(12) UK Patent Application (19) GB (11) 2 163 627 A

(43) Application published 26 Feb 1986

(21) Application No 8521222

(22) Date of filing 23 Aug 1985

(30) Priority data
(31) 8421500 (32) 24 Aug 1984 (33) GB

(51) INT CL⁴
H03K 3/84 G01R 31/02 G06F 7/58 H04M 3/34

(52) Domestic classification
H4P EUM
G4D AJ

(56) Documents cited
GB A 2131183

(58) Field of search
H4P
G4D
G1V

(71) Applicant
British Telecommunications plc (United Kingdom),
81 Newgate Street, London EC1A 7AJ

(72) Inventor
Peter Michael Attkins

(74) Agent and/or Address for Service
Kevin John Purell,
Intellectual Property Unit, Room 1304, 151 Gower Street, London WC1E 6BA

(54) Error generator

(57) The output of a pseudo-random sequence generator (SR,X1) is fed to a comparator (CP1, M1) which produces an output when a selected number of bits of the generator output matches a comparison word (from CR1); the number of bits selected (by MR1) determines the error rate. Correlation problems are reduced by changing the comparison word (eg. by loading latch CR1 from counter LC incremented at clock rate) each time a match is found.

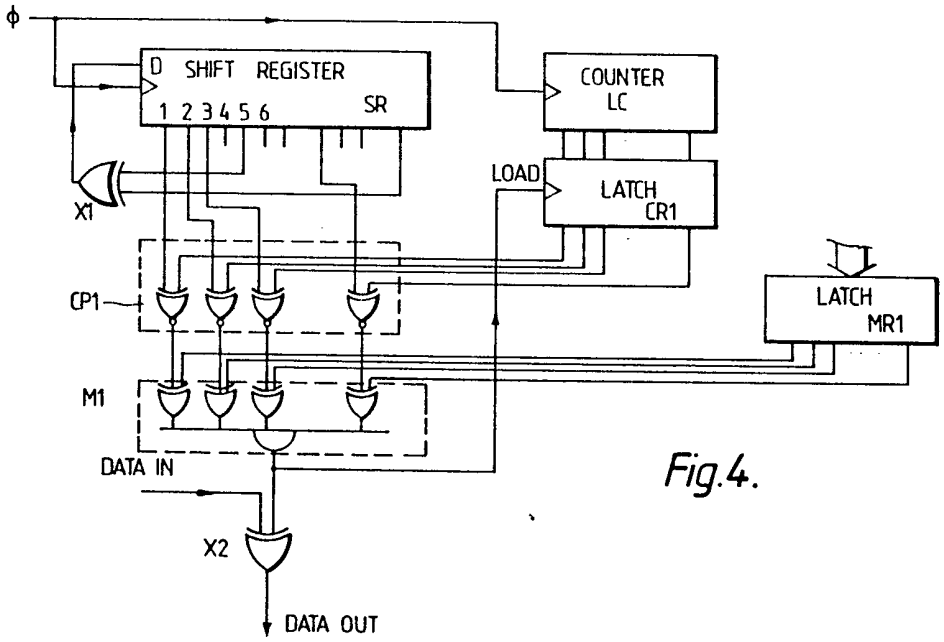
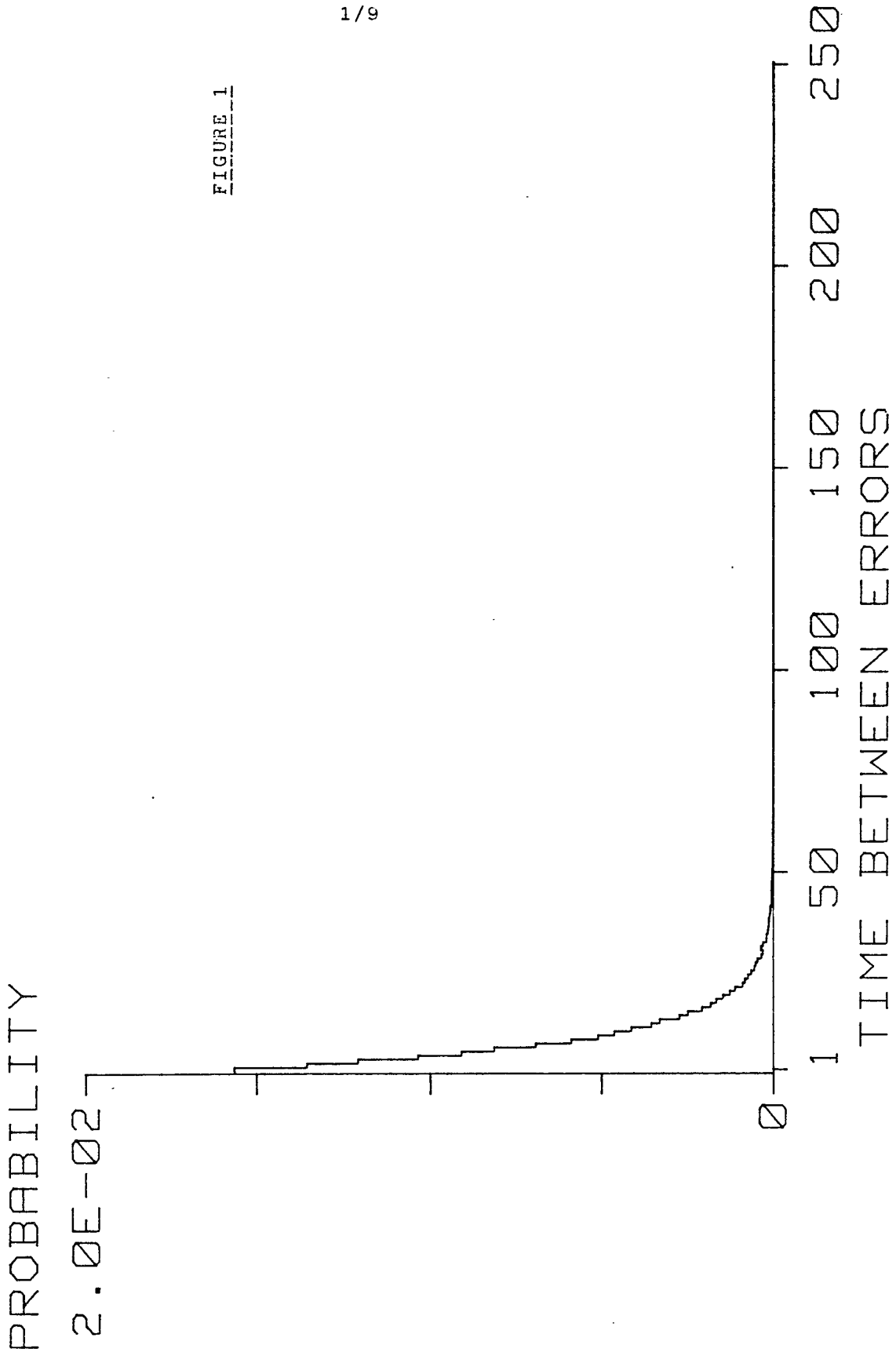


Fig.4.

