BOOK II

PROCEEDINGS
OF
THE CARMEL CONFERENCE

THE FIRST ANNUAL SYMPOSIUM ON THE APPLICATION OF COMMUNICATIONS TECHNOLOGY TO HIGH DENSITY MAGNETIC RECORDING

JANUARY 13, 14, 15, 1981

SYMPOSIUM CHAIRMAN	PROGRAM CHAIRMAN
R. L. COMSTOCK	R. C. SCHNEIDER
SAN JOSE	TUCSON

IBM CONFIDENTIAL

TABLE OF CONTENTS

BOOK I

BOOK II

MINIMUM NOISE PULSE SLIMMER

Lineu C. Barbosa
IBM Research Laboratory
Monterey and Cottle Rds., San Jose, Ca.

Pulse slimming is a technique commonly used to reduce intersymbol interference in magnetic recording, communication, radar and image processing. This technique in general has an adverse effect on the signal to noise ratio and therefore a limitation on the amount of pulse slimming is dictated by the noise deterioration associated with it. Different pulse slimming equalizers can be designed for a given amount of slimming, each one having a different effect on the total noise deterioration.

This paper introduces a method of designing a pulse slimming equalizer with the property that, among all possible equalizers of a given class of linear systems, this one maximizes the fraction of the output energy (hereon called alpha) within a pre-established time window. The noise deterioration introduced is constrained below a given level. Alternatively, for a given amount alpha of pulse slimming the present equalizer minimizes, within its class, a certain measure of noise deterioration. Control over the amount of slimming and over the noise deterioration is extremely important in face of the above considerations. This control permits tradoffs to be considered. Also extremely important is the fact that the equalizer takes into account the physical constraints of the hardware.

The above technique was used to design equalizers for a magnetic recording channel using a tapped delay line with 10 equaly spaced taps separated by 25 ns. For a maximum transition density of 6T (were T is half the clock window) and no intersymbol interference one should have 100% of the energy of the single transition pulse within a 12T time window. The digitized pulse had about 90% of its energy within the 12T window.

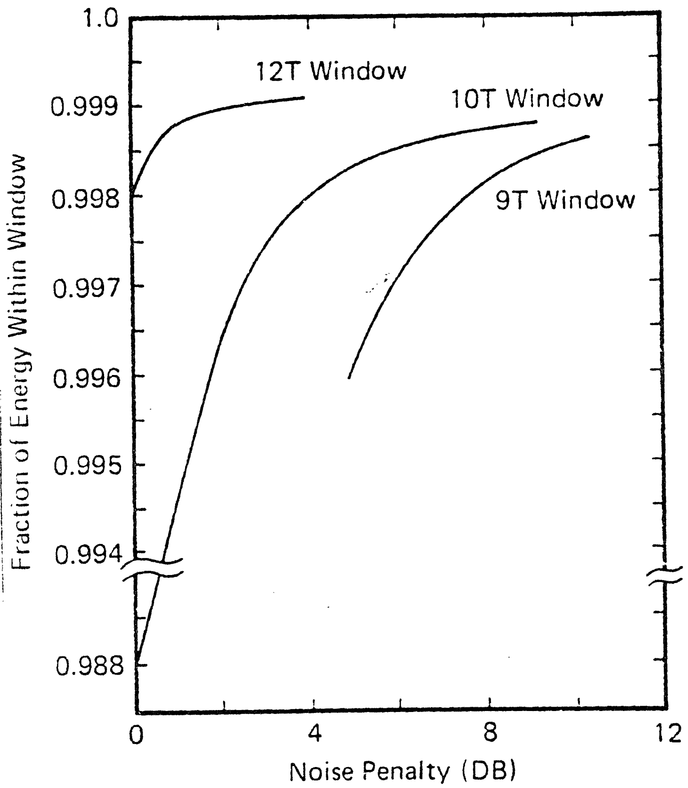


FIG1: Tradoffs between the fraction of the output energy within a given window and the equalizer's noise penalty. 0db = same noise variance before and after equalization when gain is adjusted for equal input and output signal's energies.

Fig. 1 shows the computed tradoffs between alpha and the minimized noise deterioration for several time windows. Notice that even with 0db noise penalty a substantial amount of slimming is possible.

Fig. 2 shows the effect of the 4db-noise-penalty equalizer on the resolution of a complex signal: it brings the resolution from below the 55% level to above the 95% level (resolution=ratio of min. peak amplitude and max. peak amplitude).

Fig. 3 shows the effect of the equalizers on peakshifts occurring when a dibit (double transition) separated by 6T is writen: notice the substantial reduction of the peakshift variance for low noise penalties and the gains in the average as well as in the variance of the peakshifts introduced by the 4db-noise-penalty equalizer. The unequalized dibit is also shown in the picture.

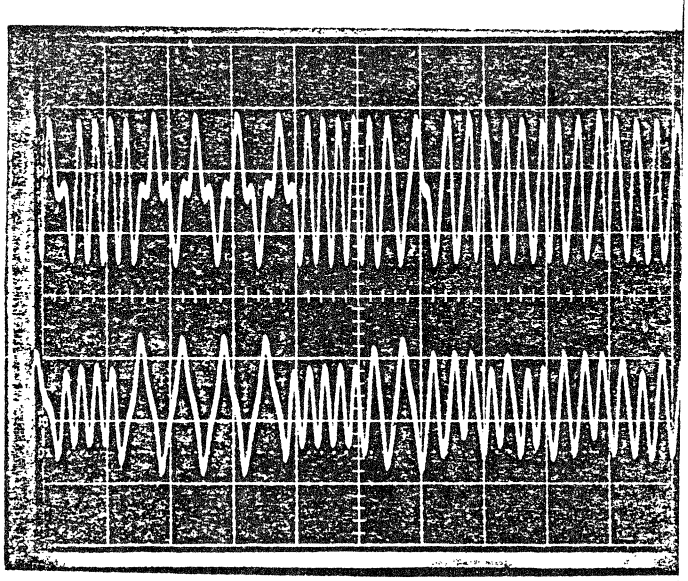


FIG2: Effect of equalization on a complex signal pattern:
top: equalized signal (4db noise penalty equalizer designed for a 10T window).
bottom: unequalized signal

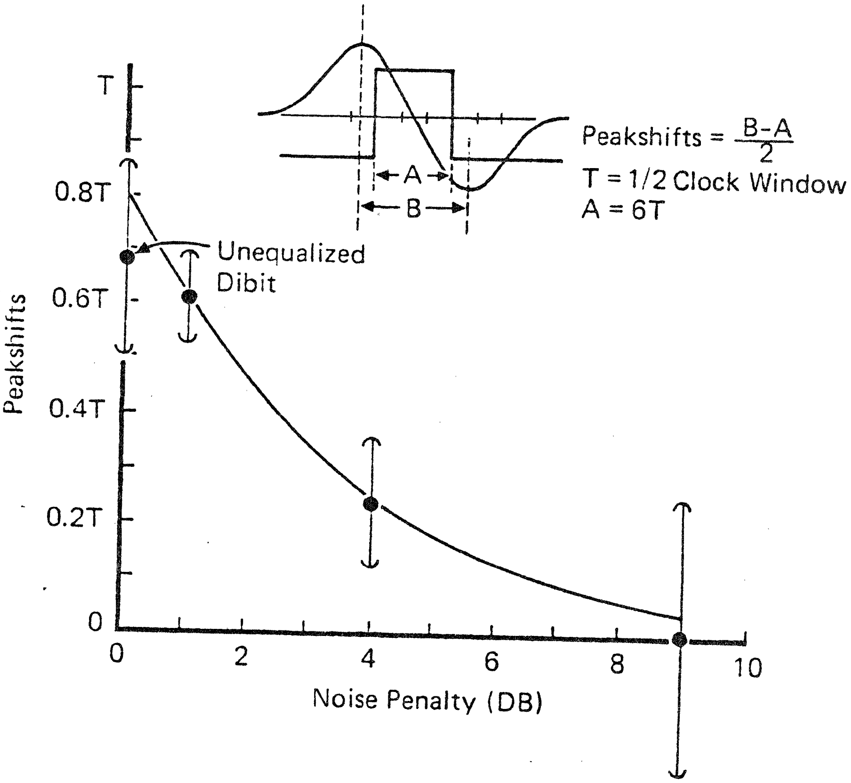
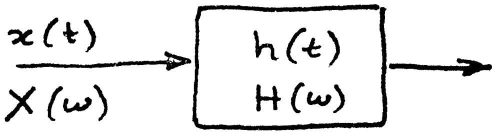


FIG3: Dibit peakshifts as function of the equalizer's noise penalty (10T window)
theoretical curve for peak shifts due to residual intersymbol interference
measured average over 1000 samples
1 standart deviations

Pulse slimming



$y(t) = x * h$

$Y(\omega) = X(\omega) \cdot H(\omega)$

Let $|X(\omega)| > 0$ for all ω

(a) Specify any $y(t)$ (as slim as desired)

and compute $Y(\omega) = \mathcal{F}[y(t)]$

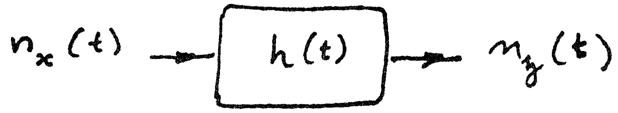
(b) Compute equalizer $H(\omega) = \dfrac{Y(\omega)}{X(\omega)}$

Observation: no matter how fat $x(t)$ is, one can equalize it so as to be as slim as wanted.

However ...

The effect of equalizers on the variance of the background noise

- General equalizers :



$$n_y(t) = \int_{-\infty}^{\infty} n_x(t-\tau)\, h(\tau)\, d\tau$$

$$\therefore \quad \sigma_y^2 = \overline{n_y(t)^2} = \iint_{-\infty}^{\infty} \overline{n_x(t-\tau)\, n_x(t-\tau')}\, h(\tau)\, h(\tau')\, d\tau\, d\tau'$$

$$= \iint_{-\infty}^{\infty} R_x(\tau-\tau')\, h(\tau)\, h(\tau')\, d\tau\, d\tau' = \|h\|_{R_x}^2$$

- TDL equalizers :

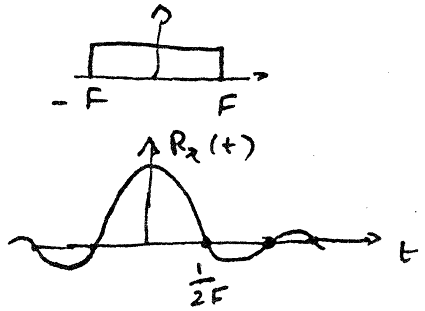
$$n_y(t) = \sum_i n_x(t-\tau_i)\, w_i$$

$$\sigma_y^2 = \sum_{ij} R_x(\tau_i-\tau_j)\, w_i w_j$$

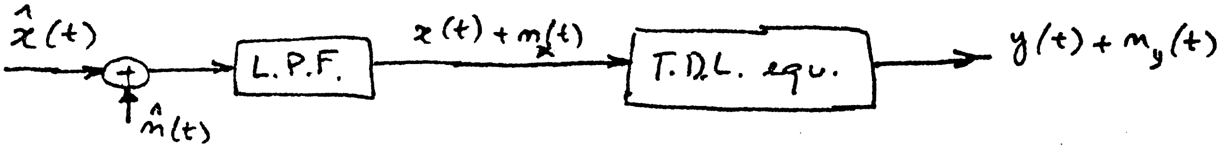
white noise + low pass filter :

$$R_x(\tau_i-\tau_j) = \delta_{ij}\, \sigma_x^2$$

$$\therefore \quad \sigma_y^2 = \sigma_x^2 \sum_i w_i^2 = \sigma_x^2 \|w\|^2$$



Deterioration on the S/N due to TDL equalizers



$$\text{Input } S/N = \frac{\|x\|_T^2}{\|m_x\|_T^2} = \frac{\|x\|_T^2}{\sigma_x^2 \cdot T} = \frac{\|x\|_T^2}{2 N_0 F_M T}$$

$(\sigma_x^2 = 2 N_0 F_M)$ (low pass filter)

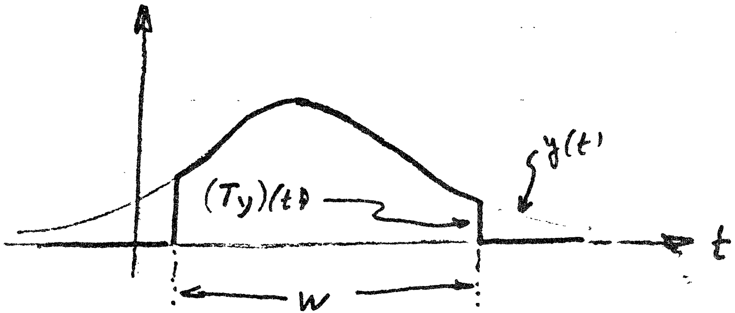
$$\text{Output } S/N = \frac{\|y\|_T^2}{\|m_y\|_T^2} = \frac{\|y\|_T^2}{\sigma_y^2 . T} = \frac{\|y\|_T^2}{2 N_0 F_M \|w\|^2 T}$$

(T.D.L. equ. $\sigma_y^2 = 2 F_M N_0 \|w\|^2$)

$$\therefore \quad S/N \text{ deterioration} \triangleq \frac{(S/N)_{input}}{(S/N)_{output}} = \frac{\|x\|_T^2 \, \|w\|^2}{\|y\|_T^2}$$

Slimming & noise deterioration compromises

P1: Find the equalizer that performs the most "squeezing" in x with a noise penalty not greater than a fixed (tolerable) amount N.



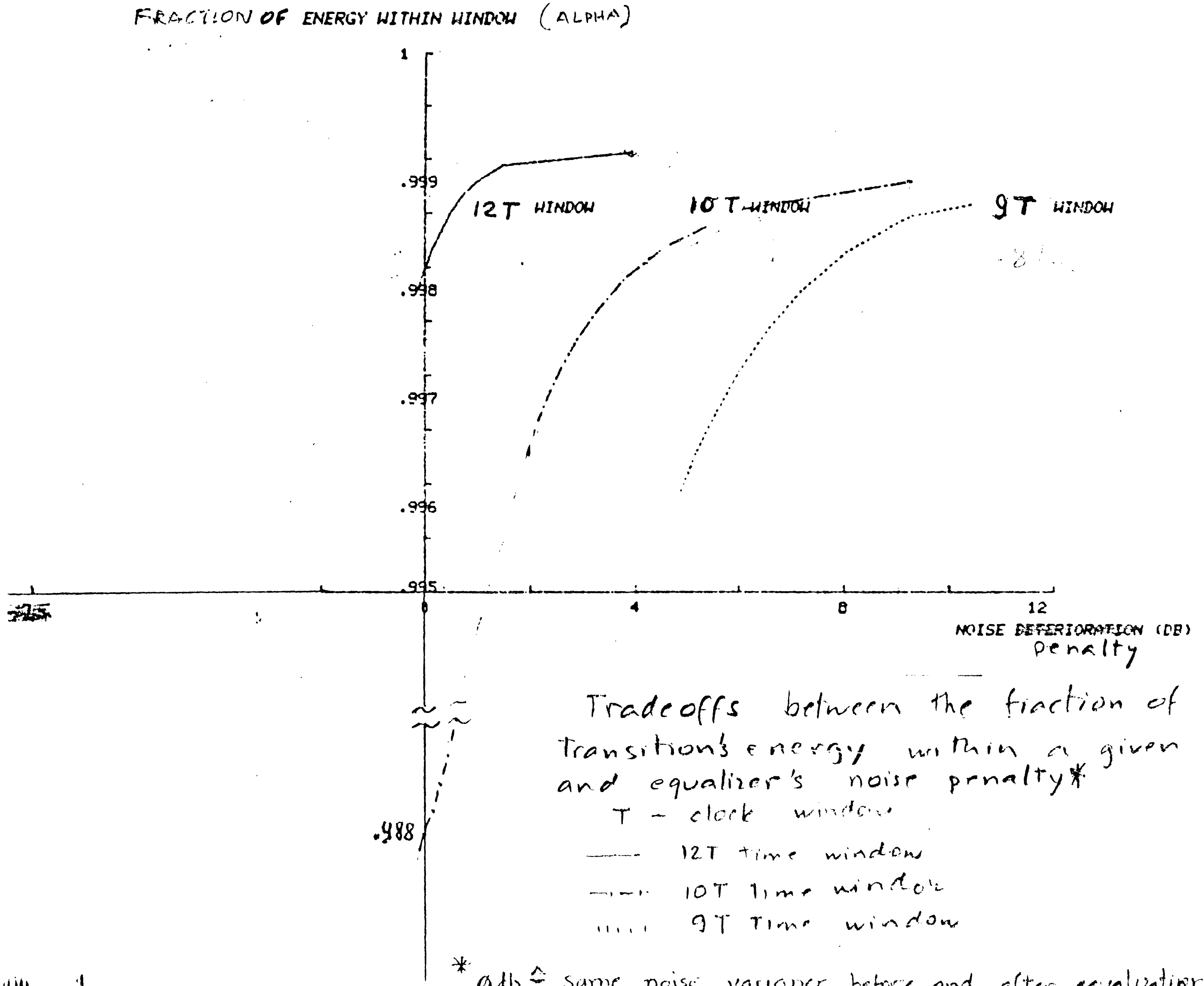
Let :

$$(Ty)(t) = y(t) \quad \text{if} \quad t \in W$$
$$= 0 \quad \text{if} \quad t \notin W$$

Then,
$$\boxed{\underset{h \in \mathcal{H}}{\text{Max}} \quad \frac{\|Ty\|^2}{\|y\|^2} \triangleq \alpha^2 \Big/ \frac{\|h\|^2 \cdot \|x\|^2}{\|y\|^2} \leq N, \quad y = h * x}$$

P2: Among all equalizers that achieve a certain amount of "squeezing" α^2 on x, find the one that pays the least noise penalty :

$$\boxed{\underset{h \in \mathcal{H}}{\text{Min}} \quad \frac{\|h\|^2 \cdot \|x\|^2}{\|y\|^2} \triangleq N \Big/ \frac{\|Ty\|^2}{\|y\|^2} \geq \alpha^2, \quad y = h * x}$$



Tradeoffs between the fraction of single transition's energy within a given time window and equalizer's noise penalty*

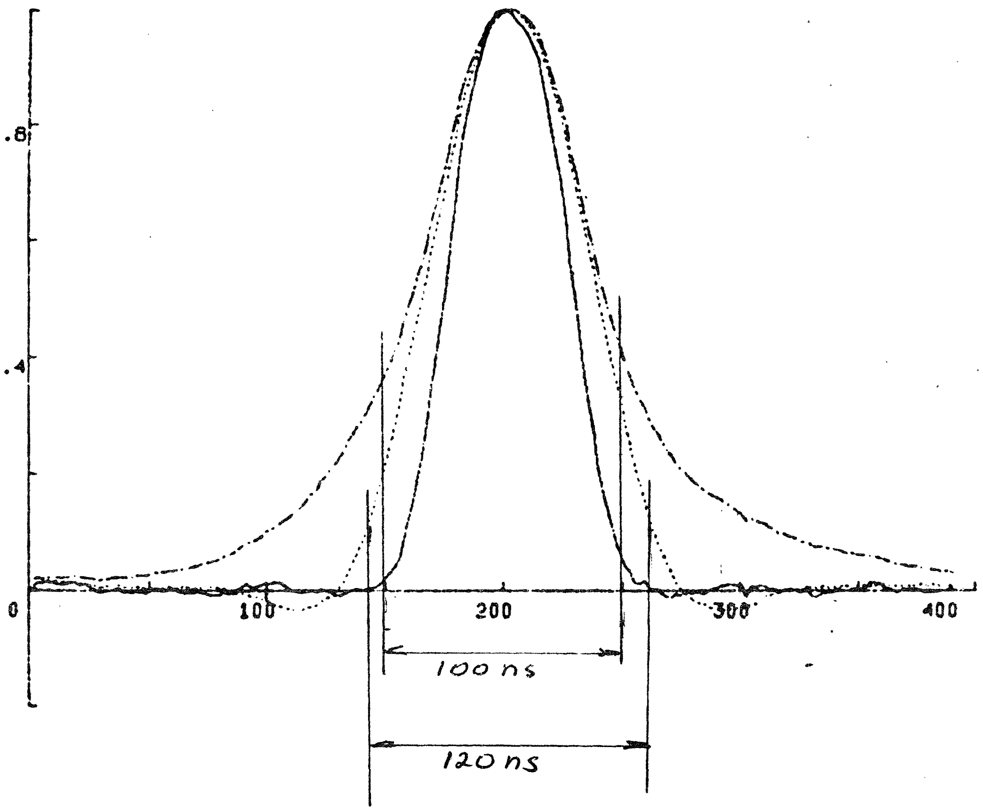
T - clock window

—— 12T time window

—·—· 10T time window

····· 9T Time window

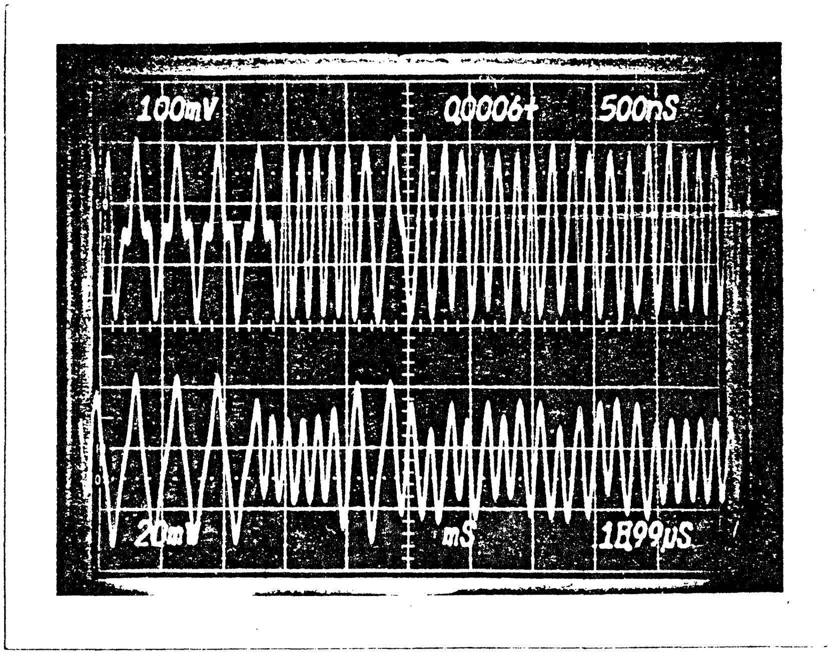
* ∅db ≙ same noise variance before and after equalization when gain is adjusted to have same input and output signals' energies.



-·-·- Input : 91.4% energy within 100 ns window

...... Equalized (0dB on 100ns) : 98.8% energy within 100ns window

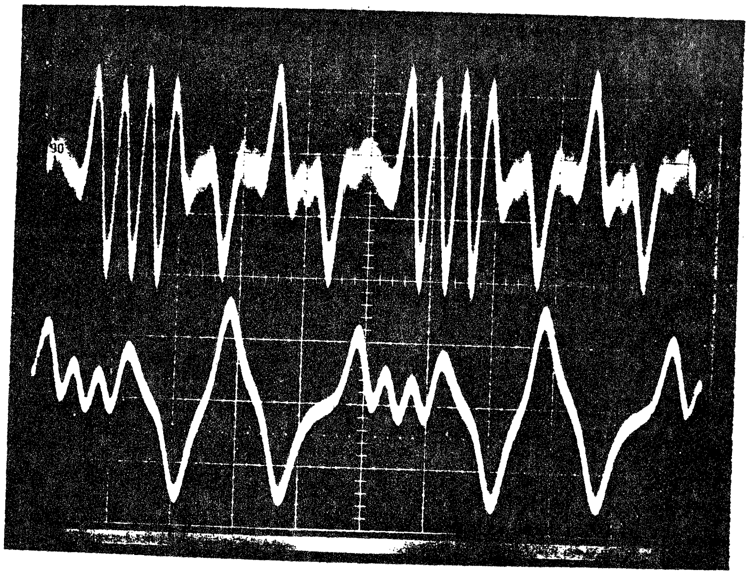
—— Equalized (9dB on 100ns) : 99.87% energy within 100 ns window

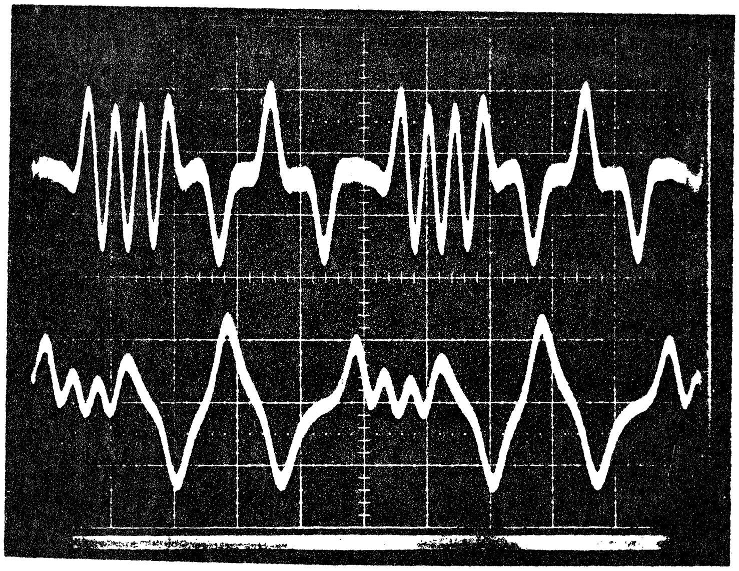


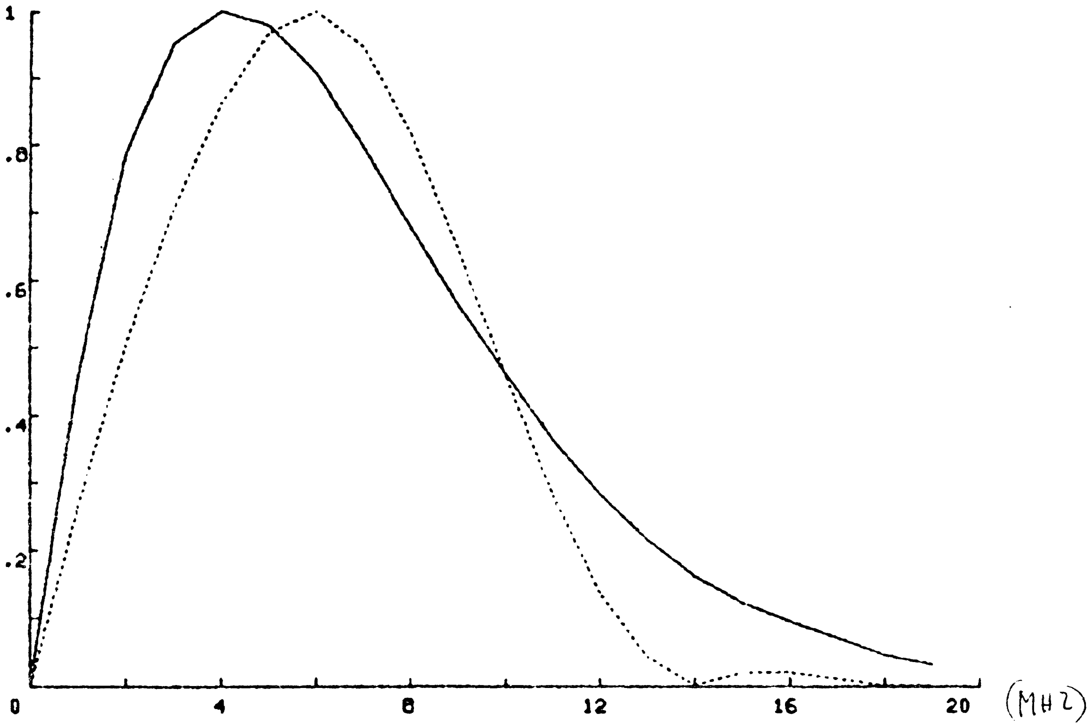
Signal after equa
(4 db noise penalty f
single transition

Unequalized signal

Effect of equalization on a complex signal pattern

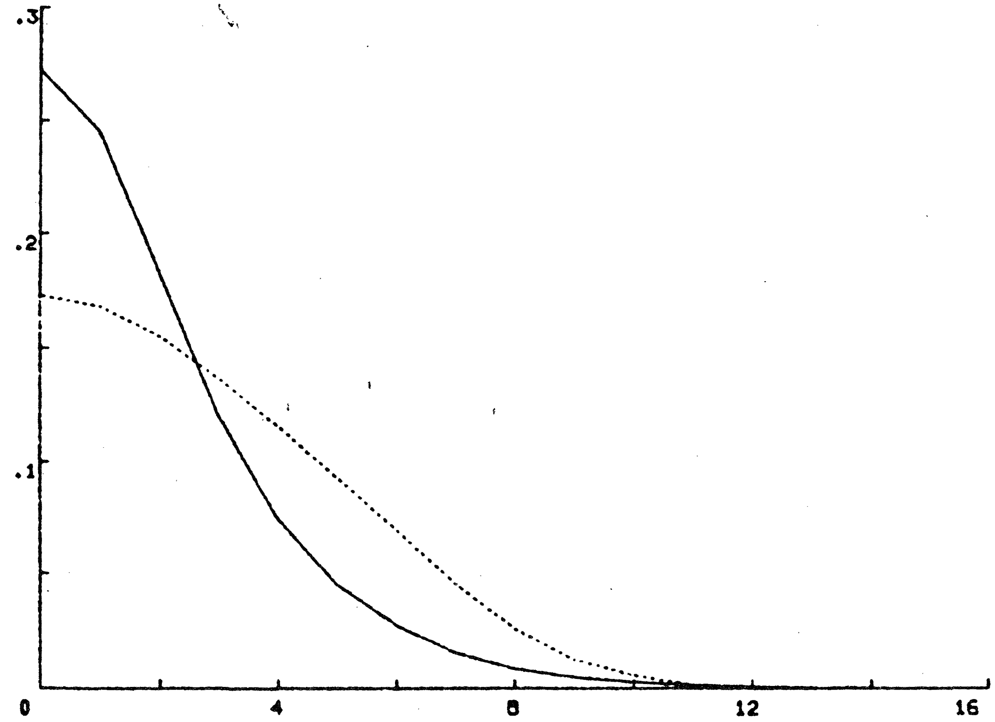

30





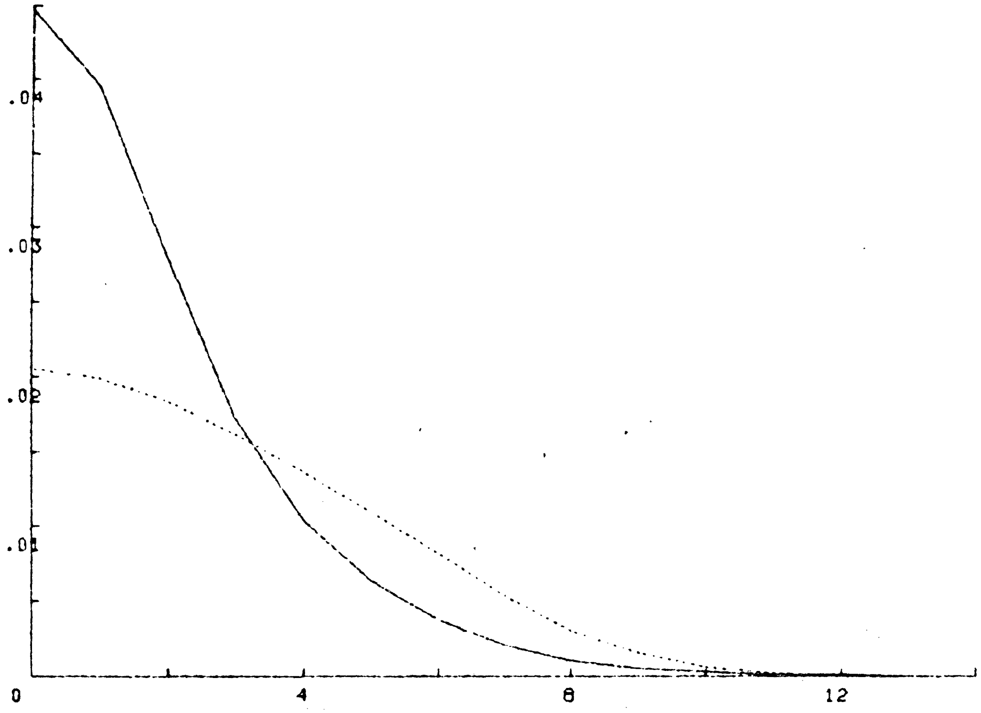
IMPULSE RESPONSE OF UNEQUALIZED (—) AND EQUALIZED CHANNELS

EQUALIZER: 0DB NOISE ENHANCEMENT (10/23/80)



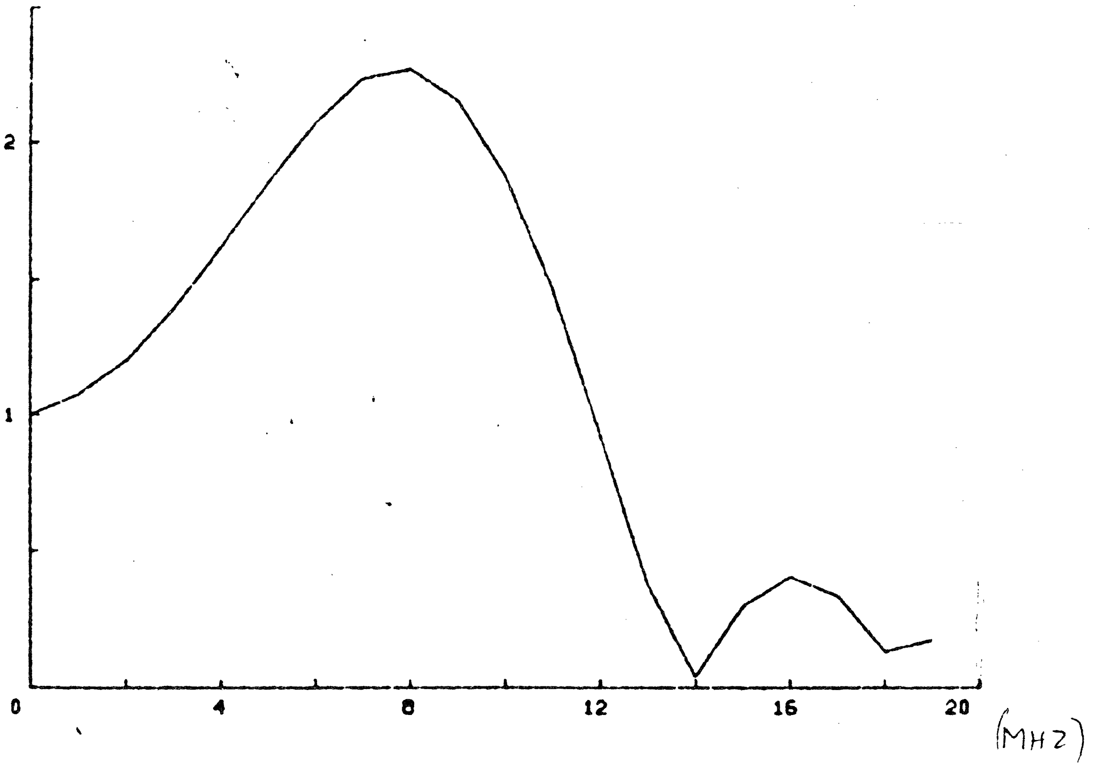
SPECTRUM OF INPUT PULSE AND EQUALIZED PULSE (SQUARED)

(same axes)



FREQUENCY RESPONSE FO UNEQUALIZED AND EQUALIZED (0DB) TRANSITIONS

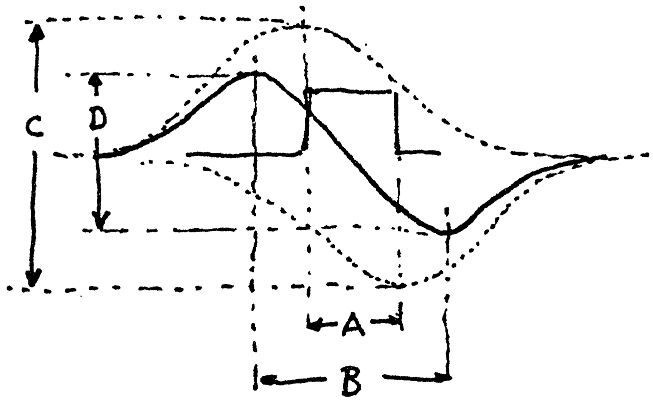
(TRANSITIONS NORMALIZED FOR SAME PEAK)



IMPULSE RESPONSE OF 0 DB EQUILIZER (10/23/80)

Some problems with Peak Detection

1) Intersymbol Interference (ISI) :

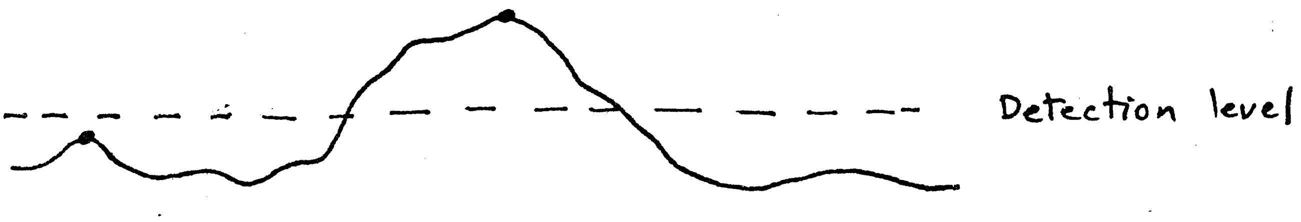


Ex.: For a symmetric dibit:

Peak shift: $sh = \frac{1}{2}(B-A)$

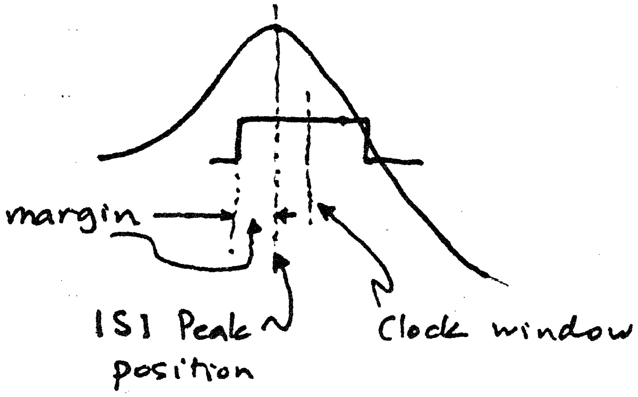
Resolution: $r = \frac{D}{C}$

2) Noise (S/N ratio):



What realy matters
(for peak detectors)

- Must have "reasonable" resolution
- Low detection probability of error



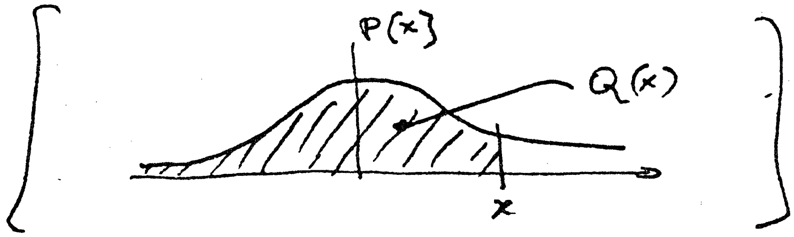
Total peak shift = (ISI shift) + noise + others

$\simeq sh_{ISI} + \text{noise}$

$$\text{Prob}[\text{error}] = \text{Prob}[\text{Peak fall out of the window}]$$

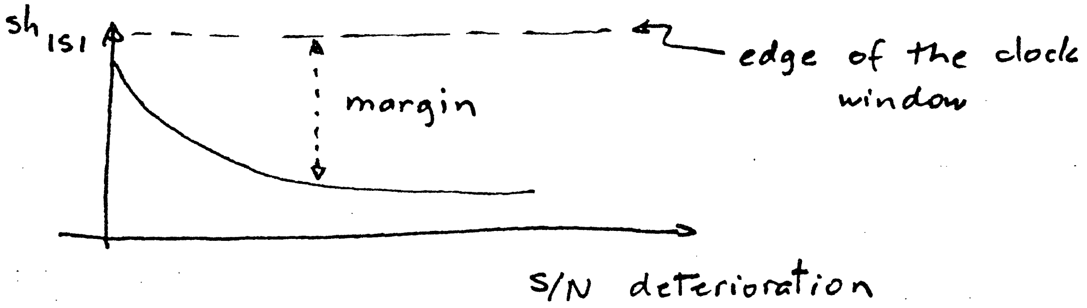
$$= \text{Prob}[\text{noise} > \text{margin} = \text{clock "edge"} - sh_{ISI}]$$

$$= 1 - Q\left\{\frac{\text{margin}}{\sigma}\right)$$

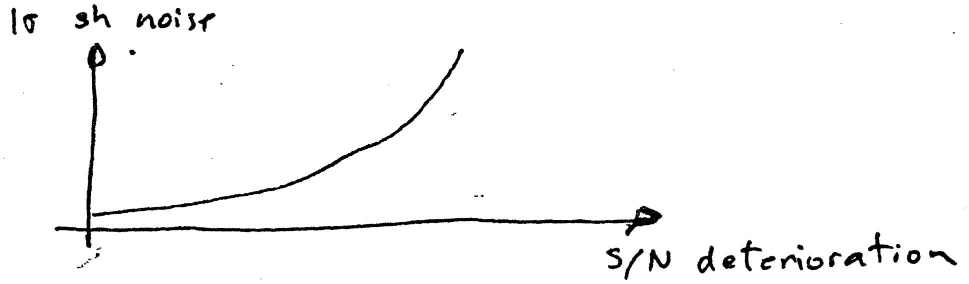


How to choose the equalizer

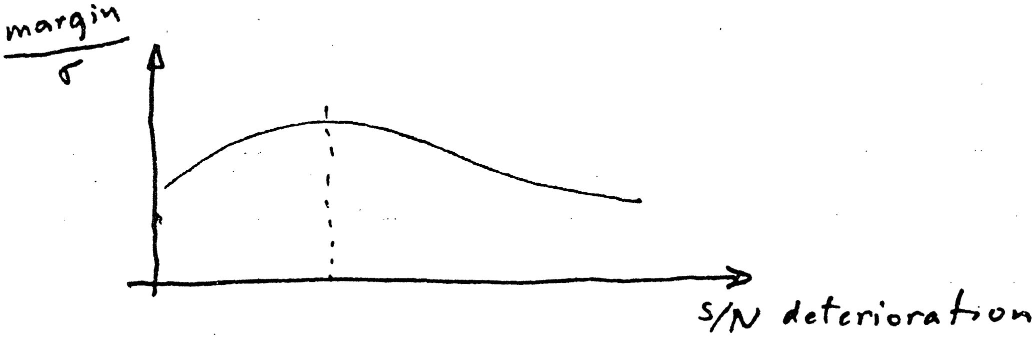
- In general, ISI decreases at expense of noise deterioration :

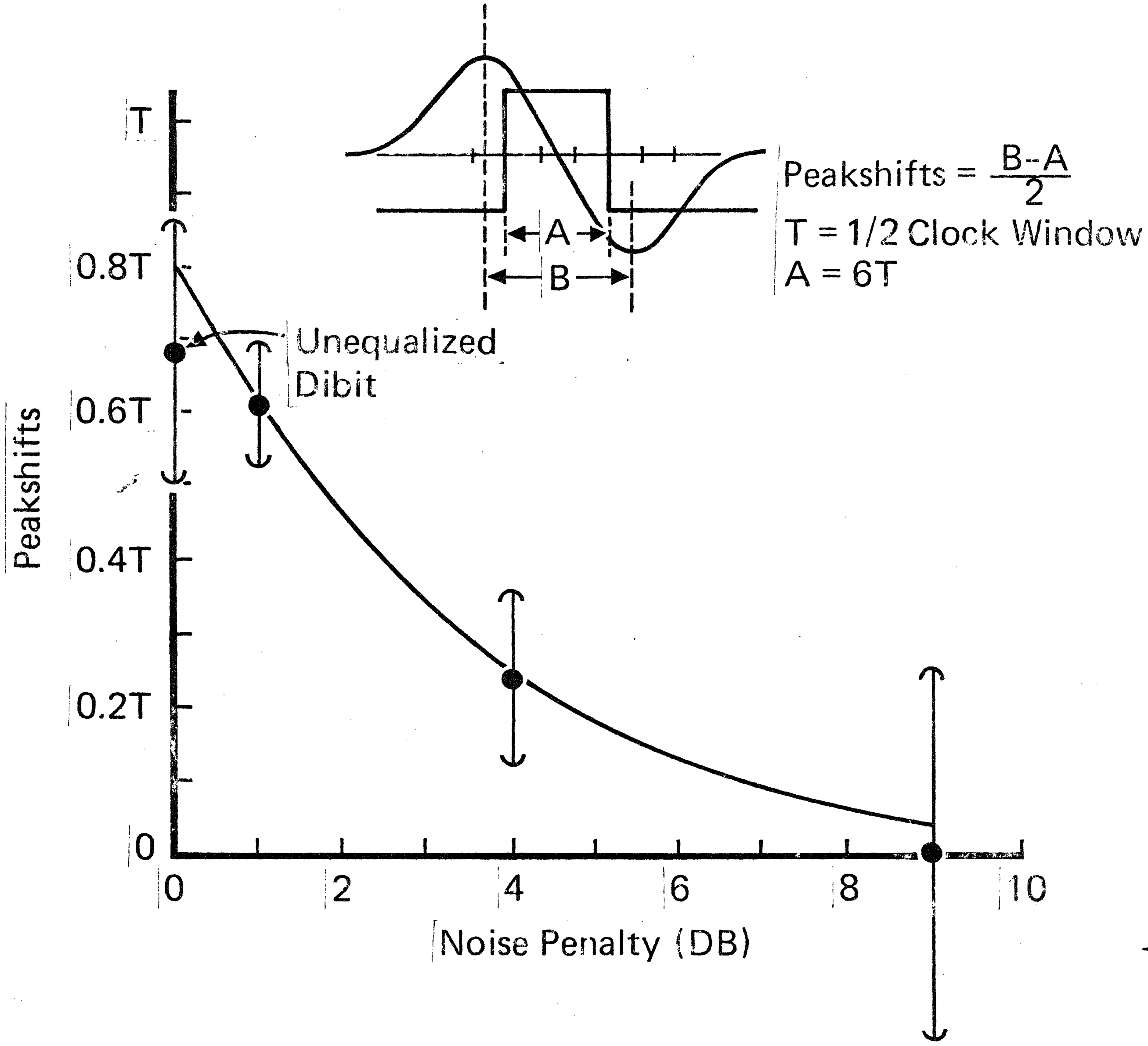


- Peak shift noise increases with background noise:



As a result




T
0.8T
0.6T
0.4T
0.2T
0
Peakshifts
0
2
4
6
8
10
Noise Penalty (DB)
Unequalized
Dibit
A
B
Peakshifts = $\frac{B-A}{2}$
T = 1/2 Clock Window
A = 6T

ABSTRACT

Two approaches in implementing adaptive equalization are presented in this talk.

The basic functional block in the first approach is a programmable transversal filter whose tap weights are adjusted according to an error correction algorithm which is a modified version of the mean square estimation. It is explained why the hybrid approach is proposed to implement in a LSI form the transversal filter which performs the convolution using a CCD tap delay line and multiplying DAC's, one for each tap. Due to the stringent cost requirement for the project at hand, it is found necessary to approximate the linear correlation required by the mean square estimation. The effects of such approximation on the convergence time and the adaptivity (to be defined later) are investigated using APL simulation.

A design for the equalizer discussed above is presented for a discrete realization.

Another approach* is given as an alternative to the transversal filter equalizer, which is especially suitable for a discrete implementation. This approach is based on three

assumptions made on the isolated pulse. They are:

(1) The pulse is spread over not more than four bit cells

(2) The pulse is symmetric

(3) The pulse peak amplitude is predetermined. In other words, an AGC loop precedes this equalizer.

It is noted that the latter can be described as a simplified version of the former, as evidenced by comparing two respective designs.

* Richard Schneider in Tucson is the originator of this approach.

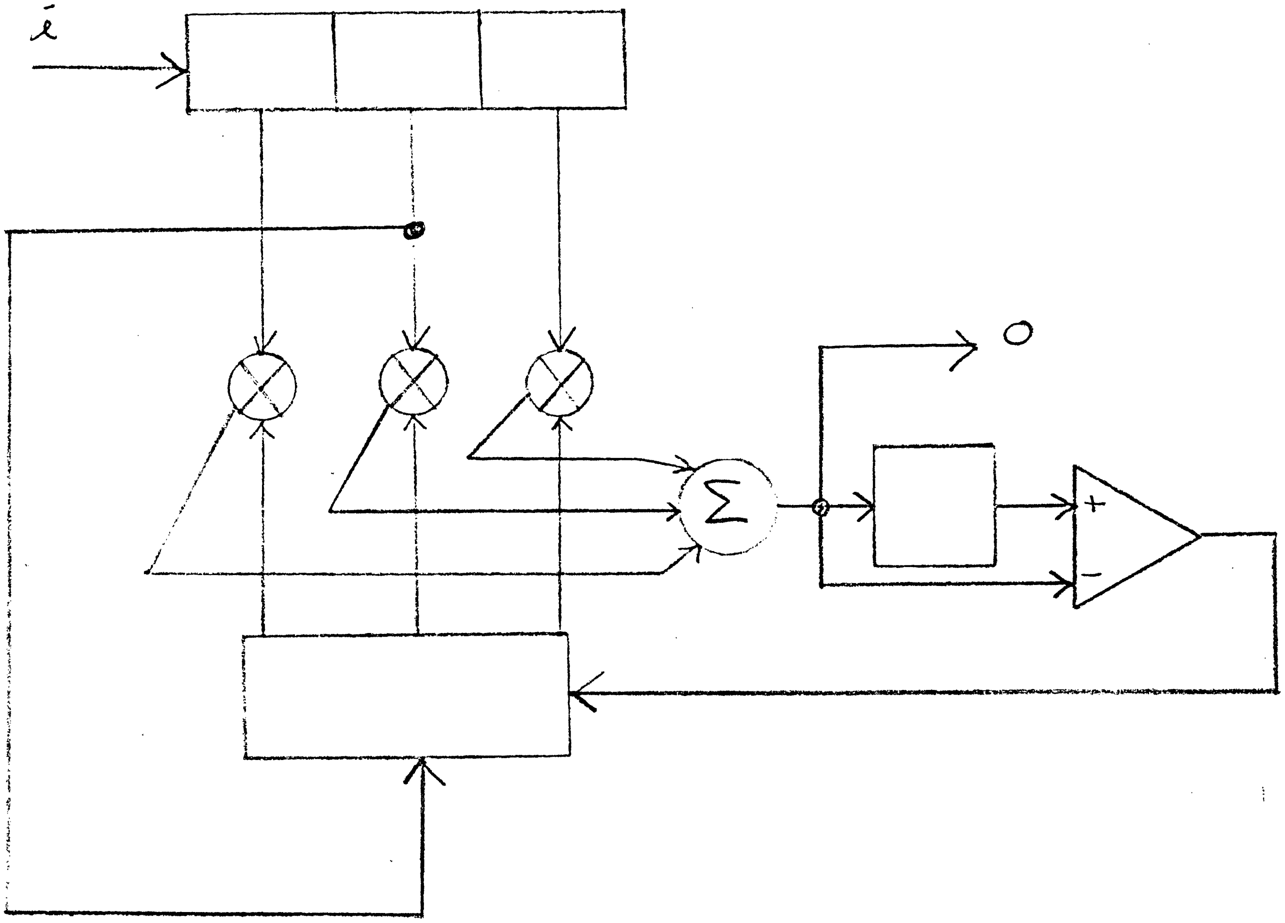
Adaptive Equalization

Gene Sonu
GPD-TUCSON
Jan. 1981

TOPICS

- ADVANTAGES
- Typical Adaptive Transversal Filter Equalizer
- Hardware
- Other Type of Adaptive Eq.
- DISCUSSION, IF ANY

Typical Adaptive Transversal Filter Equalizer



Transversal Filter Implementation

All Digital: Cost (Hardware COMPLEXITY)

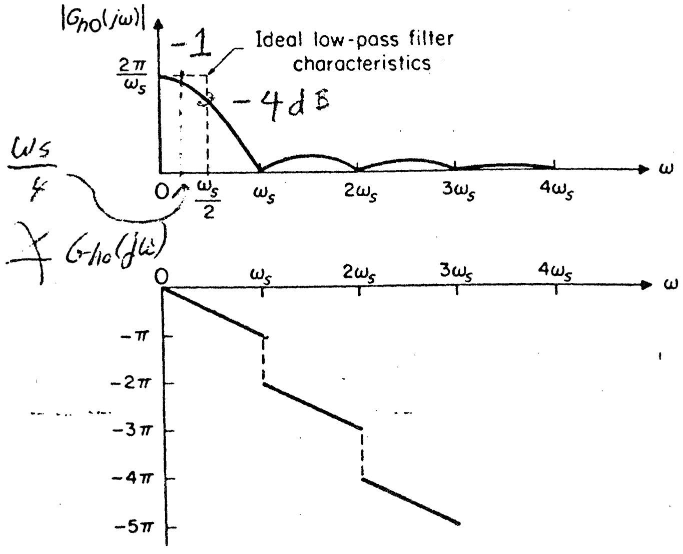
All Analog: Analog Multiplier

Sample + Hold

⇒ low freq. Applications

Hybrid:

CCD TAP Delay Line & MDAC



IBM CONFIDENTIAL

Tap Adjustment Algo : MSE

$$e_k \triangleq y_k - d_k \qquad \forall\ t = kT$$

$$\epsilon \triangleq E\{e_k^2\}$$

$$C_j^{k+1} = C_j^k + \Delta C_j^k$$

$\forall$ j th tap from ~~the~~ center one.

where

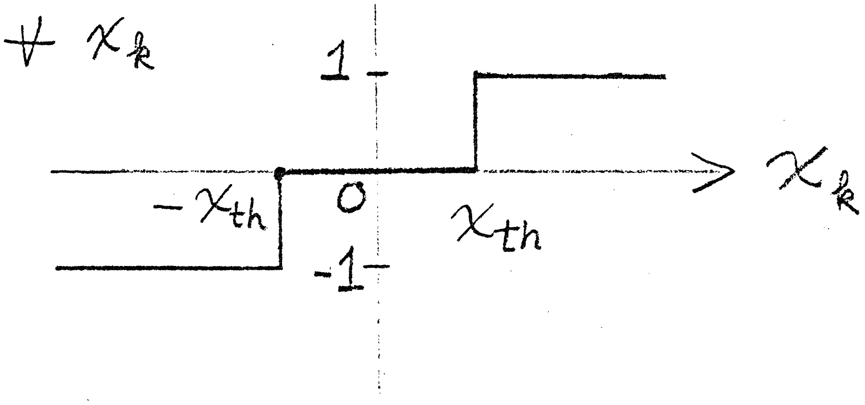
$$\Delta C_j^k = \delta \cdot E\{e_k \cdot x_{k-j}\}$$

$$\boxed{\Delta C_j^k \simeq \delta \cdot e_k \cdot x_{k-j}}$$

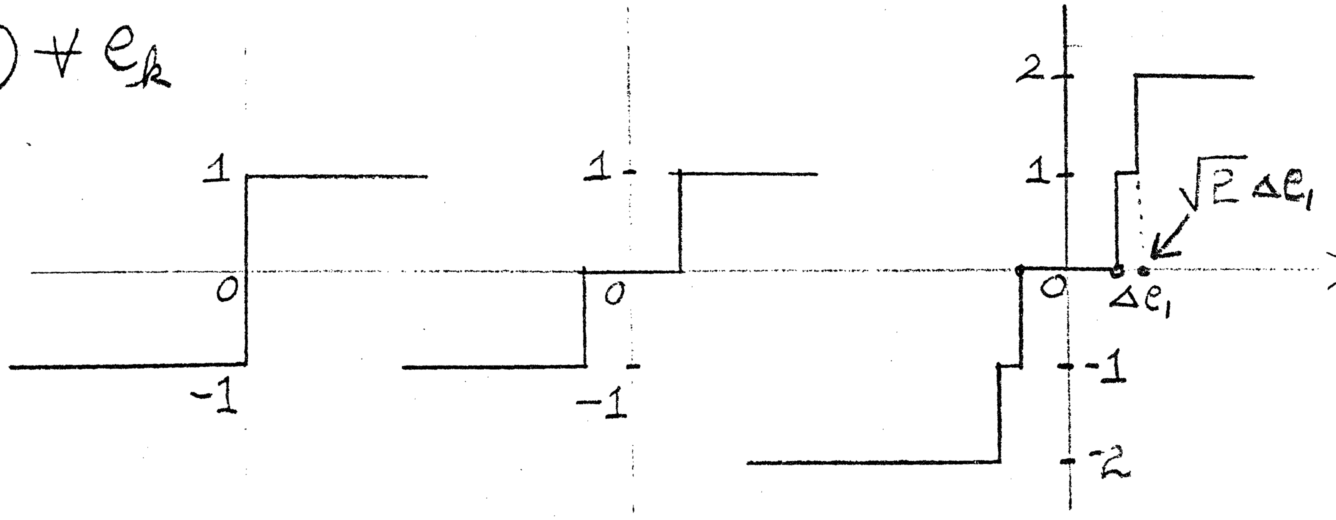
Further Modification:

