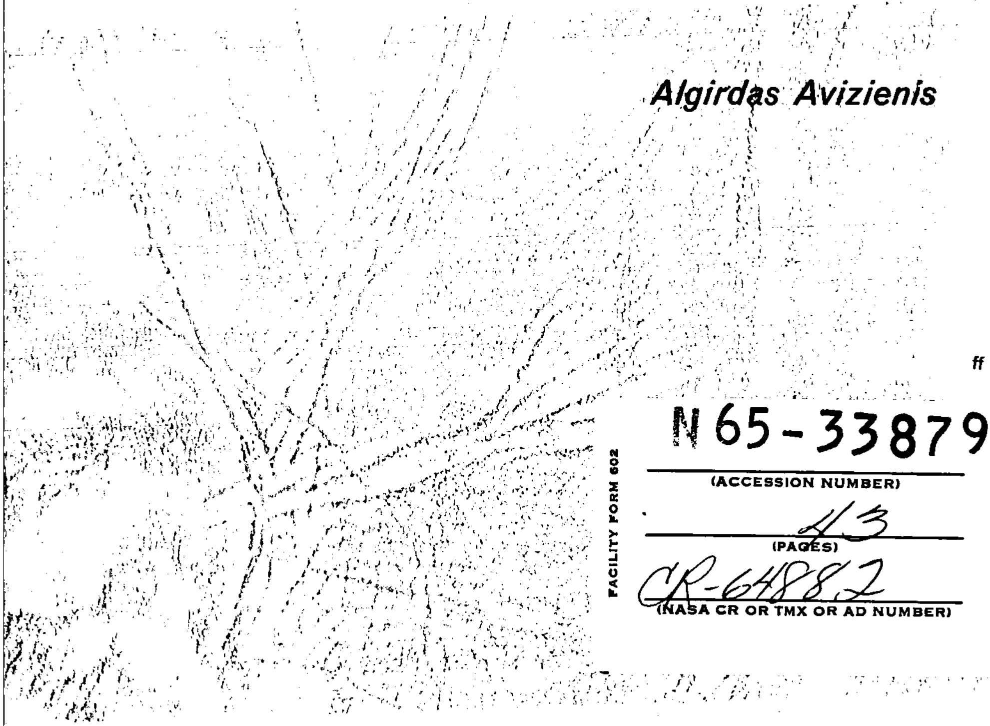
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report No. 32-711

A Study of the Effectiveness of Fault-Detecting Codes for Binary Arithmetic



Algirdas Avizienis

GPO PRICE $ ________

CFSTI PRICE(S) $ ________

Hard copy (HC) 2.00

Microfiche (MF) .50

ff 653 July 65

FACILITY FORM 602

N65-33879	
(ACCESSION NUMBER)	(THRU)
43	1
(PAGES)	(CODE)
CR-64882	08
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

September 1, 1965

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report No. 32-711

A Study of the Effectiveness of Fault-Detecting Codes for Binary Arithmetic

Algirdas Avizienis

Henry A. Curtis

Henry A. Curtis, Manager
Flight Computers and Sequencers Section

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

September 1, 1965

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

CONTENTS

Appendix.

ABSTRACT

33879

The effectiveness of a class of low-cost fault-detecting codes for binary arithmetic is investigated. Causes of arithmetical errors are studied, and error magnitudes are related to logic faults and to the arithmetic algorithms of the processor. The effectiveness of the fault-detecting codes is considered for three types of logic faults. Cases of one-use and of repeated use of the faulty logic element during an algorithm are investigated separately. Fault location and the application of several check factors are developed as extensions of the fault-detecting algorithm.

author

I. CHARACTERIZATION OF ERRORS IN DIGITAL ARITHMETIC

A. Digital Arithmetical Processors

In a digital computing system it is possible to recognize one or more functional groupings of logical and storage elements as the arithmetical processors of the given system. For the following discussion, an assembly of digital circuits is said to be an arithmetical processor when it behaves as a "black box" with the following input/output characteristics:

1. The *inputs* are an operation command (ϕ) and one or more operands (X, Y). The operation command specifies one from a set of available arithmetical algorithms $\{\phi\}$. The operands are digital numbers from the set of allowed operands $\{X\}$. Each operand represents a numerical value.
2. The *outputs* are numerical and/or logical results. The numerical results are digital numbers from the set of allowed results $\{Z\}$. The logical results represent the logical values "true" and "false." They either indicate the occurrence of singularities (nonrepresentable numerical results) or provide two-valued information regarding the values of the operands (equality, ordering, etc.).
3. The values of the results are uniquely specified arithmetical functions of the values of the operands. The required function is specified by the operation command; consequently, the expected behavior of the arithmetical processor is entirely deterministic. For any specific design of an arithmetical processor, it is possible to associate with every input combination (ϕ, X, Y) an *execution time* $t\,(\phi, X, Y)$ and a *hardware cost* $h\,(\phi, X, Y)$. These functions describe the time required to generate the complete result(s) and the count (in some standard units) of digital hardware of the processor which is employed in the specified operation.

The choices of the set of operation commands $\{\phi\}$ and of the set of allowed digital numbers $\{X\}$ depend on the class of problems which are assigned to the arithmetical processor. These problems in digital systems range from simple counting to the execution of a large set of floating-point arithmetic operations with wired-in significance indication and error checking. The choice of a specific number representation system and the logic design of the arithmetical processor depend on the required execution times and on the allowed hardware cost for the implementation of the set of operation commands $\{\phi\}$. It should be observed that the cost of arithmetic control is incorporated into the cost of the processor by the preceding description.

Two limiting cases may be distinguished in the design of digital arithmetical processors. *Stored* (or *table-lookup*) *arithmetic* is employed when all possible results are stored in their final form and execution of an operation (ϕ, X, Y) consists of reading the corresponding result(s) from the storage location specified by (ϕ, X, Y). *Primitive* (or *Turing-machine*) *arithmetic* is employed when all numerical values are represented in the unary number system (by repeated occurrences of a single symbol) and the arithmetical algorithms are executed as sequences of elementary replacement operations. Between these two extremes we find *positional arithmetic*, in which various positional (digital) number representations are employed. The results are computed by means of arithmetical algorithms, which are ordered sequences of table-lookup operations at the level of individual digits.

It may be generally observed that the longest execution times and the least hardware cost will be encountered in primitive arithmetic. Positional arithmetic offers shorter execution times at the cost of more hardware and more complex control sequences; various balances between speed and hardware cost are offered by serial, partially parallel, and fully parallel processing of operand digits. At the other extreme, stored arithmetic offers a fixed short execution time at a very high hardware cost.

B. Errors in an Arithmetical Processor

A *computational error* occurs in the operation of an arithmetical processor whenever its actual behavior deviates from the expected behavior. In the case of an error the value of the result which was produced in response to the inputs (ϕ, X, Y) is not equal to the expected value which was specified by the designer. Errors of an arithmetical processor may be grouped into two categories:

1. A *control error* occurs when the wrong algorithm is executed (ϕ' instead of the specified ϕ).
2. An *arithmetical error* occurs when execution of the specified algorithm ϕ yields a result whose value is not equal to the specified value.

Both types of errors are caused by faults of logic circuits within the arithmetical processor. A *fault* is either a transient (temporary) or a permanent deviation of a logic variable from its specified logic value. Because of the limited reliability of logic circuits, the detection of errors remains an important problem in the design and use of digital computing systems.

The currently more common of two approaches to the detection of computational errors may be called "software checking." In this case the programmer incorporates checking into the program, and the arithmetical results are verified at more or less frequent intervals by means of additional instructions which are redundant in a properly functioning computer. Correction procedures in case of an error also must be programmed. The cost of software checking consists of the additional programming effort, and of the execution time and storage requirements for the redundant instructions. The logic design of the arithmetic processor remains unaffected by software checking.

An alternative to software checking is the relegation of the checking function to logic circuits which are included in the arithmetical processor. This is the "hardware checking" approach. The results and/or the operands of each algorithm are automatically (that is, without programmed commands) tested for acceptability. The indication of an unacceptable result initiates an emergency sequence. This sequence performs an *error correction* by repetition in case of a transient fault or a *fault correction* by means of replacement or repair of the processor in the case when a permanent fault is diagnosed.

The cost of hardware checking consists of the additional logic circuits and of the potentially increased execution times of the arithmetical algorithms. Compared to software checking, the hardware approach offers two advantages. First, the programmer does not need to supply the checking instructions; this is expected to be especially important in programming complex digital systems with many processors, such as the class of "Holland machines." The second advantage is offered by the immediacy and the diagnostic nature of hardware checking. An immediate detection avoids a propagation of the error, corrects (by repetition) the effects of transient faults and facilitates the development of fault correction by means of automatic replacement or repair techniques. The verification of the acceptability of input operands supplied by another processor provides an approach to error-free operation and self-repair of systems employing many intercommunicating arithmetical processors.

The general acceptance of hardware implementations of floating-point arithmetic and the development of hardware algorithms for significant digit arithmetic, both of which replace programmed implementations, are characteristic of the trend to reduce the programmer's load by providing more sophisticated algorithms in the arithmetical processor. It is very probable that the introduction of hardware checking will provide the next step in the development of more effective arithmetical processors. The general acceptance of hardware checking depends on the development of relatively fast and economical arithmetical algorithms which allow immediate detection of the most likely faults within an arithmetical processor.

The following sections of this report present a discussion of the forms and properties of arithmetical errors in conventional number systems, especially in the binary system. An earlier study (Ref. 1) presents a class of arithmetical codes which permit low-cost checking of binary arithmetic. These codes are further evaluated in this paper. Further studies of control error detection, of arithmetical error detection in redundant and residue number systems, and of other methods of arithmetical coding are now in progress and will be reported upon completion.

C. Properties of Arithmetical Errors

There exist many types of codes for the detection and/or correction of errors occurring during the transmission of information. Words in these *transmission codes* generally do not retain their properties when subjected to arithmetical algorithms, and, consequently, transmission codes are applicable only in stored arithmetic. For the checking of positional (digital) arithmetic it is necessary to employ *arithmetical codes*, whose properties are preserved in the results of arithmetical algorithms. Since transmission is a special case of an arithmetical operation, it is evident that the arithmetical codes are subject to more general requirements and therefore are not a subclass of transmission codes, but form a field of study in their own right.

The initial prerequisite for the discussion of arithmetical codes is a description of arithmetical errors and an enumeration of their properties. Since an arithmetical error is defined as a departure from the specified value of the result, the description must indicate this change and furthermore relate it to its potential causes. The change in value is considered to be the effect of an unrequested arithmetical algorithm which has been caused by a fault and has been applied to the correct result. An unrequested addition of an *error number* to the correct result offers a convenient model to describe the effect of any possible arithmetical error. In a positional number system an arithmetical error is recognized when the actual result of an algorithm differs from the specified (i.e., correct) result in at least one digital position. It should be observed that in redundant number representations, in which two or more distinct words may represent the same numerical value, this definition allows errors of value zero. To limit the scope of this paper, further discussion will be restricted to conventional number systems, although most of the development applies to other number systems as well. A conventional number system has a constant positive base $b \geqq 2$, and the only allowed digit values are $0, 1, \cdots, b-1$. Such systems are nonredundant, and every error causes an observable change in the value of a result.

An arithmetical error has occurred if the correct (i.e., specified,) result corresponding to inputs (ϕ, X, Y) is the conventional, base b, digital number S, composed of n digits $(s_{n-1} \cdots s_i \cdots s_0)$, and the actual result is the digital number S*, composed of n digits

$$(s^*_{n-1} \cdots s^*_i \cdots s^*_0)$$

such that

$$s^*_i \neq s_i \qquad (n-1 \geqq i \geqq 0)$$

holds for one or more positions i of the actual result S*. Corresponding to every possible arithmetical error there exists an *error number* E, composed of n digits

$$(e_{n-1} \cdots e_i \cdots e_0)$$

which have the values:

$$e_i = s^*_i - s_i \qquad (n-1 \geqq i \geqq 0)$$

The error number E is a redundant form, since its digits may assume any one of $2b-1$ values:

$$-(b-1), \cdots, -1, 0, 1, \cdots, (b-1)$$

when the values of both s_i and s^*_i range from 0 to $b-1$

inclusively. Properties of redundant number representations of this type are discussed in Ref. 2.

For an example, consider the addition of two 5-bit binary numbers (X, Y) during which the rightmost bit of X is entered incorrectly into the adder:

$$\begin{array}{lll}
\text{Correct Addition:} & & \text{Actual Addition:} \\
X = 0010\underline{\underline{0}} & & X = 0010\underline{\underline{1}} \\
\underline{Y = 10011} & & \underline{Y = 10011} \\
S = 10111 & \text{actual result} & S^{*} = 11000 \\
 & \text{correct result} & \underline{S = 10111} \\
 & \text{error number} & \mathrm{E} = 01\bar{1}\bar{1}\bar{1}
\end{array}$$

In this and all following examples, negative digit values will be represented with the minus sign above the integer ($\bar{1}$ for value -1, $\bar{5}$ for -5, etc.)

The *error value* E of an error number E is given by the conventional expression

$$E = \sum_{0}^{n-1} e_i b^i$$

and is within the range

$$b^n - 1 \geqq E \geqq -(b^n - 1)$$

excluding the value zero.

The *error magnitude* $|E|$ is the absolute value of the associated error number E

$$|E| = \left| \sum_{0}^{n-1} e_i b^i \right|$$

and the range of error magnitudes $|E|$ is

$$b^n - 1 \geqq |E| \geqq 1$$

The error magnitude is the most important single property of an error number, since it determines the detectability of the error by means of an arithmetical checking algorithm.

Since there are $2b - 1$ possible values of every digit e_i, there exist $(2b - 1)^n - 1$ *different forms* of an n-digit error number (the case $e_i = 0$ in all positions is excluded). Every error number E which has a positive value $E = +|E|$ forms a *symmetric pair* with an error number E′ which has the negative value $E' = -|E|$. This negative member E′ of the pair is formed by changing the signs of all nonzero digits of the positive member E $(e'_i = -e_i)$.

Every symmetric pair of n-digit error numbers belongs to one of $b^n - 1$ *magnitude classes* according to the error magnitude $|E|$ associated with the pair (E, E′). It is evident that because of the redundancy of the error numbers most magnitude classes contain more than one symmetrical pair of error numbers. The pairs contained in one magnitude class will differ in the locations, values and the total count of nonzero digits.

For example, the binary error number of the preceding example ($\mathrm{E} = 01\bar{1}\bar{1}\bar{1}$) forms a symmetrical pair with the form $\mathrm{E}' = 0\bar{1}111$. This pair belongs to the magnitude class $|E| = 1$, which also contains four other 5-bit positive pair members 00001, $0001\bar{1}$, $001\bar{1}\bar{1}$, $1\bar{1}\bar{1}\bar{1}\bar{1}$ and the negative members of each pair.

The second significant property of every symmetrical pair of error numbers is the count of nonzero digit values, called the *multiplicity* μ of the pair. In any magnitude class $|E| = K$, one or more symmetrical pairs of error numbers will have the least value of μ. This minimal multiplicity is designated by $w(K)$ and is called the *weight* of the error magnitude K. In the example given above, the magnitude class $|E| = 1$ has the weight $w(1) = 1$, determined by the form 00001 which has the least multiplicity $\mu = 1$. The weight $w(K)$ frequently serves as an indicator of error probability. For instance, it indicates the least number of nonzero digits which an error number must contain in order to cause an error of magnitude K by means of an unlimited-length parallel addition. Carry propagation will cause the appearance of other error numbers from the magnitude class K in this case.

Another important property of an error is the *damage pattern*, which is the distribution pattern of nonzero digit values and of their signs in the associated error number. This pattern relates the error to a fault of a logic circuit, especially in the case when the circuit is employed more than once during a given algorithm.

Both weights and patterns provide means of relating arithmetical errors to their causes. The knowledge of potential causes, in turn, permits the allocation of relative probabilities to error numbers and to magnitude classes, and a subsequent evaluation of the effectiveness of various methods of checking. Unless otherwise specifically stated, the "worst case" approach will be used in evaluation: if several causes may generate a given error number, the most probable cause must be assumed.

II. ORIGINS OF ARITHMETICAL ERRORS

A. The Causes of Arithmetical Errors

In order to assign relative probabilities to various error magnitudes, it is necessary to consider the causes of arithmetical errors. An arithmetical error is caused by one or more faulty logic circuits, which fail to implement their specified functions. For the purpose of this discussion, the term *component* refers to an electronic component (resistor, capacitor, diode, transistor, etc.), the term *circuit* refers to a logic circuit (either combinational or a binary storage element), and the term *logic net* refers to a network of logic circuits.

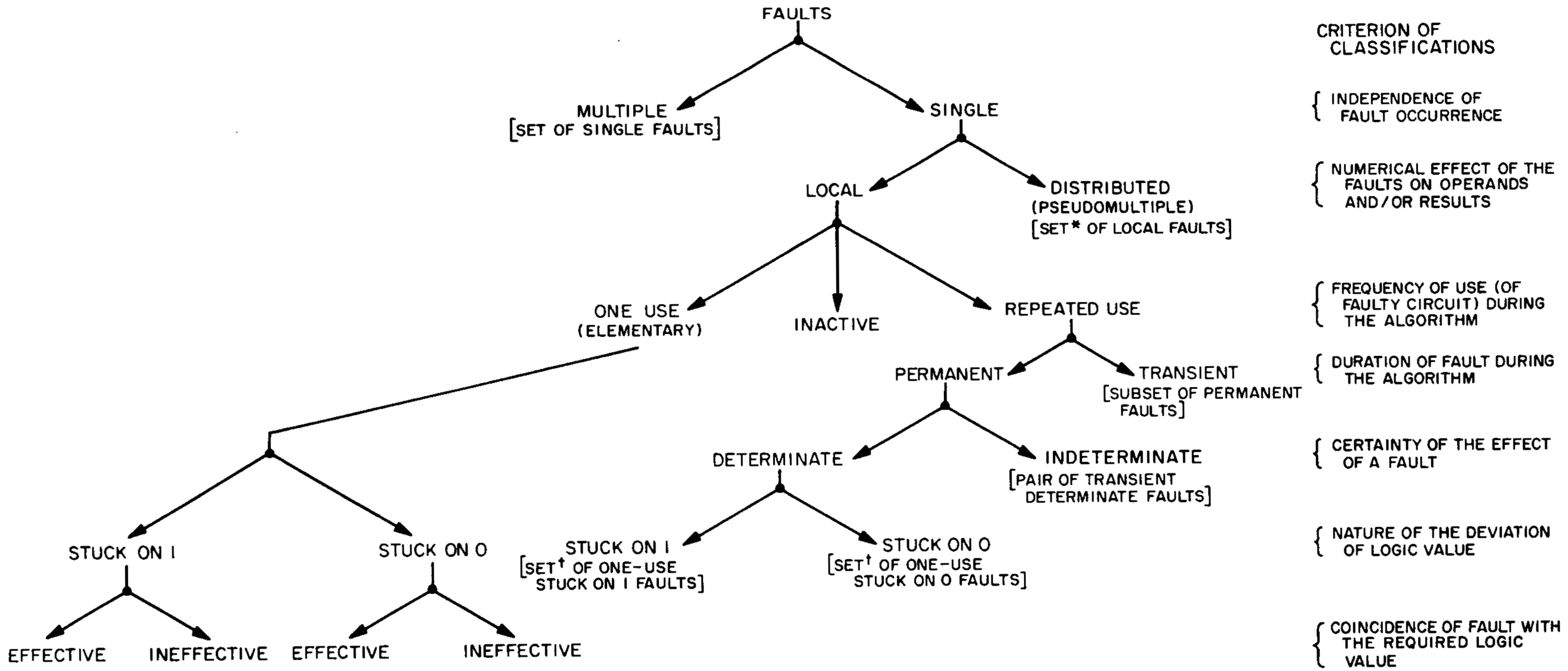
The term *fault* will be employed to designate the deviation of logic variables from their specified values at one or more points within a logic net. The fault either causes the variable to be inverted or forces it to assume a constant logic value. The term "fault" thus encompasses the undesirable effects both of the physical failures of components and of the injection of improper signals by a noisy environment. A convenient classification of faults is shown in Fig. 1, together with the different criteria applied at each level. The figure demonstrates that the effect of any fault may be expressed as the cumulative effect of a set of single, local, one-use faults. The potential membership of this set and the method by which the effect accumulates is determined by the type of the fault and by the algorithm which is being executed using the faulty circuit. The actual membership is selected from the potential membership by the operand(s) of the algorithm, since generally some faults from the potential membership will be ineffective (the fault-induced value will correspond with the specified value).

