

ЛЕКЦИЯ 2-2

ОПЕРАЦИОННЫЙ АВТОМАТ ТИПА I

ПОНЯТИЯ ОБ ЭКВИВАЛЕНТНЫХ МИКРООПЕРАЦИЯХ И ОБОБЩЕННОМ ОПЕРАТОРЕ

В микропрограммах могут встречаться микрооперации, вычисляющие значения слов с использованием одной и той же функции, применяемой к различным наборам слов, например,.

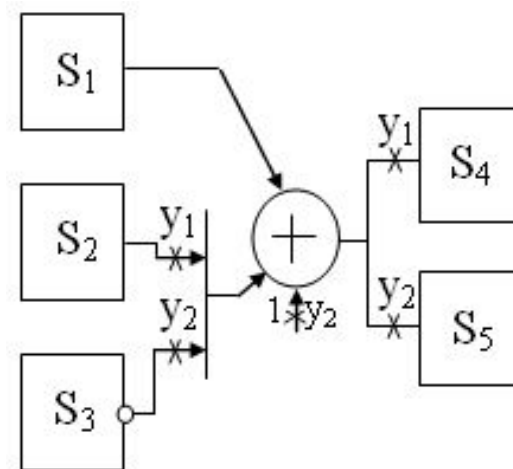
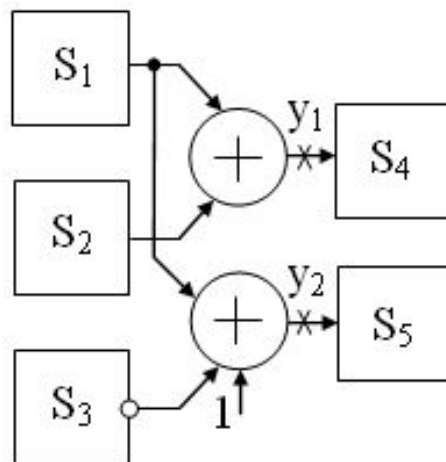
$$S_{\alpha} := \Upsilon_m(S_j)$$

$$S_{\beta} := \varphi \Upsilon_m(S_{\theta})$$

Пример:

$$y_1 : S_4 := S_1 + S_2$$

$$y_2 : S_5 := S_1 + \overline{S_3} + 1$$



Одна комбинационная схема может использоваться для выполнения нескольких микроопераций, эквивалентных в смысле реализованных ими функций. За счёт этого могут быть уменьшены затраты оборудования, но может исключаться совместимость микроопераций, что приводит к дополнительным затратам времени.

Процедура минимизации числа комбинационных схем базируется на понятии эквивалентных микроопераций и обобщенного оператора.

Микрооперации считаются эквивалентными, когда операторы содержат одну и ту же функцию. Эквивалентность микрооперации означает, что для вычисления двоичных выражений, соответствующим микрооперациям, может использоваться одна и та же комбинационная схема.

Эквивалентность устанавливается следующим образом:

Двоичные выражения $S_{\alpha_1} * S_{\alpha_2} * \dots * S_{\alpha_k}$ называются эквивалентными, $S_{\beta_1} * S_{\beta_2} * \dots * S_{\beta_n}$

если одно из них может быть приведено к другому путем:

1. Замены одного слова другим либо его инверсией.
2. Замены слова константой.
3. Равносильными преобразованиями выражений.

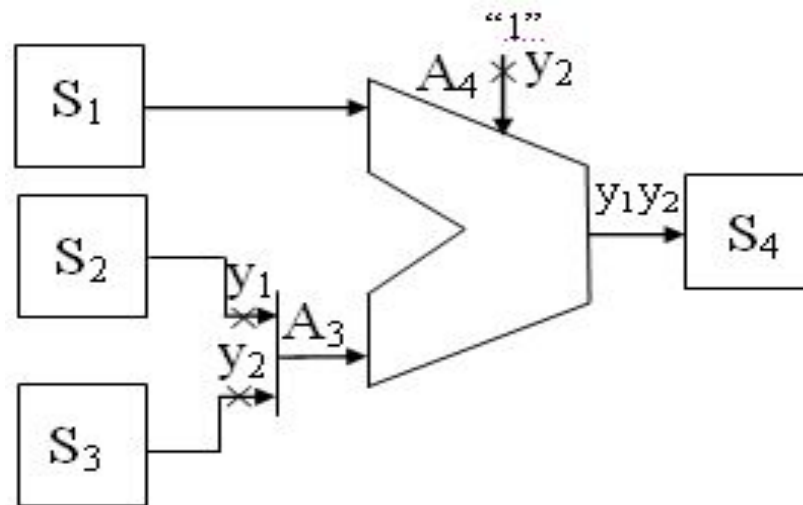
Для построения структуры реализации совокупности эквивалентных микроопераций вводится специальная форма представления – **обобщенный оператор**.

