XEROX

Century Data Systems
A Xerox Company

# "THE CHOICE OF A RECORDING CODE"

Nigel D Mackintosh, B Sc, Ph D *

**Presented at:**

IERE Conference on Video and Data Recording
Southampton, England
July 1979

**Published in:**

The Radio and Electronic Engineer
Vol. 50, No.4
April 1980

* Century Data Systems, P O Box 3056, Anaheim, CA 92803-3056

UDC 681.3.04: 681.327.63/4

*Indexing Terms:* Recording, magnetic, Data storage, digital, Coding

# The choice of a recording code

**N. D. MACKINTOSH,** B.Sc., Ph.D.*

*Based on a paper presented at the IERE Conference on Video and Data Recording held in Southampton from 24th to 27th July 1979*

**SUMMARY**

A detailed comparison of recording codes, using superposition, is used to show that there is little to choose between the popular ones, as regards maximum achievable data density. It is shown that relatively few general classifications are necessary to encompass all the codes, and that, within each classification, it is often possible to say which code is optimum. A guide to the selection of an efficient code for some common conditions of use is presented.

## 1 Introduction

The choice of a recording code is of fundamental importance in the design of any moving magnetic digital storage system. Many papers have recently appeared extolling the virtues of yet another new code, and proving how superior it is to all other codes. Such papers are generally followed by another (biased) one proving that the new code is not as good as one of the old ones. As a result, anyone faced with the choice of a 'best' code in any particular situation can be forgiven for thinking it to be a more difficult task even than the selection of the 'best' microprocessor for a given application!

In an effort to alleviate this situation, a detailed, unbiased comparison of the vast majority of possible codes has been undertaken, using a superposition program. The superposition technique itself and the choice of the 'basic pulse' (isolated reversal response) are described in a companion paper.[1] To make any comparison between different recording codes valid, it is necessary to ensure that the conditions of use are the same for each code. To do this strictly would be to ensure that all codes were tested on the same mechanical disk unit, with fixed amounts of noise present, using exactly the same writing and reading electronics, and therefore with exactly the same mechanical and electrical tolerances. In practice, however, with a comparison which involves simulation rather than actual experimentation, it is sufficient to ensure that a standard basic pulse response is used throughout, with a fixed signal-to-noise ratio and a fixed level of error, pertaining to all tolerances and inaccuracies throughout the system. This last point is very important, and the errors due to these factors can conveniently be compounded into one figure, called real time error (*RTE*), and introduced into the analysis as clocking inaccuracies. *RTE* is thus typically composed of many factors, such as crosstalk, incomplete erasure effects, print-through, plating noise, component tolerances, phase-locked loop errors, and even an allowance for margin, if desired.

The results of this analysis of codes are presented for four different values of *RTE*, to allow a system designer to interpolate between the specific figures given to relate to the actual *RTE* anticipated in his own system. The first *RTE* allowance is $RTE = 0$, resulting in the theoretical maximum possible packing factor, *PF* (data density, normalized to $PW_{50}$, the half-amplitude basic pulse width). The second and third allowances are $RTE = 8\%\ PW_{50}$ and $RTE = 16\%\ PW_{50}$, being representative of clocking errors occurring in practice. The fourth allowance is an attempt to reflect the self-clocking ability of certain codes. Codes with regular transitions (having, therefore, a low value of maximum inter-reversal time, $\hat{IRT}$) are penalized in terms of flux-reversal density, but produce a more stable clock waveform. The fourth *RTE* allowance is therefore $RTE \propto \hat{IRT}$. Obviously the exact nature of the proportionality is difficult to define, but

* *Racal Recorders Limited, Hythe, Hampshire, England; now with Burroughs Corporation, 5411 North Lindero Canyon Road, Westlake Village, CA 91361, USA.*

0033-7722/80/040177+17 $1.50/0

based on measurements of an actual disk system, the following equation was determined:

$$RTE = (6{\cdot}7 + 1{\cdot}3 \times \hat{IRT})\%\ PW_{50}.$$

Note that for a code with an $\hat{IRT}$ of 1·0 × $BP$ (bit period), this conveniently yields an $RTE$ of 3% $PW_{50}$, as in the second $RTE$ allowance.

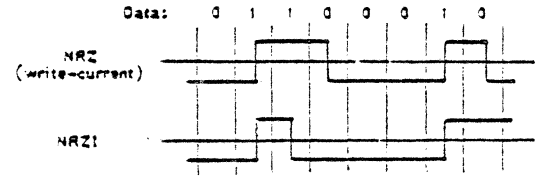
It would be unfair to use the same detection system for all of the codes, and so, in each case, an optimum detection system for the code under consideration is determined. The final answer is then presented as the maximum normalized data frequency achievable by the code, i.e. maximum packing factor. Note that $PF = PW_{50}/BP$, but for codes where data frequency does not equal reversal frequency, a distinction will be made between data bit period ($DBP$) and reversal bit period ($RBP$).

## 2 NRZ/NRZI (Non-Return-to-Zero and Non-Return-to-Zero Modified)

### 2.1 Coding Rules

(a) NRZ: Change direction of saturation at m.b.t. (mid-bit time) only if present bit $\not\equiv$ previous bit.

(b) NRZI: Change at m.b.t. only if bit is a '1'.



As both codes are able to produce the same waveforms, and differ only in such matters as error propagation, parity bit generation and, perhaps, ease of understanding, the subsequent analysis of packing density limit will be confined to just one of the codes, NRZI.

### 2.2 Rectify and Clip Detection System

#### 2.2.1 *Circuit description*

In this detection system, the signal is first amplified, then full-wave rectified, and then clipped to remove baseline noise, which would otherwise be a problem later on, in the squaring process. The noise present at the input to the linear amplifier will have come from several sources and a certain amount of this can be filtered out by correct choice of the frequency response of the linear amplifier, but a problem arises when the noise has components at frequencies lower than the maximum significant frequency in the 'basic pulse'. Attempted filtration of these components will result in integration of the data waveform, having the effect of increasing the $PW_{50}$ of the basic pulse. Since the slimness of the latter is all-important for efficient use of the recording unit, it is vital that no attempt is made to filter these components.

Experiments on a typical recording unit produced the result that, with as much filtering as possible employed, concurrent with no integration of the data, the signal-to-noise ratio is typically 20 : 1 at the output of the linear amplifier.

For the determination of the absolute limit of the code and detection method, no margins are allowed, and so the clip-level is set to the peak noise level, normalized, of 0·05. Note that this assumes zero baseline shift.

The next requirement is to detect the peaks in the waveform by converting them into zero-crossovers, using differentiation. This involves the subtraction of a delayed version of the signal from itself, and it has been shown[1,2] that an optimum value of delay is ~0·3 × $PW_{50}$.

The signal is next squared, using a comparator, to detect the zero-crossovers (z.x.o.s).

#### 2.2.2 *Packing density limit*

The packing density achievable using this detection system may be limited by timing considerations, amplitude considerations, or a mixture of the two. Considering first the timing; the worst-case pattern for peak-shift is two 1's, and this shift is shown in Fig. 1 as a function of frequency. It can be seen that the theoretical ideal frequency limit for NRZI with zero $RTE$ is 2·33, where the peak-shift equals 50%, the read resolution for this code. Further, by plotting another graph, of worst-case peak-shift (as above) + 8% $PW_{50}$ $RTE$, it is seen that the frequency limit for NRZI with a system $RTE$ of 8% $PW_{50}$ is 1·92. A similar calculation shows the timing limit for $RTE$ = 16%$PW_{50}$ to be at $PF$ = 1·62. Considering now the amplitude limit, this is where a peak in the read signal does not exceed the clip-level. The worst-case amplitude pattern at any packing density is three 1's, and is plotted in Fig. 2. It falls to the clip-level of 0·05 at $PF$ = 1·88.

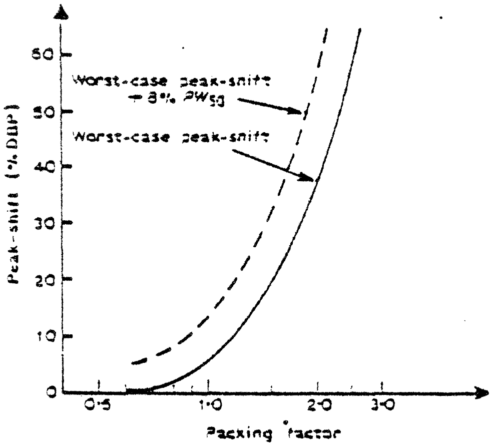


Fig. 1. Worst-case peak-shift for NRZI.

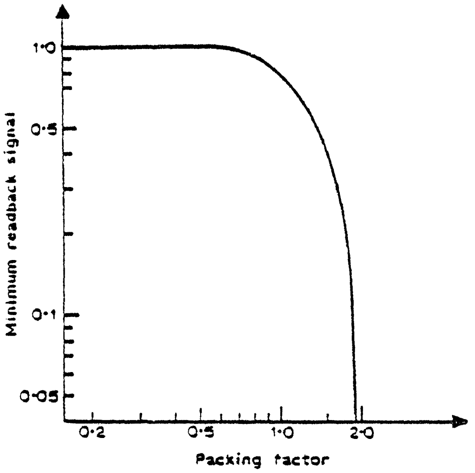


Fig. 2. Minimum read-back signal for NRZI.

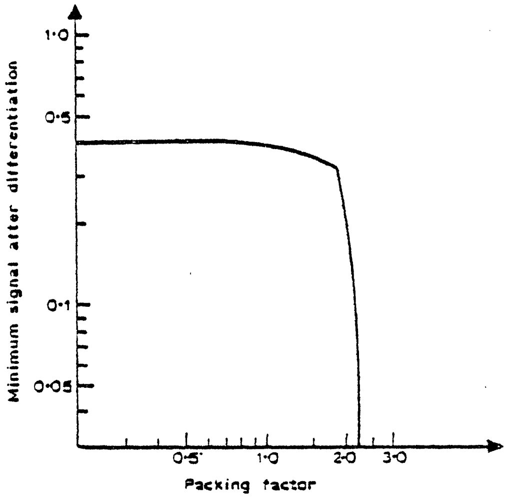


Fig. 3. Minimum signal after differentiation for NRZI.

### 2.2.3 *Summary*

The limiting frequency for each *RTE* allowance can now be summarized:

$RTE = 0$: Although the timing limit for this case is 2·33, an amplitude limit occurs first at 1·88.

$RTE = 8\% \, PW_{50}$: Again, although the timing does not limit performance until 1·92, breakdown occurs at 1·88, because of amplitude.

$RTE = 16\% \, PW_{50}$: Timing causes the breakdown in this case, at 1·62.

$RTE \propto \hat{IRT}$: Since $\hat{IRT} = \infty$, and $RTE = (6{\cdot}7 + 1{\cdot}3 \times IRT)\% \, PW_{50}$, then operation is impossible under these conditions, and the limiting frequency is zero.

## 2.3 Differentiate and Square Detection System

### 2.3.1 *Circuit description*

An alternative version of the previous detection method is to eliminate the clipping (and, therefore, the need to rectify), and to differentiate the amplified read-signal directly. Because of the large amount of noise present after differentiation, the squaring must then be biased considerably. This may be achieved by hysteresis round a comparator, so that the reference level is alternately +0·1 and −0·1 (i.e. the noise amplitude, allowing for doubling of the noise in the differentiator).