A *single fault* is a fault which is caused by one fault-inducing event; the term *multiple fault* (double, triple, etc.) refers to several single faults (caused by independent events) occurring simultaneously in time, that is, during the same algorithm. The effect of a multiple fault is described in terms of its component single faults. A single fault may be *local* or *distributed* (pseudomultiple) with respect to a logic net and a number system. Generally, a local fault inflicts a certain minimal measure of damage on the result of a specified algorithm which is elementary in the given processor. In binary arithmetic a *local fault* is defined as causing a one-unit error in one digital position of either an operand or a result of an elementary algorithm (one-bit transfer, complementation, shift, addition), while a *distributed fault* has a more extensive effect on one or more digital positions. (For example, a single distributed fault would cause both sum and carry outputs of a binary adder circuit to be 1 when the correct outputs are both 0). A distributed fault is described as a multiple occurrence of single local faults (in two adjacent positions of the sum in the above example). Prevention of distributed faults is an objective of logic design.

Faults are further classified with respect to the algorithm which is being performed. If a faulty circuit is not employed by the algorithm, the fault is *inactive*. If the faulty circuit is employed once, the fault is designated as a *one-use* fault, and if it is employed more than once, the fault is a *repeated-use fault*. In order to facilitate fault analysis, it is convenient to distinguish three types of algorithms with respect to every logic circuit in an arithmetic processor with checking provisions. An *unrelated* algorithm does not employ the given circuit. A *one-use algorithm* employs the given logic circuit only once before a checking algorithm is applied to the result. A *repeated-use algorithm* employs the given logic circuit more than once before a check is performed. It is evident that a repeated-use algorithm may be changed into a sequence of one-use algorithms by increasing the frequency of checking; however, the time required by the algorithm will be increased by the change. For example, conventional multiplication in a parallel arithmetic unit is a repeated-use algorithm with respect to the adder circuits, while the introduction of a check after every addition will convert it into a sequence of one-use algorithms and slow down its execution. On the other hand, the repeated use of a faulty circuit will compound the effects of the fault on the result, and fault detection will be less probable.

In the case of one-use local faults in binary arithmetic processors the only manifestation of the fault is the inversion of the value of one binary digit. This inversion frequently is due to a logic variable assuming a constant value; therefore, it is convenient to say that the point representing the digit value is "stuck on 0" or "stuck on 1" during the use of the faulty circuit. When the fault-induced value corresponds with the required value, the fault does not affect the result and is said to be *ineffective;* otherwise it is *effective*.

The situation becomes considerably more complicated in the case of repeated-use faults, since the duration and the variability of the fault condition need to be considered. A fault is said to be *permanent* if it persists throughout the entire algorithm, and *transient* if it does not exist



*MEMBERSHIP OF SET DETERMINED BY LOGICAL DESIGN OF THE NET AND BY THE NATURE OF THE FAULT

†MEMBERSHIP OF SET DETERMINED BY THE ALGORITHM BEING IMPLEMENTED

Fig. 1. Fault classification

during one or more uses of the circuit. A transient fault is considered as a permanent fault which is ineffective during the uses for which the fault does not exist, and is either effective or not during the remaining uses of the faulty circuit. It is also necessary to observe that a defective component may cause an arbitrary change in the logic function of a circuit (e.g., complement the required output). Furthermore, the logic value at a point may become uncertain and be interpreted randomly as either one or zero during the repeated uses of the faulty circuit. In these cases the point at which the fault occurs is said to be "stuck on X," and the fault is said to be *indeterminate*. The faults which appear as the "stuck on 0" and "stuck on 1" conditions are said to be *determinate*.

A *determinate fault* in a repeated-use situation will be either effective or ineffective during each use, depending on whether the fault condition coincides with the correct value or not. When the algorithm requires k uses, the faulty circuit will generate one of 2^k possible damage patterns, one of which is not observable (fault is ineffective during all k uses). The effect of the repeated-use fault is the same as the cumulative effect of a set of one-use faults corresponding to the uses during which the fault was effective. The details of the damage pattern are determined by the algorithm which is being executed.

An *indeterminate fault* is conveniently interpreted as a pair of transient determinate faults, since during each specific use the "stuck on X" point will behave as either "stuck on 0" or "stuck on 1," or the fault will be ineffective. Consequently, k uses of the faulty circuit will generate one of 3^k possible damage patterns (one of which is ineffective throughout all k uses). The effect of the indeterminate fault is then the cumulative effect of two determinate faults, one of the "stuck on 0" and the other of the "stuck on 1" type.

The preceding consideration of faults permits the conclusion that the effect of a fault in the circuitry of an arithmetical processor may be stated in terms of the cumulative effect of an equivalent sequence of single local one-use faults, to be called *elementary* faults, with only the effective members contributing to the error value. The type of the fault and the values of input Boolean variables determine whether a fault is effective or not during any given use of the faulty circuit. The membership in the equivalent sequence of elementary faults is determined by the type of fault, the logic design of the arithmetical processor and the specific algorithm being executed.

The purpose of this classification is to establish for any fault an equivalent sequence of elementary faults. When the error value which is contributed by an elementary fault is known, the equivalent sequence, together with the design details of the arithmetic processor, will indicate the error value or the set of potential error values which will be caused by a given fault. The probability of this fault (obtained from circuit considerations) now is related to one or several magnitude classes of error numbers.

The effects of faulty circuits will be considered for the case of most immediate interest, that is, for binary arithmetic. The direct correspondence between digit values and the values of logic variables in binary arithmetic greatly facilitates the identification of an elementary fault with the value of an error number.

B. Fault Effects in One-Use Binary Algorithms

The observable symptom of a fault within an arithmetic processor is the error number. The relationship between a fault and the resultant error number is determined by the algorithms, the number system, and the logic design of the processor. It is convenient to define an elementary fault in binary arithmetic to be a fault which causes a one-unit error in one digital position of either an operand or a result of an *elementary algorithm*. This is a minimal amount of damage in a simple operation. The elementary digit algorithms of a binary arithmetic processor with respect to error generation are one-bit transfer, complementation, shifting, and addition.

Common one-use parallel word algorithms in binary arithmetic are the transfer, shifting, and bit-wise complementation of a binary number. In these algorithms an effective elementary fault will generate an error number of magnitude 2 . The weight $w(2^j)$ and the multiplicity μ of this error number are both 1; that is, the error number has a single $+1$ or -1 digit in the fault position j. Parallel addition is a one-use algorithm which requires more detailed consideration because of the finite length of an adder.

Parallel binary addition (or subtraction) is performed modulo A (with $A_1 = 2^n - 1$ or $A_2 = 2^n$) in binary arithmetic processors. An effective elementary fault in position j will add an error number E_j to the correct result. The values which E_j may assume in parallel binary modulo A addition are:

1. $E_j = 2^j$ (a unit was added in position j of the result).
2. $E_j = -2^j$ (a unit was lost in position j of the result).

3. $E_j = -A + 2^j$ (an erroneous carry-out occurred because of 1).
4. $E_j = A - 2^j$ (a required carry-out did not occur because of 2).

The same values occur in parallel binary modulo A subtraction. An ineffective elementary fault will not be observable ($E_j = 0$). The observable error numbers will belong to two magnitude classes:

$$|E_j| = 2^j$$

and

$$|E_j| = A - 2^j$$

with $n - 1 \geqq j \geqq 0$. The error magnitude 2^j has the weight $w(2^j) = 1$, while the multiplicity depends on the carry (or borrow) propagation changes caused by the fault.

The weight $w(A - 2^j)$ and the multiplicity of the associated error number depend on the choice of A and on the value of j. Generally, the error magnitudes 2^j and $A - 2^j$ form one class with respect to parallel modulo A addition and subtraction, since both can be caused by an elementary fault. When $A = 2^n - 1$ is chosen, the error magnitude $A - 2^j$ corresponds to a form in which there is only a single zero digit in the position j; all other digits are either all $+1$, or all -1. This error number (e.g., 1101111) is recodable into a minimal form (Ref. 1, p. 3) possessing the restricted property M (e.g., $111000\bar{1}$). This form has the weight $n - j + 1$, with "1" digits in positions j to $n - 1$, inclusive, and a $\bar{1}$ digit in position 0. Exceptions are $j = 0$ (e.g., 111110) and $j = 1$ (e.g., 111101), which are already in the restricted minimal form, and have the weight $n - 1$. The choice $A = 2^n$ yields the error magnitude $2^n - 2^j = (2^n - 1) - (2^j - 1)$, which corresponds to the restricted minimal form of weight $n - j$, with $n - 1 \geqq j \geqq 0$.

In the case of two independent elementary faults, either one of which may generate the error values $\pm 2^j$ and $\pm(A - 2^j)$, the cumulative error value will be

$$E = E_j + E_i$$

with $n - 1 \geqq j \geqq i \geqq 0$. Modulo A summation of the possible error values yields 8 discrete results:

$$\pm(2^j \pm 2^i)$$

and

$$\pm[A - (2^j \pm 2^i)]$$

The error magnitudes are then

$$|E_2| = 2^j \pm 2^i$$

and

$$A - |E_2|$$

and the undetectable errors are the "weight 2" errors (Ref. 1, p. 12) and their complements with respect to A. The analysis may be directly extended to three or more independent elementary faults.

C. *Fault Effects in Repeated-Use Binary Algorithms*

The parallel binary adder and the shifting circuits are used repeatedly in multiplication and division. In these cases a local fault will contribute an error number during each use of the faulty circuit. The value of this error number will be zero when the fault is ineffective, and $\pm 2^j$ or $\pm(A - 2^j)$ when the fault is effective. In shifts, transfers, and complementation, the cumulative effect of the error numbers is addition of the values $\pm 2^j$ or zero (without carry propagation) for each use. In an addition, the effect is modulo A addition (with carry propagation) of $\pm 2^j$ or $\pm(A - 2^j)$ or zero for each use. Any one of the listed values may be contributed by an indeterminate fault ("stuck on X") during any one use. A determinate fault ("stuck on 0," or "stuck on 1") will contribute the 2^j term with one sign only. In an addition, "stuck on 0" contributes only -2^j, $A - 2^j$, or zero; "stuck on 1" contributes only 2^j, $-A + 2^j$, or zero. It is readily demonstrated that the error value $\pm(A - 2^j)$ can occur only once (or not at all) during one algorithm.

It is necessary to note that the partial products (or partial remainders) which are held in the accumulator are shifted with respect to the adder circuitry and the multiplicand (or divisor) in most parallel arithmetic processors. A local fault in position j of the adder circuitry or of a register then will contribute one-unit errors to various positions of the result in the accumulator. In this case, j designates a set $\{J\}$ of locations in the result. The membership of this set is determined by the details of the algorithm and by the detailed logic design of the arithmetic circuits. For example, a string of adjacent positions in the product will belong to $\{J\}$ when multiplication employs a right shift of one position. The preceding considerations show that the choice of algorithms and details of logic design affect the error patterns which will be generated by a local fault. A search for optimal algorithms and designs with respect to error detection is therefore an additional objective of arithmetic processor design. For example, elimination of single shifts in multiplication will alter the error pattern in a favorable direction.

The elimination or control of distributed faults is a further design consideration. For example, a faulty flip-flop in an operand register with a provision for adding the multiples ±1 and ±2 of the multiplicand (or divisor) to the accumulator will affect two adjacent digital positions of the adder and therefore is a distributed fault.

Further important cases of repeated-use algorithms are found in a *byte-organized* computer. A computer is said to be byte-organized when data transfer and addition are performed b bits at a time (in a "series-parallel" manner). One b-bit section of the n-bit word is called a *byte*, and b is the *byte length*. One effective local fault will contribute a one-unit error in the same relative position of every byte of the result. When one machine word consists of k bytes ($n = kb$), a determinate local fault will contribute one of 2^k possible error values. An indeterminate local fault ("stuck on X") in this case generates one of 3^k error values. Furthermore, the complement magnitudes $A - |E|$ also may occur during an addition.

D. Criteria for Hardware-Checking Methods

Hardware checking was described as an automatic validity test of results and/or operands in an arithmetic processor. The presently known methods of hardware checking are:

1. Partial (separate check symbols) or complete duplication of the arithmetic processor, followed by a comparison of results and decision.
2. Arithmetical coding of all operands; the coding is retained by the results and permits their verification by a checking algorithm.

The previously proposed criterion for the choice of preferred codes and check symbols is the detection of all error numbers which occur in parallel additions or transfers and belong to magnitude classes with the low weights 1, 2, and 3 (Ref. 3). This "fixed distance" requirement has been adapted from data transmission codes.

It must be noted that the fixed distance criterion does not take into account the process of error generation and the design of the arithmetic processor. A more directly relevant *criterion of effectiveness* is high probability of detection of the most likely faults by testing the results of all algorithms of the given arithmetic processor. This criterion relates the logic design and the choice of algorithms to the checking requirement and attempts to maximize the effectiveness of the entire system with respect to fault detection. Fault correction by replacement or repair is a direct consequence of fault detection, therefore "error correction" in the sense of transmission codes is not considered to be necessary. The error magnitudes which are to be detected with the greatest probability can be established by using the methods developed in the preceding sections. The *percentage* of detectable magnitudes from this class is considered to be the measure of effectiveness.

A second criterion which is essential in the evaluation of hardware checking is the *cost* associated with a given method. The cost of incorporating a method of hardware checking into a processor depends on two factors:

1. The *compatibility* of the checking method with the algorithms of the processor.
2. The *direct cost* of the checking method.

A measure of compatibility is the *additional time* required by the checked arithmetic processor to execute its algorithms and the increase in the *hardware complexity* when compared to the unchecked arithmetic processor, including its control circuits.

The direct cost of checking consists of the hardware and extra time consumed by the *checking algorithm* and the increase in *storage requirements* (redundancy) which is due to increased length of arithmetical operands and results.

It is evident that a systematic evaluation of hardware checking must be based on both effectiveness and cost considerations. A practical choice will be attained by considering these criteria for various checking techniques and by selecting the desired balance between the cost and the probability of fault detection.

III. EFFECTIVENESS OF A CLASS OF PRODUCT CODES

A. Properties of Product Codes

A *product-coded* number Z is obtained when a conventional number X is multiplied by an integer $\alpha > 1$, which is called the *check factor*. Given the in-range set of digital numbers $\{X\}$ of a digital computer, a *product code* is obtained by multiplying every number by the check factor α. When α is relatively prime to the base r of the number X and $\alpha > r$, then any arbitrary change ($\pm C$, with $C < r$) in the value of one digit (z_i) of the number $Z = \alpha X$ will change its value to $Z^* = \alpha X \pm Cr^i$. A division of Z^* by α will yield a nonzero remainder, since Cr^i is not an integer multiple of α. The nonzero remainder indicates that the change has occurred; consequently, the product code has error-detecting properties. For base 2 numbers, any odd integer $\alpha > 2$ will provide the above-discussed property, i.e., the detection of error numbers of weight 1. Values of α which provide detection of all error numbers of weights 2 and 3 and correction of weight 1 errors for a limited range of X have been described (Ref. 3, 4, 5, and 6), as well as values of α for burst-error detection and correction (Ref. 7 and 8). The cost of checking and the detection of errors due to repeated-use faults were not considered in these codes.

The search for values of α which are characterized by high compatibility with binary arithmetic and a low-cost checking algorithm (Ref. 1) led to the choice of product codes which employ check factors α of the form

$$\alpha = 2^a - 1$$

with integer $a > 1$. The parameter a is called the *check length* of the code. The check factor $2^a - 1$ is applied to encode $(k - 1)$ a-bits-long conventional binary numbers X into ka-bits-long product-coded binary numbers $Z = (2^a - 1)X$. Extension of these codes to other bases $r > 2$ is effected by choosing $\alpha = r^a - 1$.

Generally, all odd integers $\alpha > 2$ are applicable as check factors in binary product codes. The *checking algorithm* for all product codes is a zero test of the remainder after dividing the coded number $Z = \alpha X$ by the check factor α. A zero remainder indicates that either a valid result has been obtained, or an undetectable error (to be called a *miss*) has occurred. Since division is a complex arithmetic algorithm, the checking algorithms for odd $\alpha > 2$ are relatively costly. An exception occurs for the check factor $2^a - 1$. The exception is due to the congruence:

$$Kr^i \equiv K \text{ modulo } (r - 1)$$

The choice $r = 2^a$ in the above congruence permits the use of modulo $2^a - 1$ summation of the k bytes (a-bit segments of value K, with $0 \leq K \leq 2^a - 1$) which compose the number Z to compute the least positive residue of Z modulo $2^a - 1$. In this special case division is replaced by a simple "end-around carry" addition algorithm, which may be executed either sequentially or simultaneously for all k bytes. Implementations which do not require addition have been described for the $a = 2$ case (Ref. 9 and 10).

It was already noted that the check factors $2^a - 1$, with $a > 1$, will detect all weight 1 error magnitudes 2^j, with $0 \leq j \leq ka - 1$. Furthermore, all damage patterns which are confined within $a - 1$ adjacent bits of the error number (bursts of length $a - 1$ or less) will be detected, as well as the "one's complements" $(2^{ka} - 1) - C$ of all detectable error magnitudes C. All weight 2 error magnitudes will not be detected; the fraction f of undetected weight 2 error magnitudes for $a > 2$ is given by the expression (Ref. 1):

$$f = \frac{(k-1)\,a}{2a\,(ka - 3) + \dfrac{6}{k}}$$

i.e., $f < 1/2a$; for example, given $ka = 24$, $a = 3$ yields $f = 0.166$; $a = 4$ yields $f = 0.118$, and $a = 6$ yields $f = 0.071$. The case of $a = 2$ is an unfavorable exception, yielding $f = 0.5$ for any value of k. The analysis may be continued for higher weights, due to several independent elementary faults; however, errors due to repeated use of a single faulty circuit are of more immediate interest, and will be considered in later sections.

B. Binary Arithmetic for Coded Operands

The entire set of binary arithmetical algorithms is conveniently adaptable to the product codes with the check factor $2^a - 1$ and word length of ka bits (Ref. 1). An important constraint on the choice of a for binary n-bit product-coded numbers $Z = \alpha X$ is the requirement that the code should be complementable with respect to

$A = 2^n$ or $A = 2^n - 1$ in order to implement the subtraction algorithm. Assuming that the uncoded numbers are complemented with respect to an arbitrary value B, we have the requirement:

$$A - \alpha X = \alpha (B - X)$$

or

$$A = \alpha B$$

that is, α must be a factor of A. Since α is odd, this immediately excludes $A = 2^n$. Consequently, a *complementable product code* can be obtained if and only if

$$2^n - 1 = \alpha B$$

is satisfied. All possible choices of α are found in a table of the factors of $2^n - 1$ (Ref. 1, pp. 28–29). To employ other choices of α, a product-and-sum ("$An + B$") code must be used (Ref. 3 and 5). All codes with $n = ka$ and $\alpha = 2^a - 1$ are complementable; the well-known "one's complement" algorithms apply directly, including complementation, sign detection, range extension, range contraction, left and right arithmetical shifting, and addition. It must be noted that the next acceptable word length (after ka bits) is $(k + 1)a$ bits; consequently, range is extended by prefixing a replicas of the leftmost ("sign") bit. Similarly, range can be contracted by dropping a leftmost bits when $a + 1$ leftmost bits are identical. Checked variable-field-length addition is made possible by the range extension algorithm.

