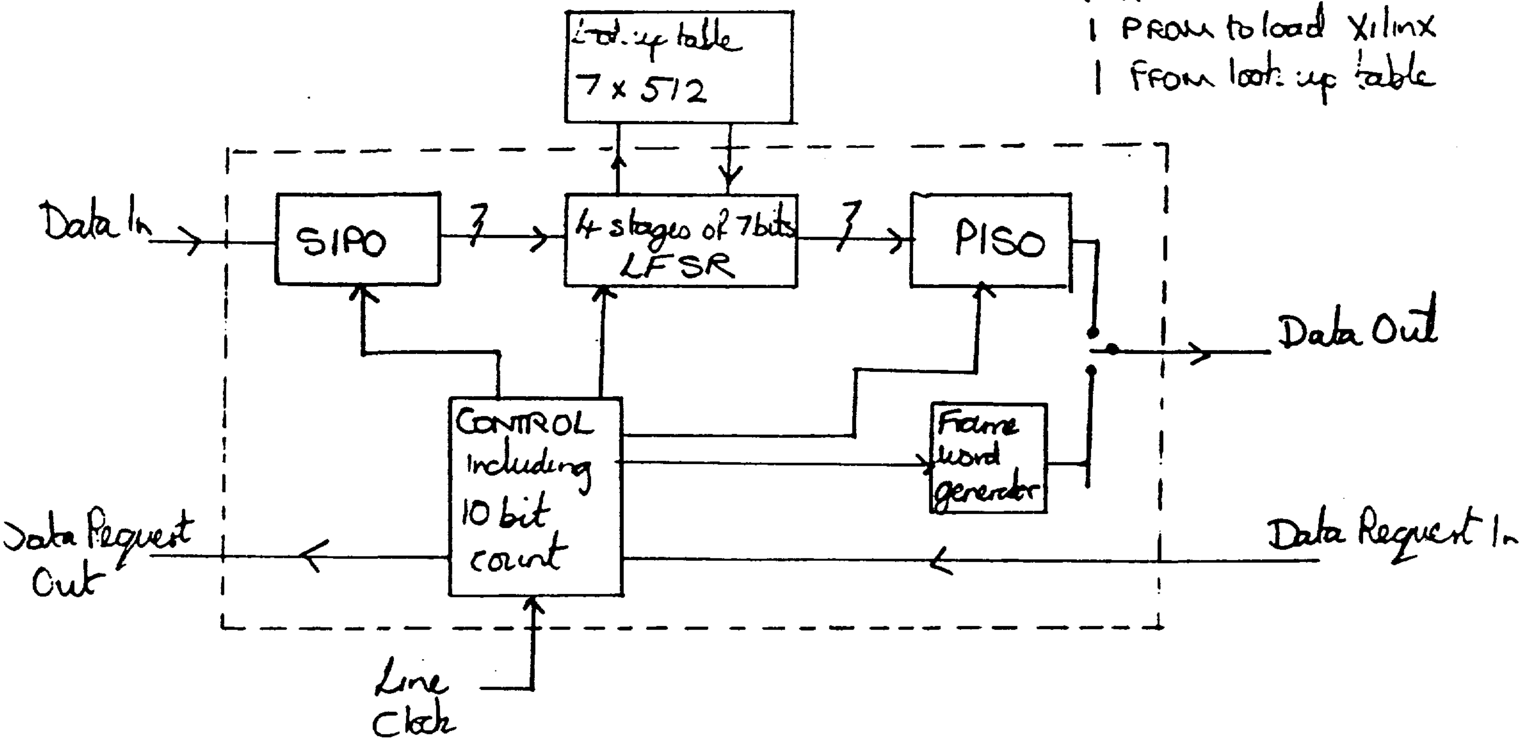


Fig 1(b)

BCH DECODER.