GB 2 163 627 A

The drawing(s) originally filed was/were informal and the print here reproduced is taken from a later filed formal copy.

FIGURE 1



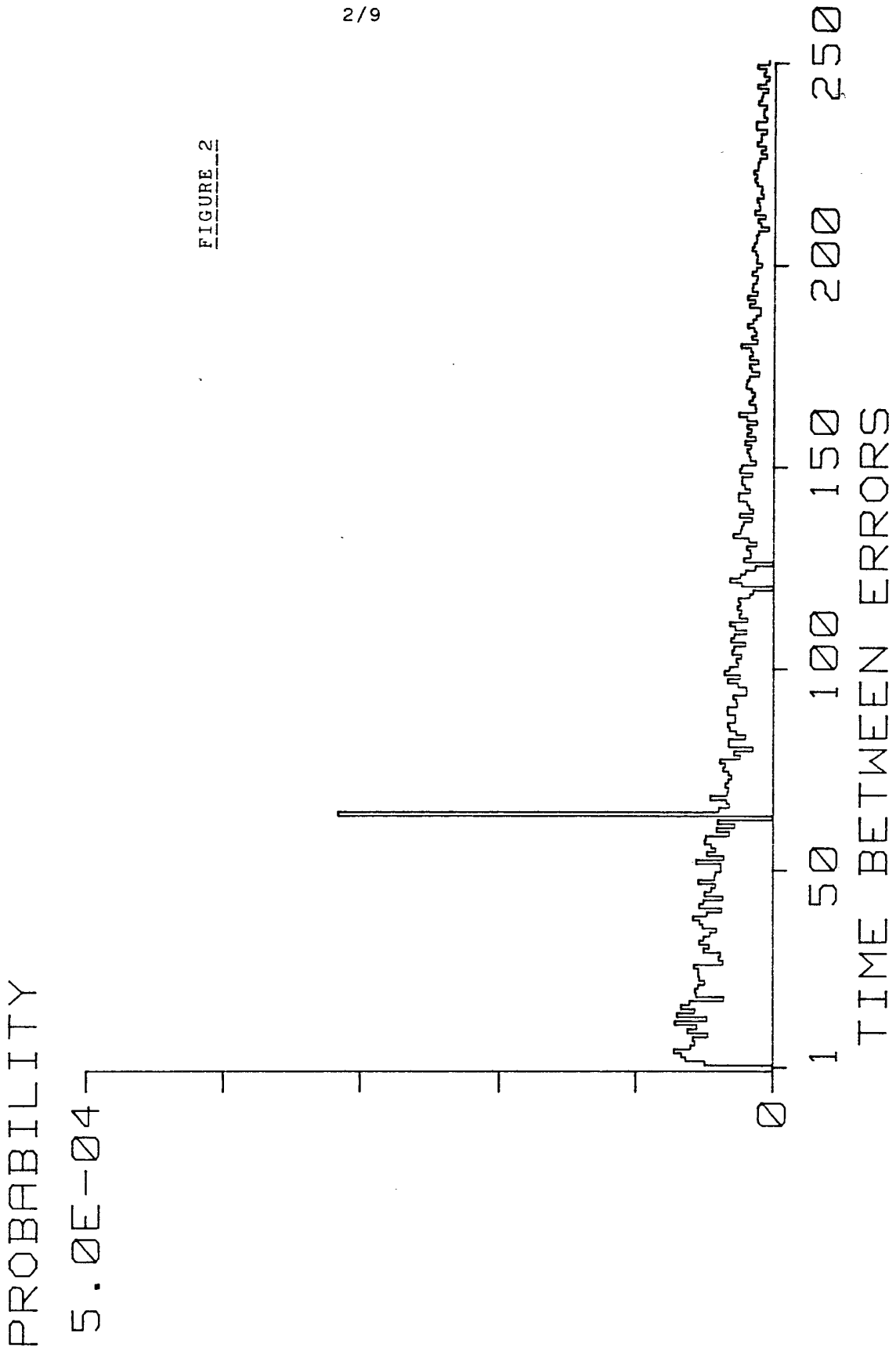
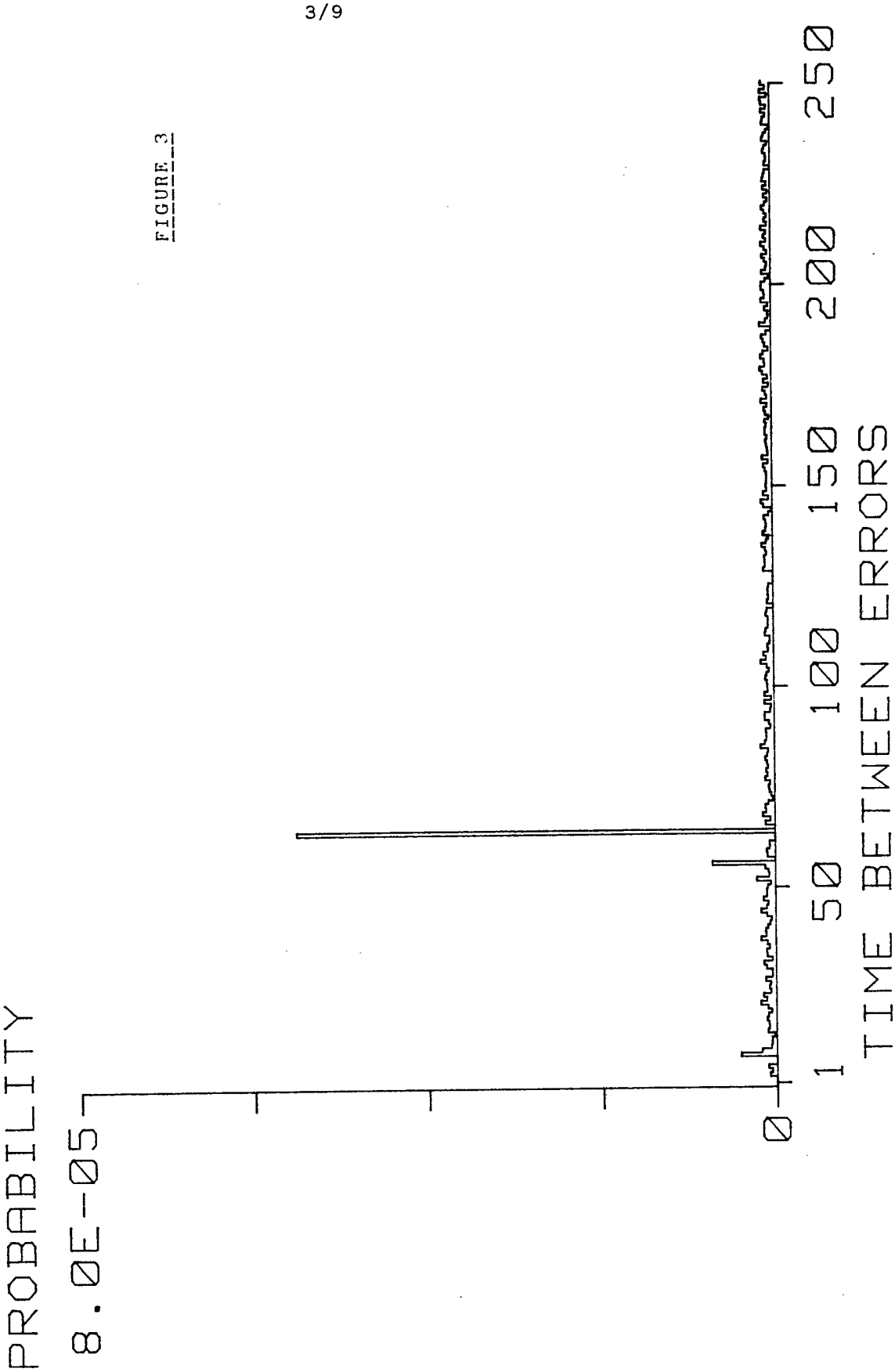