Multiplication and division of the coded number Z by the check factor $2^a - 1$ are needed in forming coded products and quotients. The $(2^a - 1)Z$ algorithm requires one parallel modulo $2^n - 1$ addition: Z is extended by a bits, shifted a positions left and added to its own extended complement $\bar{Z}$ to form:

$$2^a Z + \bar{Z} = (2^a - 1)Z$$

The inverse operation, i.e., the $(2^a - 1)Z/(2^a - 1)$ algorithm generates the result a bits at a time, employing an a-bit parallel adder to implement the solution

$$Z = \overline{(2^a Z + \bar{Z}) + \overline{2^a Z}}$$

where the bar designates "one's complement" and the addition is modulo $2^n - 1$, n being the length of operand $(2^a - 1)Z$. A double-length coded product $(2^a - 1)XY$ of the coded operands $(2^a - 1)X$ and $(2^a - 1)Y$ is obtained by forming the ordinary binary product $(2^a - 1)^2 XY$ and then dividing by $2^a - 1$. Given the dividend $(2^a - 1)X$ and divisor $(2^a - 1)Y$, the quotient $(2^a - 1)Q$ is obtained by forming $(2^a - 1)^2 X$ and then executing the ordinary binary division which yields the coded quotient $(2^a - 1)Q$ and remainder $(2^a - 1)R$ satisfying

$$(2^a - 1)^2 X = (2^a - 1)Q(2^a - 1)Y + (2^a - 1)^2 R$$

Rounding off by truncation will not yield a product-coded number; the algorithm must be modified. To round off the rightmost ca bits of the coded operand $(2^a - 1)\ X$ we add the constant $(2^a - 1)\ G$, such that in $(2^a - 1)(X + G)$ the rightmost ca bits are identical to the leftmost ("sign") bit and may be dropped. The $(2^a - 1)X/(2^a - 1)$ algorithm is employed to obtain ca rightmost bits of X. The value of G is then selected to implement either up-rounding or down-rounding.

In summary, it is noted that product codes with the check factor $2^a - 1$ display an exceptional adaptability to binary arithmetic and a low-cost checking algorithm, if the length of operands is chosen to be a multiple ka of the check length a. Variable-field-length arithmetic of coded operands is made possible by the range extension and contraction algorithms.

C. Fault Location by the Preferred Codes

The checking algorithm is implemented as the modulo $2^a - 1$ summation (with an "end-around carry" around a bits) of a bits long *groups* of the coded operand Z, executed simultaneously, or serially by group, or serially by bit. The *check value* is the least positive residue (modulo $2^a - 1$) of Z, represented by the *check result*, which is an a bits long binary number $F(f_{a-1}, \cdots, f_h, \cdots, f_0)$ of value

$$F = \sum_{h=0}^{a-1} f_h 2^h \qquad f_h = 0, 1$$

When the "all zero" coded number Z with $z_i = 0$ $(0 \leq i \leq ka - 1)$ is excluded from valid operands, all coded operands will yield the check value $F = 2^a - 1$, represented by $f_h = 1$ in all a positions $(0 \leq h \leq a - 1)$ of F. Any other check value indicates an error magnitude which is not a multiple of $2^a - 1$. The check results due to various error values which are of special interest may now be investigated. The notation $(2^a - 1)|Z$ will be employed to designate the check value, i.e., the least positive residue modulo $(2^a - 1)$ of the value Z. It is explicitly noted that the value $F = 2^a - 1$ is allowed instead of the residue value 0 (zero).

Although fault detection is sufficient for an algorithmic arithmetic unit, correction of errors due to elementary faults may be desirable for operands transferred from memory or from inputs. Fault location also may be employed in the diagnosis and subsequent repair of the faulty circuit.

To attain complete location of an elementary fault, or to effect correction of weight 1 error magnitudes and their "one's complements" it is necessary to locate the position i and to determine the sign of 2^i for the error value pairs $\{+2^i, -(2^{ka} - 1) + 2^i\}$ and $\{-2^i, (2^{ka} - 1) - 2^i\}$ which are caused by a one-use elementary fault during a transfer, complementation, shift or parallel addition. The modulo $2^a - 1$ residues of these error values are therefore of interest.

The error value $E = +2^i$ corresponds to a binary error number E containing a single nonzero digit $e_i = 1$. Stating E as a base 2^a number, we have

$$2^i = 2^{i-ja}(2^a)^j$$

with

$$0 \leqq i - ja \leqq a - 1$$

and

$$0 \leqq j \leqq k - 1$$

when E is ka bits long $(0 \leqq i \leqq ka - 1)$. The check result F obtained from E will contain a single "1" digit $f_h = 1$ in the position $h = i - ja$, since $2^i \equiv 2^{i-ja}$ modulo $(2^a - 1)$, and $(2^a - 1)|2^i = 2^{i-ja}$. Consequently, the check value for any invalid operand Z^* of value $Z + 2^i$, where Z is any valid operand, will be obtained as follows:

$$(2^a - 1)\Big|(Z + 2^i) = (2^a - 1)\Big|2^i = 2^{i-ja}$$

The error value $E = -(2^{ka} - 1) + 2^i$ yields an invalid operand of value $Z^* = Z + 2^i - (2^{ka} - 1)$, since the error caused an erroneous "end-around carry" in an addition; that is, $Z \leqq (2^{ka} - 1)$, but $(Z + 2^i) > (2^{ka} - 1)$. Since $2^{ka} - 1 = Z + \bar{Z}$, where $\bar{Z}$ is the "one's complement" of Z, we have:

$$Z^* = Z + 2^i - (Z + \bar{Z}) = 2^i - \bar{Z}$$

with $2^i > \bar{Z}$. Again, the check value F is the modulo $2^a - 1$ residue of Z^*:

$$F = (2^a - 1)\Big|(2^i - \bar{Z}) = (2^a - 1)\Big|[(2^a - 1)|2^i - (2^a - 1)\Big|\bar{Z}] = 2^h$$

We note that the error values $+ 2^i$ and $-(2^{ka} - 1) + 2^i$ both will yield the check result F containing a single "one" digit $f_h = 1$ in the position $h = i - ja$, all other $a - 1$ bits being "zeros."

Given the error value -2^i, the check value F will be computed from the invalid operand value $Z^* = Z - 2^i$ as follows:

$$F = (2^a - 1)\Big|(Z - 2^i) = (2^a - 1) - 2^{i-ja}$$

where i, j, and the word length ka are as defined earlier. The check result F generated by the checking algorithm will contain only one "zero" digit $f_h = 0$ $(h = i - ja)$; all $a - 1$ other digits will be "ones."

When the error value -2^i inhibits a required "end-around carry," the resulting invalid operand Z^* has the value:

$$Z^* = Z + (2^{ka} - 1) - 2^i$$

The check value is again $F = (2^a - 1) - 2^h$, as it was for the error value -2^i. It is observed that errors in the "end-around carry," which may occur in the modulo $2^{ka} - 1$ adder, do not affect the recognition of the position $h = i - ja$ or of the sign of the error values $\pm 2^i$.

The preceding results demonstrate that the error values which were considered cause check results which indicate the *sign* and the *group position* $h = i - ja$, with $0 \leqq h \leqq a - 1$, and $0 \leqq i \leqq ka - 1$. The *group index* j, with $0 \leqq j \leqq k - 1$ remains unknown; there are k acceptable choices of j. For complete location of the elementary fault (or for error correction) the group index must be determined. It is noted that the concept of the group position is very useful in the analysis of repeated-use fault symptoms.

D. Application of Several Check Factors

A product-coded number may be further encoded by using a second check factor. If two check factors of the $2^a - 1$ class have relatively prime check lengths a_1 and a_2, then the pair of group positions (h_1, h_2) will uniquely identify one position in a group of a_1a_2 bits. Furthermore, the undetectable error magnitudes of weight 2 are now only those which are integer multiples of $2^{a_1a_2} - 1$. With respect to fault location and weight 2 error detection the total effect of the two check factors $2^{a_1} - 1$ and $2^{a_2} - 1$ is the same as that of a single check factor $2^{a_1a_2} - 1$. The checking algorithm is executed separately for each check factor, either simultaneously or serially. The effective check length a_1a_2 has been obtained by using only $a_1 + a_2$ bits for redundancy of encoding.

The preceding argument extends directly to any set of m check factors $2^{a_i} - 1$, if the check lengths a_i are pairwise relatively prime. The combined elementary fault locating effect of m check factors is equivalent to that of the check factor $2^p - 1$, which requires only s bits for encoding, where

$$p = \prod_{i=1}^{m} a_i$$

and

$$s = \sum_{i=1}^{m} a_i$$

The choice of $p = n$ will provide complete location of i $(0 \leqq i \leqq n - 1)$ and sign determination for the error values $\pm 2^i$ and $\pm (2^n - 1 - 2^i)$ which were considered in the preceding section. All weight 2 error magnitudes will be detected, as well as their "one's complements," caused by errors in the end-around carry. All bursts of length $s - 1$ or less will also be detected.

For example, the check lengths $a_1 = 3$, $a_2 = 4$, $a_3 = 5$ will give $p = 60$ and $s = 12$; this means that 48-bit numbers will gain 12 redundant bits in encoding, while allowing the location of a one-use elementary fault in the 60-bit coded operand. One checking algorithm is sufficient to indicate the presence and the sign of the error value; the results of the remaining two will pinpoint its location by means of the three group positions (h_1, h_2, h_3).

E. Detection of Repeated-Use Determinate Faults

A defective component (resistor, transistor, etc.), or external interference may cause the deviation of a logic variable (formed or held by a logic circuit) from its specified value; this deviation is called a fault. The values of logic variables correspond directly to the values of binary digits in a digital computer employing the conventional base 2 number system; consequently, a binary number system presents the most direct relationship between a fault and the resulting change in the value of a result. An elementary fault causes a one-unit error in one digital position either of an operand or of a result in an elementary algorithm (one-bit transfer, complementation, shifting, addition). The effects of more damaging faults may be expressed as the composite effect of a set of elementary faults. An elementary fault is the most probable fault in the processor when suitable precautions are observed in logical design.

In parallel transfer, shifting and bit-wise complementation of an n-bit number, an elementary fault generates an error magnitude 2^j, which has the weight 1. In modulo A parallel addition (or subtraction), the error magnitude is either 2^j or $A - 2^j$. In multiplication and division, the parallel adder and associated circuits are used repeatedly, and an elementary fault will contribute an error number during some or all uses of the faulty circuit. The repeated-use faults are also encountered in transfers, shifts, and additions in a variable-field-length (byte-organized) arithmetic processor.

In the case of a repeated-use fault, the specific nature and the duration of the fault require further consideration. A determinate fault causes the logic variable to assume a constant value; it is either "stuck on 0" or "stuck on 1." An indeterminate fault occurs when the logic value becomes uncertain and is interpreted randomly as either "1" or "0" during the repeated uses of the faulty circuit. A fault is transient if it is not present during one or more uses of the circuit during a given algorithm; otherwise, it is permanent. During any one use of the faulty circuit, the fault-induced value may correspond with the specified value of the logic variable; the fault is ineffective during this use. Consequently, error numbers due to transient faults are a subset of those due to permanent faults.

A permanent determinate fault (used c times) will be manifested by one of 2^c possible ways of contributing to the error magnitude; the possible error magnitudes are determined by the algorithm being performed. A damage pattern of immediate interest is the *regular pattern* in which c binary positions: $i = h$, $h + b$, $h + 2b$, etc., are affected. It contributes either $e_i = 0$ or $e_i = \pm 2^i$ ($+2^i$ for "stuck on 1," -2^i for "stuck on 0") in the corresponding position of the error number. This pattern occurs in a byte-organized (variable-field-length) computer during transfer, complementation, and addition; it will be also observed in parallel multiplication and division which employs only b-bit shifts. The "one's complements" of error numbers will occur in algorithms employing the adder. The value b is called the *byte length*, and the value a is called the *check length* when the check factor

$2^a - 1$ is employed. The length of coded numbers is $n = k_b b$ bits, or k_b bytes. The product code also requires that $n = k_a a$ should be satisfied, therefore the relationship $n = k_b b = k_a a$, with integers $k_b \geqq 1$, $k_a \geqq 1$, must be maintained between a and b.

The preceding consideration of the fault location properties (Section III-C) may be employed to derive expressions for the effectiveness of elementary fault detection for various choices of the pair (a, b). A miss (undetectable error magnitude) will occur whenever the error values contributed by the damaged bytes of the coded number will cause a check result consisting of a "ones" to be generated. In the case of an elementary determinate fault, given any pair (a, b) of the check and byte lengths, the first miss occurs when the word length reaches the value

$$n_1 = c_a a \left(2^{a/c_b} - 1\right)$$

where $c_a a = c_b b$ is the least common multiple of a and b. Consequently, the choice $c_b = 1$, $c_a \geqq 1$ will yield the maximum values for n_1. The minimum value of n_1 is obtained when a and b are relatively prime; in this case $n_1 = ab$.

The choice of $b = a$ makes the byte count equal to the count of check lengths in a word: $k_b = k_a = k$; it also gives a very favorable safe length. In the case $b = a$, all $2^k - 1$ possible error magnitudes and their "one's complements" will be detected for word lengths up to $n = a\,(2^a - 2)$ bits, or $k = 2^a - 2$ bytes of a bits each. One miss occurs for $k = 2^a - 1$ bytes; with greater word lengths the count of misses ϵ for $k > 2^a - 1$ is given by the expression

$$\epsilon = \sum_{1}^{q} \epsilon_j$$

where q is the integer part of $k/(2^a - 1)$, and

$$\epsilon_j = \frac{k!}{[j(2^a - 1)]!\,[k - j(2^a - 1)]!}$$

for $1 \leqq j \leqq q$. The expressions for the miss count ϵ are derived by considering all possible ways in which a check result value $2^a - 1$ consisting of all "ones" can be generated by modulo $2^a - 1$ summation of k contributions of either 0 or 2^h, with $0 \leqq h \leqq a - 1$.

The effectiveness of any choice of the pair (a, b) can be expressed in terms of the percentage of misses among all possible $2^{k_b} - 1$ nonzero error magnitudes which can be caused by an elementary determinate fault. Given a *miss count* ϵ, the *miss percentage* is obtained as

$$\frac{100\epsilon}{2^{k_b} - 1}$$

where $n = bk_b$ is the word length of the coded operands. The miss percentages for various word lengths were obtained using a computer program which tabulated all misses for word lengths up to $k_b = 18$ bytes, check lengths $2 \leqq a \leqq 12$ and byte lengths $2 \leqq b \leqq 10$ and $b = 12$. The maximum word length of 18 bytes results in a total of $2^{18} - 1 = 262143$ possible nonzero error magnitudes. In a few special cases ($b = 2, a = 4, 8$; $b = 3, a = 3, 6$; $b = 4, a = 4, 8$; $b = 5, a = 5$; $b = 6, a = 3, 6$) the maximum word length was extended to 20 bytes, i.e., $2^{20} - 1 = 1048575$ possible nonzero error magnitudes. The miss percentages were also tabulated for the various check factors α which detect all weight 2 or all weight 2 and 3 error magnitudes (Ref. 3). The word lengths used were n, with the requirement that $\alpha B = 2^n - 1$ should be satisfied.

The results of the tabulation show that for a and b relatively prime, the percentage of misses rapidly becomes $100/(2^a - 1)$ after the first miss which occurs at word length $n_1 = ab$ (the minimal case). For other pairs (a, b), the miss percentages beyond the word length n_1 tend to overshoot $100/(2^a - 1)$ and then go below $100/(2^a - 1)$ with increasing word length. The weight 2 and weight 2, 3 detecting choices of α display miss percentages which are comparable to those of relatively prime (a, b).

The complete results obtained in the tabulation of miss percentages are presented in the Appendix. Although elaborate combinatorial expressions can be derived for miss counts of any pair (a, b), the computer tabulation was preferred as being more convenient. Complete agreement between the analytic expressions and the computer tabulation was observed for miss percentages in the case $b = a$. All tabulated values of n_1 were also in agreement with the analytic result.

F. Detection of Repeated-Use Indeterminate Faults

A permanent indeterminate fault (used c times) will contribute to the error magnitude in one of 3^c possible ways. During each use the contribution will be 0, 2^i, or -2^i. In the regular pattern with $b = a$, $k = k_a = k_b$, and word length ka, the number of misses ϵ' due to the indeterminate fault (excluding the determinate subset) is given by the expression

$$\varepsilon' = \sum_{j}^{t} \varepsilon'_j$$

where t is the integer part of $k/2$, and

$$\varepsilon'_j = \frac{k!}{2\,[(k-2j)!]\,(j!)^2}$$

for $1 \leqq j \leqq t$. The total count of possible nonzero error magnitudes is $(3^k - 1)/2$. The miss percentage $200\varepsilon'/(3^k - 1)$ is highest for $k = 2$ and gradually decreases with increasing k. It is noted that for $k > 2^a - 1$ the determinate subset contributes the miss count ε, and the total number of misses is $\varepsilon + \varepsilon'$; furthermore, the value of ε' is independent of a. Table 1 lists the miss percentages (excluding the determinate subset) for byte counts $2 \leqq k \leqq 12$.

Given any pair (a, b), the first miss due to an indeterminate fault (excluding the determinate subset) occurs when the word length exceeds the least common multiple of a and b, that is, the first miss occurs for

$$n'_1 > c_a a$$

where $c_a a = c_b b$ is the least common multiple. Consequently, the maximum safe length n is attained for a and b relatively prime, with $n'_1 > ab$. In this case the first miss is due to the determinate subset and occurs for $n_1 = ab$. For other choices of the pair (a, b), $n'_1 < n_1$ is observed.

Table 1. Miss percentages

k	$\frac{(3^k - 1)}{2}$	ε'	Miss, %
2	4	1	25.00
3	13	3	23.08
4	40	9	22.50
5	121	25	20.66
6	364	70	19.23
7	1093	196	17.93
8	3280	553	16.86
9	9841	1569	15.94
10	29524	4476	15.16
11	88573	12826	14.48
12	265720	36894	13.88

The total miss percentages

$$\frac{200\,(\varepsilon + \varepsilon')}{3^{k_b} - 1}$$

are of interest in the cases $b \neq a$ as well. An exhaustive tabulation by means of a computer program was performed for word lengths up to $k_b = 12$ bytes; that is, $(3^{12} - 1)/2 = 265,\ 720$ nonzero error magnitudes were considered. The check lengths were again $2 \leqq a \leqq 12$, and the byte lengths were $2 \leqq b \leqq 10$ and $b = 12$. It was observed that for relatively prime pairs (a, b) the miss percentages were close to $100\ (2^a - 1)$, becoming greater for pairs with common divisors, and reaching the maximal values of Table 1 for $b = a$ and $b = c_a a$. Complete results of the tabulation are presented in the Appendix.

It is noted that the most favorable choices of pairs (a, b) in the determinate case are the least desirable choices for indeterminate faults, and vice versa. The choice of the most suitable values therefore depends on the relative frequencies of these two types of faults. More elaborate damage patterns (for instance, patterns due to multiple formation in multiplication and division) will be considered in further work.

REFERENCES

1. Avizienis, A., *A Set of Algorithms for a Diagnosable Arithmetic Unit*, Technical Report No. 32-546, Jet Propulsion Laboratory, Pasadena, California, March 1, 1964.
2. Avizienis, A., "Signed-Digit Number Representations for Fast Parallel Arithmetic," *IRE Transactions*, EC-10 (1961), pp. 389–400.

REFERENCES (Cont'd)

3. Peterson, W. W., *Error Correcting Codes*, The Massachusetts Institute of Technology Press and John Wiley and Sons, Inc., New York, 1961, pp. 236–244.
4. Diamond, J. M., "Checking Codes for Digital Computers," *Proceedings of the IRE*, Vol. 43, 1955, pp. 487–488.
5. Brown, D. T., "Error Detecting and Correcting Codes for Arithmetic Operations," *IRE Transactions*, EC-9 (1960), pp. 333–337.
6. Bernstein, A. J., and Kim, W. H., "Linear Codes for Single Error Correction in Symmetric and Asymmetric Computational Processes," *IRE Transactions*, IT-8 (1962), pp. 29–34.
7. Henderson, D. S., *Residue Class Error Checking Codes*, Preprints of papers presented at the 16th National Meeting of the Association for Computing Machinery, Los Angeles, 1961.
8. Chien, R. T., *Linear Codes for Burst-Error Correction in Binary Arithmetic and Transmission*, Research Paper RC-817, IBM Corporation, T. J. Watson Research Center, November 8, 1962.
9. Rothstein, J., "Residues of Binary Numbers Modulo Three," *IRE Transactions*, EC-8 (1959), p. 229.
10. Germeroth, J. H., "Casting Out Threes in Binary Numbers," *IRE Transactions*, EC-9 (1960), p. 373.