Level Quantization

1) $\forall\ x_k$



2) $\forall\ e_k$



Sign

3-level

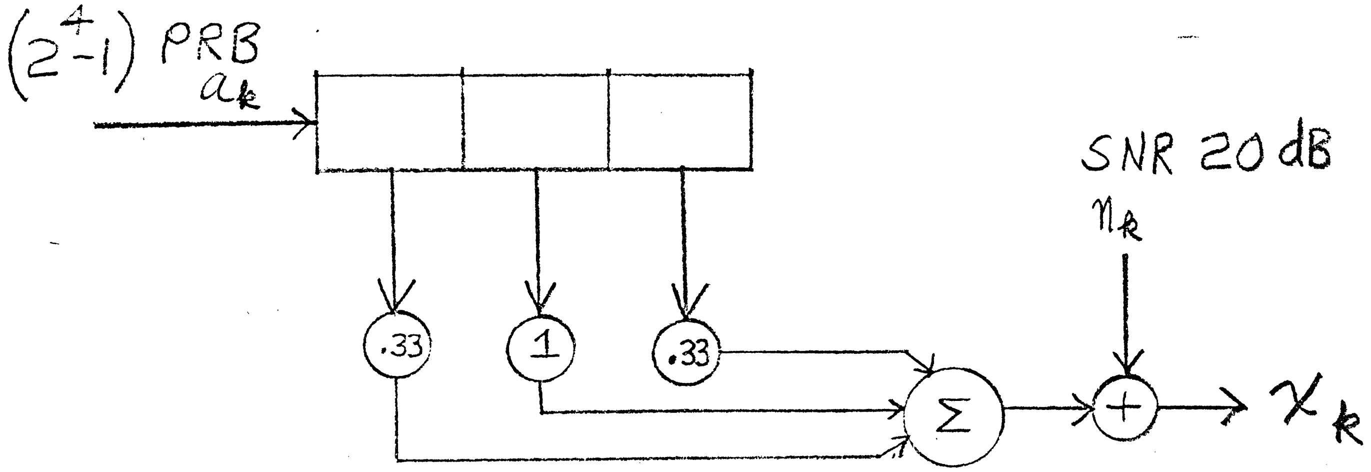
5-level

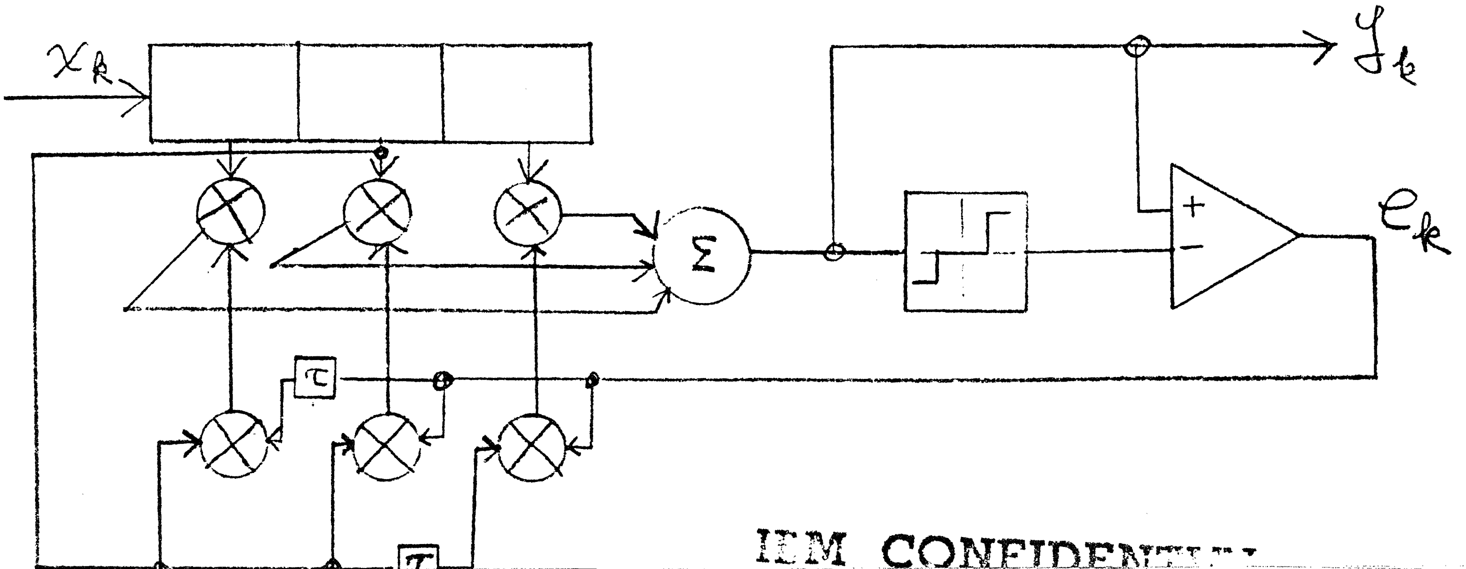
Effect of Quantization

1 Converge?
2. Convergence Time
3. Adaptivity (Riegler, 1973)

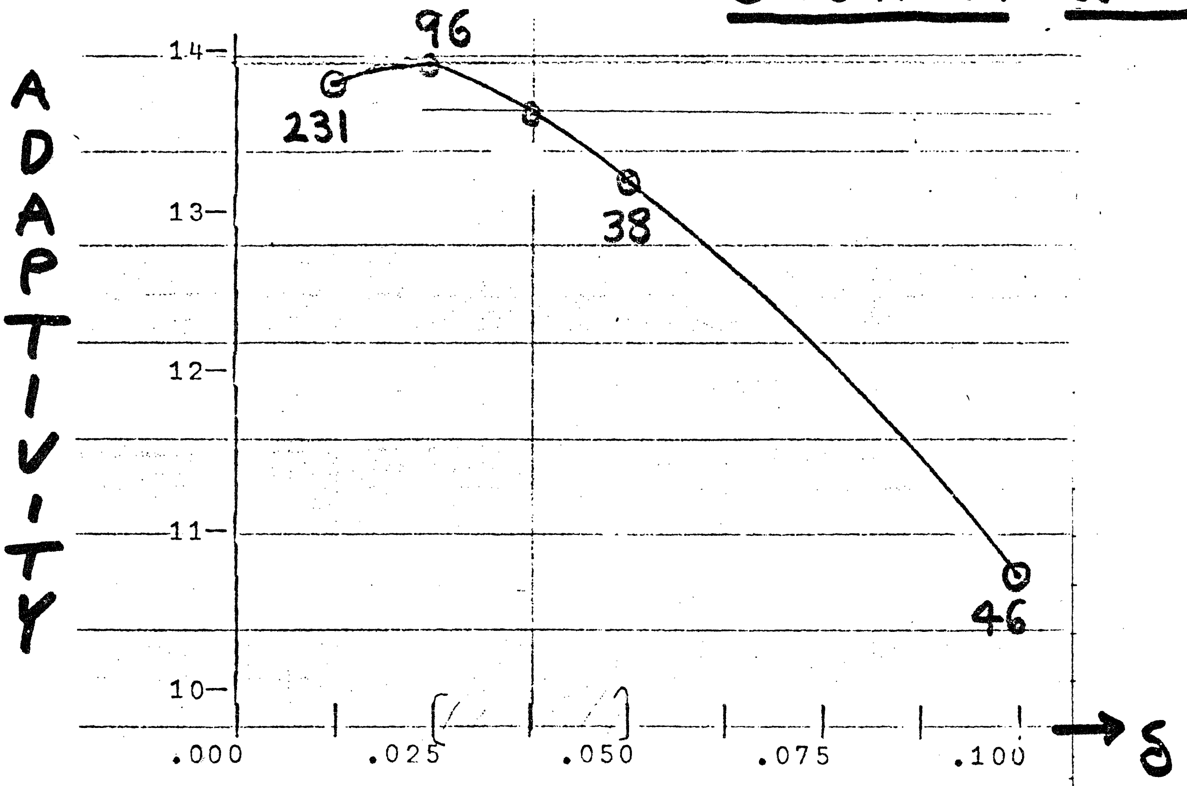
$$\hat{=} 20 \log_{10} \frac{\sigma_d^2}{\sigma_e^2}$$

APL Implementation



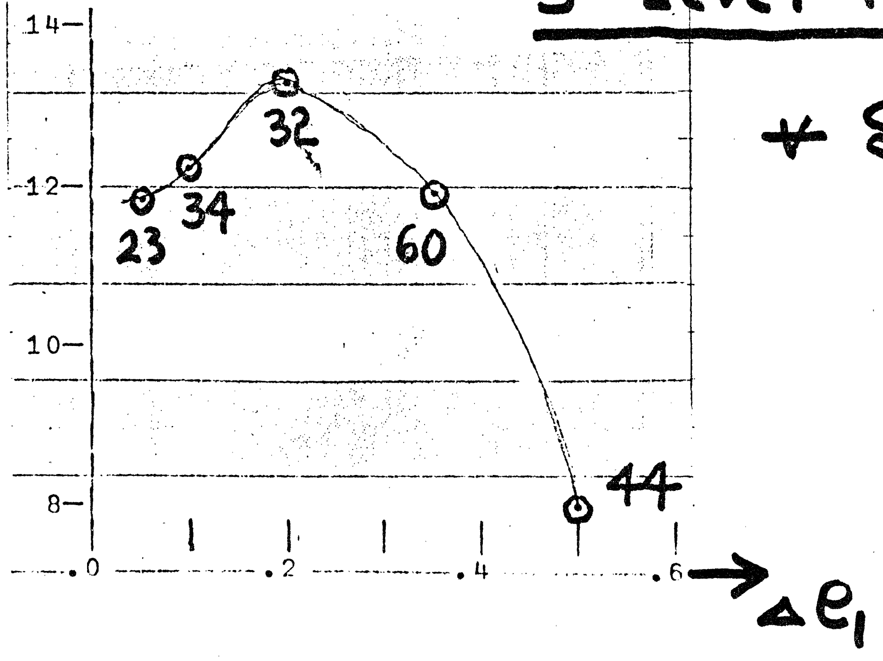


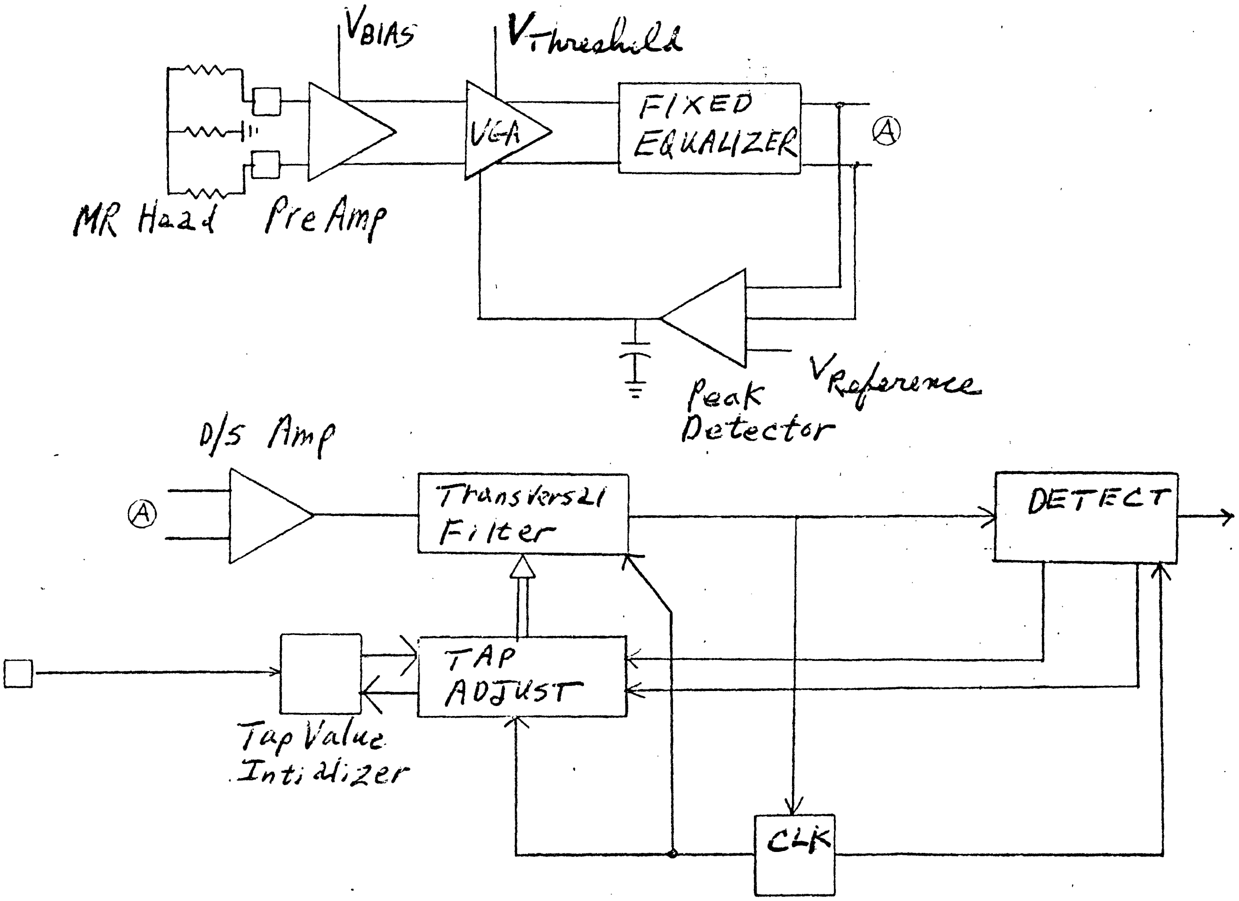
SIGNUM QUANTIZER



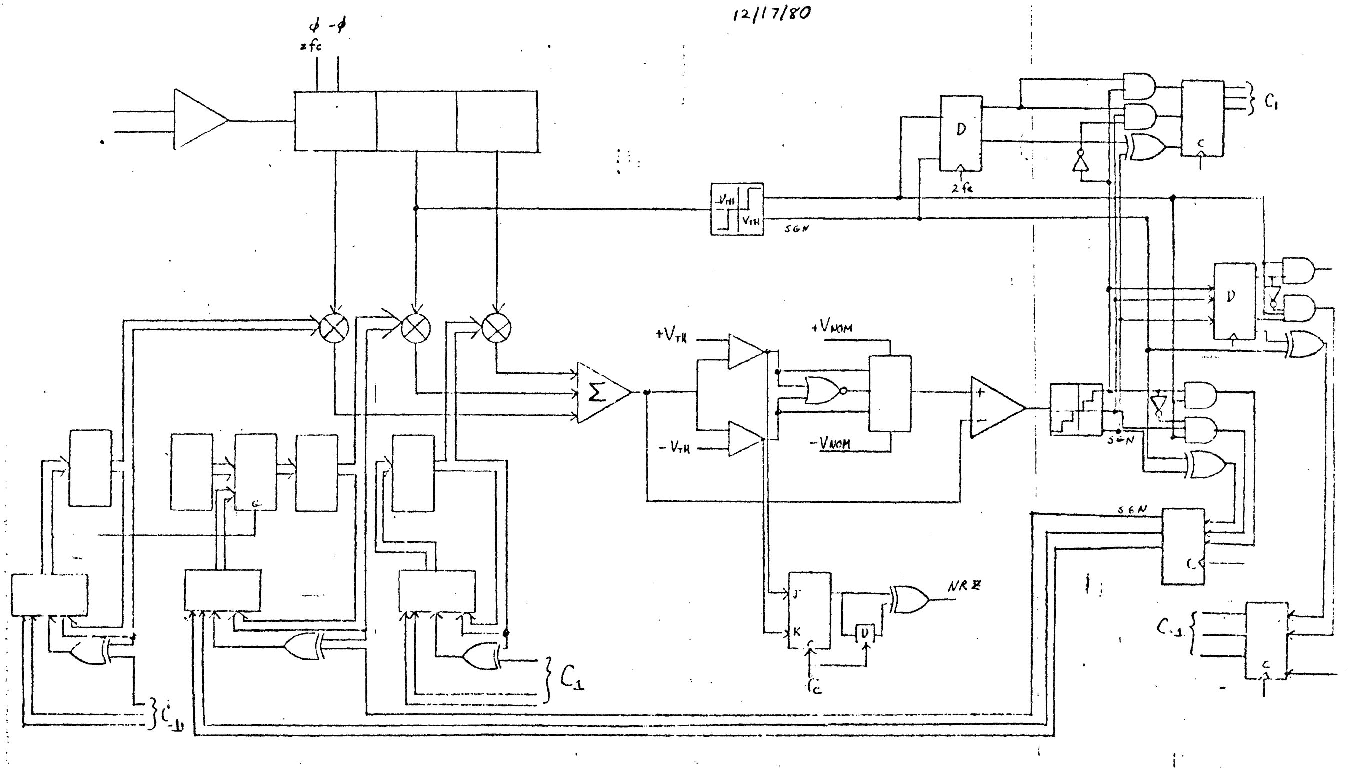
5-Level Parabolic

∀ δ = .04




VBIAS
VThreshold
VGA
FIXED EQUALIZER
A
MR Head
PreAmp
Peak Detector
VReference
D/S Amp
A
Transversal Filter
DETECT
TAP ADJUST
Tap Value Intializer
CLK

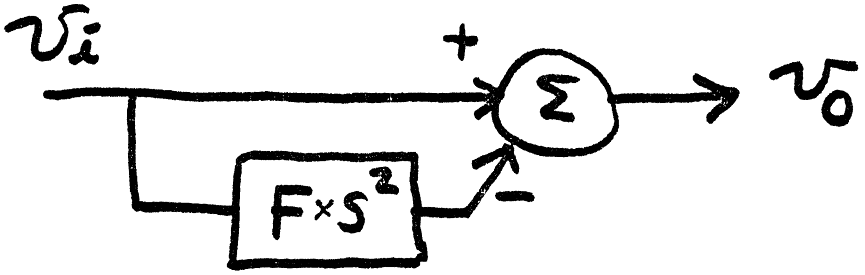
G. Sonu
12/17/80

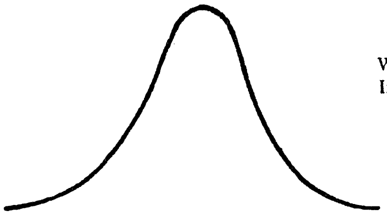


Other Type of Adaptive Eq.

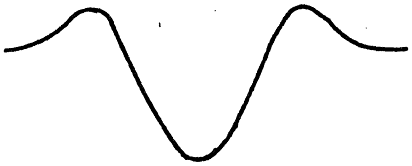
$$H(s,t) = 1 - F(t)\, s^2$$

$$\forall\ F(t) = \text{Const},$$

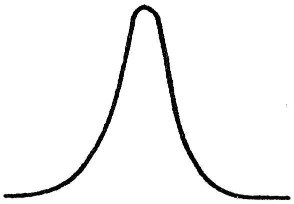




Wide
Input Pulse

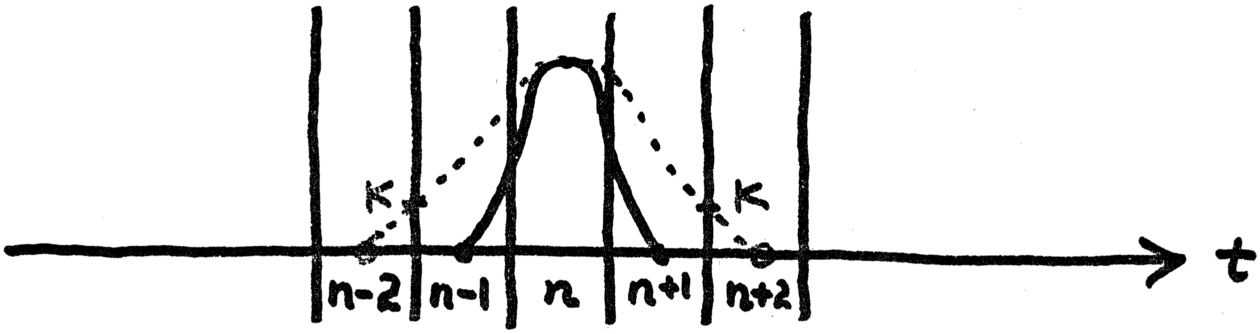


Second Derivative
of Input Pulse



Slimmed Pulse

3 Assumptions



$$S_n = K\, D_{n-1} + D_n + K\, D_{n+1}$$

$$\Rightarrow \quad K = \frac{S_n - D_n}{D_{n-1} + D_{n+1}}$$

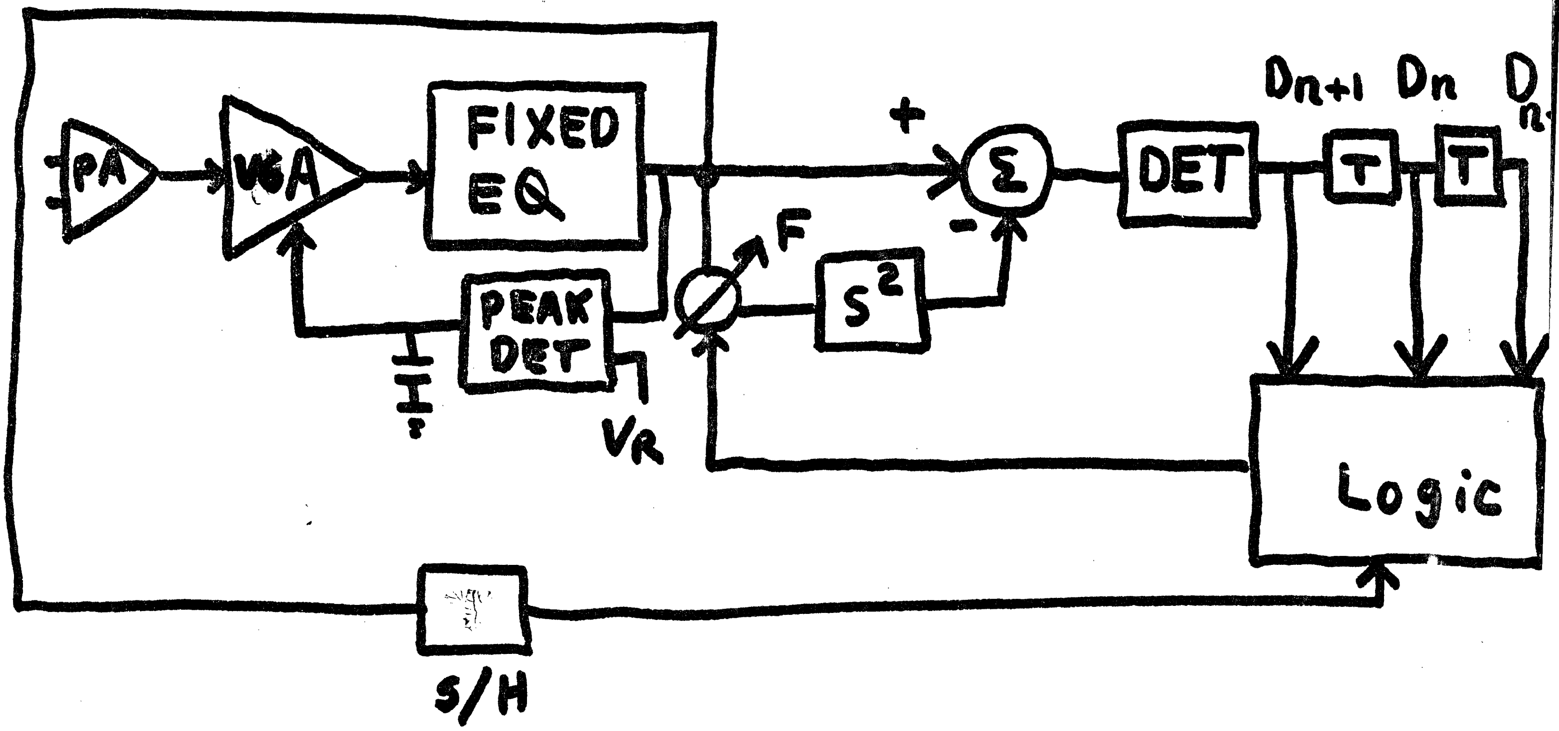
$$E_{n+1} = S_{n+1} + (F_n + \Delta F)\, S''_{n+1}$$

where

if $K > 0$ $\quad + \Delta F$

if $K < 0$ $\quad - \Delta F$

if $D_{n-1} + D_{n+1} = 0,$ $\quad 0$



Magnetic Recording:

"It's not just an industry,
its a way of life"

Equalization

and

Signal-Noise Ratio

D P Turner

1/14/81

IBM CONFIDENTIAL

Abstract: "Equalization and S/N Ratio"
January 14, 1981 D. P. Turner

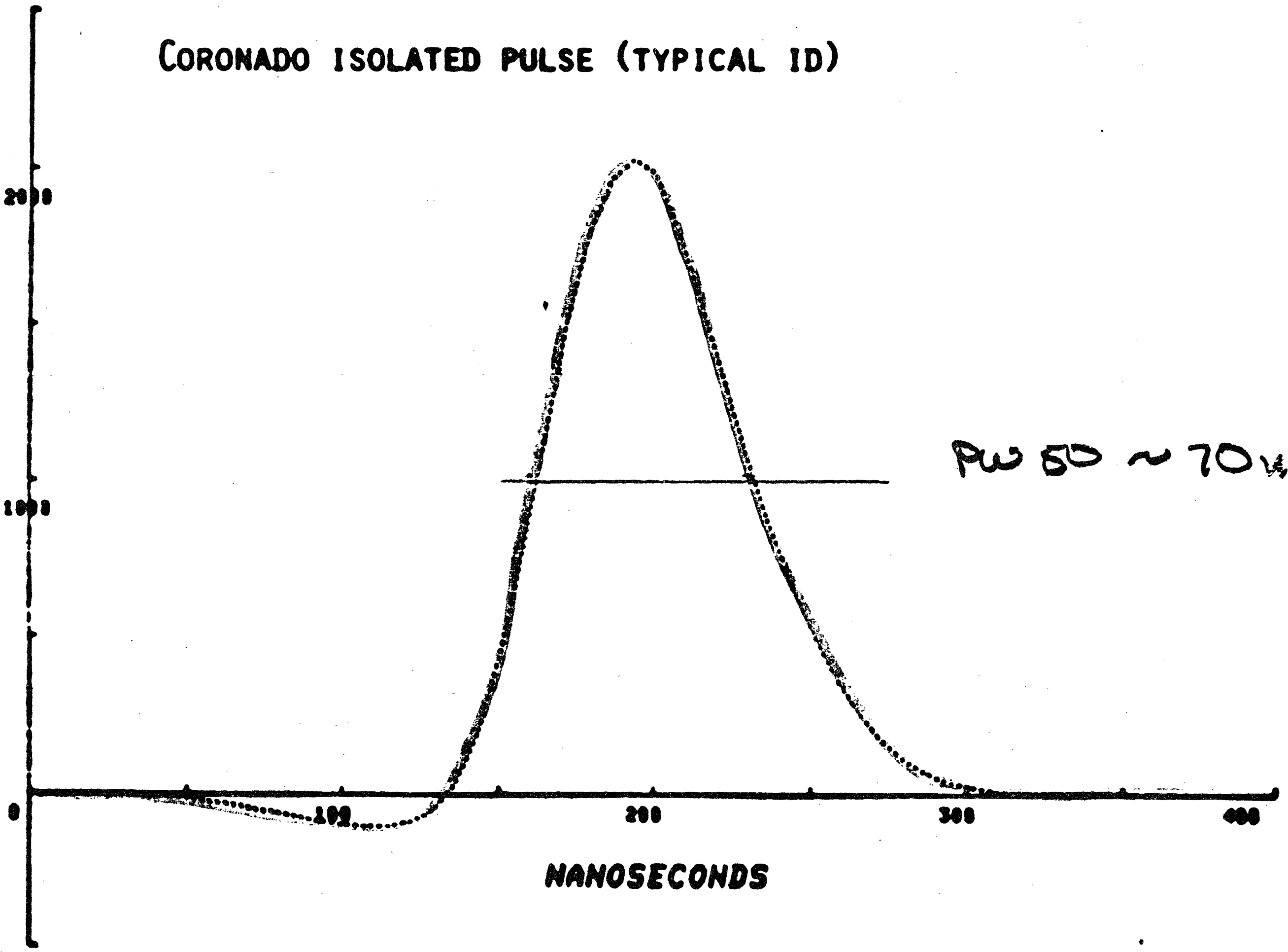
Our early work in equaization has been based on the frequency domain equalizer programs developed by Jud McDowell in Tuscon. We have attempted to design an equalizer to operate the 3380 head-disk combination at 1.25 times the 3380 data rate. The equalizer design involved measuring the unequalized channel frequency response and deriving an equalizer which would match the desired "sine channel". The equalizer so designed resulted in reduced on track bit shift compared to the unequalized channel. However, when tested on a precision test stand the equalized channel, although it was a better performer than the unequalized channel, did not perform adequately for a product.

We are very concerned that future programs are going to be faced with serious signal to noise ratio problems. We have made experiments which show error rates dropping very rapidly with increased noise levels. Scaling exercises predict Frontier signal-noise ratios of 13 to 26 dB. These signal-noise ratios may make equalizer design extremely difficult.

We have recently discovered three papers on equalization by Kameyama, MacIntosh, and Taub which treat equalization in the time domain and which all come up with similar results predicting limits to which equalization may be gainfully employed. The pulse width reduction obtainable by reasonable equalizers indicates that doubling the data rate would be possible, but further detailed analysis of dibit bitshift and tribit amplitude loss indicates that the potential gains are considerably less, say a 20 - 30 % improvement in linear density. Kameyama even shows a case where if the initial signal-noise ratio is high enough the linear density obtainable with equalization is less than that obtainable without. We are presently working to verify some of these results using 3380 components.

PLA LPLT ISPBAVE VS XXS
BAB

IBM CONFIDENTIAL

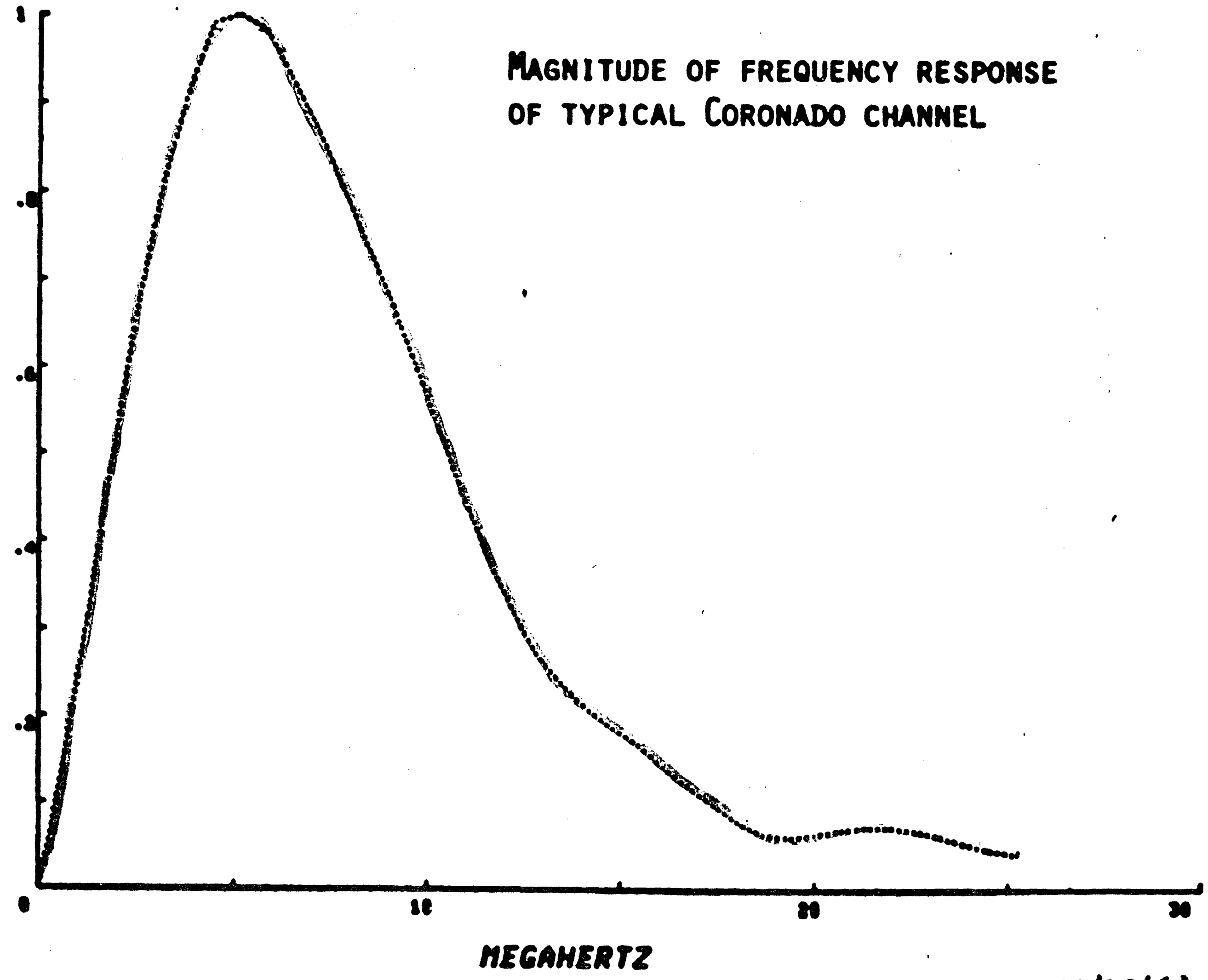


PW50 ~ 70 nsec

10/20/80
OPT

PLA LPLT CIDSP28COR(1)) VS CIDSP28COR(3))
BAB

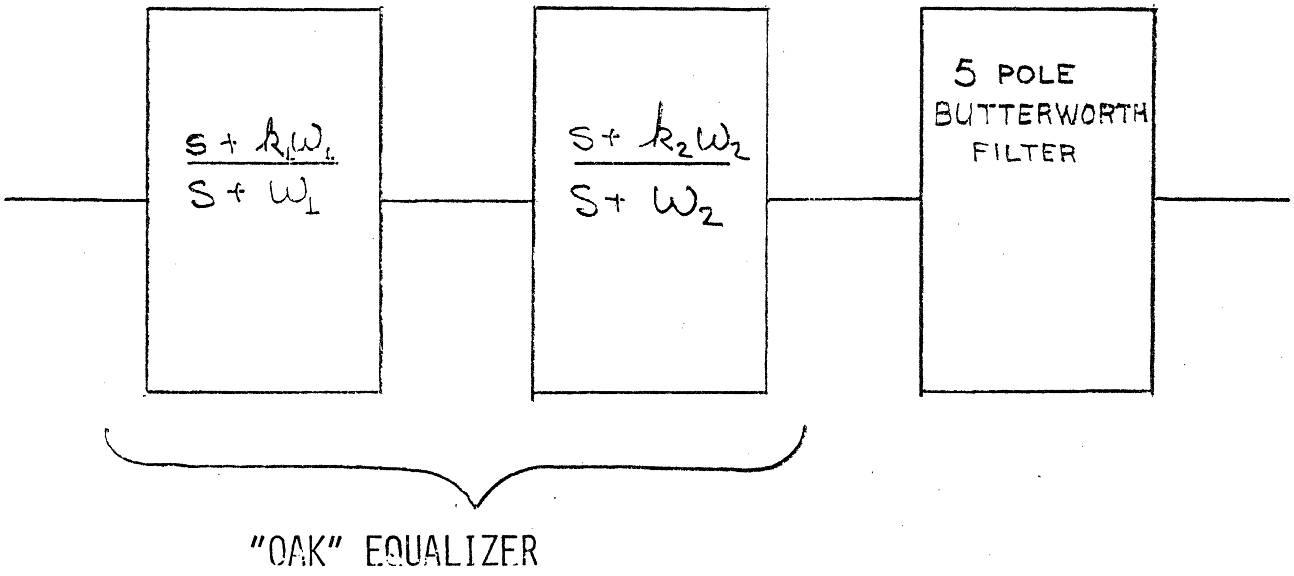
IBM CONFIDENTIAL



MAGNITUDE OF FREQUENCY RESPONSE OF TYPICAL CORONADO CHANNEL

10/20/80 DPT

EQUALIZER FOR 1.25 TIMES CORONADO DATA RATE

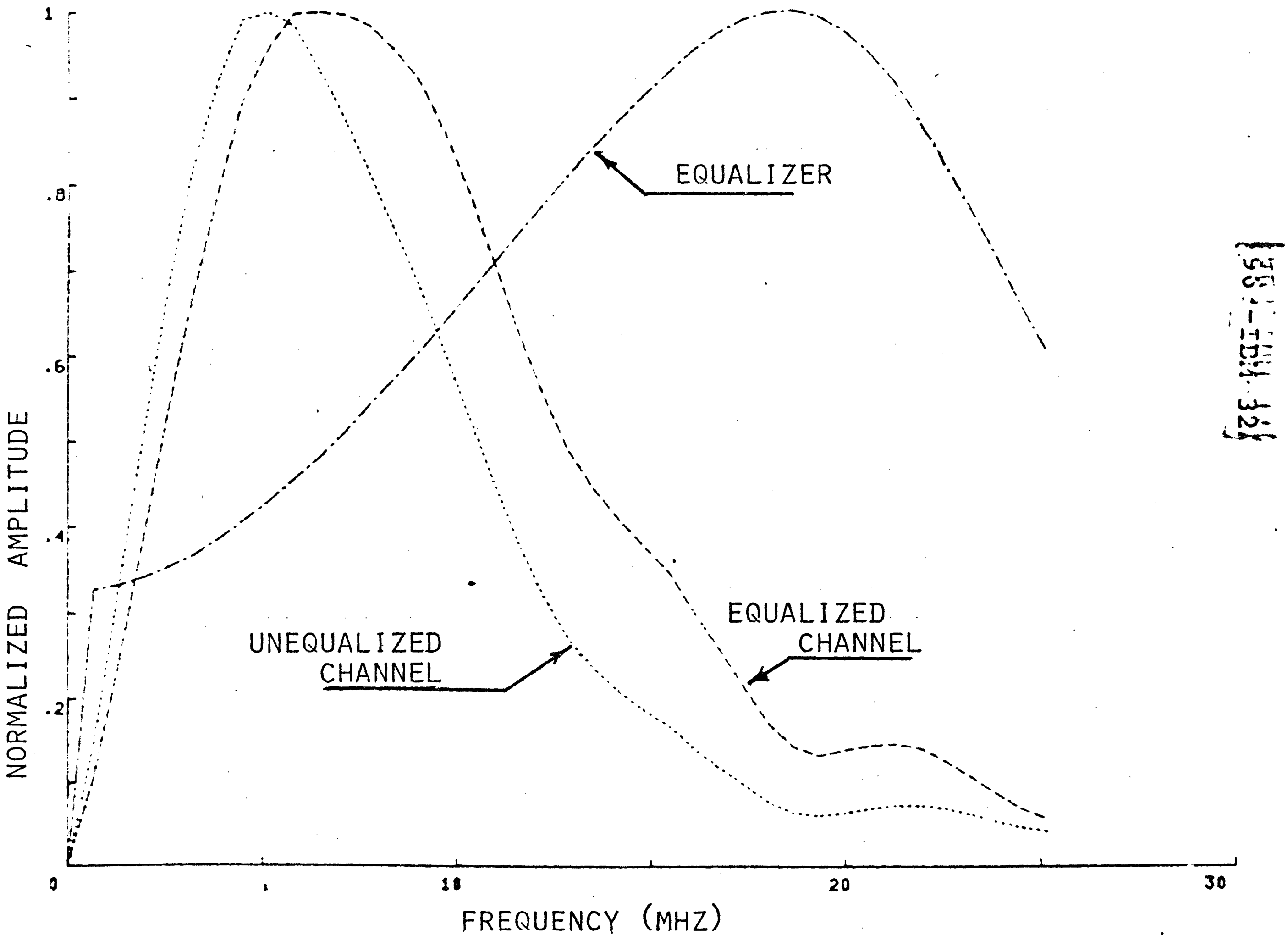

$\frac{s+k_1\omega_1}{s+\omega_1}$
$\frac{s+k_2\omega_2}{s+\omega_2}$
5 POLE BUTTERWORTH FILTER
"OAK" EQUALIZER

$K_1 = .38$ $\omega_1 = 2\pi \times 20 \times 10^6$

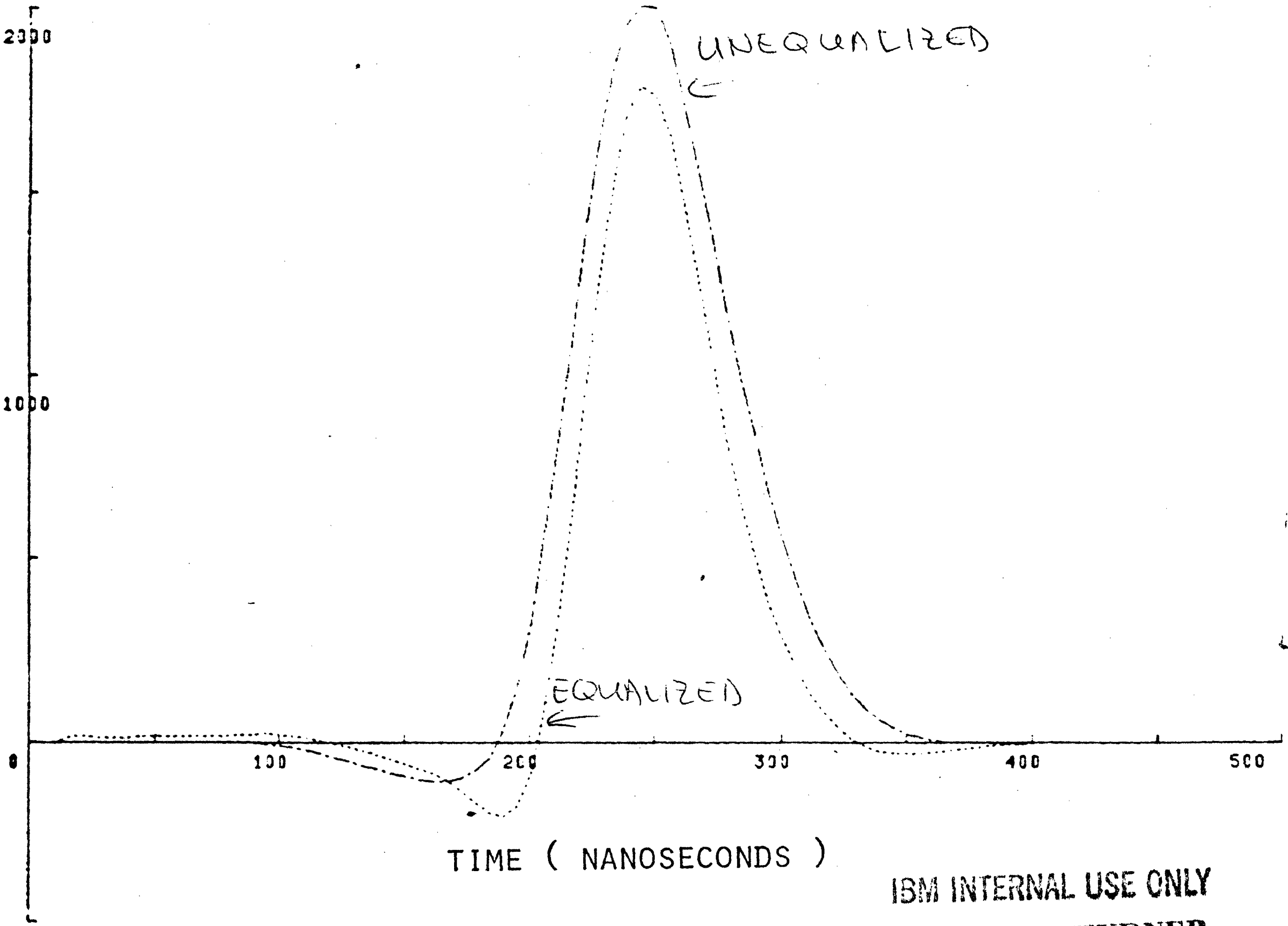
$K_2 = .43$ $\omega_2 = 2\pi \times 20 \times 10^6$

5 POLE BUTTERWORTH FILTER: $\omega_N = 2\pi \times 22 \times 10^6$

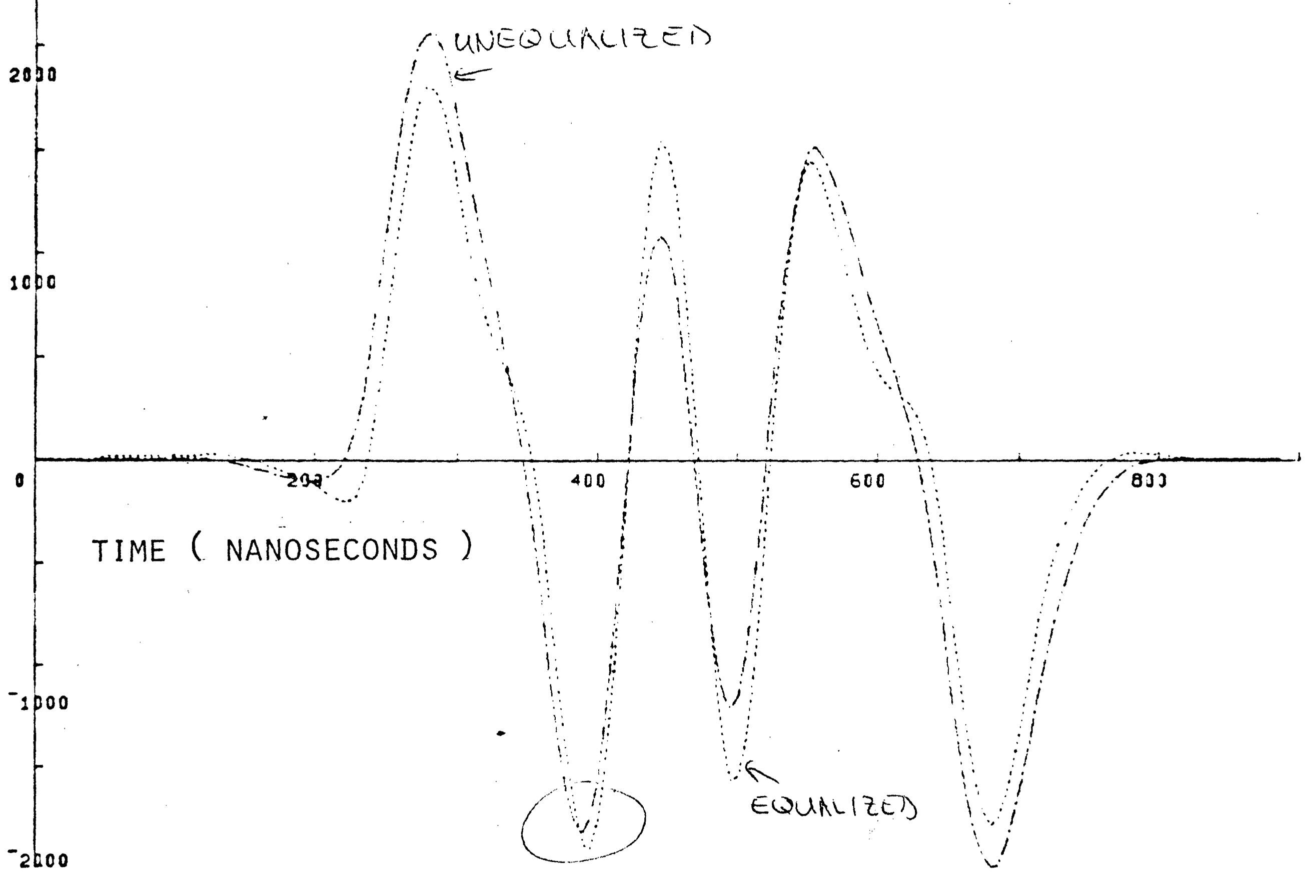
IBM INTERNAL USE ONLY
12/1/80 DPT


1
.8
.6
.4
.2
NORMALIZED AMPLITUDE
0
10
20
30
FREQUENCY (MHZ)
EQUALIZER
EQUALIZED
CHANNEL
UNEQUALIZED
CHANNEL

ISOLATED PULSE



READBACK SIGNAL SEQUENCE



NOTE BIT SHIFT IMPROUEMENT

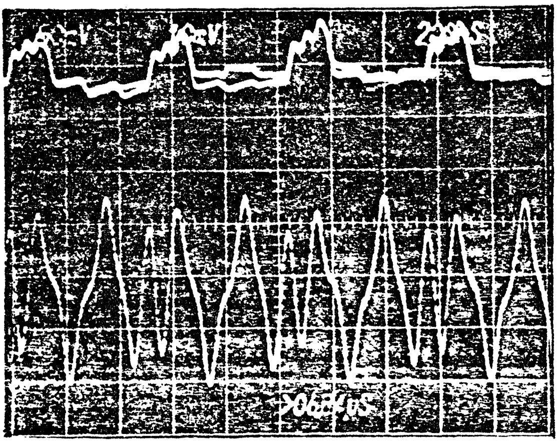
D. P. TURNER

RESULTS ON SPINNER

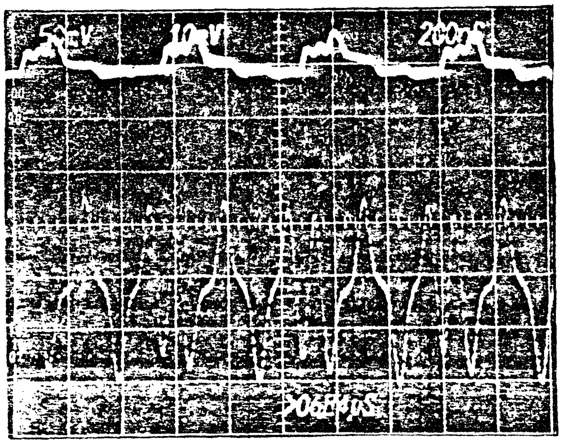
CLOCK RATE = 60 MHz (1.25 TIMES CORONADO)

WITHOUT EQUALIZER (BUTTERWORTH FILTER ONLY)

ID

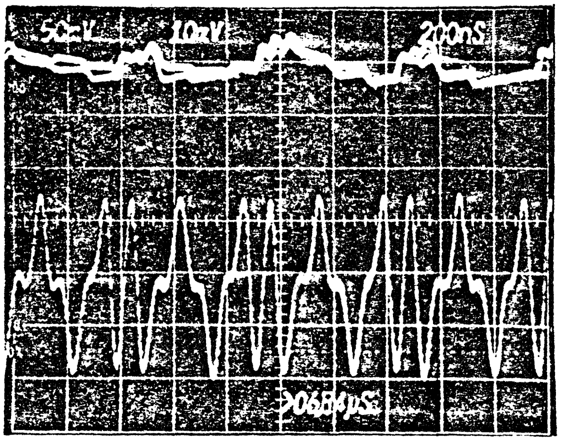


MD

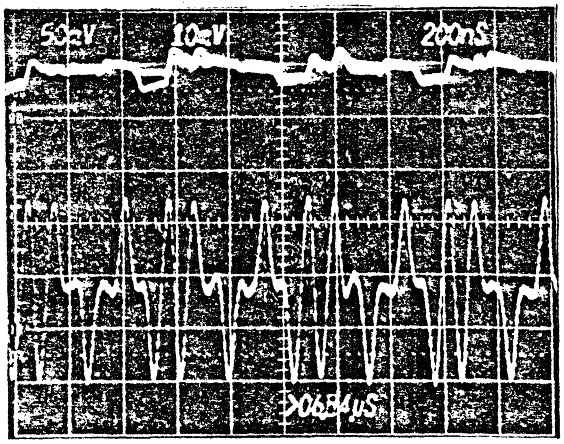


WITH EQUALIZER

ID

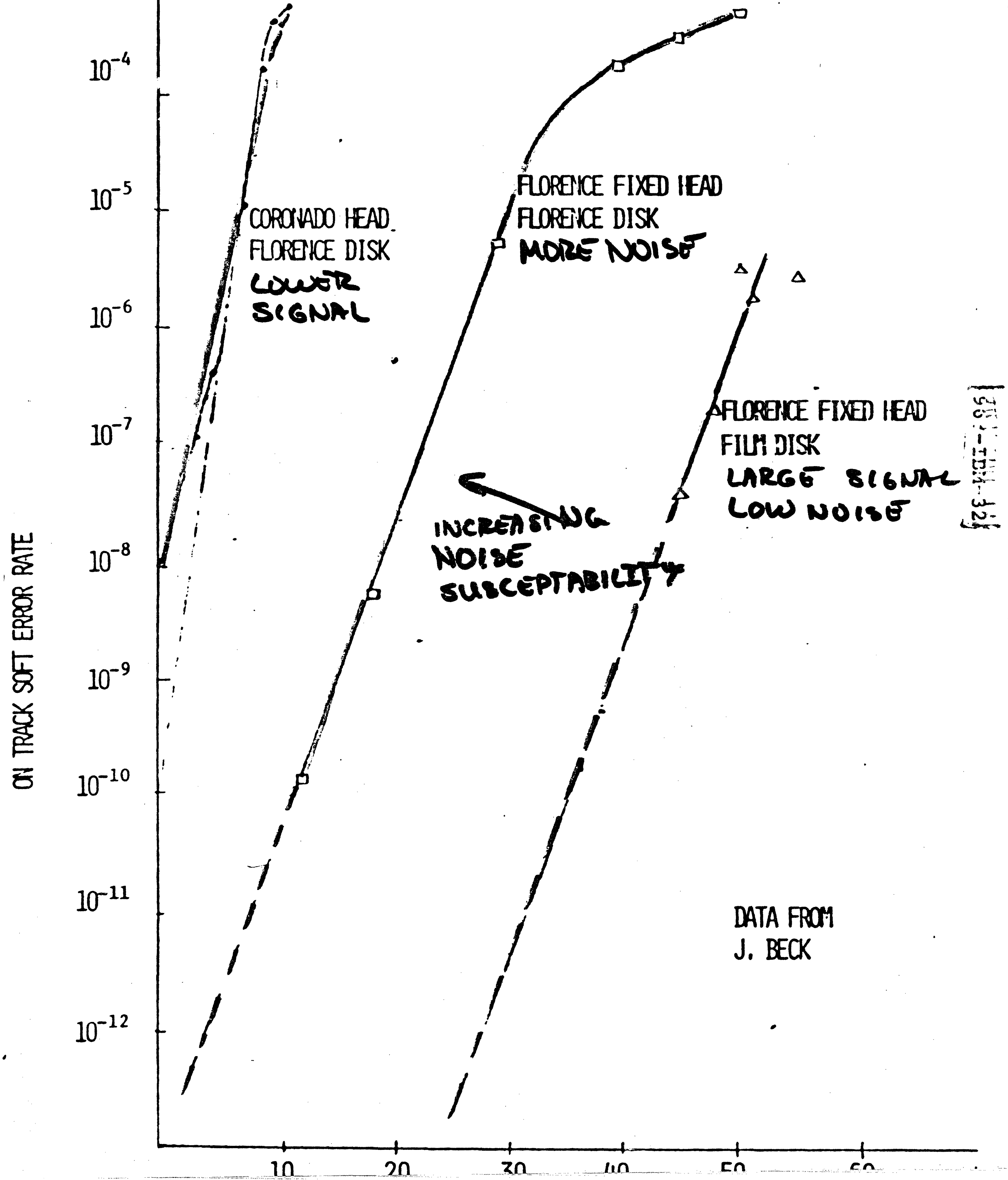


MD



IBM INTERNAL USE ONLY
12/1/80 DPT

IBM CONFIDENTIAL
TNK 3/12/80



DATA FROM
J. BECK

IBM Confidential

SIGNAL, NOISE AND CHANNEL CAPACITY
PARTICULATE AND AURORA DISKS

	Particulate Disk		Aurora		Channel Capacity (bits/sec/Hz)	
Environmental Noise (µV rms)	Total Noise(µVrms)	Signal (µVrms)	Total Noise(µVrms)	Signal (µVrms)	Particulate	Aurora
1.0	3.3	170	2.4	112	0.653	0.76
2.0	3.7		3.0		0.62	0.68
3.0	4.4		3.7		0.58	0.62
4.0	5.1	(≈1 order	4.6		0.54	0.57
5.0	5.9	in SER)	5.5			
6.0	6.8		6.4			
7.0	7.7		7.3			
8.0	8.6		8.3			
9.0	9.5		9.3			
10.0	10.5		10.2			

10/17/80

RLComstock

Signal / Noise

Disk	RMS Signal	RMS Noise	S/N	S/N (dB)
thin film	109 μV	4.3 μV	25.4	28.1
particulate	153	6.5	23.6	27.4
	110	6.0	18.2	25.2
	115	6.4	17.9	25.1
	107	6.7	15.9	24.0
	111	8.0	13.8	22.8
	117	9.7	12.0	21.6

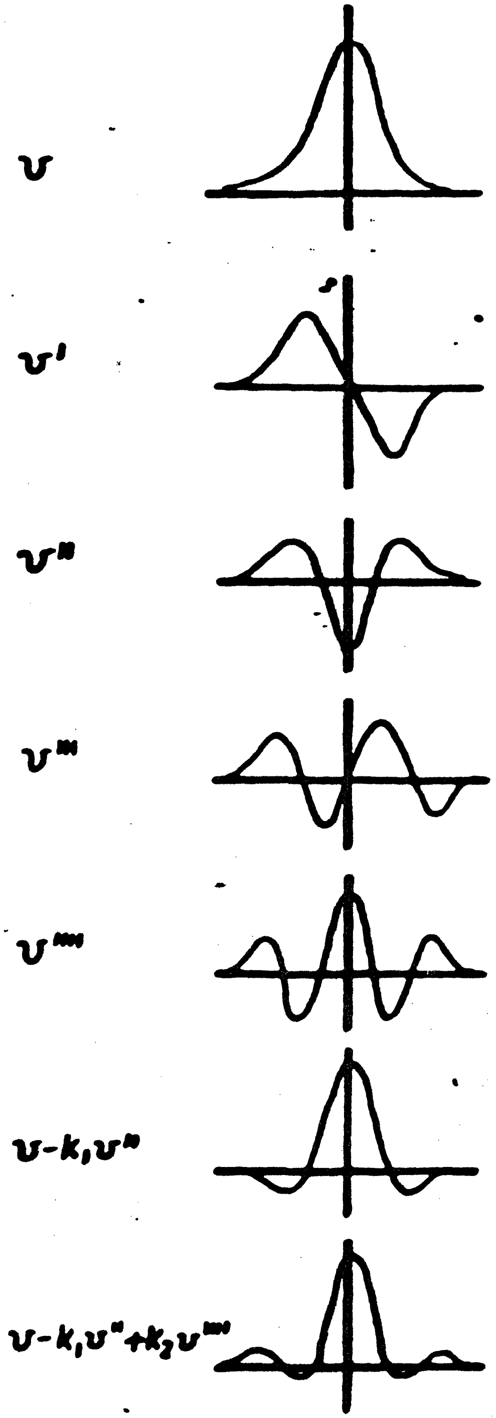
THREE PAPERS ON EQUALIZATION

1. T. KAMEYAMA, S. TAKANAMI AND R. ARAI
   "IMPROVEMENT OF RECORDING DENSITY BY MEANS OF COSINE EQUALIZER"
   IEEE TRANS. MAGNETICS, VOL MAG-12 #6 PP 746-748 NOV, 1976.