$S_j := A_{\alpha_1} * A_{\alpha_2} * \dots * A_{\alpha_m}$ где A_{α_i} вспомогательные переменные

$$\begin{cases} y_1 : S_4 := S_1 + S_2 \\ y_2 : S_4 := S_1 + \overline{S_3} + 1 \end{cases} \Rightarrow S_4 := S_1 + A_3 + A_4$$

$$A_3 = \begin{cases} S_2, y_1 = 1 \\ \overline{S_3}, y_2 = 1 \end{cases}$$

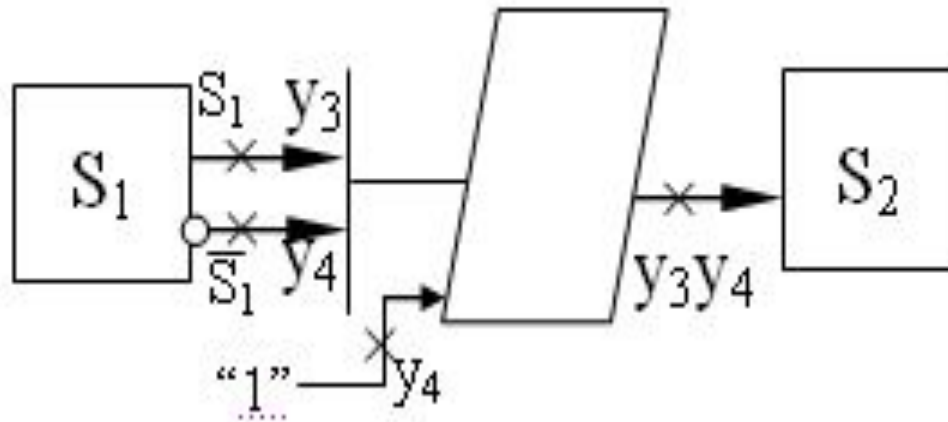
$$A_4 = \begin{cases} 1, y_2 = 1 \\ 0, \text{otherwise} \end{cases}$$



$$\begin{cases} y_3 : S_2 := R_1(0.S_1) \\ y_4 : S_2 := R_1(1.\bar{S}_1) \end{cases} \Rightarrow S_2 := R_1(A_1 A_2)$$