FIGURE 2

FIGURE_3



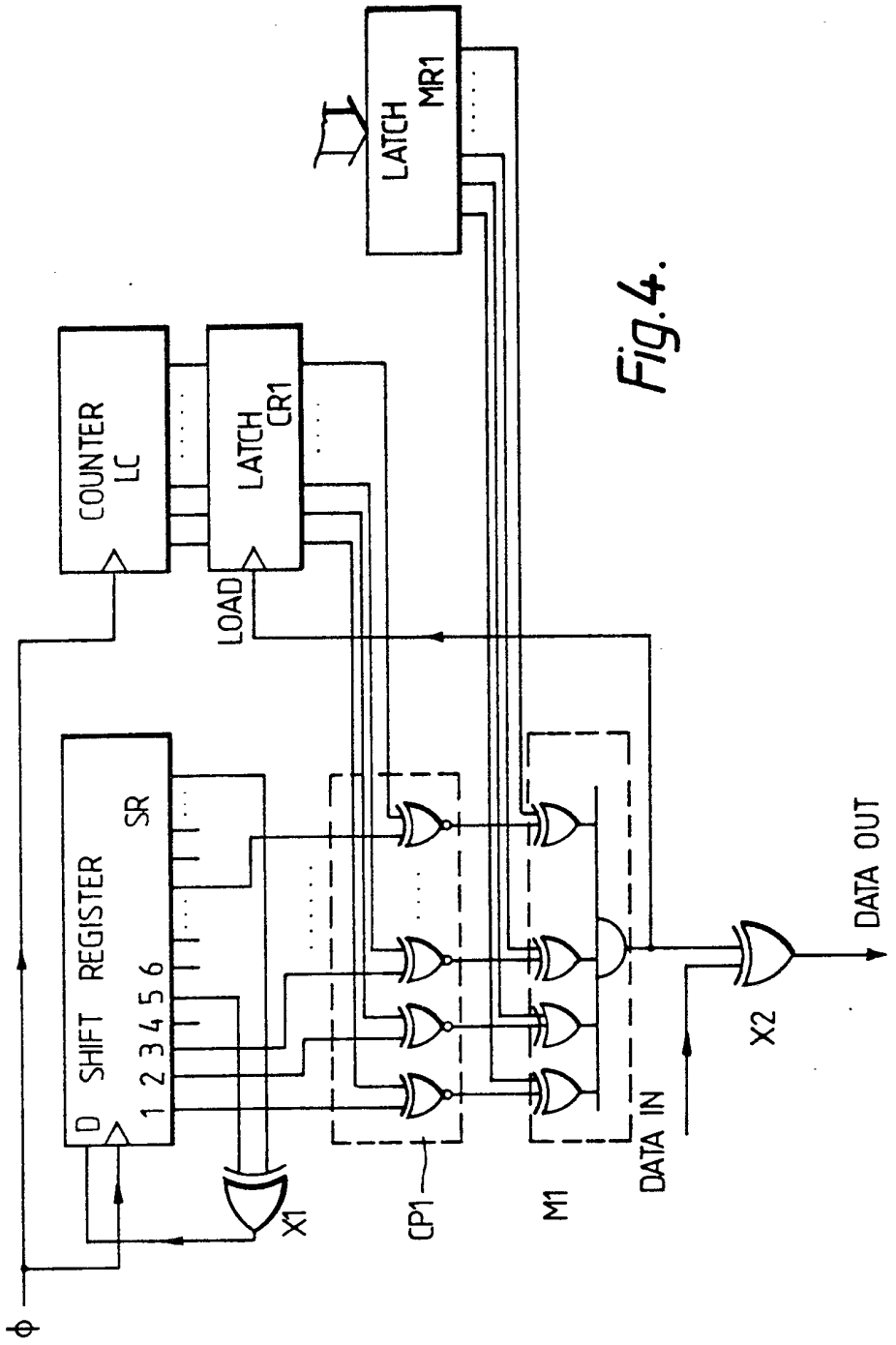

LATCH MR1
COUNTER LC
LATCH CR1
LOAD
D SHIFT REGISTER SR
1 2 3 4 5 6
X1
CP1
M1
DATA IN
X2
DATA OUT
ϕ

Fig.4.