### 2.3.2 *Packing density limit*

The main disadvantage of this method is that, in the worst-case derivative waveform, there is an appreciable time delay between a zero cross-over and the 0·1 or −0·1 level, resulting in a timing error. Neglecting this for the moment, the amplitude limit will be where the derivative does not exceed the bias level of ±0·1. This worst-case derivative amplitude is plotted against frequency in Fig. 3.

The portion before the 'break-point' at 1·87 is due to the trailing edge of any pattern of consecutive 1's, whilst that after is due to the centre portion of a four 1's pattern. This latter breakdown features in many codes and so is shown in Fig. 4 in detail, at the amplitude limit for this method of 2·14. The low slope of the read-back waveform between the two inner 1's produces a very small peak in the derivative.

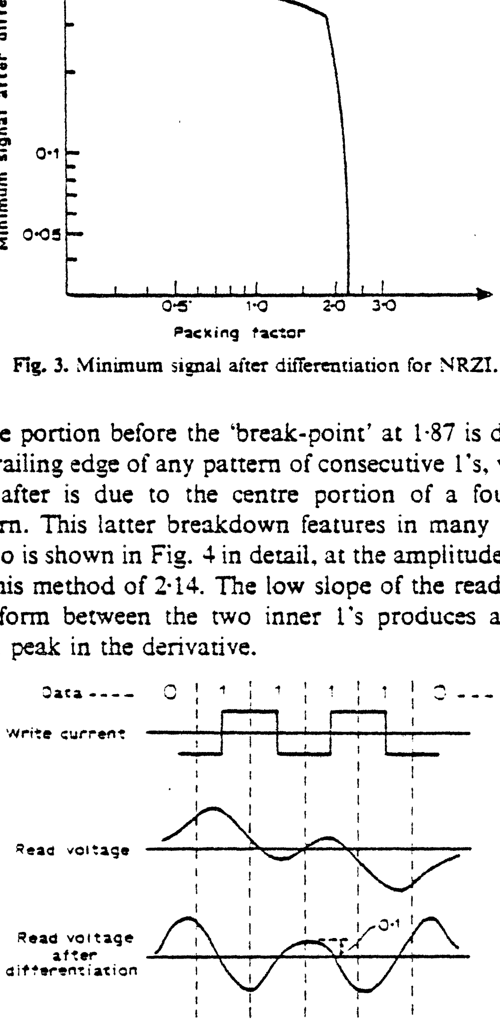


Fig. 4. Amplitude limit due to 'four-1's' pattern.

$RTE = 0$: On top of the normal timing loss due to worst-case peak-shift, a further timing loss is introduced by the biased squaring. At the amplitude limit of 2·14, this latter loss is ±17% *BP*, whilst the peak-shift on two 1's is ±43·5% *BP*. The total is therefore much greater than the permissible 50% *BP*. The timing limit is found to be at $PF = 2{\cdot}00$, where worst-case peak-shift is 37·5% *BP*, and the biased-squaring loss is 12·5% *BP*, a total of 50% *BP*. So, for $RTE = 0$, the limit is 2·00.

$RTE = 8\% \, PW_{50}$: At $PF = 1{\cdot}85$, worst-case peak-shift = 31·5% *BP*, $RTE = 14{\cdot}8\%$ *BP*, and squaring loss = 3·7% *BP*, a total of 50% *BP*. Thus the limit is 1·85.

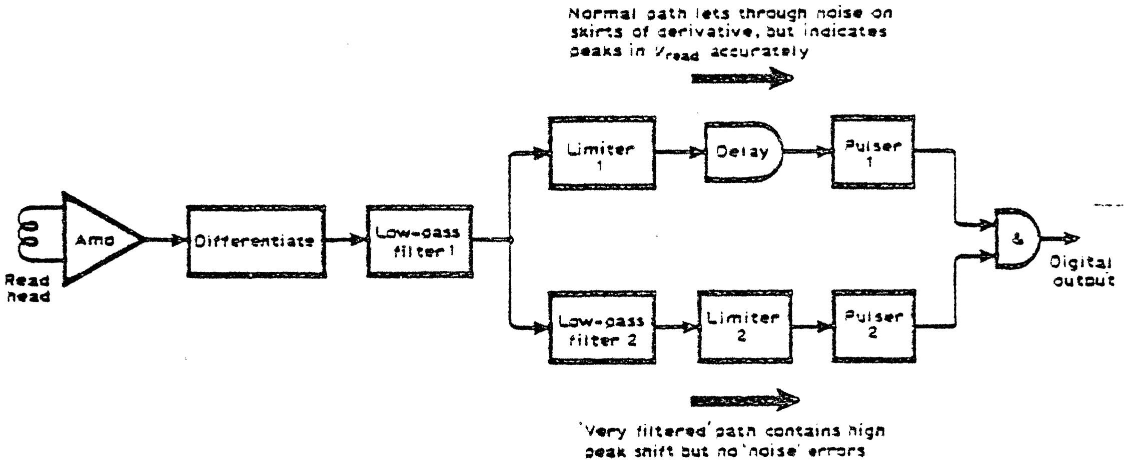

Normal path lets through noise on skirts of derivative, but indicates peaks in $V_{read}$ accurately
Read head
Amp
Differentiate
Low-pass filter 1
Limiter 1
Delay
Pulser 1
&
Digital output
Low-pass filter 2
Limiter 2
Pulser 2
'Very filtered' path contains high peak shift but no 'noise' errors

Fig. 5. Gated crossover detection method.

$RTE =$ 16% $PW_{50}$: A similar calculation shows the limit to be 1·60.

$RTE \propto \hat{IRT}$: As in the previous detection method, the limit here is zero.

### 2.3.3 *Summary*

The four packing factor limits for NRZI with this detection system are:

| | | |
|---|---|---|
| $RTE = 0$: | $\hat{PF} = 2{\cdot}00$ | All due to timing |
| $RTE = 8\%\ PW_{50}$: | $\hat{PF} = 1{\cdot}85$ | |
| $RTE = 16\%\ PW_{50}$: | $\hat{PF} = 1{\cdot}60$ | |
| $RTE \propto \hat{IRT}$: | $\hat{PF} = 0{\cdot}0$ | |

## 2.4 Gated Cross-over Detection System

### 2.4.1 *Circuit description*

The main disadvantage of the previous method is the timing loss introduced by the biased squaring. In the gated cross-over method, the zero cross-overs are used to indicate the peak positions in the read waveform, so that there is no timing loss, and the '0·1 cross-overs' are used to discriminate against noise on the baseline.

The method was first proposed by Dunstan and Whitehouse[3,4] and a similar system also appears to have been used by Tamura *et al.*[5] As shown in Fig. 5 two paths are formed after differentiation, one 'very filtered' path which contains high peak-shift, but no errors due to noise, and one unfiltered path which lets baseline noise through but indicates peaks in the readback waveform accurately. The two are then combined to give a noise-free, accurate, digital output.

### 2.4.2 *Packing density limit*

$RTE = 0$: The timing limit for this case is where the worst-case peak shift = 50% $BP$, i.e. at $PF = 2{\cdot}33$. However, an amplitude limit occurs where a signal peak after differentiation = 0·1, at $PF = 2{\cdot}14$. The limit is therefore at 2·14.

$RTE =$ 8% $PW_{50}$: The timing limit for this case is where (worst-case peak shift + jitter) = 50% $BP$, i.e. at 1·92. As the amplitude limit occurs at a higher frequency, 1·92 is the limit in this case.

$RTE =$ 16% $PW_{50}$: The limit is 1·62 due to timing.

$RTE \propto \hat{IRT}$: As before, operation is not possible in this case.

### 2.4.3 *Summary*

The four limits for this detection system are:

| | | |
|---|---|---|
| $RTE = 0$: | $\hat{PF} = 2{\cdot}14$ | due to amplitude |
| $RTE = 8\%\ PW_{50}$: | $\hat{PF} = 1{\cdot}92$ | due to timing |
| $RTE = 16\%\ PW_{50}$: | $\hat{PF} = 1{\cdot}62$ | |
| $RTE \propto \hat{IRT}$: | $\hat{PF} = 0$ | |

## 2.5 Pattern-adaptive Writing

### 2.5.1 *Basic technique*

This technique involves the modification of the timing of the write-current for certain specific data patterns. It can be used to trade-off amplitude against timing, or vice versa, depending on which is causing the most problems. As an example, in NRZI it has been shown that the peak-shift produced by a 'two-1's' pattern can become intolerable at high frequencies. Thus, in Fig. 6 it can be seen that by writing the two transitions at m.b.t., as is normal, the peaks in the read waveform are shifted, say 50% of a bit period away from the correct m.b.t. positions. If, however, this two-1's pattern is recognized before writing, and the two transitions are written closer than they would normally be, the peaks in the read waveform are found to be shifted less, relative to m.b.t., than in the normal method. The penalty, as can be seen, is a reduction in peak amplitude. Conversely, by writing the two transitions further apart than normal, an increase in amplitude can be obtained, at the expense of greater peak-shift. It should be noted that pattern

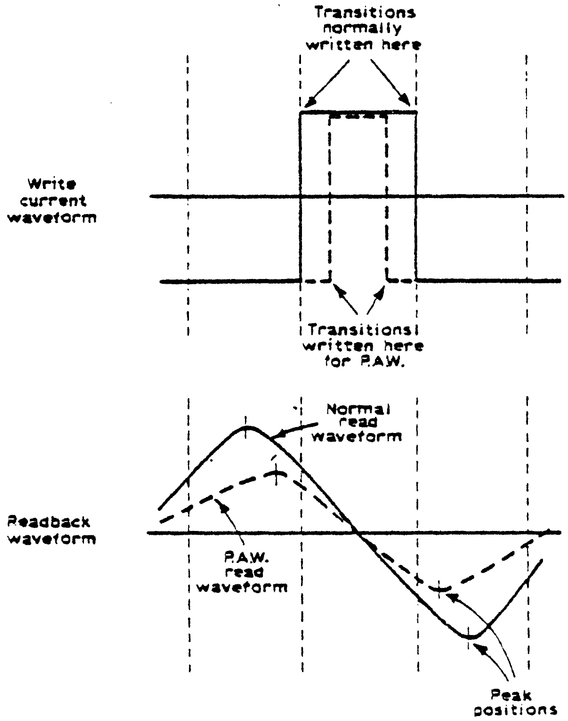
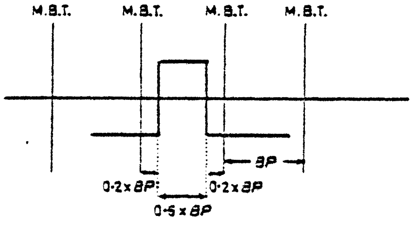


**Fig. 6.** Pattern adaptive write technique.

adaptive write (p.a.w.) techniques can be applied to any pattern, not just two-1's.