APPENDIX

Tabulation of Misses and Miss Percentages for Faults in Repeated-Use Algorithms

I. Determinate Faults, Check Factors 2^4-1.

Table A-1. Byte length $B = 2$

N = KB	ERROR MAGNITUDES		B = 2 A = 2				B = 2 A = 3				B = 2 A = 4			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	7	4	1	14.29	1	25.00	1	14.29	1	25.00	0	.00	0	.00
8	15	8	4	26.67	3	37.50	2	13.33	1	12.50	0	.00	0	.00
10	31	16	10	32.26	6	37.50	4	12.90	2	12.50	0	.00	0	.00
12	63	32	21	33.33	11	34.38	9	14.29	5	15.63	1	1.59	1	3.13
14	127	64	42	33.07	21	32.81	18	14.17	9	14.06	4	3.15	3	4.69
16	255	128	84	32.94	42	32.81	36	14.12	18	14.06	16	6.27	12	9.38
18	511	256	169	33.07	85	33.20	73	14.29	37	14.45	40	7.83	24	9.38
20	1023	512	340	33.24	171	33.40	146	14.27	73	14.26	100	9.78	60	11.72
22	2047	1024	682	33.32	342	33.40	292	14.26	146	14.26	200	9.77	100	9.77
24	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	401	9.79	201	9.81
26	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	722	8.81	321	7.84
28	16383	8192	5460	33.33	2730	33.33	2340	14.28	1170	14.28	1316	8.03	594	7.25
30	32767	16384	10921	33.33	5461	33.33	4681	14.29	2341	14.29	2352	7.18	1036	6.32
32	65535	32768	21844	33.33	10923	33.33	9362	14.29	4681	14.29	4368	6.67	2016	6.15
34	131071	65536	43690	33.33	21846	33.33	18724	14.29	9362	14.29	8352	6.37	3984	6.08
36	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	16705	6.37	8353	6.37
38	524287	262144									33796	6.45	17091	6.52
40	1048575	524288									69904	6.67	36108	6.89

N = KB	ERROR MAGNITUDES		B = 2 A = 5				B = 2 A = 6				B = 2 A = 7			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
12	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
14	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
16	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
18	511	256	16	3.13	8	3.13	1	.20	1	.39	4	.78	2	.78
20	1023	512	33	3.23	17	3.32	4	.39	3	.59	8	.78	4	.78
22	2047	1024	66	3.22	33	3.22	16	.78	12	1.17	16	.78	8	.78
24	4095	2048	132	3.22	66	3.22	64	1.56	48	2.34	32	.78	16	.78
26	8191	4096	264	3.22	132	3.22	160	1.95	96	2.34	64	.78	32	.78
28	16383	8192	528	3.22	264	3.22	400	2.44	240	2.93	129	.79	65	.79
30	32767	16384	1057	3.23	529	3.23	1000	3.05	600	3.66	258	.79	129	.79
32	65535	32768	2114	3.23	1057	3.23	2000	3.05	1000	3.05	516	.79	258	.79
34	131071	65536	4228	3.23	2114	3.23	4000	3.05	2000	3.05	1032	.79	516	.79
36	262143	131072	8456	3.23	4228	3.23	8001	3.05	4001	3.05	2064	.79	1032	.79
38	524287	262144												
40	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 2 A = 8				B = 2 A = 9				B = 2 A = 10			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	511	256	0	.00	0	.00	1	.20	1	.39	0	.00	0	.00
20	1023	512	0	.00	0	.00	2	.20	1	.20	0	.00	0	.00
22	2047	1024	0	.00	0	.00	4	.20	2	.20	0	.00	0	.00
24	4095	2048	1	.02	1	.05	8	.20	4	.20	0	.00	0	.00
26	8191	4096	4	.05	3	.07	16	.20	8	.20	0	.00	0	.00
28	16383	8192	16	.10	12	.15	32	.20	16	.20	0	.00	0	.00
30	32767	16384	64	.20	48	.29	64	.20	32	.20	1	.00	1	.01
32	65535	32768	256	.39	192	.59	128	.20	64	.20	4	.01	3	.01
34	131071	65536	640	.49	384	.59	256	.20	128	.20	16	.01	12	.02
36	262143	131072	1600	.61	960	.73	513	.20	257	.20	64	.02	48	.04
38	524287	262144	4000	.76	2400	.92								
40	1048575	524288	10000	.95	6000	1.14								

N = KB	ERROR MAGNITUDES		B = 2 A = 11				B = 2 A = 12			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00
4	3	2	0	.00	0	.00	0	.00	0	.00
6	7	4	0	.00	0	.00	0	.00	0	.00
8	15	8	0	.00	0	.00	0	.00	0	.00
10	31	16	0	.00	0	.00	0	.00	0	.00
12	63	32	0	.00	0	.00	0	.00	0	.00
14	127	64	0	.00	0	.00	0	.00	0	.00
16	255	128	0	.00	0	.00	0	.00	0	.00
18	511	256	0	.00	0	.00	0	.00	0	.00
20	1023	512	0	.00	0	.00	0	.00	0	.00
22	2047	1024	1	.05	1	.10	0	.00	0	.00
24	4095	2048	2	.05	1	.05	0	.00	0	.00
26	8191	4096	4	.05	2	.05	0	.00	0	.00
28	16383	8192	8	.05	4	.05	0	.00	0	.00
30	32767	16384	16	.05	8	.05	0	.00	0	.00
32	65535	32768	32	.05	16	.05	0	.00	0	.00
34	131071	65536	64	.05	32	.05	0	.00	0	.00
36	262143	131072	128	.05	64	.05	1	.00	1	.00
38	524287	262144								
40	1048575	524288								

Table A-2. Byte length $B = 3$

N = KB	ERROR MAGNITUDES		B = 3 A = 2				B = 3 A = 3				B = 3 A = 4			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	3	2	1	33.33	1	50.00	0	.00	0	.00	0	.00	0	.00
9	7	4	2	28.57	1	25.00	0	.00	0	.00	0	.00	0	.00
12	15	8	5	33.33	3	37.50	0	.00	0	.00	1	6.67	1	12.50
15	31	16	10	32.26	5	31.25	0	.00	0	.00	2	6.45	1	6.25
18	63	32	21	33.33	11	34.38	0	.00	0	.00	4	6.35	2	6.25
21	127	64	42	33.07	21	32.81	1	.79	1	1.56	8	6.30	4	6.25
24	255	128	85	33.33	43	33.59	8	3.14	7	5.47	17	6.67	9	7.03
27	511	256	170	33.27	85	33.20	36	7.05	28	10.94	34	6.65	17	6.64
30	1023	512	341	33.33	171	33.40	120	11.73	84	16.41	68	6.65	34	6.64
33	2047	1024	682	33.32	341	33.30	330	16.12	210	20.51	136	6.64	68	6.64
36	4095	2048	1365	33.33	683	33.35	792	19.34	462	22.56	273	6.67	137	6.69
39	8191	4096	2730	33.33	1365	33.33	1716	20.95	924	22.56	546	6.67	273	6.67
42	16383	8192	5461	33.33	2731	33.34	3433	20.95	1717	20.96	1092	6.67	546	6.67
45	32767	16384	10922	33.33	5461	33.33	6450	19.68	3017	18.41	2184	6.67	1092	6.67
48	65535	32768	21845	33.33	10923	33.33	11560	17.64	5110	15.59	4369	6.67	2185	6.67
51	131071	65536	43690	33.33	21845	33.33	20128	15.36	8568	13.07	8738	6.67	4369	6.67
54	262143	131072	87381	33.33	43691	33.33	34884	13.31	14756	11.26	17476	6.67	8738	6.67
57	524287	262144					62016	11.83	27132	10.35				
60	1048575	524288					116280	11.09	54264	10.35				

N = KB	ERROR MAGNITUDES		B = 3 A = 5				B = 3 A = 6				B = 3 A = 7			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
9	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
15	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
18	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
21	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
24	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
27	511	256	16	3.13	8	3.13	0	.00	0	.00	4	.78	2	.78
30	1023	512	33	3.23	17	3.32	0	.00	0	.00	8	.78	4	.78
33	2047	1024	66	3.22	33	3.22	0	.00	0	.00	16	.78	8	.78
36	4095	2048	132	3.22	66	3.22	0	.00	0	.00	32	.78	16	.78
39	8191	4096	264	3.22	132	3.22	0	.00	0	.00	64	.78	32	.78
42	16383	8192	528	3.22	264	3.22	1	.01	1	.01	129	.79	65	.79
45	32767	16384	1057	3.23	529	3.23	8	.02	7	.04	258	.79	129	.79
48	65535	32768	2114	3.23	1057	3.23	64	.10	56	.17	516	.79	258	.79
51	131071	65536	4228	3.23	2114	3.23	288	.22	224	.34	1032	.79	516	.79
54	262143	131072	8456	3.23	4228	3.23	1296	.49	1008	.77	2064	.79	1032	.79
57	524287	262144					4320	.82	3024	1.15				
60	1048575	524288					14400	1.37	10080	1.92				

N = KB	ERROR MAGNITUDES		B = 3 A = 8				B = 3 A = 9				B = 3 A = 10			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
9	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
15	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
21	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	255	128	1	.39	1	.78	0	.00	0	.00	0	.00	0	.00
27	511	256	2	.39	1	.39	0	.00	0	.00	0	.00	0	.00
30	1023	512	4	.39	2	.39	0	.00	0	.00	1	.10	1	.20
33	2047	1024	8	.39	4	.39	0	.00	0	.00	2	.10	1	.10
36	4095	2048	16	.39	8	.39	0	.00	0	.00	4	.10	2	.10
39	8191	4096	32	.39	16	.39	0	.00	0	.00	8	.10	4	.10
42	16383	8192	64	.39	32	.39	0	.00	0	.00	16	.10	8	.10
45	32767	16384	128	.39	64	.39	0	.00	0	.00	32	.10	16	.10
48	65535	32768	257	.39	129	.39	0	.00	0	.00	64	.10	32	.10
51	131071	65536	514	.39	257	.39	0	.00	0	.00	128	.10	64	.10
54	262143	131072	1028	.39	514	.39	0	.00	0	.00	256	.10	128	.10
57	524287	262144												
60	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 3 A = 11				B = 3 A = 12			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00
6	3	2	0	.00	0	.00	0	.00	0	.00
9	7	4	0	.00	0	.00	0	.00	0	.00
12	15	8	0	.00	0	.00	0	.00	0	.00
15	31	16	0	.00	0	.00	0	.00	0	.00
18	63	32	0	.00	0	.00	0	.00	0	.00
21	127	64	0	.00	0	.00	0	.00	0	.00
24	255	128	0	.00	0	.00	0	.00	0	.00
27	511	256	0	.00	0	.00	0	.00	0	.00
30	1023	512	0	.00	0	.00	0	.00	0	.00
33	2047	1024	1	.05	1	.10	0	.00	0	.00
36	4095	2048	2	.05	1	.05	0	.00	0	.00
39	8191	4096	4	.05	2	.05	0	.00	0	.00
42	16383	8192	8	.05	4	.05	0	.00	0	.00
45	32767	16384	16	.05	8	.05	0	.00	0	.00
48	65535	32768	32	.05	16	.05	0	.00	0	.00
51	131071	65536	64	.05	32	.05	0	.00	0	.00
54	262143	131072	128	.05	64	.05	0	.00	0	.00
57	524287	262144								
60	1048575	524288								

Table A-3. Byte length $B = 4$

N = KB	ERROR MAGNITUDES		B = 4 A = 2				B = 4 A = 3				B = 4 A = 4			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	7	4	1	14.29	1	25.00	1	14.29	1	25.00	0	.00	0	.00
16	15	8	4	26.67	3	37.50	2	13.33	1	12.50	0	.00	0	.00
20	31	16	10	32.26	6	37.50	4	12.90	2	12.50	0	.00	0	.00
24	63	32	21	33.33	11	34.38	9	14.29	5	15.63	0	.00	0	.00
28	127	64	42	33.07	21	32.81	18	14.17	9	14.06	0	.00	0	.00
32	255	128	84	32.94	42	32.81	36	14.12	18	14.06	0	.00	0	.00
36	511	256	169	33.07	85	33.20	73	14.29	37	14.45	0	.00	0	.00
40	1023	512	340	33.24	171	33.40	146	14.27	73	14.26	0	.00	0	.00
44	2047	1024	682	33.32	342	33.40	292	14.26	146	14.26	0	.00	0	.00
48	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	0	.00	0	.00
52	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	0	.00	0	.00
56	16383	8192	5460	33.33	2730	33.33	2340	14.28	1170	14.28	0	.00	0	.00
60	32767	16384	10921	33.33	5461	33.33	4681	14.29	2341	14.29	1	.00	1	.01
64	65535	32768	21844	33.33	10923	33.33	9362	14.29	4681	14.29	16	.02	15	.05
68	131071	65536	43690	33.33	21846	33.33	18724	14.29	9362	14.29	136	.10	120	.18
72	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	816	.31	680	.52
76	524287	262144									3876	.74	3060	1.17
80	1048575	524288									15504	1.48	11628	2.22

N = KB	ERROR MAGNITUDES		B = 4 A = 5				B = 4 A = 6				B = 4 A = 7			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
24	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
28	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
32	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
36	511	256	16	3.13	8	3.13	1	.20	1	.39	4	.78	2	.78
40	1023	512	33	3.23	17	3.32	4	.39	3	.59	8	.78	4	.78
44	2047	1024	66	3.22	33	3.22	16	.78	12	1.17	16	.78	8	.78
48	4095	2048	132	3.22	66	3.22	64	1.56	48	2.34	32	.78	16	.78
52	8191	4096	264	3.22	132	3.22	160	1.95	96	2.34	64	.78	32	.78
56	16383	8192	528	3.22	264	3.22	400	2.44	240	2.93	129	.79	65	.79
60	32767	16384	1057	3.23	529	3.23	1000	3.05	600	3.66	258	.79	129	.79
64	65535	32768	2114	3.23	1057	3.23	2000	3.05	1000	3.05	516	.79	258	.79
68	131071	65536	4228	3.23	2114	3.23	4000	3.05	2000	3.05	1032	.79	516	.79
72	262143	131072	8456	3.23	4228	3.23	8001	3.05	4001	3.05	2064	.79	1032	.79
76	524287	262144												
80	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 4 A = 8				B = 4 A = 9				B = 4 A = 10			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
28	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
32	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	511	256	0	.00	0	.00	1	.20	1	.39	0	.00	0	.00
40	1023	512	0	.00	0	.00	2	.20	1	.20	0	.00	0	.00
44	2047	1024	0	.00	0	.00	4	.20	2	.20	0	.00	0	.00
48	4095	2048	0	.00	0	.00	8	.20	4	.20	0	.00	0	.00
52	8191	4096	0	.00	0	.00	16	.20	8	.20	0	.00	0	.00
56	16383	8192	0	.00	0	.00	32	.20	16	.20	0	.00	0	.00
60	32767	16384	0	.00	0	.00	64	.20	32	.20	1	.00	1	.01
64	65535	32768	0	.00	0	.00	128	.20	64	.20	4	.01	3	.01
68	131071	65536	0	.00	0	.00	256	.20	128	.20	16	.01	12	.02
72	262143	131072	0	.00	0	.00	513	.20	257	.20	64	.02	48	.04
76	524287	262144	0	.00	0	.00								
80	1048575	524288	0	.00	0	.00								

N = KB	ERROR MAGNITUDES		B = 4 A = 11				B = 4 A = 12			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00
8	3	2	0	.00	0	.00	0	.00	0	.00
12	7	4	0	.00	0	.00	0	.00	0	.00
16	15	8	0	.00	0	.00	0	.00	0	.00
20	31	16	0	.00	0	.00	0	.00	0	.00
24	63	32	0	.00	0	.00	0	.00	0	.00
28	127	64	0	.00	0	.00	0	.00	0	.00
32	255	128	0	.00	0	.00	0	.00	0	.00
36	511	256	0	.00	0	.00	0	.00	0	.00
40	1023	512	0	.00	0	.00	0	.00	0	.00
44	2047	1024	1	.05	1	.10	0	.00	0	.00
48	4095	2048	2	.05	1	.05	0	.00	0	.00
52	8191	4096	4	.05	2	.05	0	.00	0	.00
56	16383	8192	8	.05	4	.05	0	.00	0	.00
60	32767	16384	16	.05	8	.05	0	.00	0	.00
64	65535	32768	32	.05	16	.05	0	.00	0	.00
68	131071	65536	64	.05	32	.05	0	.00	0	.00
72	262143	131072	128	.05	64	.05	0	.00	0	.00
76	524287	262144								
80	1048575	524288								

Table A-4. Byte length $B = 5$

N = KB	ERROR MAGNITUDES		B = 5 A = 2				B = 5 A = 3				B = 5 A = 4			
	TOTAL	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	3	2	1	33.33	1	50.00	0	.00	0	.00	0	.00	0	.00
15	7	4	2	28.57	1	25.00	1	14.29	1	25.00	0	.00	0	.00
20	15	8	5	33.33	3	37.50	2	13.33	1	12.50	1	6.67	1	12.50
25	31	16	10	32.26	5	31.25	4	12.90	2	12.50	2	6.45	1	6.25
30	63	32	21	33.33	11	34.38	9	14.29	5	15.63	4	6.35	2	6.25
35	127	64	42	33.07	21	32.81	18	14.17	9	14.06	8	6.30	4	6.25
40	255	128	85	33.33	43	33.59	36	14.12	18	14.06	17	6.67	9	7.03
45	511	256	170	33.27	85	33.20	73	14.29	37	14.45	34	6.65	17	6.64
50	1023	512	341	33.33	171	33.40	146	14.27	73	14.26	68	6.65	34	6.64
55	2047	1024	682	33.32	341	33.30	292	14.26	146	14.26	136	6.64	68	6.64
60	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	273	6.67	137	6.69
65	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	546	6.67	273	6.67
70	16383	8192	5461	33.33	2731	33.34	2340	14.28	1170	14.28	1092	6.67	546	6.67
75	32767	16384	10922	33.33	5461	33.33	4681	14.29	2341	14.29	2184	6.67	1092	6.67
80	65535	32768	21845	33.33	10923	33.33	9362	14.29	4681	14.29	4369	6.67	2185	6.67
85	131071	65536	43690	33.33	21845	33.33	18724	14.29	9362	14.29	8738	6.67	4369	6.67
90	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	17476	6.67	8738	6.67
95	524287	262144												
100	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 5 A = 5				B = 5 A = 6				B = 5 A = 7			
	TOTAL	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
15	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
25	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	63	32	0	.00	0	.00	1	1.59	1	3.13	0	.00	0	.00
35	127	64	0	.00	0	.00	2	1.57	1	1.56	1	.79	1	1.56
40	255	128	0	.00	0	.00	4	1.57	2	1.56	2	.78	1	.78
45	511	256	0	.00	0	.00	8	1.57	4	1.56	4	.78	2	.78
50	1023	512	0	.00	0	.00	16	1.56	8	1.56	8	.78	4	.78
55	2047	1024	0	.00	0	.00	32	1.56	16	1.56	16	.78	8	.78
60	4095	2048	0	.00	0	.00	65	1.59	33	1.61	32	.78	16	.78
65	8191	4096	0	.00	0	.00	130	1.59	65	1.59	64	.78	32	.78
70	16383	8192	0	.00	0	.00	260	1.59	130	1.59	129	.79	65	.79
75	32767	16384	0	.00	0	.00	520	1.59	260	1.59	258	.79	129	.79
80	65535	32768	0	.00	0	.00	1040	1.59	520	1.59	516	.79	258	.79
85	131071	65536	0	.00	0	.00	2080	1.59	1040	1.59	1032	.79	516	.79
90	262143	131072	0	.00	0	.00	4161	1.59	2081	1.59	2064	.79	1032	.79
95	524287	262144	0	.00	0	.00								
100	1048575	524288	0	.00	0	.00								