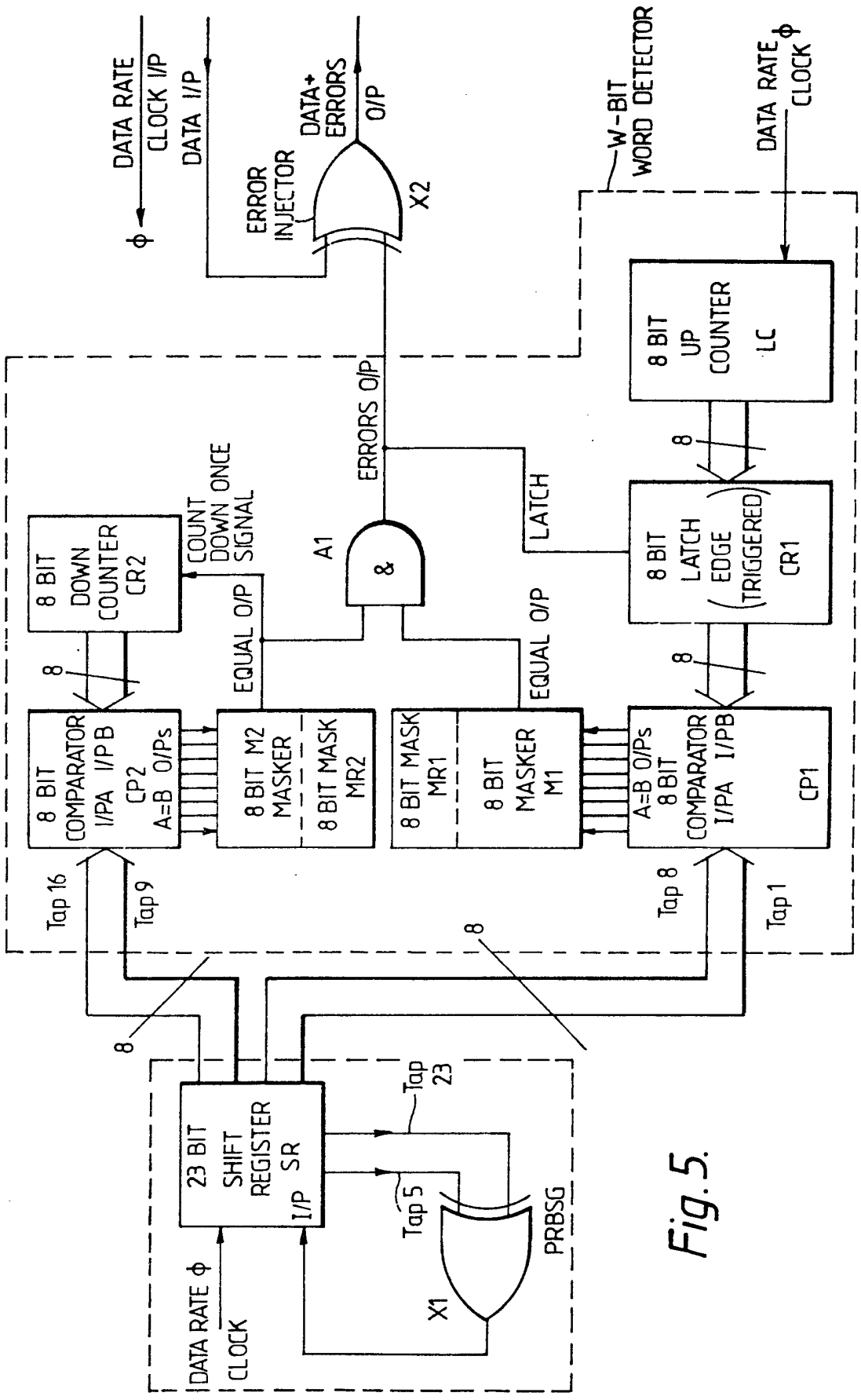

DATA RATE
CLOCK I/P
DATA I/P
ϕ
ERROR INJECTOR
DATA+ ERRORS O/P
X2
W-BIT WORD DETECTOR
DATA RATE ϕ CLOCK
8 BIT UP COUNTER LC
8
8 BIT LATCH EDGE (TRIGGERED) CR1
LATCH
ERRORS O/P
A1
&
8 BIT DOWN COUNTER CR2
COUNT DOWN ONCE SIGNAL
EQUAL O/P
EQUAL O/P
8 BIT COMPARATOR I/PA I/P B CP2 A=B O/Ps
8 BIT M2 MASKER
8 BIT MASK MR2
8 BIT MASK MR1
8 BIT MASKER M1
A=B O/Ps 8 BIT COMPARATOR I/PA I/PB CP1
Tap 16
Tap 9
Tap 8
Tap 1
8
8
23 BIT SHIFT REGISTER SR I/P
Tap 23
Tap 5
X1
PRBSG
DATA RATE ϕ CLOCK

Fig. 5.