The notation used to describe the extent to which p.a.w. is being used will be as follows: if a transition is written a distance of $n\%$ $BP$ away from its normal position, and in such a direction as to produce less effective peak-shift than it would in the normal position, then $x\%$ p.a.w. is being used, where $x = (50-n) \times 2$. For the case where p.a.w. is being used to produce *more* peak-shift than normal $n$ will be negative, and $x$ will be greater than 100. For example, in the case shown below, 60% p.a.w. is being used.



It can be seen, therefore, that for the common case of 'two-1's', $x\%$ conveniently represents the separation of the two 1's, in terms of bit-periods. Note that 100% p.a.w. is equivalent to *no* p.a.w.

### 2.5.2 *Application to NRZI*

Consider the case of 8% $PW_{50}$ $RTE$ with the gated cross-over method. This breaks down at 1·92 because of excessive peak-shift in the two-1's pattern. It is necessary to know how much p.a.w. to apply to this pattern to take the achievable packing density up to the next limit, at 2·14, where the amplitude of the derivative of four-1's causes breakdown.

Allowable peak-shift at 2·14

$$\begin{aligned} &= 50\% BP - RTE \\ &= 50\%\ BP - 8\% \times 2{\cdot}14 \times BP \\ &= 32{\cdot}9\%\ BP \end{aligned}$$

or, in terms of $PW_{50}$,

allowable peak separation for two-1's

$$\begin{aligned} &= BP + 2 \times 0{\cdot}329 \times BP \\ &= 1{\cdot}66 \times BP \\ &= (1{\cdot}66 - 2{\cdot}14) \times PW_{50} \\ &= 0{\cdot}78 \times PW_{50}. \end{aligned}$$

The amount of p.a.w. necessary to produce this separation is most conveniently determined by the use of Fig. 7(a) which is a plot of written-transition separation against read-back-peak separation, for the two-1's pattern. Note that at low packing densities, i.e. written separation greater than approximately $1{\cdot}8 \times PW_{50}$, no interaction takes place between the two readback pulses; there is therefore no peak-shift, and so read-back separation = written separation. As the two transitions are written progressively closer, however, intersymbol interference produces peak-shift, and so read-back separation > written separation. In the limit, as the written separation tends to zero, the read-back separation tends to a finite limit of $0{\cdot}81 \times PW_{50}$.

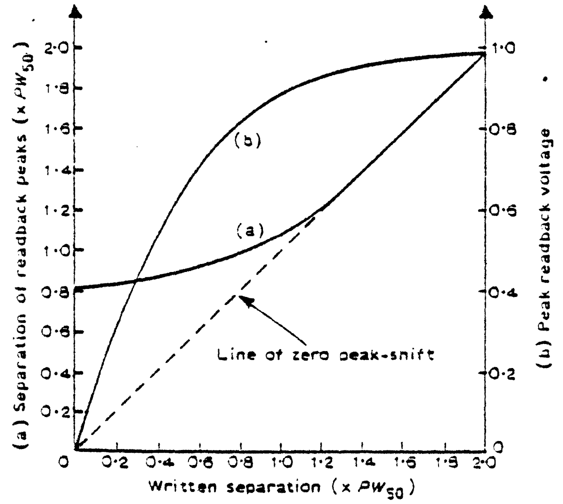


**Fig. 7.** Read-back separation and peak voltage for two isolated flux-reversals.

To see why this arises, we can consider that by placing a second 'isolated' pulse close to the first one, as we are doing, the effect is to differentiate the first pulse. As the separation decreases, more accurate differentiation is achieved, and so at the limit of zero separation, the ideal derivative is produced. This will have peaks at the points of maximum slope on the original pulse. These points occur at a distance of $0{\cdot}405 \times PW_{50}$ either side of the peak, thus agreeing with the asymptote of the graph.

The graph shows, therefore, that no amount of p.a.w. will allow NRZI to work at 2·14 with a $RTE$ of 8% $PW_{50}$, because a two-1's read-back separation of $0{\cdot}78 \times PW_{50}$ cannot be achieved. A theoretical timing limit occurs at 2·04 when 0% p.a.w. is used, though of course the peak amplitude of the read-back waveform is zero as shown in Fig. 7(b), which plots peak amplitude against peak separation, and operation under such circumstances is impossible.

One further point is that the peak-shift on the end 1's of three-1's, four-1's, five-1's, or, indeed, any number of 'isolated' ones, is nearly as bad as the absolute worst-case two-1's pattern. This means that if p.a.w. was used in a system, the criterion used would probably be: apply p.a.w. to any 1 if it has three or more 0's on one side of it, and one or more 1's on the other side. To determine the practical limit for p.a.w. using the gated crossover detection system, therefore, involves the simultaneous study of both amplitude and timing effects for all patterns, at all frequencies, and with all extents of p.a.w.

### 2.5.3 *Summary*

The result of such an analysis is that p.a.w. can only produce a practical packing density increase from 1·92 to 2·00, for 8% $PW_{50}$ $RTE$. This is an increase of only 4%. For 16% $PW_{50}$ $RTE$, the improvement is even less. The conclusion is, therefore, that p.a.w. is not worthwhile as a means of increasing packing density for NRZI.

### 2.6 Summary of NRZ/NRZI Systems

| $RTE$ | Rectify and clip limit | Differentiate and square limit | Gated cross-over limit |
|---|---|---|---|
| 0 | 1·88 | 2·00 | 2·14 |
| 8% $PW_{50}$ | 1·88 | 1·85 | 1·92 |
| 16% $PW_{50}$ | 1·62 | 1·60 | 1·62 |

## 3 Enhanced NRZI

It should be noted that enhancement can also be applied to NRZ, but, as before, NRZI only will be discussed.

### 3.1 Coding Rules

Type A. Code as in NRZI, but after every $n$ bits include one compulsory '1'.

Type B. Code as in NRZI, but after every $n$ bits include one odd-parity bit.†

### 3.2 Description

In this code, it is convenient to make a distinction between recorded-bit-frequency, or flux-reversal frequency, and average data frequency, since for every $n+1$ bits recorded, only $n$ of them are data bits. Thus, we have data frequency $= n/(n+1) \times$ reversal frequency, and since data bit period $DBP = PW_{50}/$data frequency, and reversal bit period $RBP = PW_{50}/$reversal frequency, then $DBP = RBP \times (n+1)/n$.

The purpose of this code is to utilize the good qualities of NRZI, i.e. ±50% read resolution, absence of 'double frequency' components, and ease of coding/decoding, whilst removing its main disadvantage, its inability to provide self-clocking. With enhanced NRZI (ENRZI), by appropriate choice of $n$, any required degree of self-clocking can be obtained, using the following formulae:

for type A:

$$I\hat{R}T = (n+1) \times RBP = n \times DBP$$

for type B:

$$I\hat{R}T = (2n+1) \times RBP = [(2n+1)n/(n+1)] \times DBP.$$

For all values of $n$, but $n = 1$, both types of ENRZI exhibit effectively the same worst-case patterns as NRZI, and will break down, therefore, at the same reversal frequency as NRZI, and, more importantly, at the same reversal and data frequency as each other. Thus, given any value of $n$ other than 1, both types of ENRZI will break down at the same data frequency, whilst type A will exhibit an $I\hat{R}T$ typically half that of type B, and will therefore be better at self-clocking. Detailed analysis of worst-case patterns shows that for $n = 1$, type A is also superior in terms of achievable data frequency, as well as in terms of self-clocking ability.

It can be seen that although type B assists error detection by virtue of the fact that the compulsory bit is a parity bit, it is never better than type A as regards achievable packing density, and is always worse at providing self-clocking. For this reason, type A is considered superior to type B, and the former alone will now be analysed in detail.

### 3.3 Packing Density Limit

### 3.3.1 $n \geq 2$

For $n \geq 2$, ENRZI exhibits effectively the same worst-case patterns as NRZI, and, therefore, if the gated crossover detection method is used, the packing density limits can be easily calculated by applying the dilution factor to the corresponding NRZI limit.

(a) $RTE = 0$. For NRZI the limit here is 2·14. Thus for ENRZI the limits are:

| $n$ and $I\hat{R}T$ ($\times DBP$) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | $x$ | $\infty$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{PF}$ | 1·43 | 1·61 | 1·71 | 1·78 | 1·83 | 1·87 | 1·90 | 1·92 | 1·94 | $\dfrac{2{\cdot}14x}{(x+1)}$ | 2·14 |

† It can be easily shown that, for NRZ, a further stipulation is that $n$ must be odd, to ensure a finite $I\hat{R}T$.

(b) $RTE = 8\%\ PW_{50}$. For NRZI the limit here is 1·92, and so for ENRZI the limits are:

| $n$ and $I\hat{R}T$ ($\times DBP$) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | $x$ | $\infty$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{PF}$ | 1·28 | 1·44 | 1·54 | 1·60 | 1·65 | 1·68 | 1·71 | 1·73 | 1·75 | $\frac{1{\cdot}92x}{x+1}$ | 1·92 |

(c) $RTE = 16\%\ PW_{50}$. For NRZI, the limit here is 1·62, and so for ENRZI the limits are:

| $n$ and $I\hat{R}T$ ($\times DBP$) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | $x$ | $\infty$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{PF}$ | 1·08 | 1·22 | 1·30 | 1·35 | 1·39 | 1·42 | 1·44 | 1·46 | 1·47 | $\frac{1{\cdot}62x}{x+1}$ | 1·62 |

(d) $RTE \propto I\hat{R}T$. For this case, it is not possible to work straight from the NRZI figure, as there is no exact parallel between the two codes. For each value of $n$, therefore, the procedure for determining the data frequency limit is as follows:

e.g. for $n = 3$:

$$RTE = (6{\cdot}7 + 1{\cdot}3 \times I\hat{R}T/DBP)\%\ PW_{50}$$
$$= (6{\cdot}7 + 3{\cdot}9)\%\ PW_{50}$$

therefore

$$RTE = 10{\cdot}6\%\ PW_{50}.$$

The timing limit occurs where ($RTE$ + worst-case peak-shift) = 50% $RBP$. From the graph of worst-case NRZI peak-shift, this occurs at 1·81, where (peak-shift $+RTE$) = 30·8% $RBP$ + 19·2% $RBP$ = 50% $RBP$.

As this occurs before the amplitude limit at 2·14, the reversal-frequency limit is 1·81.

Therefore data frequency limit

$$= n/(n+1) \times \text{reversal frequency limit}$$
$$= (\tfrac{3}{4}) \times 1{\cdot}81.$$

Hence for $n = 3$, data frequency limit = 1·36.

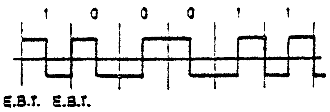
By a similar process, the corresponding limits for other values of $n$ can be found, resulting in:

| $n$ and $I\hat{R}T$ ($\times DBP$) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 34 | $\infty$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{PF}$ | 1·23 | 1·36 | 1·42 | 1·44 | 1·44 | 1·47 | 1·40 | 1·40 | 1·37 | 1·16 | 0 | 0 |
| $RTE$ (% $PW_{50}$) | 9·3 | 10·6 | 11·9 | 13·2 | 14·5 | 15·8 | 17·1 | 18·4 | 19·7 | 32·7 | 50·1 | $\infty$ |

It is interesting to note that, for this $RTE$ case, maximum packing density is achieved by using $n = 7$, which is the version most often used in practice.