2. N. D. MACKINTOSH
   "A SUPERPOSITION-BASED ANALYSIS OF PULSE-SLIMMING TECHNIQUES FOR DIGITAL MAGNETIC RECORDING"
   PROCEEDINGS OF THE CONFERENCE ON VIDEO AND DATA RECORDING,
   UNIVERSITY OF SOUTHHAMPTON, HAMPS, ENGLAND, JULY 24-27, 1979
   PUBLISHED BY IERE, LONDON, ENGLAND - IERE CONFERENCE PRODEEDING #43

3. D. M. TAUB
   "PROCESSING THE 'READ' SIGNAL IN HIGH-DENSITY MAGNETIC RECORDING"
   IBM U.K. LABORATORIES INTERNAL FILE MEMORANDUM, APRIL 1976

$v - k v''$



FROM MACINTOSH

Fig.2 Pulse-slimming by addition of derivatives.

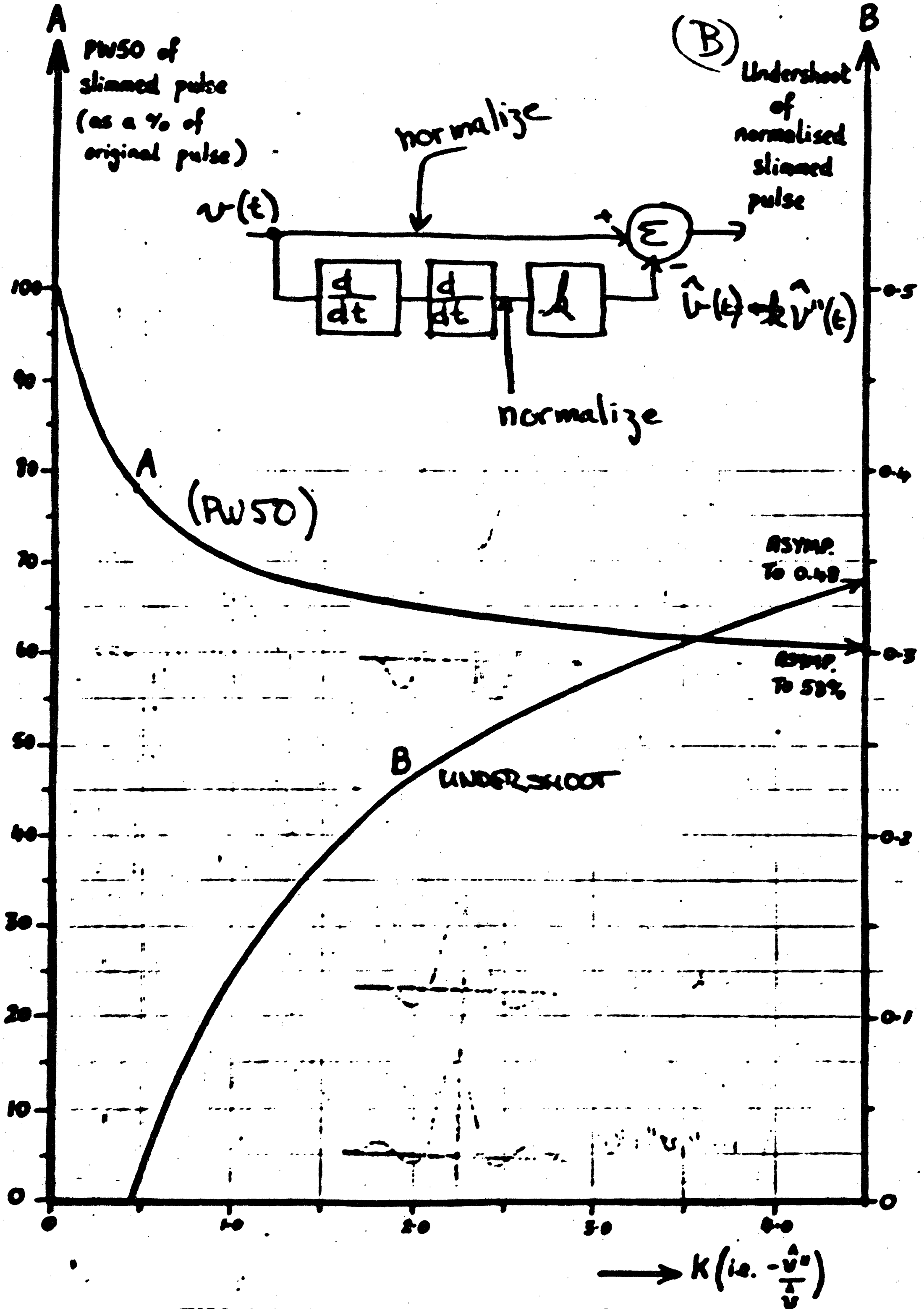


Fig.3 PW50 & Undershoot of slimmed pulse.

FROM MACINTOSH

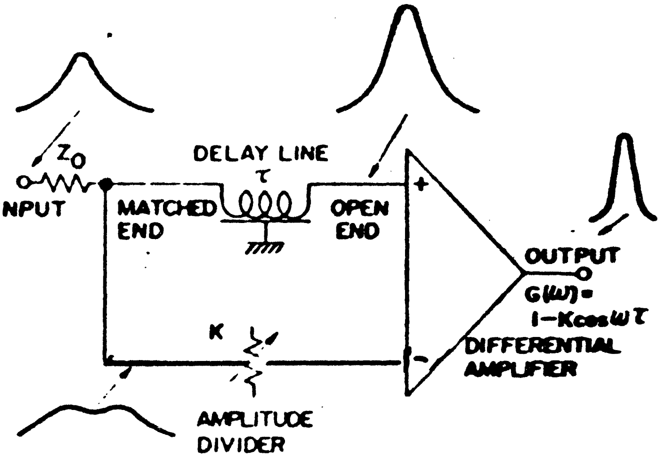


Fig.1 Waveform and blockdiagram.

FROM KAMEYAMA

[illegible]

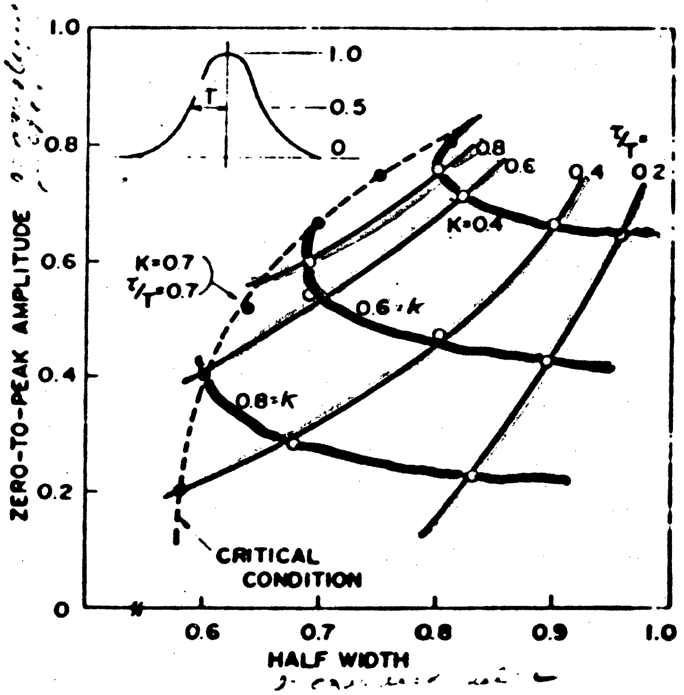
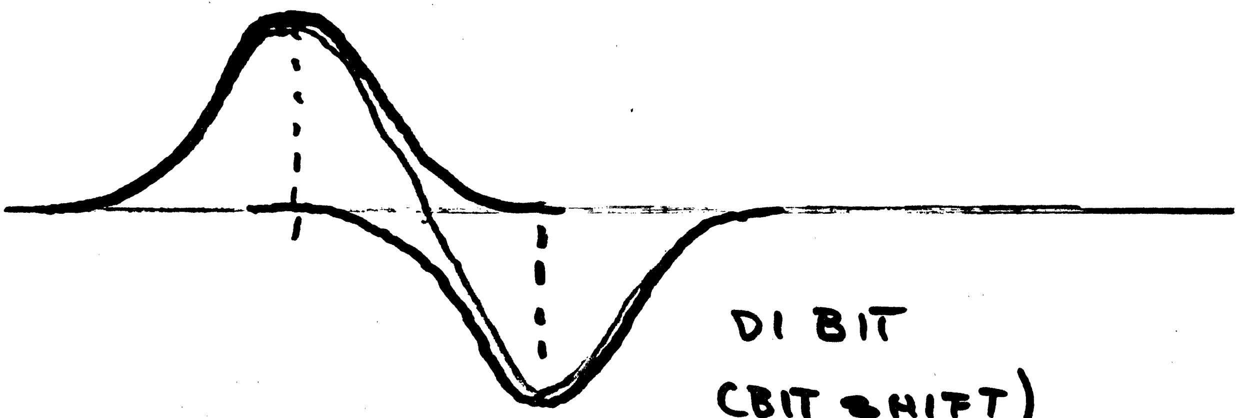
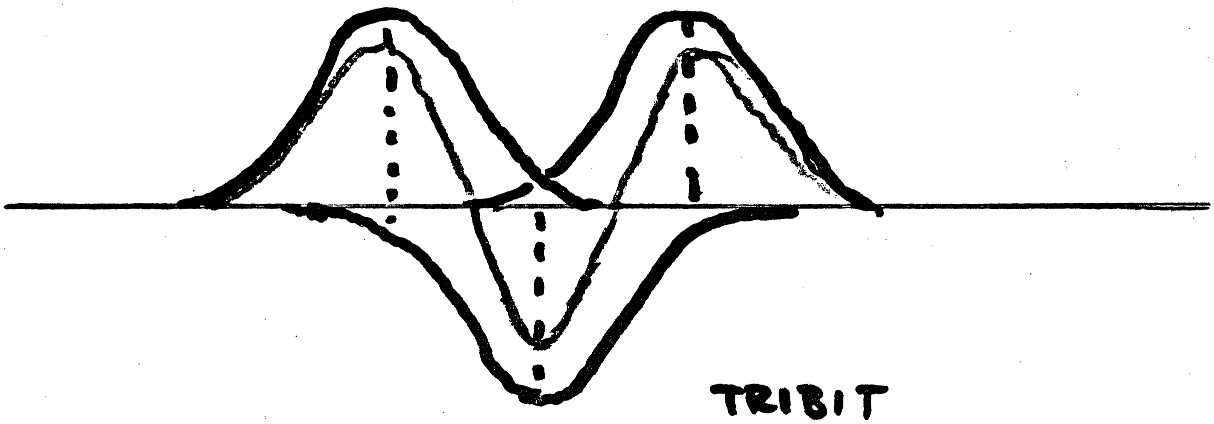


Fig.2 Resulting half width and zero-to-peak amplitude for various parameters, where half width and zero-to-peak amplitude without compensation are normalized to unity.

FROM KAMEYAMA



DI BIT
(BIT SHIFT)



TRIBIT
(AMPLITUDE LOSS
AND BIT SHIFT)

12/22/80 DPT.

unequalized system.

amplitude of middle bit of 3 "1"s pattern

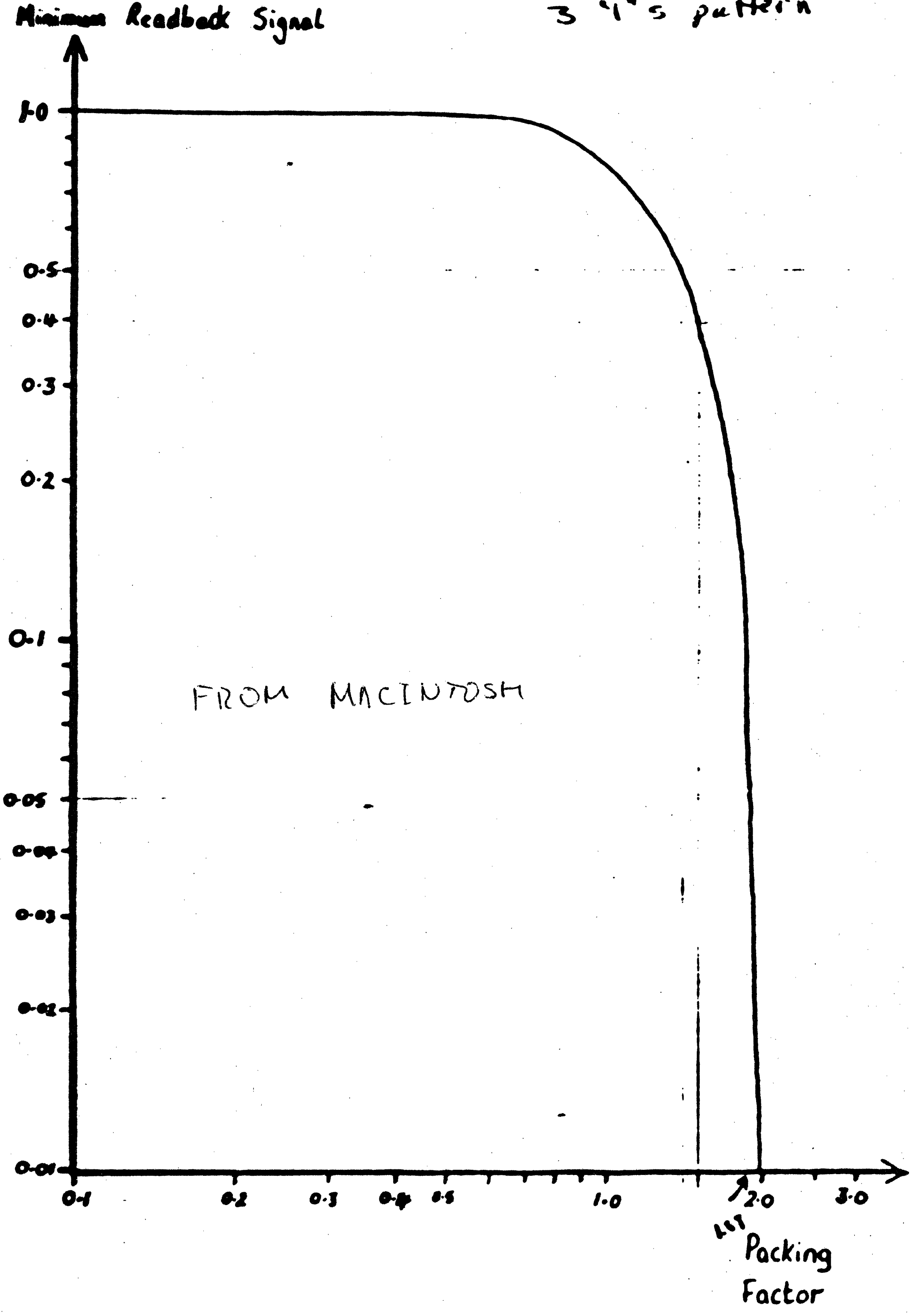


Fig.9 Minimum readback signal vs Packing Factor. for NRZ1 'three-ones'.

1.55 for 40% clips!

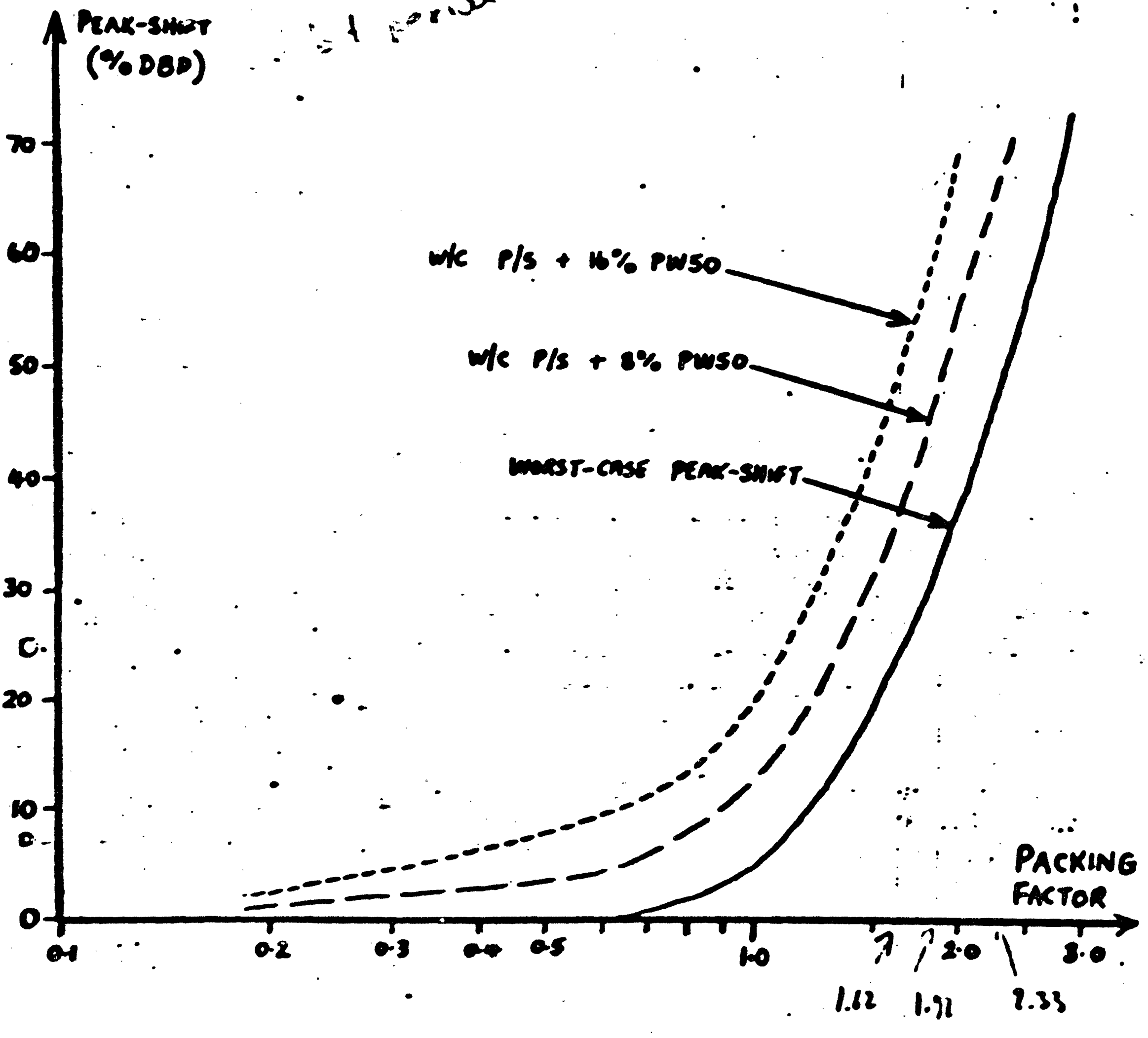
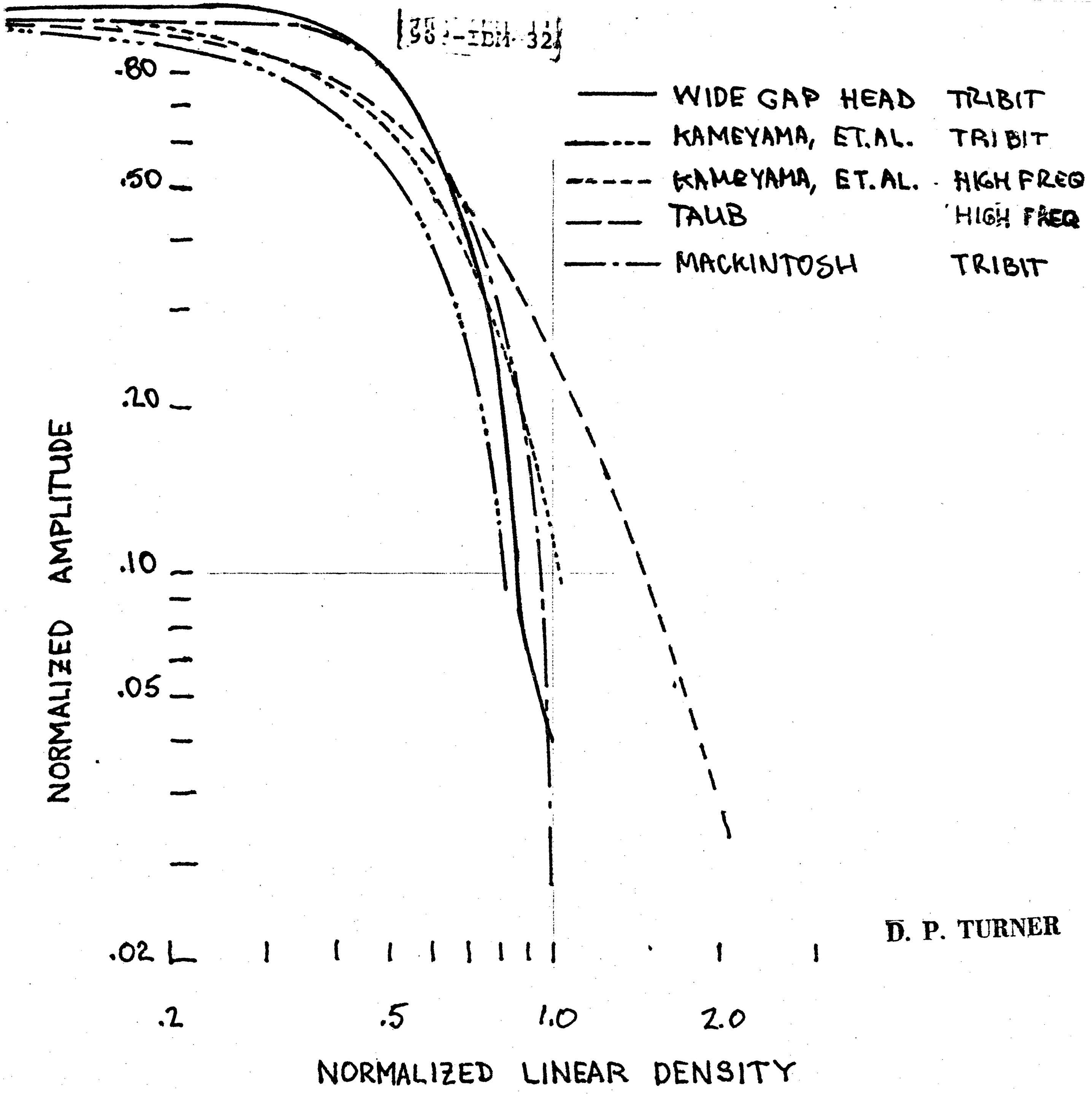


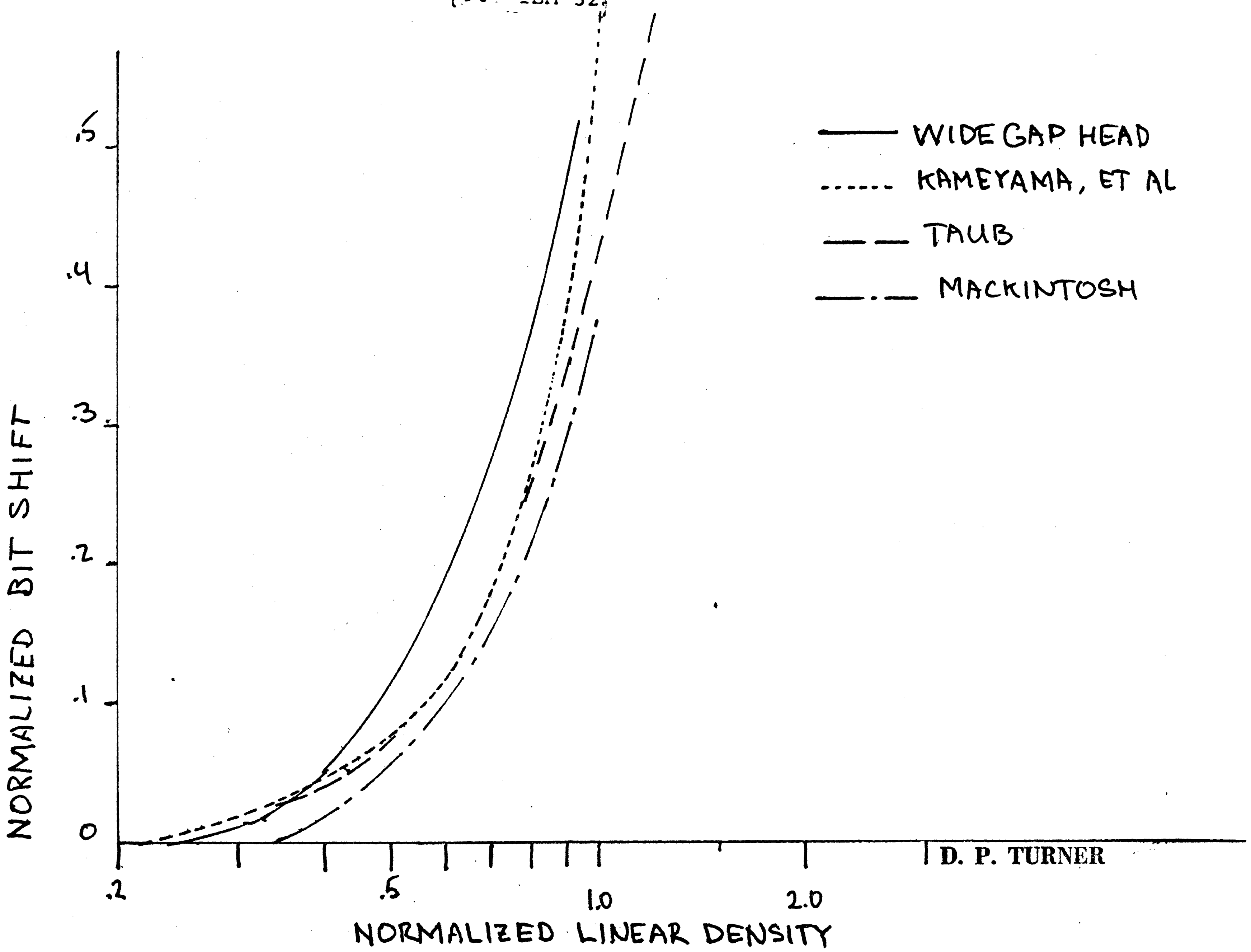
Fig. 10 Worst-case peak-shift for WR21.

UNEQUALIZED

FROM MACINTOSH


WIDE GAP HEAD TRIBIT
KAMEYAMA, ET.AL. TRIBIT
KAMEYAMA, ET.AL. HIGH FREQ
TAUB HIGH FREQ
MACKINTOSH TRIBIT
.80
.50
.20
.10
.05
.02
.2
.5
1.0
2.0
NORMALIZED AMPLITUDE
NORMALIZED LINEAR DENSITY

D. P. TURNER


WIDE GAP HEAD
KAMEYAMA, ET AL
TAUB
MACKINTOSH
NORMALIZED BIT SHIFT
.5
.4
.3
.2
.1
0
.2
.5
1.0
2.0
NORMALIZED LINEAR DENSITY

D. P. TURNER

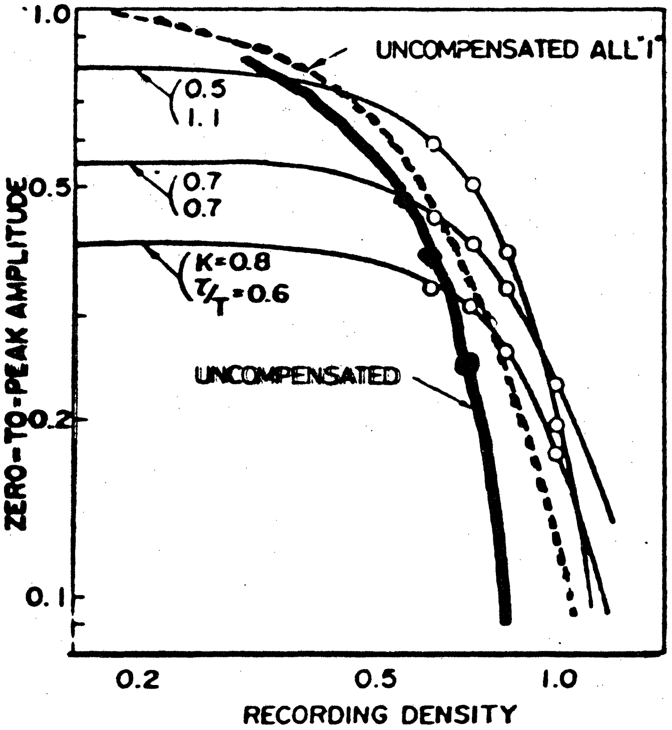
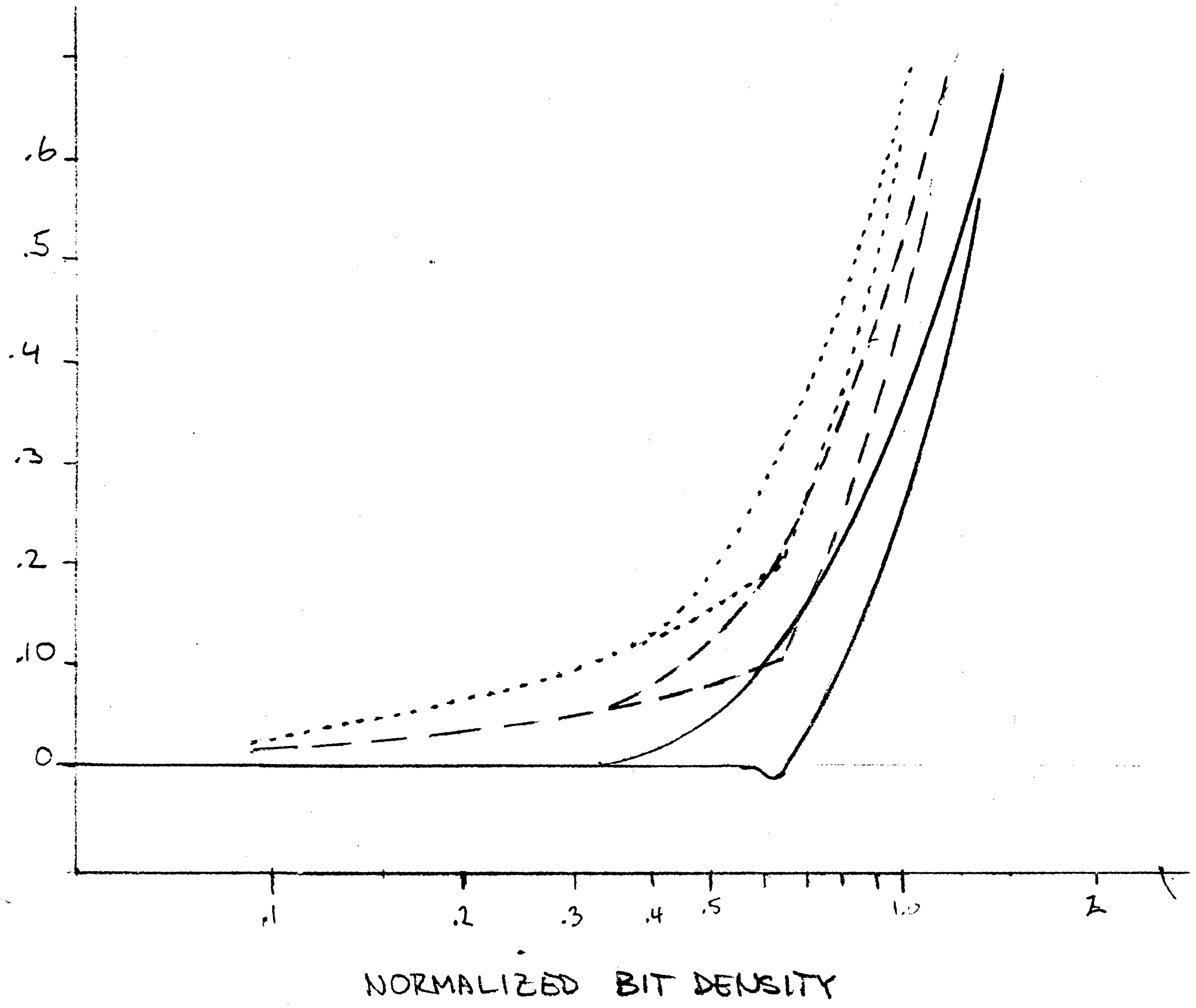


Fig.4 Zero-to-peak amplitude vs. recording density for the worst pattern cases (isolated three-bit in NRZ-I).

KAMEYAMA

[5B.-IDM.32]



MACINTOSH

[56 - IDM - 32]

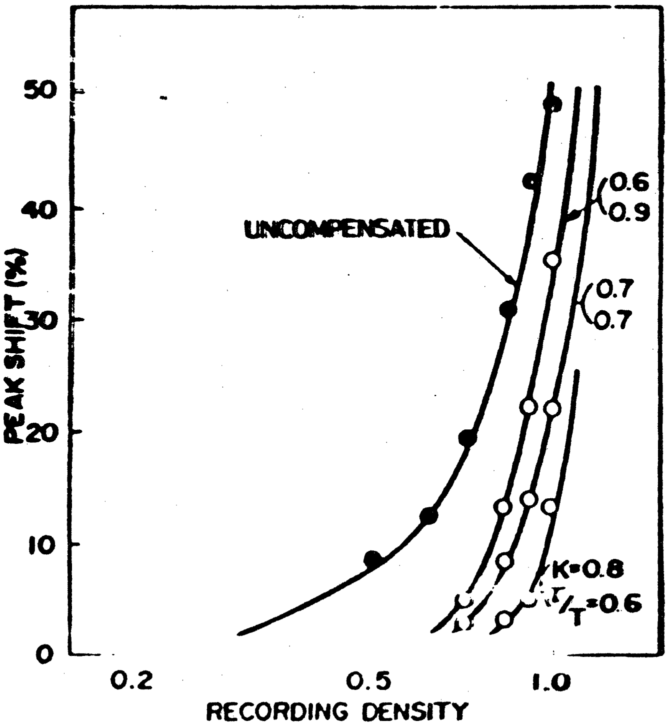


Fig.3 Peak shift vs. recording density for the worst pattern cases (isolated two-bit in NRZ-I).

KĀMEYAMA

[illegible]-IDA-32

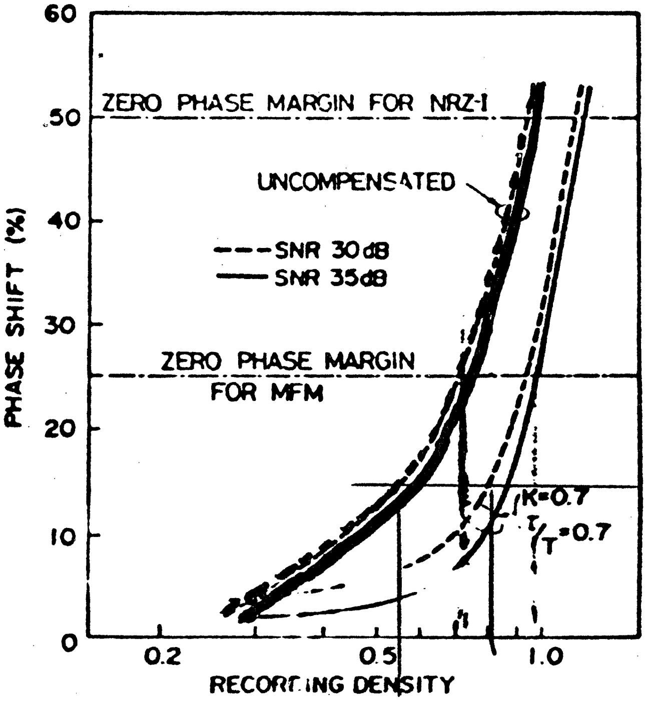


Fig.6 Phase margin vs. recording density.

KAMEYAMA

[SS-IECM-32]

~ 50% improvement.

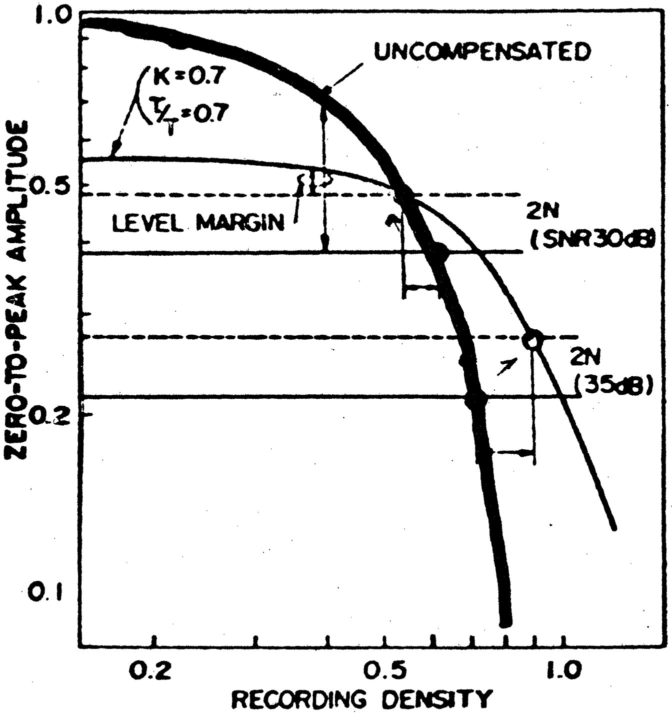
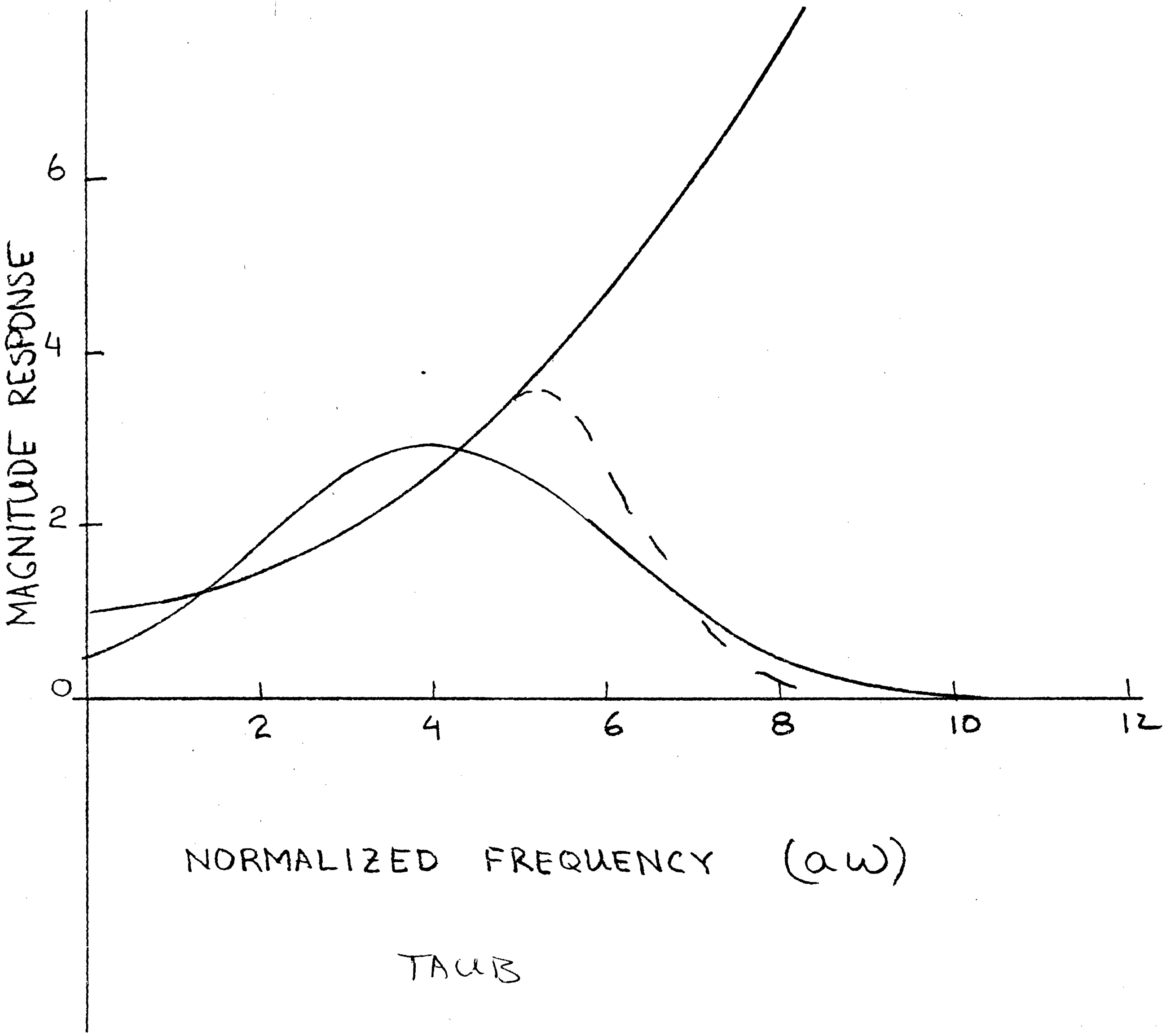


Fig.7 Amplitude margin vs. recording density.

KAMEYAMA

[56-IDM-32]


MAGNITUDE RESPONSE
0
2
4
6
2
4
6
8
10
12
NORMALIZED FREQUENCY (aω)

TAUB

SUMMARY

The objective of this talk is to demonstrate how one can achieve write equalization by shaping the write current transition in the saturate type recording.

Before presenting a novel technique for the write equalization, we will first characterize an 18-track double-gap head manufactured by Nortronics. Then, we will proceed to demonstrate the effectiveness of the proposed technique in terms of the pulse slimming and the peak shift improvement using a test set-up with this head. It is mentioned that this Nortronics head has been highly publicized as a major innovation in the recording field.

For a given test set-up, the maximum flux density one can achieve with this head is obtained by taking into account the following:

(1) The signal-to-noise ratio is 25dB at minimum.

(2) The peak shift with a worst case pattern is 50% at maximum.

(3) The amplitude variation from $3f_1$ to f_1 with the same pattern in (2) is 4:1 at maximum.

The results show the maximum flux density is 17.2kfci with the test set-up previously discussed.

The effectiveness of write equalization depends on dominance of linearity over nonlinearity associated with the write process. For a given recording channel where the above is true, the write equalization provides SNR enhancement and economical hardware implementation.

A general form of the write equalizer functions to be presented can be respresented by a transversal filter structure. The first-order function has been reported in the literature to be very effective as a read equalizer and also as a write equalizer in AC bias recording. If the first-order function is realized as a write equalizer in saturate type recording by shaping the write current transition, the major transition is modified such that two minor transitions are added before and after the transition, respectively.

In order to provide some perception on how to select an optimum minor transition, the plot is obtained from APL simulation, where the effects of the minor transition on the Lorentzian pulse are shown.

The transfer function for the second-order case is plotted and a hardware realization of this function in a transversal filter configuration is given. This realization can be seen as an extension of the first-order realization.

The write equalizers discussed so far are compared in terms of the isolated pulse response and the pattern response. It is shown in this comparison that the pulse was slimmed by the factor of two with the second-order equalizer. The performance of the second-order equalizer is also shown to be most satisfactory in terms of peak shift and maintaining the symmetry property the input pattern possesses. The peak shifts associated with the three bits in the pattern are improved from 53, 53 and 20% to 21, 14 and-21%, respectively.

Write Equalization by Current Shaping

Gene Sonu
GPD-Tucson
Jan., 1981

Topics

- Test Set-Up
- Write Current Shaping
- Hardware
- Effectiveness
  - Pulse Slimming (High Freq Boost)
  - Peak Shift Improvement (Pattern Response)

Characterization of A 18-Track Double-Gap Permalloy Head (Nortronics)

Test Set-Up

✓ B-H Drive

✓ Tape Tension 9.5 oz

✓ Tape Speed 1 m/sec

✓ γ-Fe_2O_3 (low Energy Tape)

✓ Saturate Type Recording

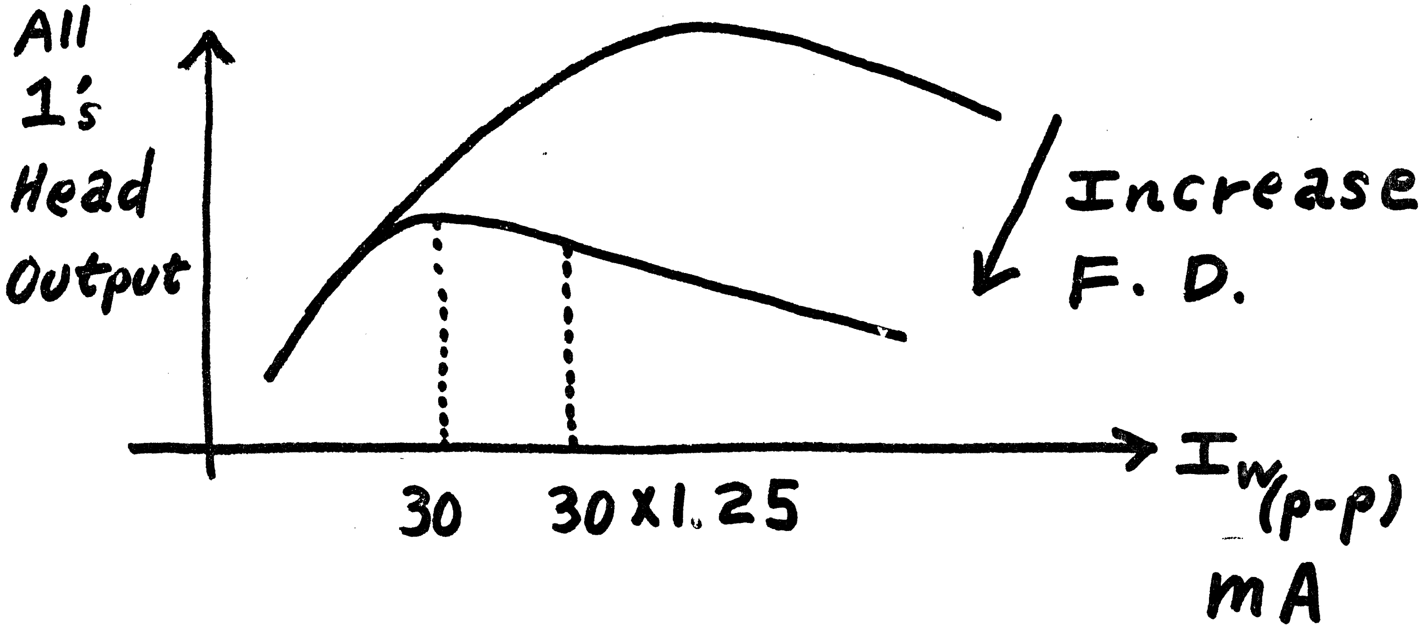
Flux Density

✓ SNR > 25 dB

✓ Peak Shift ≃ 50%

✓ Dynamic Range ≃ 4.0

SNR



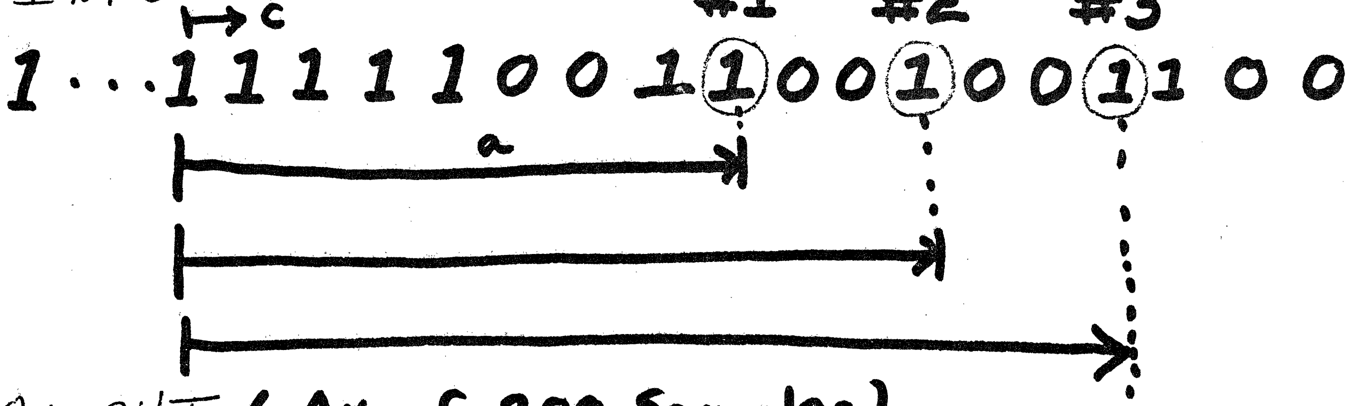
>25 dB

$I_{W_{p-p}}$ (operating) = 37.5 mA

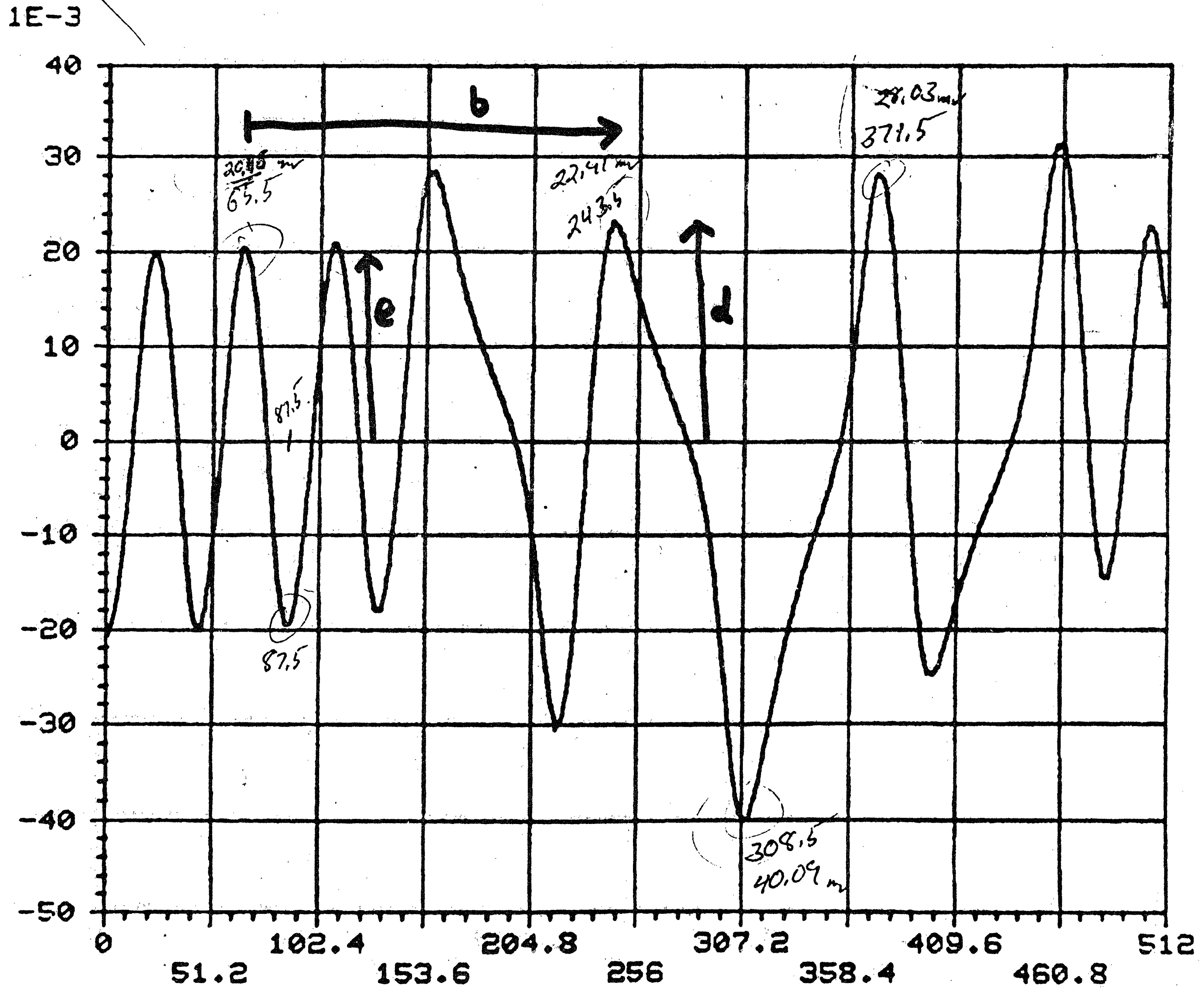
FD ≃ 18 K fci

A Worst Case P. S. Pattern

✓ INPUT



✓ OUTPUT (Av. of 200 Samples)

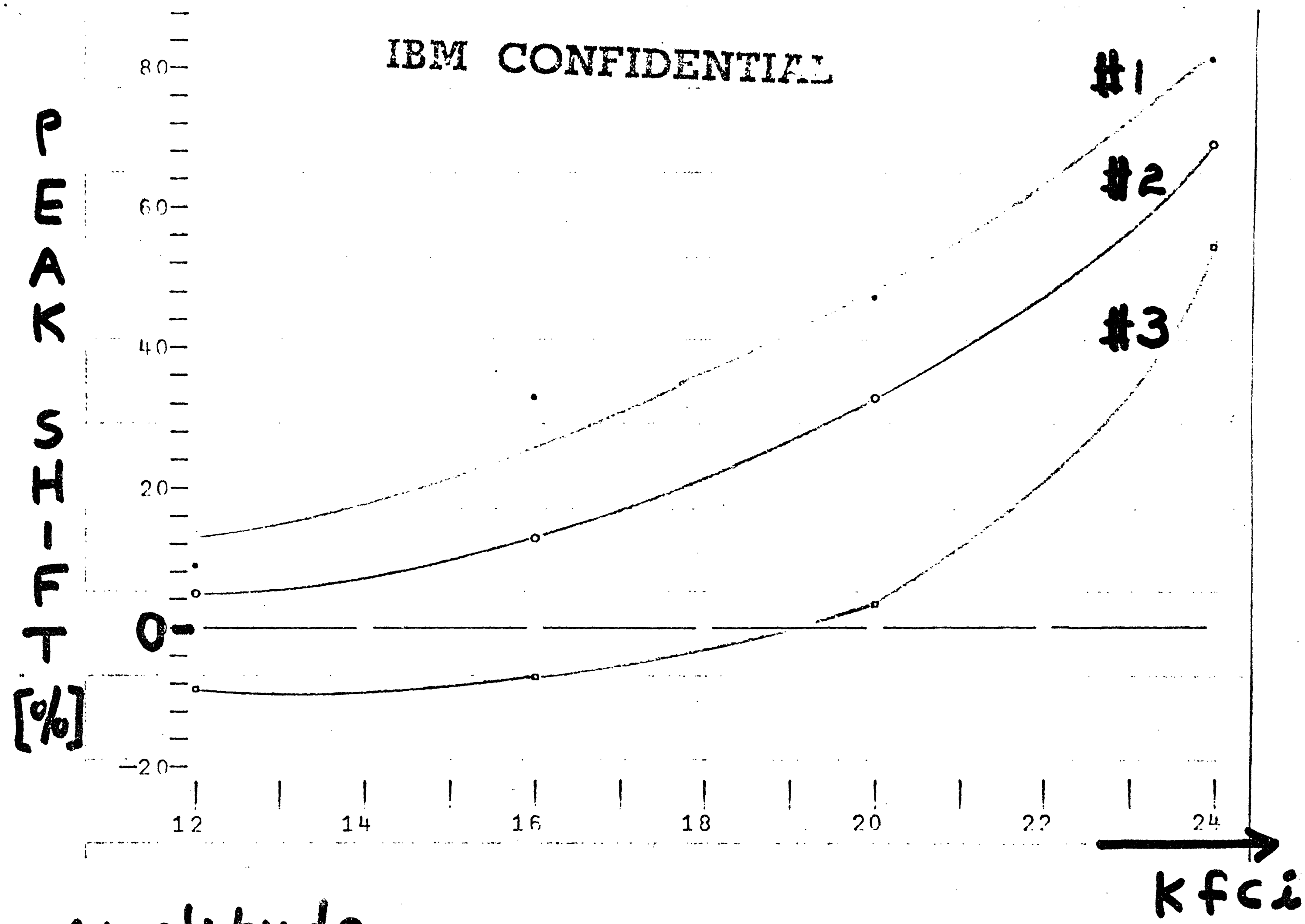


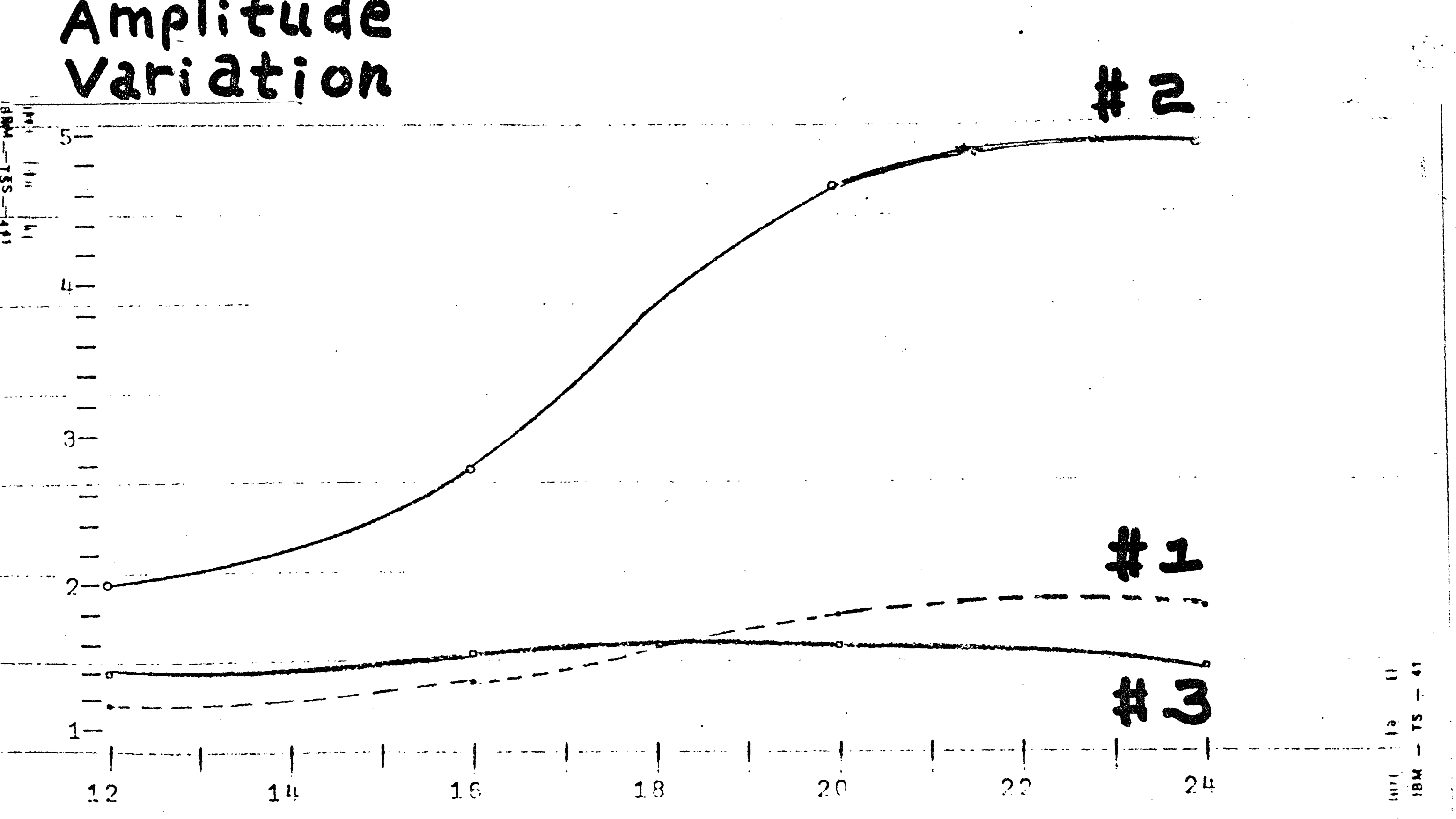
P.S for # 1 $\hat{=}$ $(b-a)/c$ %

AMP. VAR. for # 1 $\hat{=}$ d/e %

IBM CONFIDENTIAL

IBM CONFIDENTIAL





Why Write Eq?