FIGURE 6

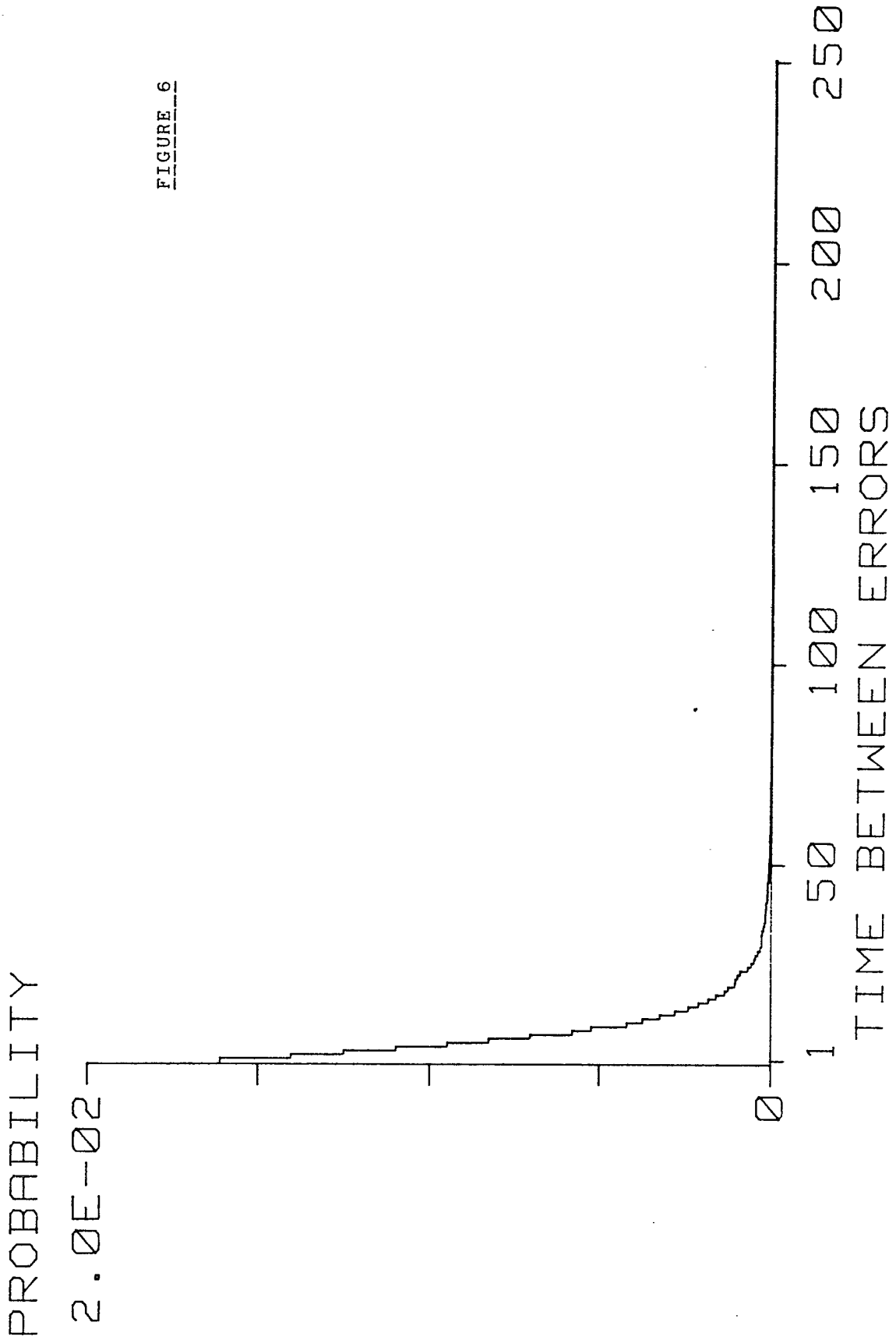
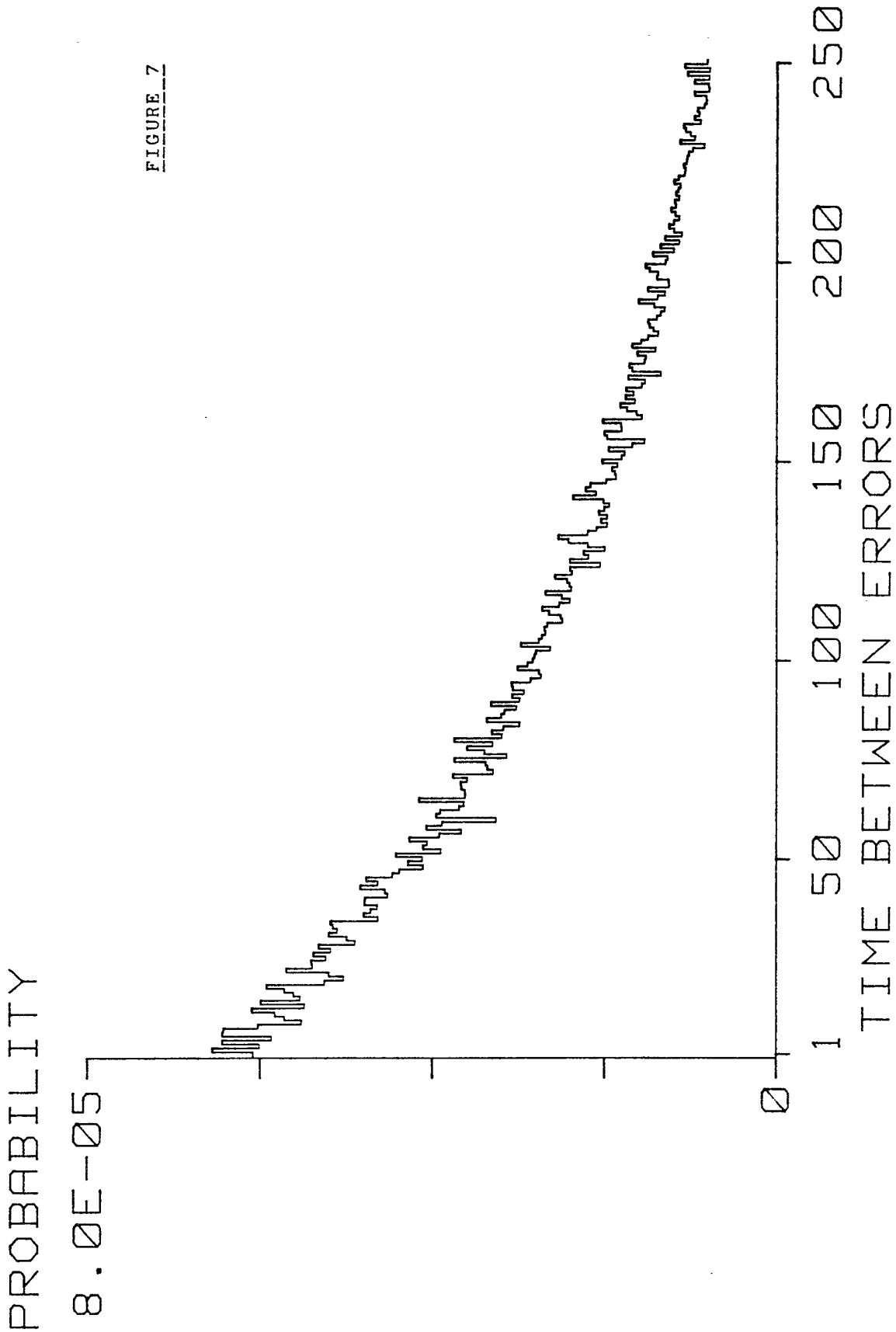
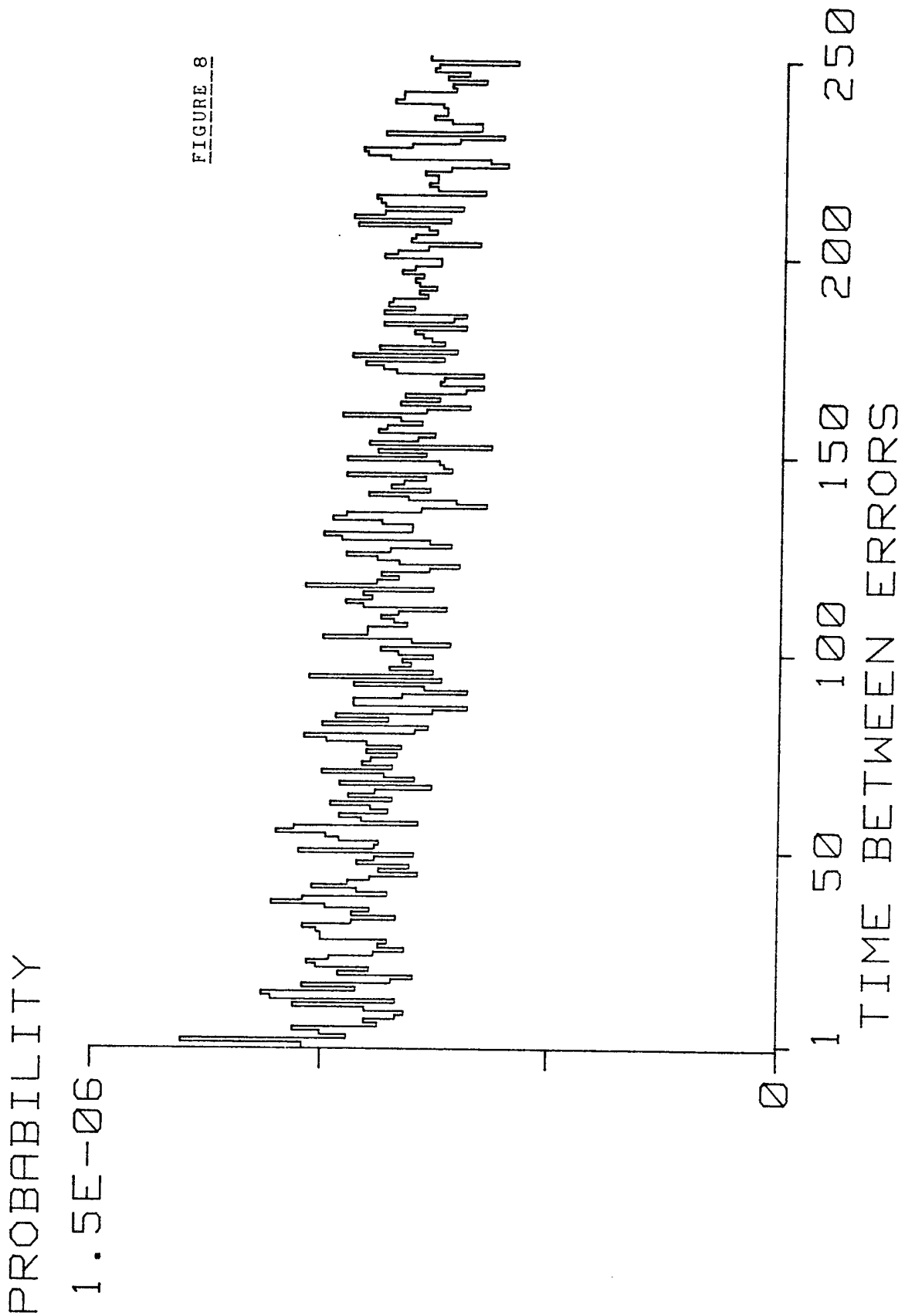