### 3.3.2 $n = 1$

ENRZI with $n = 1$ produces the well-known frequency modulation code (FM), the usual definition of which is: always change at m.b.t., but if the bit is a '1', change at e.b.t. also, e.g.:



Another very similar code is phase modulation code (PM), and other names for the two codes include 'Manchester', Biphase and FSK, but, as with NRZI and NRZ, there is no need to consider both PM and FM in detail, as they produce exactly the same waveform sets. Several detection techniques were applied to FM, including gated crossover, polarity strobing at m.b.t., polarity strobing at e.b.t., and derivative polarity strobing at t.q.b.t. (three-quarter bit-time).

The exceedingly simple detection method of strobing polarity at m.b.t. proves to be superior to all others considered, yielding the following figures:

| | | |
|---|---|---|
| (i) | for $RTE = 0$: | $\hat{PF} = 1{\cdot}17$; |
| (ii) | for $RTE = 8\%\ PW_{50}$ or $RTE \propto I\hat{R}T$ : | $\hat{PF} = 1{\cdot}15$; |
| (iii) | for $RTE = 16\%\ PW_{50}$: | $\hat{PF} = 1{\cdot}09$. |

## 4 Modified NRZI

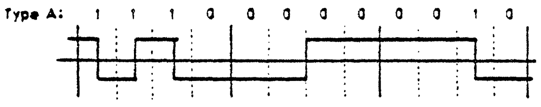
As before, modification can also be applied to NRZ, but NRZI only will be considered.

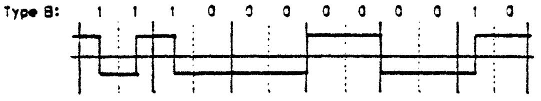
### 4.1 Coding Rules

Type A. Code groups of $2n$ bits at a time, and code each group as in NRZI, except code a group of all 0's as a change at mid-group time.

Type B. Code groups of $n$ bits at a time, as in NRZI, but code a group of all 0's followed by the same as a change on the junction of the two groups.

For example $n = 2$:





### 4.2 Description

Like ENRZI, modified NRZI (MNRZI) is an attempt to produce a self-clocking version of NRZI, in this case by breaking up long runs of zero's with an occasional e.b.t. (end bit time) pulse. It might appear at first that by inserting e.b.t. pulses in isolated areas, in this manner, a self-clocking code will be produced without affecting the normal operation of NRZI, and so the code will work to

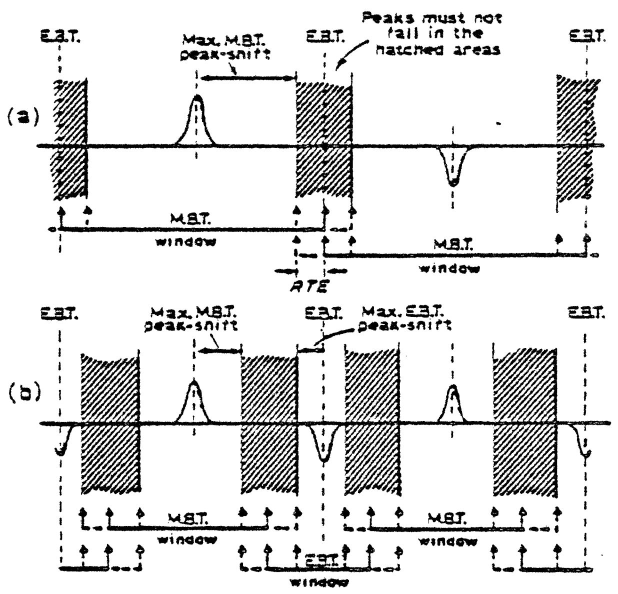


Fig. 8. Reduction of margin by adding e.b.t. transitions to NRZI.

the same limits as NRZI. Unfortunately, this is not the case, as can be seen with reference to Fig. 8. (a) signifies NRZI with a m.b.t. window which has non-zero jitter (*RTE*). The window has a width of *BP*, and the jitter, or *RTE*, is effectively shared between adjacent windows, so that the peak-shift allowable on the m.b.t. pulse = (50% *BP* − *RTE*). In (b), which depicts any code that contains both m.b.t. and e.b.t. pulses, the jitter is no longer shared by adjacent m.b.t. windows, but by adjacent m.b.t. and e.b.t. windows (whether or not there is an actual e.b.t. window). It is clear that the allowable peak-shift on the m.b.t. pulse is now (50% $BP - 2 \times RTE -$ max. e.b.t. peak-shift). So even if the e.b.t. shift is negligible, the allowable m.b.t. peak-shift is *RTE* less than for 'straight' NRZI.

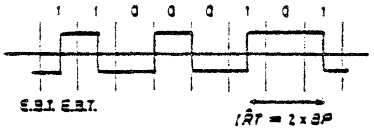
From this it can be expected that for very small *RTE* values, MNRZI will closely resemble NRZI in achievable packing density, the exact proximity of the two depending on the extent of self-clocking required. As the *RTE* figure increases, however, the MNRZI performance will rapidly decrease, showing MNRZI to be not a worthwhile code for high *RTE* values. This will be borne out in the ensuing detailed analyses, in which the various MNRZI codes are referred to as MNRZI$_{xn}$, where $x$ is either 'A' or 'B', indicating the type, and $n$ is as defined in the coding rules.

### 4.3 MNRZI$_{31}$

#### 4.3.1 *Coding rules*

This code is the well-known Modified Frequency Modulation, described by Padalino[6] (attributed by Padalino to Pouliart[7]), but also described by Woo.[3] It is also known as Miller Code, Delay Modulation (DM), and, in a slightly different form, Modified Phase Modulation (MPM).

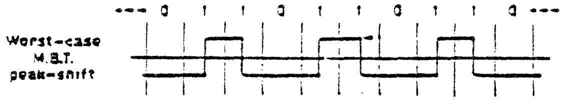
The coding rules are: change at m.b.t. for a 1, and change at e.b.t. between two 0's. For example:

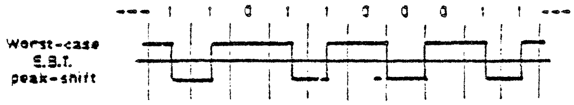


#### 4.3.2 *Gated crossover detection*

As shown in Section 4.2, a timing limit occurs for MNRZI when (worst-case m.b.t. peak-shift + 2 × *RTE* + worst-case e.b.t. peak-shift) = 50% *BP*.

For Miller code the two worst-case patterns are shown below:





The graphs of peak-shift against packing factor for the two patterns are shown in Fig. 9, from which the timing limit for any value of *RTE* can be found:

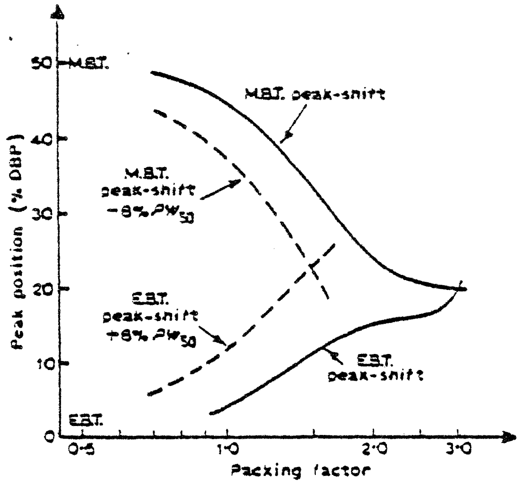
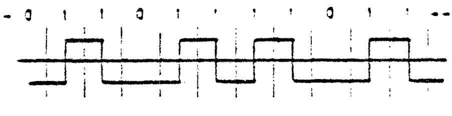


Fig. 9. Worst-case peak-shift for Miller code.

(a) *RTE* = 0: The intercept of the two curves gives directly the timing limit for zero *RTE* as 2·95. An amplitude limit will occur when the peak amplitude of the derivative is equal to the noise level, i.e. 0·1. This occurs at 2·14 for the four-1's-type pattern:



Thus, for zero *RTE*, DM breaks down at 2·14, due to amplitude.

(b) $RTE = 8\%\ PW_{50}$: The timing limit for this value of $RTE$ is obtained by replotting the two curves to include the $RTE$, as shown. The intercept of the two, at 1·47, is the timing limit. As the amplitude limit is unchanged at 2·14, then for 8% $PW_{50}$ $RTE$, DM breaks down at 1·47.

(c) $RTE = 16\%\ PW_{50}$: By replotting the curves (though this is not shown, for clarity), the timing limit is found to be 1·12. As this is below the amplitude limit, breakdown is at 1·12.

(d) $RTE \propto I\hat{R}T$: For DM,

$$I\hat{R}T = 2 \times BP.$$

$$RTE = (6{\cdot}7 + 2 \times 1{\cdot}2)\%\ PW_{50} = 9{\cdot}3\%\ PW_{50}.$$

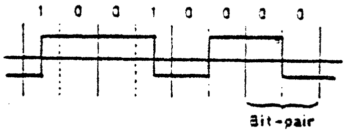
The timing limit is found to be 1·4 and as this is below the amplitude limit, DM breaks down at 1·4, for the fourth $RTE$ case.

## 4.4 MNRZI$_{A1}$

### 4.4.1 *Coding rules*

Code bits in pairs, and code as in NRZI, except code a pair of 0's as a change at mid-pair-time (m.p.t.).

This yields an $I\hat{R}T$ of $3 \times BP$, as in the following pattern:



### 4.4.2 *Packing density limit*

Analysis of the worst case peak-shift and amplitude patterns yields the following results:

| | |
|---|---|
| $RTE = 0$: | $P\hat{F} = 1{\cdot}97$ |
| $RTE = 8\%\ PW_{50}$: | $P\hat{F} = 1{\cdot}47$ |
| $RTE = 16\%\ PW_{50}$: | $P\hat{F} = 1{\cdot}15$ |
| $RTE \propto I\hat{R}T = 10{\cdot}6\% \times PW_{50}$: | $P\hat{F} = 1{\cdot}36.$ |

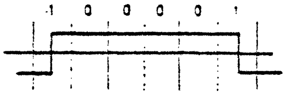
All the limits are due to timing, as the amplitude limit (with the gated crossover detection method) is at 2·14.

## 4.5 MNRZI$_{B2}$

### 4.5.1 *Coding rules*

Code bits in pairs, and code pairs as in NRZI, except code a pair of 0's followed by the same as a change on the boundary of the two pairs.

