

## 相関回路の作り方

相関値を求める式は厳密には積分計算だが、ロジックで処理する場合は A/D 値と拡散符号との積を求め、それをシグマ ( $\Sigma$ ) して総和を求めることで得られる。

例えば、拡散率 32 ビットの場合、チップ周期間隔の受信 A/D 値 32 個と拡散符号 32 ビットをそれぞれのチップタイミング毎に掛け算して、掛け算結果を 32 個分加算することで求める。

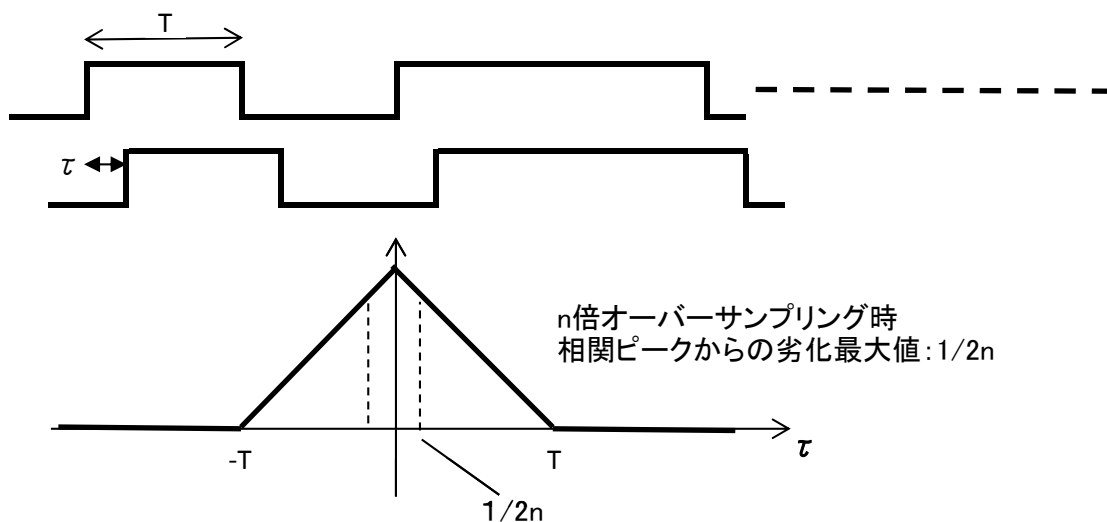
拡散符号は 1 または -1 の値を持つので、掛け算の演算は実際には符号をそのままにするか反転するかビット操作のみになる。そのためわざわざ掛け算回路を設ける必要はない。

送信符号と受信逆拡散符号のタイミングは一致しないのでチップレートよりも速いオーバーサンプリングで A/D 値をサンプリングして時間経過とともに徐々にタイミングをずらしながら相関検出を行っていく。

相関検出出力は、時間経過とともにピークが発生し、ピークの発生する間隔がビット周期になり、ピークが正のときが +1 のデータ (1)、負のときが -1 のデータ (0) になる。

相関出力のビット数は A/D 値のビット数よりも大きな値となる。その後の処理で使用するビット数は、これらの中から必要なビット数を抜き出して仕様することとなる。

# 相関検出



相関値 
$$R \frac{1}{T_o} \int_{-T_o/2}^{T_o/2} v_1(t) \cdot v_2(t+\tau) dt$$

$T_o$ : 拡散符号長  
 $T_o = NT$

拡散符号長  $N$  の相関 
$$\sum_{k=1}^N v_1(kT) \cdot v_2(kT)$$

$T$ : 拡散符号ビット周期

$v_1(t)$  を受信データ、 $v_2(t)$  を拡散符号とすれば、  
 $v_2(kT)$  は符号1のときは+1、符号0のときは-1を掛け算することになる。

|    |    |    |    |    |    |
|----|----|----|----|----|----|
| 0  | 1  | 0  | 0  | 1  | 1  |
| -1 | +1 | -1 | -1 | +1 | +1 |