FIGURE 7



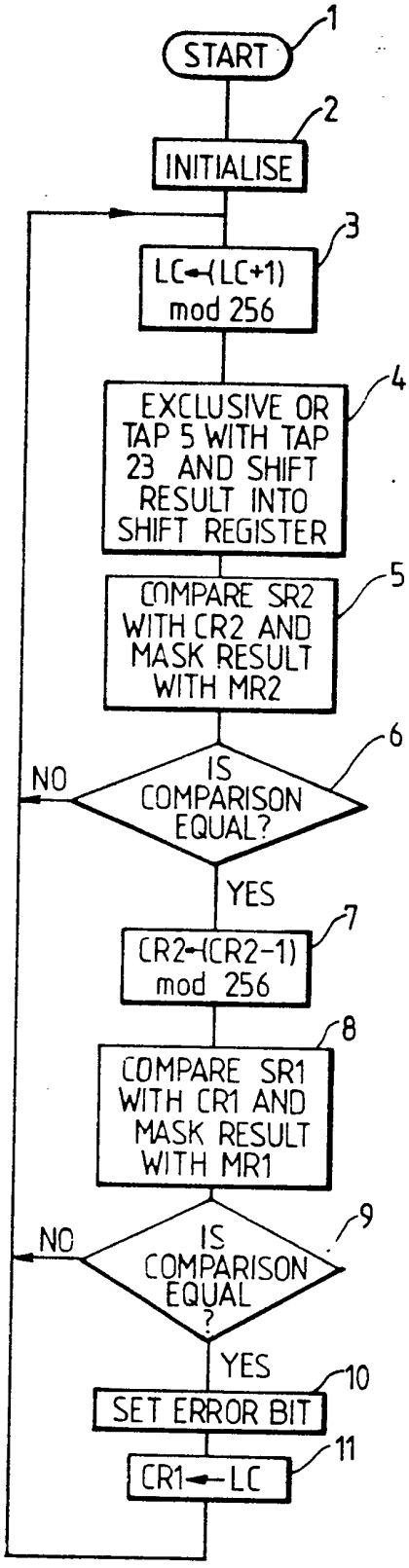
2163627

FIGURE 8



2163627

Fig.9.



SPECIFICATION

Error generation

There are various types of random and pseudo random error generator principles. For example, a white noise source connected to a variable threshold comparator and sampling circuit will produce an output signal with probability altered by the threshold level. This type of generator has many failings such as; small noise power changes caused by temperature etc. will have a considerable effect on the error probability; the noise power bandwidth and comparator bandwidth must be many times greater than the sampling frequency–if this is not attained then there will be correlation between consecutive samples and the output signal will reflect this deficiency.