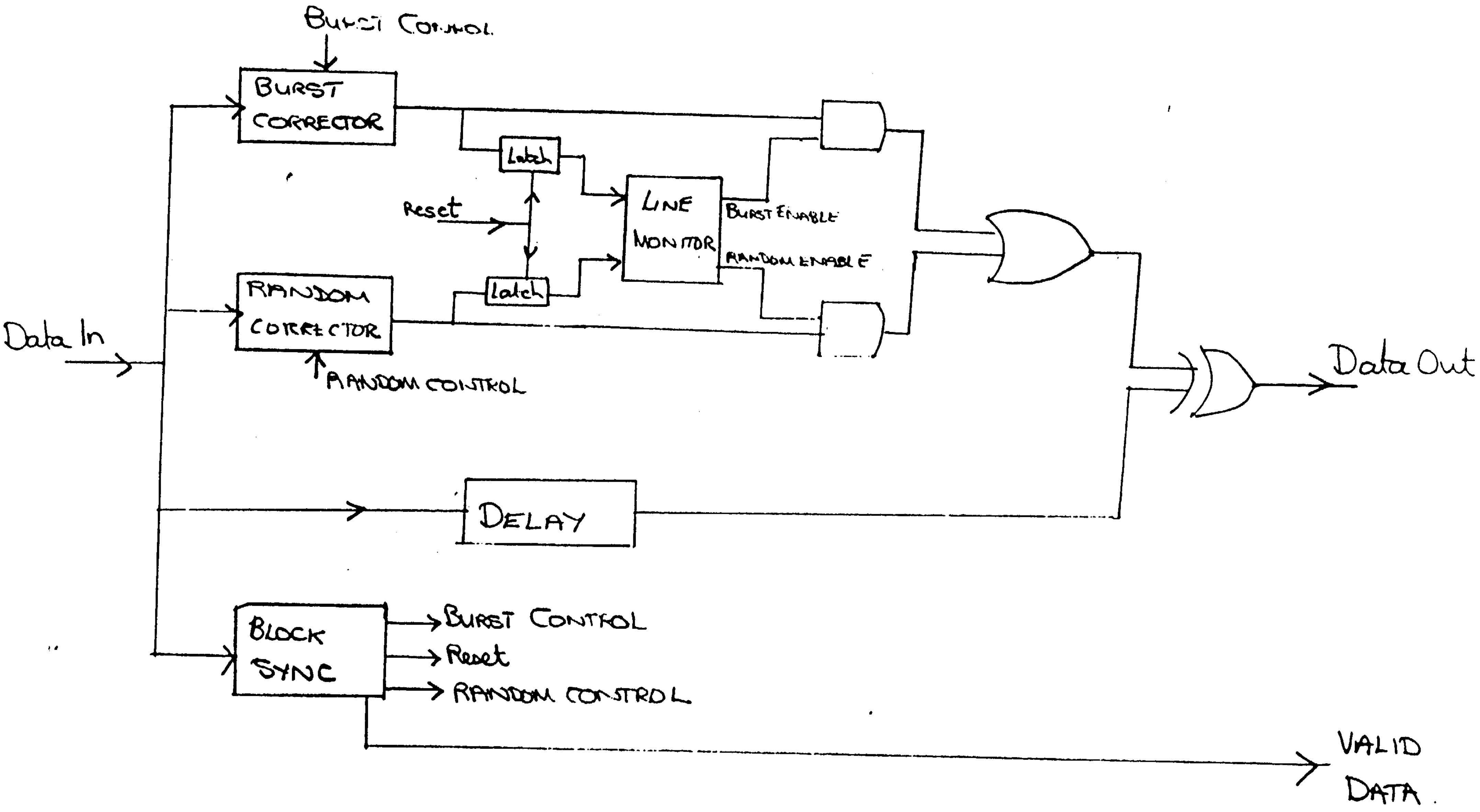
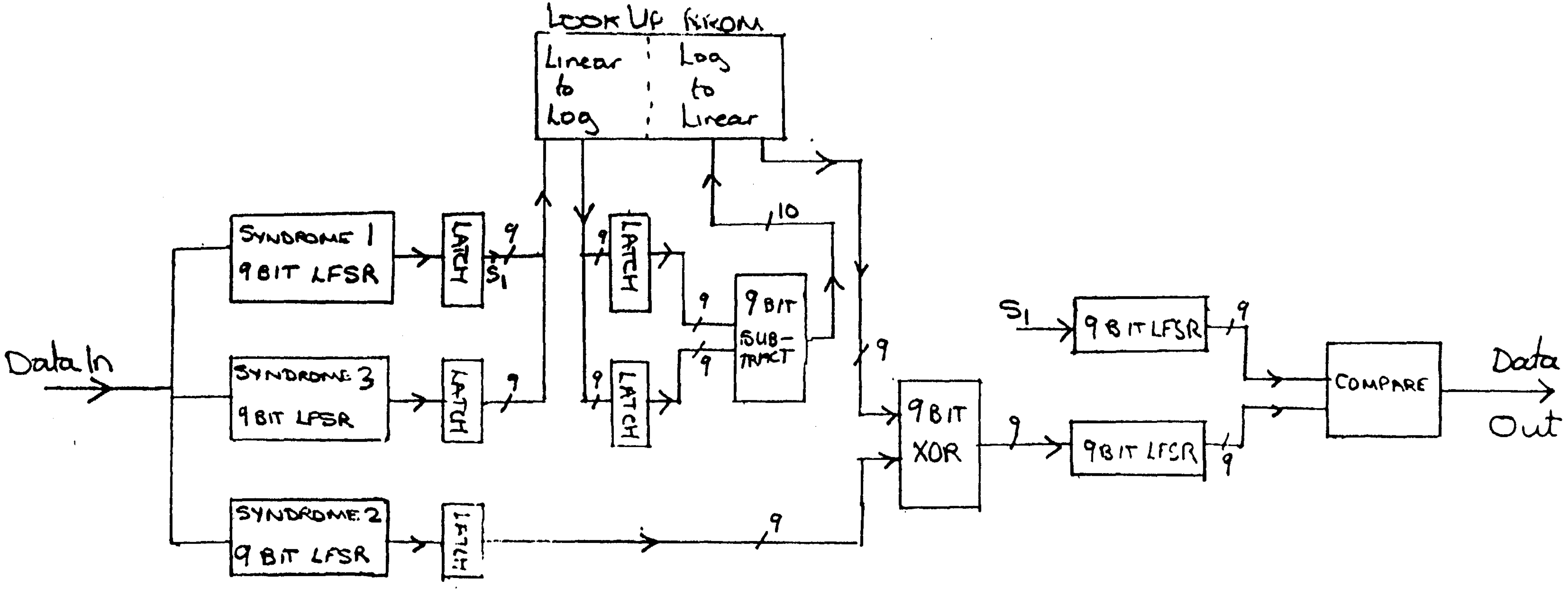