N = KB	ERROR MAGNITUDES		B = 5 A = 8				B = 5 A = 9				B = 5 A = 10			
	TOTAL	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TUTAL	MISS	INCREMENT
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
15	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
25	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
35	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
40	255	128	1	.39	1	.78	0	.00	0	.00	0	.00	0	.00
45	511	256	2	.39	1	.39	1	.20	1	.39	0	.00	0	.00
50	1023	512	4	.39	2	.39	2	.20	1	.20	0	.00	0	.00
55	2047	1024	8	.39	4	.39	4	.20	2	.20	0	.00	0	.00
60	4095	2048	16	.39	8	.39	8	.20	4	.20	0	.00	0	.00
65	8191	4096	32	.39	16	.39	16	.20	8	.20	0	.00	0	.00
70	16383	8192	64	.39	32	.39	32	.20	16	.20	0	.00	0	.00
75	32767	16384	128	.39	64	.39	64	.20	32	.20	0	.00	0	.00
80	65535	32768	257	.39	129	.39	128	.20	64	.20	0	.00	0	.00
85	131071	65536	514	.39	257	.39	256	.20	128	.20	0	.00	0	.00
90	262143	131072	1028	.39	514	.39	513	.20	257	.20	0	.00	0	.00
95	524287	262144												
100	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 5 A = 11				B = 5 A = 12			
	TOTAL	INCREMENT	MISS	TOTAL	MISS	INCREMENT	MISS	TOTAL	MISS	INCREMENT
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00
10	3	2	C	.00	0	.00	0	.00	0	.00
15	7	4	0	.00	0	.00	0	.00	0	.00
20	15	8	0	.00	0	.00	0	.00	0	.00
25	31	16	0	.00	0	.00	0	.00	0	.00
3C	63	32	0	.00	0	.00	0	.00	0	.00
35	127	64	0	.00	0	.00	0	.00	0	.00
40	255	128	0	.00	0	.00	0	.00	0	.00
45	511	256	0	.00	0	.00	0	.00	0	.00
50	1023	512	0	.00	0	.00	0	.00	0	.00
55	2047	1024	1	.05	1	.10	0	.00	0	.00
60	4095	2048	2	.05	1	.05	1	.02	1	.05
65	8191	4096	4	.05	2	.05	2	.02	1	.02
70	16383	8192	8	.05	4	.05	4	.02	2	.02
75	32767	16384	16	.05	8	.05	8	.02	4	.02
80	65535	32768	32	.05	16	.05	16	.02	8	.02
85	131071	65536	64	.05	32	.05	32	.02	16	.02
90	262143	131072	128	.05	64	.05	64	.02	32	.02
95	524287	262144								
100	1048575	524288								

Table A-5. Byte length $B = 6$

N = KB	ERROR MAGNITUDES		B = 6 A = 2				B = 6 A = 3				B = 6 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	7	4	1	14.29	1	25.00	0	.00	0	.00	0	.00	0	.00
24	15	8	4	26.67	3	37.50	0	.00	0	.00	0	.00	0	.00
30	31	16	10	32.26	6	37.50	0	.00	0	.00	0	.00	0	.00
36	63	32	21	33.33	11	34.38	0	.00	0	.00	1	1.59	1	3.13
42	127	64	42	33.07	21	32.81	1	.79	1	1.56	4	3.15	3	4.69
48	255	128	84	32.94	42	32.81	8	3.14	7	5.47	16	6.27	12	9.38
54	511	256	169	33.07	85	33.20	36	7.05	28	10.94	40	7.83	24	9.38
60	1023	512	340	33.24	171	33.40	120	11.73	84	16.41	100	9.78	60	11.72
66	2047	1024	682	33.32	342	33.40	330	16.12	210	20.51	200	9.77	100	9.77
72	4095	2048	1365	33.33	683	33.35	792	19.34	462	22.56	401	9.79	201	9.81
78	8191	4096	2730	33.33	1365	33.33	1716	20.95	924	22.56	722	8.81	321	7.84
84	16383	8192	5460	33.33	2730	33.33	3433	20.95	1717	20.96	1316	8.03	594	7.25
90	32767	16384	10921	33.33	5461	33.33	6450	19.68	3017	18.41	2352	7.18	1036	6.32
96	65535	32768	21844	33.33	10923	33.33	11560	17.64	5110	15.59	4368	6.67	2016	6.15
102	131071	65536	43690	33.33	21846	33.33	20128	15.36	8568	13.07	8352	6.37	3984	6.08
108	262143	131072	87381	33.33	43691	33.33	34884	13.31	14756	11.26	16705	6.37	8353	6.37
114	524287	262144					62016	11.83	27132	10.35				
120	1048575	524288					116280	11.09	54264	10.35				

N = KB	ERROR MAGNITUDES		B = 6 A = 5				B = 6 A = 6				B = 6 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
36	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
42	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
48	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
54	511	256	16	3.13	8	3.13	0	.00	0	.00	4	.78	2	.78
60	1023	512	33	3.23	17	3.32	0	.00	0	.00	8	.78	4	.78
66	2047	1024	66	3.22	33	3.22	0	.00	0	.00	16	.78	8	.78
72	4095	2048	132	3.22	66	3.22	0	.00	0	.00	32	.78	16	.78
78	8191	4096	264	3.22	132	3.22	0	.00	0	.00	64	.78	32	.78
84	16383	8192	528	3.22	264	3.22	0	.00	0	.00	129	.79	65	.79
90	32767	16384	1057	3.23	529	3.23	0	.00	0	.00	258	.79	129	.79
96	65535	32768	2114	3.23	1057	3.23	0	.00	0	.00	516	.79	258	.79
102	131071	65536	4228	3.23	2114	3.23	0	.00	0	.00	1032	.79	516	.79
108	262143	131072	8456	3.23	4228	3.23	0	.00	0	.00	2064	.79	1032	.79
114	524287	262144					0	.00	0	.00				
120	1048575	524288					0	.00	0	.00				

N = KB	ERROR MAGNITUDES		B = 6 A = 8				B = 6 A = 9				B = 6 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
42	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
48	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
54	511	256	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
60	1023	512	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
66	2047	1024	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
72	4095	2048	1	.02	1	.05	0	.00	0	.00	0	.00	0	.00
78	8191	4096	4	.05	3	.07	0	.00	0	.00	0	.00	0	.00
84	16383	8192	16	.10	12	.15	0	.00	0	.00	0	.00	0	.00
90	32767	16384	64	.20	48	.29	0	.00	0	.00	1	.00	1	.01
96	65535	32768	256	.39	192	.59	0	.00	0	.00	4	.01	3	.01
102	131071	65536	640	.49	384	.59	0	.00	0	.00	16	.01	12	.02
108	262143	131072	1600	.61	960	.73	0	.00	0	.00	64	.02	48	.04
114	524287	262144												
120	1048575	524288												

N = KB	ERROR MAGNITUDES		B = 6 A = 11				B = 6 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00
12	3	2	0	.00	0	.00	0	.00	0	.00
18	7	4	0	.00	0	.00	0	.00	0	.00
24	15	8	0	.00	0	.00	0	.00	0	.00
30	31	16	0	.00	0	.00	0	.00	0	.00
36	63	32	0	.00	0	.00	0	.00	0	.00
42	127	64	0	.00	0	.00	0	.00	0	.00
48	255	128	0	.00	0	.00	0	.00	0	.00
54	511	256	0	.00	0	.00	0	.00	0	.00
60	1023	512	0	.00	0	.00	0	.00	0	.00
66	2047	1024	1	.05	1	.10	0	.00	0	.00
72	4095	2048	2	.05	1	.05	0	.00	0	.00
78	8191	4096	4	.05	2	.05	0	.00	0	.00
84	16383	8192	8	.05	4	.05	0	.00	0	.00
90	32767	16384	16	.05	8	.05	0	.00	0	.00
96	65535	32768	32	.05	16	.05	0	.00	0	.00
102	131071	65536	64	.05	32	.05	0	.00	0	.00
108	262143	131072	128	.05	64	.05	0	.00	0	.00
114	524287	262144								
120	1048575	524288								

Table A-6. Byte length $B = 7$

N = KB	ERROR MAGNITUDES		B = 7 A = 2				B = 7 A = 3				B = 7 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	3	2	1	33.33	1	50.00	0	.00	0	.00	0	.00	0	.00
21	7	4	2	28.57	1	25.00	1	14.29	1	25.00	0	.00	0	.00
28	15	8	5	33.33	3	37.50	2	13.33	1	12.50	1	6.67	1	12.50
35	31	16	10	32.26	5	31.25	4	12.90	2	12.50	2	6.45	1	6.25
42	63	32	21	33.33	11	34.38	9	14.29	5	15.63	4	6.35	2	6.25
49	127	64	42	33.07	21	32.81	18	14.17	9	14.06	8	6.30	4	6.25
56	255	128	85	33.33	43	33.59	36	14.12	18	14.06	17	6.67	9	7.03
63	511	256	170	33.27	85	33.20	73	14.29	37	14.45	34	6.65	17	6.64
70	1023	512	341	33.33	171	33.40	146	14.27	73	14.26	68	6.65	34	6.64
77	2047	1024	682	33.32	341	33.30	292	14.26	146	14.26	136	6.64	68	6.64
84	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	273	6.67	137	6.69
91	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	546	6.67	273	6.67
98	16383	8192	5461	33.33	2731	33.34	2340	14.28	1170	14.28	1092	6.67	546	6.67
105	32767	16384	10922	33.33	5461	33.33	4681	14.29	2341	14.29	2184	6.67	1092	6.67
112	65535	32768	21845	33.33	10923	33.33	9362	14.29	4681	14.29	4369	6.67	2185	6.67
119	131071	65536	43690	33.33	21845	33.33	18724	14.29	9362	14.29	8738	6.67	4369	6.67
126	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	17476	6.67	8738	6.67

N = KB	ERROR MAGNITUDES		B = 7 A = 5				B = 7 A = 6				B = 7 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
21	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
28	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
35	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
42	63	32	2	3.17	1	3.13	1	1.59	1	3.13	0	.00	0	.00
49	127	64	4	3.15	2	3.13	2	1.57	1	1.56	0	.00	0	.00
56	255	128	8	3.14	4	3.13	4	1.57	2	1.56	0	.00	0	.00
63	511	256	16	3.13	8	3.13	8	1.57	4	1.56	0	.00	0	.00
70	1023	512	33	3.23	17	3.32	16	1.56	8	1.56	0	.00	0	.00
77	2047	1024	66	3.22	33	3.22	32	1.56	16	1.56	0	.00	0	.00
84	4095	2048	132	3.22	66	3.22	65	1.59	33	1.61	0	.00	0	.00
91	8191	4096	264	3.22	132	3.22	130	1.59	65	1.59	0	.00	0	.00
98	16383	8192	528	3.22	264	3.22	260	1.59	130	1.59	0	.00	0	.00
105	32767	16384	1057	3.23	529	3.23	520	1.59	260	1.59	0	.00	0	.00
112	65535	32768	2114	3.23	1057	3.23	1040	1.59	520	1.59	0	.00	0	.00
119	131071	65536	4228	3.23	2114	3.23	2080	1.59	1040	1.59	0	.00	0	.00
126	262143	131072	8456	3.23	4228	3.23	4161	1.59	2081	1.59	0	.00	0	.00

N = KB	ERROR MAGNITUDES		B = 7 A = 8				B = 7 A = 9				B = 7 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
21	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
28	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
35	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
42	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
49	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
56	255	128	1	.39	1	.78	0	.00	0	.00	0	.00	0	.00
63	511	256	2	.39	1	.39	1	.20	1	.39	0	.00	0	.00
70	1023	512	4	.39	2	.39	2	.20	1	.20	1	.10	1	.20
77	2047	1024	8	.39	4	.39	4	.20	2	.20	2	.10	1	.10
84	4095	2048	16	.39	8	.39	8	.20	4	.20	4	.10	2	.10
91	8191	4096	32	.39	16	.39	16	.20	8	.20	8	.10	4	.10
98	16383	8192	64	.39	32	.39	32	.20	16	.20	16	.10	8	.10
105	32767	16384	128	.39	64	.39	64	.20	32	.20	32	.10	16	.10
112	65535	32768	257	.39	129	.39	128	.20	64	.20	64	.10	32	.10
119	131071	65536	514	.39	257	.39	256	.20	128	.20	128	.10	64	.10
126	262143	131072	1028	.39	514	.39	513	.20	257	.20	256	.10	128	.10

N = KB	ERROR MAGNITUDES		B = 7 A = 11				B = 7 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00
14	3	2	0	.00	0	.00	0	.00	0	.00
21	7	4	0	.00	0	.00	0	.00	0	.00
28	15	8	0	.00	0	.00	0	.00	0	.00
35	31	16	0	.00	0	.00	0	.00	0	.00
42	63	32	0	.00	0	.00	0	.00	0	.00
49	127	64	0	.00	0	.00	0	.00	0	.00
56	255	128	0	.00	0	.00	0	.00	0	.00
63	511	256	0	.00	0	.00	0	.00	0	.00
70	1023	512	0	.00	0	.00	0	.00	0	.00
77	2047	1024	1	.05	1	.10	0	.00	0	.00
84	4095	2048	2	.05	1	.05	1	.02	1	.05
91	8191	4096	4	.05	2	.05	2	.02	1	.02
98	16383	8192	8	.05	4	.05	4	.02	2	.02
105	32767	16384	16	.05	8	.05	8	.02	4	.02
112	65535	32768	32	.05	16	.05	16	.02	8	.02
119	131071	65536	64	.05	32	.05	32	.02	16	.02
126	262143	131072	128	.05	64	.05	64	.02	32	.02

Table A-7. Byte length $B = 8$

N = KB	ERROR MAGNITUDES		B = 8 A = 2				B = 8 A = 3				B = 8 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	7	4	1	14.29	1	25.00	1	14.29	1	25.00	0	.00	0	.00
32	15	8	4	26.67	3	37.50	2	13.33	1	12.50	0	.00	0	.00
40	31	16	10	32.26	6	37.50	4	12.90	2	12.50	0	.00	0	.00
48	63	32	21	33.33	11	34.38	9	14.29	5	15.63	0	.00	0	.00
56	127	64	42	33.07	21	32.81	18	14.17	9	14.06	0	.00	0	.00
64	255	128	84	32.94	42	32.81	36	14.12	18	14.06	0	.00	0	.00
72	511	256	169	33.07	85	33.20	73	14.29	37	14.45	0	.00	0	.00
80	1023	512	340	33.24	171	33.40	146	14.27	73	14.26	0	.00	0	.00
88	2047	1024	682	33.32	342	33.40	292	14.26	146	14.26	0	.00	0	.00
96	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	0	.00	0	.00
104	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	0	.00	0	.00
112	16383	8192	5460	33.33	2730	33.33	2340	14.28	1170	14.28	0	.00	0	.00
120	32767	16384	10921	33.33	5461	33.33	4681	14.29	2341	14.29	1	.00	1	.01
128	65535	32768	21844	33.33	10923	33.33	9362	14.29	4681	14.29	16	.02	15	.05
136	131071	65536	43690	33.33	21846	33.33	18724	14.29	9362	14.29	136	.10	120	.18
144	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	816	.31	680	.52

N = KB	ERROR MAGNITUDES		B = 8 A = 5				B = 8 A = 6				B = 8 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
32	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
40	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
48	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
56	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	.56
64	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
72	511	256	16	3.13	8	3.13	1	.20	1	.39	4	.78	2	.78
80	1023	512	33	3.23	17	3.32	4	.39	3	.59	8	.78	4	.78
88	2047	1024	66	3.22	33	3.22	16	.78	12	1.17	16	.78	8	.78
96	4095	2048	132	3.22	66	3.22	64	1.56	48	2.34	32	.78	16	.78
104	8191	4096	264	3.22	132	3.22	160	1.95	96	2.34	64	.78	32	.78
112	16383	8192	528	3.22	264	3.22	400	2.44	240	2.93	129	.79	65	.79
120	32767	16384	1057	3.23	529	3.23	1000	3.05	600	3.66	258	.79	129	.79
128	65535	32768	2114	3.23	1057	3.23	2000	3.05	1000	3.05	516	.79	258	.79
136	131071	65536	4228	3.23	2114	3.23	4000	3.05	2000	3.05	1032	.79	516	.79
144	262143	131072	8456	3.23	4228	3.23	8001	3.05	4001	3.05	2064	.79	1032	.79

N = KB	ERROR MAGNITUDES		B = 8 A = 8				B = 8 A = 9				B = 8 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
32	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
40	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
48	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
56	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
64	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
72	511	256	0	.00	0	.00	1	.20	1	.39	0	.00	0	.00
80	1023	512	0	.00	0	.00	2	.20	1	.20	0	.00	0	.00
88	2047	1024	0	.00	0	.00	4	.20	2	.20	0	.00	0	.00
96	4095	2048	0	.00	0	.00	8	.20	4	.20	0	.00	0	.00
104	8191	4096	0	.00	0	.00	16	.20	8	.20	0	.00	0	.00
112	16383	8192	0	.00	0	.00	32	.20	16	.20	0	.00	0	.00
120	32767	16384	0	.00	0	.00	64	.20	32	.20	1	.00	1	.01
128	65535	32768	0	.00	0	.00	128	.20	64	.20	4	.01	3	.01
136	131071	65536	0	.00	0	.00	256	.20	128	.20	16	.01	12	.02
144	262143	131072	0	.00	0	.00	513	.20	257	.20	64	.02	48	.04

N = KB	ERROR MAGNITUDES		B = 8 A = 11				B = 8 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT	MISS TOTAL COUNT	PERCENT	MISS INCREMENT COUNT	PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00
16	3	2	0	.00	0	.00	0	.00	0	.00
24	7	4	0	.00	0	.00	0	.00	0	.00
32	15	8	0	.00	0	.00	0	.00	0	.00
40	31	16	0	.00	0	.00	0	.00	0	.00
48	63	32	0	.00	0	.00	0	.00	0	.00
56	127	64	0	.00	0	.00	0	.00	0	.00
64	255	128	0	.00	0	.00	0	.00	0	.00
72	511	256	0	.00	0	.00	0	.00	0	.00
80	1023	512	0	.00	0	.00	0	.00	0	.00
88	2047	1024	1	.05	1	.10	0	.00	0	.00
96	4095	2048	2	.05	1	.05	0	.00	0	.00
104	8191	4096	4	.05	2	.05	0	.00	0	.00
112	16383	8192	8	.05	4	.05	0	.00	0	.00
120	32767	16384	16	.05	8	.05	0	.00	0	.00
128	65535	32768	32	.05	16	.05	0	.00	0	.00
136	131071	65536	64	.05	32	.05	0	.00	0	.00
144	262143	131072	128	.05	64	.05	0	.00	0	.00

Table A-8. Byte length $B = 9$

N = KB	ERROR MAGNITUDES		B = 9 A = 2				B = 9 A = 3				B = 9 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	3	2	1	33.33	1	50.00	0	.00	0	.00	0	.00	0	.00
27	7	4	2	28.57	1	25.00	0	.00	0	.00	0	.00	0	.00
36	15	8	5	33.33	3	37.50	0	.00	0	.00	1	6.67	1	12.50
45	31	16	10	32.26	5	31.25	0	.00	0	.00	2	6.45	1	6.25
54	63	32	21	33.33	11	34.38	0	.00	0	.00	4	6.35	2	6.25
63	127	64	42	33.07	21	32.81	1	.79	1	1.56	8	6.30	4	6.25
72	255	128	85	33.33	43	33.59	8	3.14	7	5.47	17	6.67	9	7.03
81	511	256	170	33.27	85	33.20	36	7.05	28	10.94	34	6.65	17	6.64
90	1023	512	341	33.33	171	33.40	120	11.73	84	16.41	68	6.65	34	6.64
99	2047	1024	682	33.32	341	33.30	330	16.12	210	20.51	136	6.64	68	6.64
108	4095	2048	1365	33.33	683	33.35	792	19.34	462	22.56	273	6.67	137	6.69
117	8191	4096	2730	33.33	1365	33.33	1716	20.95	924	22.56	546	6.67	273	6.67
126	16383	8192	5461	33.33	2731	33.34	3433	20.95	1717	20.96	1092	6.67	546	6.67
135	32767	16384	10922	33.33	5461	33.33	6450	19.68	3017	18.41	2184	6.67	1092	6.67
144	65535	32768	21845	33.33	10923	33.33	11560	17.64	5110	15.59	4369	6.67	2185	6.67
153	131071	65536	43690	33.33	21845	33.33	20128	15.36	8568	13.07	8738	6.67	4369	6.67
162	262143	131072	87381	33.33	43691	33.33	34884	13.31	14756	11.26	17476	6.67	8738	6.67

