Coding Theory

(Convolutional and Turbo codes)

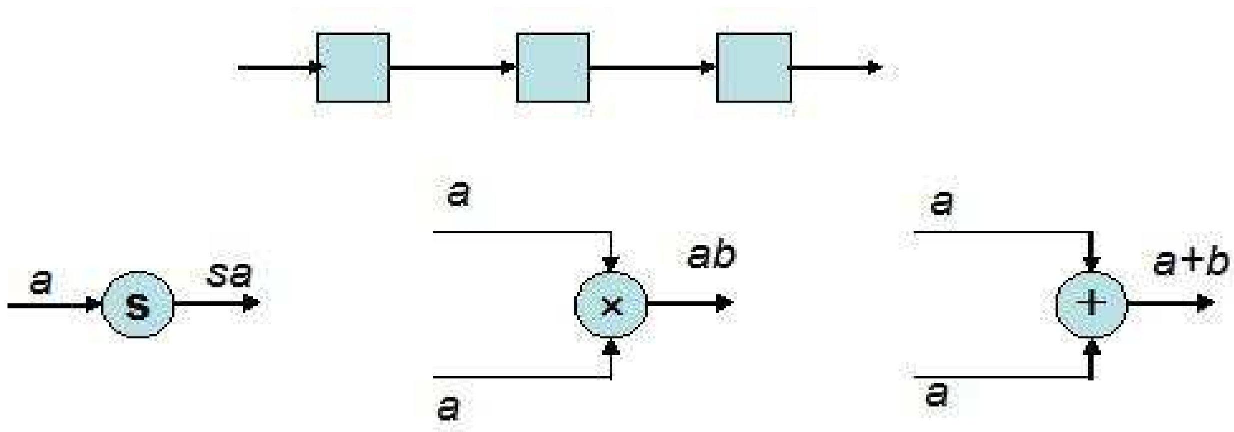
Lector: Nikolai L. Manev

Institute of Mathematics and Informatics, Sofia, Bulgaria

CHONBUK NATIONAL UNIVERSITY

Shift registers

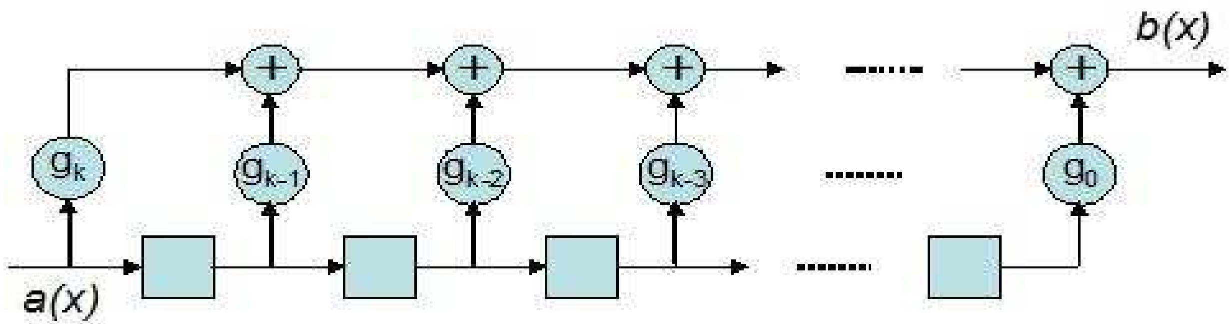
Shift registers are bank of memory units which are capable of shifting the contents of one unit to the next at every clock pulse.



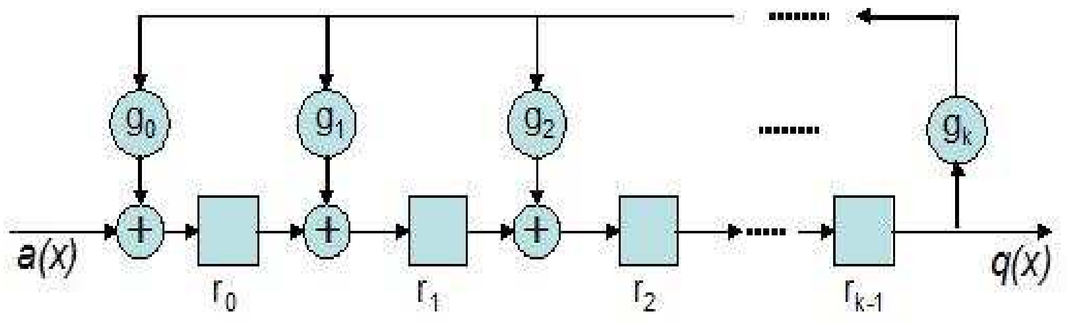
The basic elements of a shift register.