The most consistent and predictable error generators are based on digital techniques, ideally using random number generation. A realistic approach is to generate a long pseudo random sequence and use a digital word detector.

Assume that the generator consists of an n-bit shift register with feedback, and a w-bit word detector produces an output whenever a specified word appears in the output sequence.

The generator will produce 2^n-1 unique words of n-bits (with exclusive OR feedback all zeros is illegal and with exclusive NOR feedback all ones is illegal). This is true for certain tap configurations only–some combinations will produce far fewer words. The word detector can 'look' for any w-bit word in the sequence except the illegal word therefore it is better to make $w<n$ to avoid such a situation.

The probability of a w-bit word being detected is slightly greater than 2^{-w} because there are only 2^n-1words and not 2^n. The difference depends on the actual value of n. An obvious limitation to this generator is that the probability values obtainable are in powers of 2 and not infinitely variable. In practice, however, there is normally more than enough resolution.

This method of error generation does produce the correct probability, but the nature of error distribution along the length of pseudorandom sequence is sequence related. For a given w-bit word the output sequence from the detector may be charged by altering the generator, for example by changing n or just changing the other shift resistor tap/taps and the word order will change. Similarly an exclusive NOR feedback would give a different distribution from exclusive OR. This is irrelvant if the distribution is reasonable anyway, but certain configurations may result in the output signals being bunched together and ten a long gap until the pattern repeats, especially when w approaches n.

Closer examination of a typical PRBSG will reveal another shortcoming of this type of error generator and that is of inter-word correlation similar to bandwidth limitation in a noise source based design. A PRBSG has word correlation at an interval of 2^m-n-1 (a maximal length sequence)–this is ONLY for a one bit word though. If we use more than one bit there may be correlation over a much shorter interval. If, for example, the feedback is an EXCLUSIVE OR of shift register taps 28 and 31, the sequence is very long, but correlation will occur at intervals of Tap 28+1, Tap 31+1, Tap 31−Tap 28 because each new bit entering the shift register is related to bits 28 and 31 further down. It causes an unwanted distribution of output signals which gets worse as w increases from 2.

An error generator of this type is described in CCITT Study Group XVII Document No: 18, Annex 1 (14 July 1982). This uses a 31-bit register with exclusive or feedback from taps 28 and 31; a w-bit word detector produces an error output when taps 12 to (11+w) are all "1".

This error generator attempts to circumvent the problem of correlation by sub-sampling the PRBSG at a ratio of 1:32. This is achieved by clocking the generator 32 times between each test by the word detector thereby removing all correlations<32. In practical terms the shift register clock is 2.048MHz while the sampling/data rate is 64KHz. The feedback taps used (Taps 28 and 31) produce a maximal length pseudo random binary sequence of length $2^{31}-1$ bits or approximately 2.1×10^9 bits before a repeat of the sequence.

The sub-sampling rate cannot be a submultiple of the pattern length due to the −1 term in 2^n-1 so the actual sub-sampled pattern would repeat approximately:

$$\frac{32\times2.1\times10^9 \text{ bits at } 2.048\text{Mbit/s (each pattern is 'short' bu 1 bit)}}{32 \text{ (sub-sampling rate)}}$$

which is the same length as the original pattern of 2.1×10^9 bits or a repetition once every 32812.5 seconds (9 hours approximately).

Unfortunately this sub-sampling only shifts the correlation intervals instead of eliminating them. Computer simulation of this error generator has been carried out and results are shown in Figs. 1 to 3 for an order w of 3, 7 and 10 respectively, the probability p of an interval of the samples between errors being plotted against tbe. Further data are set out below:

Fig	Order (w)	Error Rate	Equivalent duration of test at 64kbit/s (seconds)	Total Errors	% of errors beyond tbe=250
1	3	120×10^{-3}	13	100,000	0
2	7	7.8×10^{-3}	20	10,000	14
3	10	0.97×10^{-3}	161	10,000	75

It will be observed that the output shows major correlation problems at intervals 63 and 64 and less important problems at higher intervals. If P(tbe) is the probability of consecutive errors occurring at an interval of the then:

P(63)- 0 for all values of ORDER–ie. once an error has occurred it is certain that no error will occur 63 samples later

P(64)>P(x) where $x\neq63$ and x is 'near' to 64 for all values of ORDER

For ORDER> 3:P(64) is 4 to 6 percent of the total error rate or there is roughly a 5 percent chance of an error being produced 64 bits after the previous error. This is clearly a failing of the error generation method.

The relationship between error probability/rates and equivalent mean-time-between-errors for 64 kbit/s data rates, and the ORDER of the system is as follows:

ORDER=w
Error probability (approximately)=2^{w}
Error rate=1 in 2^{w}
Equivalent mean-time between errors=$2^{w}/64000$ (seconds)

If the generator was set to ORDER =20 the mean time between errors would be 16s and this would be very like having a push-switch to generate a single error occasionally. One important difference though is that the generator will, about 5 percent of the time, produce a second error 64 bits (1ms) later–the ear would detect this as a single disturbance of greater magnitude than one error. Therefore the transmission system under test may well sound worse than it should as some of the time the double errors will be considered, by the listener, to be single errors as indeed they should be. Any transmission system with data structure at a sub-multiple of 64 (byte derived systems such as PCM) could be affected by the problem at P(64) and any system which has framing information will not respond to a given error in the same manner as it would with properly (randomly) distributed errors. Indeed it may be possible to build transmission systems which deliberately respond very favourably to the error injector and thereby give a better measured performance than they would otherwise.

According to the present invention there is provided a variable probability error generator comprising a pseudo-random sequence generator (SR,X1) and detector means (CP1, CP2) arranged to examine a selected number of bits of the generated sequence and to produce an output upon recognition that those bits correspond to a comparison word, characterised by modification means (LC, CR1, CR2) operable, following each such recognition, to change the comparison word.

Some embodiments of the present invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figures 1 to 3 are graphs illustrating the performance of a known error generator;

Figures 4 and 5 are block diagrams of two alternative error generators according to the invention;

Figures 6 to 8 are graphs illustrating the performance of the generator of Fig. 5; and

Figure 9 is a flowchart illustrating a third embodiment of the invention.