The maximum inter-reversal time is $5 \times BP$, as in the pattern:



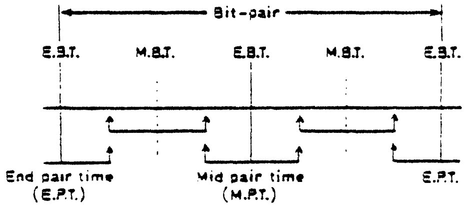
### 4.5.2 *Packing density limit*

| | |
|---|---|
| $RTE = 0$: | $P\hat{F} = 2{\cdot}06$ |
| $RTE = 8\%\ PW_{50}$: | $P\hat{F} = 1{\cdot}57$ |
| $RTE = 16\%\ PW_{50}$: | $P\hat{F} = 1{\cdot}22$ |
| $RTE \propto I\hat{R}T = 13{\cdot}2\%\ PW_{50}$: | $P\hat{F} = 1{\cdot}30.$ |

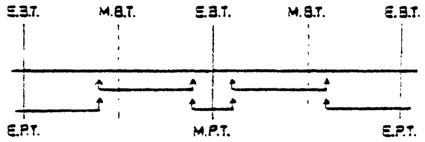
All limits are again due to timing, as the amplitude breakdown (gated crossover detection) is 2·14.

### 4.5.3 *Window modification*

In this code, for the first time, a slightly different method of arranging the m.b.t. and e.b.t. windows is possible. In the normal method, the windows are exactly the same for each bit, regardless of its position in the pair, thus:



However, if (i) the worst-case peak-shift from m.b.t. → e.p.t. is not the same as that from m.b.t. → m.p.t., and (ii) e.p.t. → m.b.t. is different to m.p.t. → m.b.t., then the windows can be different for each bit in the pair, e.g.:

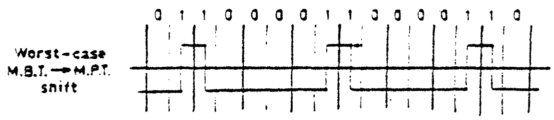


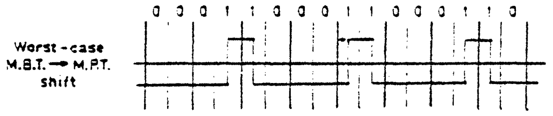
The structure just drawn is the one that could be applied to this code, because:

(a) There is no m.p.t. pulse, so condition (ii) is met, and

(b) The worst-case m.b.t. → m.p.t. shift is greater than that from m.b.t. → e.p.t., so condition (i) is satisfied.

Unfortunately, however, the two shifts mentioned in (b) above are very close to each other, as might be guessed from the patterns producing them:





Thus, the increase in performance possible in this case is found to be only 2% (up from 1·57 to 1·60) (for the 8% $PW_{50}$ $RTE$ case), and is not considered worthwhile.

### 4.6 Summary

The performance of all the MNRZI codes possible is summarized in Fig. 10. The results verify the original postulate that MNRZI would not be worthwhile for high *RTE* values. Even for the 8% $PW_{50}$ *RTE* allowance, the best MNRZI code will only work up to 1·65, where (two-1's peak-shift + 2 × *RTE*) = 50% *BP*. If no e.b.t. pulses have to be allowed for, as in NRZI, the limit is 1·92 where (two-1's peak-shift + 1 × *RTE*) = 50% *BP*.

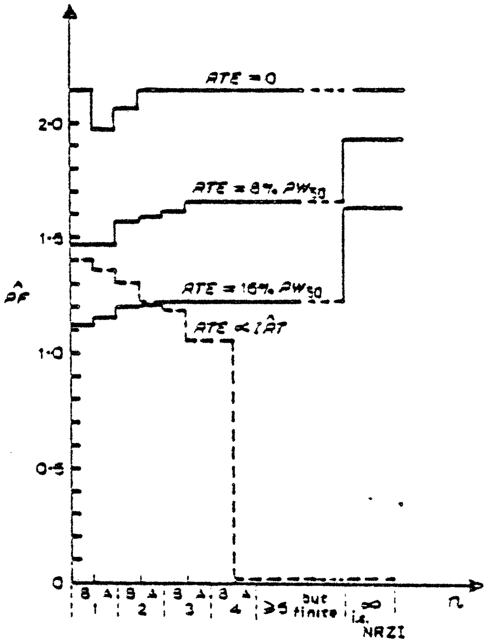


Fig. 10. MNRZI performance summary.

One important conclusion is that if $RTE \propto I\check{R}T$, no trade-off is necessary, as $\text{MNRZI}_{B1}$ (Miller) provides both the highest achievable frequency and the lowest $I\check{R}T$, showing why Miller code is a very popular code in practice.

## 5 Group Codes

Codes such as NRZ, NRZI, PM and FM operate on the principle of 'one symbol for one bit'. There is no reason, however, why codes should not be constructed whereby groups of bits are coded with unique patterns. Such codes are called group codes, and indeed, MNRZI could be viewed as one.

A similar class of codes is adaptive codes, whereby groups are viewed, as before, but the waveform symbol depends not only on the group presently being coded, but also on previous and subsequent groups. The distinction between the two classes is vague, however, and both will herein be referred to as group codes.

Franaszek's paper[9] is an attempt to provide a means of producing optimum group codes, given the constraints of minimum and maximum inter-reversal times.

The symbols used by Franaszek are:

$N$ = number of equal subdivisions per bit.
= number of possible flux-reversal positions per bit (e.g. for FM, $N = 2$).
$d$ = minimum number of empty flux-reversal positions between two flux-reversals.
$k$ = maximum number of empty flux-reversal positions between two flux-reversals: thus

$$I\check{R}T = [(d+1)/N] \times BP$$

and

$$I\hat{R}T = [(k+1)/N] \times BP.$$

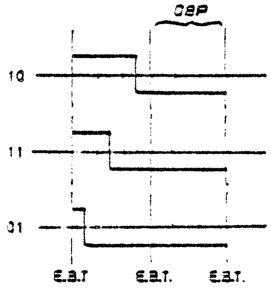
$M$ = maximum number of bits required for coding at any one time (e.g. for FM, $M = 1$: but for Watson code, $M = 2$).

This notation will be used for all the group codes presented here.

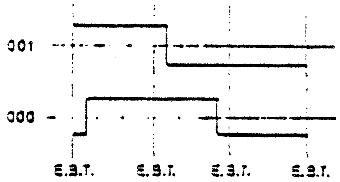
### 5.2 Group Code (4, 9)

#### 5.2.1 *Coding rules*

This code is given by Franaszek as an example of a run-length limited code. Coding commences by the viewing of the first two data bits. If these are anything other than a pair of 0's, the waveforms shown below are applied.



If, however, the pair is 00, the third bit is viewed also, and the three bits together are then coded thus:



The parameter values are thus:

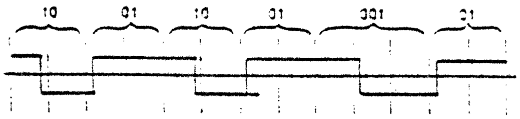
$$N = 3, \quad d = 4, \quad k = 9, \quad M = 3$$

and

$$I\hat{R}T = 3{\cdot}33 \times BP. \quad I\check{R}T = 1{\cdot}67 \times BP.$$

#### 5.2.2 *Packing density limit*

The read resolution for this code is only ±16·7% *BP*. The worst-case peak-shift pattern is:



(a) $RTE = 0$: The peak-shift equals 16·7% at $PF = 1{\cdot}70$. There is no problem with the amplitude with any detection system at this frequency, so breakdown is at 1·70.
(b) $RTE = 8\%\ PW_{50}$: The timing limit here is at 1·30.
(c) $RTE = 16\%\ PW_{50}$: This gives a timing limit of 1·06.
(d) $RTE \propto I\hat{R}T$: Since $I\hat{R}T = 3{\cdot}33 \times BP$, $RTE = 11{\cdot}0\%\ PW_{50}$, and this breaks down at 1·20.

### 5.3 Rice Code

#### 5.3.1 *Coding rules*

The first six bits are viewed initially. If they are '010101', they are coded as shown in Fig. 11; otherwise the last two bits are returned to the input stream for now. If the remaining bits commence with '11', then all four are coded, as shown. Otherwise, the last two bits are returned to the input stream, and the remaining two bits are coded as shown. The process is then repreated, starting with the next six bits in the input stream (including returned bits).

The parameters are:

$$N = 1\tfrac{1}{2}, \quad d = 1, \quad k = 11, \quad M = 6$$

and

$$I\hat{R}T = 8 \times BP, \quad I\check{R}T = 1{\cdot}33 \times BP.$$

The read resolution is ±33·3% *BP*.

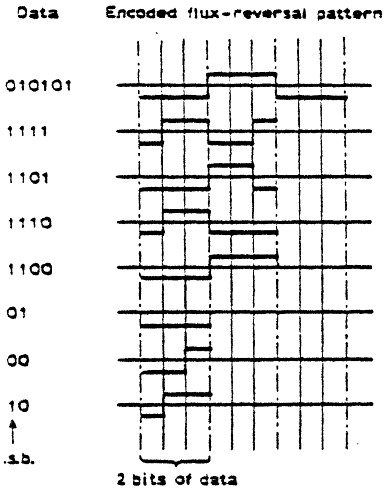


Fig. 11. Coding rules for Rice code.

#### 5.3.2 *Packing density limit*

The worst-case peak-shift pattern is shown in Fig. 12(a). An amplitude limit occurs for a four-1's-type pattern, shown in Fig. 12(b), at 2·84, assuming a gated-crossover detection system.

(a) $RTE = 0$: The timing limit is where worst-case peak-shift = $33\tfrac{1}{3}\%$ *BP*, at $PF = 2{\cdot}23$. This occurs before the amplitude limit, and therefore breakdown is at 2·23.

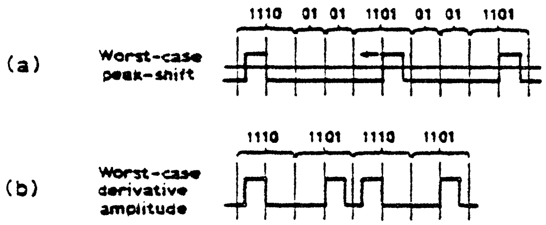


Fig. 12. Rice code worst-case patterns.

(b) $RTE = 8\%\ PW_{50}$: The timing limit for this case, and therefore breakdown, occurs at 1·77.
(c) $RTE = 16\%\ PW_{50}$: This gives a timing limit at 1·47.
(d) $RTE \propto I\hat{R}T$: Since $I\hat{R}T = 8 \times BP$, $RTE = (6{\cdot}7 + 1{\cdot}3 \times 8)\%\ PW_{50} = 17{\cdot}1\%\ PW_{50}$. The timing limit for this is at 1·41.

#### 5.3.3 *Summary*

This code, used by Digital Development Operation, is obviously of considerable interest as it provides a performance comparable with that of NRZI, whilst allowing some degree of self-clocking. Because of the large difference between the amplitude and timing limits, it might appear prudent to use p.a.w. However, because of the complexity of the code, it is difficult to decide where exactly to apply the p.a.w. An alternative is pulse-slimming, which may well improve its performance even further.

### 5.4 Gabor Code

#### 5.4.1 *Coding rules*

This code, proposed by Gabor,[10] is a very complicated adaptive code. Bits are viewed in groups of two, and each double bit-period is subdivided into three parts, each of which is a possible flux reversal position.