Fig 2

BCH RANDOM CORRECTOR



REQUIREMENTS

1 XILINX 3042

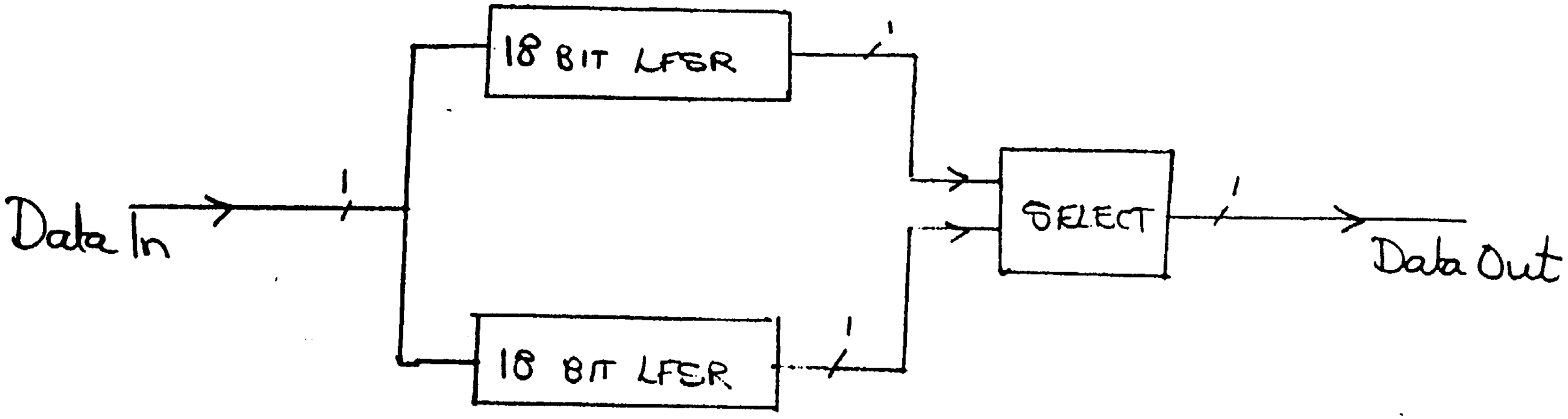
1 PROM to load XILINX

1 LOOK UP PROM

Fig 3

12c

BCH BURST CORRECTOR



REQUIREMENTS

NONE
(Included in Xilinx of
Random Corrector)

