すでに述べたことであるが、デジタル信号では、とびとびの物理量しか使わない。その最も単純な形が「１・０信号」で、２種類の基本信号しか使わない。そのため、通信の途中で雑音などのために少しぐらい信号の形がゆがんでも、受信側で「これは１だろう」、「これは０に違いない」と判定できれば正しい情報を受け取れるし、あらためてきれいな形の信号を送り直すことができる（信号の整形）。

しかし長距離の通信では電気的な「雑音」の影響で、信号の形が大きく変わって「１が０になってしまう」とか逆に「０が１に近い信号に化けてしまう」ことも、小さな確率で起こる。そして大量の情報を送信すると、「小さな確率」のことが、現実に多数回起こってしまう____そこで工夫されたのが、「誤りを検出・訂正する」技術である。

7.1　誤りの検出

（1）反復送信

たとえば暗号文

TFOENPSFNPOFZ　（原文は何でしょう？）

を送るとき、安全のために２回繰り返して

TFOENPSFNPOFZ　TFOENPSFNPOFZ

を送信すれば、それが雑音のためにどこかが変わってしまっても、「どこが間違っているか」を検出できることがある。たとえば、

TFOGNPSFNPOFZ　TFOENPSFTPOFZ

が受信されたときは、前半と後半の先頭をそろえて並べてみれば、いくつかの間違いがわかる。

TFO G NPSF N POFZ

TFO E NPSF T POFZ

この場合は、４文字めと11文字めが合わないので、それぞれの場所の前半か後半の、少なくともどちらかが間違っている。このように「誤りがある」とわかれば____誤りを検出することができれば、「その部分を再度送信してもらう」などの対策がとれる。

〈間違いが検出されない場合〉このやり方では「同じ位置の文字に、同じ誤りが発生した」場合には、その誤りは見逃される。しかし誤りの発生が場所に関係なく「万に一つ」（確率$p = 0.0001$）なら、同じ位置で誤りが発生する割合は「億に一つ」（$p^2 = 0.00000001$）となるし、「同じ文字に誤る」確率はさらに小さい。

（2）奇偶検査（パリティ・チェック）

用語　偶数：0, 2, 4, 6,……など、２で割り切れる数（０も偶数である）。

奇数：1, 3, 5, 7,……など、２で割り切れない数。２で割ると１余る数。

パリティ（parity）：情報関係では「奇数か偶数か」を意味し、「奇偶性」と訳される。

JIS符号では、１文字を表すには１バイト＝８ビット、すなわち８個の０・１が使われる。これにもう１ビットを次のようにつけ加えて、送・受信に使うことがある。

１文字を表す０・１の組み合わせの中に、１が必ず偶数個である

ようにする。

例　JIS符号系では、文字T，F，Pの符号を次のように定めている。

T：01010100　　F：01000110　　P：01010000

これらの末尾に「1が偶数個になるように」0か1をつけ加えると、次のようになる。

T：010101001（1が4個）、F：010001101（1が4個）、

P：010100000（1が2個）

つけ加えられたビット（太字で示す）を、**パリティ・ビット**という。またこのように「1の個数を偶数個にする」方式を、偶数パリティ方式という。

注意　「奇数パリティ」（1の数を奇数個にする）方式も考えられるが、ここでは偶数パリティ方式に限って話をすすめる。

文字列を送信するとき、本来の符号だけでなく「文字ごとにパリティ・ビットをつけ加えた符号」を送ると、受信する側で、「文字ごとの1の数が偶数かどうか」を検査すること（奇偶検査、パリティ・チェック）によって、ある確率で誤りが検出できる。

問1　次の符号に、パリティ・ビットをつけ加えなさい。

（ア）01011100　　（イ）01000101　　（ウ）01001011

問2　パリティ・ビットを含む、次の信号が受信された。これらは正しいか？　ただし「誤りは、1文字9ビットの中で、せいぜい1つ」と仮定する。

（ア）010000001　　（イ）010101000　　（ウ）010001010

（解答は100ページに示す）

応用　タテとヨコにパリティ・ビットをつける：たとえば0・1の列

01010100	010101001
01000110	010001101
01010000	010100000
01001111	010011111
01010110	010101100
	010110111

を送信するとき、前のように「文字ごとにパリティ・ビットをつける」だけでなく、さいごにも「列ごとのパリティ・ビット」をつけて、右枠内のような文字列を送信することがある。

参考　さいごの行は、「**サムチェック**（和の検査）」と呼ばれる。

さいごの「サムチェック」は、次のような規則で作られている：

その列の中の1の個数が、偶数個になる。

サムチェックをも含めて全部の信号を受信し終わってから、行ごとのパリティ（1は偶数個か？）を検査すれば、「1ヶ所だけ間違った符号」が検出できる。さらに列ごとのパリティ（1は偶数個か？）を検査すれば、「1ヶ所だけ間違った列」が検出できる。ある符号で2ヶ所が間違った場合、「符号の検査」では見落とされるが、サムチェックで検出できる場合がある。ただし「誤りの検出」だけで、どの場所で誤ったかは特定できない（送信された信号全体を再送信してもらうことになろう）。ただし「誤りが全体で1ヶ所だけ」なら、それが起こった行と列がわかるのだから、間違えた場所が特定でき、