The notation used is as follows:

| | Data bits | Code bits |
|---|---|---|
| Preceding bit pair | $B_{1p}\ B_{2p}$ | $P_{1p}\ P_{2p}\ P_{3p}$ |
| Present bit pair | $B_1\ B_2$ | $P_1\ P_2\ P_3$ |
| Following bit pair | $B_{1f}\ B_{2f}$ | $P_{1f}\ P_{2f}\ P_{3f}$ |

The code is constructed so as to obey:

$$I\hat{R}T = 4/3 \times BP; \quad I\check{R}T = 2/3 \times BP.$$

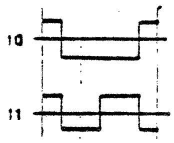
The formal encoding rules are:

$$P_1 = \overline{P_{3p}} + B_1 + \overline{B}_2 B_{1f}$$
$$P_2 = P_{3p}\overline{B}_1 + B_2$$
$$P_3 = \overline{P_{3p}} + B_1 + B_2$$

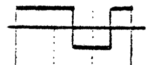
This is most easily visualized as:

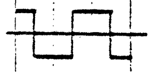
*If* the bit pair is '10' *or* '11'

*then* code as follows:

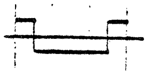


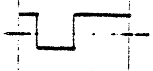
*else if* the bit pair is '01';

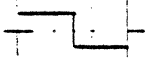
*then if* $P_{3p}$ code as: → 

*or if* $\overline{P_{3p}}$ code as: → 

'00' is the awkward case, and is coded as:

*if* $\overline{P_{3p}}$: → 

*if* $P_{3p}$ *and* $B_{1f}$: → 

*if* $P_{3p}$ *and* $\overline{B_{1f}}$: → 

The decoding rules are:

$$B_1 = [\overline{P_{3p}}P_{1p} + P_{3p}P_1P_3]; \quad B_2 = [P_2P_3].$$

### 5.4.2 *Packing density limit*

The read resolution for this code is again 33·3% *BP*, and analysis of worst-case patterns yields the following results:

(a) $RTE = 0$: A peak-shift of 33·3% is never reached, so breakdown is due to amplitude at 1·45.

(b) $RTE = 8\%\ PW_{50}$: A timing limit now occurs at 1·75 but breakdown is still due to amplitude at 1·45.

(c) $RTE = 16\%\ PW_{50}$: Timing now causes breakdown at 1·15.

(d) $RTE \propto I\hat{R}T$: Since $I\hat{R}T = 1{\cdot}33 \times BP$, $RTE = 8{\cdot}43\%\ PW_{50}$, and breakdown is at 1·45 due to amplitude, as the timing limit is at 1·7.

In such a complex code as this, p.a.w. would be difficult to apply.

## 5.5 Octal Coded Binary

### 5.5.1 *Coding rules*

In this code, data are coded in groups of three. The waveform sets for octal-coded-binary (OCB) code are shown in Fig. 13. As in all the other çodes presented here, inverses of waveforms are non-distinct, and are used to ensure that $I\hat{R}T = 4{\cdot}2 \times BP$, subject to $I\hat{R}T = 1{\cdot}2 \times BP$.

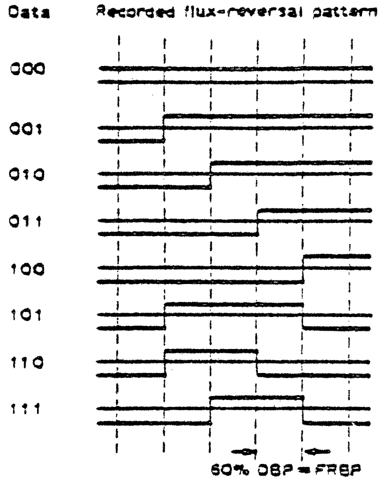


Fig. 13. Waveforms for octal-coded-binary.

### 5.5.2 *Packing density limit*

(a) $RTE = 0$: The worst-case peak-shift curve shows that 30% *DBP* (the read-resolution for this code) occurs at 2·03. As the amplitude limit is at 2·56, breakdown is at 2·03 due to timing.

(b) $RTE = 8\%\ PW_{50}$: This breakdown occurs at 1·62, again due to timing.

(c) $RTE = 16\%\ PW_{50}$: Breakdown here is at 1·30.

(d) $RTE \propto I\hat{R}T$: Since $I\hat{R}T = 4{\cdot}2 \times DBP$, $RTE = 12{\cdot}16\%\ PW_{50}$. With this *RTE* figure, the new (timing) breakdown is at 1·45.

### 5.5.3 *Pattern adaptive write*

Because the amplitude limit is significantly higher than the timing limit, it might be expected that p.a.w. could be profitably used. However, because the timing limit is quite high anyway, very little reduction in peak-shift can be obtained, even with large amounts of p.a.w. The increase obtainable is in fact only ~5%.

## 5.6 'GCR'

### 5.6.1 *Coding rules*

In this code, also known as '4/5 code' and analysed also by Tamura *et al.*,[5] four data bits are represented by a five-bit pattern. The constraints placed upon the code are that

$$I\hat{R}T = 2{\cdot}4 \times DBP\ (= 3 \times FRBP)$$

and

$$I\check{R}T = 0{\cdot}8 \times DBP\ (= 1 \times FRBP).$$

From the 32 possible combinations of 5 bits, 15 can be eliminated because of these constraints, leaving 17, from which one can be discarded to produce the 16 unique patterns required. This one can then be used as a special pattern, for checking or error detection, as it obeys the constraints and is therefore detectable.

### 5.6.2 *Packing density limit*

Considering the gated crossover type, amplitude breakdown will occur at $PF = 1{\cdot}74$ for a four-1's type pattern. The read resolution of this code is $50\%\ FRBP = 40\%\ DBP$, so:

(a) $RTE = 0$: The timing limit occurs at $PF = 2{\cdot}12$. As the amplitude limit occurs earlier, however, the code breaks down at 1·74.

(b) $RTE = 8\%\ PW_{50}$: The modified graph shows that the new timing limit is at 1·57, and as this is below the amplitude limit, breakdown therefore occurs at 1·57.

(c) $RTE = 16\%\ PW_{50}$. Breakdown here is at 1·31, against due to timing.

(d) $RTE \propto I\hat{R}T$. Since

$$I\hat{R}T = 3 \times FRBP = 3 \times 0{\cdot}8 \times DBP$$
$$= 2{\cdot}4 \times DBP,$$

then

$$RTE = (6{\cdot}7 + 1{\cdot}3 \times 2{\cdot}4)\%\ PW_{50}$$
$$= 9{\cdot}82\%\ PW_{50}.$$

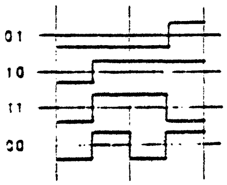
This timing limit occurs at $PF = 1{\cdot}50$, and this is where breakdown occurs.

## 5.7 Watson Code

### 5.7.1 *Coding rules*

Code bits in pairs, and code as in NRZI, except code a pair of 0's as a change at both m.b.t.s *and* a change at mid-pair-time (m.p.t.).

Thus the four possible patterns for a pair are:



In this code, proposed by Watson,[11] runs of zeros are broken up by the use of a unique flux-reversal pattern, which the originator hopes will be still easily identifiable at high packing densities.

### 5.7.2 *Packing density limit*

Using a slightly modified gated crossover detection method, an amplitude limit occurs at 1·30 due to the inherent three-1's-type pattern of the '00' bit pair.

(a) $RTE = 0$: A timing limit occurs at 2·33, where the peak-shift = $50\%\ BP$. However, amplitude causes an earlier limit at 1·30.

(b) $RTE = 8\%\ PW_{50}$: Although the timing limit is at 1·92, amplitude again causes breakdown at 1·30.

(c) $RTE = 16\%\ PW_{50}$: Timing limit = 1·65, but amplitude limit = 1·30.

(d) $RTE \propto I\hat{R}T$: $I\hat{R}T = 3 \times BP$, $RTE = 10{\cdot}6\%\ PW_{50}$, giving a timing limit at 1·82, but amplitude fails at 1·30.

### 5.7.3 *P.A.W.*

This code provides an opportunity for 'reverse' p.a.w., because of its early amplitude breakdown. If the two outer flux-reversals in the '00' pattern are written further away from the centre one, the timing margin will be reduced, but the amplitude of the centre peak will increase. Note that this is not strictly p.a.w., but merely a modification of the coding rules. For each separate $RTE$ allowance, the amount of 'p.a.w.' can be optimized to ensure that the timing limit and the amplitude limit occur simultaneously. With the gated crossover detection method, this yields:

(a) $RTE = 0$: By using $-40\%$ p.a.w., on the outer peaks of the '00' pattern only, the frequency limit becomes 1·60, where the shift on one of these peaks takes it to e.b.t., and, simultaneously, the four-1's type pattern breaks down on amplitude.

(b) $RTE = 8\%\ PW_{50}$: The optimum in this case is $-70\%$ p.a.w., causing simultaneous timing and amplitude breakdowns at 1·50.

(c) $RTE = 16\%\ PW_{50}$: In this case, the limit is at 1·36, with $-90\%$ p.a.w.

(d) $RTE \propto I\hat{R}T$: With $-80\%$ p.a.w., this breakdown is at 1·42.

## 5.8 Miller$^2$ Code

### 5.8.1 *Coding rules*

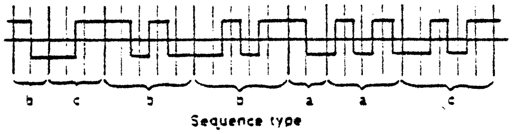
The bit stream to be encoded is broken into sequences of three types:

(a) Any number of consecutive ones.

(b) Two zeros separated by either no ones, or an odd number of ones.

(c) One zero followed by an even number of ones (terminated by a zero not counted as part of the sequence).

Sequences type (a) and (b) are coded as in normal Miller code. Sequences type (c) have the transition corresponding to the final '1' inhibited, e.g.:

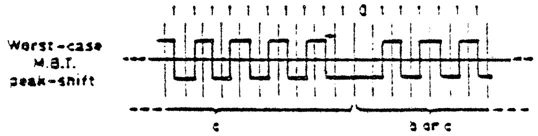


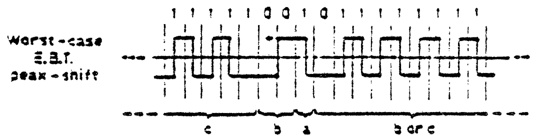
This code, invented by J. W. Miller,[12] is a modification of the original Miller code (sometimes attributed to A. Miller[13]), in such a way as to remove its d.c. content. The d.c. content of a code manifests itself as a baseline shift of either the read-back waveform, or its derivative. This can cause errors in a detection system which is particularly sensitive to amplitude variations, but can be overcome to a large extent by d.c. restoration circuitry. The alternative is to use a code such as Miller,[2] with zero d.c. content, but the disadvantage that results is a greater $I\hat{R}T$ ($3 \times DBP$), yielding higher peak-shift and requiring more sophisticated clocking circuitry.

Note that this code is also known as $M^2$, or $M^2FM$, but is not the same as MMFM, which is very similar to $MNRZI_{B2}$.