Scaler: $a \to sa$;
Adder: $(a, b) \to a + b$;
Multiplier: $(a, b) \to ab$.

The shift register presented below realizes the multiplication of an arbitrary polynomial $a(x)$ by the polynomial $g(x) = g_k x^k + g_{k-1}x^{k-1} + g_{k-2}x^{k-2} + \cdots + g_1 x + g_0$.



Finite Impulse Response (FIR) Filter.



Dividing circuit.

The above shift register realizes the division of an arbitrary polynomial $a(x) = a_n x^n + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$ by the polynomial
$g(x) = g_k x^k + g_{k-1} x^{k-1} + g_{k-2} x^{k-2} + \cdots + g_1 x + g_0$. Initial states of the cells are zeros. First enters the coefficient a_n, then a_{n-1}, and so on. In the first k shifts the register is fill out, and after the next $n-k+1$ pulse clocks the quotient is passed out and the content of the register is the reminder.

Convolutional codes

The convolutional encoder manipulates the input information symbols in blocks of length k, but the resulted block depend on the previously transmitted ones. More precisely, the encoder transforms the input sequence of blocks of length k, $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots,$ into the sequence of block of length n: $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots,$ where

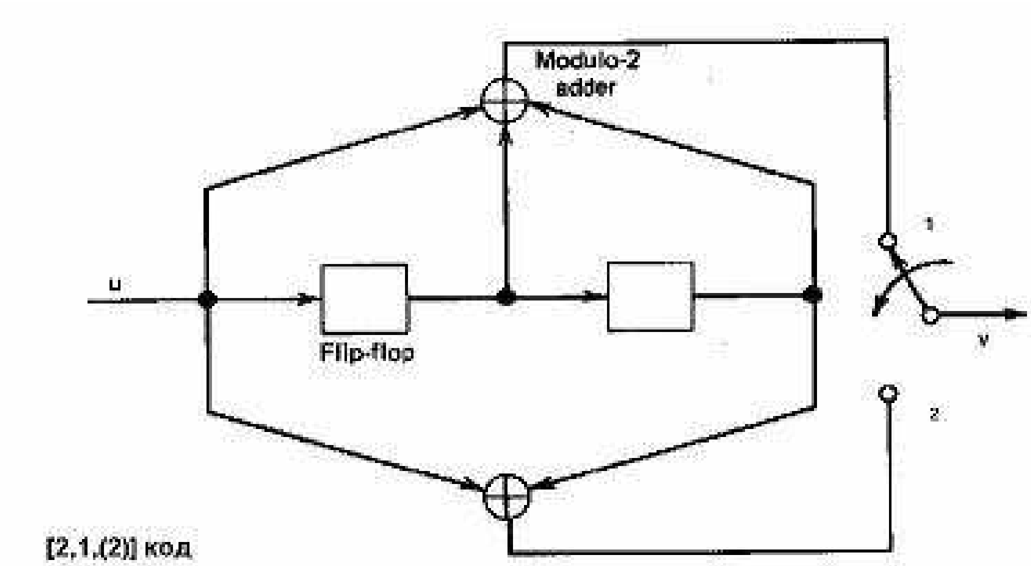
$$\mathbf{v}_t = l(\mathbf{u}_t, \mathbf{u}_{t-1}, \ldots, \mathbf{u}_{t-m})$$

In this case we say that the encoder realizes an $[n, k, (m)]$ code. The number m is called *memory*, and $L = m + 1$ *constrain length* of the code.
Indeed the memory m is the number of the k-bits cells of the shift register realizes the convolutional $[n, k, (m)]$ code.

Example. Consider a $[2, 1, (2)]$ code, that is $n = 2,\ k = 1$ and $L = 3$. The encoder transforms the input sequence of bits, $u_0, u_1, u_2, \ldots,$ into $(v_{10}, v_{20}), (v_{11}, v_{21}), (v_{12}, v_{22}), \ldots$, where

Convolutional codes

$$v_{1t} = u_t + u_{t-1} + u_{t-2}; \quad v_{2t} = u_t + u_{t-2}.$$



The output consists of two sequences:

$$v^{(1)} = v_{10}, v_{11}, v_{12}, \ldots,$$

$$v^{(2)} = v_{20}, v_{21}, v_{22}, \ldots$$

The end of the part

Thank You for Attention!