N = KB	ERROR MAGNITUDES		B = 9 A = 5				B = 9 A = 6				B = 9 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
27	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
45	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
54	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
63	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
72	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
81	511	256	16	3.13	8	3.13	0	.00	0	.00	4	.78	2	.78
90	1023	512	33	3.23	17	3.32	0	.00	0	.00	8	.78	4	.78
99	2047	1024	66	3.22	33	3.22	0	.00	0	.00	16	.78	8	.78
108	4095	2048	132	3.22	66	3.22	0	.00	0	.00	32	.78	16	.78
117	8191	4096	264	3.22	132	3.22	0	.00	0	.00	64	.78	32	.78
126	16383	8192	528	3.22	264	3.22	1	.01	1	.01	129	.79	65	.79
135	32767	16384	1057	3.23	529	3.23	8	.02	7	.04	258	.79	129	.79
144	65535	32768	2114	3.23	1057	3.23	64	.10	56	.17	516	.79	258	.79
153	131071	65536	4228	3.23	2114	3.23	288	.22	224	.34	1032	.79	516	.79
162	262143	131072	8456	3.23	4228	3.23	1296	.49	1008	.77	2064	.79	1032	.79

N = KB	ERROR MAGNITUDES		B = 9 A = 8				B = 9 A = 9				B = 9 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
27	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
45	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
54	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
63	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
72	255	128	1	.39	1	.78	0	.00	0	.00	0	.00	0	.00
81	511	256	2	.39	1	.39	0	.00	0	.00	0	.00	0	.00
90	1023	512	4	.39	2	.39	0	.00	0	.00	1	.10	1	.20
99	2047	1024	8	.39	4	.39	0	.00	0	.00	2	.10	1	.10
108	4095	2048	16	.39	8	.39	0	.00	0	.00	4	.10	2	.10
117	8191	4096	32	.39	16	.39	0	.00	0	.00	8	.10	4	.10
126	16383	8192	64	.39	32	.39	0	.00	0	.00	16	.10	8	.10
135	32767	16384	128	.39	64	.39	0	.00	0	.00	32	.10	16	.10
144	65535	32768	257	.39	129	.39	0	.00	0	.00	64	.10	32	.10
153	131071	65536	514	.39	257	.39	0	.00	0	.00	128	.10	64	.10
162	262143	131072	1028	.39	514	.39	0	.00	0	.00	256	.10	128	.10

N = KB	ERROR MAGNITUDES		B = 9 A = 11				B = 9 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00
18	3	2	0	.00	0	.00	0	.00	0	.00
27	7	4	0	.00	0	.00	0	.00	0	.00
36	15	8	0	.00	0	.00	0	.00	0	.00
45	31	16	0	.00	0	.00	0	.00	0	.00
54	63	32	0	.00	0	.00	0	.00	0	.00
63	127	64	0	.00	0	.00	0	.00	0	.00
72	255	128	0	.00	0	.00	0	.00	0	.00
81	511	256	0	.00	0	.00	0	.00	0	.00
90	1023	512	0	.00	0	.00	0	.00	0	.00
99	2047	1024	1	.05	1	.10	0	.00	0	.00
108	4095	2048	2	.05	1	.05	0	.00	0	.00
117	8191	4096	4	.05	2	.05	0	.00	0	.00
126	16383	8192	8	.05	4	.05	0	.00	0	.00
135	32767	16384	16	.05	8	.05	0	.00	0	.00
144	65535	32768	32	.05	16	.05	0	.00	0	.00
153	131071	65536	64	.05	32	.05	0	.00	0	.00
162	262143	131072	128	.05	64	.05	0	.00	0	.00

Table A-9. Byte length $B = 10$

N = KB	ERROR MAGNITUDES		B = 10 A = 2				B = 10 A = 3				B = 10 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	7	4	1	14.29	1	25.00	1	14.29	1	25.00	0	.00	0	.00
40	15	8	4	26.67	3	37.50	2	13.33	1	12.50	0	.00	0	.00
50	31	16	10	32.26	6	37.50	4	12.90	2	12.50	0	.00	0	.00
60	63	32	21	33.33	11	34.38	9	14.29	5	15.63	1	1.59	1	3.13
70	127	64	42	33.07	21	32.81	18	14.17	9	14.06	4	3.15	3	4.69
80	255	128	84	32.94	42	32.81	36	14.12	18	14.06	16	6.27	12	9.38
90	511	256	169	33.07	85	33.20	73	14.29	37	14.45	40	7.83	24	9.38
100	1023	512	340	33.24	171	33.40	146	14.27	73	14.26	100	9.78	60	11.72
110	2047	1024	682	33.32	342	33.40	292	14.26	146	14.26	200	9.77	100	9.77
120	4095	2048	1365	33.33	683	33.35	585	14.29	293	14.31	401	9.79	201	9.81
130	8191	4096	2730	33.33	1365	33.33	1170	14.28	585	14.28	722	8.81	321	7.84
140	16383	8192	5460	33.33	2730	33.33	2340	14.28	1170	14.28	1316	8.03	594	7.25
150	32767	16384	10921	33.33	5461	33.33	4681	14.29	2341	14.29	2352	7.18	1036	6.32
160	65535	32768	21844	33.33	10923	33.33	9362	14.29	4681	14.29	4368	6.67	2016	6.15
170	131071	65536	43690	33.33	21846	33.33	18724	14.29	9362	14.29	8352	6.37	3984	6.08
180	262143	131072	87381	33.33	43691	33.33	37449	14.29	18725	14.29	16705	6.37	8353	6.37

N = KB	ERROR MAGNITUDES		B = 10 A = 5				B = 10 A = 6				B = 10 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
40	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
50	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
60	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
70	127	64	0	.00	0	.00	0	.00	0	.00	1	.79	1	1.56
80	255	128	0	.00	0	.00	0	.00	0	.00	2	.78	1	.78
90	511	256	0	.00	0	.00	1	.20	1	.39	4	.78	2	.78
100	1023	512	0	.00	0	.00	4	.39	3	.59	8	.78	4	.78
110	2047	1024	0	.00	0	.00	16	.78	12	1.17	16	.78	8	.78
120	4095	2048	0	.00	0	.00	64	1.56	48	2.34	32	.78	16	.78
130	8191	4096	0	.00	0	.00	160	1.95	96	2.34	64	.78	32	.78
140	16383	8192	0	.00	0	.00	400	2.44	240	2.93	129	.79	65	.79
150	32767	16384	0	.00	0	.00	1000	3.05	600	3.66	258	.79	129	.79
160	65535	32768	0	.00	0	.00	2000	3.05	1000	3.05	516	.79	258	.79
170	131071	65536	0	.00	0	.00	4000	3.05	2000	3.05	1032	.79	516	.79
180	262143	131072	0	.00	0	.00	8001	3.05	4001	3.05	2064	.79	1032	.79

N = KB	ERROR MAGNITUDES		B = 10 A = 8				B = 10 A = 9				B = 10 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
30	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
40	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
50	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
60	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
70	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
80	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
90	511	256	0	.00	0	.00	1	.20	1	.39	0	.00	0	.00
100	1023	512	0	.00	0	.00	2	.20	1	.20	0	.00	0	.00
110	2047	1024	0	.00	0	.00	4	.20	2	.20	0	.00	0	.00
120	4095	2048	1	.02	1	.05	8	.20	4	.20	0	.00	0	.00
130	8191	4096	4	.05	3	.07	16	.20	8	.20	0	.00	0	.00
140	16383	8192	16	.10	12	.15	32	.20	16	.20	0	.00	0	.00
150	32767	16384	64	.20	48	.29	64	.20	32	.20	0	.00	0	.00
160	65535	32768	256	.39	192	.59	128	.20	64	.20	0	.00	0	.00
170	131071	65536	640	.49	384	.59	256	.20	128	.20	0	.00	0	.00
180	262143	131072	1600	.61	960	.73	513	.20	257	.20	0	.00	0	.00

N = KB	ERROR MAGNITUDES		B = 10 A = 11				B = 10 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00
20	3	2	0	.00	0	.00	0	.00	0	.00
30	7	4	0	.00	0	.00	0	.00	0	.00
40	15	8	0	.00	0	.00	0	.00	0	.00
50	31	16	0	.00	0	.00	0	.00	0	.00
60	63	32	0	.00	0	.00	0	.00	0	.00
70	127	64	0	.00	0	.00	0	.00	0	.00
80	255	128	0	.00	0	.00	0	.00	0	.00
90	511	256	0	.00	0	.00	0	.00	0	.00
100	1023	512	0	.00	0	.00	0	.00	0	.00
110	2047	1024	1	.05	1	.10	0	.00	0	.00
120	4095	2048	2	.05	1	.05	0	.00	0	.00
130	8191	4096	4	.05	2	.05	0	.00	0	.00
140	16383	8192	8	.05	4	.05	0	.00	0	.00
150	32767	16384	16	.05	8	.05	0	.00	0	.00
160	65535	32768	32	.05	16	.05	0	.00	0	.00
170	131071	65536	64	.05	32	.05	0	.00	0	.00
180	262143	131072	128	.05	64	.05	1	.00	1	.00

Table A-10. Byte length $B = 12$

N = KB	ERROR MAGNITUDES		B = 12 A = 2				B = 12 A = 3				B = 12 A = 4			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	7	4	1	14.29	1	25.00	0	.00	0	.00	0	.00	0	.00
48	15	8	4	26.67	3	37.50	0	.00	0	.00	0	.00	0	.00
60	31	16	10	32.26	6	37.50	0	.00	0	.00	0	.00	0	.00
72	63	32	21	33.33	11	34.38	0	.00	0	.00	0	.00	0	.00
84	127	64	42	33.07	21	32.81	1	.79	1	1.56	0	.00	0	.00
96	255	128	84	32.94	42	32.81	8	3.14	7	5.47	0	.00	0	.00
108	511	256	169	33.07	85	33.20	36	7.05	28	10.94	0	.00	0	.00
120	1023	512	340	33.24	171	33.40	120	11.73	84	16.41	0	.00	0	.00
132	2047	1024	682	33.32	342	33.40	330	16.12	210	20.51	0	.00	0	.00
144	4095	2048	1365	33.33	683	33.35	792	19.34	462	22.56	0	.00	0	.00
156	8191	4096	2730	33.33	1365	33.33	1716	20.95	924	22.56	0	.00	0	.00
168	16383	8192	5460	33.33	2730	33.33	3433	20.95	1717	20.96	0	.00	0	.00
180	32767	16384	10921	33.33	5461	33.33	6450	19.68	3017	18.41	1	.00	1	.01
192	65535	32768	21844	33.33	10923	33.33	11560	17.64	5110	15.59	16	.02	15	.05
204	131071	65536	43690	33.33	21846	33.33	20128	15.36	8568	13.07	136	.10	120	.18
216	262143	131072	87381	33.33	43691	33.33	34884	13.31	14756	11.26	816	.31	680	.52

N = KB	ERROR MAGNITUDES		B = 12 A = 5				B = 12 A = 6				B = 12 A = 7			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
48	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
60	31	16	1	3.23	1	6.25	0	.00	0	.00	0	.00	0	.00
72	63	32	2	3.17	1	3.13	0	.00	0	.00	0	.00	0	.00
84	127	64	4	3.15	2	3.13	0	.00	0	.00	1	.79	1	1.56
96	255	128	8	3.14	4	3.13	0	.00	0	.00	2	.78	1	.78
108	511	256	16	3.13	8	3.13	0	.00	0	.00	4	.78	2	.78
120	1023	512	33	3.23	17	3.32	0	.00	0	.00	8	.78	4	.78
132	2047	1024	66	3.22	33	3.22	0	.00	0	.00	16	.78	8	.78
144	4095	2048	132	3.22	66	3.22	0	.00	0	.00	32	.78	16	.78
156	8191	4096	264	3.22	132	3.22	0	.00	0	.00	64	.78	32	.78
168	16383	8192	528	3.22	264	3.22	0	.00	0	.00	129	.79	65	.79
180	32767	16384	1057	3.23	529	3.23	0	.00	0	.00	258	.79	129	.79
192	65535	32768	2114	3.23	1057	3.23	0	.00	0	.00	516	.79	258	.79
204	131071	65536	4228	3.23	2114	3.23	0	.00	0	.00	1032	.79	516	.79
216	262143	131072	8456	3.23	4228	3.23	0	.00	0	.00	2064	.79	1032	.79

N = KB	ERROR MAGNITUDES		B = 12 A = 8				B = 12 A = 9				B = 12 A = 10			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	3	2	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	7	4	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
48	15	8	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
60	31	16	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
72	63	32	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
84	127	64	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
96	255	128	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
108	511	256	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
120	1023	512	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
132	2047	1024	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
144	4095	2048	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
156	8191	4096	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
168	16383	8192	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
180	32767	16384	0	.00	0	.00	0	.00	0	.00	1	.00	1	.01
192	65535	32768	0	.00	0	.00	0	.00	0	.00	4	.01	3	.01
204	131071	65536	0	.00	0	.00	0	.00	0	.00	16	.01	12	.02
216	262143	131072	0	.00	0	.00	0	.00	0	.00	64	.02	48	.04

N = KB	ERROR MAGNITUDES		B = 12 A = 11				B = 12 A = 12			
			MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
	TOTAL	INCREMENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00
24	3	2	0	.00	0	.00	0	.00	0	.00
36	7	4	0	.00	0	.00	0	.00	0	.00
48	15	8	0	.00	0	.00	0	.00	0	.00
60	31	16	0	.00	0	.00	0	.00	0	.00
72	63	32	0	.00	0	.00	0	.00	0	.00
84	127	64	0	.00	0	.00	0	.00	0	.00
96	255	128	0	.00	0	.00	0	.00	0	.00
108	511	256	0	.00	0	.00	0	.00	0	.00
120	1023	512	0	.00	0	.00	0	.00	0	.00
132	2047	1024	1	.05	1	.10	0	.00	0	.00
144	4095	2048	2	.05	1	.05	0	.00	0	.00
156	8191	4096	4	.05	2	.05	0	.00	0	.00
168	16383	8192	8	.05	4	.05	0	.00	0	.00
180	32767	16384	16	.05	8	.05	0	.00	0	.00
192	65535	32768	32	.05	16	.05	0	.00	0	.00
204	131071	65536	64	.05	32	.05	0	.00	0	.00
216	262143	131072	128	.05	64	.05	0	.00	0	.00

II. Determinate Faults, Fixed-Distance Check Factors α.

Table A-11. Check factors α for weights 1 and 2 (distance 3)

Byte length →		$B = 2$			$B = 3$			$B = 4$			$B = 5$			$B = 6$			$B = 7$			$B = 8$			$B = 9$			$B = 10$			$B = 12$		
α	N	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B
55	20	2.25	23	10	0.79	1	[b]	3.23	1	5	6.67	1	4	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	2	0	0	[b]
75	20	1.96	20	10	0.79	1	[b]	0	0	5	6.67	1	4	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	2	0	0	[b]
49	21	2.25	46	[b]	0	0	7	1.59	1	[b]	0	0	[b]	0	0	[b]	14.29	1	3	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
69	22	1.61	33	11	1.57	4	[b]	1.59	1	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
47	23	2.20	90	[b]	1.57	4	[b]	3.17	2	[b]	6.45	2	[b]	0	0	[b]	6.67	1	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
87	28	1.20	196	14	0.88	9	[b]	0	0	7	0	0	[b]	0	0	[b]	6.67	1	4	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
77	30	1.28	421	15	0.98	10	10	2.75	7	[b]	1.59	1	6	0	0	5	0	0	[b]	0	0	[b]	0	0	[b]	0	0	3	0	0	[b]
71	35	1.41	3700	[b]	1.44	59	[b]	0.78	4	[b]	0.79	1	7	3.17	2	[b]	3.23	1	5	3.23	1	[b]	0	0	[b]	0	0	[b]	0	0	[b]
95	36	1.10	2881	18	1.68	69	12	1.76	9	9	1.18	3	[b]	1.59	1	6	1.59	1	[b]	0	0	[b]	6.67	1	4	0	0	[b]	0	0	3
79	39	—	—	—	1.28	105	13	1.37	14	[b]	0.39	1	[b]	0	0	[b]	1.59	1	[b]	0	0	[b]	3.23	1	[b]	0	0	[b]	0	0	[b]
103	51	—	—	—	0.91	1191	17	0.94	77	[b]	0.64	13	[b]	1.37	7	[b]	1.18	3	[b]	0.79	1	[b]	0	0	[b]	1.59	1	[b]	0	0	[b]

[a] Miss % $= 100 \times$ (Miss count)$/(2^M - 1)$, with integer M satisfying $M - 1 < N/B \leq M$.

[b] N/B is not an integer; processing is not convenient.

Table A-12. Check factors α for weights 1, 2, and 3 (distance 4)

Byte length →		$B = 2$			$B = 3$			$B = 4$			$B = 5$			$B = 6$			$B = 7$			$B = 8$			$B = 9$			$B = 10$			$B = 12$		
α	N	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B	Miss %[a]	Miss Count	N/B
89	11	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
105	12	1.59	1	6	0	0	4	0	0	3	0	0	[b]	0	0	2	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	1
267	22	0	0	11	1.18	3	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]	0	0	[b]
357	24	0.81	33	12	0	0	8	1.59	1	6	0	0	[b]	0	0	4	0	0	[b]	0	0	3	0	0	[b]	0	0	[b]	0	0	2
555	36	0.18	469	18	0.51	21	12	0	0	9	0.39	1	[b]	1.59	1	6	1.59	1	[b]	0	0	[b]	6.67	1	4	0	0	[b]	0	0	3

[a] Miss % $= 100 \times$ (Miss count)$/(2^M - 1)$, with integer M satisfying $M - 1 < N/B \leq M$.

[b] N/B is not an integer; processing is not convenient.