- SNR
- Simple Hardware Implementatio

IF

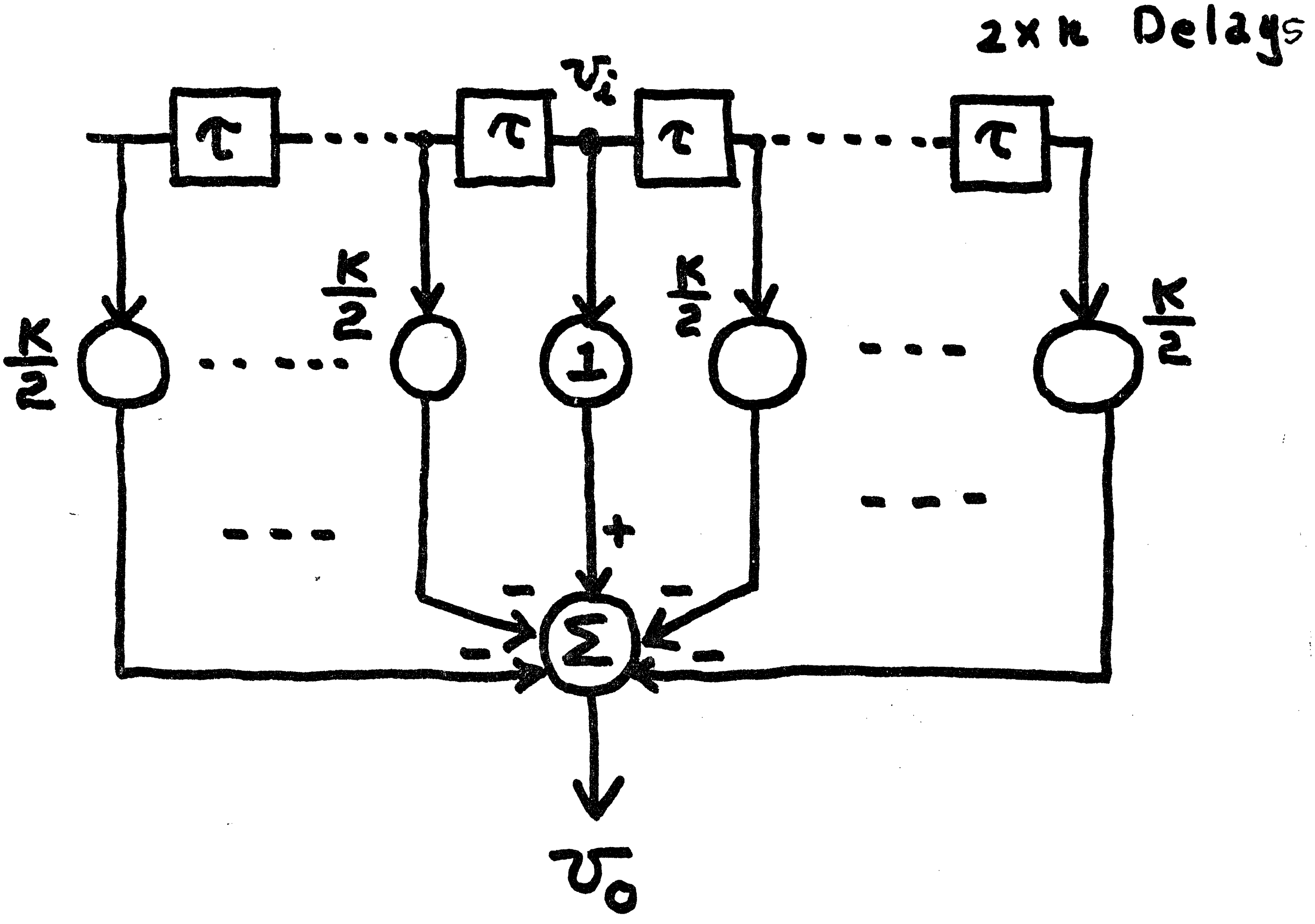
LINEARITY IS GUARANTEED

Applications:

NOT FOR VERY HIGH DENSITY RECORDING UNLESS LINEARITY IS MAINTAINED

A Transversal Filter Function

$$H(\omega) = 1 - K \sum_{k=1}^{n} \cos k\omega\tau$$



History

$\forall \; n = 1,$

$$H(\omega) = 1 - K \cos \omega \tau$$

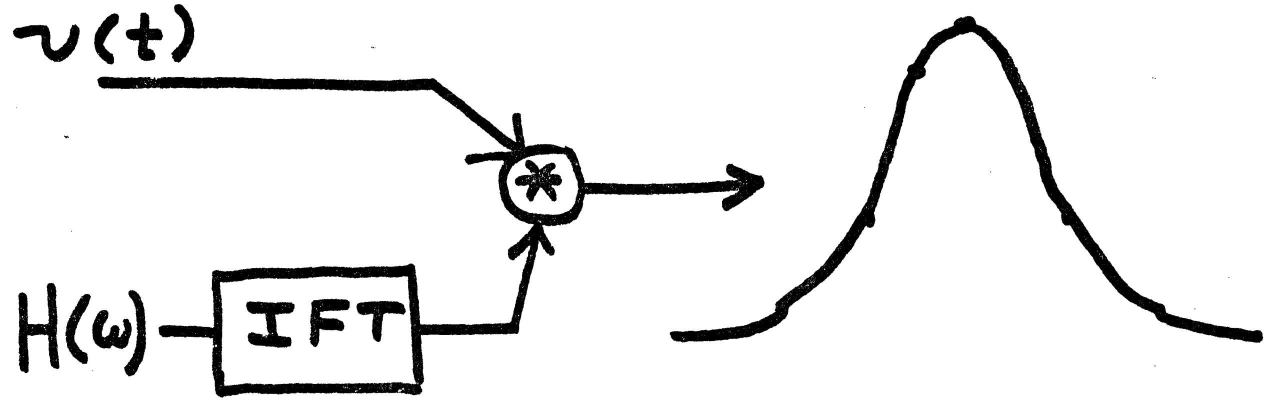
- o Read Eq : ✓ Pulse Slimming (2:1)
  ✓ SNR DEGRADATION

- o WRITE EQ : Jacoby 1979
  AC Bias
  3PM
  ✓ SNR IMPROVE
  ✓ Double Linear Density

Determination of K and τ

Based on
Isolated Pulse Output

Lorentzian Pulse



$$v(t) = \left[1 + \left(\frac{2t}{PW_{50\%}}\right)^2\right]^{-1}$$

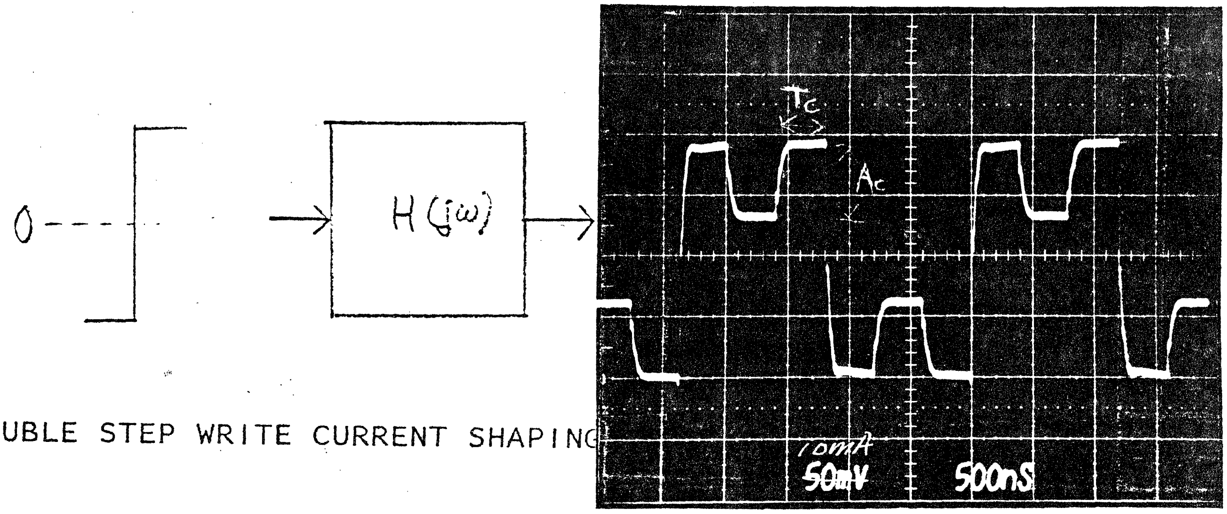


FIGURE 4. DOUBLE STEP WRITE CURRENT SHAPIN

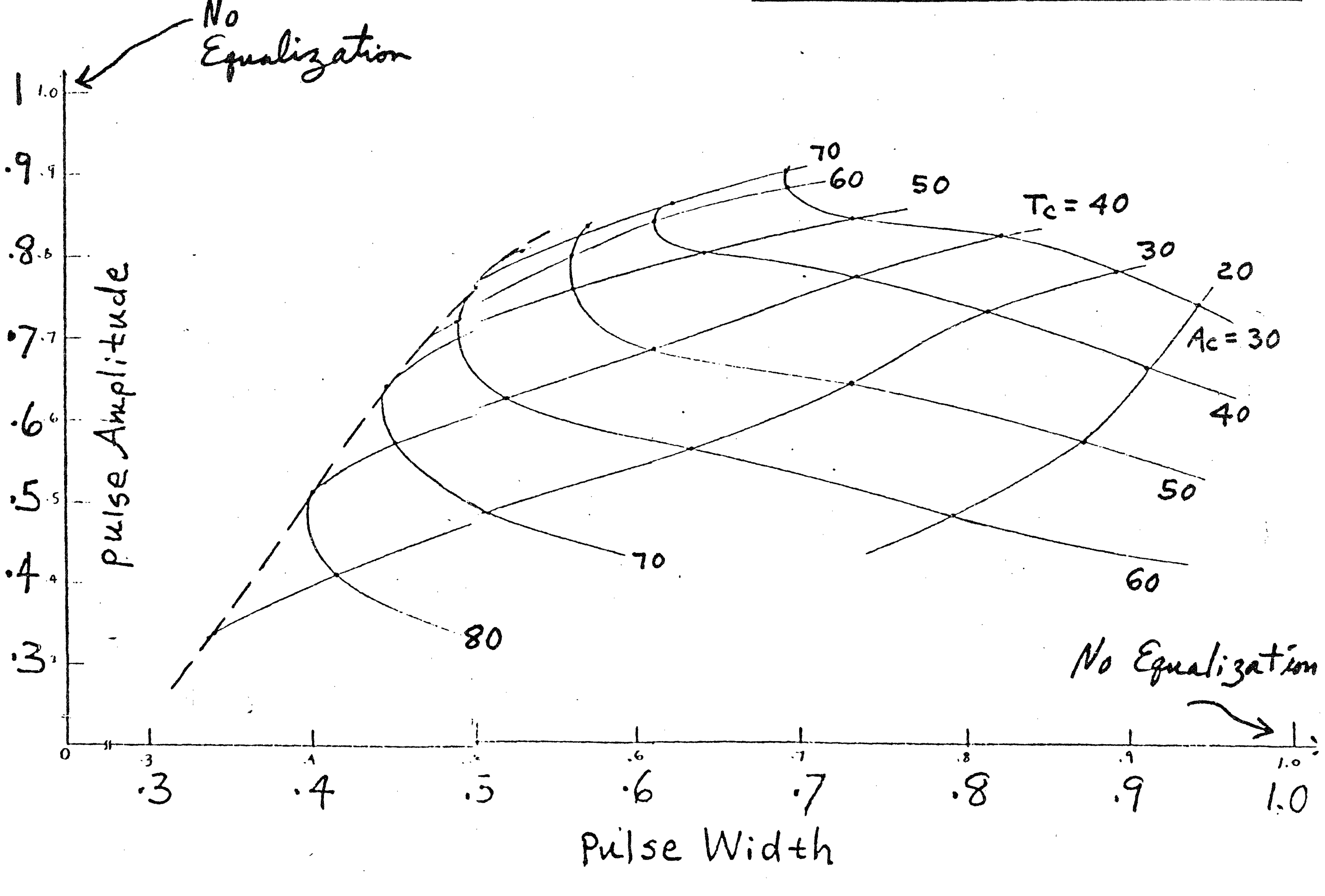


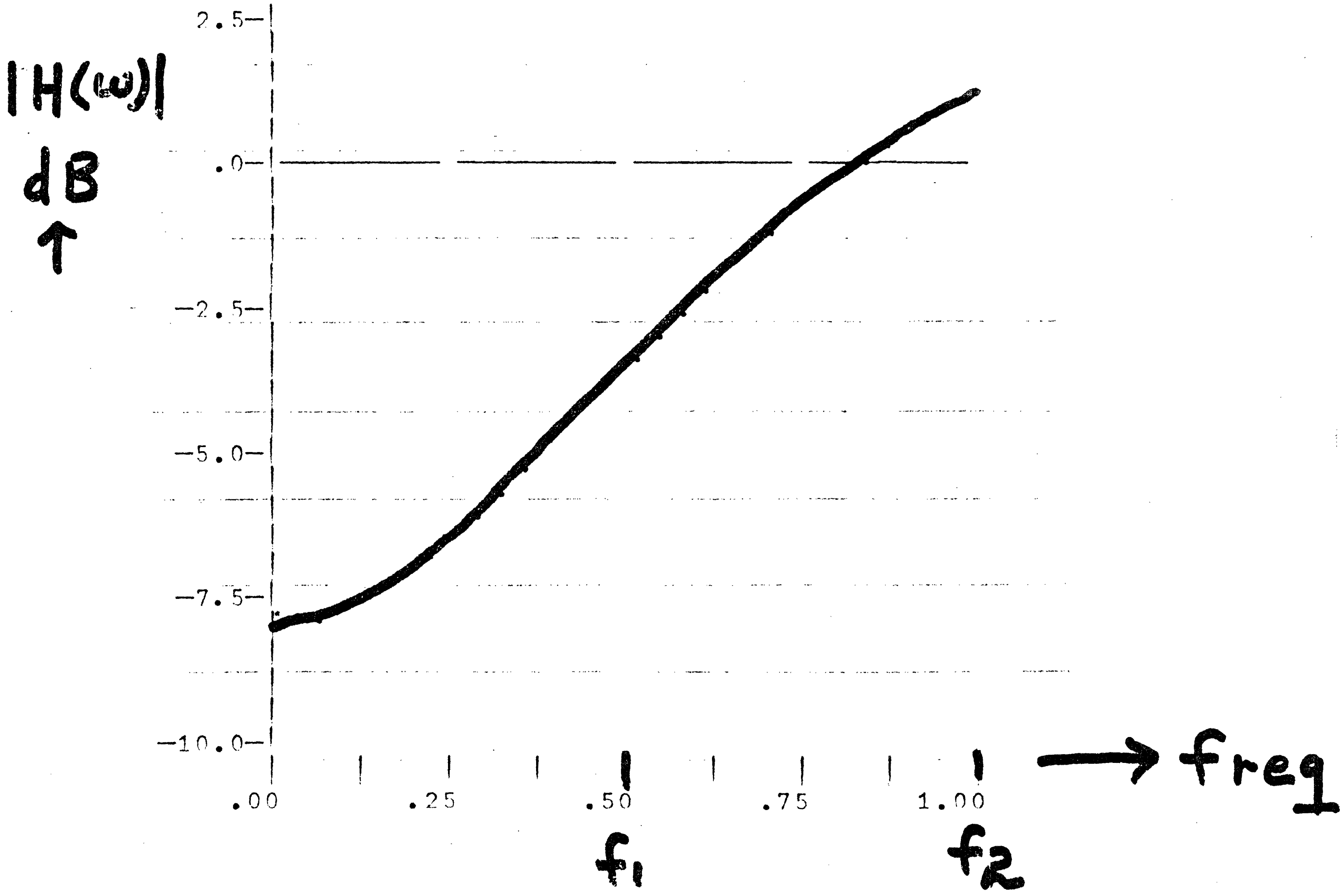
FIGURE 5. COMPUTER RESULTS RELATING PULSE WIDTH TO PULSE AMPLITUDE IN THE ISOLATED PULSE RESPONSE OF THE DOUBLE STEP WRITE EQUALIZER.

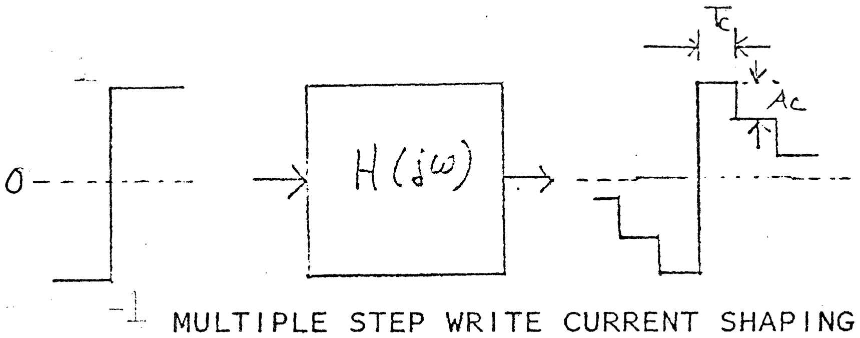
n = 2

$K = .3\ I_{w(o-p)}$

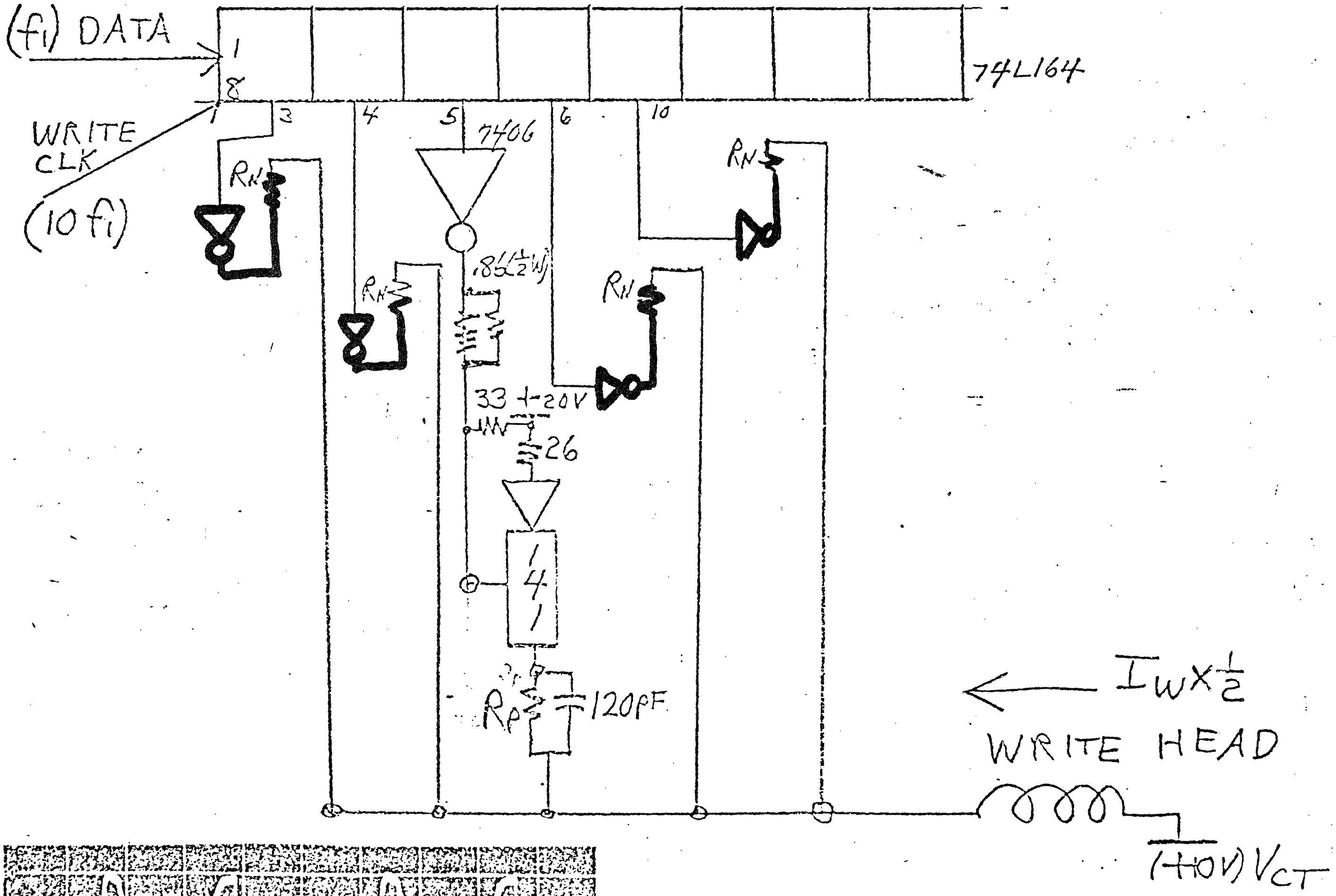
$\tau = .2T$ where $T = 1$

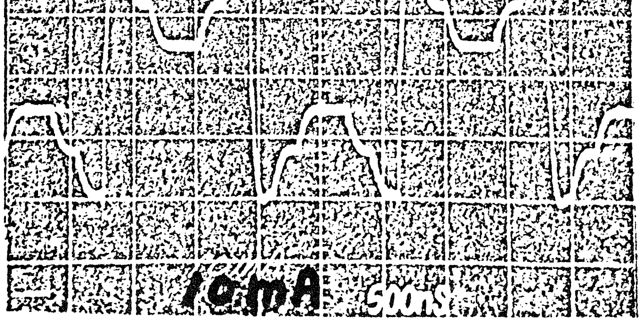
$$H(\omega) = 1 - K \sum_{k=1}^{2} \cos k\omega\tau$$





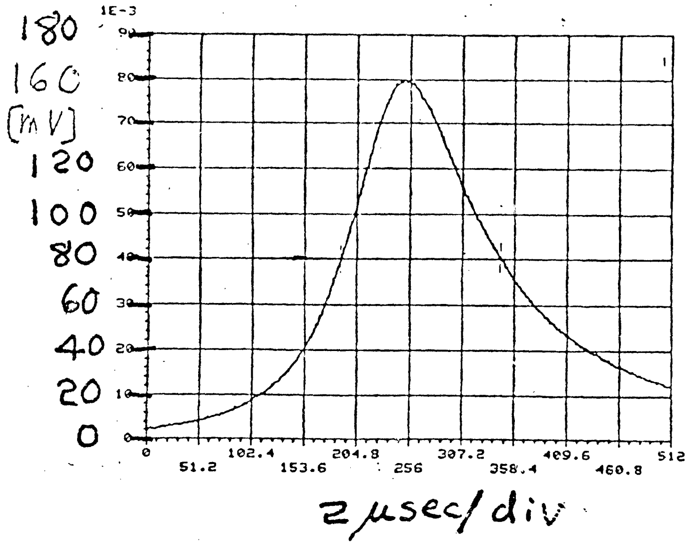
MULTIPLE STEP WRITE CURRENT SHAPING



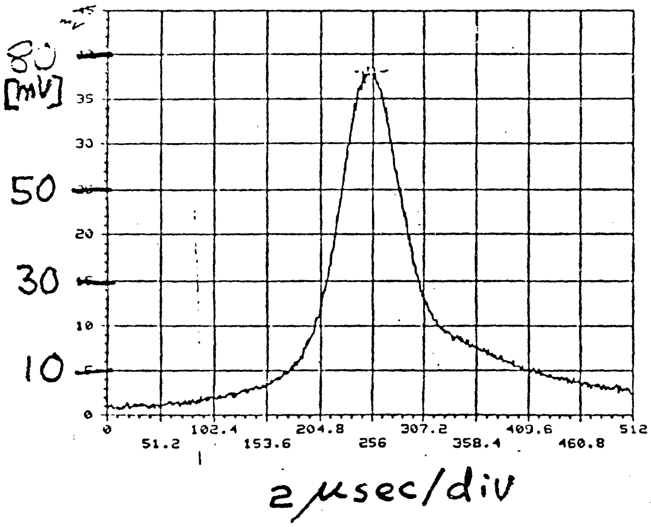


10mA/div
500nsec/div

200 Samples are Averaged



NO EQ



n = 2

Peak Shift [%]

		PS_1	PS_2	PS_3
No Eq		53	53	20
$n=1$	$\tau = .34T$ $K = .6\, I_{W\,(0\text{-}p)}$	33	33	-13
$n=2$	$\tau = .2T$ $K = .3\, I_{W\,(0\text{-}p)}$	21	14	-21

IBM CONFIDENTIAL

NOISELESS SLIDING BLOCK CODES

FOR

FINITE STATE CHANNELS

BY R. ADLER AND M. HASSNER

[RESEARCH, YORKTOWN]

ABSTRACT

WE DESCRIBE AN ALGORITHM FOR GENERATING NONLINEAR CONVOLUTIONAL MAPS, TERMED AS SLIDING BLOCK CODES, THAT MAP UNCONSTRAINED (OR CONSTRAINED) DATA INTO FINITE STATE CHANNELS. THE CHANNELS CONSIDERED ARE BOTH OF FINITE AND INFINITE MEMORY TYPE. FOR THE LATTER THE ALGORITHM PRODUCES SLIDING BLOCK CODES THAT HAVE (CHANNEL) STATE INDEPENDENT DECODERS, THUS AVOIDING INFINITE (CATASTROPHIC) ERROR PROPAGATION. THE SLIDING BLOCK CODES TRANSFER INFORMATION AT A RATE ARBITRARILY CLOSE TO THE CHANNEL CAPACITY (WHICH IS THE MAXIMAL INFORMATION RATE TRANSMISSIBLE OVER THE CHANNEL).

THE MAIN PRACTICAL APPLICATION CONSIDERED IS MAGNETIC RECORDING. THE MAGNETIC MEDIUM LIMITATIONS GIVE RISE TO A VARIETY OF CONSTRAINTS ALL OF WHICH CAN BE REPRESENTED IN THE FORMAT OF FINITE STATE CHANN'LS. THE CONSTRAINTS CONSIDERED ARE (d, k) RUN-LENGTH-LIMITS DUE TO INTERSYMBOL INTERFERENCE AND CLOCKING REQUIREMENTS, SPECTRAL NULLS DUE TO A-C COUPLING OR THE NEED FOR PILOT SIGNALS THAT WILL NOT INTERFERE WITH THE DATA, EXCLUDED WORST CASE PEAK SHIFT BIT PATTERNS, ETC.

IN SOME OF THE APPLICATIONS THESE CONSTRAINTS HAVE TO BE IMPOSED SIMULTANEOUSLY AS WELL AS IN CONJUNCTION WITH OTHER PREPROCESSING TECHNIQUES SUCH AS PARTIAL RESPONSE (CORRELATIVE) CODING.

THE CONSTRAINTS, PROPERTIES OF THE SLIDING BLOCK CODE SOLUTIONS, AS WELL AS THE IBM DIVISIONS THAT SUGGESTED THE PROBLEMS WHICH THEY SOLVE ARE SUMMARIZED IN THE FOLLOWING TABLE.

d	k	SPECTRAL NULL	W.C. PEAK SHIFT PATTERN EXCLUDED	PARTIAL RESPONSE	CODING RATE	ERROR PROPAG. (BITS)	IBM DIVISION
1	6			NRZI	$2/3$	26	GPD, TUCSON
1	7			NRZI	$2/3$.	6	GPD, TUCSON
0	4	$\omega = 0$ $\omega = \pi/6$		NRZI	$4/6$	4	RES., SJ
0	5	$\omega = 0$		INTERLEAVED NRZI	$3/4$	3	GPD, TUCSON
0	2	$\omega = 0$		NRZI	$3/4$	3	GPD, TUCSON
2	7		$1 0^2 1 0^7 1$ $1 0^7 1 0^2 1$	NRZI	$1/2$	4	ISD, ROCHESTER

THE CONSTRAINTS ABOVE AS WELL AS THE CORRESPONDING SLIDING BLOCK CODE SOLUTIONS SHOULD BE VIEWED AS EXAMPLES THAT DEMONSTRATE THE POWER OF THE ALGORITHM PRESENTED. IN OUR VIEWPOINT THIS ALGORITHM WOULD BE BETTER EXPLOITED IF USED IN CONJUNCTION WITH A SYSTEMATIC ANALYSIS OF THE CHANNEL, EQUALIZATION AND DETECTION SCHEME. SUCH AN ANALYSIS SHOULD GIVE AS AN OUTPUT MATCHING SETS OF CONSTRAINTS TO OPTIMIZE OVERALL SYSTEM PERFORMANCE AND WOULD PROVIDE AN INPUT TO OUR ALGORITHM RESULTING IN OPTIMAL SLIDING

NOISELESS SLIDING BLOCK CODES FOR FINITE-STATE CHANNELS

ROY ADLER AND MARTIN HASSNER

[RESEARCH, YORKTOWN]

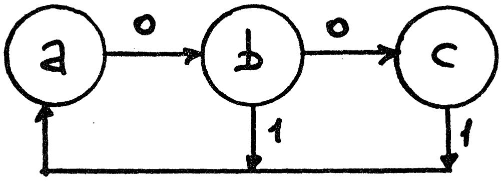
PRACTICAL MOTIVATION

FINITE STATE CHANNELS MODEL PHYSICAL LIMITATIONS OF DIGITAL TRANSMISSION OR STORAGE MEDIA.

FINITE-MEMORY CHANNEL MODELS OF MAGNETIC RECORDING

RUN-LENGTH-LIMITATIONS DESCRIBED BY FINITE-MEMORY CHANNELS OF $\underline{(d,k)}$-$\underline{\text{TYPE}}$ MODEL CONSTRAINTS DUE TO INTERSYMBOL INTERFERENCE AND CLOCKING REQUIREMENTS

EXAMPLE :



$d=1$

$k=2$

FINITE MEMORY PROPERTY : CHANNEL STATES $\{a,b,c\}$ ARE UNIQUELY DETERMINED BY THE NUMBER OF 0'S THAT FOLLOW THE LAST OCCURRENCE OF A 1 (I.E. BY A FINITE NUMBER OF PAST CHANNEL SYMBOLS)

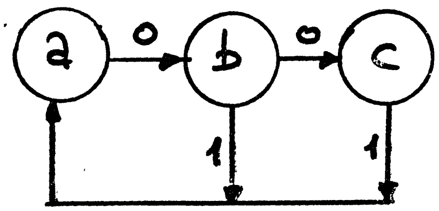
CHANNEL SYMBOL SEQUENCE : 1 0 0 1 0

CHANNEL STATE SEQUENCE : $a \rightarrow b \rightarrow c \rightarrow a \rightarrow b$

SHANNON'S NOISELESS CODING THEOREM

NECESSARY AND SUFFICIENT CONDITION FOR THE <u>EXISTENCE</u> OF AN INVERTIBLE (NOISELESS) CODE THAT MAPS DATA INTO A CONSTRAINED FORMAT.

MATHEMATICAL FORMULATION



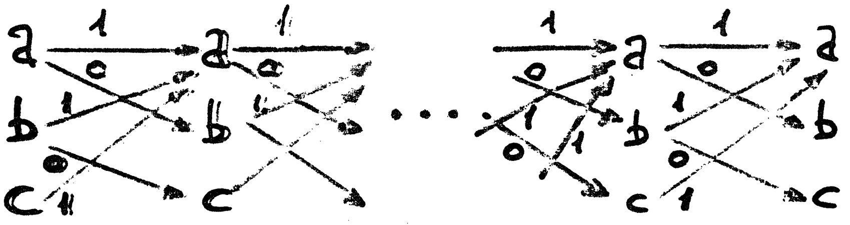
STATE TRANSITION MATRIX REPRESENTATION

$$T = \begin{array}{c|ccc} & a & b & c \\ \hline a & 0 & 1 & 0 \\ b & 1 & 0 & 1 \\ c & 1 & 0 & 0 \end{array}$$

CHANNEL CAPACITY $C_T = \log_2 \lambda_T$ bps

λ_T IS THE MAXIMAL EIGENVALUE (OR SPECTRAL RADIUS) OF T OBTAINED AS A SOLUTION OF THE CHARACTERISTIC EQUATION $\det(T-\lambda I)=$

OPERATIONAL INTERPRETATION



TRELLIS DESCRIPTION OF T- ADMISSIBLE n-BLOCKS.

$$\theta_n(T) = \sum_{i,j} T^{(n)}(i,j) = \|T^n\| \quad , i,j=a,b,c$$

(IS CARDINALITY OF SET OF n-BLOCKS DESCRIBED BY TRELLIS)

SPECTRAL RADIUS THEOREM

$$\text{GROWTH RATE} = \lim_n \|T^n\|^{1/n} = \lambda_T$$

$$C_T = \log_2 \lambda_T = \lim_n n^{-1} \log \theta_n(T)$$

CHANNEL CAPACITY C_T IS THE MAXIMUM INFORMATION RATE THAT CAN BE TRANSMITTED OVER CHANNEL BY USING T- ADMISSIBLE BLOCKCODES OF ARBITRAR: BLOCKLENGTH.

(UNCONSTRAINED BINARY) DATA SOURCE

$$M = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

SOURCE ENTROPY $\quad H_M = \log_2 \lambda_M$ bps

(FOR THIS PARTICULAR CASE $\lambda_M = 2$)

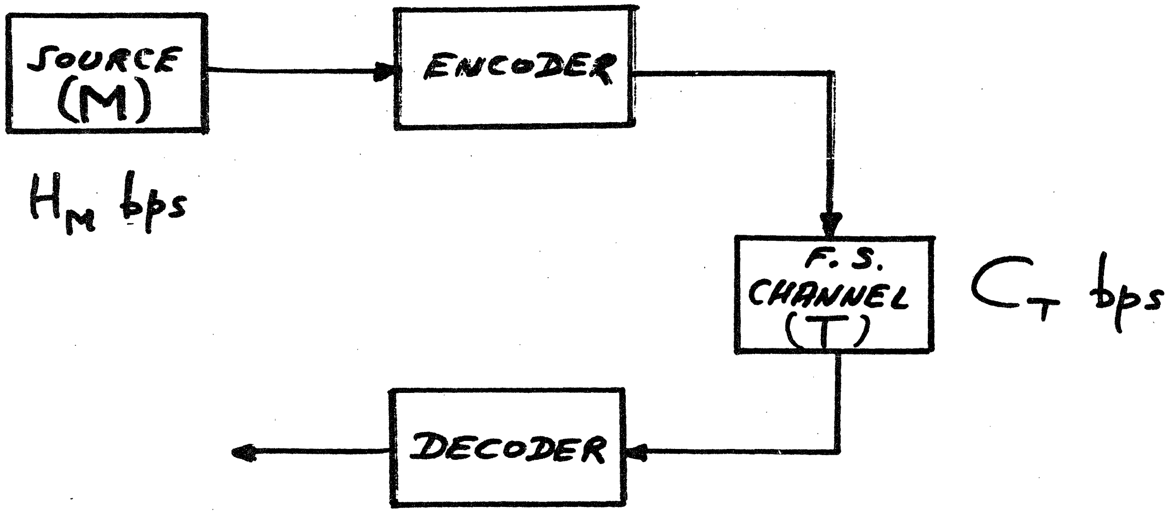
OPERATIONAL INTERPRETATION

$\lambda_M = \lim_n n^{-1} \| M^n \|^{1/n}$ is THE GROWTH RATE OF THE SOURCE WORD SETS

$H_M = \log_2 \lambda_M$ is THE <u>MINIMAL</u> <u>INFORMATION</u> <u>RATE</u> REQUIRED FOR ERRORLESS TRANSMISSION.

NOISELESS CODING THEOREM

A SOURCE M CAN BE INVERTIBLY CODED INTO A FINITE STATE CHANNEL T IF AND ONLY IF $H_M \leq C_T$



THE THEOREM IMPLIES THE EXISTENCE OF ENCODER / DECODER SCHEMES.

BLOCK CODE INTERPRETATION OF THEOREM

FOR LARGE n THERE EXISTS $\Theta_n(T) \approx 2^{nC_T}$ AND SINCE THE NUMBER OF n - SOURCE BLOCKS IS 2^{nH_M} (FOR OUR PARTICULAR SOURCE $H_M = 1$) THE CONDITION $H_M \leq C_T$* IMPLIES THE EXISTENCE OF INVERTIBLE n - BLOCK MAPS AS FOLLOWS

LIST OF SOURCE n-BLOCKS LIST OF CHANNEL n-BL

$$2^{nH_M}\left\{\begin{matrix} x_1^1 \cdots x_n^1 \\ x_1^2 \cdots x_n^2 \\ \vdots \end{matrix}\right. \longrightarrow \left.\begin{matrix} y_1^1 \cdots y_n^1 \\ y_1^2 \cdots y_n^2 \\ \vdots \end{matrix}\right\} 2^{nC_T}$$

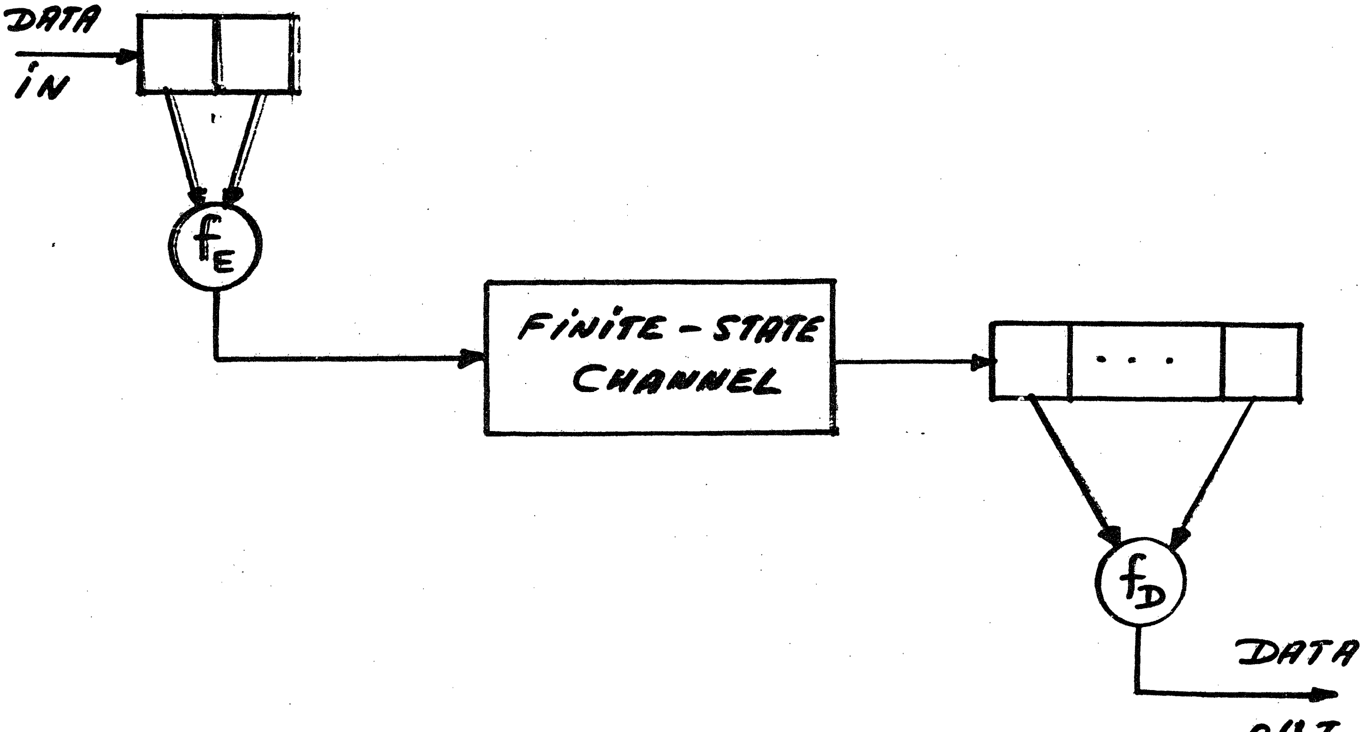
* IF $H_M > C_T$ WE CAN ENCODE INVERTIBLY AT THE EXPENSE OF LOWER INFORMATION RATES

$$m, n : 2^{nH_M} \leq 2^{mC_T} , \frac{n}{m} < 1$$

CRITIQUE OF SHANNON THEORY

1. THERE IS NO ALGORITHM FOR GENERATING BLOCK SCHEMES BUT SEARCH TECHNIQUES.
2. LARGE BLOCKLENGTHS ARE REQUIRED TO ACHIEVE INFO RATES CLOSE TO CHANNEL CAPACITY.

SLIDING BLOCK ALGORITHM



SCHEMATIC DIAGRAM

SEQUENTIAL CODING SCHEME THAT CODES NOISELESSLY (CONSTRAINED OR UNCONSTRAINED) DATA INTO AN ARBITRARY FINITE CONSTRAINT FORMAT AT AN INFORMATION RATE ARBITRARILY CLOSE TO THE CHANNEL CAPACITY.

THE ENCODER f_E AND DECODER f_D ARE DERIVED FROM AN ALGORITHM.

NONPROBABILISTIC ISOMORPHISM THEOREM
(ADLER, GOODWYN, MARCUS AND WEISS)

A SEQUENCE SPACE $Y(M_1)$ THAT CONSISTS OF ALL M_1-ADMISSIBLE SEQUENCES, WHERE M_1 IS AN (IRREDUCIBLE) 0-1 MATRIX, IS <u>ISOMORPHIC</u> TO A SEQUENCE SPACE $Y(M_2)$ IF AND ONLY IF $\boxed{\lambda_{M_1} = \lambda_{M_2}}$.

(THE 0-1 MATRICES M_1 AND M_2 DESCRIBE FINITE EXCLUDED BLOCK LISTS)

$$\begin{tikzcd}
 & Y(M_3) \arrow[dl] \arrow[dr] & \\
Y(M_2) & & Y(M_1)
\end{tikzcd}$$

COUPLING

$$\lambda_{M_3} = \lambda_{M_2} = \lambda_{M_1}$$

THEOREM ILLUSTRATION

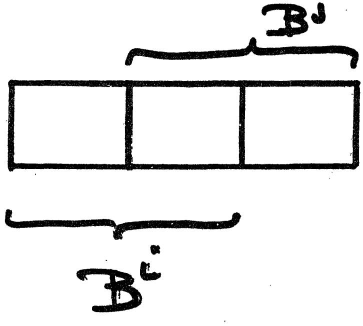
ENCODER $\quad f_E = \pi_2 \circ \pi_1^{-1}$

DECODER $\quad f_D = \pi_1 \circ \pi_2^{-1}$

FINITE EXCLUDED BLOCK LIST DESCRIPTION

EXAMPLE: $\mathcal{B}_3 = \{101\}$

$$\mathcal{B} = b_0 b_1 b_2$$



$$i, j \in \{a_1 = 00,\ a_2 = 01,\ a_3 = 10,\ a_4 = 11\}$$

0-1 MATRIX REPRESENTATION OF $\mathcal{B}_3$

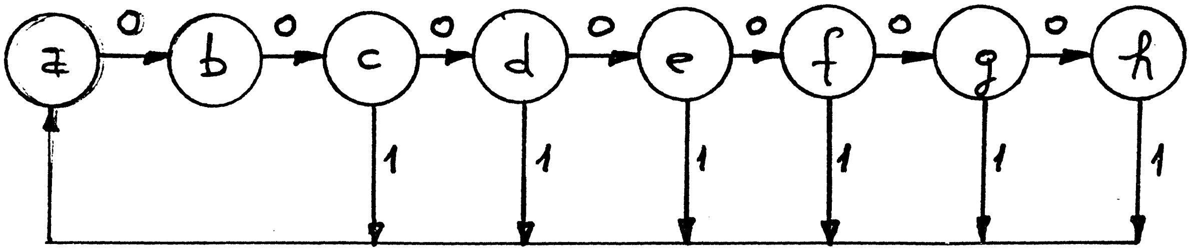
$$M = \begin{array}{c|cccc} & a_1 & a_2 & a_3 & a_4 \\ \hline a_1 & 1 & 1 & 0 & 0 \\ a_2 & 0 & 0 & 1 & 1 \\ a_3 & 1 & 0 & 0 & 0 \\ a_4 & 0 & 0 & 1 & 1 \end{array}$$

$Y(M)$ IS THE COLLECTION OF M-ADMISSIBLE SEQUENCES

OBSERVATION: A FINITE EXCLUDED BLOCK LIST DESCRIPTION IS EQUIVALENT TO A FINITE MEMORY CHANNEL.

APPLICATION TO MAGNETIC RECORDING – (d, k) CHANNELS
(RUN – LENGTH – LIMITED)

EXAMPLE :



FINITE STATE DESCRIPTION OF THE (2,7) – CHANNEL

STATE TRANSITION MATRIX

$$T = \begin{array}{c|cccccccc} & a & b & c & d & e & f & g & h \\ \hline a & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ d & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ e & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ f & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ g & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ h & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

CHANNEL CAPACITY COMPUTATION

THE CHARACTERISTIC EQUATION FOR A (d,k)-CHANNEL is $\lambda^{k+1} - \frac{\lambda^{k-d+1}-1}{\lambda-1}$ [TANG AND BAHL]

FOR THE $(2,7)$-CHANNEL $C_T = \log_2 \lambda_T$ WHERE λ_T IS THE LARGEST ROOT OF THE EQUATION

$$\lambda^8 - \frac{\lambda^6-1}{\lambda-1} = 0 \qquad \left(\sqrt{2} < \lambda_T \simeq 1{,}4314\right)$$

DATA ENTROPY

$$H_M = \log_2 \lambda_M = 1 \quad \text{SINCE} \quad \lambda_M = 2 \ , \quad M = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

SINCE $H_M > C_T$ WE HAVE TO LOSE INFORMATION RATE

$$m, n : \lambda_T^m \geq \lambda_M^n$$

$n = 1$

$m = 2$

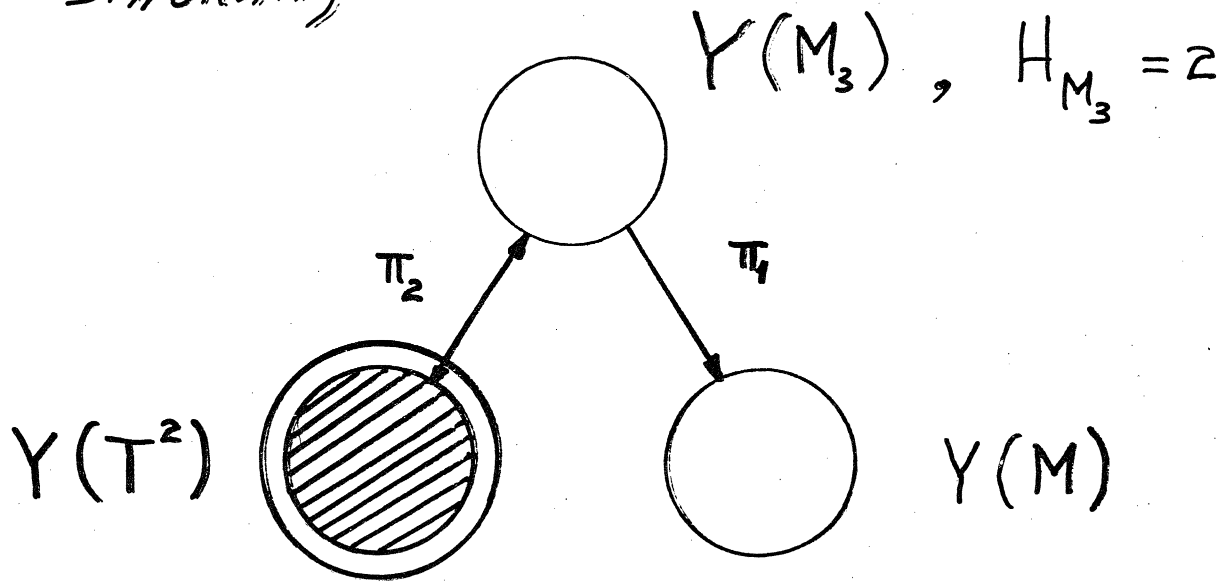
CODING RATE CHOSEN IS $1/2$

TWO-STEP STATE TRANSITION MATRIX DESCRIBES THE CHANNEL IN WHICH EACH STATE TRANSITION IS LABELLED BY 2 BITS.

$$T^2 = \begin{array}{c|cccccccc} & a & b & c & d & e & f & g & h \\ \hline a & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ c & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ d & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ e & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ f & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ g & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ h & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\lambda_{T^2} = \lambda_T^2 > 2 = \lambda_M^1$$

GENERALIZED ISOMORPHISM THEOREM (SCHEMATIC DIAGRAM)



π_1 IS ONTO 1-1 A.E.

π_2 IS ONTO 1-1

MAXIMUM EIGENVALUE- EIGENVECTOR INEQUALITY

$$\left(T^m \underline{r} \geq \lambda_M^n \underline{r} \right)$$

$$T^2 \underline{r} \geq 2 \underline{r}$$

THERE EXISTS AN <u>INTEGRAL</u> SOLUTION SINCE T^2 IS AN INTEGRAL MATRIX. $\underline{r}$ IS AN APPROX. PERRON EIGENVECTOR

INTEGRAL RIGHT APPROXIMATE EIGENVECTOR SOLUTION

$$\underline{r} = [2, 3, 4, 4, 3, 3, 1, 1]$$

SPLITTING ALGORITHM

THE CHANNEL STATES THAT LABEL THE ROWS AND COLUMNS OF T^2 ARE SPLIT IN A CONTROLLED MANNER AND T^2 IS TRANSFORMED INTO $\hat{T}^2$ SUCH THAT

$$\hat{T}^2 \underline{r} \geq 2 \underline{\hat{r}}$$

a. $\underline{\hat{r}} = [1, \dots, 1]^T$

b. $\dim(\hat{T}^2) = \sum r_i \; (=21)$

c. row sum $(\hat{T}^2) \geq 2$

d. column sum $(\hat{T}^2) =$ column sum (T^2)

ONE STEP OF ITERATIVE SPLITTING ALGORITHM (GENERAL)

a. CHOOSE STATE i^* SUCH THAT $\boxed{r_{i^*} \geq r_i}$ AND ITS SET OF SUCCESSORS $S(i^*)$ IS OF SIZE $\boxed{|S(i^*)| > \lambda_M^n}$. (THE EXISTENCE OF i^* IS GUARANTEED BY THE IRREDUCIBILITY OF T^m).

b. AMONG ALL SUBSETS $E \subset S(i^*)$ OF SIZE $|E| \leq \lambda_M^n$ THERE IS ONE (OR MANY) SUCH THAT $\boxed{\sum_{j \in E} r_j \equiv 0 \bmod \lambda_M^n}$

c. SPLIT AND ASSIGN NEW TRANSITIONS AS FOLLOWS

$$\begin{tikzcd}
 & i_1^* \arrow[r] & E \\
i^* \arrow[ur] \arrow[dr] & & \\
 & i_2^* \arrow[r] & E^c = S(i^*) - E
\end{tikzcd}$$

SUCCESSOR TABLE	r
a → c	$r_a = 2$
b → a, d	$r_b = 3$
c → a, b, e	$r_c = 4$ ←
d → a, b, f	$r_d = 4$
e → a, b, g	$r_e = 3$
f → a, b, h	$r_f = 3$
g → a, b	$r_g = 1$

ONE STEP OF ITERATIVE SPLITTING ALGORITHM (SPECIFIC EXAMPLE)

a. CHOOSE MAXIMAL COMPONENT $r_c = 4$

b.
$$\begin{tikzcd} & & E = \{a\} \quad , \; r_a = 2 \\ c \arrow[r] & \{a, b, e\} \arrow[ur] \arrow[dr] & \\ & & E^c = \{b, e\} \end{tikzcd}$$

c.
$$\begin{tikzcd} & c_1 \arrow[r] & a \\ c \arrow[ur] \arrow[dr] & & \\ & c_2 \arrow[r] & b, e \end{tikzcd}$$

$$r_{c_1} = \frac{r_a}{2} = 1$$

$$r_{c_2} = \frac{r_b + r_e}{2} = 3$$

d. ALL OTHER TRANSITIONS ARE LEFT UNCHANGED.

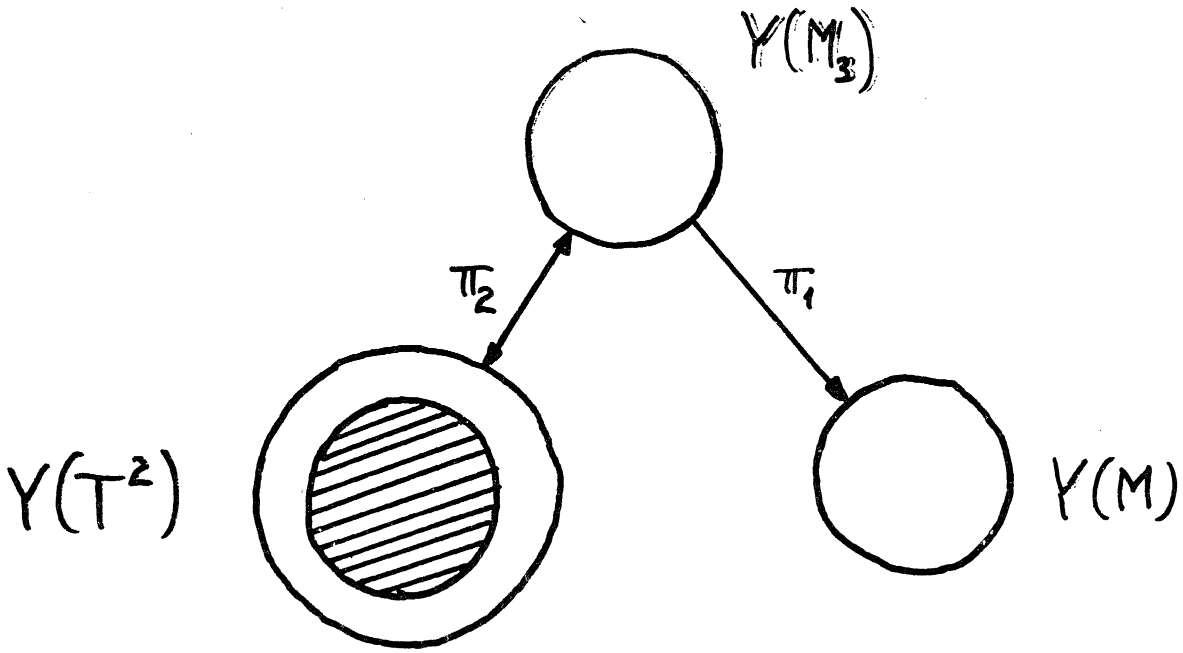
NEW SUCCESSOR TABLE		$\underline{r}'$
$a \longrightarrow$	c_1, c_2	$r_a = 2$
$b \longrightarrow$	a, d	$r_b = 3$
$c_1 \longrightarrow$	a	$r_{c_1} = 1$
$c_2 \longrightarrow$	b, e	$r_{c_2} = 3$
$d \longrightarrow$	a, b, f	$r_d = 4$
$e \longrightarrow$	a, b, g	$r_e = 3$
$f \longrightarrow$	a, b, h	$r_f = 3$
$g \longrightarrow$	a, b	$r_g = 1$
$h \longrightarrow$	b	$r_h = 1$

$$\begin{array}{c|cccccccc}
 & a & b & c & d & e & f & g & h \\
\hline
a & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
b & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
d & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
e & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
f & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
g & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
h & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
 & 6 & 6 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
\begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix}
\geqslant 2
\begin{bmatrix} 2 \\ 3 \\ 4 \\ 4 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{array}{c|ccccccccc}
 & a & b & c_1 & c_2 & d & e & f & g & h \\
\hline
a & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
b & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
c_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
d & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
e & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
f & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
g & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
h & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
 & 6 & 6 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
\begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 4 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix}
\geqslant 2
\begin{bmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 4 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix}$$

PROPERTIES OF SPLITTING MAP

THE "OFFSPRING" CHANNEL STATES AND THE "PARENT" STATES HAVE IDENTICAL PREDECESSORS. (THE COLUMN SUMS STAY INVARIANT UNDER SPLITTING). THIS GUARANTEES THAT THE SPLITTING MAP BE INVERTIBLE RESULTING IN A FINITE SLIDING BLOCK DECODER.