Referring to Fig. 4, a pseudo random binary sequence generator is formed by a shift register SR having feedback via an exclusive–or gate X1 from taps 5 and 23, giving a pattern length of

more than 8 million bits. The register is loaded initially (by means not shown) with a non-zero number to ensure proper startup. The first W taps are used in conjunction with a w-bit word detector where w is less than or equal to W. The errors output are injected into the incoming data stream with an exclusive OR gate X2.

It is the implementation of the w-bit word detector that controls the performance of the error generator with regard to distribution of time between errors, the problems with conventional generators having been shown already. In Fig. 4, the W bits are compared bit by bit in comparator CP1 with a comparison word stored in a latch CR1. The outputs of the comparator are fed to a masker M1 in which the first w bits are AND-ed to form the output; the remainder if any being forced to 1 by the output of a latch MR1 by presetting appropriate bits of the latter.

This generator does not even attempt to employ sub-sampling techniques, instead, the w-bit word detector uses a variable word for its comparison. Conventional generators use a fixed word–the actual word will affect the error distribution, but any fixed word produces an unwanted distribution similar to the graphs shown in Figs. 1 to 3 (this used 'all ones' as the word). Using a variable word overcomes the correlation problem providing that the word is not directly related to the sequence. The word is charged each time the w-bit word detector produces an output, the new word being stored in the latch CR1 and is the value of a modulo 256 counter LC clocked by the data rate clock ϕ.

Fig. 5 shows a modified version, in which the word detector examines up to 16 bits (ie. W=16). The lower 8 bits are dealt with in the same way as in Fig. 4, whilst the upper 8 are compared in comparator CP2 with a separate comparison word from an 8-bit modulo 256 down counter CR2, decremented by the equal output of the comparator. The comparator output is masked by masker M2 and mask register MR2 and the output combined with that of MR1 in an AND gate A1.

The error rate selection is controlled by the mask registers MR2, MR1 eg:

ORDER	MR2	MR1	COMMENT
1	00000000	00000001	Look at tap 1 only
13	00011111	11111111	Look at tap 1 to tap 13

Note however that the bits set do not have to be consecutive; any pattern can be used provided the total number of bits set to 1 is w.

These values are used by the 8-bit maskers to select only the required taps for the selected ORDER. A 'zero' mask bit means–assume the corresponding tap was equal to the compare register bit value (ignore it). A 'one' means pass the comparator value to the equal detector in the masker. Note that this is the reverse convention to that employed in Fig. 4. Thus if w is less than or equal to 8, then operation is identical to that of Fig. 4; if w is greater than 8 then the upper half of the detector is used. The upper EQUAL O/P will detect w-8 bits from the PRBSG (the detector always 'looks' at tap 1 to tap w) and when equality is found then the upper 8-bit word is decremented by one. Obviously to produce an error output both halves of the word detector must be presented with their correct words simultaneously.

Computer generated results for the arrangement of Figs. 5 are shown in Fig. 6 to 8 (Figs. 6 and 7 are also valid for the version of Fig. 4, of course). Further data are as follows:

Fig	Order (w)	Equivalent Error Rate	duration of test at 64kbit/s (seconds)	Total Errors	% of errors beyond tbe=250
6	3	120×10^{-3}	13	10,000	0
7	7	7.9×10^{-3}	198	10,000	14
8	10	0.97×10^{-3}	1610	10,000	78

It will be observed that there are *no* zero probability points, and that there are no longer any excessively high probability points. Fig. 8 shows a wider variation than the others but this is due to the low error rate in relation to the duration of the test and would also be seen for a true random sequence. It is found that the graph "smoothes" as the test is continued.

It should also be noted that the vertical scales of the graphs of Figs. 6 to 8 are different from

those of Figs. 1 to 3. Interestingly, the differences between the two sets of results cannot be discerned using a spectrum analyser.

As well as the substantial improvement in probability distribution, the generators described also have the advantage, compared with generators using sub-sampling, of not requiring logic operating at rates in excess of the data clock rate. Whilst this would be of importance in the hardware versions described only in the case of extremely high data rates, it is significant for the microprocessor based version described below. Using a Motorola 68000 microprocessor with an 8 MHz processor clock, programmed in machine code, it has been found that it is possible to achieve real-time error generation for 64kbit/s data. Were sub-sampling employed, this would not be possible.

A microprocessor-based version has also been implemented on an 8-bit Intel 8051 microcontroller chip. The prime requirement was for speed of execution as the application was a 16kbit/s error injector. The microcontroller will produce errors at 32kbit/s, but it is insufficiently fast for 64kbit/s. All practical tests (to conform the simulation) were performed at 16kbit/s. A flowchart for this implementation is shown in Fig. 9.

This uses the same algorithm as embodied in the hardware version of Fig. 5. The same references are used to refer to the CPU registers (or memory locations) as were used to refer to the register of Fig. 5. The 23-bit shift registers BR is represented by three 8-bit registers as follows:

SR1:–tap 1 –tap 8
SR2:–tap 9 –tap 15
SR3:–tap 16–tap 24

Only those steps of the flowchart which are not self-explanatory are discussed below

2 Initialise: Set SR1=5; SR2=SR3=0
CR1=CR2=0
LC=0
Mask registers filled with N ones
starting from the l.s.b of MR1

3 The loop counter is incremented.

When listening to speech over a 64kbit/s transmission system a single error may be noticed; however, there is little point in using an error generator with an order greater than 20 because it would approximate single errors.

CLAIMS

1. A variable probability error generator comprising a pseudo-random sequence generator (SR,X1) and detector means (CP1, CP2) arranged to examine a selected number of bits of the generated sequence and to produce an output upon recognition that those bits correspond to a comparison word, characterised by modification means (LC, CR1, CR2) operable, following each such recognition, to change the comparison word.

2. An error generator according to claim 1 characterised in that the detector means is arranged to examine the selected number of consecutivie bits of the generated sequence.

3. An error generator according to Claim 2 characterised in that the detector means is arranged, for each bit of the generated sequence, to examine that bit and the appropriate number of succeeding bits.

4. An error generator according to claim 2 or claim 3 characterised in that the modification means is arranged, upon such recognition, to make the comparison word, considered as a binary number, is made equal to the number of comparisons, counted modulo-s^w, where w is the selected number of bits, which have occurred since an arbitary datum point.

5. An error generator substantially as herein described with reference to the accompanying drawings.

Printed in the United Kingdom for Her Majesty's Stationery Office, Dd 8818935, 1986, 4235.
Published at The Patent Office, 25 Southampton Buildings, London, WC2A 1AY, from which copies may be obtained.