III. Indeterminate Faults, Check Factors 2^A-1.

Table A-13. Byte length $B = 2$

N = KB	ERROR MAGNITUDES		B = 2 A = 2				B = 2 A = 3				B = 2 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
6	13	9	4	30.77	3	33.33	1	7.69	1	11.11	1	7.69	1	11.11
8	40	27	13	32.50	9	33.33	5	12.50	4	14.81	4	10.00	3	11.11
10	121	81	40	33.06	27	33.33	17	14.05	12	14.81	10	8.26	6	7.41
12	364	243	121	33.24	81	33.33	52	14.29	35	14.40	25	6.87	15	6.17
14	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	76	6.95	51	7.00
16	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	228	6.95	152	6.95
18	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	669	6.80	441	6.72
20	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	1976	6.69	1307	6.64
22	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	5926	6.69	3950	6.69
24	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	17769	6.69	11843	6.69

N = KB	ERROR MAGNITUDES		B = 2 A = 5				B = 2 A = 6				B = 2 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	40	27	0	.00	0	.00	1	2.50	1	3.70	0	.00	0	.00
10	121	81	1	.83	1	1.23	4	3.31	3	3.70	0	.00	0	.00
12	364	243	7	1.92	6	2.47	13	3.57	9	3.70	0	.00	0	.00
14	1093	729	31	2.84	24	3.29	31	2.84	18	2.47	1	.09	1	.14
16	3280	2187	107	3.26	76	3.48	73	2.23	42	1.92	9	.27	8	.37
18	9841	6561	329	3.34	222	3.38	172	1.75	99	1.51	45	.46	36	.55
20	29524	19683	1004	3.40	675	3.43	517	1.75	345	1.75	201	.68	156	.79
22	88573	59049	2940	3.32	1936	3.28	1545	1.74	1028	1.74	701	.79	500	.85
24	265720	177147	8644	3.25	5704	3.22	4597	1.73	3052	1.72	2171	.82	1470	.83

N = KB	ERROR MAGNITUDES		B = 2 A = 8				B = 2 A = 9				B = 2 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
4	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	40	27	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	121	81	1	.83	1	1.23	0	.00	0	.00	0	.00	0	.00
12	364	243	4	1.10	3	1.23	0	.00	0	.00	1	.27	1	.41
14	1093	729	13	1.19	9	1.23	0	.00	0	.00	4	.37	3	.41
16	3280	2187	40	1.22	27	1.23	0	.00	0	.00	13	.40	9	.41
18	9841	6561	94	.96	54	.82	1	.01	1	.02	40	.41	27	.41
20	29524	19683	220	.75	126	.64	11	.04	10	.05	121	.41	81	.41
22	88573	59049	514	.58	294	.50	59	.07	48	.08	283	.32	162	.27
24	265720	177147	1201	.45	687	.39	295	.11	236	.13	661	.25	378	.21

N = KB	ERROR MAGNITUDES		B = 2 A = 11				B = 2 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
2	1	1	0	.00	0	.00	0	.00	0	.00
4	4	3	0	.00	0	.00	0	.00	0	.00
6	13	9	0	.00	0	.00	0	.00	0	.00
8	40	27	0	.00	0	.00	0	.00	0	.00
10	121	81	0	.00	0	.00	0	.00	0	.00
12	364	243	0	.00	0	.00	0	.00	0	.00
14	1093	729	0	.00	0	.00	1	.09	1	.14
16	3280	2187	0	.00	0	.00	4	.12	3	.14
18	9841	6561	0	.00	0	.00	13	.13	9	.14
20	29524	19683	0	.00	0	.00	40	.14	27	.14
22	88573	59049	1	.00	1	.00	121	.14	81	.14
24	265720	177147	13	.00	12	.01	364	.14	243	.14

Table A-14. Byte length $B = 3$

N = KB	ERROR MAGNITUDES		B = 3 A = 2				B = 3 A = 3				B = 3 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	4	3	1	25.00	1	33.33	1	25.00	1	33.33	0	.00	0	.00
9	13	9	4	30.77	3	33.33	3	23.08	2	22.22	0	.00	0	.00
12	40	27	13	32.50	9	33.33	9	22.50	6	22.22	1	2.50	1	3.70
15	121	81	40	33.06	27	33.33	25	20.66	16	19.75	6	4.96	5	6.17
18	364	243	121	33.24	81	33.33	70	19.23	45	18.52	22	6.04	16	6.58
21	1093	729	364	33.30	243	33.33	197	18.02	127	17.42	74	6.77	52	7.13
24	3280	2187	1093	33.32	729	33.33	561	17.10	364	16.64	225	6.86	151	6.90
27	9841	6561	3280	33.33	2187	33.33	1614	16.40	1053	16.05	664	6.75	439	6.69
30	29524	19683	9841	33.33	6561	33.33	4686	15.87	3072	15.61	1976	6.69	1312	6.67
33	88573	59049	29524	33.33	19683	33.33	13706	15.47	9020	15.28	5896	6.66	3920	6.64
36	265720	177147	88573	33.33	59049	33.33	40326	15.18	26620	15.03	17681	6.65	11785	6.65

N = KB	ERROR MAGNITUDES		B = 3 A = 5				B = 3 A = 6				B = 3 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
9	13	9	0	.00	0	.00	1	7.69	1	11.11	0	.00	0	.00
12	40	27	0	.00	0	.00	4	10.00	3	11.11	0	.00	0	.00
15	121	81	1	.83	1	1.23	10	8.26	6	7.41	0	.00	0	.00
18	364	243	7	1.92	6	2.47	24	6.59	14	5.76	0	.00	0	.00
21	1093	729	31	2.84	24	3.29	66	6.04	42	5.76	1	.09	1	.14
24	3280	2187	107	3.26	76	3.48	180	5.49	114	5.21	9	.27	8	.37
27	9841	6561	329	3.34	222	3.38	484	4.92	304	4.63	49	.50	40	.61
30	29524	19683	1004	3.40	675	3.43	1300	4.40	816	4.15	183	.62	134	.68
33	88573	59049	2940	3.32	1936	3.28	3595	4.06	2295	3.89	639	.72	456	.77
36	265720	177147	8644	3.25	5704	3.22	9940	3.74	6345	3.58	2191	.82	1552	.88

N = KB	ERROR MAGNITUDES		B = 3 A = 8				B = 3 A = 9				B = 3 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
6	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
9	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	40	27	0	.00	0	.00	1	2.50	1	3.70	0	.00	0	.00
15	121	81	0	.00	0	.00	4	3.31	3	3.70	0	.00	0	.00
18	364	243	0	.00	0	.00	13	3.57	9	3.70	0	.00	0	.00
21	1093	729	0	.00	0	.00	31	2.84	18	2.47	0	.00	0	.00
24	3280	2187	1	.03	1	.05	73	2.23	42	1.92	0	.00	0	.00
27	9841	6561	10	.10	9	.14	171	1.74	98	1.49	0	.00	0	.00
30	29524	19683	58	.20	48	.24	465	1.57	294	1.49	1	.00	1	.01
33	88573	59049	262	.30	204	.35	1263	1.43	798	1.35	12	.01	11	.02
36	265720	177147	918	.35	656	.37	3429	1.29	2166	1.22	76	.03	64	.04

N = KB	ERROR MAGNITUDES		B = 3 A = 11				B = 3 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
3	1	1	0	.00	0	.00	0	.00	0	.00
6	4	3	0	.00	0	.00	0	.00	0	.00
9	13	9	0	.00	0	.00	0	.00	0	.00
12	40	27	0	.00	0	.00	0	.00	0	.00
15	121	81	0	.00	0	.00	1	.83	1	1.23
18	364	243	0	.00	0	.00	4	1.10	3	1.23
21	1093	729	0	.00	0	.00	13	1.19	9	1.23
24	3280	2187	0	.00	0	.00	40	1.22	27	1.23
27	9841	6561	0	.00	0	.00	94	.96	54	.82
30	29524	19683	0	.00	0	.00	220	.75	126	.64
33	88573	59049	1	.00	1	.00	514	.58	294	.50
36	265720	177147	13	.00	12	.01	1200	.45	686	.39

Table A-15. Byte length $B = 4$

N = KB	ERROR MAGNITUDES		B = 4 A = 2				B = 4 A = 3				B = 4 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	4	3	1	25.00	1	33.33	0	.00	0	.00	1	25.00	1	33.33
12	13	9	4	30.77	3	33.33	1	7.69	1	11.11	3	23.08	2	22.22
16	40	27	13	32.50	9	33.33	5	12.50	4	14.81	9	22.50	6	22.22
20	121	81	40	33.06	27	33.33	17	14.05	12	14.81	25	20.66	16	19.75
24	364	243	121	33.24	81	33.33	52	14.29	35	14.40	70	19.23	45	18.52
28	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	196	17.93	126	17.28
32	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	553	16.86	357	16.32
36	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	1569	15.94	1016	15.49
40	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	4476	15.16	2907	14.77
44	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	12826	14.48	8350	14.14
48	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	36894	13.88	24068	13.59

N = KB	ERROR MAGNITUDES		B = 4 A = 5				B = 4 A = 6				B = 4 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	40	27	0	.00	0	.00	1	2.50	1	3.70	0	.00	0	.00
20	121	81	1	.83	1	1.23	4	3.31	3	3.70	0	.00	0	.00
24	364	243	7	1.92	6	2.47	13	3.57	9	3.70	0	.00	0	.00
28	1093	729	27	2.47	20	2.74	31	2.84	18	2.47	1	.09	1	.14
32	3280	2187	97	2.96	70	3.20	73	2.23	42	1.92	9	.27	8	.37
36	9841	6561	329	3.34	232	3.54	172	1.75	99	1.51	49	.50	40	.61
40	29524	19683	1004	3.40	675	3.43	517	1.75	345	1.75	183	.62	134	.68
44	88573	59049	2940	3.32	1936	3.28	1545	1.74	1028	1.74	639	.72	456	.77
48	265720	177147	8708	3.28	5768	3.26	4597	1.73	3052	1.72	2191	.82	1552	.88

N = KB	ERROR MAGNITUDES		B = 4 A = 8				B = 4 A = 9				B = 4 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
8	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	13	9	1	7.69	1	11.11	0	.00	0	.00	0	.00	0	.00
16	40	27	4	10.00	3	11.11	0	.00	0	.00	0	.00	0	.00
20	121	81	10	8.26	6	7.41	0	.00	0	.00	0	.00	0	.00
24	364	243	24	6.59	14	5.76	0	.00	0	.00	1	.27	1	.41
28	1093	729	66	6.04	42	5.76	0	.00	0	.00	4	.37	3	.41
32	3280	2187	180	5.49	114	5.21	0	.00	0	.00	13	.40	9	.41
36	9841	6561	484	4.92	304	4.63	1	.01	1	.02	40	.41	27	.41
40	29524	19683	1300	4.40	816	4.15	11	.04	10	.05	121	.41	81	.41
44	88573	59049	3595	4.06	2295	3.89	71	.08	60	.10	283	.32	162	.27
48	265720	177147	9940	3.74	6345	3.58	279	.10	208	.12	661	.25	378	.21

N = KB	ERROR MAGNITUDES		B = 4 A = 11				B = 4 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
4	1	1	0	.00	0	.00	0	.00	0	.00
8	4	3	0	.00	0	.00	0	.00	0	.00
12	13	9	0	.00	0	.00	0	.00	0	.00
16	40	27	0	.00	0	.00	1	2.50	1	3.70
20	121	81	0	.00	0	.00	4	3.31	3	3.70
24	364	243	0	.00	0	.00	13	3.57	9	3.70
28	1093	729	0	.00	0	.00	31	2.84	18	2.47
32	3280	2187	0	.00	0	.00	73	2.23	42	1.92
36	9841	6561	0	.00	0	.00	171	1.74	98	1.49
40	29524	19683	0	.00	0	.00	465	1.57	294	1.49
44	88573	59049	1	.00	1	.00	1263	1.43	798	1.35
48	265720	177147	13	.00	12	.01	3429	1.29	2166	1.22

Table A-16. Byte length $B = 5$

N = KB	ERROR MAGNITUDES		B = 5 A = 2				B = 5 A = 3				B = 5 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
15	13	9	4	30.77	3	33.33	1	7.69	1	11.11	0	.00	0	.00
20	40	27	13	32.50	9	33.33	5	12.50	4	14.81	1	2.50	1	3.70
25	121	81	40	33.06	27	33.33	17	14.05	12	14.81	6	4.96	5	6.17
30	364	243	121	33.24	81	33.33	52	14.29	35	14.40	22	6.04	16	6.58
35	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	74	6.77	52	7.13
40	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	225	6.86	151	6.90
45	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	664	6.75	439	6.69
50	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	1976	6.69	1312	6.67
55	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	5896	6.66	3920	6.64
60	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	17681	6.65	11785	6.65

N = KB	ERROR MAGNITUDES		B = 5 A = 5				B = 5 A = 6				B = 5 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
15	13	9	3	23.08	2	22.22	0	.00	0	.00	0	.00	0	.00
20	40	27	9	22.50	6	22.22	0	.00	0	.00	0	.00	0	.00
25	121	81	25	20.66	16	19.75	0	.00	0	.00	0	.00	0	.00
30	364	243	70	19.23	45	18.52	1	.27	1	.41	0	.00	0	.00
35	1093	729	196	17.93	126	17.28	8	.73	7	.96	1	.09	1	.14
40	3280	2187	553	16.86	357	16.32	32	.98	24	1.10	9	.27	8	.37
45	9841	6561	1569	15.94	1016	15.49	120	1.22	88	1.34	45	.46	36	.55
50	29524	19683	4476	15.16	2907	14.77	434	1.47	314	1.60	201	.68	156	.79
55	88573	59049	12826	14.48	8350	14.14	1482	1.67	1048	1.77	701	.79	500	.85
60	265720	177147	36894	13.88	24068	13.59	4537	1.71	3055	1.72	2171	.82	1470	.83

N - KB	ERROR MAGNITUDES		B = 5 A = 8				B = 5 A = 9				B = 5 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
10	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
15	13	9	0	.00	0	.00	0	.00	0	.00	1	7.69	1	11.11
20	40	27	0	.00	0	.00	0	.00	0	.00	4	10.00	3	11.11
25	121	81	0	.00	0	.00	0	.00	0	.00	10	8.26	6	7.41
30	364	243	0	.00	0	.00	0	.00	0	.00	24	6.59	14	5.76
35	1093	729	0	.00	0	.00	0	.00	0	.00	66	6.04	42	5.76
40	3280	2187	1	.03	1	.05	0	.00	0	.00	180	5.49	114	5.21
45	9841	6561	10	.10	9	.14	1	.01	1	.02	484	4.92	304	4.63
50	29524	19683	58	.20	48	.24	11	.04	10	.05	1300	4.40	816	4.15
55	88573	59049	262	.30	204	.35	71	.08	60	.10	3595	4.06	2295	3.89
60	265720	177147	918	.35	656	.37	279	.10	208	.12	9940	3.74	6345	3.58

N = KB	ERROR MAGNITUDES		B = 5 A = 11				B = 5 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT	MISS COUNT	TOTAL PERCENT	MISS INCREMENT COUNT	PERCENT
5	1	1	0	.00	0	.00	0	.00	0	.00
10	4	3	0	.00	0	.00	0	.00	0	.00
15	13	9	0	.00	0	.00	0	.00	0	.00
20	40	27	0	.00	0	.00	0	.00	0	.00
25	121	81	0	.00	0	.00	0	.00	0	.00
30	364	243	0	.00	0	.00	0	.00	0	.00
35	1093	729	0	.00	0	.00	0	.00	0	.00
40	3280	2187	0	.00	0	.00	0	.00	0	.00
45	9841	6561	0	.00	0	.00	0	.00	0	.00
50	29524	19683	0	.00	0	.00	0	.00	0	.00
55	88573	59049	1	.00	1	.00	0	.00	0	.00
60	265720	177147	13	.00	12	.01	1	.00	1	.00

Table A-17. Byte length $B = 6$

N = KB	ERROR MAGNITUDES		B = 6 A = 2				B = 6 A = 3				B = 6 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	4	3	1	25.00	1	33.33	1	25.00	1	33.33	0	.00	0	.00
18	13	9	4	30.77	3	33.33	3	23.08	2	22.22	1	7.69	1	11.11
24	40	27	13	32.50	9	33.33	9	22.50	6	22.22	4	10.00	3	11.11
30	121	81	40	33.06	27	33.33	25	20.66	16	19.75	10	8.26	6	7.41
36	364	243	121	33.24	81	33.33	70	19.23	45	18.52	25	6.87	15	6.17
42	1093	729	364	33.30	243	33.33	197	18.02	127	17.42	76	6.95	51	7.00
48	3280	2187	1093	33.32	729	33.33	561	17.10	364	16.64	228	6.95	152	6.95
54	9841	6561	3280	33.33	2187	33.33	1614	16.40	1053	16.05	669	6.80	441	6.72
60	29524	19683	9841	33.33	6561	33.33	4686	15.87	3072	15.61	1976	6.69	1307	6.64
66	88573	59049	29524	33.33	19683	33.33	13706	15.47	9020	15.28	5926	6.69	3950	6.69
72	265720	177147	88573	33.33	59049	33.33	40326	15.18	26620	15.03	17769	6.69	11843	6.69

N = KB	ERROR MAGNITUDES		B = 6 A = 5				B = 6 A = 6				B = 6 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	4	3	0	.00	0	.00	1	25.00	1	33.33	0	.00	0	.00
18	13	9	0	.00	0	.00	3	23.08	2	22.22	0	.00	0	.00
24	40	27	0	.00	0	.00	9	22.50	6	22.22	0	.00	0	.00
30	121	81	1	.83	1	1.23	25	20.66	16	19.75	0	.00	0	.00
36	364	243	7	1.92	6	2.47	70	19.23	45	18.52	0	.00	0	.00
42	1093	729	27	2.47	20	2.74	196	17.93	126	17.28	1	.09	1	.14
48	3280	2187	97	2.96	70	3.20	553	16.86	357	16.32	9	.27	8	.37
54	9841	6561	329	3.34	232	3.54	1569	15.94	1016	15.49	37	.38	28	.43
60	29524	19683	1004	3.40	675	3.43	4476	15.16	2907	14.77	143	.48	106	.54
66	88573	59049	2940	3.32	1936	3.28	12826	14.48	8350	14.14	539	.61	396	.67
72	265720	177147	8708	3.28	5768	3.26	36894	13.88	24068	13.59	1961	.74	1422	.80

N = KB	ERROR MAGNITUDES		B = 6 A = 8				B = 6 A = 9				B = 6 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
12	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	40	27	0	.00	0	.00	1	2.50	1	3.70	0	.00	0	.00
30	121	81	1	.83	1	1.23	4	3.31	3	3.70	0	.00	0	.00
36	364	243	4	1.10	3	1.23	13	3.57	9	3.70	1	.27	1	.41
42	1093	729	13	1.19	9	1.23	31	2.84	18	2.47	4	.37	3	.41
48	3280	2187	40	1.22	27	1.23	73	2.23	42	1.92	13	.40	9	.41
54	9841	6561	94	.96	54	.82	171	1.74	98	1.49	40	.41	27	.41
60	29524	19683	220	.75	126	.64	465	1.57	294	1.49	121	.41	81	.41
66	88573	59049	514	.58	294	.50	1263	1.43	798	1.35	283	.32	162	.27
72	265720	177147	1201	.45	687	.39	3429	1.29	2166	1.22	661	.25	378	.21

N = KB	ERROR MAGNITUDES		B = 6 A = 11				B = 6 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
6	1	1	0	.00	0	.00	0	.00	0	.00
12	4	3	0	.00	0	.00	0	.00	0	.00
18	13	9	0	.00	0	.00	1	7.69	1	11.11
24	40	27	0	.00	0	.00	4	10.00	3	11.11
30	121	81	0	.00	0	.00	10	8.26	6	7.41
36	364	243	0	.00	0	.00	24	6.59	14	5.76
42	1093	729	0	.00	0	.00	66	6.04	42	5.76
48	3280	2187	0	.00	0	.00	180	5.49	114	5.21
54	9841	6561	0	.00	0	.00	484	4.92	304	4.63
60	29524	19683	0	.00	0	.00	1300	4.40	816	4.15
66	88573	59049	1	.00	1	.00	3595	4.06	2295	3.89
72	265720	177147	13	.00	12	.01	9940	3.74	6345	3.58

Table A-18. Byte length $B = 7$

N = KB	ERROR MAGNITUDES		B = 7 A = 2				B = 7 A = 3				B = 7 A = 4			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
21	13	9	4	30.77	3	33.33	1	7.69	1	11.11	0	.00	0	.00
28	40	27	13	32.50	9	33.33	5	12.50	4	14.81	1	2.50	1	3.70
35	121	81	40	33.06	27	33.33	17	14.05	12	14.81	6	4.96	5	6.17
42	364	243	121	33.24	81	33.33	52	14.29	35	14.40	22	6.04	16	6.58
49	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	74	6.77	52	7.13
56	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	225	6.86	151	6.90
63	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	664	6.75	439	6.69
70	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	1976	6.69	1312	6.67
77	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	5896	6.66	3920	6.64
84	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	17681	6.65	11785	6.65

N = KB	ERROR MAGNITUDES		B = 7 A = 5				B = 7 A = 6				B = 7 A = 7			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	4	3	0	.00	0	.00	0	.00	0	.00	1	25.00	1	33.33
21	13	9	0	.00	0	.00	0	.00	0	.00	3	23.08	2	22.22
28	40	27	0	.00	0	.00	0	.00	0	.00	9	22.50	6	22.22
35	121	81	1	.83	1	1.23	0	.00	0	.00	25	20.66	16	19.75
42	364	243	7	1.92	6	2.47	1	.27	1	.41	70	19.23	45	18.52
49	1093	729	31	2.84	24	3.29	8	.73	7	.96	196	17.93	126	17.28
56	3280	2187	107	3.26	76	3.48	32	.98	24	1.10	553	16.86	357	16.32
63	9841	6561	329	3.34	222	3.38	120	1.22	88	1.34	1569	15.94	1016	15.49
70	29524	19683	1004	3.40	675	3.43	434	1.47	314	1.60	4476	15.16	2907	14.77
77	88573	59049	2940	3.32	1936	3.28	1482	1.67	1048	1.77	12826	14.48	8350	14.14
84	265720	177147	8644	3.25	5704	3.22	4537	1.71	3055	1.72	36894	13.88	24068	13.59