FINAL OUTCOME

$a_1 \longrightarrow c_2 , c_3$

$a_2 \longrightarrow c_1 , c_4$

$b_1 \longrightarrow d_1 , d_2$

$b_2 \longrightarrow a_2 , a_1$

$b_3 \longrightarrow d_3 , d_4$

$c_1 \longrightarrow a_2 , a_1$

$c_2 \longrightarrow b_1 , b_3$

$c_3 \longrightarrow e_2 , e_3$

$c_4 \longrightarrow b_2 , e_1$

$d_1 \longrightarrow f_2 , f_3$

$d_2 \longrightarrow b_1 , b_3$

$d_3 \longrightarrow a_2 , a_1$

$d_4 \longrightarrow b_2 , f_1$

$e_1 \longrightarrow a_1 , a_2$

$e_2 \longrightarrow b_1 , b_3$

$e_3 \longrightarrow b_2 , g_1$

$f_1 \longrightarrow a_2 , a_1$

$f_2 \longrightarrow h_1 , b_2$

$f_3 \longrightarrow b_1 , b_3$

$g_1 \longrightarrow a_2 , b_1 , (b_2 , b_3 , a_1)$

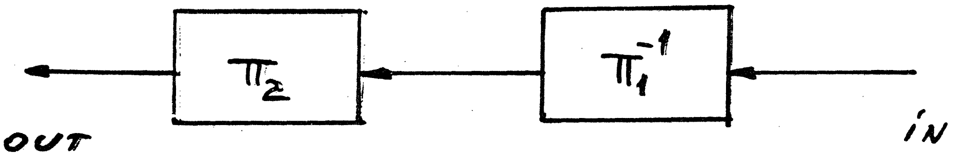
$h_1 \longrightarrow b_2 , b_3 , (b_1)$

SUCCESSOR TABLE REPRESENTATION OF A ROW SUM 2 MATRIX M_3 (EXCESS TRANSITIONS DELETED)

π_1 - ASSIGNMENT ("COLORING")

	0			1	
a_1	$\longrightarrow$	c_3	a_1	$\longrightarrow$	e_2
a_2	$\longrightarrow$	c_1	a_2	$\longrightarrow$	c_4
b_1	$\longrightarrow$	d_1	b_1	$\longrightarrow$	d_2
b_2	$\longrightarrow$	a_2	b_2	$\longrightarrow$	a_1
b_3	$\longrightarrow$	d_3	b_3	$\longrightarrow$	d_4
c_1	$\longrightarrow$	a_2	c_1	$\longrightarrow$	a_1
c_2	$\longrightarrow$	b_3	c_2	$\longrightarrow$	b_1
c_3	$\longrightarrow$	e_2	c_3	$\longrightarrow$	e_3
c_4	$\longrightarrow$	e_1	c_4	$\longrightarrow$	b_2
d_1	$\longrightarrow$	f_2	d_1	$\longrightarrow$	f_3
d_2	$\longrightarrow$	b_3	d_2	$\longrightarrow$	b_1
d_3	$\longrightarrow$	a_2	d_3	$\longrightarrow$	a_1
d_4	$\longrightarrow$	f_1	d_4	$\longrightarrow$	b_2
e_1	$\longrightarrow$	a_2	e_1	$\longrightarrow$	a_1
e_2	$\longrightarrow$	b_3	e_2	$\longrightarrow$	b_1
e_3	$\longrightarrow$	g_1	e_3	$\longrightarrow$	b_2
f_1	$\longrightarrow$	a_2	f_1	$\longrightarrow$	a_1
f_2	$\longrightarrow$	h_1	f_2	$\longrightarrow$	b_2
f_3	$\longrightarrow$	b_3	f_3	$\longrightarrow$	b_1
g_1	$\longrightarrow$	a_2	g_1	$\longrightarrow$	b_1
h_1	$\longrightarrow$	b_3	h_1	$\longrightarrow$	b_2

THE "COLORING" IS AN (ARBITRARY) ASSIGNMENT OF DATA SYMBOLS TO (SPLIT) CHANNEL STATE TRANSITIONS



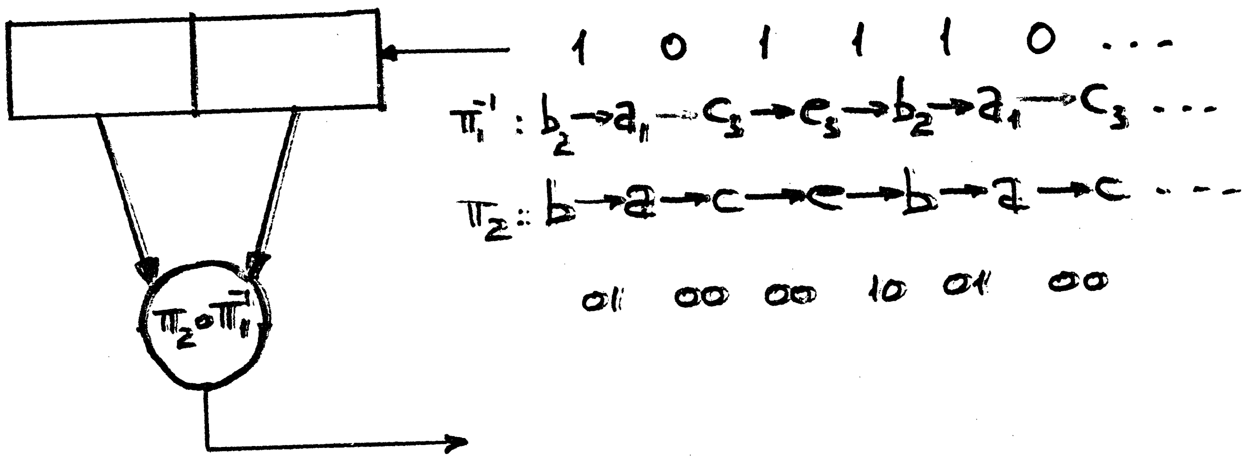
SCHEMATIC DIAGRAM OF ENCODER

RIGHT RESOLVING PROPERTY OF π_1^{-1}

THE ENCODING AMBIGUITY IS RESOLVED TO THE RIGHT IN A FINITE NUMBER OF STEPS RESULTING IN A <u>FIXED</u> <u>POINT</u> (THE RIGHT END SPLIT CHANNEL STATE IS FIXED INDEPENDENTLY OF THE PARTICULAR "ROAD" CHOSEN)

$$\begin{tikzcd}
a_1 \arrow[r, "0"] & c_3 \arrow[r, "0"] & e_2 \arrow[r, "0"] & b_3 \arrow[r, "0"] & d_3 \arrow[r, "1"] & a_1 \arrow[r, "1"] & c_2 \arrow[r, "0"] & b_3 \arrow[r, "1"] & d_4 \arrow[ddr, "1"] & \\
a_2 \arrow[r] & c_1 \arrow[r] & a_2 \arrow[r] & c_1 \arrow[r] & a_2 \arrow[r] & c_4 \arrow[r] & b_2 \arrow[r] & a_2 \arrow[r] & c_4 \arrow[dr] & \\
b_1 \arrow[r] & d_1 \arrow[r] & f_2 \arrow[r] & h_1 \arrow[r] & b_3 \arrow[r] & d_4 \arrow[r] & b_2 \arrow[r] & a_2 \arrow[r] & c_4 \arrow[r] & b_2 \\
\vdots & & & & & & & & & \\
h_1 \arrow[r] & b_3 \arrow[r] & d_3 \arrow[r] & a_2 \arrow[r] & c_1 \arrow[r] & a_1 \arrow[r] & c_2 \arrow[r] & b_3 \arrow[r] & d_4 \arrow[uur] &
\end{tikzcd}$$

RIGHT RESOLVING BLOCK $\mathcal{R} = 000011011$ RESULT. IN FIXED POINT b_2


1 0 1 1 1 0 ...
π_1^{-1} : $b_2 \to a_1 \to c_3 \to e_3 \to b_2 \to a_1 \to c_3 \cdots$
π_2 :: $b \to a \to c \to e \to b \to a \to c \cdots$
01 00 00 10 01 00
$\pi_2 \circ \pi_1^{-1}$

ONE STEP SEQUENTIAL ENCODER

$$f_E = \pi_2 \circ \pi_1^{-1}$$

$b_2, c_1, d_3, e_1, f_1 \longrightarrow a_1$

$b_2, c_1, d_3, e_1, f_1, g_1 \longrightarrow a_2$

$c_2, d_2, e_2, f_3, g_1 \longrightarrow b_1$

$c_4, d_4, e_3, f_2, h_1 \longrightarrow b_2$

$c_2, d_2, e_2, f_3, h_1 \longrightarrow b_3$

$a_2 \longrightarrow c_1$

$a_1 \longrightarrow c_2$

$a_1 \longrightarrow c_3$

$a_2 \longrightarrow c_4$

$b_1 \longrightarrow d_1$

$b_1 \longrightarrow d_2$

$b_3 \longrightarrow d_3$

$b_3 \longrightarrow d_4$

$c_4 \longrightarrow e_1$

$c_3 \longrightarrow e_2$

$c_3 \longrightarrow e_3$

$d_4 \longrightarrow f_1$

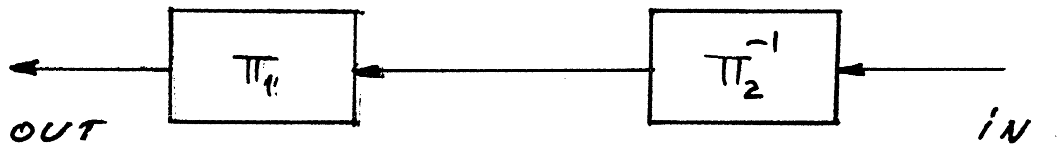
$d_1 \longrightarrow f_2$

$d_1 \longrightarrow f_3$

$e_3 \longrightarrow g_1$

$f_2 \longrightarrow h_1$

PREDECESSOR TABLE REPRESENTATION OF M_3



SCHEMATIC DIAGRAM OF DECODER

UNIFORM LEFT RESOLVING PROPERTY OF π_1^{-1}

EVERY CHANNEL SYMBOL IS UNIQUELY DECODABLE GIVEN A FIXED AMOUNT OF "LOOK-AHEAD". FOR THE PARTICULAR MATRIX M_3 CHOSEN THIS PARAMETER IS 3.

LENGTH 4 DECODING TREE STRUCTURE

a → c → e → g

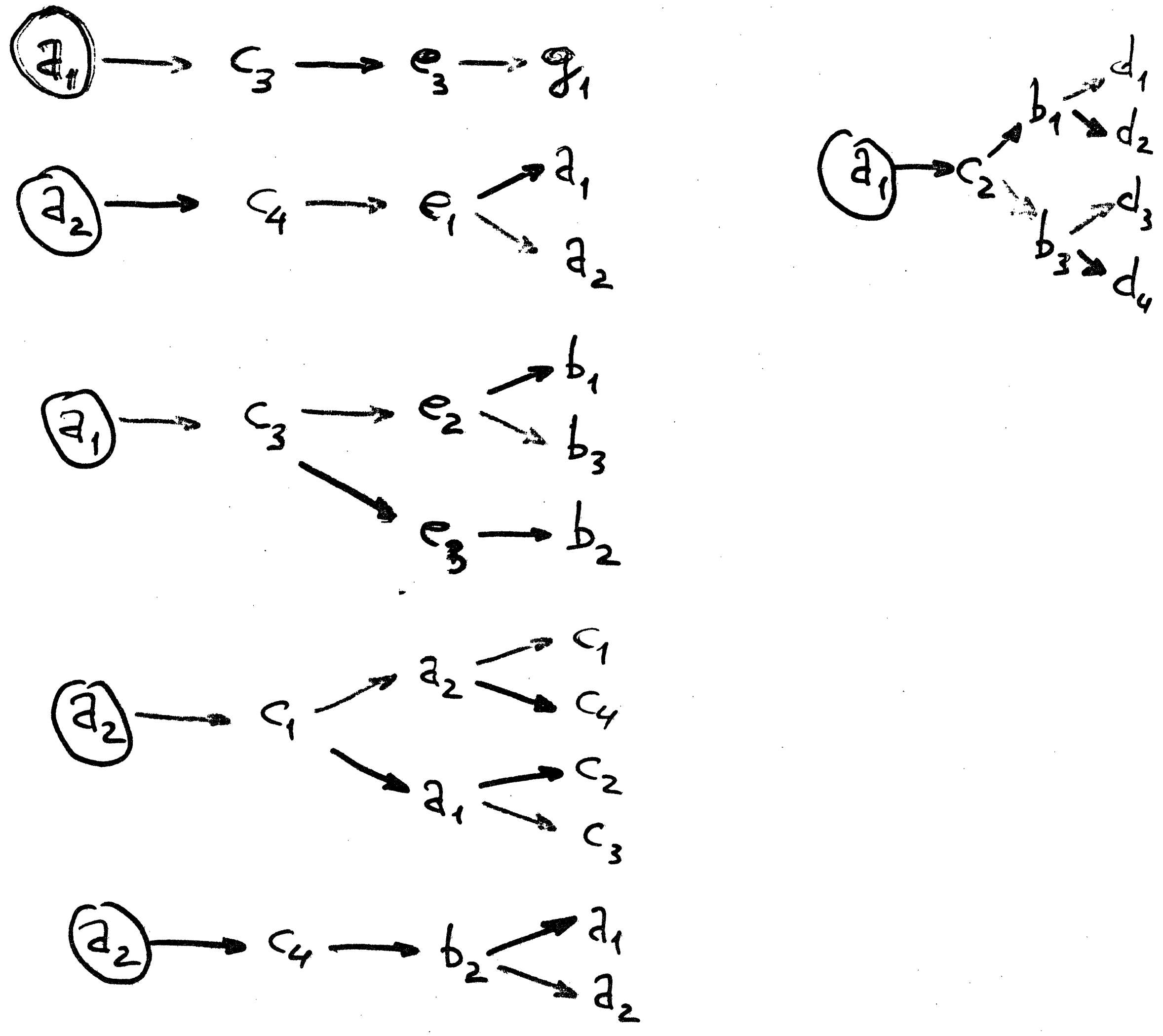
a → c → e → a

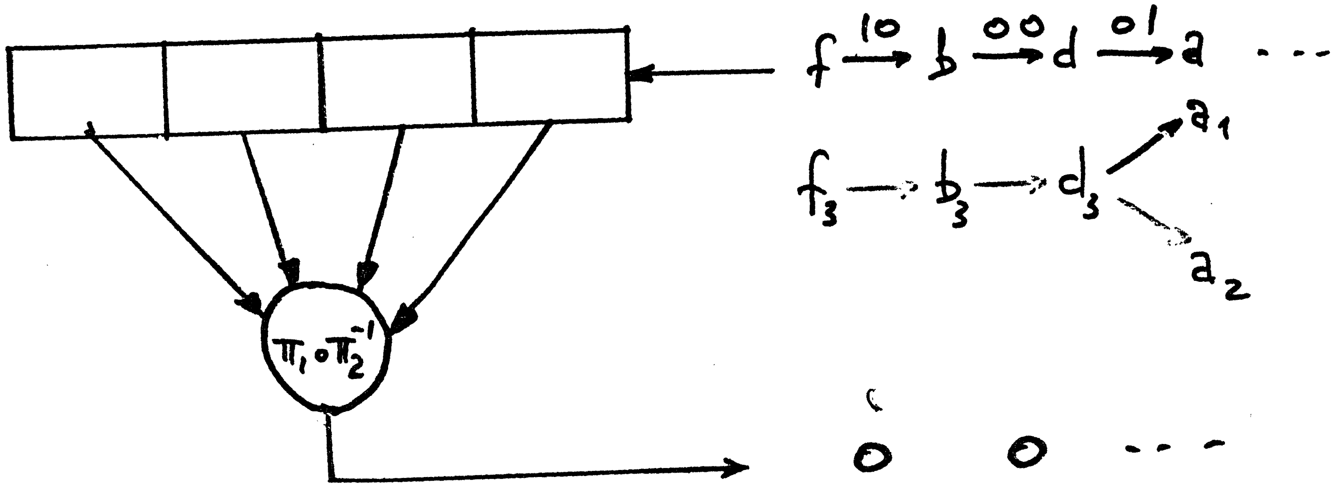
a → c → e → b

a → c → a → c

a → c → b → a

a → c → b → d





4-STEP LOOK-AHEAD SEQUENTIAL DECODER

$$f_D = \pi_1 \circ \pi_2^{-1}$$

RUN-LENGTH-LIMITED CHANNEL APPLICATIONS CONSIDERED (FOR MAGNETIC RECORDING)

1. SLIDING BLOCK CODE FOR (1,7)-CHANNEL WITH CODING RATE 2/3

2. SLIDING BLOCK CODE FOR (1,6)-CHANNEL WITH CODING RATE 2/3

APPLICATIONS TO FINITE STATE CHANNELS DEFINED BY RUN-LENGTH AND SPECTRAL NULL CONSTRAINTS

OBSERVATION

THESE ARE NOT FINITE-MEMORY CHANNELS SINCE THE CHANNEL STATES ARE NOT UNIQUELY DETERMINED BY A FINITE NUMBER OF PAST CHANNEL SYMBOLS.

CHANNEL STATE IDENTIFICATION REQUIRES IN ADDITION (CONSTRAINED) VALUES WHICH ARE CALCULATED FROM CUMULATIVE SPECTRAL SUMS, EVALUATED AT THE FREQUENCIES OF THE SPECTRAL NULLS SPECIFIED.

MAIN PROBLEM

THE OCCURRENCE OF AN ERROR IN THE CHANNEL SYMBOL SEQUENCE CAN BE CATASTROPHIC I.E IT MAY PROPAGATE INDEFINITELY UNLESS WE USE A STATE INDEPENDENT DECODER.

(STATE DEPENDENT DECODING CAN RESULT IN LARGE HARDWARE COMPLEXITY)

CODE FOR MAGNETIC CHANNEL OF TYPE $(0, k)$
WITH TWO SPECTRAL NULLS (ONE AT DC)

THIS PROBLEM WAS SUGGESTED BY T. HOWELL (SJ RES) AS APPLICABLE TO THE <u>BURIED</u> <u>SERVO</u> <u>PROJECT</u>.

FINITE STATE CHANNEL REPRESENTATION

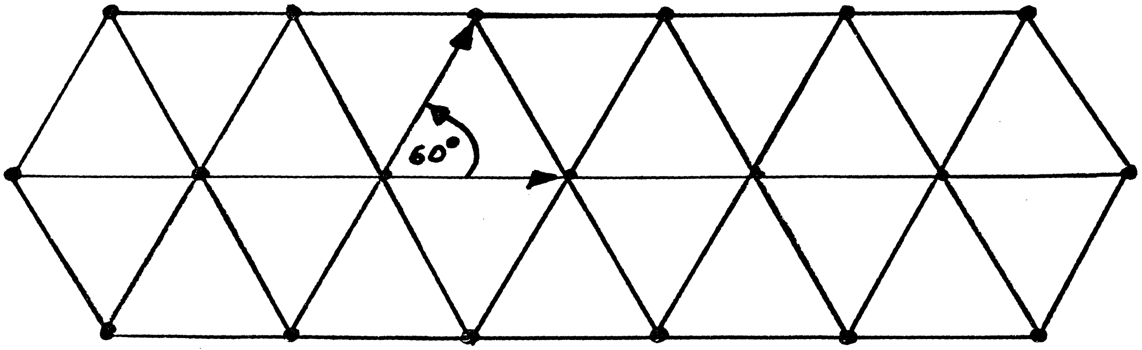
A NECESSARY AND SUFFICIENT CONDITION FOR THE EXISTENCE OF A SPECTRAL ZERO IN THE <u>AVERAGE</u> <u>POWER</u> <u>SPECTRUM</u> $S_X(\omega)$ OF A $\{-1, 1\}$- VALUED SEQUENCE $\{x_k\}$

$\left| \sum_{k=0}^{N} x_k e^{j\omega k} \right| \leq C$ FOR EVERY N IS EQUIVALENT TO $S_X(\omega) = 0$

OBSERVATION

FOR $\omega = 0, \frac{W}{4}, \frac{W}{6}$ THE VALUES OF $\left| \sum_{k=0}^{N} x_k e^{j\omega k} \right.$ OCCUR AT SPECIAL POINTS IN THE COMPLEX PLANE (FOR $\omega = 0$ THESE ARE THE INTEGERS ON THE REAL LINE).

FOR $\omega = \frac{W}{6}$ THESE ARE THE LATTICE POINTS OF A TILING OF THE PLANE BY EQUILATERAL TRIANGLES



STATE DESCRIPTION OF $\left\{ S_X(0)=0,\ S_X\left(\frac{W}{6}\right)=0 \right\}$

A STATE σ IS SPECIFIED BY A TRIPLET (S1, S2, S3)

$$S1 = \sum_{k=0}^{m} x_k$$

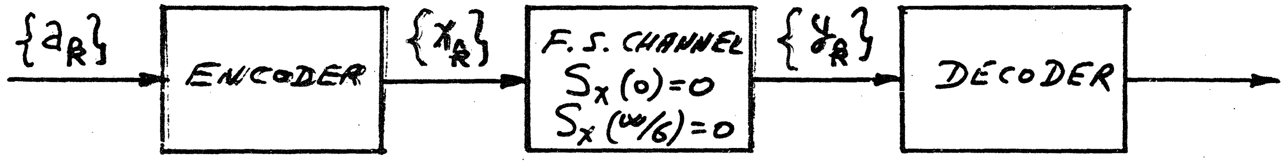
$$S2 = \mathrm{Re} \sum_{k=0}^{m} x_k e^{j \frac{W}{6} k}$$

S3 = VALUE OF COMPONENT OF $\sum_{k=0}^{m} x_k e^{j\frac{W}{6}k}$ IN THE DIRECTION OF $e^{j\frac{W}{6}}$

WE CONSTRAIN THE STATE TRIPLETS TO BE INTEGERS IN THE RANGE $(-C, -C+1, \dots, C-1, C)$. FOR EACH VALUE OF C WE COMPUTE THE CHANNEL CAPACITY AS THE MAXIMAL EIGENVALUE OF A STATE TRANSITION MATRIX.

FOR C=4 THE FEASIBLE (NOISELESS) CODING RATE IS 4/6

CODE PROPERTIES



1. THE CODE MAPS 4 UNCONSTRAINED BITS INTO A CHANNEL STATE TRANSITION THAT GENERATES 6 CONSTRAINED (OUTPUT) BITS

2. THE RUNLENGTHS OF 0'S AND 1'S DO NOT EXCEED 5 (EQUIVALENTLY THERE ARE NO MORE THAN 4 ZEROS BETWEEN TWO ONES)

3. THE CODE HAS A SPECTRAL ZERO AT $\omega/2$

4. THE DECODER IS STATE INDEPENDENT AND HENCE DOES NOT PROPAGATE ERRORS. THIS IS ACHIEVED BY ENCODING THE SAME DATA BLOCK INTO STATE TRANSITIONS THAT CORRESPOND TO IDENTICAL BIT PATTERNS (AFTER DROPPING THE STATE LABELS). THIS ENSURES THAT A CHANNEL ERROR WILL NOT PROPAGATE SINCE WE CAN DECODE CORRECTLY EVEN THOUGH THE DECODER MAY BE IN AN ERRONEOUS STATE (WHICH IS INDISTINGUISHABLE FROM THE CORRECT ONE)

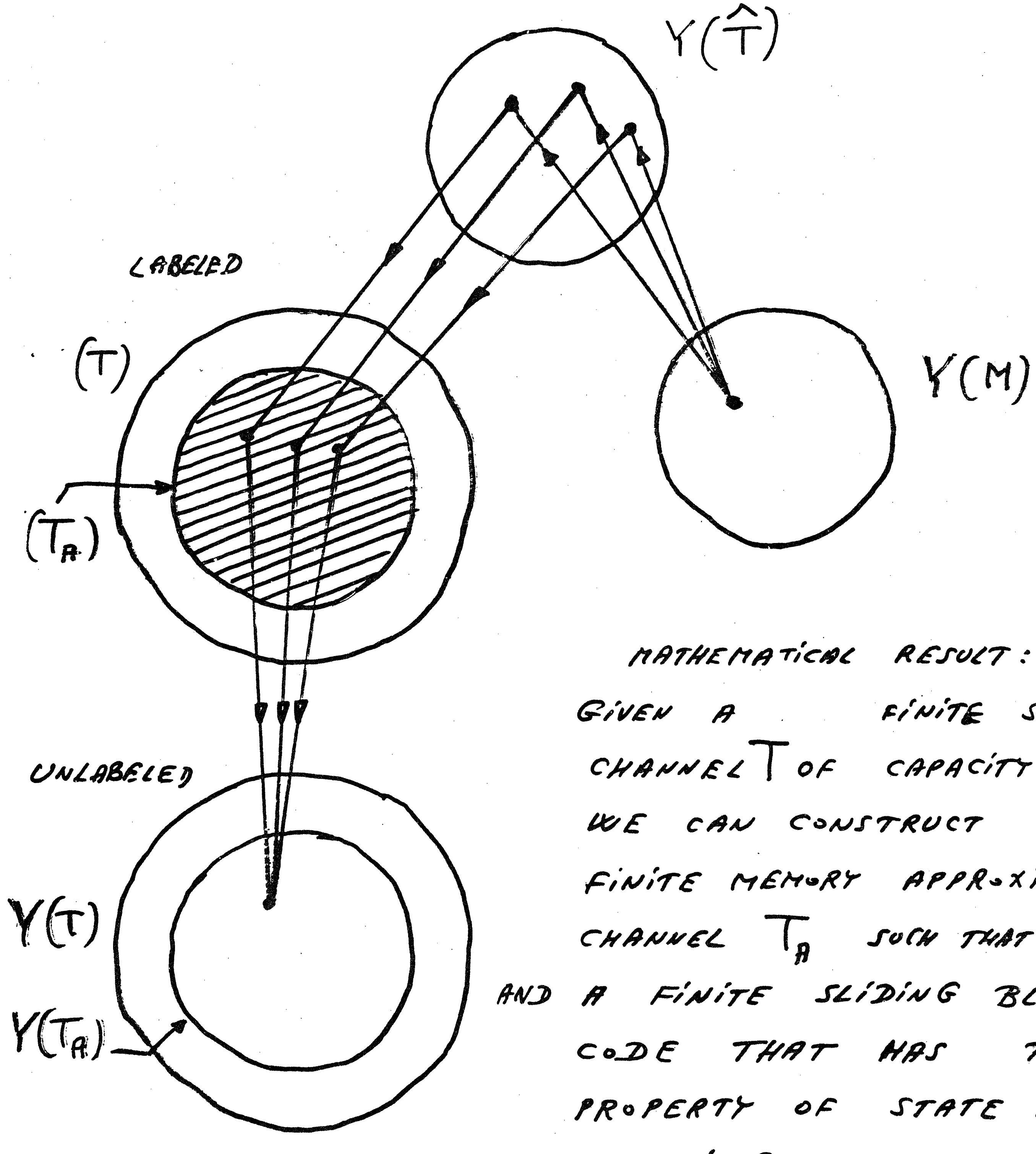
FOR OUR PARTICULAR CASE THE MAXIMUM ERROR PROPAGATION IS 4 BITS.

OBSERVATION

THIS CODING STRATEGY TO ACHIEVE STATE INDEPENDENT DECODING IS FEASIBLE PROVIDED THE ROW SUM OF THE STATE TRANSITION MATRIX $\geq \lambda_M^n$. IT ALSO SEEMS THAT FOR THIS CASE STATE INDEPENDENT BLOCK CODING SCHEMES CAN BE FOUND.

IN THE GENERAL CASE STATE INDEPENDENT BLOCK CODING SCHEMES WITHOUT ADDITIONAL OVERHEAD DO NOT EXIST.

STATE INDEPENDENT SLIDING BLOCK CODE SOLUTION



MATHEMATICAL RESULT:

GIVEN A FINITE STATE CHANNEL T OF CAPACITY $C_T > H_M$ WE CAN CONSTRUCT A FINITE MEMORY APPROXIMATION CHANNEL T_A SUCH THAT $C_{T_A} = H_M$ AND A FINITE SLIDING BLOCK CODE THAT HAS THE PROPERTY OF STATE INDEP. DECODING.

UNSOLVED (YET) PROBLEM : $C_T = H_M$ (ZERO-MODULATION)

APPLICATIONS

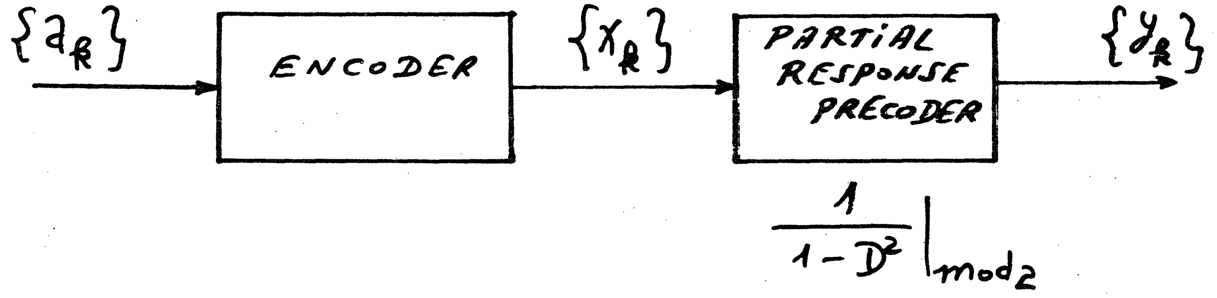
1. (0,2) DC- FREE CODE AT CODING RATE 3/4 WITH ERROR PROPAGATION OF 3 BITS

2. R. SCHNEIDER SUGGESTED (2,7) DC FREE CODE

CODE FOR PARTIAL RESPONSE (CORRELATIVE) CHANNEL WITH $(0,k)$ RUN-LENGTH CONSTRAINT AND SPECTRAL ZERO AT DC

THIS PROBLEM WAS SUGGESTED BY R. SCHNEIDER (GPD, TUCSON) AND WORKED JOINTLY WITH HIM. P. FRANASZEK (RES. YORKTOWN) HAS EQUIVALENT BLOCK CODE SOLUTION.

FINITE STATE CHANNEL DESCRIPTION


$\{a_k\}$
ENCODER
$\{x_k\}$
PARTIAL RESPONSE PRECODER
$\{y_k\}$
$\left.\frac{1}{1-D^2}\right|_{\text{mod}\,2}$

A NECESSARY AND SUFFICIENT CONDITION FOR $S_y(0)=0$ CAN BE STATED IN TERMS OF THE SPECTRAL PRODUCT SUMS OF $\{x_k\}$

(EVEN SUM) $\sigma_m^e(0) = \sum_{k=0}^{m} x_0 \cdots x_{2k}$

(ODD SUM) $\sigma_m^o(0) = \sum_{k=\text{[illegible]}}^{m} x_1 \cdots x_{2k+1}$

$$\sigma_m^e(0) + \sigma_m^o(0) = \sum_{k=0}^{m} y_{-2}\, y_{2k} + \sum_{k=1}^{m} y_{-1}\, y_{2k+1}$$

$$S_y(0)=0 \Longleftrightarrow \left| \sum_{k=0}^{m} y_k \right| \leq \left| \sigma_m^e(0) + \sigma_m^o(0) \right| \leq C$$

AN EQUIVALENT STATE DESCRIPTION WITH LESS STATES RESULTS IF WE USE "PATEL'S CHARGE RULE" (A. PATEL, GPD SJ, SEE ZERO MODULATION)

$$\tau_i(0) \xrightarrow{0} \tau_i(0)+1$$

$$\tau_i(0) \xrightarrow{1} -\tau_i(0)+1$$

$$\tau_m(\omega) = \sum_{k=0}^{m} x_k \dots x_m e^{jk\omega} + 1 = \left[\sigma_m(\omega) - \sigma_{m-1}(\omega)\right]\left[\sigma_m(\omega)+1\right]$$

$$\boxed{\tau_m(\omega) = \Delta\sigma_m(\omega)\left[\sigma_m(\omega)+1\right]}$$

CODE PROPERTIES

1. CODING RATE IS 3/4 WHERE 3 BITS ARE ENCODED INTO A CHANNEL STATE TRANSITION THAT OUTPUTS 4 CONSTRAINED BITS

2. IN THE SEQUENCE $\{y_k\}$ AT THE OUTPUT OF THE PARTIAL RESPONSE PRECODER THE RUNLENGTH OF ZEROS BETWEEN TWO ONES DOES NOT EXCEED 5.

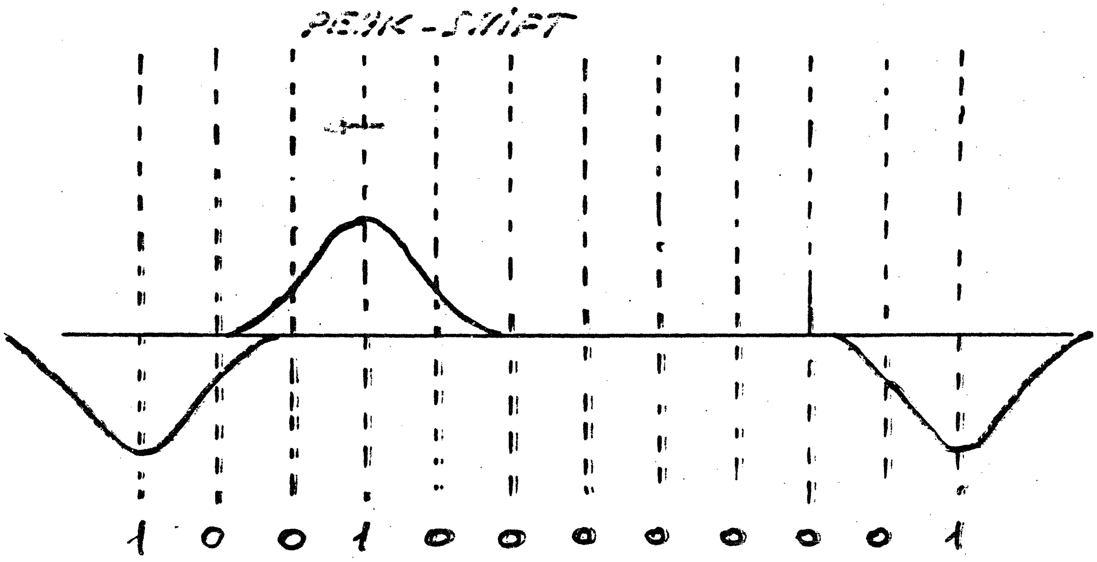
3. A CHANNEL ERROR DUE TO RANDOM NOISE WILL NOT PROPAGATE BEYOND 3 BITS OF DECODED DATA

4. PART OF THE ERRORS CAN BE CORRECTED

NOTE: THIS CHANNEL HAS A TRANSITION MATRIX WHOSE ROWSUM $\geq 2^3 \left(= \lambda_M^n\right)$ AND THUS STATE-INDEP DECODING IS ALSO ACHIEVABLE BY BLOCK SCHEMES [FRANASZEK]

CODE FOR (d, k) – CHANNEL THAT ELIMINATES WORST CASE PEAK SHIFT BIT PATTERNS

THIS PROBLEM WAS SUGGESTED BY E. CUNNINGHAM (ISD ROCH.)



WORST CASE PEAK-SHIFT FOR (2,7)-CODE IS CREATED SHORT-TO-LONG AND LONG-TO-SHORT RUNLENGTH TRANSITIONS

PROBLEM : ELIMINATE $10^{2}10^{7}1$

$10^{7}10^{2}1$

$10^{2}10^{6}1$

$10^{6}10^{2}1$

$10^{2}10^{5}1$

FINITE MEMORY MODEL

EACH CHANNEL STATE IS DEFINED BY A TRIPLET (S1, S2, S3)

$S1 \in \{2, \dots, 7\}$	FIRST RLL
$S2 \in \{2, \dots, 7\}$	SECOND RLL
$S3 \in \{0, 1\}$	LAST BIT

FROM STATE TRANSITION MATRIX WE CALCULATE CHANNEL CAPACITY

PRELIMINARY CONCLUSIONS

1. $10^{2}10^{7}1$ AND $10^{7}10^{2}1$ CAN BE ELIMINATED FROM (2,7)-CODE WITHOUT LOWERING THE CODING RATE.

2. IN ADDITION $10^{2}10^{6}1$ AND $10^{6}10^{2}1$ CAN BE ELIMINATED HOWEVER k HAS TO BE INCREASED FROM 7 TO 8 POSSIBLY 9

3. THE PATTERNS $10^{2}10^{5}1$ AND $10^{5}10^{2}1$ CANNOT BE EXCLUDED FROM A $(2, k)$-CODE WITH ARBITRARY UNLESS WE LOWER THE CODING RATE.

SUMMARY

THIS IS A GENERAL METHOD FOR THE CONSTRUCTION OF MODULATION CODES DEFINED BY AN ARBITRARY FINITE SET OF CONSTRAINTS. THE SLIDING BLOCK CODES FURTHERMORE POSSESS THE PROPERTY OF STATE INDEP DECODING.

PROBLEMS

1. HOW TO DERIVE CONSTRAINTS OF INTEREST, BESIDES THOSE MENTIONED ((d, k), SPECTRAL NULLS ETC.

2. HOW TO QUANTIFY THE IMPROVEMENTS DUE TO CODING.

ERROR CONTROL CODES -- A TUTORIAL -- R. E. Blahut

The attached charts were used as part of a tutorial on error-control codes. The tutorial discussed the nature of the coding problem, solutions, and applications. A description of a simple Hamming (7,4) code illustrates many of the elementary ideas. A discussion of the role of Galios fields in coding follows. Next comes some circuits for decoding simple codes. The tutorial ends with a description of concatenated codes and convolutional codes.

ERROR CONTROL CODES

A TUTORIAL

SHANNON SAYS —

- TO ACHIEVE CHANNEL CAPACITY ONE MUST USE AN ECC
- DIVIDE COMMUNICATION PROBLEM INTO MODULATION AND ECC TO MAKE LIFE SIMPLE
- DESIGN MODULATION AND ECC SIMULTANEOUSLY

APPLICATIONS

- SATELLITE COMMUNICATIONS
  > 100 MBPS
- DEEP SPACE COMMUNICATION
  (255, 223) $t=16$ REED-SOLOMON (8 BIT-BYTE
- MILATARY COMMUNICATION
  10 MBPS
  JAM PROTECTION
- TAPES / DISCS
- DATA BUS
- DIGITAL VIDEO & AUDIO (TV & PHONO)
- VOICEBAND MODEMS
- FACSIMILE

MASSIVE CODES ARE NOW POSSIBLE

TRANSMITTED MESSAGE

THE QUICK BROWN FOX JUMPED ...

RECEIVED MESSAGE WITH RANDOM ERRORS ...

THE QUIRK BROWN SOXTJUMPRD ...

RECEIVED MESSAGE WITH BURST ERRORS

THE QUICK BRZATCLTX JUMPED ...

RECEIVED MESSAGE WITH ERASURES

THE QUI_K BROWN _OX J_MP_D ...

RECEIVED MESSAGE WITH SOFT DECISIONS

THE QUIRK BROWN ...
.9 .7 .2 .9 .8 .2 3 .9

OR LIST DECISIONS

THE QUIRK BROWN ...
S P C TL

OR READABLE ERASURES

THE QUIRK BROWN SOX ...

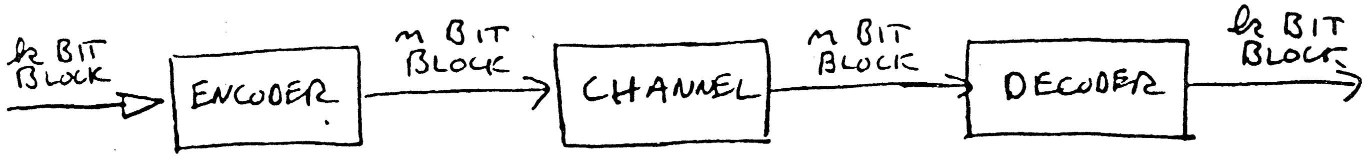
COMPLETE DECODER

GIVES ITS BEST GUESS ON EVERY MESSAGE - NO MATTER HOW BAD

INCOMPLETE DECODER

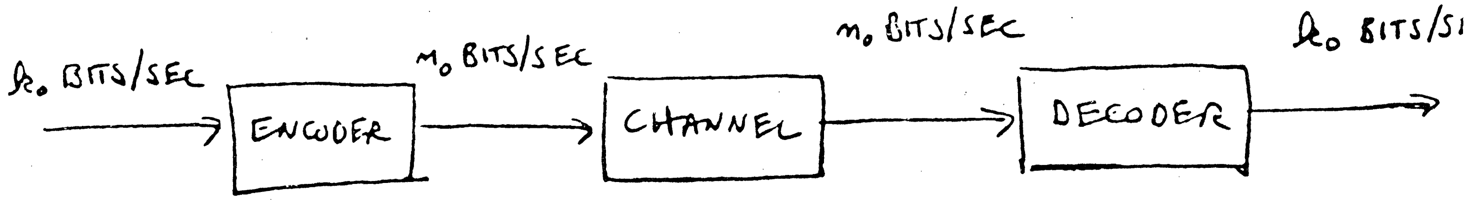
ONLY CORRECTS UP TO SOME LIMIT - OTHERWISE FLAGS MESSAGE AS BAD

INFORMATION SEQUENCE — k BITS
CODEWORD — n BITS
RATE = k/n

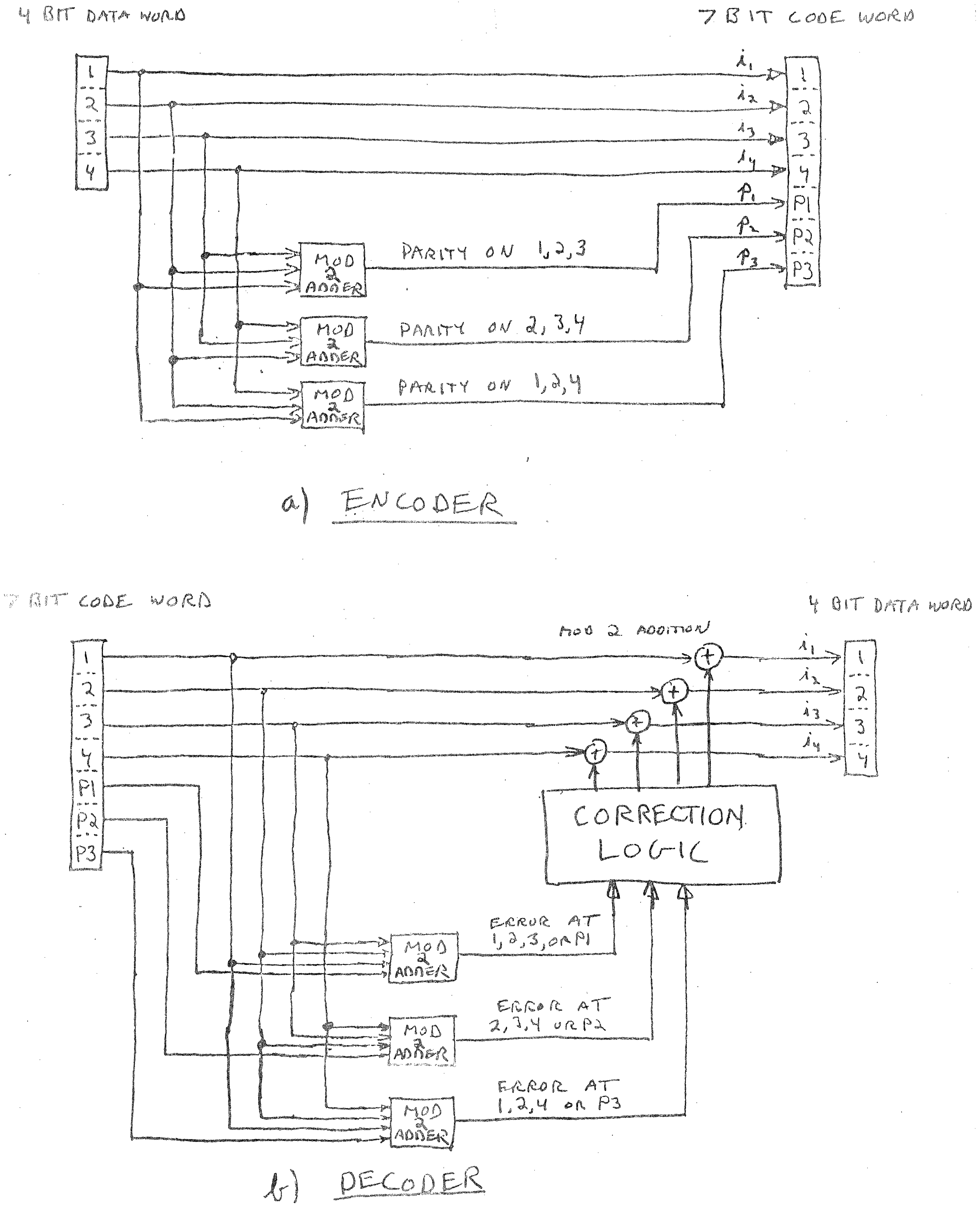


BLOCK CODE

INFORMATION RATE - k_0 BITS/SEC
CHANNEL RATE - n_0 BITS/SEC
CODE RATE - k_0/n_0 (BITS/BIT)



TREE CODE



a) ENCODER

b) DECODER

A SIMPLE HAMMING (7,4) ENCODER/DECODER

$2^{80} \cdot 2^{20}$

$2^{4 \times 10^7}$

0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

Table 1.5-1

Hamming (7,4) Code

H CODES

NONSYSTEMATIC ENCODING

0	0	0	0	0	0	0	⟷	0 0 0 0
0	0	0	1	0	1	1		1 0 0 0
0	0	1	0	1	1	0		0 1 0 0
0	0	1	1	1	0	1		1 1 0 0
0	1	0	0	1	1	1		0 0 1 0
0	1	0	1	1	0	0		1 0 1 0
0	1	1	0	0	0	1		0 1 1 0
0	1	1	1	0	1	0		1 1 1 0
1	0	0	0	1	0	1		0 0 0 1
1	0	0	1	1	1	0		1 0 0 1
1	0	1	0	0	1	1		0 1 0 1
1	0	1	1	0	0	0		1 1 0 1
1	1	0	0	0	1	0		0 0 1 1
1	1	0	1	0	0	1		1 0 1 1
1	1	1	0	1	0	0		0 1 1 1
1	1	1	1	1	1	1		1 1 1 1

Table 1.5-1

Hamming (7,4) Code

SYSTEMATIC ENCODING
←→

0	0	0	0	0	0	0	0000
0	0	0	1	0	1	1	0001
0	0	1	0	1	1	0	0010
0	0	1	1	1	0	1	0011
0	1	0	0	1	1	1	0100
0	1	0	1	1	0	0	0101
0	1	1	0	0	0	1	0110
0	1	1	1	0	1	0	0111
1	0	0	0	1	0	1	1000
1	0	0	1	1	1	0	1001
1	0	1	0	0	1	1	1010
1	0	1	1	0	0	0	1011
1	1	0	0	0	1	0	1100
1	1	0	1	0	0	1	1101
1	1	1	0	1	0	0	1110
1	1	1	1	1	1	1	1111

Table 1.5-1

Hamming (7,4) Code

GF(2)	GF(4)	GF(8)	GF(16)
(7,4)	(5,3)	(9,7)	(17,15)
(15,11)	(21,18)	(73,70)	(273,270)
(31,26)	(85,81)	(585,581)	
(63,57)	(341,336)		
(127,120)			

Table 3.4-1

Descriptors of Some Hamming Codes

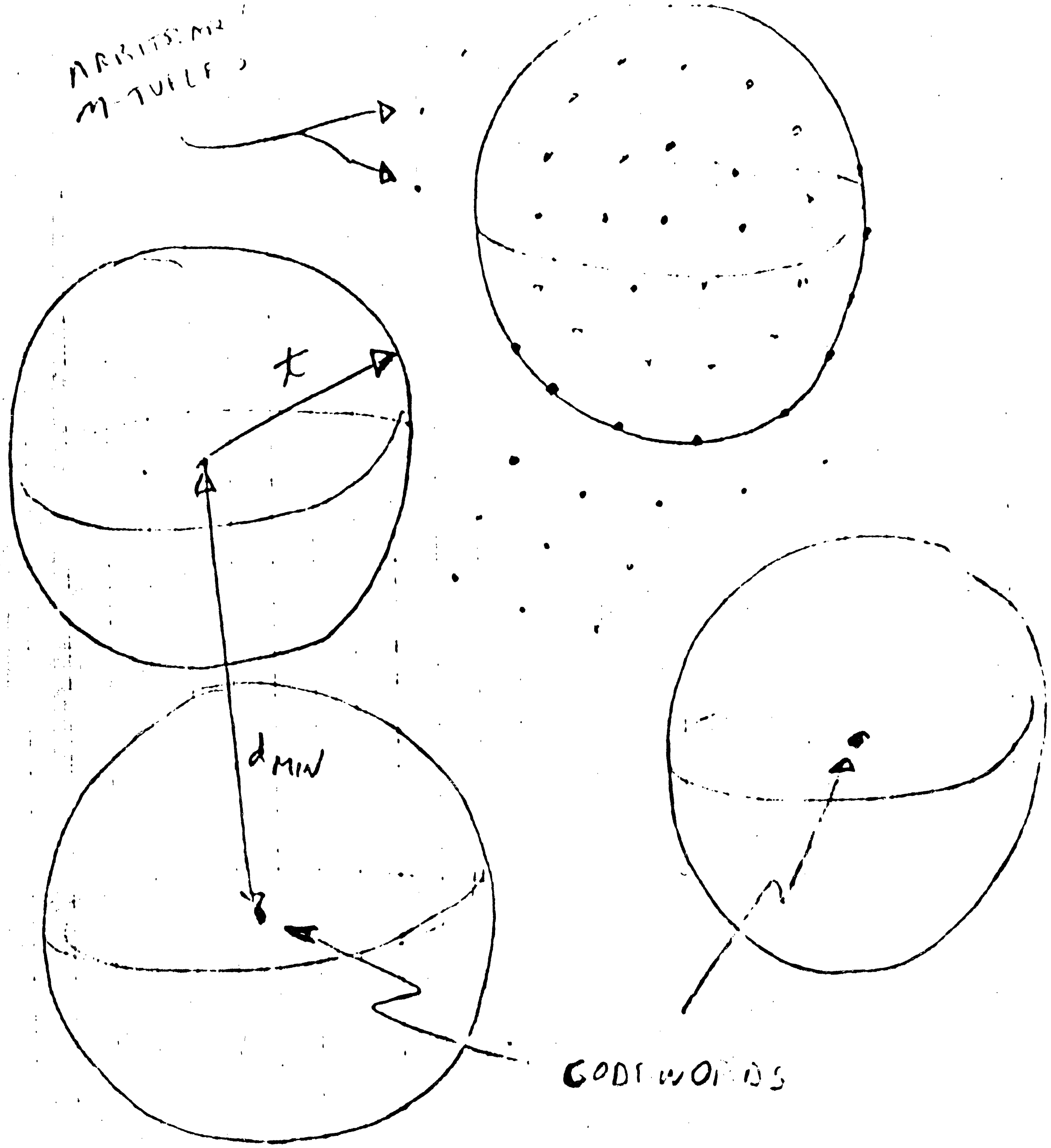


Figure 1.4-2

Decoding Spheres

ÉVARISTE GALOIS
(1811-1832)

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
2	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
3	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
4	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
5	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
7	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
8	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
A	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
B	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
C	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
D	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
E	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Addition Table

The Hexadecimal Field

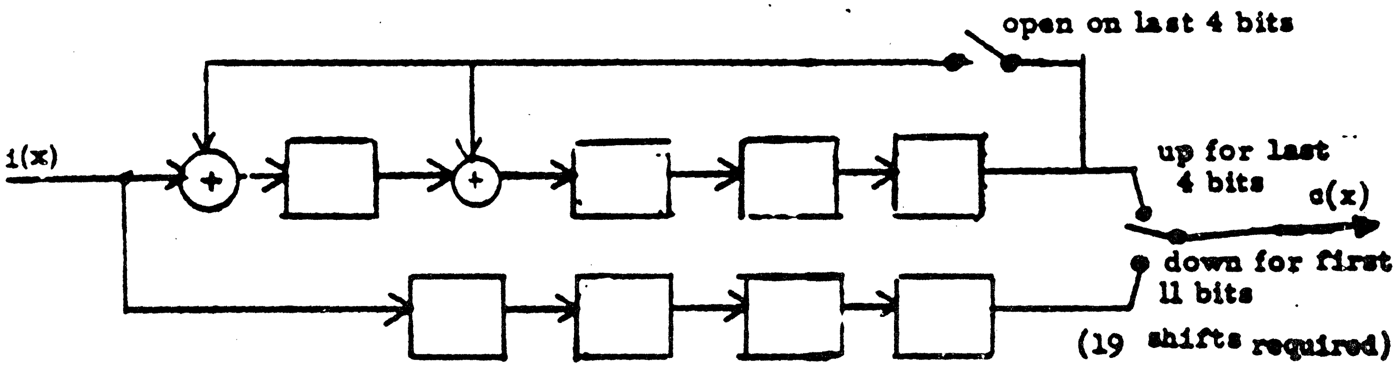
Figure 2.1-1a

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	3	1	7	5	B	9	F	D
3	0	3	6	5	C	F	A	9	B	8	D	E	7	4	1	2
4	0	4	8	C	3	7	B	F	6	2	E	A	5	1	D	9
5	0	5	A	F	7	2	D	8	E	B	4	1	9	C	3	6
6	0	6	C	A	B	D	7	1	5	3	9	F	E	8	2	4
7	0	7	E	9	F	8	1	6	D	A	3	4	2	5	C	B
8	0	8	3	B	6	E	5	D	C	4	F	7	A	2	9	1
9	0	9	1	8	2	B	3	A	4	D	5	C	6	F	7	E
A	0	A	7	D	E	4	9	3	F	5	8	2	1	B	6	C
B	0	B	5	E	A	1	F	4	7	C	2	9	D	6	8	3
C	0	C	B	7	5	9	E	2	A	6	1	D	F	3	4	8
D	0	D	9	4	1	C	8	5	2	F	B	6	3	E	A	7
E	0	E	F	1	D	3	2	C	9	7	6	8	4	A	B	5
F	0	F	D	2	9	6	4	B	1	E	C	3	8	7	5	A

Multiplication Table

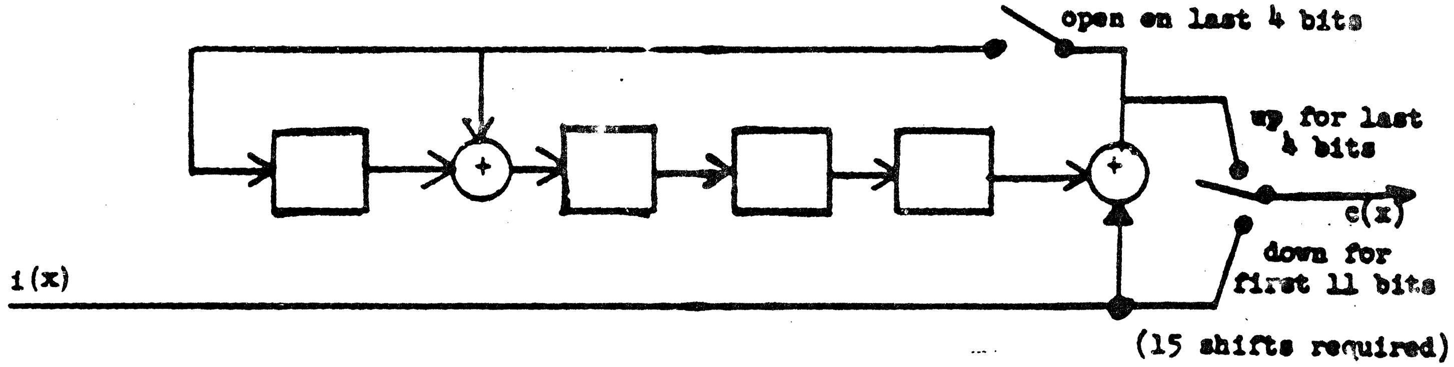
The Hexadecimal Field

Figure 2.1-1b



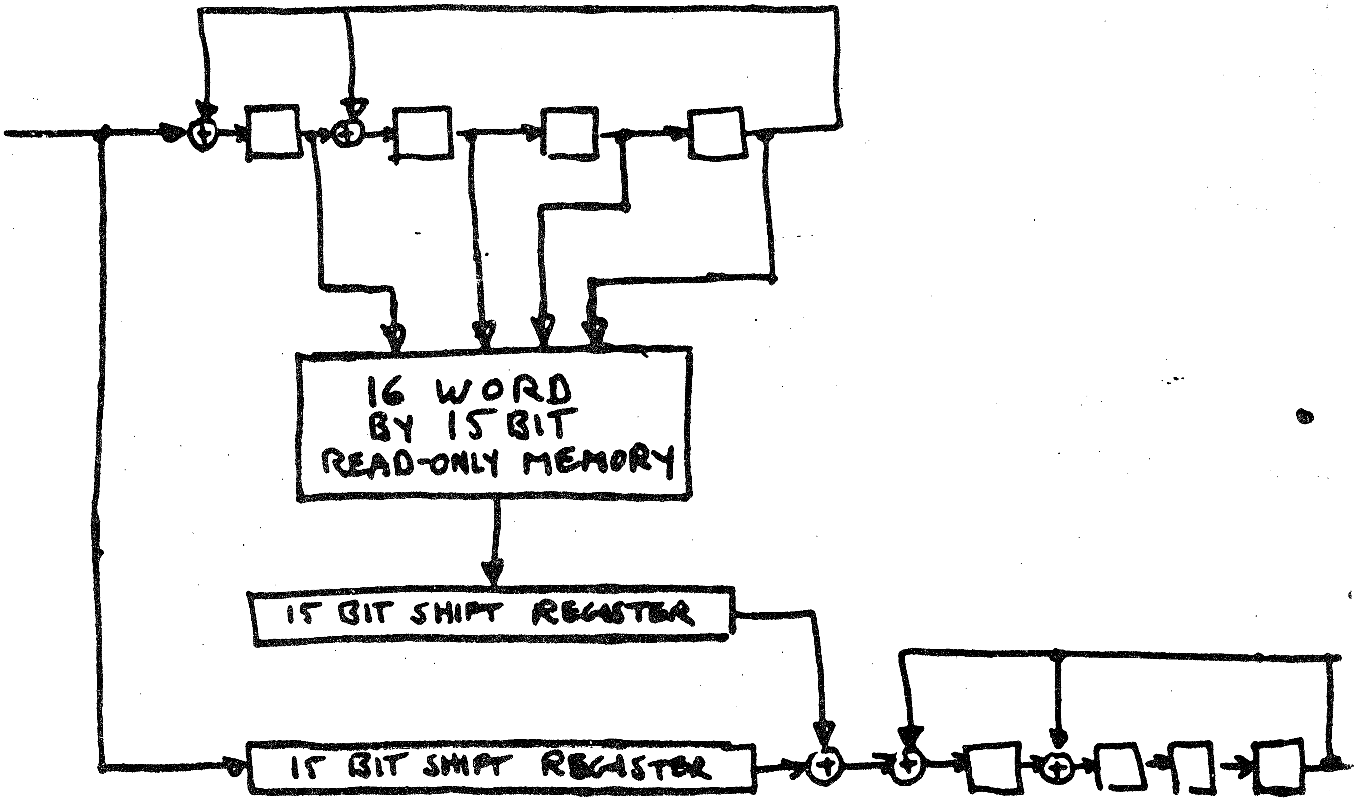
A SYSTEMATIC ENCODER
FOR HAMMING (15, 11) CODE