### 5.8.2 *Packing density limit*

The worst case m.b.t. and e.b.t. peak-shift occurs for the following patterns:





These patterns are almost identical to the worst-case patterns for $MNRZI_{A1}$, for which $I\hat{R}T$ is also $3 \times DBP$. Performance limits are thus (from the corresponding $MNRZI_{A1}$ figures):

$$\left.\begin{array}{ll} RTE = 0: & P\hat{F} = 1{\cdot}97 \\ RTE = 8\%\ PW_{50}: & P\hat{F} = 1{\cdot}47 \\ RTE = 16\%\ PW_{50}: & P\hat{F} = 1{\cdot}5 \\ RTE \propto I\hat{R}T: & P\hat{F} = 1{\cdot}36 \end{array}\right\} \text{all due to timing.}$$

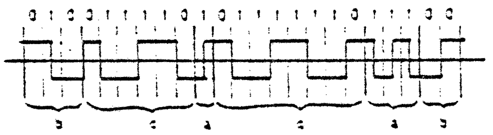
## 5.9 Zero Modulation Code

### 5.9.1 *Coding rules*

The bit stream to be encoded is broken into sequences of three types:

(a) Any number of consecutive ones.
(b) Two zeros separated by either no ones, or an odd number of ones.
(c) Two zeros separated by an even number of ones.

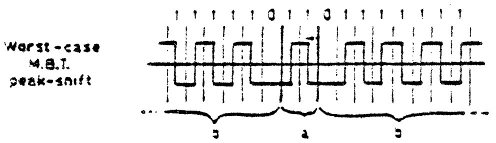
Sequences type (a) and (b) are coded as in normal Miller code. In sequences type (c), ZM encodes the zeros in the Miller manner, but the ones are encoded as though they were zeros but with alternate transitions deleted:

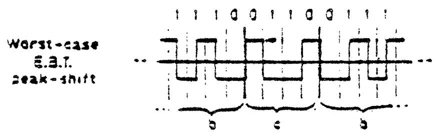


This code, invented by Patel,[14] is another attempt at improving Miller code by removing its d.c. content. It has the same disadvantages as Miller,[2] though $I\hat{R}T$ is only $2 \times DBP$.

### 5.9.2 *Packing density limit*

The worst-case peak-shift patterns are as follows:





Pattern (a) is the same as the worst-case m.b.t. shift for Miller code, whilst pattern (b) has greater peak-shift than the corresponding pattern for Miller. Analysis of these patterns yields the following packing density limits:

$$\left.\begin{array}{ll} RTE = 0: & P\hat{F} = 1{\cdot}94 \\ RTE = 8\%\ PW_{50}: & P\hat{F} = 1{\cdot}40 \\ RTE = 16\%\ PW_{50}: & P\hat{F} = 1{\cdot}11 \\ RTE \propto I\hat{R}T: & P\hat{R} = 1{\cdot}32 \end{array}\right\} \text{all due to timing.}$$

This analysis is based on ideal zero modulation code, which is very difficult to implement since complete sequences have to be modified, requiring infinite look-forward and look-back memories (unlike Miller,[2] where changes are introduced only at the end of sequences). Patel recognized this difficulty, and suggested that the memory could be reduced by blocking the data into groups of bits followed by a parity. Naturally, this dilutes the data to an extent dependent on the length of a block, but performance can never be better than the figures given above for the infinite memory.

## 5.10 3PM Code

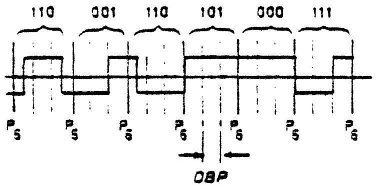
### 5.10.1 *Coding rules*

The data stream is split into groups of three bits, which are then encoded into a six-bit word for recording. The code is adaptive, in that the bit pattern for the word currently being encoded depends on the previous and the following word. The code has been designed to produce $I\hat{R}T = 1{\cdot}5 \times DBP$ and $I\hat{R}T = 6 \times DBP$. The coding is represented by:

| | | | Recorded transitions | | | | | |
|---|---|---|---|---|---|---|---|---|
| Data | Previous $P_5$ | Following $P_1$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ |
| 000 | × | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 001 | × | × | 0 | 0 | 0 | 1 | 0 | 0 |
| 010 | × | × | 0 | 1 | 0 | 0 | 0 | 0 |
| 011 | × | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| | | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 100 | × | × | 0 | 0 | 1 | 0 | 0 | 0 |
| 101 | 0 | × | 1 | 0 | 0 | 0 | 0 | 0 |
| | 1 | | 0 | 0 | 0 | 0 | 0 | 0 |
| 110 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 1 | | 0 | 0 | 0 | 0 | 0 | 1 |
| 111 | 0 | × | 1 | 0 | 0 | 1 | 0 | 0 |
| | 1 | | 0 | 0 | 0 | 1 | 0 | 0 |

(× = Don't care)

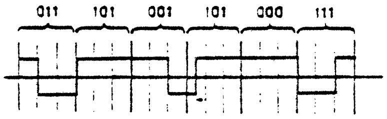
e.g.



This code, presented by Jacoby,[15] is essentially a standard type of adaptive block code, but is unusual in achieving an $I\hat{R}T$ of $1{\cdot}5 \times DBP$. This lessens the worst-case peak shift, but because $I\hat{R}T = 6 \times DBP$, the code is difficult to clock accurately.

5.10.2 *Packing density limit*

The worst-case peak-shift pattern is:



This is plotted in Fig. 14, from which the following packing factor limits can be obtained (since the detection window is $\pm 25\%$ $DBP$):

$$RTE = 0: \quad P\hat{F} = 2{\cdot}13$$
$$RTE = 8\%\ PW_{50}: \quad P\hat{F} = 1{\cdot}67$$
$$RTE = 16\%\ PW_{50}: \quad P\hat{F} = 1{\cdot}29$$
$$RTE \propto I\hat{R}T: \quad P\hat{F} = 1{\cdot}36$$

All these limits are due to timing, as amplitude is no problem for any detection system.

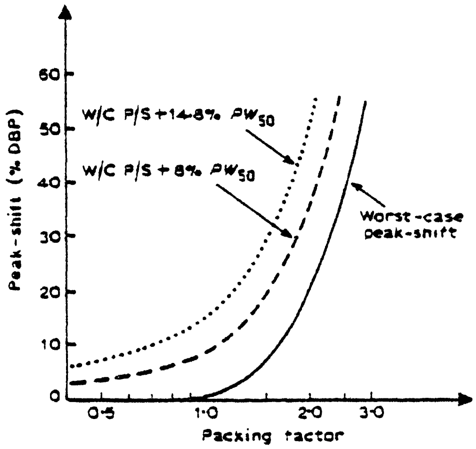


Fig. 14. Worst-case peak shift for 3PM code.

### 6 Summary of Code Performances

The performance of the best of the codes is summarized in Figs. 15 and 16. Figure 15 shows the maximum packing factor achievable by each code for the three fixed values of $RTE$. For $RTE = 0$, NRZI is surprisingly beaten for first place by the relatively unknown Rice code, because of its very good $I\hat{R}T$ of $1{\cdot}33 \times DBP$, and yet surprisingly large window of $33\frac{1}{3}\% \times DBP$. There is little to choose between the top five codes, in fact, in this $RTE$ category.

For $RTE = 8\%\ PW_{50}$, NRZI is slightly better than its nearest rivals—Rice code, $ENRZI_7$, 3PM and OCB. Again, for $RTE = 16\%\ PW_{50}$, NRZI achieves maximum performance, with Rice code and $ENRZI_7$ fairly close behind.

Perhaps the most useful indication of the performance of the codes is for $RTE \propto I\hat{R}T$, shown in Fig. 16. The striking result is that, apart from PM (which is only used when packing density is unimportant), all the other codes can achieve packing densities which are within $\pm 6\%$ of each other! Additionally, the three most recent codes, i.e. Miller,[2] 3PM and ZM, are of below average ability, showing that it may be more prudent to spend money on d.c. restoration circuitry rather than complicated encode and decode electronics.

In support of the theoretical work presented here, and in particular the general implications of Fig. 16, consider the results presented by several other authors:

Tamura *et al.*[5] compared GCR, FM and Miller, and found packing density limits of GCR : FM : Miller = 1·08 : 0·73 : 1·0.

Huber[16] found MFM : Miller[2] : 3 PM = 1·0 : 1·05 : 1·10.

King[17] found RNRZ : ENRZ : MFM : PM = 1·07 : 0·92 : 1 : 0·6.

Stein[18] found RNRZ : Miller[2] : ENRZ = 1·04 : 1·00 : 0·88.

Davidson *et al.*[19] found MFM : '4, 6 : 0' code : '6, 8 ; 0' code = 1·0 : 1·04 : 1·09.

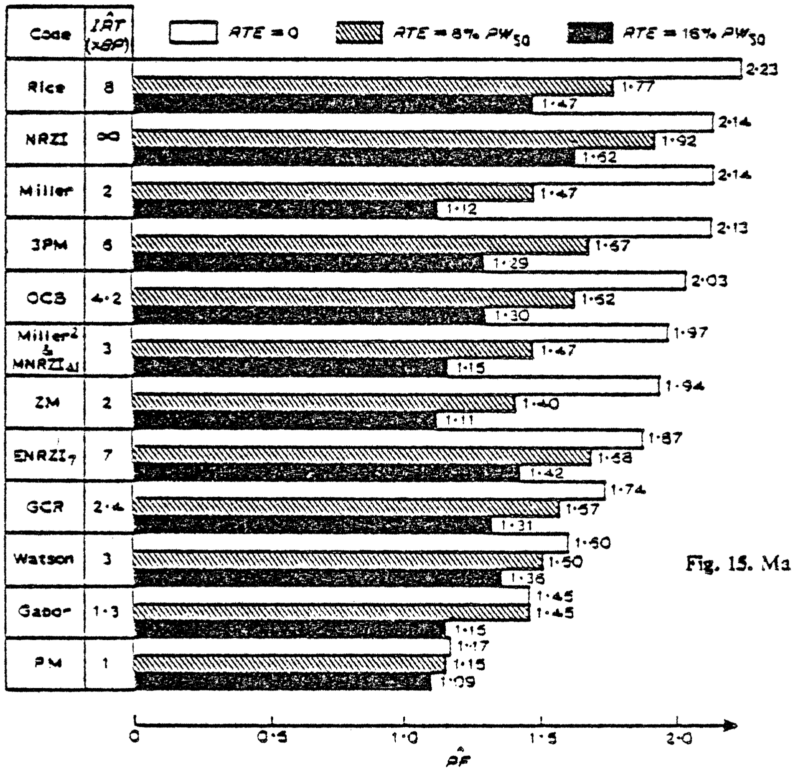


Fig. 15. Magnetic recording codes performance summary.

The clear implication is that, whilst no one agrees which is the best code, there is very little difference between all of the popular codes. Indeed, it has been shown in this study and in the excellent study by King[17] that the choice of detection system can have far more effect on the ultimate performance of a memory than can the choice of the code.