Fig 4

RS(127,123) Decoder

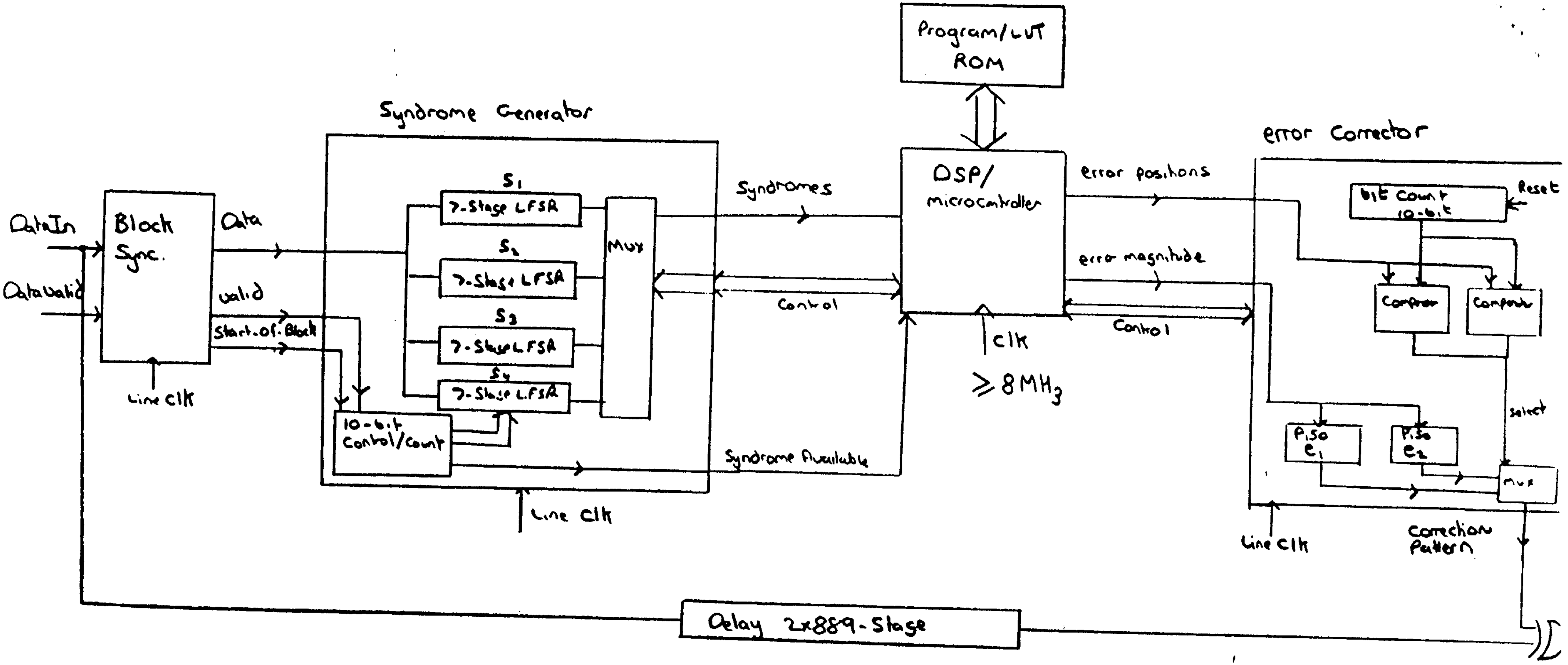


Fig5

Hardware Requrements.

1 Xilinx 3042 (Asic)
1 DSP+ Support chips.
1 PROM (Program)
1 Delay (SR-2k)

Block Sync

Frame Store (256k)
1 PROM (control)
1 PAL (10-bit Count)