Figure 6.3-3



ANOTHER SYSTEMATIC ENCODER
FOR HAMMING (15, 11) CODE

Figure 6.3-4



SYNDROME DECODER FOR NONSYSTEMATIC
HAMMING (15, 11) CODE

FIGURE 63-3

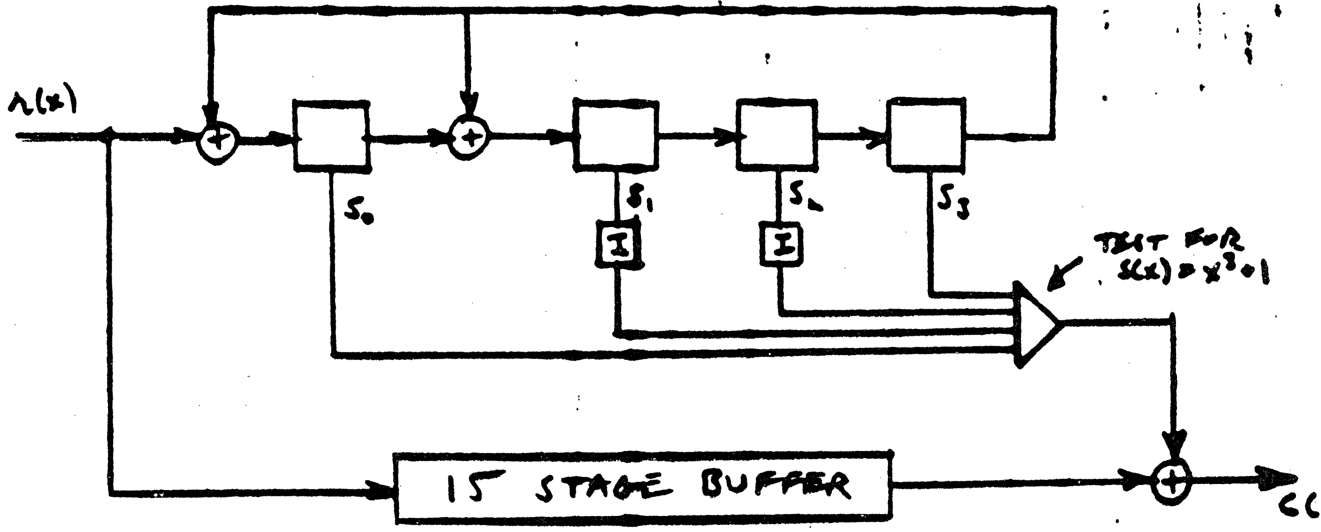
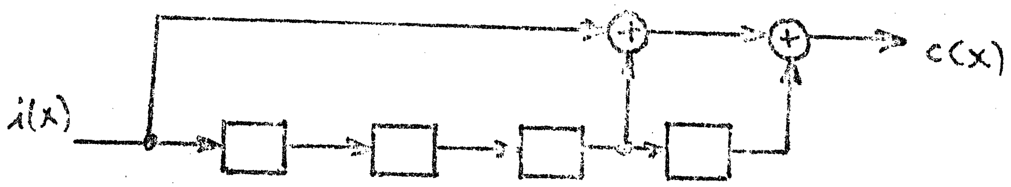
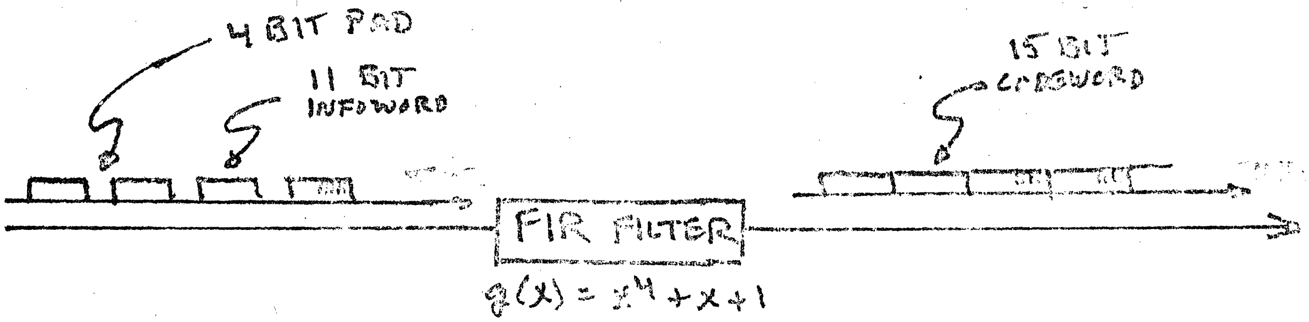


FIGURE 6.4-2
Meggitt Decoder for a (15,11) Hamming Code

$$c(x) = i(x)(x^4+x+1)$$



NON-SYSTEMATIC (15,11) ENCODER
FIGURE 6.3-1



ENCODING A LONG BIT STREAM
FIGURE 6.3-2

NOTES

- All Data Paths are 2 Bits Wide
- 36 Shifts to Decode

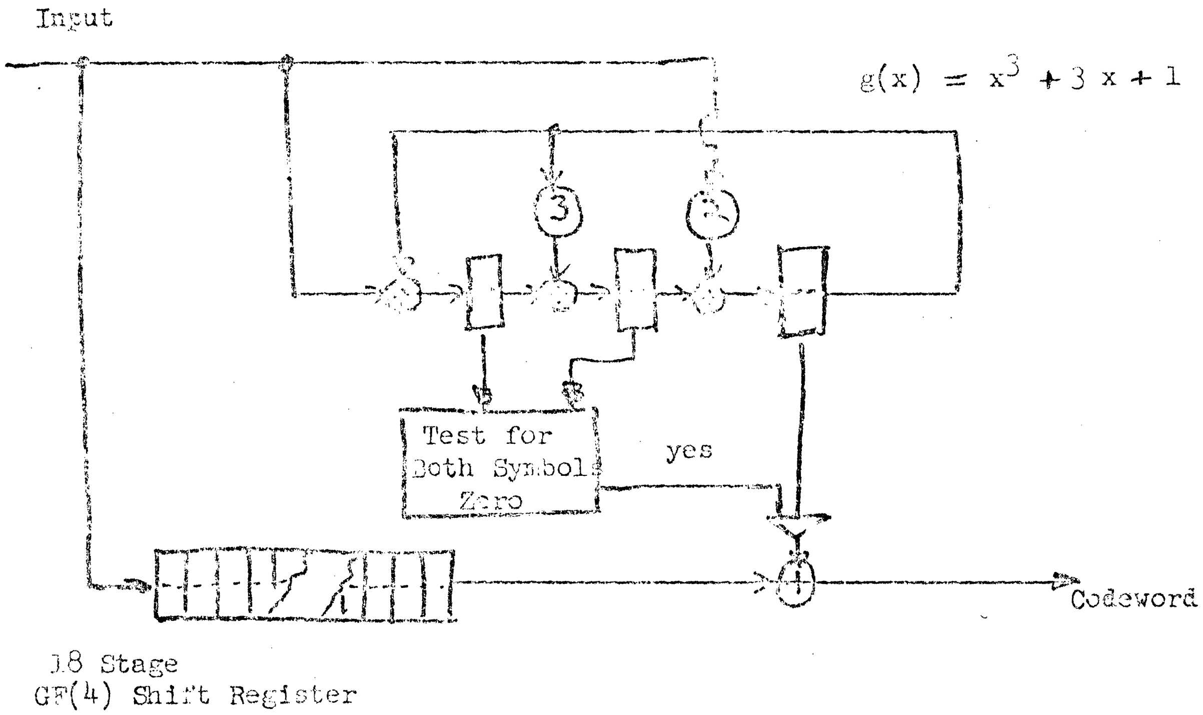


Figure 6.6-2

A Meggitt Decoder for an (18,15) Shortened Hamming Code Over GF(4)

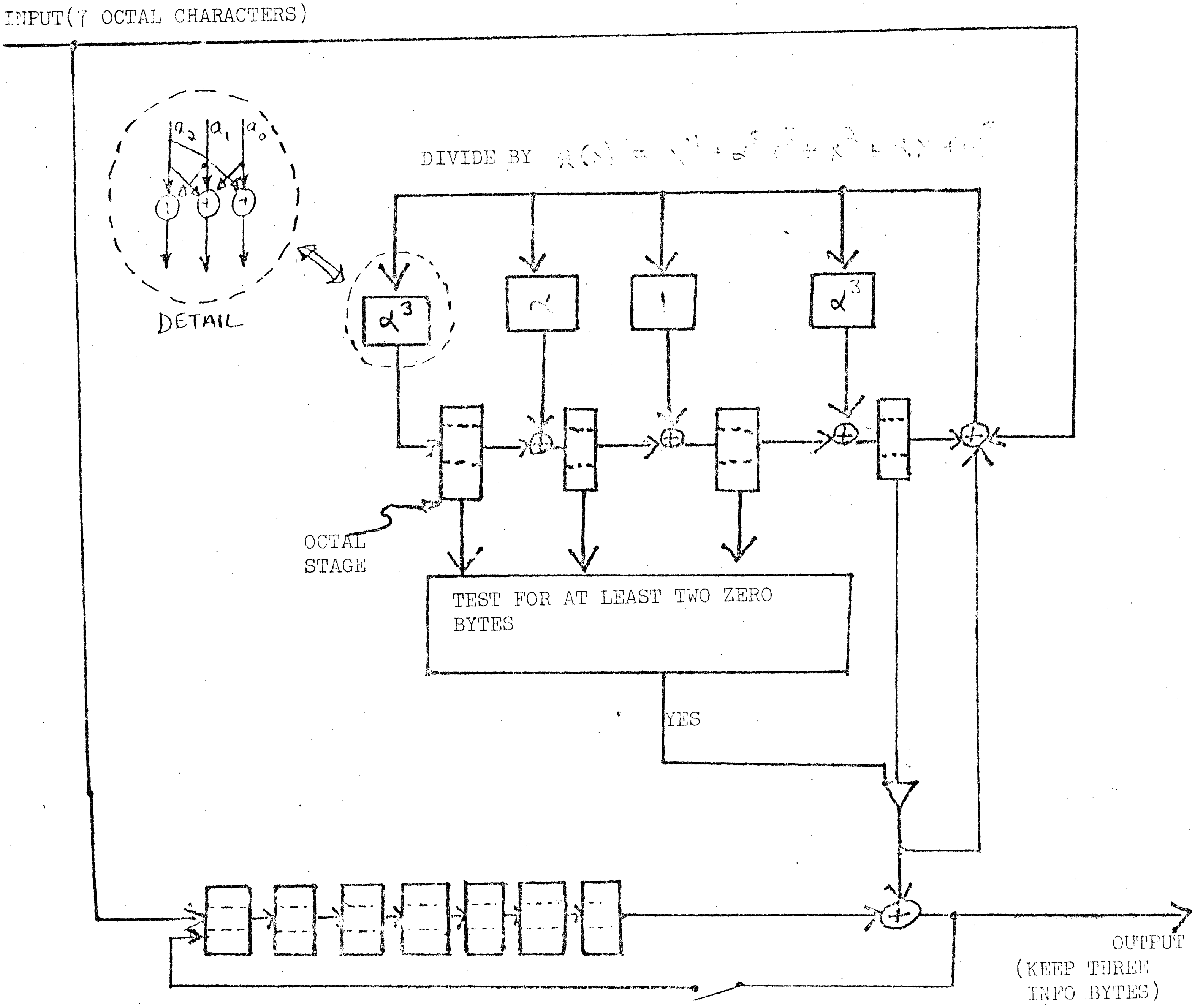
FIGURE 6.5-4

ERROR TRAPPING

FOR (7,3) REED-SOLOMON

DOUBLE SYMBOL ERROR CORRECTION

Note: All Data Paths are
Three Bits Wide


INPUT(7 OCTAL CHARACTERS)
a_2
a_1
a_0
DETAIL
DIVIDE BY $g(x) = x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3$
α^3
α
1
α^3
OCTAL STAGE
TEST FOR AT LEAST TWO ZERO BYTES
YES
OUTPUT
(KEEP THREE INFO BYTES)

21 SHIFTS REQUIRED

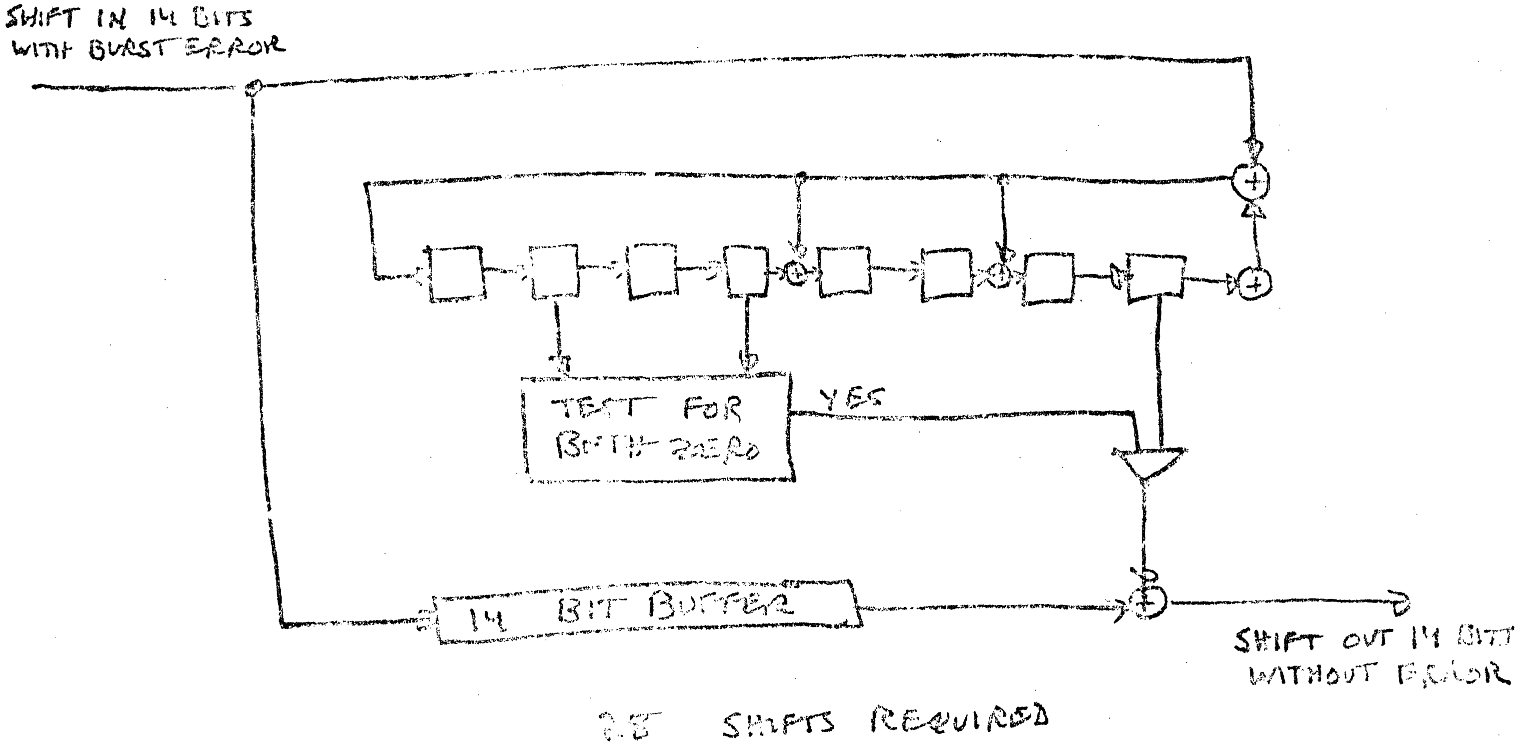
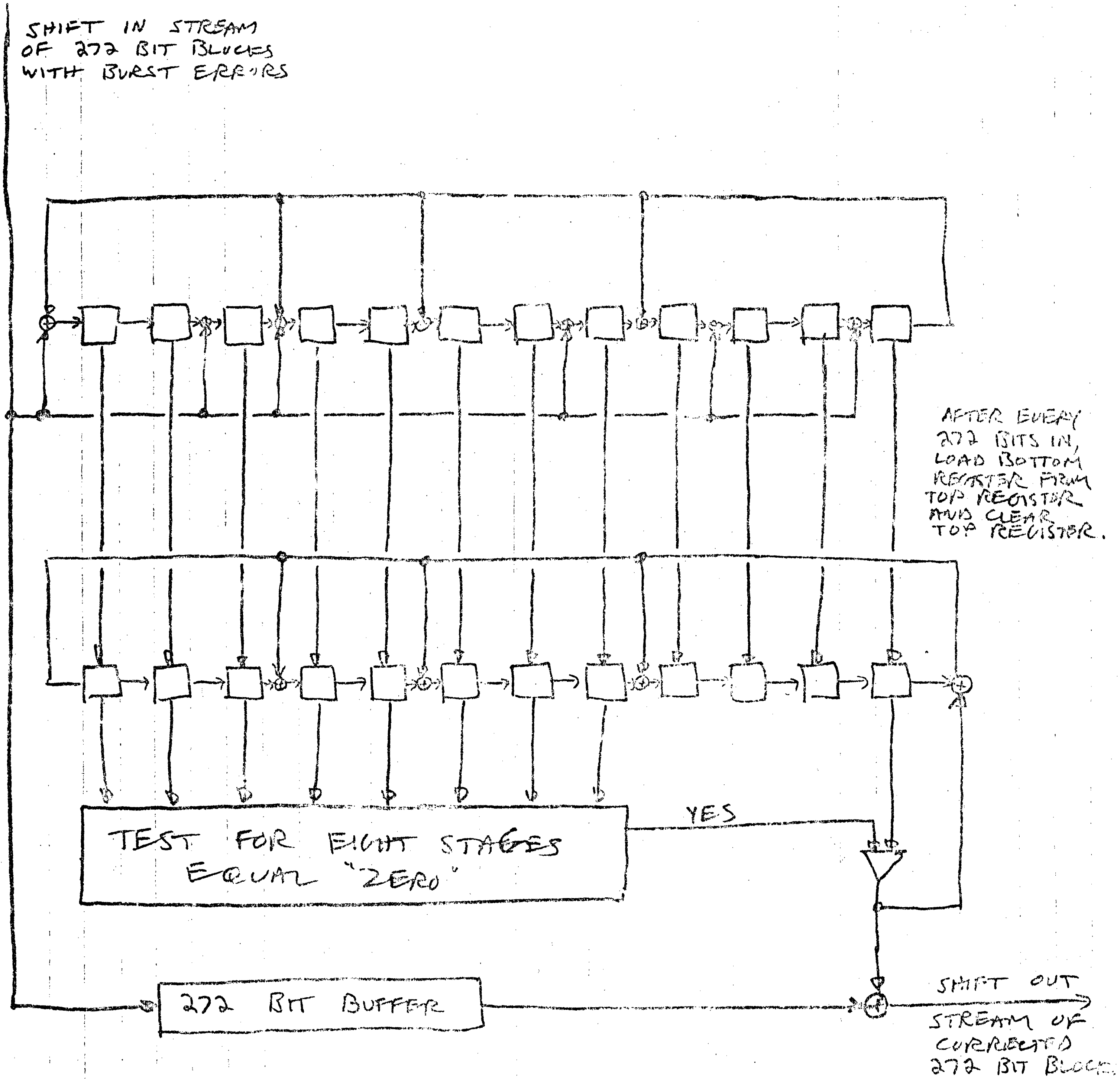


FIGURE 6.5-6
ERROR TRAPPING DECODER FOR
AN INTERLEAVED BURST-ERROR CORRECTING CODE



ERROR TRAPPING DECODER FOR (272, 260)
BURST-ERROR-CORRECTING CODE

FIGURE 6.6-1

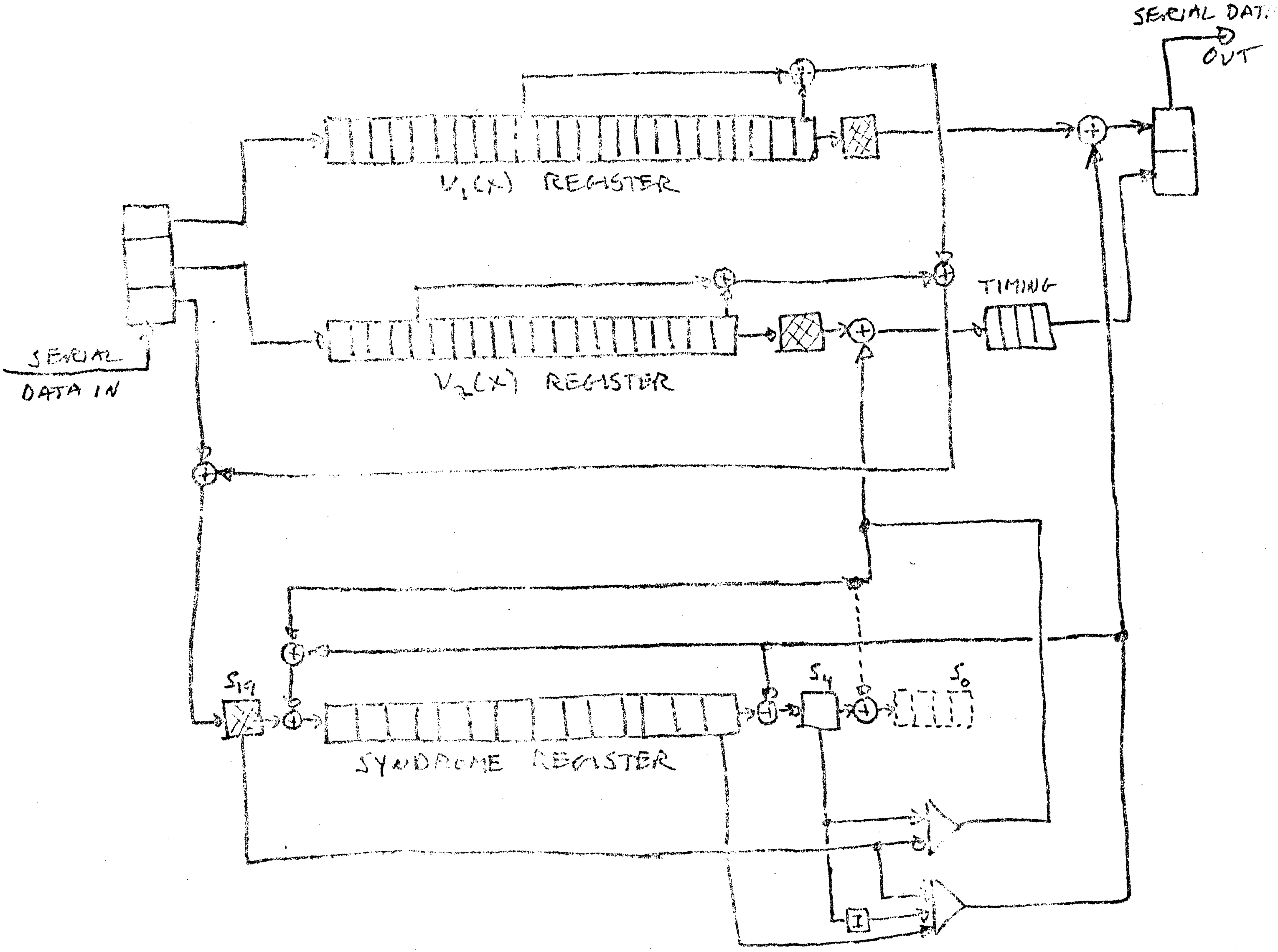
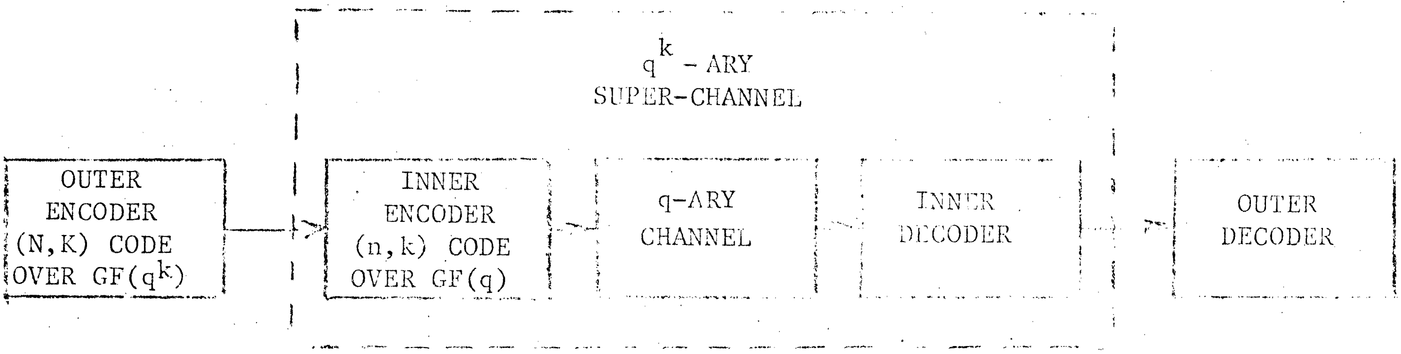


FIGURE 12.7-3

DECODER FOR (72,48) IWADARE CODE



q^k - ARY
SUPER-CHANNEL
OUTER ENCODER (N,K) CODE OVER GF(q^k)
INNER ENCODER (n,k) CODE OVER GF(q)
q-ARY CHANNEL
INNER DECODER
OUTER DECODER

Concatenation

FIGURE 7.8-1

```
ORIGINAL DATA:
 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 1 2
 3 4 5 6 7 6 5 4 3 2 1 0 0 1 2 3 4 5
 6 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6
 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6 5 4 3


OUTER ENCODED DATA: (PARITY BLOCK FIRST 4 COLUMNS)
 0 4 6 5|0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 1 2
 2 4 6 5|3 4 5 6 7 6 5 4 3 2 1 0 0 1 2 3 4 5
 4 0 0 2|6 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6
 2 1 7 2|5 4 3 2 1 0 0 1 2 3 4 5 6 7 6 5 4 3


INNER ENCODED MESSAGE:(PARITY BLOCK TOP 3 ROWS)
 7 7 6 7 4 1 4 5 4 1 0 4 2 5 4 3 6 2 6 4 1 4
 4 4 5 6 0 4 7 2 6 5 2 4 2 1 5 4 3 4 6 0 4 7
 2 2 5 2 4 6 1 7 1 1 2 5 3 0 7 3 3 2 2 4 6 1
 ---------------------------------------------
 0 4 5 5|0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 1 2
 2 4 6 5|3 4 5 6 7 6 5 4 3 2 1 0 0 1 2 3 4 5
 4 0 0 2|6 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6
 2 1 7 2|5 4 3 2 1 0 0 1 2 3 4 5 6 7 6 5 4 3


THE MESSAGE WITH ERRORS AND ERASURES:
 4 7 - 7 4 - 4 5 4 1 - 4 2 1 4 - 6 2 6 4 - 4
 4 3 - - 0 2 7 2 - - 2 4 2 4 5 4 3 4 5 0 4 7
 2 2 - 2 4 4 1 7 1 1 2 5 3 5 7 3 3 - 2 4 6 1
 0 4 - - 1 6 0 - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - - 1 0 0 1 2 2 4 5
 4 0 - 2 6 1 6 5 4 3 2 - - - 4 - 0 - 5 6 - 6
 2 1 5 2 5 2 3 - 1 0 0 1 2 6 4 5 6 7 6 5 4 1

 INNER DECODE

THE CORRECTED MESSAGE IS:
 0 4 - 5 0 - 2 3 4 5 6 7 6 7 4 3 2 1 0 0 1 2
 2 4 - 5 3 - 5 6 7 6 5 4 3 3 1 0 0 1 2 3 4 5
 4 0 - 2 6 - 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6
 2 1 - 2 5 - 3 2 1 0 0 1 2 6 4 5 6 7 6 5 4 3



    OUTER DECODE

   THE RESTORED MESSAGE:
    0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 0 1 2
    3 4 5 6 7 6 5 4 3 2 1 0 0 1 2 3 4 5
    6 7 6 5 4 3 2 1 0 0 1 2 3 4 5 6 7 6
    5 4 3 2 1 0 0 1 2 3 4 5 6 7 6 5 4 3
```

(22,18) RS
$q = 2^{12}$
$= 4096$
$t = 2$
$d = 5$

EACH COLUMN

(7,4) RS
$q = 8$
$d = 4$

Figure 7.8-3

An Octal Concatenation Code

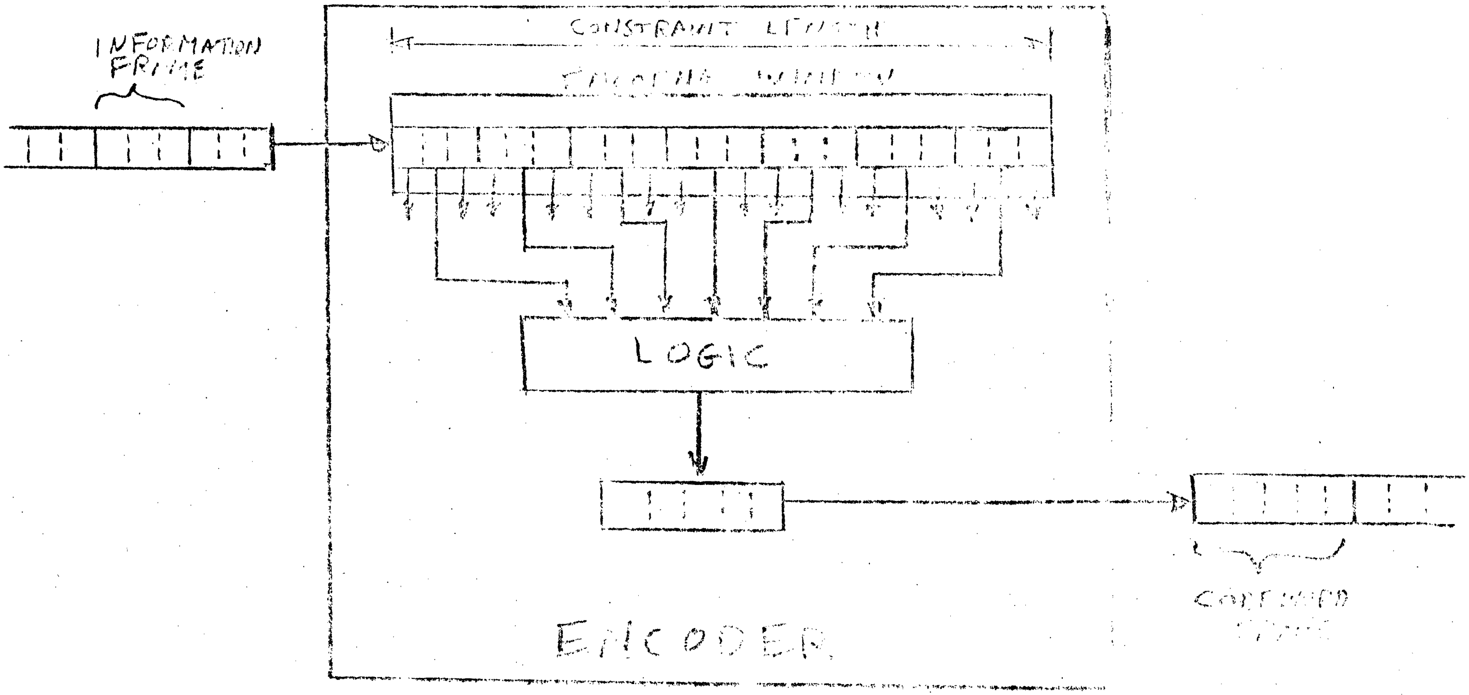

INFORMATION FRAME
CONSTRAINT LENGTH
ENCODING WINDOW
LOGIC
ENCODER

FIGURE 12.1-1

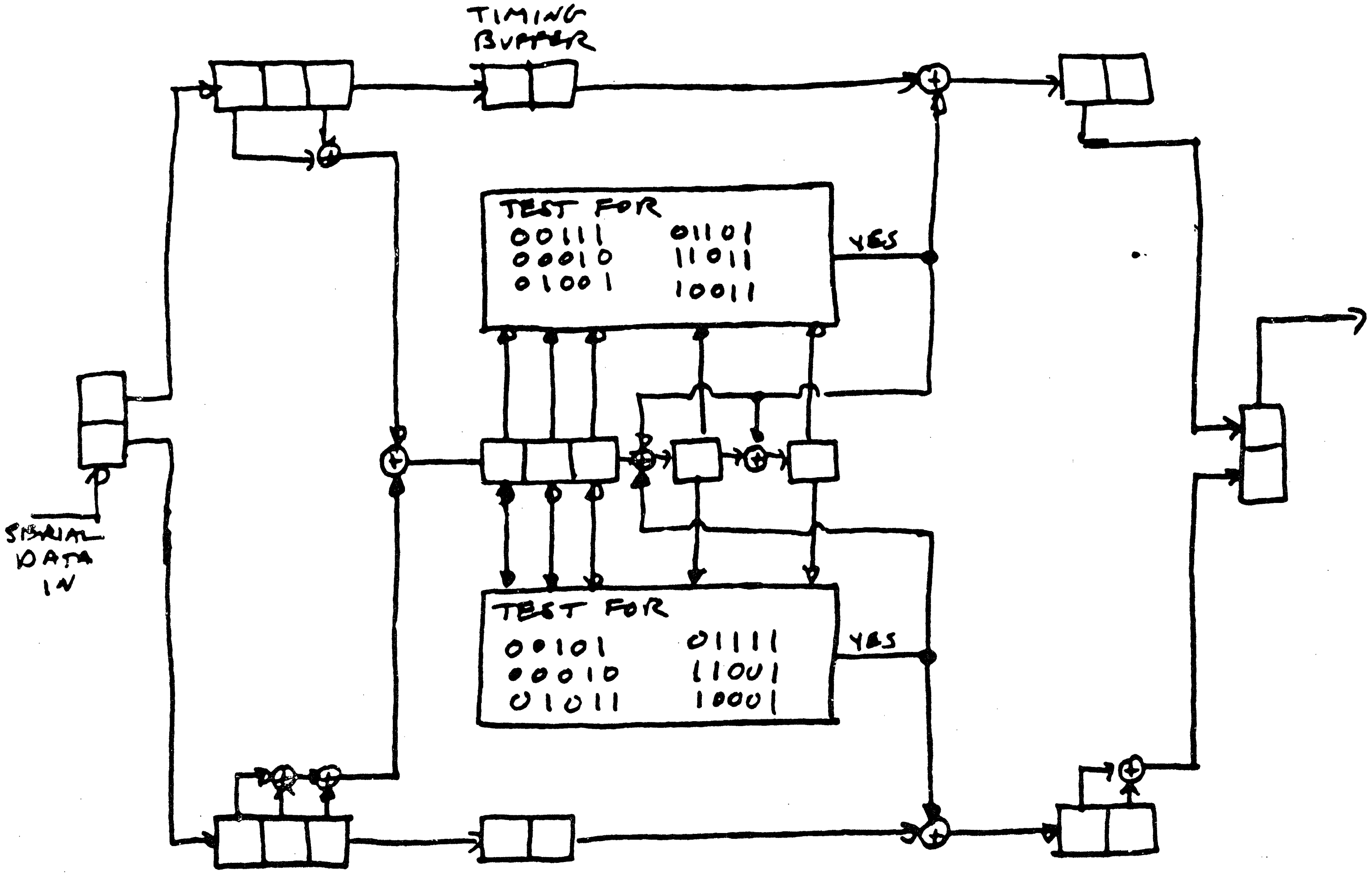
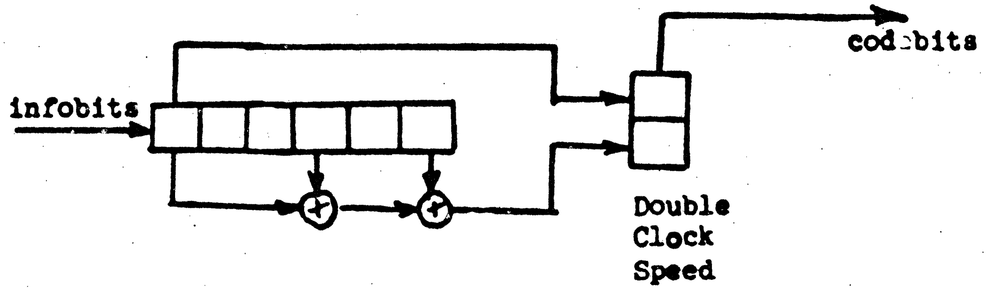
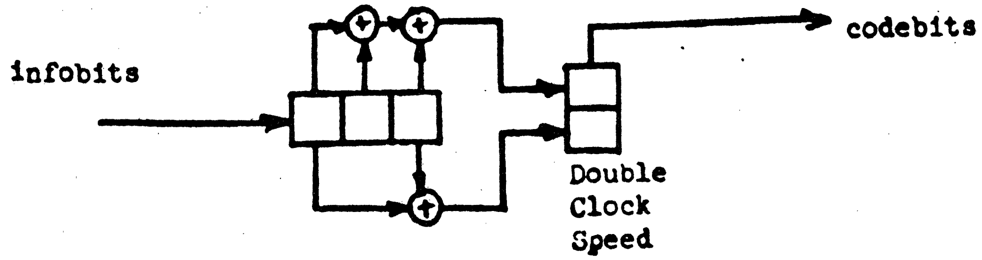


FIGURE 12-6-5

DECODER FOR A (6,3) CODE



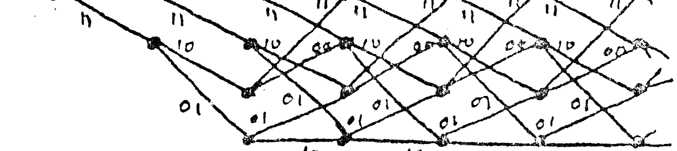
AN ENCODER FOR A BINARY (12,6) CONVOLUTIONAL CODE



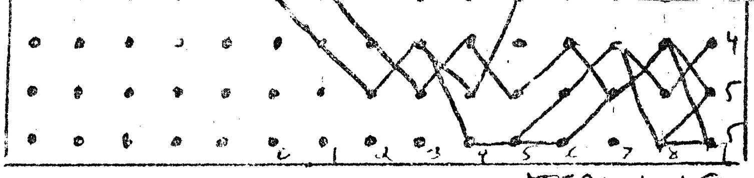
AN ENCODER FOR A BINARY (6,3) CONVOLUTIONAL CODE

FIGURE 12.1-3

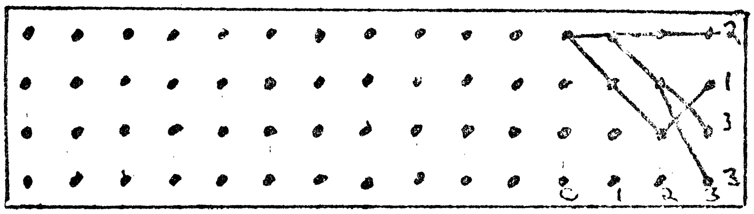
SOME CONVOLUTIONAL ENCODERS



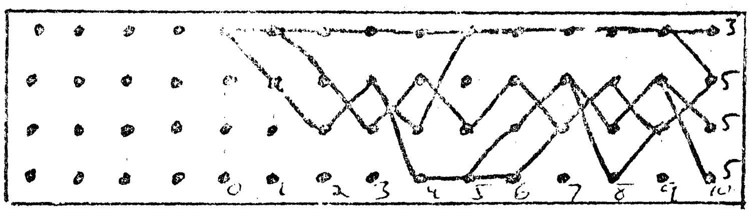
e=10 10 00 00 10 00 00 00 00 00.....



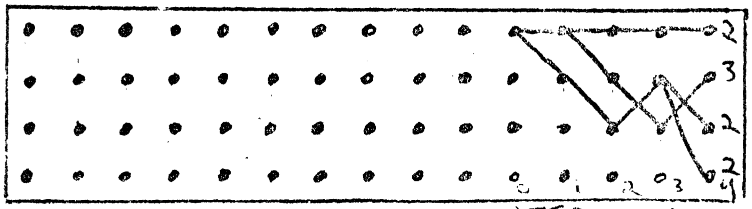
ITERATION 9



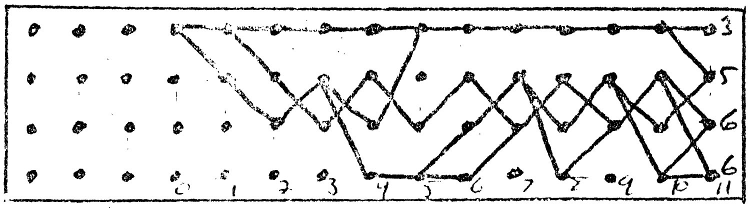
ITERATION 3



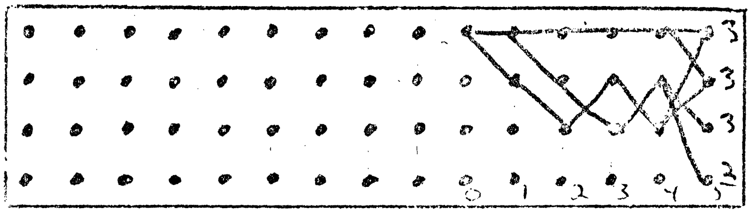
ITERATION 10



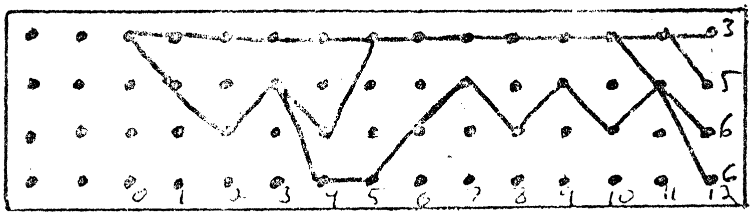
ITERATION 4



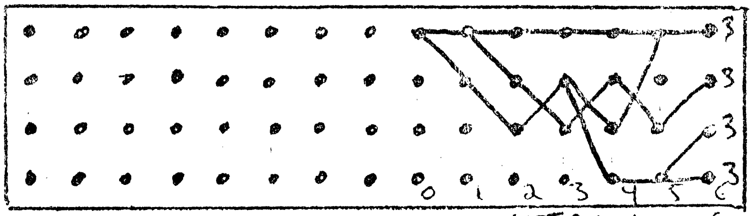
ITERATION 11



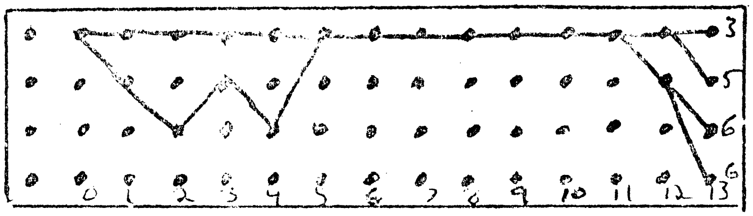
ITERATION 5



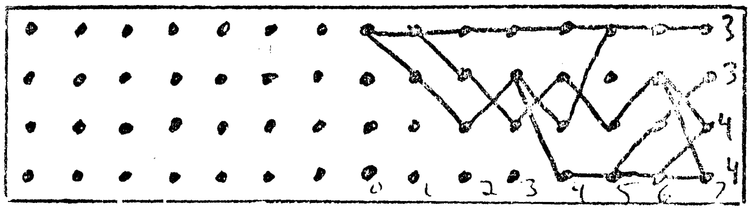
ITERATION 12



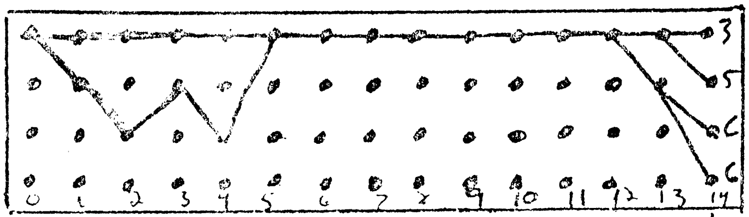
ITERATION 6



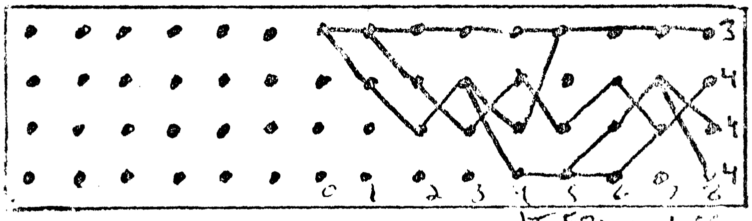
ITERATION 13



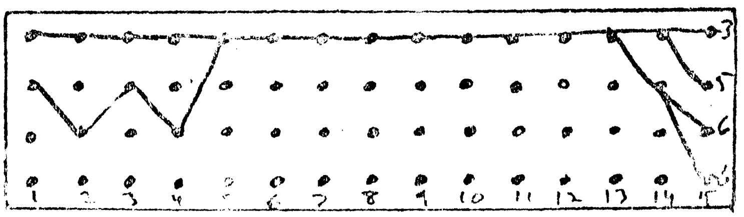
ITERATION 7



ITERATION 14



ITERATION 8



ITERATION 15 :

FIGURE 13.2.3 [illegible]

ECC FOR TAPES AND DISCS -- R. E. Blahut

This presentation is a companion to the following presentation by Steve West. Between the two presentations a proposed universal LSI decoder is described. This decoder will decode any Reed-Solomon code for magnetic tape or disc applications. Very high decoding speed is possible, and any number of errors can be corrected by the decoder. The user need only provide enough redundancy in the recorded message to allow correction of the desired number of errors.

The talk considers the theory upon which the design is based.

UNIVERSAL LSI DECODER FOR BLOCK CODES

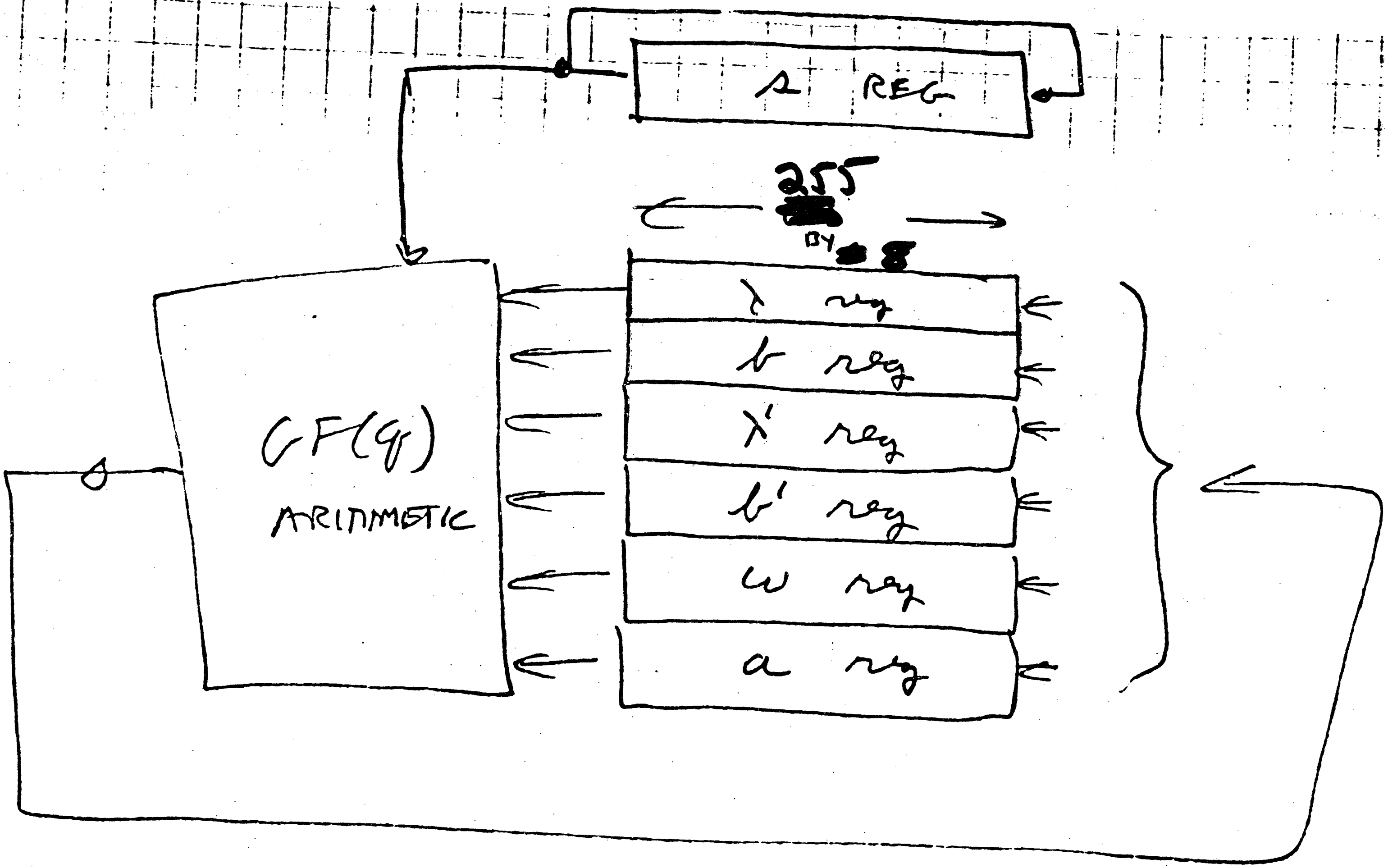
- $n \leq 256$ BYTES
- $q = 2, 4, 8, 16, 32, 64, 128, 256$ (1 TO 8 BIT BYTES)
- BCH & REED-SOLOMON CODES
- ERRORS & ERASURES DECODING
- DECODE TIME $n \times 2t$ CLOCKS
- 7 GATES / CLOCK
- CHIP SIZE = ?

PERFORMANCE PREDICTIONS
AT A 30 μSEC DECODE CYCLE

BLOCKLENGTH (BYTES)	2t	DECODE TIME	DECODE TIME/BIT	# DECODERS FOR 50 MBPS
n = 256	20	133 μsec	.08 μsec	4
n = 86	20	50 μsec	.08 μsec	4
n = 52	20	30 μsec	.08 μsec	4
n = 18	4	2 μsec	.015 μsec	0.8

t = # BYTE ERRORS

2t = # BYTE ERASURES



DECODER

NOW IT GETS HARD

BCH CODES & REED-SOLOMON CODES

- CODES OVER $GF(q)$
- BLOCKLENGTH $n = q^m - 1$ (7, 15, 31, 63, ...)

  IF $m = 1 \Rightarrow$ REED-SOLOMON

  $m \neq 1 \Rightarrow$ BCH

- DEFINE $g(x)$ BY SPECIFYING ROOTS IN $GF(q^m)$
- $g(\alpha^i) = 0$ FOR $i = 1, \ldots, 2t$

  α IS AN ELEMENT OF $GF(q^m)$

EXAMPLE: HAMMING (7,4) OVER GF(2)

$$n = 2^m - 1$$

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

BECAUSE IN $GF(8)$, $g(\alpha) = 0$

$g(\alpha^2) = 0$

SIMPLEST CASE REED-SOLOMON CODES

$$n = q - 1$$

$$k = q - 1 - 2t$$ ← 2 PARITY BYTES PER ERROR CORRECTED

FOURIER TRANSFORMS

- DISCRETE FOURIER TRANSFORM OVER COMPLEX FIELD

$$P_k = \sum_{i=0}^{n-1} e^{-j2\pi \frac{ik}{N}} p_i \qquad k=0,\dots,n-1$$

$$p_i = \sum_{k=0}^{n-1} e^{j2\pi \frac{ik}{N}} P_k$$

- DISCRETE FOURIER TRANSFORM OVER GALOIS FIELD

$$E_j = \sum_{i=0}^{n-1} \alpha^{ij} e_i \qquad j=0,\dots,n-1$$

$$e_i = \frac{1}{(n \bmod p)} \sum_{j=0}^{n-1} \alpha^{-ij} E_j$$

BUT n DIVIDES $q^m - 1$

α HAS ORDER n

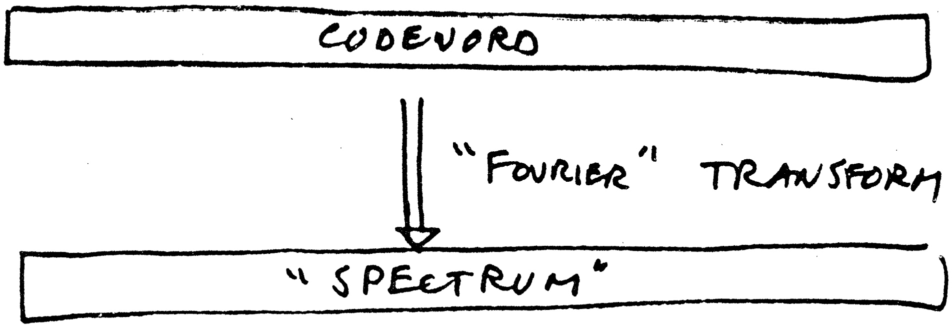
IN GENERAL

$$[GF(q)]^n \xrightarrow{FT} [GF(q^m)]^n$$

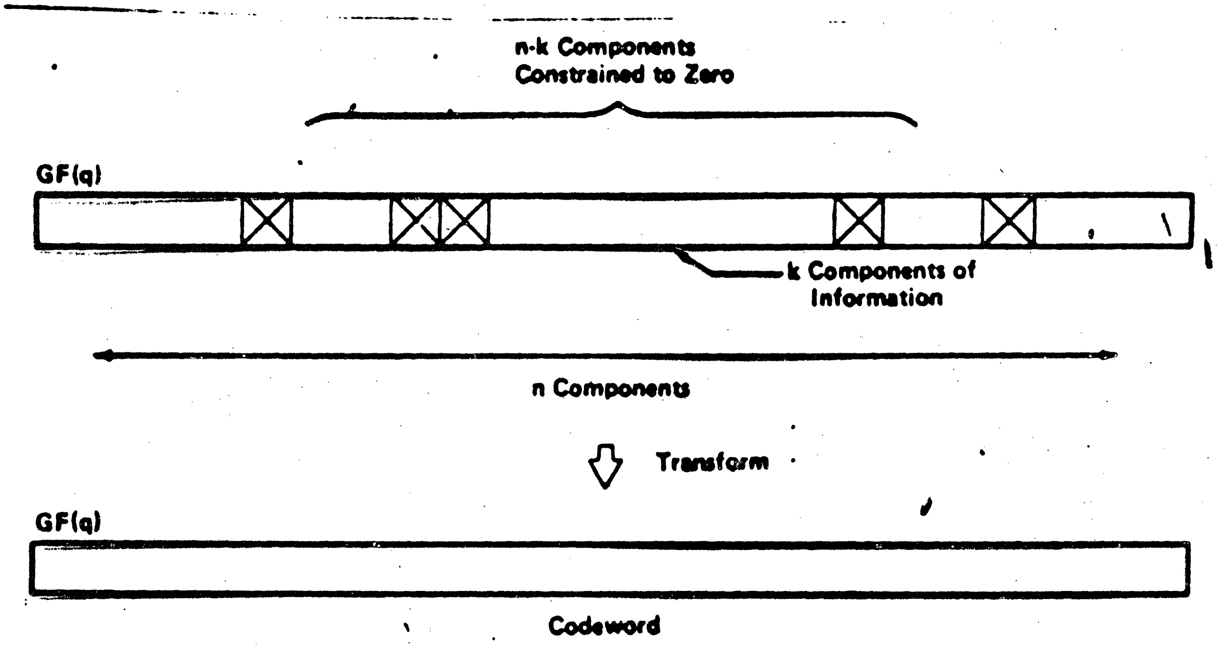
$$a_i b_i \longleftrightarrow A * B$$

FREQUENCY DOMAIN DECODER

- CODES ARE IN $GF(q)$ $\quad n = q^m - 1$
- CODING ALGORITHMS ARE IN $GF(q^m)$
- "FREQUENCY DOMAIN" IS DEFINED BY ANALOGY WITH TRUE FREQUENCY



- "SPECTRUM" IS A SET OF q^m-ARY SYMBOLS
- "SPECTRUM" IS PROCESSED TO FIND ERRORS IN ORIGINAL CODEWORD



Encoding Via the Transform

IF $2t$ FREQUENCIES CONSTRAINED TO ZERO ARE CONTIGUOUS, THEN CODE IS CALLED t-ERROR-CORRECTING BCH CODE

Frequency Domain Codewords							Time Domain Codewords						
c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_0	c_1	c_2	c_3	c_4	c_5	c_6
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	α^0	0	α^0	α^0	1	1	1	0	1	0	0
0	0	0	α^1	0	α^4	α^2	0	0	1	1	1	0	1
0	0	0	α^2	0	α^1	α^4	0	1	0	0	1	1	1
0	0	0	α^3	0	α^5	α^6	1	1	0	1	0	0	1
0	0	0	α^4	0	α^2	α^1	0	1	1	1	0	1	0
0	0	0	α^5	0	α^6	α^3	1	0	0	1	1	1	0
0	0	0	α^6	0	α^3	α^5	1	0	1	0	0	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	α^0	0	α^0	α^0	0	0	0	1	0	1	1
1	0	0	α^1	0	α^4	α^2	1	1	0	0	0	1	0
1	0	0	α^2	0	α^1	α^4	1	0	1	1	0	0	0
1	0	0	α^3	0	α^5	α^6	0	0	1	0	1	1	0
1	0	0	α^4	0	α^2	α^1	1	0	0	0	1	0	1
1	0	0	α^5	0	α^6	α^3	0	1	1	0	0	0	1
1	0	0	α^6	0	α^3	α^5	0	1	0	1	1	0	0

Hamming (7,4) Code

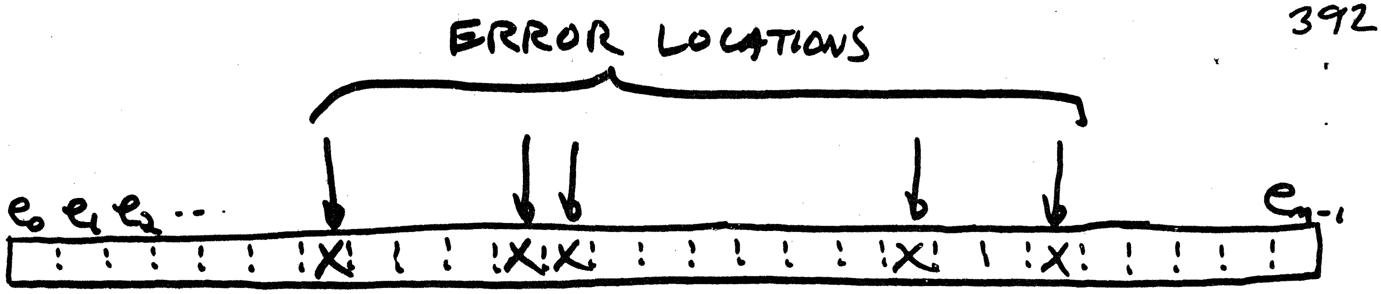
F.T.

