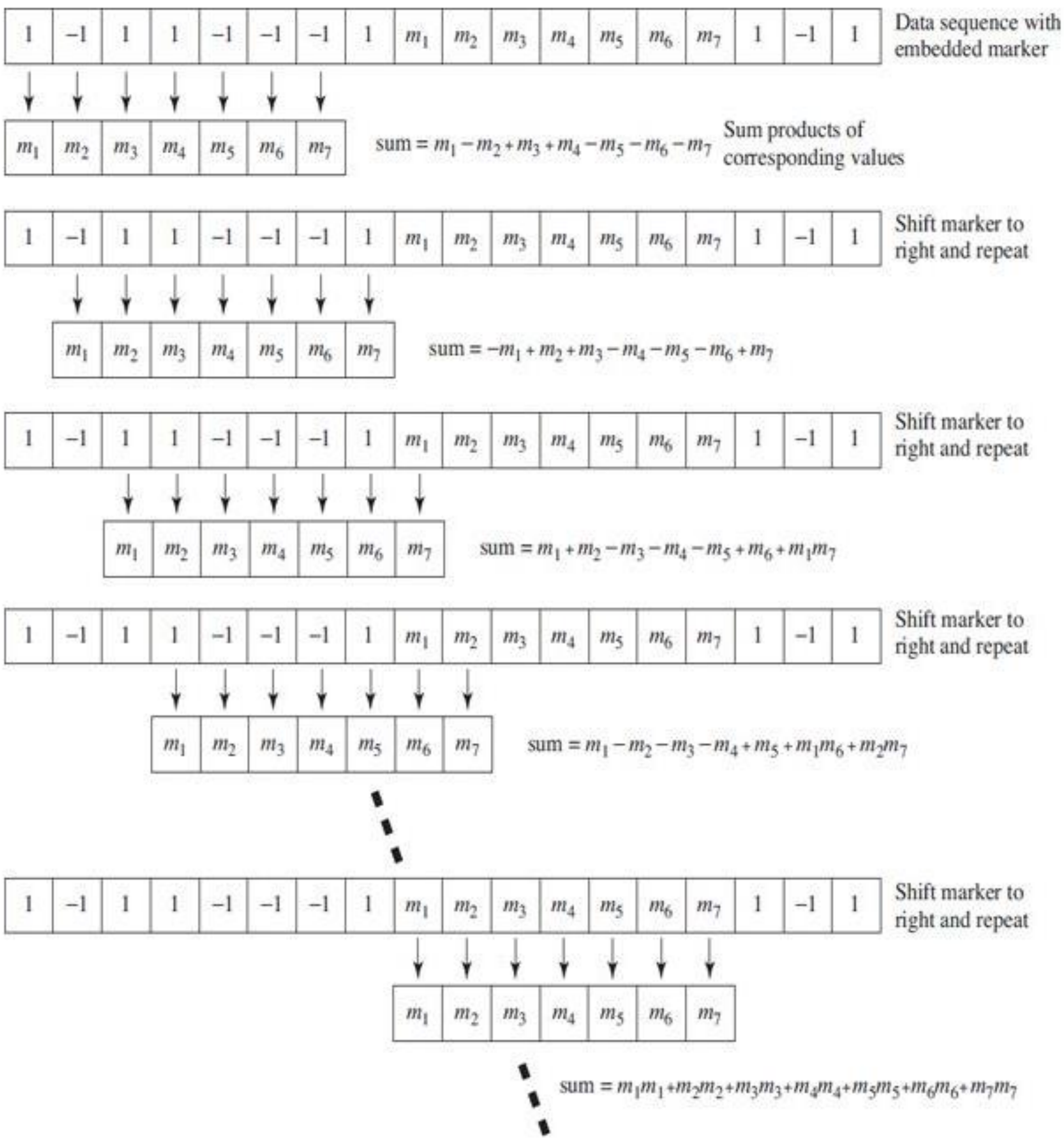
Frame synchronization

The idea of frame synchronization is to identify the byte boundaries in a stream of data $c_n = \pm 1$

Consider the example stream in the figure below. A known marker sequence $M = [m_1, m_2, m_3, m_4, m_5, m_6, m_7]$ is inserted in the data stream c_n , where each of the 7 bits in the marker sequence is ± 1.
To find the location of the marker sequence in the data stream c_n , we do cross-correlation of c_n with the marker sequence to obtain the sync output $f_n = \sum_{k=1}^{7} m_k c_{n+k}$



We then look for the maximum value of f_n which is 7 for this example, assuming no errors in the data stream c_n

The diagram look similar to a digital filter with coefficients $M = [m_1, m_2, m_3, m_4, m_5, m_6, m_7]$
However, in this case the marker sequence is shown as sliding to the right, and the data sequence is fixed, or equivalently, the marker sequence fixed and data sliding from right to left.
For a digital filter, we would typically show the filter coefficients as fixed, and the data sliding in from the left to right, and the cross-correlation becomes a convolution (with c_{n-k} instead of c_{n+k})

If there are some errors in the data stream, then the maximum value of f_n may be less than 7. To allow for such errors, we may set the sync threshold to 6 instead of 7 and declare correct frame sync when $f_n = 6$. However, we want to choose the marker sequence so that the probability of getting $f_n = 6$ when the marker is not correctly synchronized (i.e offset by one or more bits) is low.

A marker sequence of length 7 is too short for a practical system because a random data sequence has a probability of one in $2^7 = 128$ to match the marker sequence exactly, and we would get a false frame sync quite often.

In practical wireless systems where signals may be weak, a longer frame sequence is used to minimize the probability of incorrect frame sync or false frame sync in noise. The FLEX pager system uses a marker of length 32.