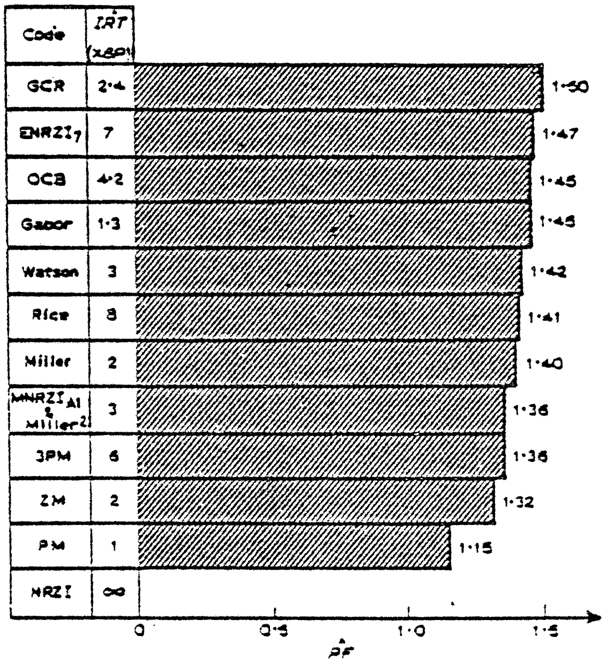
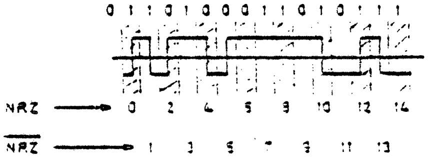
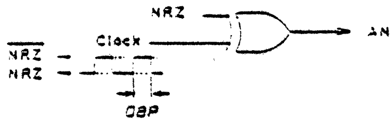


Fig. 16. Performance limits for $RTE \propto \hat{IRT}$.

NRZI remains outstanding for any fixed value of $RTE$ and because of this, a new code to emerge recently, Randomized NRZ (RNRZ), has been gaining in acceptance. It attempts to turn NRZ into a self-clocking code by scrambling the data before recording. The idea is that long runs of data without transitions are broken up, and the chances of having a long $IRT$ after scrambling are small. It is difficult to see, however, how a typically random data pattern is, in fact, improved by scrambling, i.e. randomizing. If the idea is that data often consist of all 0's, or all 1's (in NRZ), it is very simple to break these up, with much less circuitry than RNRZ requires, by alternating the NRZ definitions, i.e. Alternating NRZ (ANRZ): In even bit periods, code a change from a '0' to a '1' or a '1' to a '0' as a transition at m.b.t.; in odd bit periods, code no change in the data as a transition at m.b.t.:



It can be generated simply by exclusive-ORing the NRZ data with its own clock:



Obviously this code breaks down for an input pattern of 00110011 . . ., but there exists a similar pattern which causes RNRZ to break down.

Although some systems designers are limited in their choice of code by existing standards, e.g. the ECMA and ANSI standards for cassette drives, or the IBM 'standard' for the single density floppy drive, in cases where there is no standard, or where there is one to be set, it behoves the designer to choose a reasonably efficient code. This should not involve much effort, as there are far more important design studies to undertake, such as choice of block structure and detection method, both of which can have more effect on the ultimate formatted capacity of the drive than can haggling over the last few percent achievable by different codes.

Low-cost devices, such as cassette transports and mini diskette drives, often cannot afford the luxury of servo-controlled media speed, or sophisticated encode, decode and detection electronics. For these applications, a code which is very self-clocking, such as PM, should be used. Gabor code could be used for its low $I\hat{R}T$, but the code conversion is quite complex.

To use NRZI in a system really needs a separate dedicated clock track. This is uneconomical in most cases; but for multi-track tape, in either computer data or p.c.m. applications, and certainly for fixed head disks, the overhead involved becomes minimal. The biggest problem is then skew, but l.s.i. chips are now available to overcome this.

For security-conscious applications, the inherent scrambling of Randomized NRZI is an attraction, though the scrambling technique could be applied to any code, at the expense of an extra stage of processing.

In the majority of applications, it is difficult to *really* justify the choice of any particular code, but the original 'compromise' code, Miller code, has much to offer. It is efficient, simple to encode and decode, does not require odd-integral clocks as do some codes, and does not need a sophisticated phase-locked loop.

## 7 Conclusions

It has been shown how worst-case peak-shift and amplitude patterns can be used to determine the margins in a system. A detailed comparison of many recording codes shows that there is very little to choose between all of the popular ones, as regards the maximum packing density achievable by each.

Pattern-adaptive write (pre-compensation) is seen to be beneficial for some codes (e.g. Watson code), but of little use for most (e.g. NRZI). Similarly, window modification, or pattern-adaptive read, has little to offer. A comparison of detection techniques has shown how they can significantly influence the capabilities of a memory.

The conclusion is that system designers should choose any code which appears suitable for their particular system, and the system should then be designed around the code.

## 8 References

1 Mackintosh, N. D., 'A superposition-based analysis of pulse-slimming techniques for digital recording', Video and Data Recording Conference, 1979. (IERE Conf. Proc. 43.) *The Radio and Electronic Engineer*, **50**, No. 6, June 1980. (To be published.)

2 Kameyama, T. *et al.*, 'Improvement of recording density by means of cosine equaliser', *IEEE Trans. on Magnetics*, **MAG-12**, no. 6, pp. 746–48, November 1976.

3 Whitehouse, A. E., Ph.D. Thesis, Manchester University, 1970.

4 Dunstan, E. M. and Whitehouse, A. E., 'A 2000 bpi magnetic drum channel with high noise immunity', Conf. on Computer Technology, Manchester 1967. (IEE Conf. Pub. 32.)

5 Tamura, T., *et al.*, 'A coding method in digital magnetic recording', Intermag 1972, *IEEE Trans.*, **MAG-8**, No. 3, pp. 612–14, September 1972.

6 Padalino, M., Workshop on Magnetic Recording, Intermag 1968.

7 Pouliart, W. H. P. and Vandevenne, J. P. H., 'Electrical Intelligence Storage Arrangement', U.S. Patent No. 2,807,004, September 17th, 1957.

8 Woo, W. D., 'Binary Magnetic Recording Àpparatus', U.S. Patent No. 3,235,855, February 15th, 1966.

9 Franaszek, P. A., 'Sequence-state methods for run-length-limited coding', *IBM J. Res. & Dev.*, **14**, No. 4, pp. 376–83, July 1970.

10 Gabor, A., 'Adaptive coding for self-clocking recording', *IEEE Trans. on Electronic Computers*, **EC-16**, no. 6, pp. 866–68, 1967.

11 Watson, I., Ph.D. Thesis, Manchester University, 1973.

12 Miller, J. W., U.S. Patent 4,027,335, May 31st 1977.

13 Miller, A., U.S. Patent 3,108,261, October 22nd, 1963.

14 Patel, A. M., 'Zero modulation encoding in magnetic recording', *IBM J. Res. & Dev.*, **19**, no. 4, pp. 366–78, July 1975.

15 Jacoby, G. V., 'A new look-ahead code for increased data density', *IEEE Trans.*, **MAG-13**, no. 5, pp. 1202–4, September 1977.

16 Huber, W. D., 'Maximization of lineal recording density', *IEEE Trans.*, **MAG-13**, no. 5, pp. 1208–10, September, 1977.

17 King, D. A., 'Comparison of pcm codes for direct recording', Intern. Telemetering Conf., September 1976, pp. 526–40.

18 Stein, J. H., 'High Density Recording: Facts relating to Standardization', BSI Doc. no. 78/60564.

19 Davidson, M., *et al.*, 'High density magnetic recording using digital block codes of low disparity', *IEEE Trans.*, **MAG-12**, no. 5, pp. 584–86, September 1976.

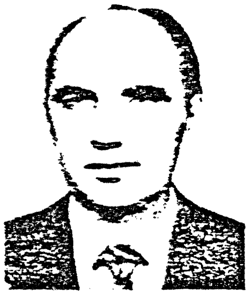
*Manuscript received by the Institution on 2nd April 1979*
(*Paper No. 1928/Comp 195*)

# Contributors to this issue

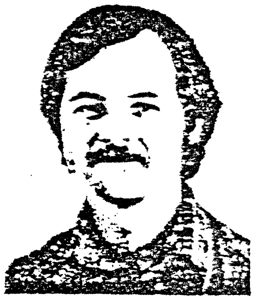
**Ralph Benjamin** (Fellow 1975) has been Chief Scientist and Superintending Director at the Government Communications Headquarters since 1971. A graduate of the University of London, he joined the Royal Naval Scientific Service in 1944 and after individual merit promotions to Senior Principal Scientific Officer and Deputy Chief Scientific Officer, he was appointed Head of Research and Deputy Director of the Admiralty Surface Weapons Establishment in 1961. In 1964 he transferred to the Admiralty Underwater Weapons Establishment as Director and Chief Scientist, also becoming a qualified Naval Diving Officer. From 1965 to 1971 he combined these appointments with that of Director of Underwater Weapons R & D at the Ministry of Defence. In 1964 London University awarded Dr Benjamin the Ph.D. degree for a thesis on signal processing, and in 1970 the D.Sc. for his contributions to general electronics. He has published a book on 'Modulation, Resolution and Signal Processing in Radar, Sonar and Related Systems' and is author of numerous papers in this Journal and elsewhere.

He was appointed C.B. in the recent New Year Honours List.

**Neil MacKinnon** (Fellow 1979, Member 1970) studied at Worcester College, Oxford, and the Royal Military College of Science whilst serving in the Royal Corps of Signals. During his military service he spent eight years in electronic warfare environments and other appointments included those of Airborne Forces Development Officer for electronic devices such as transponders and locator systems, and Weapons Staff Officer at SRDE, Christchurch working on the Clansman project. After leaving the Army in 1973 Mr MacKinnon spent three years with the Racal Electronics Group during the formative period of their communication security company, Racal Datacom, to whom he is now a Consultant. His current interests are in the application of analogue solid-state devices, such as c.c.d.s, to signal processing, and the system planning of secure diplomatic radio networks.



**'Nick' Kouvaras** graduated first from the Athens Higher School of Electronics Engineering, after which he attended the Faculty of Mathematics of the University of Athens and obtained his B.Sc. degree in mathematics. In 1960 he joined the Electronics Department of the 'Demokritos' Nuclear Research Centre, and he has done research and development work on digital systems which has led to several publications including a paper in this Journal in September 1978 on delta modulation; he is also involved in a programme of work on digital processing systems for speech.



**Nigel Mackintosh** received his B.Sc. in computer science at the University of Manchester in 1971 and his subsequent work on magnetic aspects of the MU5 research computer gained him his Ph.D. in 1975. At Burroughs Machines in Scotland he helped develop the world's first 1 Mbyte floppy-disk drive, and later at Racal Recorders in Southampton he applied microprocessor technology to communications recorders and equipment. Dr Mackintosh is now with the Advanced Techniques Group of Burroughs Machines in California evaluating the limits of digital data recording.

**Ian Vance** joined Standard Telecommunication Laboratories in 1968 after receiving the degree of Bachelor of Engineering from the University of Liverpool. In 1973 he was admitted as Master of Science at Aston University. Mr Vance has worked in many aspects of the field of radio communication and microwaves, leading in recent years to an interest in highly integrated forms of such equipment. He is currently manager of the Radio Frequency Sub-Systems Department at the Harlow establishment of STL.