0	0	0	0	0	1	1
0	0	0	1	0	1	0
1	0	0	0	0	0	0

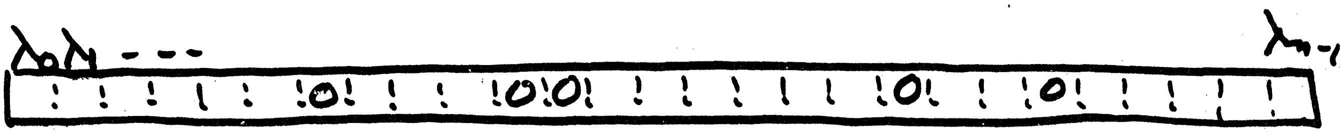
[illegible]

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	1

$\underline{e}$



$\underline{\lambda}$



TO GET $\underline{\lambda}$, DEFINE $\Lambda(x) = \prod_{k=1}^{\nu} (1 - x\alpha^{i_k})$

THEN

$$\lambda_i = \sum_{j=0}^{n-1} \alpha^{-ij} \Lambda_j$$

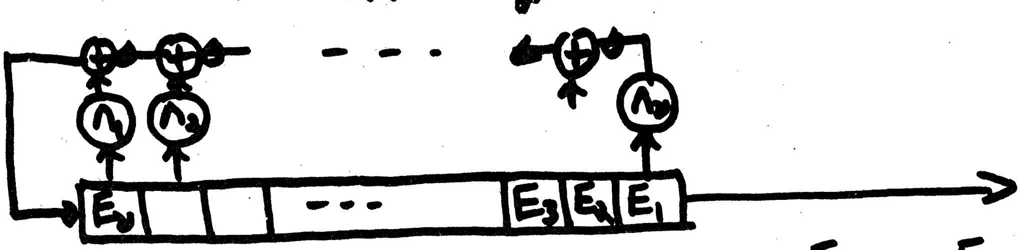
GIVES DESIRED ZEROS

AND

$$\text{degree}[\Lambda(x)] = \nu \leq t$$

$$\lambda_i e_i = 0 \quad \Rightarrow \quad \underline{\Lambda} * \underline{E} = 0$$

$$E_k = -\sum_{j=1}^{\nu} \Lambda_j E_{k-j}$$

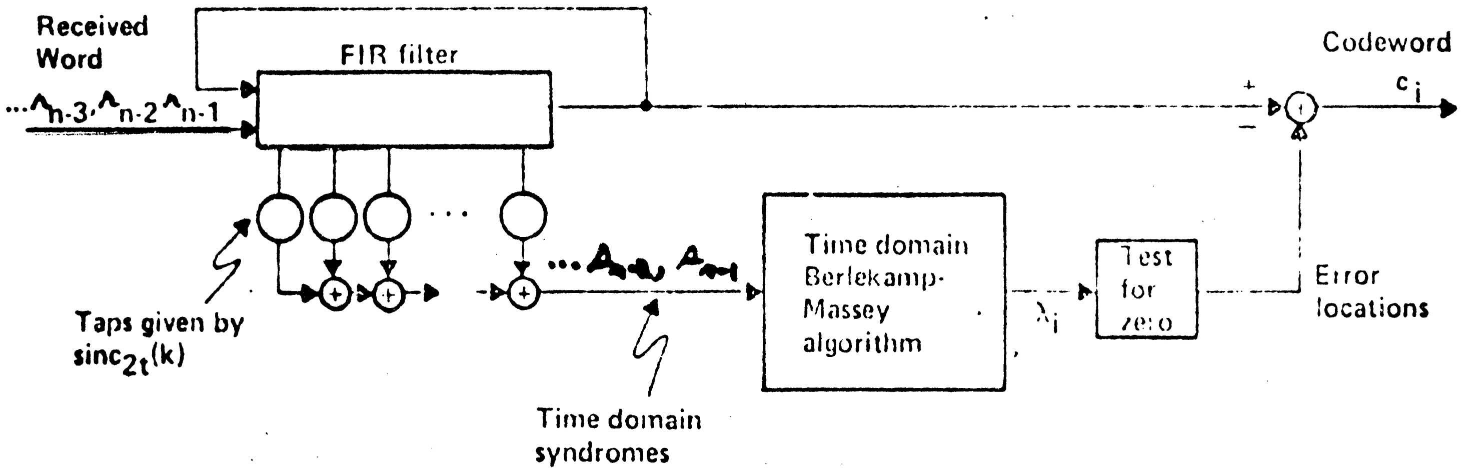


$\nu \leq t$

$E_n, \ldots, E_{\nu+1}, E_\nu, \ldots, E_2, E_1$

$$\left.\begin{aligned} E_{\nu+1} &= -\Lambda_1 E_\nu - \Lambda_2 E_{\nu-1} - \cdots - \Lambda_\nu E_1 \\ &\vdots \\ E_{2\nu} &= -\Lambda_1 E_{2\nu-1} - \Lambda_2 E_{2\nu-2} - \cdots - \Lambda_\nu E_\nu \end{aligned}\right\} \nu \text{ EQUATIONS}$$

$$E_{2\nu+1} = -\Lambda_1 E_{2\nu} - \Lambda_2 E_{2\nu-1} - \cdots - \Lambda_\nu E_{\nu+1}$$



Time Domain Decoder For Binary BCH Codes

$$S_j = R_j \qquad j = 1, \ldots, 2t$$

$$S_j = \text{RECT}_j R_j \qquad \text{RECT}_j = \begin{cases} 1 & j = 1, \ldots, 2t \\ 0 & \text{OTHERWISE} \end{cases}$$

$$s_i = \text{SINC}_{2t}(i) * r_i$$

BERLEKAMP-MASSEY ITERATION

$$\Delta_r = \sum_{j=0}^{n-1} \Lambda_j^{(r-1)} S_{r-j}$$

$$\begin{vmatrix} \Lambda^{(r)}(x) \\ B^{(r)}(x) \end{vmatrix} = \begin{vmatrix} 1 & -\Delta_r x \\ \Delta_r^{-1} \delta_r x & \bar{\delta}_r x \end{vmatrix} \begin{vmatrix} \Lambda^{(r-1)}(x) \\ B^{(r-1)}(x) \end{vmatrix}$$

TIME DOMAIN BERLEKAMP-MASSEY ITERATION

$$\Delta_r = \sum_{i=0}^{n-1} \alpha^{ir} \lambda_i^{(r)} s_i$$

$$\begin{vmatrix} \lambda_i^{(r)} \\ b_i^{(r)} \end{vmatrix} = \begin{vmatrix} 1 & -\Delta_r \alpha^{-i} \\ \Delta_r^{-1} \delta_r \alpha^{-i} & \bar{\delta}_r \alpha^{-i} \end{vmatrix} \begin{vmatrix} \lambda_i^{(r-1)} \\ b_i^{(r-1)} \end{vmatrix}$$

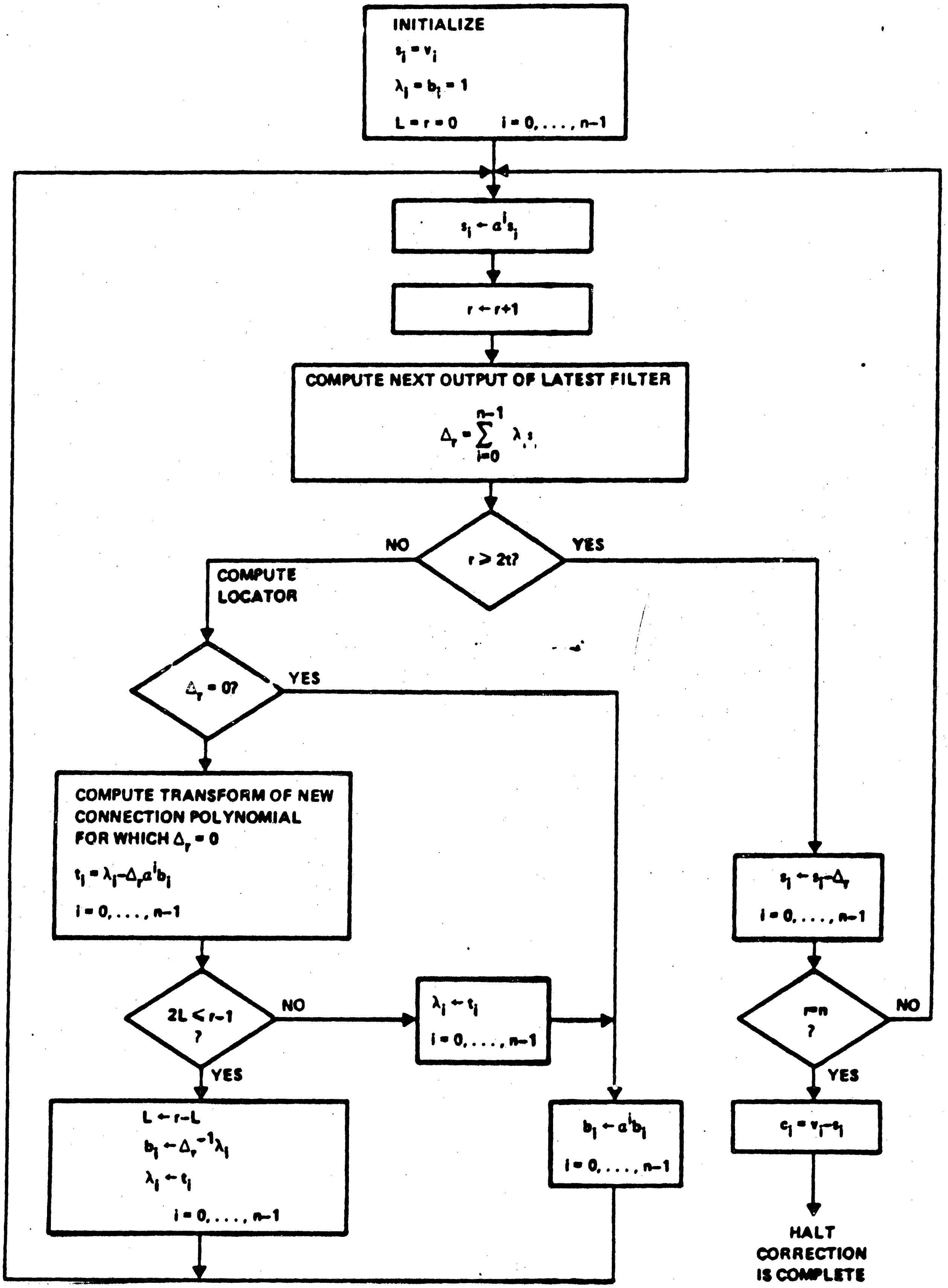

INITIALIZE
$s_i = v_i$
$\lambda_i = b_i = 1$
$L = r = 0 \quad i = 0, \ldots, n-1$
$s_i \leftarrow \alpha^i s_i$
$r \leftarrow r+1$
COMPUTE NEXT OUTPUT OF LATEST FILTER
$\Delta_r = \sum_{i=0}^{n-1} \lambda_i s_i$
NO
$r > 2t?$
YES
COMPUTE LOCATOR
$\Delta_r = 0?$
YES
COMPUTE TRANSFORM OF NEW CONNECTION POLYNOMIAL FOR WHICH $\Delta_r = 0$
$t_i = \lambda_i - \Delta_r \alpha^i b_i$
$i = 0, \ldots, n-1$
$2L < r-1$?
NO
YES
$\lambda_i \leftarrow t_i$
$i = 0, \ldots, n-1$
$L \leftarrow r-L$
$b_i \leftarrow \Delta_r^{-1} \lambda_i$
$\lambda_i \leftarrow t_i$
$i = 0, \ldots, n-1$
$b_i \leftarrow \alpha^i b_i$
$i = 0, \ldots, n-1$
$s_i \leftarrow s_i - \Delta_r$
$i = 0, \ldots, n-1$
$r = n$?
NO
YES
$c_i = v_i - s_i$
HALT CORRECTION IS COMPLETE

Figure 9.5-3 Time Domain Decoder

Reed-Solomon Time Domain Decoder Architecture and Galois Field Multipliers

Stephen C. West

GPD Tucson

There is an Advanced Technology effort underway at GPD Tucson to investigate the implementation and performance of a Reed-Solomon error control decoder. This presentation reports on two aspects of the investigation, the decoder architectures being investigated and a recently developed Galois Field multiplier.

Reed-Solomon Decoder Architectures

One of the primary objectives of the current effort is to develop a general purpose Reed-Solomon decoder. The motivation for this objective is two fold: first, to allow selection of the precise code parameters to occur as late as possible in the product development cycle, and second, to support a range of products (low, medium and high end products) within a generation of products.

We are currently trying to put a plan in place to design and build a VTL prototype decoder to be followed by an LSI version.

The significant features being considered for the VTL decoder are listed in Figure 1.

There are four architectures being examined, all of which are based on two decoding algorithms developed by R. E. Blahut at FSD Owego. The four architectures are summarized in Figure 2. Architectures III and IV are particularly interesting, if multiple LSI ships are required to implement the decoder. A properly partitioned decoder would allow an application, needing only N^2 speed, to purchase and implement a lower cost decoder than an application which required N x 2t speed.

Figures 3 thru 9 illustrate the algorithms and block diagrams for the four architectures. The algorithm of Figure 3 (Architecture I) is a subset of the algorithm of Figure 5 (Architectures II and III) with the exception of a slight change in the YES branch of the $R > 2t$ check (Architecture II). The algorithm of Architecture III is expanded, over that shown in Figure 5, to include the operations in the YES branch of the $R > 2t$ check as a selectable mode of operation.

The algorithm for Architecture IV (Figure 8) differs from the Architecture I algorithm only in the assignment of the variable used to accumulate the α^i multiplications (S_i versus λ_i). Architecture IV has organized the N^2 decoder into two processors:

- o Core Processor - N x 2t clocks per block decode,
- o Auxillary Processor - N x (N - 2t) clocks per block decode.

By replicating and multiplexing the Auxillary Processors, the resultant decoder can achieve the N x 2t speed of Architectures II and III. The Core Processor is also a stand alone N^2 decoder.

Galois Field (GF) Multipliers

One of the more critical elements of the decoder, in terms of delay and chip real estate, is the GF multiplier. A GF multiplier design which has a low circuit count and delay was recently developed at GPD Tucson. A design procedure and implementation for this multiplier, called the Direct GF Multiplier in this presentation, are given in Figures 11 thru 18.

Figure 10 compares a number of different implementations of GF multipliers. The top two configurations use log and anti-log, to the base alpha, look up tables. The second configuration requires the MOD 255 Adder for each multiplier element within the decoder plus a complex GF adder for each adder element within the decoder.

The apparent choice for an 8-bit LSI decoder implementation would be the Direct GF Multiplier, which has both low circuit count and low delay. A procecure for generating the equations implemented by the Direct GF Multiplier is given in Figure 11 and a GF(8) example of the procedure is given in Figure 12.

Figure 13 lists the equations for GF(2) thru GF(128) multipliers, and Figure 14 lists the equations for the GF(256) multiplier.

Figure 15 shows the block diagram of the Direct GF Multiplier. The three basic operations (blocks) in the multiplier are:

- "AND" Product Array - "AND" each bit of one input with each bit of the second input
- Product Compression - generation of the Y variables shown in Figure 14
- Final Selection - collecting the Y's into the final output equations.

Figures 16, 17 and 18 illustrate the implementations of the three blocks of Figure 15. This particular implementation employs 64 2-input "AND" circuits and 76 2-input "XOR" circuits.

1/8/81
SCW

- ERRORS AND ERASURES (POINTERS) DECODING
- VARIABLE BLOCK LENGTH
- ONE SYMBOL EXTENSION BEYOND THE PRIMITIVE BLOCK LENGTH
- VARIABLE REDUNDANCY
- VARIABLE BITS PER SYMBOL
- EXTENDABLE ARCHITECTURE (FROM N^2 CLOCKS PER BLOCK DECODE TO $N \times 2t$ CLOCKS PER BLOCK DECODE)
- ARCHITECTED FOR LSI IMPLEMENTATION
- PIPELINING FOR SPEED
- CODE POINTER GENERATION

FEATURES BEING CONSIDERED FOR THE RS DECODER

FIGURE 1

1/8/81
SCW

IBM CONFIDENTIAL

ARCHITECTURE	SPEED	HARDWARE*	REMARKS
I	N^2	1	
II	$N \times 2t$	2^{+}	
III	$N \times 2t$	2^{++}	ARCHITECTURE I IS A SUBSET
IV	$N \times 2t \leq \left\| \overline{\underline{N \times \frac{N}{I}}} \right\| \leq N^2$	$1 + 1^{--} \times I$	AUXILLARY PROCESSORS ARE VERY SIMPLE. I IS NUMBER OF AUXILLARY PROCESSORS. ARCHITECTURE I IS A SUBSET.

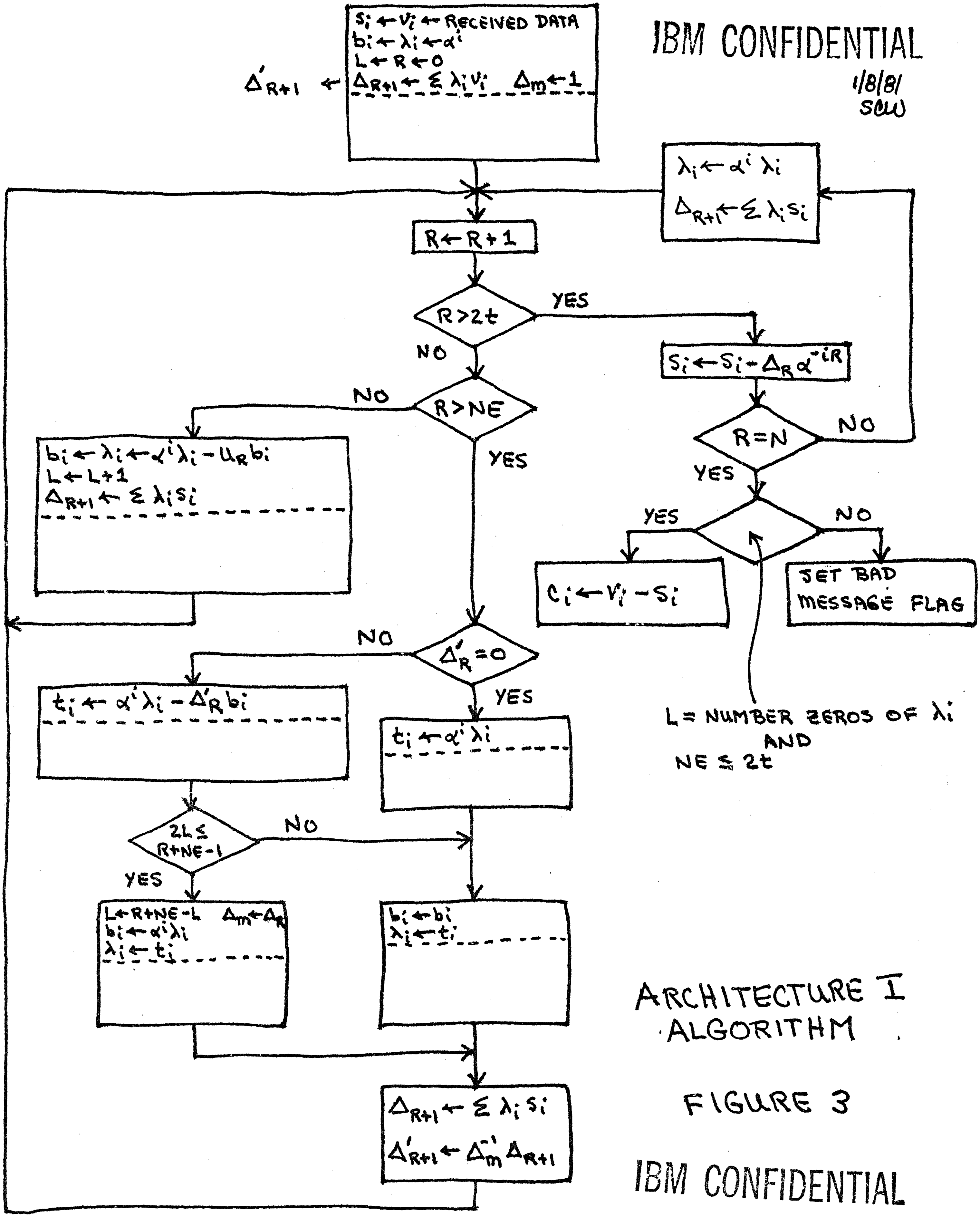
* ARCHITECTURE I IS THE BASE FOR THE HARDWARE COMPARISON. 2^{+} IMPLIES ARCHITECTURE II REQUIRES SOMETHING MORE THAN TWICE THE HARDWARE FOR ARCHITECTURE I.

ARCHITECTURE SUMMARY

FIGURE 2

IBM CONFIDENTIAL

1/8/81
SCW


$s_i \leftarrow v_i \leftarrow$ RECEIVED DATA
$b_i \leftarrow \lambda_i \leftarrow \alpha^i$
$L \leftarrow R \leftarrow 0$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i v_i$ $\Delta_m \leftarrow 1$
$\Delta'_{R+1} \leftarrow$
$\lambda_i \leftarrow \alpha^i \lambda_i$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$R \leftarrow R+1$
$R > 2t$
YES
NO
$s_i \leftarrow s_i - \Delta_R \alpha^{-iR}$
NO
$R > NE$
$R = N$
NO
YES
YES
$b_i \leftarrow \lambda_i \leftarrow \alpha^i \lambda_i - u_R b_i$
$L \leftarrow L+1$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
YES
NO
$c_i \leftarrow v_i - s_i$
SET BAD MESSAGE FLAG
NO
$\Delta'_R = 0$
YES
$t_i \leftarrow \alpha^i \lambda_i - \Delta'_R b_i$
$t_i \leftarrow \alpha^i \lambda_i$
L = NUMBER ZEROS OF λ_i AND NE $\le$ 2t
$2L \le R+NE-1$
NO
YES
$L \leftarrow R+NE-L$ $\Delta_m \leftarrow \Delta_R$
$b_i \leftarrow \alpha^i \lambda_i$
$\lambda_i \leftarrow t_i$
$b_i \leftarrow b_i$
$\lambda_i \leftarrow t_i$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$\Delta'_{R+1} \leftarrow \Delta_m^{-1} \Delta_{R+1}$

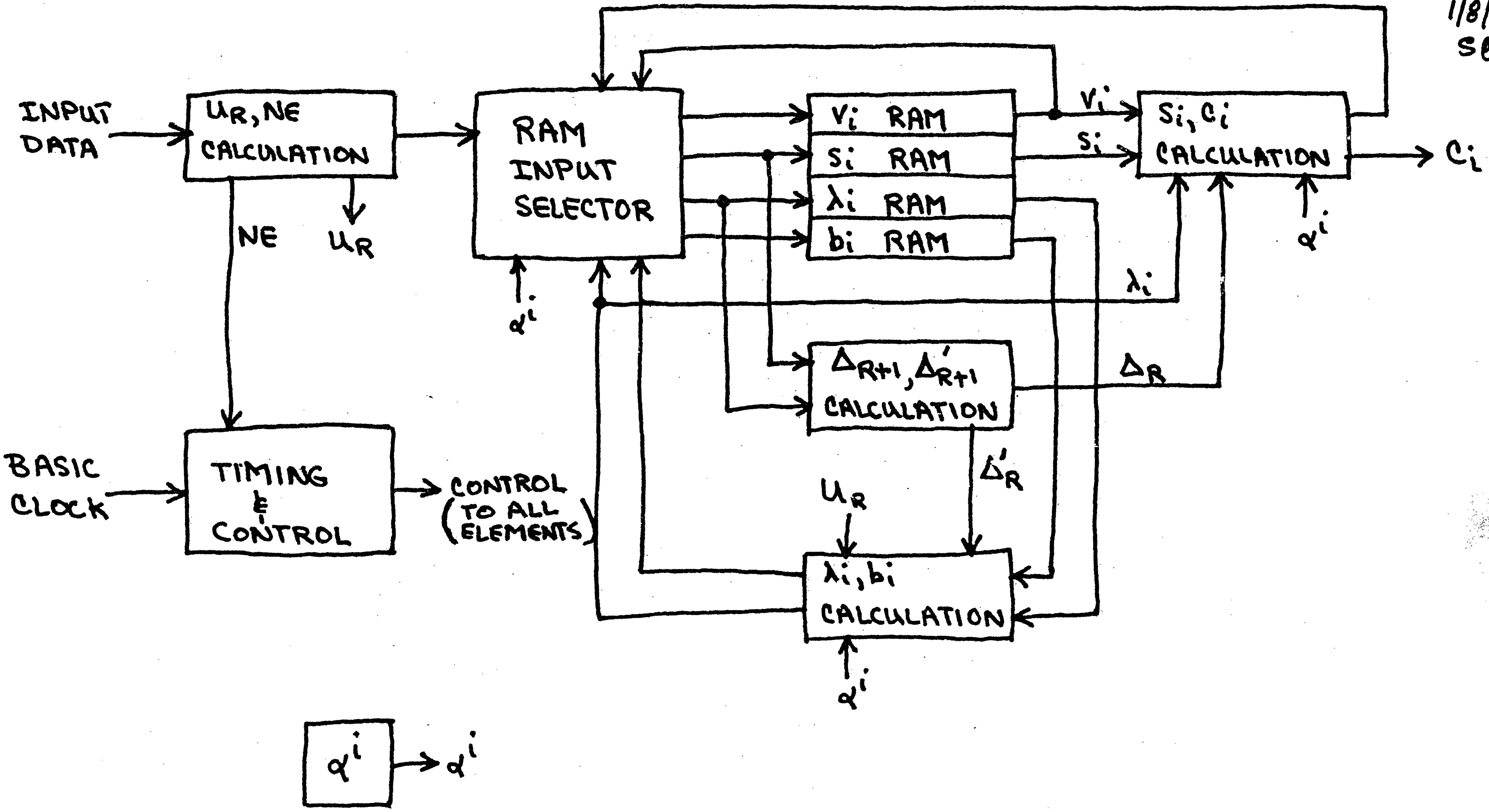
ARCHITECTURE I
ALGORITHM

FIGURE 3

IBM CONFIDENTIAL

IBM CONFIDENTIAL

1/8/81
sew

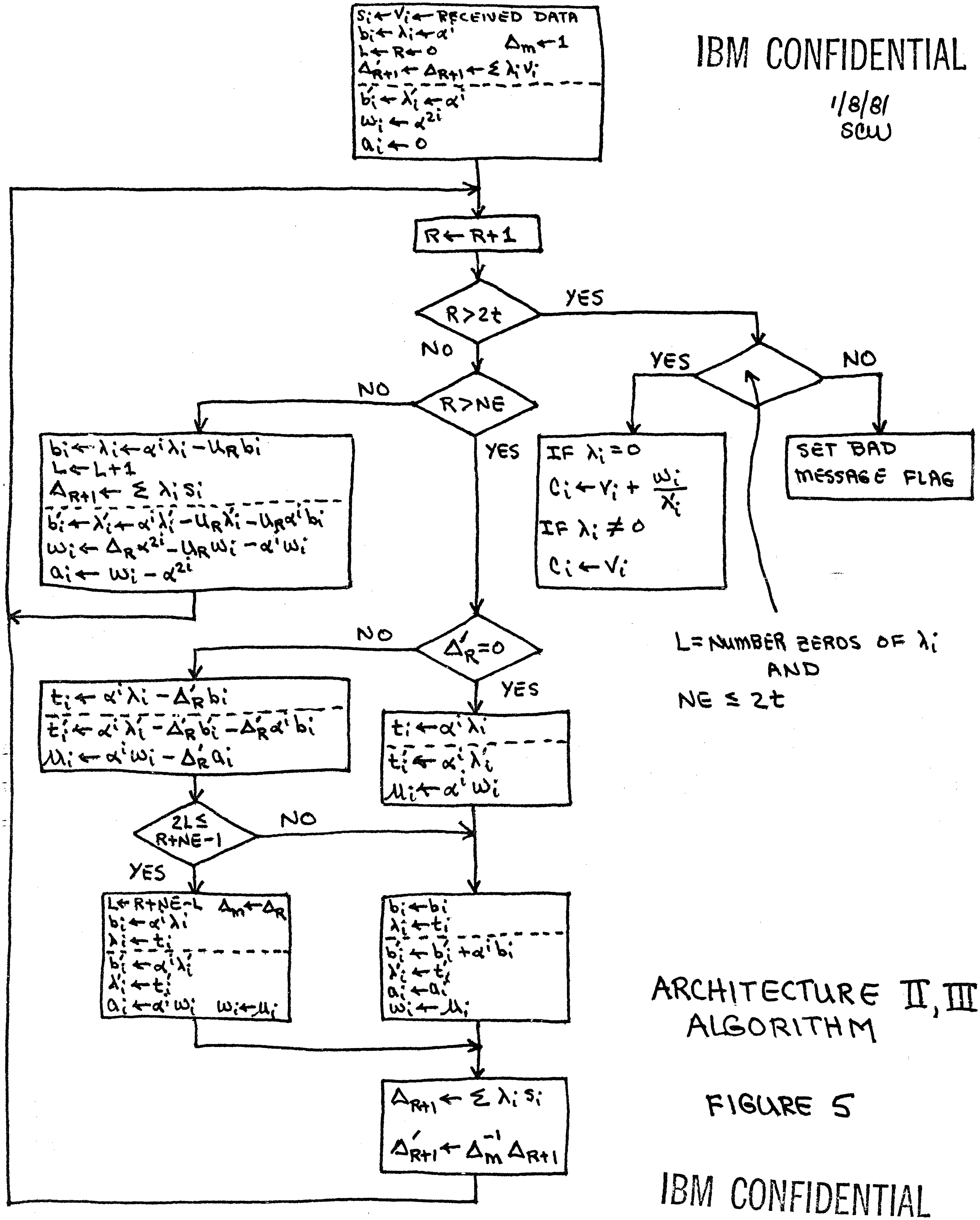


ARCHITECTURE I
BLOCK DIAGRAM

FIGURE 4

IBM CONFIDENTIAL

1/8/81
SCW

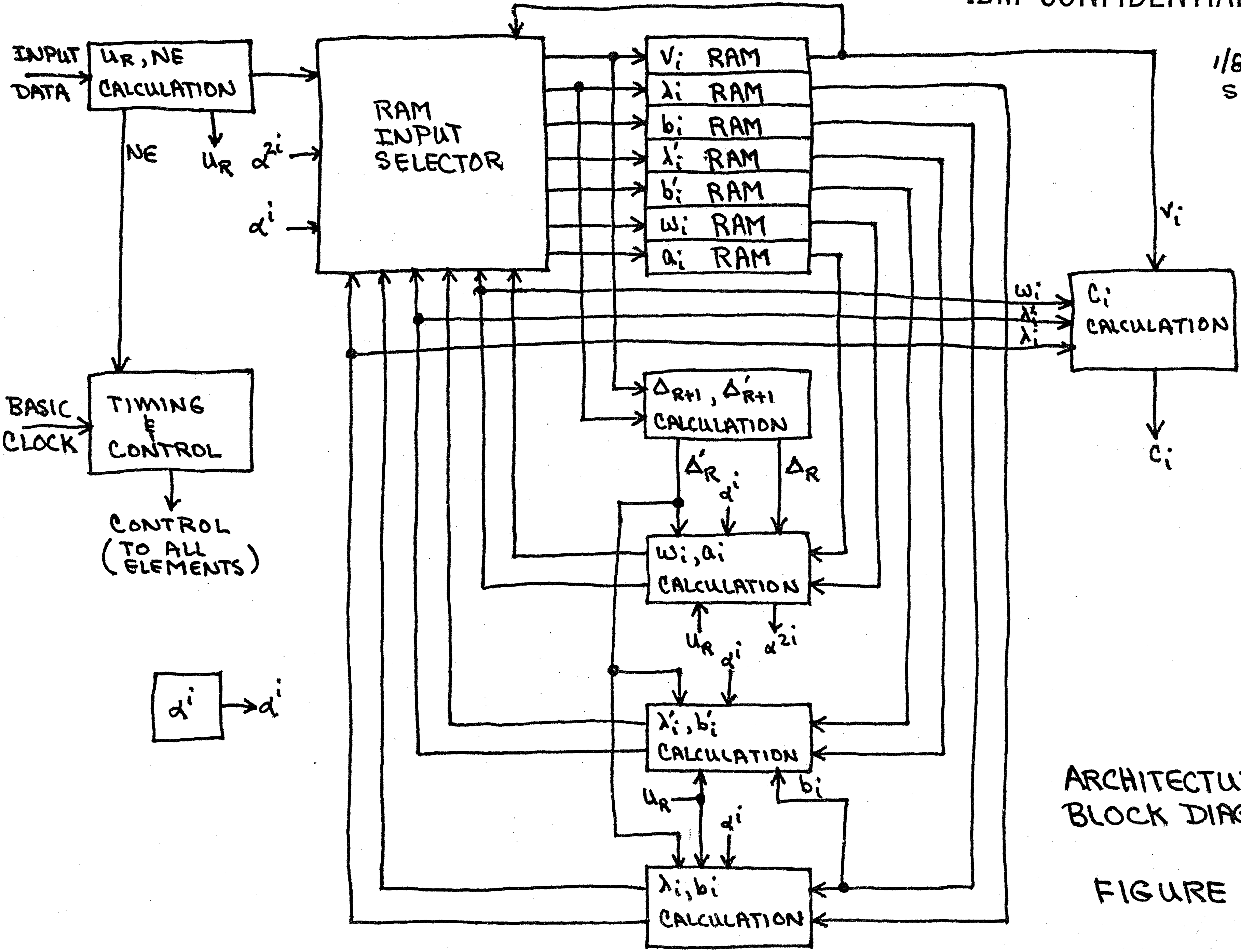

$s_i \leftarrow v_i \leftarrow$ RECEIVED DATA
$b_i \leftarrow \lambda_i \leftarrow \alpha^i$
$L \leftarrow R \leftarrow 0$
$\Delta_m \leftarrow 1$
$\Delta'_{R+1} \leftarrow \Delta_{R+1} \leftarrow \Sigma \lambda_i v_i$
$b'_i \leftarrow \lambda'_i \leftarrow \alpha^i$
$\omega_i \leftarrow \alpha^{2i}$
$a_i \leftarrow 0$
$R \leftarrow R+1$
$R>2t$
YES
NO
YES
NO
$R>NE$
NO
YES
$b_i \leftarrow \lambda_i \leftarrow \alpha^i \lambda_i - U_R b_i$
$L \leftarrow L+1$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$b'_i \leftarrow \lambda'_i \leftarrow \alpha^i \lambda'_i - U_R \lambda'_i - U_R \alpha^i b_i$
$\omega_i \leftarrow \Delta_R \alpha^{2i} - U_R \omega_i - \alpha^i \omega_i$
$a_i \leftarrow \omega_i - \alpha^{2i}$
IF $\lambda_i = 0$
$C_i \leftarrow v_i + \frac{\omega_i}{\lambda'_i}$
IF $\lambda_i \neq 0$
$C_i \leftarrow v_i$
SET BAD MESSAGE FLAG
L=NUMBER ZEROS OF λ_i AND
NE ≤ 2t
$\Delta'_R = 0$
NO
YES
$t_i \leftarrow \alpha^i \lambda_i - \Delta'_R b_i$
$t'_i \leftarrow \alpha^i \lambda_i - \Delta'_R b'_i - \Delta'_R \alpha^i b_i$
$\mu_i \leftarrow \alpha^i \omega_i - \Delta'_R a_i$
$t_i \leftarrow \alpha^i \lambda_i$
$t'_i \leftarrow \alpha^i \lambda'_i$
$\mu_i \leftarrow \alpha^i \omega_i$
$2L \leq R+NE-1$
NO
YES
$L \leftarrow R+NE-L$ $\Delta_m \leftarrow \Delta_R$
$b_i \leftarrow \alpha^i \lambda_i$
$\lambda_i \leftarrow t_i$
$b'_i \leftarrow \alpha^i \lambda'_i$
$\lambda'_i \leftarrow t'_i$
$a_i \leftarrow \alpha^i \omega_i$ $\omega_i \leftarrow \mu_i$
$b_i \leftarrow b_i$
$\lambda_i \leftarrow t_i$
$b'_i \leftarrow b'_i + \alpha^i b_i$
$\lambda'_i \leftarrow t'_i$
$a_i \leftarrow a_i$
$\omega_i \leftarrow \mu_i$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$\Delta'_{R+1} \leftarrow \Delta_m^{-1} \Delta_{R+1}$

ARCHITECTURE II, III
ALGORITHM

FIGURE 5

IBM CONFIDENTIAL

1/8/81
sew



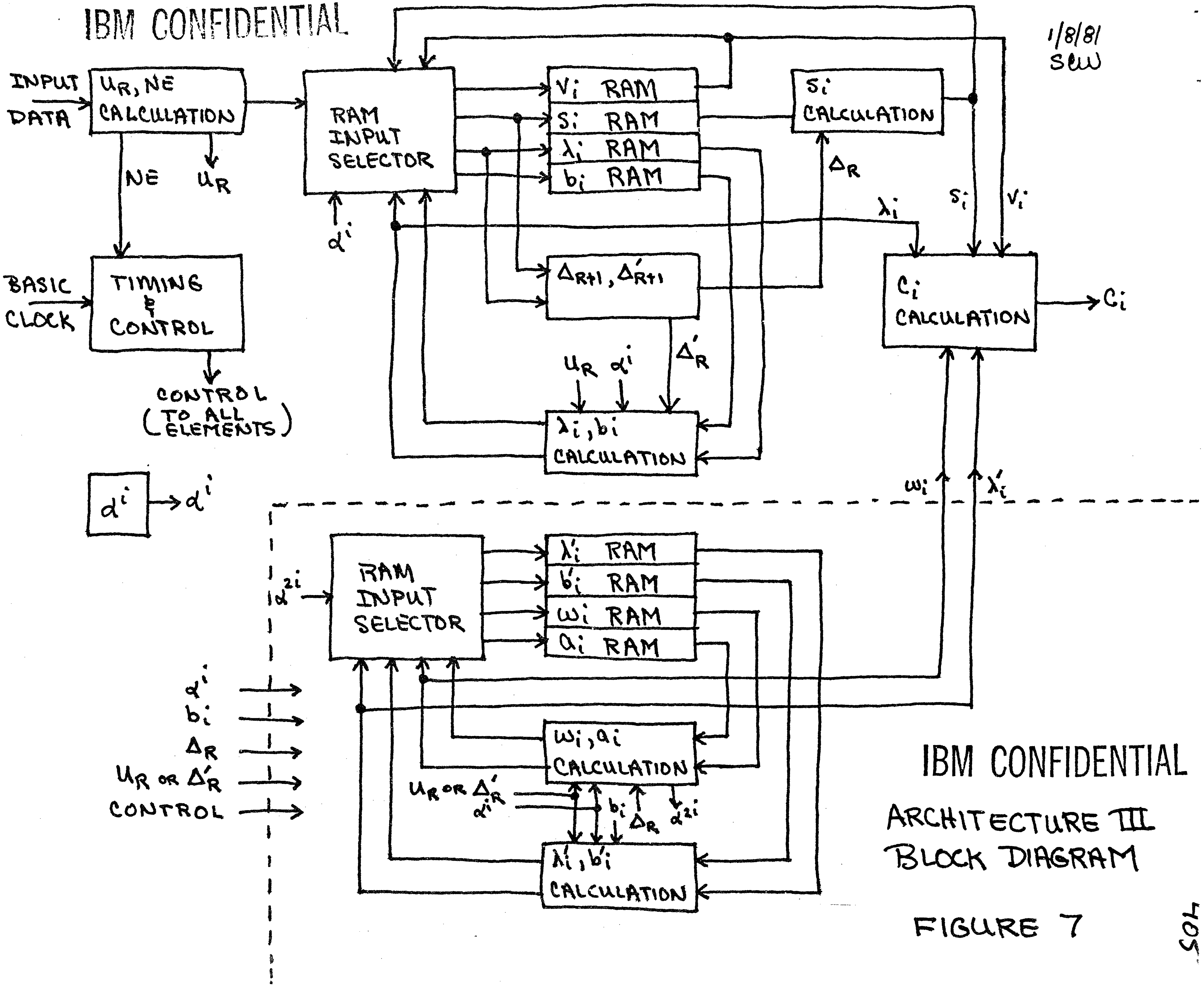
ARCHITECTURE II
BLOCK DIAGRAM

FIGURE 6

IBM CONFIDENTIAL

IBM CONFIDENTIAL

1/8/81
SEW



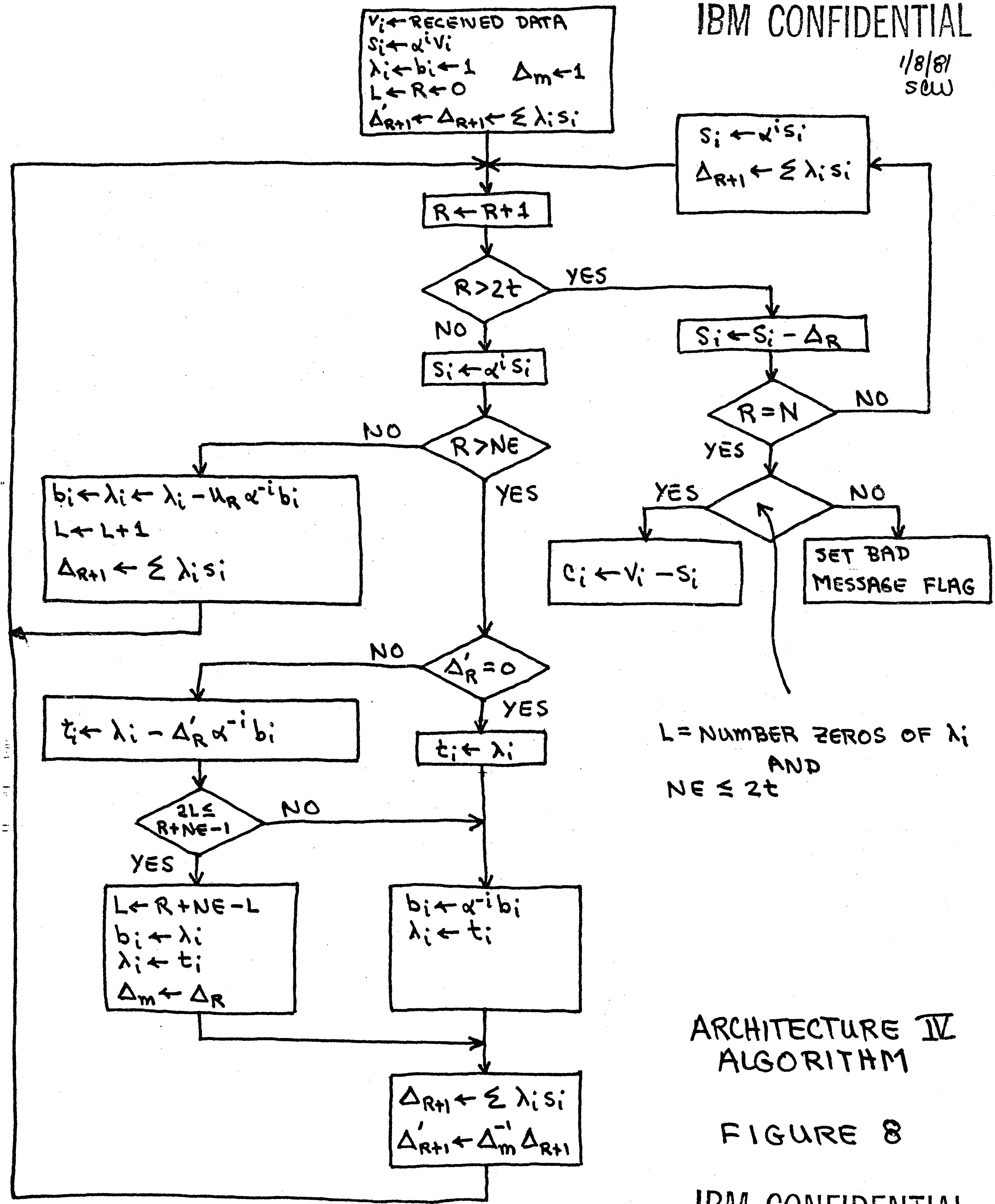
IBM CONFIDENTIAL

ARCHITECTURE III BLOCK DIAGRAM

FIGURE 7

IBM CONFIDENTIAL

1/8/81
SCW

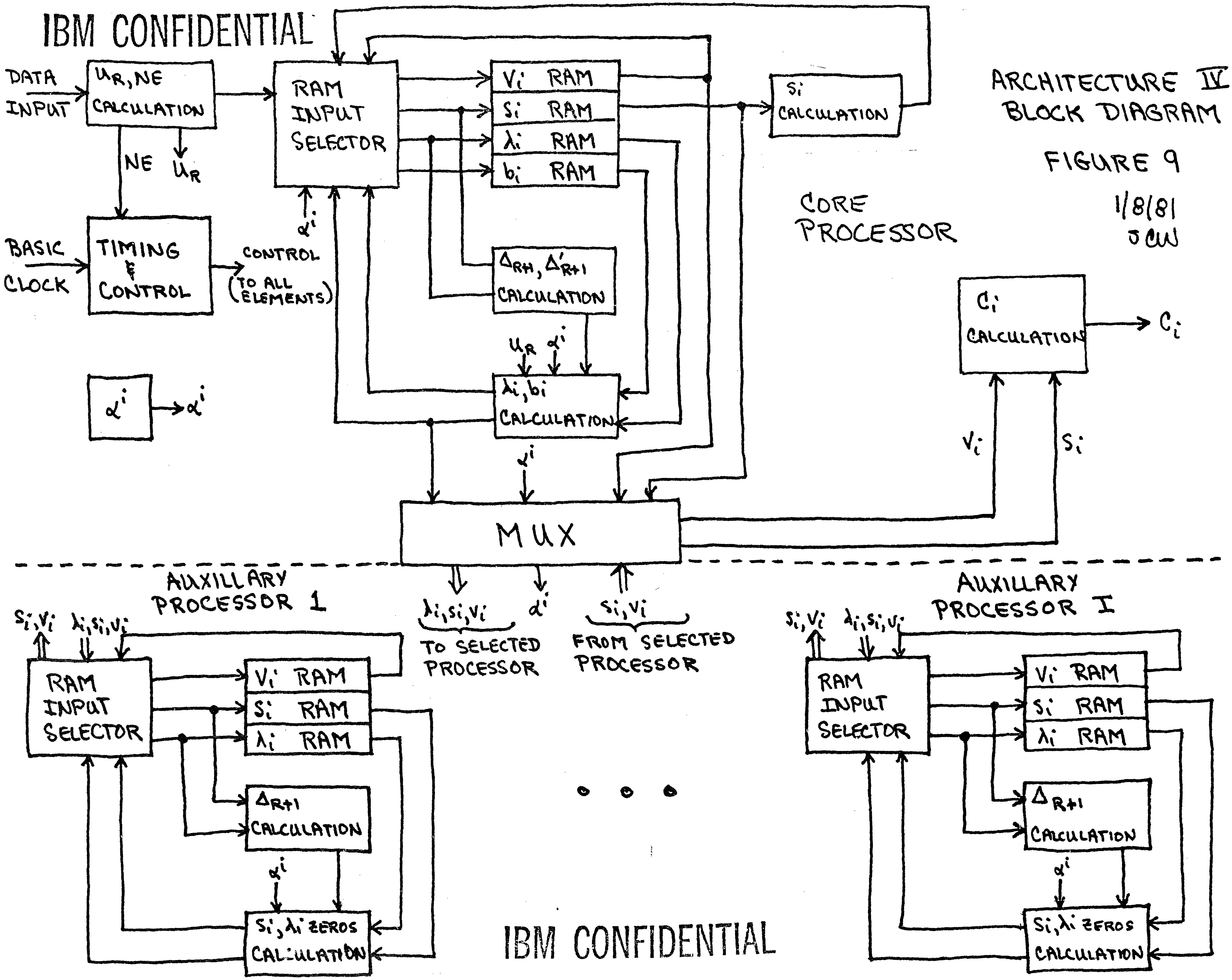

$V_i \leftarrow$ RECEIVED DATA
$S_i \leftarrow \alpha^i V_i$
$\lambda_i \leftarrow b_i \leftarrow 1$ $\Delta_m \leftarrow 1$
$L \leftarrow R \leftarrow 0$
$\Delta'_{R+1} \leftarrow \Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$R \leftarrow R+1$
$R > 2t$
YES
NO
$S_i \leftarrow S_i - \Delta_R$
$R = N$
NO
YES
$s_i \leftarrow \alpha^i s_i$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
YES
NO
$c_i \leftarrow V_i - S_i$
SET BAD MESSAGE FLAG
L = NUMBER ZEROS OF λ_i
AND
NE ≤ 2t
$s_i \leftarrow \alpha^i s_i$
$R > NE$
NO
YES
$b_i \leftarrow \lambda_i \leftarrow \lambda_i - u_R \alpha^{-i} b_i$
$L \leftarrow L+1$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$\Delta'_R = 0$
NO
YES
$t_i \leftarrow \lambda_i - \Delta'_R \alpha^{-i} b_i$
$t_i \leftarrow \lambda_i$
$2L \leq R+NE-1$
NO
YES
$L \leftarrow R+NE-L$
$b_i \leftarrow \lambda_i$
$\lambda_i \leftarrow t_i$
$\Delta_m \leftarrow \Delta_R$
$b_i \leftarrow \alpha^{-i} b_i$
$\lambda_i \leftarrow t_i$
$\Delta_{R+1} \leftarrow \Sigma \lambda_i s_i$
$\Delta'_{R+1} \leftarrow \Delta_m^{-1} \Delta_{R+1}$

ARCHITECTURE IV
ALGORITHM

FIGURE 8

IBM CONFIDENTIAL

IBM CONFIDENTIAL



ARCHITECTURE IV BLOCK DIAGRAM

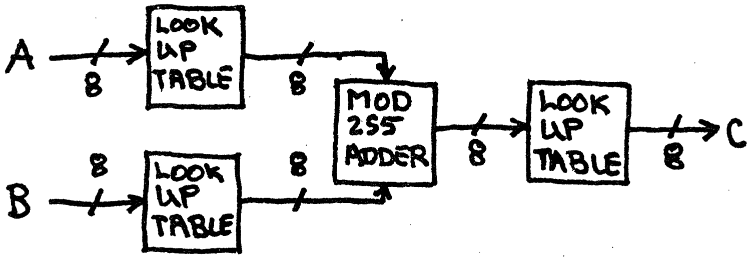
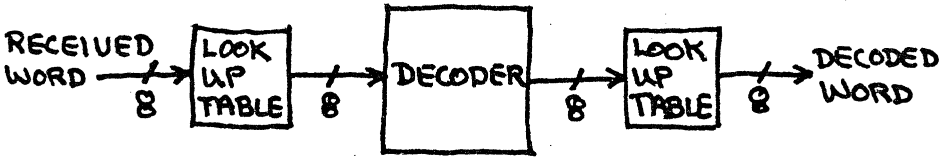
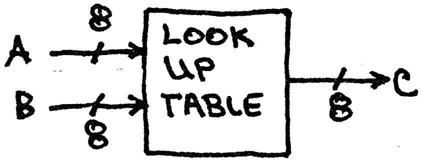
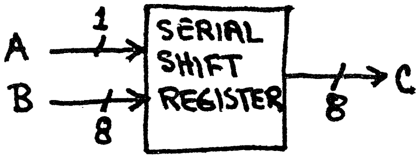
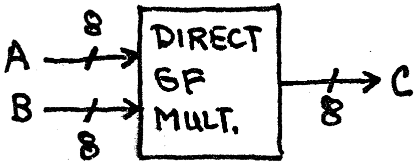
FIGURE 9

1/8/81
JCW

IBM CONFIDENTIAL

1/8/81
SCW

IBM CONFIDENTIAL

CONFIGURATIONS	# CIRCUITS	DELAY	GF ADDER
 A 8 LOOK UP TABLE 8 B 8 LOOK UP TABLE 8 MOD 255 ADDER 8 LOOK UP TABLE 8 C	3-256X8 ROM MEMORIES PER MULTIPLIER ~ 100 "AND", "OR", "XOR" CIRCUITS PER MULTIPLIER	2 MEMORY ACCESSES PLUS ~ 25 CIRCUIT DELAYS	8 "XOR" CIRCUITS 1 CIRCUIT DELAY
 RECEIVED WORD 8 LOOK UP TABLE 8 DECODER 8 LOOK UP TABLE 8 DECODED WORD	2-256X8 ROM MEMORIES PER DECODER ~ 100 "AND", "OR", "XOR" CIRCUITS PER MULTIPLIER	~ 25 CIRCUIT DELAYS	COMPLEX MULTI-CIRCUIT AND DELAY
	1-65536 x 8 ROM MEMORY PER MULTIPLIER	1 MEMORY ACCESS	8 "XOR" CIRCUITS 1 CIRCUIT DELAY
	8 SHIFT REGISTER STAGES PER MULTIPLIER ~ 20 "AND", "XOR" CIRCUITS PER MULTIPLIER	8 CLOCKS	8 "XOR" CIRCUITS 1 CIRCUIT DELAY
	~ 140 "AND", "XOR" CIRCUITS PER MULTIPLIER	7 CIRCUIT DELAYS	8 "XOR" CIRCUITS 1 CIRCUIT DELAY

COMPARISON OF GF MULTIPLIER IMPLEMENTATIONS

FIGURE 10

IBM CONFIDENTIAL

1/8/81
SCW

IBM CONFIDENTIAL

1. DEFINE THE FIRST $2r-1$ ELEMENTS OF THE GALOIS FIELD AND ARRANGE IN AN r BY r^2 MATRIX AS FOLLOWS:

$$M = \left[\alpha^0 \alpha^1 \alpha^2 \cdots \alpha^{r-1} \alpha^1 \alpha^2 \alpha^3 \cdots \alpha^r \cdots \alpha^{r-1} \alpha^r \alpha^{r+1} \cdots \alpha^{2r-2}\right]$$

(EACH α IS REPRESENTED BY AN r BIT COLUMN VECTOR {LSB AT TOP} AND THERE ARE r^2 α'S)

2. DEFINE THE r^2 "AND" PRODUCTS AND ARRANGE IN AN r^2 ELEMENT VECTOR AS FOLLOWS:

"AND" PRODUCTS

$$A_i \cdot B_j = X_{(i+j \bmod r)+1,\, j+1} \quad \begin{cases} i = 0, 1, \cdots, r-1 \\ j = 0, 1, \cdots, r-1 \end{cases}$$

VECTOR

$$V = \left[X_{11}\, X_{21}\, X_{31} \cdots X_{r1}\, X_{22}\, X_{32}\, X_{42} \cdots X_{12} \cdots X_{rr}\, X_{1r}\, X_{2r} \cdots X_{r-1\, r}\right]$$

3. THE GALOIS FIELD PRODUCT OF A, B (EACH r ELEMENT VECTORS) IS THE VECTOR C, I.E., $A \times B = C$. C CAN BE FOUND USING THE MATRIX M AND THE VECTOR V AS FOLLOWS:

$$C = M \underbrace{\neq . \wedge}_{\text{APL INNER PRODUCT}} V^T$$

PROCEDURE FOR DEVELOPING DIRECT GF MULTIPLIER EQUATIONS

FIGURE 11

IBM CONFIDENTIAL

IBM CONFIDENTIAL

1/8/81
SCW

THE ELEMENTS OF GF(8) ARE :

0	0 0 0
α^0	0 0 1
α^1	0 1 0
α^2	1 0 0
α^3	0 1 1
α^4	1 1 0
α^5	1 1 1
α^6	1 0 1

(DEFINED BY THE POLYNOMIAL $X^3 + X + 1$)