N = KB	ERROR MAGNITUDES		B = 7 A = 8				B = 7 A = 9				B = 7 A = 10			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
14	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
21	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
28	40	27	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
35	121	81	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
42	364	243	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
49	1093	729	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
56	3280	2187	1	.03	1	.05	0	.00	0	.00	0	.00	0	.00
63	9841	6561	10	.10	9	.14	1	.01	1	.02	0	.00	0	.00
70	29524	19683	42	.14	32	.16	11	.04	10	.05	1	.00	1	.01
77	88573	59049	166	.19	124	.21	59	.07	48	.08	12	.01	11	.02
84	265720	177147	644	.24	478	.27	295	.11	236	.13	76	.03	64	.04

N = KB	ERROR MAGNITUDES		B = 7 A = 11				B = 7 A = 12			
	TOTAL	INCREMENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT	MISS COUNT	TOTAL PERCENT	MISS COUNT	INCREMENT PERCENT
7	1	1	0	.00	0	.00	0	.00	0	.00
14	4	3	0	.00	0	.00	0	.00	0	.00
21	13	9	0	.00	0	.00	0	.00	0	.00
28	40	27	0	.00	0	.00	0	.00	0	.00
35	121	81	0	.00	0	.00	0	.00	0	.00
42	364	243	0	.00	0	.00	0	.00	0	.00
49	1093	729	0	.00	0	.00	0	.00	0	.00
56	3280	2187	0	.00	0	.00	0	.00	0	.00
63	9841	6561	0	.00	0	.00	0	.00	0	.00
70	29524	19683	0	.00	0	.00	0	.00	0	.00
77	88573	59049	1	.00	1	.00	0	.00	0	.00
84	265720	177147	13	.00	12	.01	1	.00	1	.00

Table A-19. Byte length $B = 8$

N = KB	ERROR MAGNITUDES		B = 8 A = 2				B = 8 A = 3				B = 8 A = 4			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	4	3	1	25.00	1	33.33	0	.00	0	.00	1	25.00	1	33.33
24	13	9	4	30.77	3	33.33	1	7.69	1	11.11	3	23.08	2	22.22
32	40	27	13	32.50	9	33.33	5	12.50	4	14.81	9	22.50	6	22.22
40	121	81	40	33.06	27	33.33	17	14.05	12	14.81	25	20.66	16	19.75
48	364	243	121	33.24	81	33.33	52	14.29	35	14.40	70	19.23	45	18.52
56	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	196	17.93	126	17.28
64	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	553	16.86	357	16.32
72	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	1569	15.94	1016	15.49
80	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	4476	15.16	2907	14.77
88	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	12826	14.48	8350	14.14
96	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	36894	13.88	24068	13.59

N = KB	ERROR MAGNITUDES		B = 8 A = 5				B = 8 A = 6				B = 8 A = 7			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	13	9	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
32	40	27	0	.00	0	.00	1	2.50	1	3.70	0	.00	0	.00
40	121	81	1	.83	1	1.23	4	3.31	3	3.70	0	.00	0	.00
48	364	243	7	1.92	6	2.47	13	3.57	9	3.70	0	.00	0	.00
56	1093	729	31	2.84	24	3.29	31	2.84	18	2.47	1	.09	1	.14
64	3280	2187	107	3.26	76	3.48	73	2.23	42	1.92	9	.27	8	.37
72	9841	6561	329	3.34	222	3.38	172	1.75	99	1.51	37	.38	28	.43
80	29524	19683	1004	3.40	675	3.43	517	1.75	345	1.75	143	.48	106	.54
88	88573	59049	2940	3.32	1936	3.28	1545	1.74	1028	1.74	539	.61	396	.67
96	265720	177147	8644	3.25	5704	3.22	4597	1.73	3052	1.72	1961	.74	1422	.80

N = KB	ERROR MAGNITUDES		B = 8 A = 8				B = 8 A = 9				B = 8 A = 10			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
16	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
24	13	9	3	23.08	2	22.22	0	.00	0	.00	0	.00	0	.00
32	40	27	9	22.50	6	22.22	0	.00	0	.00	0	.00	0	.00
40	121	81	25	20.66	16	19.75	0	.00	0	.00	0	.00	0	.00
48	364	243	70	19.23	45	18.52	0	.00	0	.00	1	.27	1	.41
56	1093	729	196	17.93	126	17.28	0	.00	0	.00	4	.37	3	.41
64	3280	2187	553	16.86	357	16.32	0	.00	0	.00	13	.40	9	.41
72	9841	6561	1569	15.94	1016	15.49	1	.01	1	.02	40	.41	27	.41
80	29524	19683	4476	15.16	2907	14.77	11	.04	10	.05	121	.41	81	.41
88	88573	59049	12826	14.48	8350	14.14	47	.05	36	.06	283	.32	162	.27
96	265720	177147	36894	13.88	24068	13.59	189	.07	142	.08	661	.25	378	.21

N = KB	ERROR MAGNITUDES		B = 8 A = 11				B = 8 A = 12			
	TOTAL	INCREMENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT	MISS TOTAL COUNT	MISS TOTAL PERCENT	MISS INCREMENT COUNT	MISS INCREMENT PERCENT
8	1	1	0	.00	0	.00	0	.00	0	.00
16	4	3	0	.00	0	.00	0	.00	0	.00
24	13	9	0	.00	0	.00	0	.00	0	.00
32	40	27	0	.00	0	.00	1	2.50	1	3.70
40	121	81	0	.00	0	.00	4	3.31	3	3.70
48	364	243	0	.00	0	.00	13	3.57	9	3.70
56	1093	729	0	.00	0	.00	31	2.84	18	2.47
64	3280	2187	0	.00	0	.00	73	2.23	42	1.92
72	9841	6561	0	.00	0	.00	171	1.74	98	1.49
80	29524	19683	0	.00	0	.00	465	1.57	294	1.49
88	88573	59049	1	.00	1	.00	1263	1.43	798	1.35
96	265720	177147	13	.00	12	.01	3429	1.29	2166	1.22

Table A-20. Byte length $B = 9$

N = KB	ERROR MAGNITUDES		B = 10 A = 2				B = 10 A = 3				B = 10 A = 4			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
30	13	9	4	30.77	3	33.33	1	7.69	1	11.11	1	7.69	1	11.11
40	40	27	13	32.50	9	33.33	5	12.50	4	14.81	4	10.00	3	11.11
50	121	81	40	33.06	27	33.33	17	14.05	12	14.81	10	8.26	6	7.41
60	364	243	121	33.24	81	33.33	52	14.29	35	14.40	25	6.87	15	6.17
70	1093	729	364	33.30	243	33.33	156	14.27	104	14.27	76	6.95	51	7.00
80	3280	2187	1093	33.32	729	33.33	468	14.27	312	14.27	228	6.95	152	6.95
90	9841	6561	3280	33.33	2187	33.33	1405	14.28	937	14.28	669	6.80	441	6.72
100	29524	19683	9841	33.33	6561	33.33	4217	14.28	2812	14.29	1976	6.69	1307	6.64
110	88573	59049	29524	33.33	19683	33.33	12653	14.29	8436	14.29	5926	6.69	3950	6.69
120	265720	177147	88573	33.33	59049	33.33	37960	14.29	25307	14.29	17769	6.69	11843	6.69

N = KB	ERROR MAGNITUDES		B = 10 A = 5				B = 10 A = 6				B = 10 A = 7			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	4	3	1	25.00	1	33.33	0	.00	0	.00	0	.00	0	.00
30	13	9	3	23.08	2	22.22	0	.00	0	.00	0	.00	0	.00
40	40	27	9	22.50	6	22.22	1	2.50	1	3.70	0	.00	0	.00
50	121	81	25	20.66	16	19.75	4	3.31	3	3.70	0	.00	0	.00
60	364	243	70	19.23	45	18.52	13	3.57	9	3.70	0	.00	0	.00
70	1093	729	196	17.93	126	17.28	31	2.84	18	2.47	1	.09	1	.14
80	3280	2187	553	16.86	357	16.32	73	2.23	42	1.92	9	.27	8	.37
90	9841	6561	1569	15.94	1016	15.49	172	1.75	99	1.51	49	.50	40	.61
100	29524	19683	4476	15.16	2907	14.77	517	1.75	345	1.75	183	.62	134	.68
110	88573	59049	12826	14.48	8350	14.14	1545	1.74	1028	1.74	639	.72	456	.77
120	265720	177147	36894	13.88	24068	13.59	4597	1.73	3052	1.72	2191	.82	1552	.88

N = KB	ERROR MAGNITUDES		B = 10 A = 8				B = 10 A = 9				B = 10 A = 10			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
20	4	3	0	.00	0	.00	0	.00	0	.00	1	25.00	1	33.33
30	13	9	0	.00	0	.00	0	.00	0	.00	3	23.08	2	22.22
40	40	27	0	.00	0	.00	0	.00	0	.00	9	22.50	6	22.22
50	121	81	1	.83	1	1.23	0	.00	0	.00	25	20.66	16	19.75
60	364	243	4	1.10	3	1.23	0	.00	0	.00	70	19.23	45	18.52
70	1093	729	13	1.19	9	1.23	0	.00	0	.00	196	17.93	126	17.28
80	3280	2187	40	1.22	27	1.23	0	.00	0	.00	553	16.86	357	16.32
90	9841	6561	94	.96	54	.82	1	.01	1	.02	1569	15.94	1016	15.49
100	29524	19683	220	.75	126	.64	11	.04	10	.05	4476	15.16	2907	14.77
110	88573	59049	514	.58	294	.50	47	.05	36	.06	12826	14.48	8350	14.14
120	265720	177147	1201	.45	687	.39	189	.07	142	.08	36894	13.88	24068	13.59

N = KB	ERROR MAGNITUDES		B = 10 A = 11				B = 10 A = 12			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
10	1	1	0	.00	0	.00	0	.00	0	.00
20	4	3	0	.00	0	.00	0	.00	0	.00
30	13	9	0	.00	0	.00	0	.00	0	.00
40	40	27	0	.00	0	.00	0	.00	0	.00
50	121	81	0	.00	0	.00	0	.00	0	.00
60	364	243	0	.00	0	.00	0	.00	0	.00
70	1093	729	0	.00	0	.00	1	.09	1	.14
80	3280	2187	0	.00	0	.00	4	.12	3	.14
90	9841	6561	0	.00	0	.00	13	.13	9	.14
100	29524	19683	0	.00	0	.00	40	.14	27	.14
110	88573	59049	1	.00	1	.00	121	.14	81	.14
120	265720	177147	13	.00	12	.01	364	.14	243	.14

Table A-21. Byte length B = 10

N = KB	ERROR MAGNITUDES		B = 9 A = 2				B = 9 A = 3				B = 9 A = 4			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	4	3	1	25.00	1	33.33	1	25.00	1	33.33	0	.00	0	.00
27	13	9	4	30.77	3	33.33	3	23.08	2	22.22	0	.00	0	.00
36	40	27	13	32.50	9	33.33	9	22.50	6	22.22	1	2.50	1	3.70
45	121	81	40	33.06	27	33.33	25	20.66	16	19.75	6	4.96	5	6.17
54	364	243	121	33.24	81	33.33	70	19.23	45	18.52	22	6.04	16	6.58
63	1093	729	364	33.30	243	33.33	197	18.02	127	17.42	74	6.77	52	7.13
72	3280	2187	1093	33.32	729	33.33	561	17.10	364	16.64	225	6.86	151	6.90
81	9841	6561	3280	33.33	2187	33.33	1614	16.40	1053	16.05	664	6.75	439	6.69
90	29524	19683	9841	33.33	6561	33.33	4686	15.87	3072	15.61	1976	6.69	1312	6.67
99	88573	59049	29524	33.33	19683	33.33	13706	15.47	9020	15.28	5896	6.66	3920	6.64
108	265720	177147	88573	33.33	59049	33.33	40326	15.18	26620	15.03	17681	6.65	11785	6.65

N = KB	ERROR MAGNITUDES		B = 9 A = 5				B = 9 A = 6				B = 9 A = 7			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
27	13	9	0	.00	0	.00	1	7.69	1	11.11	0	.00	0	.00
36	40	27	0	.00	0	.00	4	10.00	3	11.11	0	.00	0	.00
45	121	81	1	.83	1	1.23	10	8.26	6	7.41	0	.00	0	.00
54	364	243	7	1.92	6	2.47	24	6.59	14	5.76	0	.00	0	.00
63	1093	729	27	2.47	20	2.74	66	6.04	42	5.76	1	.09	1	.14
72	3280	2187	97	2.96	70	3.20	180	5.49	114	5.21	9	.27	8	.37
81	9841	6561	329	3.34	232	3.54	484	4.92	304	4.63	45	.46	36	.55
90	29524	19683	1004	3.40	675	3.43	1300	4.40	816	4.15	201	.68	156	.79
99	88573	59049	2940	3.32	1936	3.28	3595	4.06	2295	3.89	701	.79	500	.85
108	265720	177147	8708	3.28	5768	3.26	9940	3.74	6345	3.58	2171	.82	1470	.83

N = KB	ERROR MAGNITUDES		B = 9 A = 8				B = 9 A = 9				B = 9 A = 10			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
18	4	3	0	.00	0	.00	1	25.00	1	33.33	0	.00	0	.00
27	13	9	0	.00	0	.00	3	23.08	2	22.22	0	.00	0	.00
36	40	27	0	.00	0	.00	9	22.50	6	22.22	0	.00	0	.00
45	121	81	0	.00	0	.00	25	20.66	16	19.75	0	.00	0	.00
54	364	243	0	.00	0	.00	70	19.23	45	18.52	0	.00	0	.00
63	1093	729	0	.00	0	.00	196	17.93	126	17.28	0	.00	0	.00
72	3280	2187	1	.03	1	.05	553	16.86	357	16.32	0	.00	0	.00
81	9841	6561	10	.10	9	.14	1569	15.94	1016	15.49	0	.00	0	.00
90	29524	19683	42	.14	32	.16	4476	15.16	2907	14.77	1	.00	1	.01
99	88573	59049	166	.19	124	.21	12826	14.48	8350	14.14	12	.01	11	.02
108	265720	177147	644	.24	478	.27	36894	13.88	24068	13.59	52	.02	40	.02

N = KB	ERROR MAGNITUDES		B = 9 A = 11				B = 9 A = 12			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
9	1	1	0	.00	0	.00	0	.00	0	.00
18	4	3	0	.00	0	.00	0	.00	0	.00
27	13	9	0	.00	0	.00	0	.00	0	.00
36	40	27	0	.00	0	.00	0	.00	0	.00
45	121	81	0	.00	0	.00	1	.83	1	1.23
54	364	243	0	.00	0	.00	4	1.10	3	1.23
63	1093	729	0	.00	0	.00	13	1.19	9	1.23
72	3280	2187	0	.00	0	.00	40	1.22	27	1.23
81	9841	6561	0	.00	0	.00	94	.96	54	.82
90	29524	19683	0	.00	0	.00	220	.75	126	.64
99	88573	59049	1	.00	1	.00	514	.58	294	.50
108	265720	177147	13	.00	12	.01	1200	.45	686	.39

Table A-22. Byte length $B = 12$

N = KB	ERROR MAGNITUDES		B = 12 A = 2				B = 12 A = 3				B = 12 A = 4			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	4	3	1	25.00	1	33.33	1	25.00	1	33.33	1	25.00	1	33.33
36	13	9	4	30.77	3	33.33	3	23.08	2	22.22	3	23.08	2	22.22
48	40	27	13	32.50	9	33.33	9	22.50	6	22.22	9	22.50	6	22.22
60	121	81	40	33.06	27	33.33	25	20.66	16	19.75	25	20.66	16	19.75
72	364	243	121	33.24	81	33.33	70	19.23	45	18.52	70	19.23	45	18.52
84	1093	729	364	33.30	243	33.33	197	18.02	127	17.42	196	17.93	126	17.28
96	3280	2187	1093	33.32	729	33.33	561	17.10	364	16.64	553	16.86	357	16.32
108	9841	6561	3280	33.33	2187	33.33	1614	16.40	1053	16.05	1569	15.94	1016	15.49
120	29524	19683	9841	33.33	6561	33.33	4686	15.87	3072	15.61	4476	15.16	2907	14.77
132	88573	59049	29524	33.33	19683	33.33	13706	15.47	9020	15.28	12826	14.48	8350	14.14
144	265720	177147	88573	33.33	59049	33.33	40326	15.18	26620	15.03	36894	13.88	24068	13.59

N = KB	ERROR MAGNITUDES		B = 12 A = 5				B = 12 A = 6				B = 12 A = 7			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	4	3	0	.00	0	.00	1	25.00	1	33.33	0	.00	0	.00
36	13	9	0	.00	0	.00	3	23.08	2	22.22	0	.00	0	.00
48	40	27	0	.00	0	.00	9	22.50	6	22.22	0	.00	0	.00
60	121	81	1	.83	1	1.23	25	20.66	16	19.75	0	.00	0	.00
72	364	243	7	1.92	6	2.47	70	19.23	45	18.52	0	.00	0	.00
84	1093	729	31	2.84	24	3.29	196	17.93	126	17.28	1	.09	1	.14
96	3280	2187	107	3.26	76	3.48	553	16.86	357	16.32	9	.27	8	.37
108	9841	6561	329	3.34	222	3.38	1569	15.94	1016	15.49	45	.46	36	.55
120	29524	19683	1004	3.40	675	3.43	4476	15.16	2907	14.77	201	.68	156	.79
132	88573	59049	2940	3.32	1936	3.28	12826	14.48	8350	14.14	701	.79	500	.85
144	265720	177147	8644	3.25	5704	3.22	36894	13.88	24068	13.59	2171	.82	1470	.83

N = KB	ERROR MAGNITUDES		B = 12 A = 8				B = 12 A = 9				B = 12 A = 10			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
24	4	3	0	.00	0	.00	0	.00	0	.00	0	.00	0	.00
36	13	9	1	7.69	1	11.11	0	.00	0	.00	0	.00	0	.00
48	40	27	4	10.00	3	11.11	1	2.50	1	3.70	0	.00	0	.00
60	121	81	10	8.26	6	7.41	4	3.31	3	3.70	0	.00	0	.00
72	364	243	24	6.59	14	5.76	13	3.57	9	3.70	1	.27	1	.41
84	1093	729	66	6.04	42	5.76	31	2.84	18	2.47	4	.37	3	.41
96	3280	2187	180	5.49	114	5.21	73	2.23	42	1.92	13	.40	9	.41
108	9841	6561	484	4.92	304	4.63	171	1.74	98	1.49	40	.41	27	.41
120	29524	19683	1300	4.40	816	4.15	465	1.57	294	1.49	121	.41	81	.41
132	88573	59049	3595	4.06	2295	3.89	1263	1.43	798	1.35	283	.32	162	.27
144	265720	177147	9940	3.74	6345	3.58	3429	1.29	2166	1.22	661	.25	378	.21

N = KB	ERROR MAGNITUDES		B = 12 A = 11				B = 12 A = 12			
	TOTAL	INCREMENT	MISS TOTAL		MISS INCREMENT		MISS TOTAL		MISS INCREMENT	
			COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT	COUNT	PERCENT
12	1	1	0	.00	0	.00	0	.00	0	.00
24	4	3	0	.00	0	.00	1	25.00	1	33.33
36	13	9	0	.00	0	.00	3	23.08	2	22.22
48	40	27	0	.00	0	.00	9	22.50	6	22.22
60	121	81	0	.00	0	.00	25	20.66	16	19.75
72	364	243	0	.00	0	.00	70	19.23	45	18.52
84	1093	729	0	.00	0	.00	196	17.93	126	17.28
96	3280	2187	0	.00	0	.00	553	16.86	357	16.32
108	9841	6561	0	.00	0	.00	1569	15.94	1016	15.49
120	29524	19683	0	.00	0	.00	4476	15.16	2907	14.77
132	88573	59049	1	.00	1	.00	12826	14.48	8350	14.14
144	265720	177147	13	.00	12	.01	36894	13.88	24068	13.59

ACKNOWLEDGMENT

The author wishes to acknowledge stimulating discussions with J. J. Wedel and A. D. Weeks. Valuable assistance of A. D. Weeks in the preparation of the tables in the Appendix is also gratefully acknowledged. The programs for the tabulation of misses and miss percentages were written by K. Oslund.