$(r = 3)$

$$M = \left[\alpha^0 \alpha^1 \alpha^2 \alpha^1 \alpha^2 \alpha^3 \alpha^2 \alpha^3 \alpha^4\right] = \begin{bmatrix} 1&0&0&0&0&1&0&1&0 \\ 0&1&0&1&0&1&0&1&1 \\ 0&0&1&0&1&0&1&0&1 \end{bmatrix} \leftarrow \text{LSB}$$

$$V = \left[X_{11}\ X_{21}\ X_{31}\ X_{22}\ X_{32}\ X_{12}\ X_{33}\ X_{13}\ X_{23}\right]$$

$$= \left[A_0B_0\ \ A_1B_0\ \ A_2B_0\ \ A_0B_1\ \ A_1B_1\ \ A_2B_1\ \ A_0B_2\ \ A_1B_2\ \ A_2B_2\right]$$

$$C = \begin{bmatrix} X_{11} \oplus X_{12} \oplus X_{13} \\ X_{21} \oplus X_{22} \oplus X_{12} \oplus X_{13} \oplus X_{23} \\ X_{31} \oplus X_{32} \oplus X_{33} \oplus X_{23} \end{bmatrix}$$

$$= \begin{bmatrix} A_0B_0 \oplus A_2B_1 \oplus A_1B_2 \\ A_1B_0 \oplus A_0B_1 \oplus A_2B_1 \oplus A_1B_2 \oplus A_2B_2 \\ A_2B_0 \oplus A_1B_1 \oplus A_0B_2 \oplus A_2B_2 \end{bmatrix}$$

$$\alpha^3 \times \alpha^6 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \oplus 0 \oplus 1 \\ 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \\ 0 \oplus 0 \oplus 1 \oplus 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \alpha^2$$

$$\alpha^1 \times \alpha^1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \oplus 0 \oplus 0 \\ 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \\ 0 \oplus 1 \oplus 0 \oplus 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \alpha^2$$

GF(8) EXAMPLE OF DIRECT GF MULTIPLIER

FIGURE 12

IBM CONFIDENTIAL

IBM CONFIDENTIAL

1/8/81
SCW

GF(2)

$C_0 = X_{11}$

GF(4)

$C_0 = \oplus / X_{11}, X_{12}$
$C_1 = \oplus / X_{12}, X_{21}, X_{22}$

GF(8)

$C_0 = \oplus / X_{11}, X_{12}, X_{13}$
$C_1 = \oplus / X_{12}, X_{13}, X_{21}, X_{22}, X_{23}$
$C_2 = \oplus / X_{23}, X_{31}, X_{32}, X_{33}$

GF(16)

$C_0 = \oplus / X_{11}, X_{12}, X_{13}, X_{14}$
$C_1 = \oplus / X_{12}, X_{13}, X_{14}, X_{21}, X_{22}, X_{23}, X_{24}$
$C_2 = \oplus / X_{23}, X_{24}, X_{31}, X_{32}, X_{33}, X_{34}$
$C_3 = \oplus / X_{34}, X_{41}, X_{42}, X_{43}, X_{44}$

GF(32)

$C_0 = \oplus / X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{45}$
$C_1 = \oplus / X_{21}, X_{22}, X_{23}, X_{24}, X_{25}$
$C_2 = \oplus / X_{12}, X_{13}, X_{14}, X_{15}, X_{31}, X_{32}, X_{33}, X_{34}, X_{35}, X_{45}$
$C_3 = \oplus / X_{23}, X_{24}, X_{25}, X_{41}, X_{42}, X_{43}, X_{44}, X_{45}$
$C_4 = \oplus / X_{34}, X_{35}, X_{51}, X_{52}, X_{53}, X_{54}, X_{55}$

GF(64)

$C_0 = \oplus / X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}$
$C_1 = \oplus / X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}$
$C_2 = \oplus / X_{23}, X_{24}, X_{25}, X_{26}, X_{31}, X_{32}, X_{33}, X_{34}, X_{35}, X_{36}$
$C_3 = \oplus / X_{34}, X_{35}, X_{36}, X_{41}, X_{42}, X_{43}, X_{44}, X_{45}, X_{46}$
$C_4 = \oplus / X_{45}, X_{46}, X_{51}, X_{52}, X_{53}, X_{54}, X_{55}, X_{56}$
$C_5 = \oplus / X_{56}, X_{61}, X_{62}, X_{63}, X_{64}, X_{65}, X_{66}$

GF(128)

$C_0 = \oplus / X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{56}, X_{57}$
$C_1 = \oplus / X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{67}$
$C_2 = \oplus / X_{31}, X_{32}, X_{33}, X_{34}, X_{35}, X_{36}, X_{37}$
$C_3 = \oplus / X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{41}, X_{42}, X_{43}, X_{44}, X_{45}, X_{46}, X_{47}, X_{56}, X_{57}$
$C_4 = \oplus / X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{51}, X_{52}, X_{53}, X_{54}, X_{55}, X_{56}, X_{57}, X_{67}$
$C_5 = \oplus / X_{34}, X_{35}, X_{36}, X_{37}, X_{61}, X_{62}, X_{63}, X_{64}, X_{65}, X_{66}, X_{67}$
$C_6 = \oplus / X_{45}, X_{46}, X_{47}, X_{71}, X_{72}, X_{73}, X_{74}, X_{75}, X_{76}, X_{77}$

DIRECT GF MULTIPLIER EQUATIONS
FOR GF(2) - GF(128)

FIGURE 13

IBM CONFIDENTIAL

1/8/81
SCW

$$C_0 = \oplus / \underbrace{X_{11}}_{Y_1}, \underbrace{X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}}_{Y_2}, \underbrace{X_{56}, X_{57}, X_{58}}_{Y_{10}}, \underbrace{X_{67}, X_{68}}_{Y_{12}}, \underbrace{X_{78}}_{Y_{14}}$$

$$C_1 = \oplus / \underbrace{X_{21}, X_{22}}_{Y_3}, \underbrace{X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}}_{Y_4}, \underbrace{X_{67}, X_{68}}_{Y_{12}}, \underbrace{X_{78}}_{Y_{14}}$$

$$C_2 = \oplus / \underbrace{X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}}_{Y_2}, \underbrace{X_{31}, X_{32}, X_{33}}_{Y_5}, \underbrace{X_{34}, X_{35}, X_{36}, X_{37}, X_{38}}_{Y_6}, \underbrace{X_{56}, X_{57}, X_{58}}_{Y_{10}}, \underbrace{X_{78}}_{Y_{14}}$$

$$C_3 = \oplus / \underbrace{X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}}_{Y_2}, \underbrace{X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}}_{Y_4}, \underbrace{X_{41}, X_{42}, X_{43}, X_{44}}_{Y_7}, \underbrace{X_{45}, X_{46}, X_{47}, X_{48}}_{Y_8}$$

$$C_4 = \oplus / \underbrace{X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}}_{Y_2}, \underbrace{X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}}_{Y_4}, \underbrace{X_{34}, X_{35}, X_{36}, X_{37}, X_{38}}_{Y_6}, \underbrace{X_{51}, X_{52}, X_{53}, X_{54}, X_{55}}_{Y_9}, \underbrace{X_{78}}_{Y_{14}}$$

$$C_5 = \oplus / \underbrace{X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}}_{Y_4}, \underbrace{X_{34}, X_{35}, X_{36}, X_{37}, X_{38}}_{Y_6}, \underbrace{X_{45}, X_{46}, X_{47}, X_{48}}_{Y_8}, \underbrace{X_{61}, X_{62}, X_{63}, X_{64}, X_{65}, X_{66}}_{Y_{11}}$$

$$C_6 = \oplus / \underbrace{X_{34}, X_{35}, X_{36}, X_{37}, X_{38}}_{Y_6}, \underbrace{X_{45}, X_{46}, X_{47}, X_{48}}_{Y_8}, \underbrace{X_{56}, X_{57}, X_{58}}_{Y_{10}}, \underbrace{X_{71}, X_{72}, X_{73}, X_{74}, X_{75}, X_{76}, X_{77}}_{Y_{13}}$$

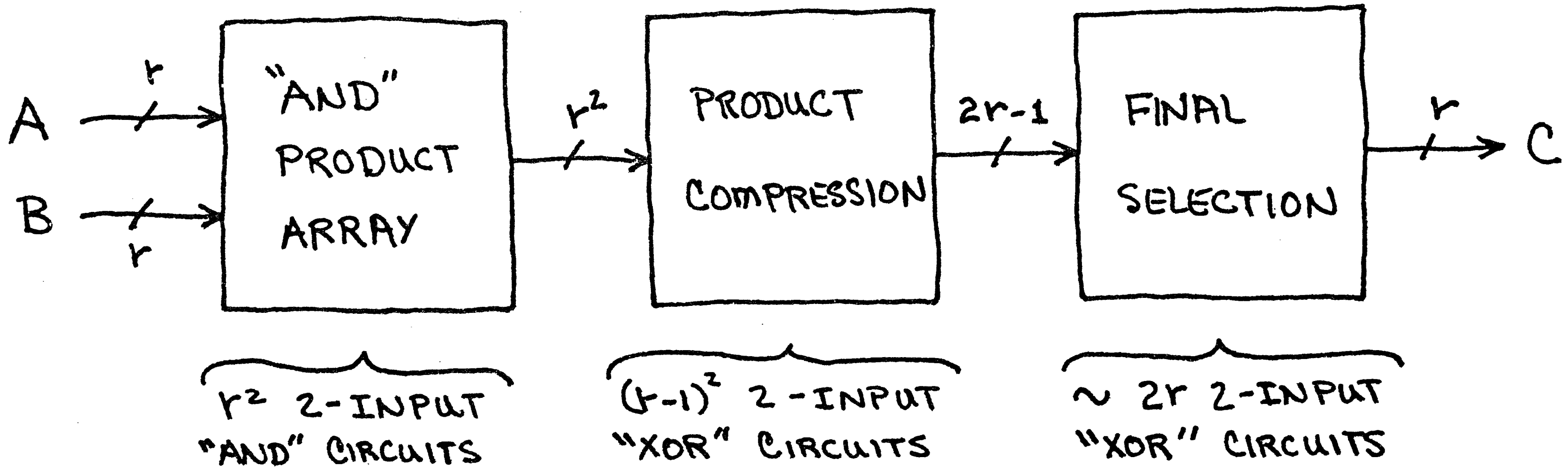
$$C_7 = \oplus / \underbrace{X_{45}, X_{46}, X_{47}, X_{48}}_{Y_8}, \underbrace{X_{56}, X_{57}, X_{58}}_{Y_{10}}, \underbrace{X_{67}, X_{68}}_{Y_{12}}, \underbrace{X_{81}, X_{82}, X_{83}, X_{84}, X_{85}, X_{86}, X_{87}, X_{88}}_{Y_{15}}$$

DIRECT GF MULTIPLIER EQUATIONS
FOR GF(256)

FIGURE 14

IBM CONFIDENTIAL

1/8/81
SCW


A
B
r
r
"AND" PRODUCT ARRAY
r^2
PRODUCT COMPRESSION
2r-1
FINAL SELECTION
r
C
r^2 2-INPUT "AND" CIRCUITS
$(r-1)^2$ 2-INPUT "XOR" CIRCUITS
~ 2r 2-INPUT "XOR" CIRCUITS

$C = \underbrace{A \times B}$ MOD (GF GENERATOR POLYNOMIAL)

↳ POLYNOMIAL MULTIPLY

r^2 2-INPUT "AND" CIRCUITS

~ r^2+1 2-INPUT "XOR" CIRCUITS

(r = NUMBER BITS/GF ELEMENT)

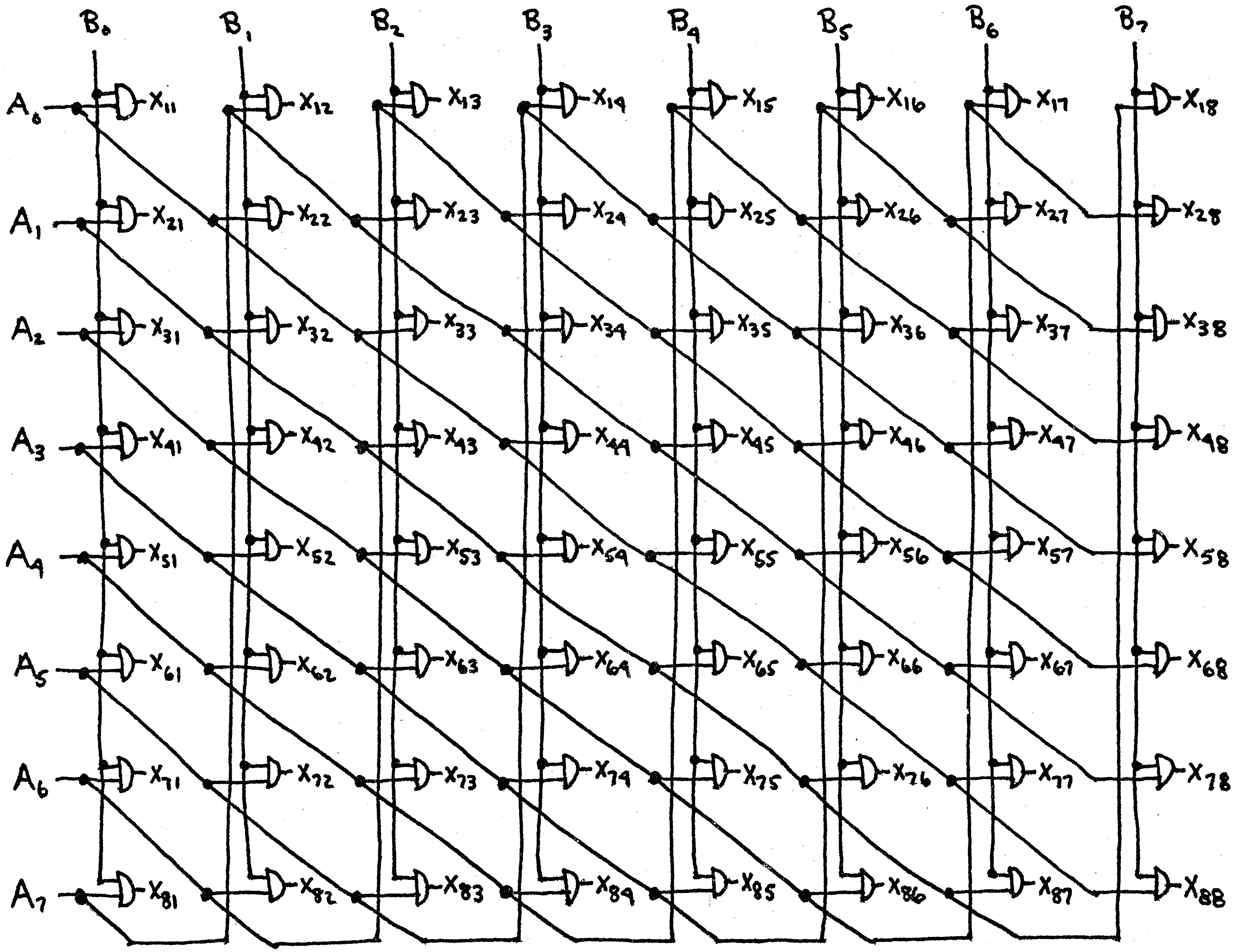
DIRECT GF MULTIPLIER
BLOCK DIAGRAM

FIGURE 15

IBM CONFIDENTIAL

IBM CONFIDENTIAL

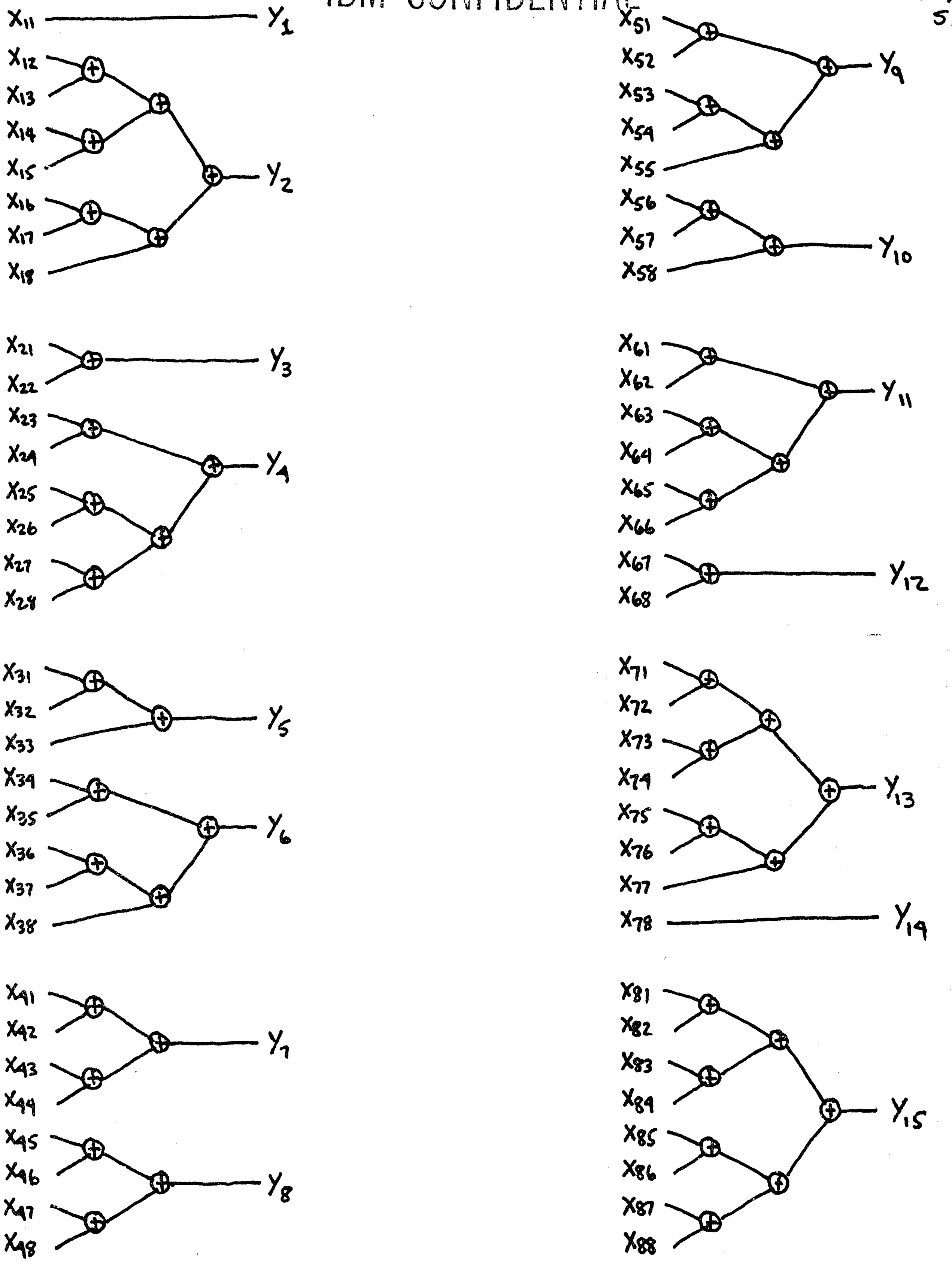
1/8/81
SCW



"AND" PRODUCT ARRAY
FIGURE 16

IBM CONFIDENTIAL

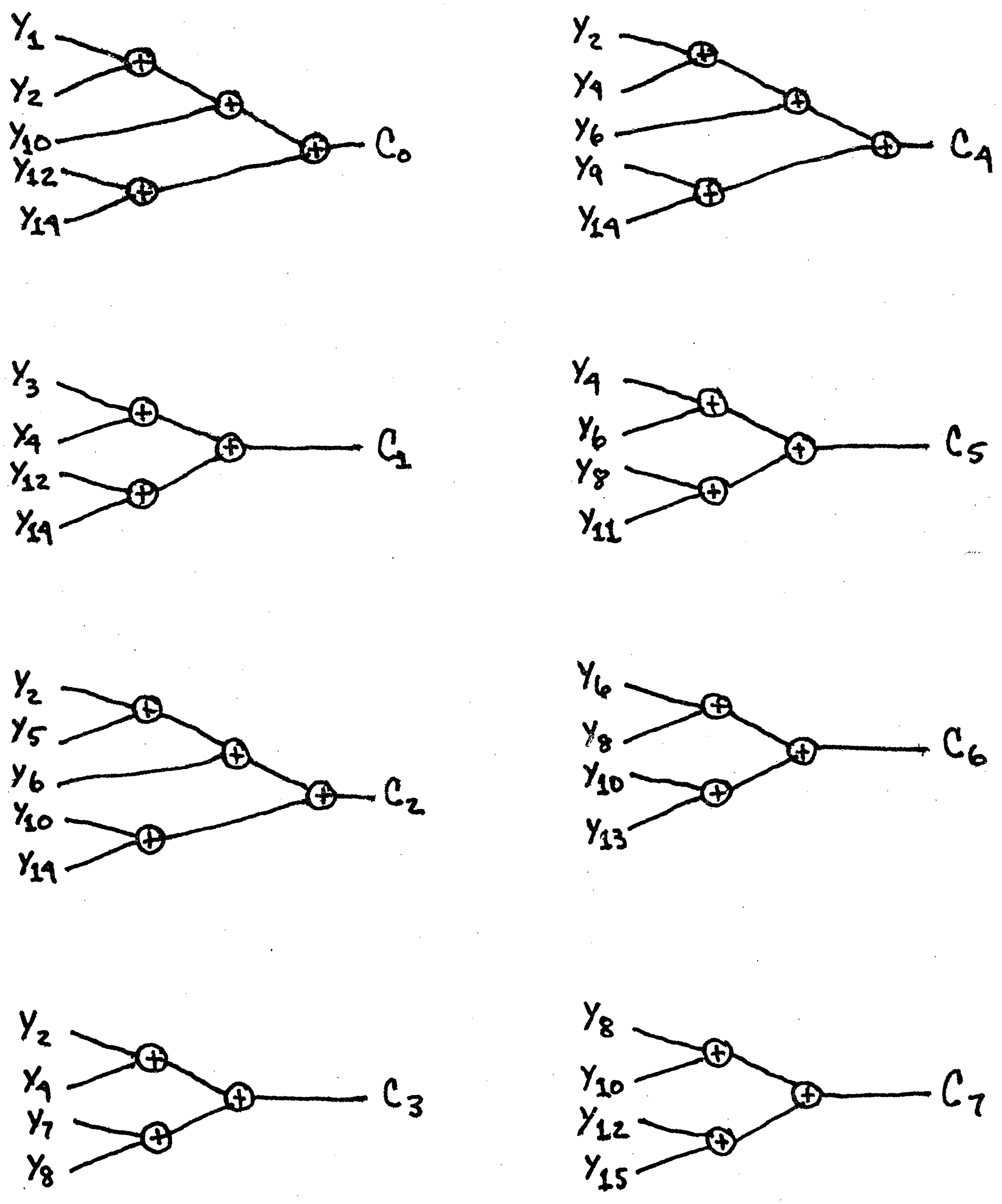
1/8/81
SCW



PRODUCT COMPRESSION

FIGURE 17

1/8/81
SCW



FINAL SELECTION

FIGURE 18

ON THE FLY ERROR CORRECTION FOR DISK FILES

BY William Bliss 208/050-1 Rochester, Mn. Jan. 10, 1981

Present error correcting systems for disk files typically can only correct a single error burst in a block and require the system to 'hickup' to achieve correction. The decoding of multiple burst error correcting codes can be implemented at a minimal expense by using slower software techniques instead of dedicated hardware solutions. Methods do exist to utilize the encoder hardware/standard syndrome generator to assist in the calculations of the syndrome components (fig. 1&2). If throughput degradation is acceptable then no special high speed circuitry is required. The software does utilize a modest amount of hardware

As the need for multiple burst correction becomes greater the soft error rate will worsen until an unacceptable level is reached. At this raw error rate on the fly correction of single error bursts will become cost effective. This will necessitate an additional block of buffering to store data while it is checked for correctibility under the constraint of one burst (fig. 3). The system will be allowed to hickup if more than one burst error occurs in a block. In the time of one data block the syndrome components must be calculated, the error solved for under the assumption of only one occurance, and that solution checked for validity (fig. 4). We are probably in the realm of 'special Galois processor hardware' to achieve these goals. A hardware syndrome component calculator could be used to ease these high speed calculation constraints and also to speed up multiple burst correction (fig. 5). Alternately, a duplicate encoder/standard syndrome generator could be used with a traditional burst error trapping method (fig. 6).

For systems with even higher raw error probabilities it may be desirable to achieve on the fly double burst error correction with Reed-Solomon codes. The solution is similar to that of single burst correction except that a lookup table is used to solve for the roots of the error location polynomial (fig. 7).

On the fly correction of three bursts in a block is more difficult for all B.C.H. codes. A Chien root searcher (hardware) is needed in all cases. The root searcher can operate synchronously with the appropriate output buffer but the validity of the correction is not known until tested after the fact. A Reed-Solomon code requires additional data buffering to allow time for calculation of the values of the errors at the locations indicated by the recipricals of the roots of the error location polynomial.

$r(x)$ is recieved word, degree $< n$

$g(x)$ is generator polynomial, degree $= r$

S_j is the jth Syndrome component, $\in GF(2^m)$

$S_j = r(\alpha^j)$ $\quad$ $n-1$ multiplications

The encoder hardware can easily generate an 'overall syndrome' $\varrho(x)$

$\varrho(x) = \text{Rem}\left[\frac{r(x)}{g(x)}\right]$ $\qquad$ $\varrho_i \in GF(2)$ binary BCH
$\in GF(2^m)$ R-S

$$r(x) = q(x)\, g(x) + \varrho(x)$$

$$S_j = r(\alpha^j) = q(x) \cdot 0 + \varrho(\alpha^j) = \varrho(\alpha^j)$$

which is $r-1$ multiplications

Reduction by $\frac{r-1}{n-1} \simeq \frac{r}{n}$

Syndrome Component Calculation Using Standard Syndrome Remainder

fig 1

SYNDROME COMPONENT CALCULATIONS for Binary BCH codes

$$S_i = \delta(\alpha^i) = \delta_{r-1}\alpha^{(r-1)i} + \delta_{r-2}\alpha^{(r-2)i} + \cdots + \delta_1\alpha^i + \delta_0$$

$$= (\cdots((\delta_{r-1})\alpha^i + \delta_{r-2})\alpha^i + \cdots \delta_1)\alpha^i + \delta_0$$

$$= [\cdots[[(\delta_{r-1}\alpha^{i(b-1)} + \delta_{r-2}\alpha^{i(b-2)} + \cdots + \delta_{r-b}\alpha^{i(b-b)}]\alpha^{ib} +$$

$$(\delta_{r-b-1}\alpha^{i(b-1)} + \delta_{r-b-2}\alpha^{i(b-2)} + \cdots + \delta_{r-2b}\alpha^0)]\alpha^{ib} +$$

$$\vdots$$

$$(\delta_{2b-1}\alpha^{i(b-1)} + \delta_{2b-2}\alpha^{i(b-2)} + \cdots + \delta_b)]\alpha^{ib} +$$

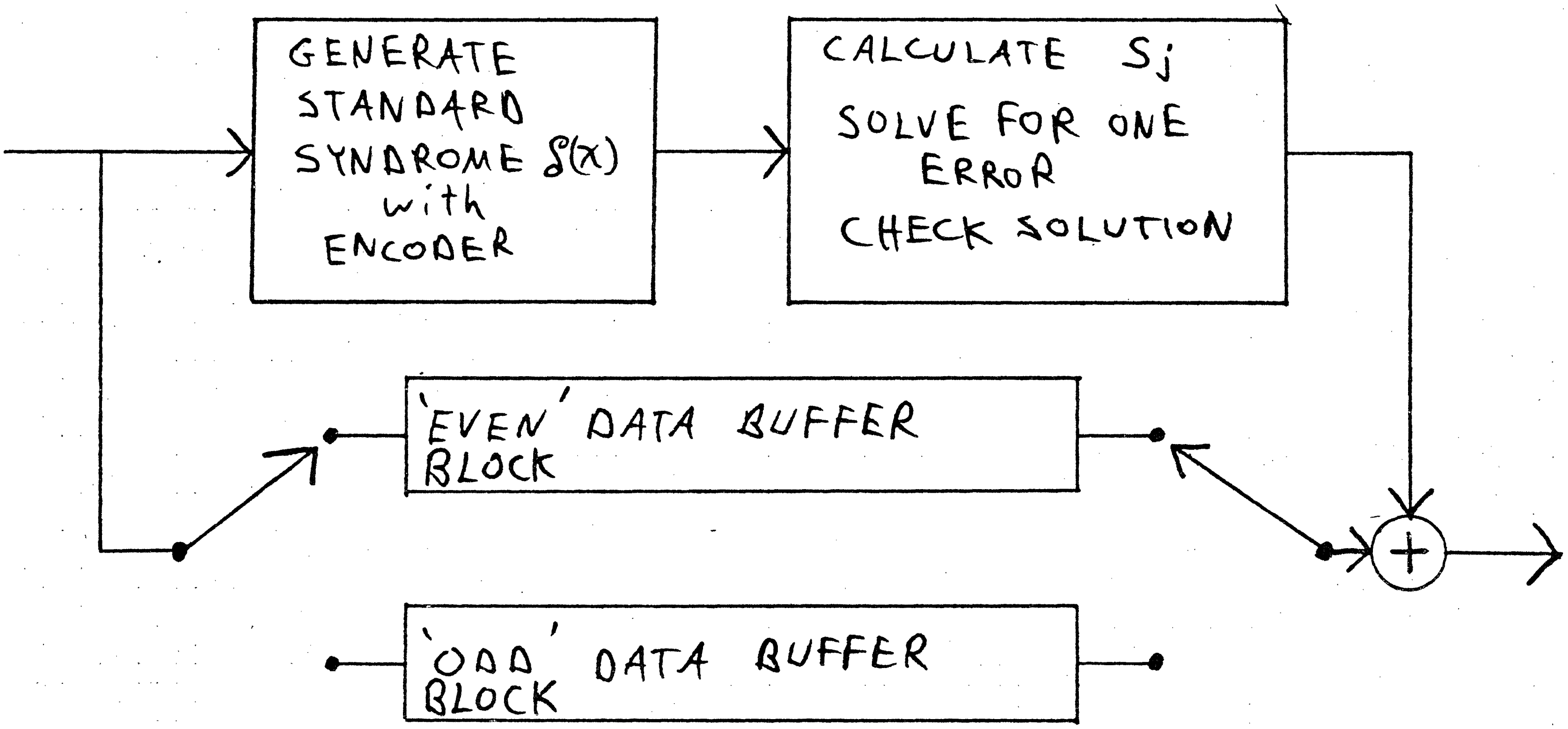
$$(\delta_{b-1}\alpha^{i(b-1)} + \delta_{b-2}\alpha^{i(b-2)} + \cdots + \delta_0)$$

Let $\frac{r}{b} = R$(an integer)# of multiplies over $GF(2^m)$. $\alpha^{i(b-1)}, \alpha^{i(b-2)}, \cdots, \alpha^i$ are pre-computed constants.

Fig. 2

$$X_1 = \frac{S_2}{S_1}$$

$$Y_1 = \frac{S_1^2}{S_2}$$



On the Fly Correction of Single Bursts with Software

fig. 3

$t=1$

Error Location

$$\begin{bmatrix} S_1 & \cdots & S_r \\ \vdots & & \\ S_r & & S_{2r-1} \end{bmatrix} \begin{bmatrix} \sigma_r \\ \vdots \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} S_{r+1} \\ \vdots \\ S_{2r} \end{bmatrix}$$

$$S_1 \quad \sigma_1 = S_2$$

$$\sigma_1 = \frac{S_2}{S_1}$$

$$\sigma'(x) = x + \sigma_1 = 0$$

$$x + \frac{S_2}{S_1} = 0$$

$$X_1 = \frac{S_2}{S_1}$$

Error Value

$$\begin{bmatrix} X_1 & \cdots & X_r \\ \vdots & & \vdots \\ X_1^r & \cdots & X_r^r \end{bmatrix} \begin{bmatrix} Y_1 \\ \vdots \\ Y_r \end{bmatrix} = \begin{bmatrix} S_1 \\ \vdots \\ S_r \end{bmatrix}$$

$$X_1 \quad Y_1 = S_1$$

$$Y_1 = \frac{S_1}{X_1} = \frac{S_1}{S_2/S_1} = \frac{S_1^2}{S_2}$$

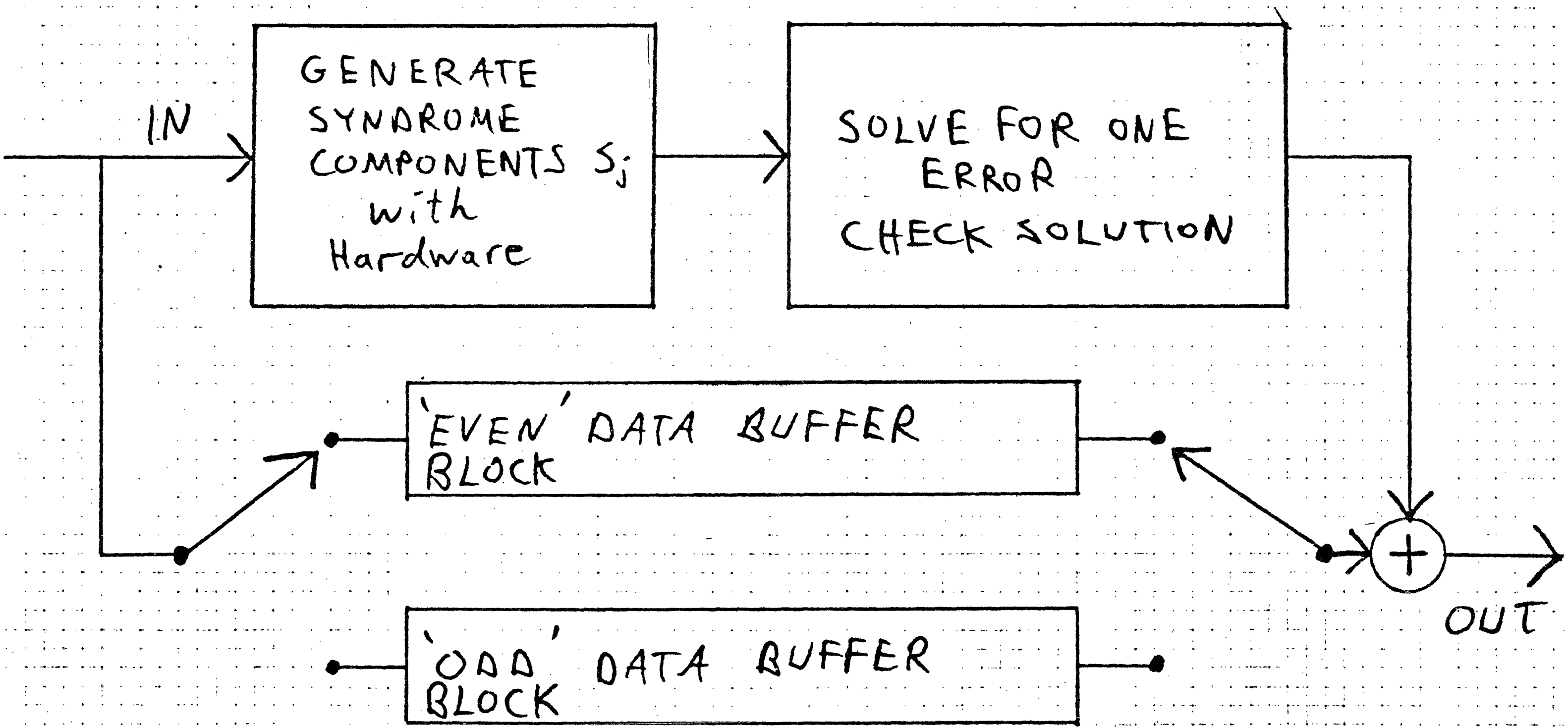
Check Solution

$$S_j' = S_j + Y_1(X_1)^j \overset{?}{=} 0 \quad \text{for} \quad 1 \leq j \leq 2t$$

Fig. 4 Solution and Check for One Burst

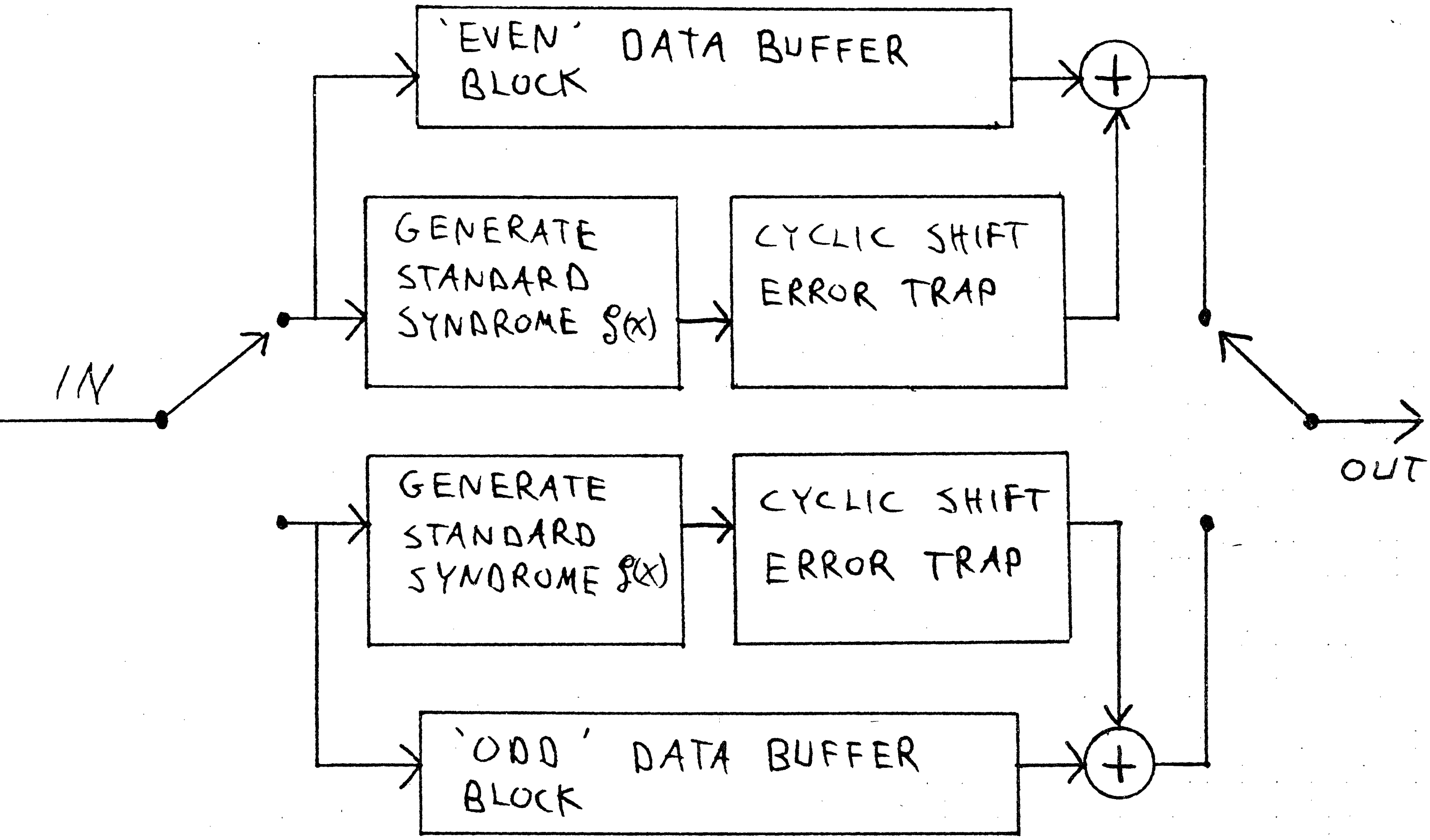
$$X_1 = \frac{S_2}{S_1}$$

$$Y_1 = \frac{S_1^2}{S_2}$$


IN
GENERATE SYNDROME COMPONENTS S_j with Hardware
SOLVE FOR ONE ERROR CHECK SOLUTION
'EVEN' DATA BUFFER BLOCK
'ODD' DATA BUFFER BLOCK
+
OUT

On the Fly Correction of Single Bursts with Dedicated Hardware Syndrome Component Generator

Fig. 5



On the Fly Correction of Single Bursts with Duplicate Encoder and Simple Error Trapping Decoding

fig. 6

$$\sigma'(x) = x^2 + \sigma_1 x + \sigma_2 = 0$$

where $\sigma_1 = \dfrac{S_2 S_3 + S_1 S_4}{S_2^2 + S_1 S_3}$

$$\sigma_2 = \frac{S_3^2 + S_2 S_4}{S_2^2 + S_1 S_3}$$

Let $x = y\sigma_1$

$$\sigma_1^2 y_2^2 + \sigma_1^2 y = \sigma_2$$

$$y^2 + y = \frac{\sigma_2}{\sigma_1^2} = d$$

Look-up d to get a root 'y_p'

$y_p + 1$ is other root

$$X_1 = \sigma_1 y_p$$

$$X_2 = \sigma_1 (y_p + 1)$$

Lookup Table to Find Roots of Second Degree Equation

fig. 7

ERROR RECOVERY STRATEGY FOR FUTURE DASD

ARVIND M. PATEL
F97 /060 x 6-2598

IBM CONFIDENTIAL 12/16/8

WHITNEY: ERROR CORRECTION CODE.

- Each Area Count
  Key
  DATA
  has 12 check bytes ECC.
  (WORD = 13 bytes to 47476 bytes)

- Correct One group of 3 Consecutive bytes
  (50% of 4 byte groups).

- High degree of Error Detection.

- Syndrome Computation in Whitney.
  Syndrome Decoding in Storage Control
  Error correction in Host Memory.

- ECC Uncorrectable Error → Reread with Offsets.
  Upto. 37 reread operations in ERP

- Unrecoverable Error → Track flagged Unusable
  Alternate Track assigned.

$\frac{12 \times \overset{CKD}{3}}{1000} \Rightarrow$ 3.6% for 1000 byte record
1.8% for 2000 " "
.36% for 10000 byte "

FUTURE ENVIRONMENT.

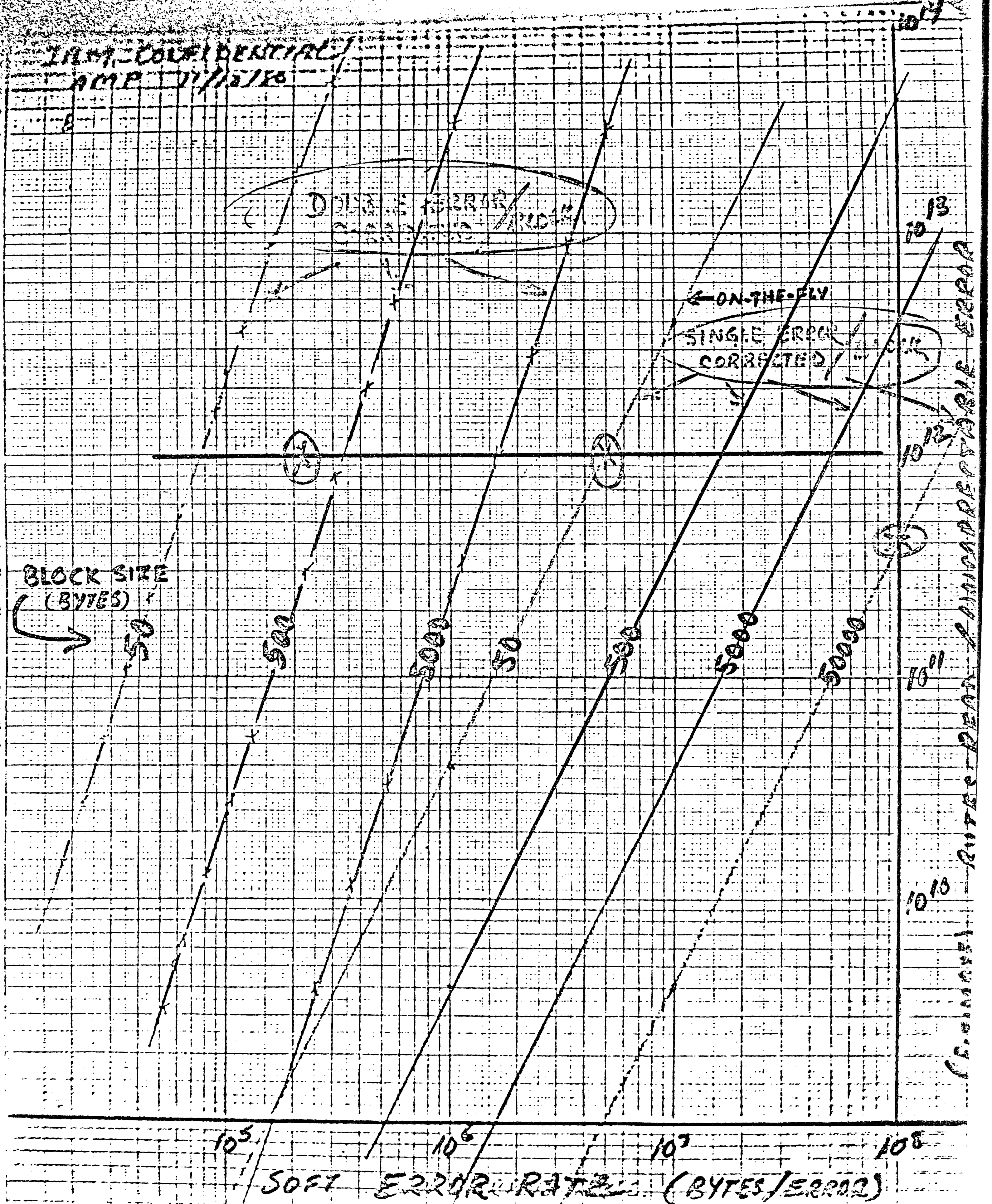
- HIGHER DENSITY/DATA RATE.
  - SHORT MULTIPLE ERRORS.
- DEMAND FOR HIGHER RELIABILITY./DATA INTEGRITY.
- RELIABILITY AWARENESS OF CUSTOMER.
  - RELIABILITY +
- IBM's EMPHASIS ON QUALITY.
- COST OF UNSCHEDULED MAINTENANCE IS HIGH and INCREASING.
- IMPACT OF HIGHER SOFT ERROR RATE.
  - ON PERFORMANCE
  - CUSTOMER ACCEPTANCE

IBM CONFIDENTIAL
AMP 11/13/80

RATIONALE AND DESIGN POINT.

- COST OF ECC IS IN
  - DECODING HARDWARE
  - DECODING TIME
- REDUNDANCY IS NOT COST. (≈5%)
  - BUY BACK WITH DENSITY.
  - REDUCE DECODING COST. - HARDWARE
- TIME
- ON-THE-FLY CORRECTION OF MOST ERRORS
  - WITHOUT CPU INVOLVEMENT
  - NO EFFECT ON THRU-PUT
  - NO NEED TO LOG ERRORS.
- PLAN ON CORRECTING LOT OF ERRORS.
  - INCREASE DENSITY
  - TRADE-OFF IN HEAD-DISK TOLERANCES.
  - PARTIAL COVERAGE FOR SMALL DEFECTS
- RESERVE CAPABILITY TO COVER WEAKER DEVICE
  - SIGNAL WEAKER DEVICE
  - AVOID UNSCHEDULED SERVICE CALLS.

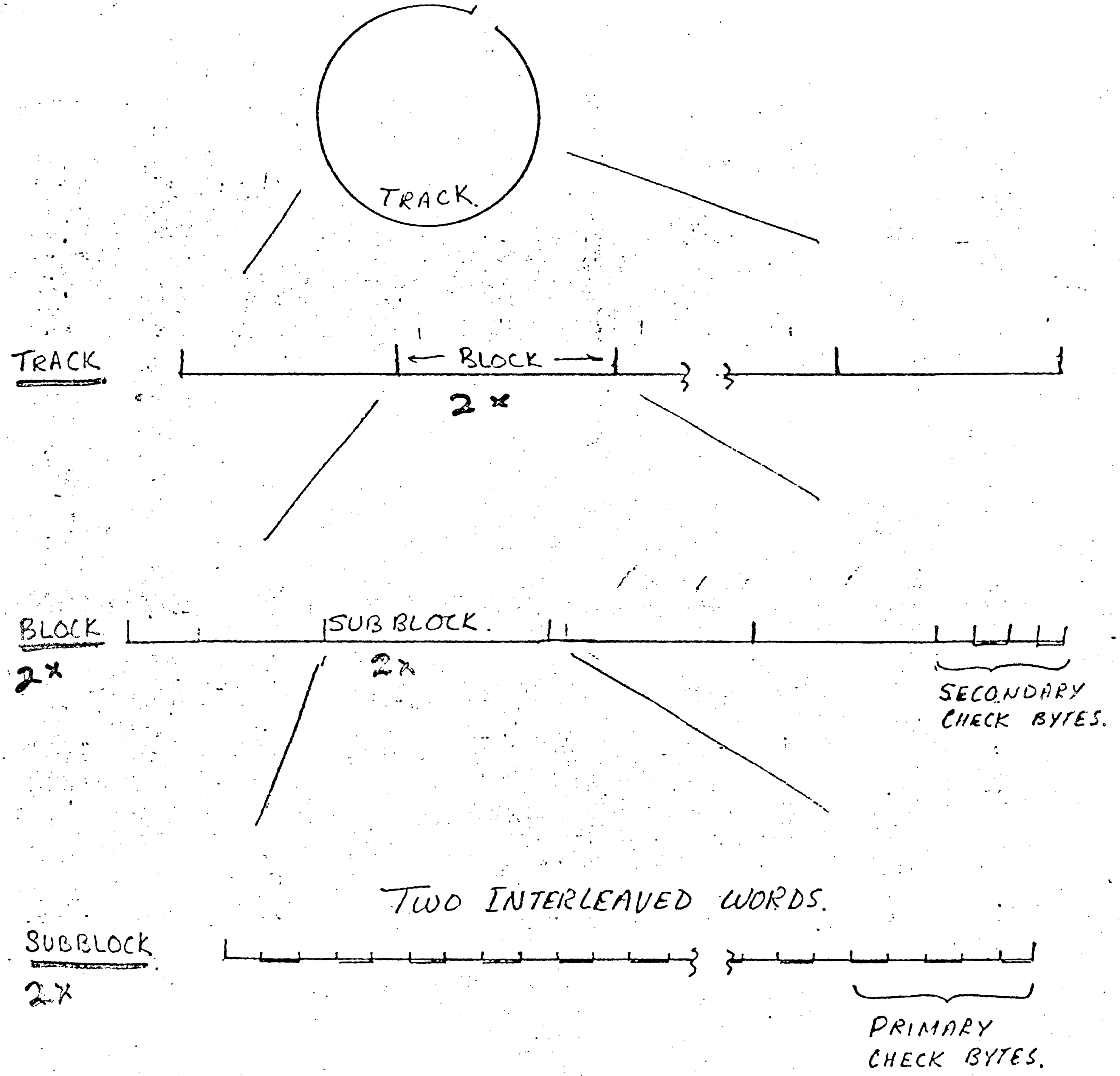
IBM-CONFIDENTIAL
AMP 11/18/80



TWO LEVEL CODING SCHEME

- SEPERABLE CODING STRUCTURE WITH TWO LEVEL CAPABILITY
- PRIMARY CAPABILITY ON-THE-FLY.
- SECONDARY CAPABILITY ON FURTHER PROCESSING
- SIMPLE CHECK AND EASY TRANSITION FROM PRIMARY TO SECONDARY
- BYTE PROCESSING AND BYTE CORRECTION.
- INTERLEAVED FOR BURST CORRECTION.
- FLEXIBILITY ON BLOCK SIZE.

IBM CONFIDENTIAL
AMP 1/12/81.



ECC DATA FORMAT

IBM CONFIDENTIAL

PRIMARY WORD (SUBBLOCK) REDUNDANCY ≈ 3%

64 DATA + 2 CHECK = 66 BYTES/WORD

- CORRECT ONE BYTE/WORD (BYTE ERROR)
- MAXIMUM DECODE TIME ≈ 66 SHIFTS.
- ON-THE-FLY CORRECTION (~~66~~ 132 BYTE BUFFER)
- INTERLEAVE TWO WORDS (BURST 9-bit MIN)
- MORE INTERLEAVING FOR LONGR BURST. (⇒BUFFER

SECONDARY WORD (BLOCK) REDUNDANCY ≈ 1%

4 SUBBLOCK + 2 CHECK BYTES = 266 BYTES/BLOCK

- SIMPLE CHECK DETECTS DOUBLE ERROR
- CORRECTS TWO BYTE ERROR IN ONE SUBBLOCK
- DECODING VIA MICROCODE PROCESSING
- PROVIDES RESERVE CAPABILITY FOR WEAKER DEVI
- SIGNALS WEAKER DEVICE
- (WILL WORK ON ANY NUMBER OF SUBBLOCKS)

END-OF-RECORD CHECK. (Optional, 2 bytes) IBM CONFIDENTIAL

- INVOKES REREAD

AMP 11/13/80

CONCLUSIONS & COMMENTS

✓ DESIGN FOR HIGH SOFT ERROR RATE.

✓ TWO LEVEL CAPABILITY

FIRST LEVEL ⟺ AVERAGE DEVICE ?

SECOND LEVEL ⟺ WEAKER DEVICE ?

✓ OPTIMAL USEAGE OF REDUNDACY

✓ ON-THE-FLY CORRECTION OF MOST ERRORS

✓ FLEXIBILITY OF DESIGN TO FIT ERROR MODES (INCLUDING MOST DEFECTS).

✓ SECOND LEVEL PROCESSING CAN BE ELIMINATED LATER AND/OR IN ALTERNATE OFFERING.

✓ USE SUBBLOCK CODE AS INDEPENDENT CODE ON SMALL FIELDS (COUNT, KEY etc)

✓ TRUNCATE (OR PAD) SUBBLOCK A/O BLOCK IN CKD

✓ NEW CONCEPT IN ERROR PROCESSING IN DASD.

IBH CONFIDENTIAL
AMP 11/13/80

ENCODING PROCESS.

EACH SUBBLOCK

2 CHECK BYTES $B_0 \ B_1$
64 DATA BYTES $B_2 \ B_3 \ --- \ B_{65}$.

$$\left.\begin{aligned} \sum_{i=0}^{65} T^i B_i &= 0 \\ \sum_{i=0}^{65} T^{2i} B_i &= 0 \end{aligned}\right\}$$

16-bit SHIFT REGISTER w/ T^m & T^n MULTIPLIER.

- ON-THE-FLY COMPUTATIONS.

BLOCK.

2 CHECK BYTES P C.
ANY NUMBER OF SUBBLOCKS!

$$P = \sum_{\text{SUBBLOCK}} \left(\sum_i B_i \right)$$

OVERALL PARITY

$$C = \sum_{\text{SUBBLOCK}} \left(\sum_i T^{3i} B_i \right)$$

8-bit SHIFT REGISTER w/ T^3 MULTIPLIER.

- ON-THE-FLY COMPUTATIONS

SYNDROME GENERATION

EACH SUBBLOCK SYNDROMES S_1 S_2

$S_1 = \sum_i T^i \hat{B}_i$ 8-bit SR. w/ T MULT

$S_2 = \sum_i T^{2i} \hat{B}_i$ 8-bit SR w/ T^2 MULT

- ON-THE-FLY COMPUTATIONS.

BLOCK

$$S_0 = \hat{P} \oplus \sum_{\text{SUBBLOCK}} \left(\sum_i \hat{B}_i \right)$$

$$S_3 = \hat{C} \oplus \sum_{\text{SUBBLOCK}} \left(\sum_i T^{3i} \hat{B}_i \right)$$

- ON-THE-FLY COMPUTATIONS.

SYNDROME DECODING.

EACH SUBBLOCK e = BYTE ERROR in x^{th} byte

$$S_1 = T^{x} e$$

$$S_2 = T^{2x} e$$

$$e = T^{-x} S_1$$ 8-bit SR1 w/ T^{-1} MULT.

$$e = T^{-2x} S_2$$ 8-bit SR2 w/ T^{-2} MULT.

65 shifts max

x = # of SHIFTS IN SR1 and SR2 FOR CONTENTS TO BECOME EQUAL } ($x \leq 65$)

e = CONTENT OF BOTH REGISTERS.

CORRECT ERROR.

HOLD S_1 S_2 x e (FOR ONE SUBBLOCK W/ ERROR)*

HOLD Σe when MORE THAN ONE SUBBLOCK CORRECTED.

* SUBBLOCK THAT IS MOST LIKELY W/ MULTIPLE ERROR
e.g.
1. only subblock w/ SYNDROMES NONZERO
2. SUBBLOCK W/ UNDECODABLE SYNDROMES
3. SUBBLOCK W/ MORE THAN ONE DROPOUTS.

IBM CONFIDENTIAL.
AMP 12/12/80

SYNDROME DECODING (CONTD.)

BLOCK LEVEL CHECK BLOCK SYNDROME S_0

$S_0 = \Sigma e \Longrightarrow$ ALL CORRECT

$S_0 \neq \Sigma e \Longrightarrow$ MULTIPLE ERROR PRESENT

A DOUBLE ERROR IN ONE SUBBLOCK*

CORRECTED BY PROCESSING S_0 S_1 S_2 S_3

IBM CONFIDENTIAL
AMP 12/15/80

BLOCK CORRECTION

- ONE SUBBLOCK MISCORRECTED (S_1 S_2 x e on HOLD)
- OTHER SUBBLOCKS OK (S_1 S_2 ZERO)

S_0 S_3 ARE BLOCK SYNDROMES.

COMPUTE:

$$\left.\begin{aligned} A &= S_0 S_2 \oplus S_1 S_1 \\ B &= S_2 S_1 \oplus S_3 S_0 \\ R &= S_3 S_1 \oplus S_2 S_2 \end{aligned}\right\}$$

256 x 16
TABLE LOOK-UP
w/ ADDER FUNCTION
EX-OR

CHECK:

$$T^{2y} A \oplus T^{y} B \overset{?}{=} R \quad \Big\}$$

TWO SHIFT REGISTE
w/ T MULTIPLIER
T^2 MULTIPLIER

y (also z) = { # of SHIFTS IN T AND T^2 MULTIPLIE
REGISTERS FOR CONTENTS TO ADD-UPTO R
(IF $R=0$, THEN P IN ERROR, NO MISCORRECTION

COMPUTE ERROR PATTERNS:

FOR LOCATION y: $e_y = A/(T^{2y} S_0 \oplus S_2)$

FOR LOCATION z: $e_z = S_0 \oplus e_y$

CORRECT ERRORS:

$B_x = \hat{B}_x \oplus e$ (MISCORRECTION REMOVED)

$$\left.\begin{aligned} B_y &= \hat{B}_y \oplus e_y \\ B_z &= \hat{B}_z \oplus e_z. \end{aligned}\right\} \text{TWO ERRORS}$$

IBM CONFIDENTIAL