$$A_1 = \begin{cases} 0, y_3 = 1 \\ 1, y_4 = 1 \end{cases}$$

$$A_2 = \begin{cases} S_1, y_3 = 1 \\ \bar{S}_1, y_4 = 1 \end{cases}$$



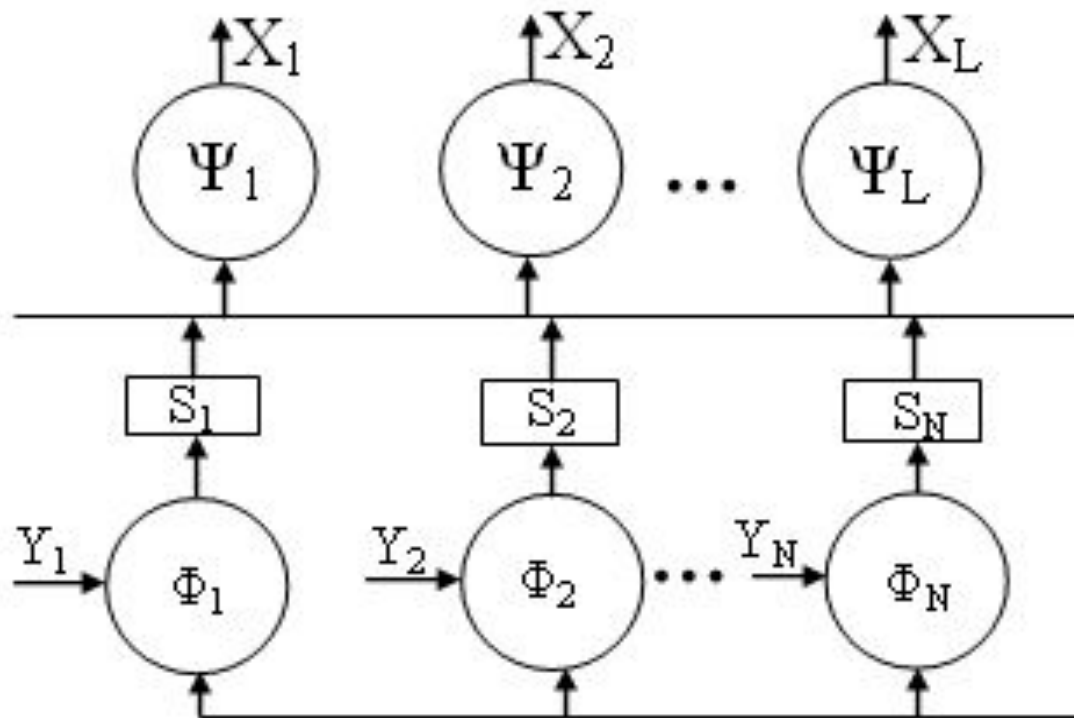
ОПЕРАЦИОННЫЕ АВТОМАТЫ ТИПА I

Определим структуру ОА, производительность которого не ниже производительности канонического ОА, а затраты оборудования меньше. Это может быть обеспечено в случае, если синтезируемая структура не будет вносить ограничений на совместимость микроопераций.

Это может быть выполнено тогда, когда каждая комбинационная схема используется для выполнения эквивалентных микроопераций, связанных с вычислением значений одного слова.

Следовательно, для минимизации аппаратных средств необходимо обобщать комбинационные схемы для выполнения нескольких микроопераций, которые принадлежат одному подмножеству микроопераций, вычисляющему одно слово.

ОА, структура которых обеспечивает возможность одновременного выполнения всех функционально совместимых микроопераций при использовании минимально возможного числа комбинационных схем, называются **I-автоматами**. Особенностью **I-автоматов** является то, что каждый регистр обслуживается своей комбинационной схемой.



ПРОЕКТИРОВАНИЕ I-АВТОМАТОВ

Сводится к преобразованию заданного набора микроопераций в совокупность обобщенных операторов и выполняется следующим образом:

1. Составляется список микроопераций и логических условий:

$$y_1 : A := D_1$$

$$y_2 : B := D_2$$

$$y_3 : C := 0$$

$$y_4 : CnT := n$$

$$y_5 : C := C + A$$

$$y_6 : TS := B(0)$$

$$y_7 : B := R_1(C(0).B)$$

$$y_8 : CnT := CnT - 1$$

$$y_9 : C := R_1(C(n-1)C)$$

$$y_{10} := R_1(CFC)$$

$$y_{11} : C := C + \overline{A} + 1$$

$$y_{12} : R := C.B$$

$$x_1 : B(0)$$

$$x_2 : OF$$

$$x_3 : CnT = 0$$

$$x_4 : TS$$

2. Выбирается множество регистров и определяется их разрядность. Используются следующие регистры:

$$A(n-1:0); B(n-1:0); C(n-1:0);$$

$$CnT(m-1:0); CF, OF, TS.$$

3. Всё множество микроопераций разбивается на подмножества, соответствующие внутренним словам (регистрам):

$$Y_A = \{y_1\}$$

$$Y_B = \{y_2 y_7\}$$

$$Y_C = \{y_3 y_5 y_9 y_{10} y_{11}\}$$

$$Y_{CnT} = \{y_4 y_8\}$$

$$Y_{TS} = \{y_6\}$$

4. На выделенных подмножествах определяются классы эквивалентных микроопераций.

$$K_A = \{y_1\}, K_{B_1} = \{y_2\}, K_{B_2} = \{y_7\}, K_{C_1} = y_3, K_{C_2} = \{y_5 y_{11}\}, \\ K_{C_3} = \{y_9 y_{10}\}, K_{CnT_1} = \{y_4\}, K_{CnT_2} = \{y_8\}, K_{TS} = \{y_6\}.$$

5. Для каждого класса, содержащего не менее 2-х микроопераций, находится обобщенный оператор:

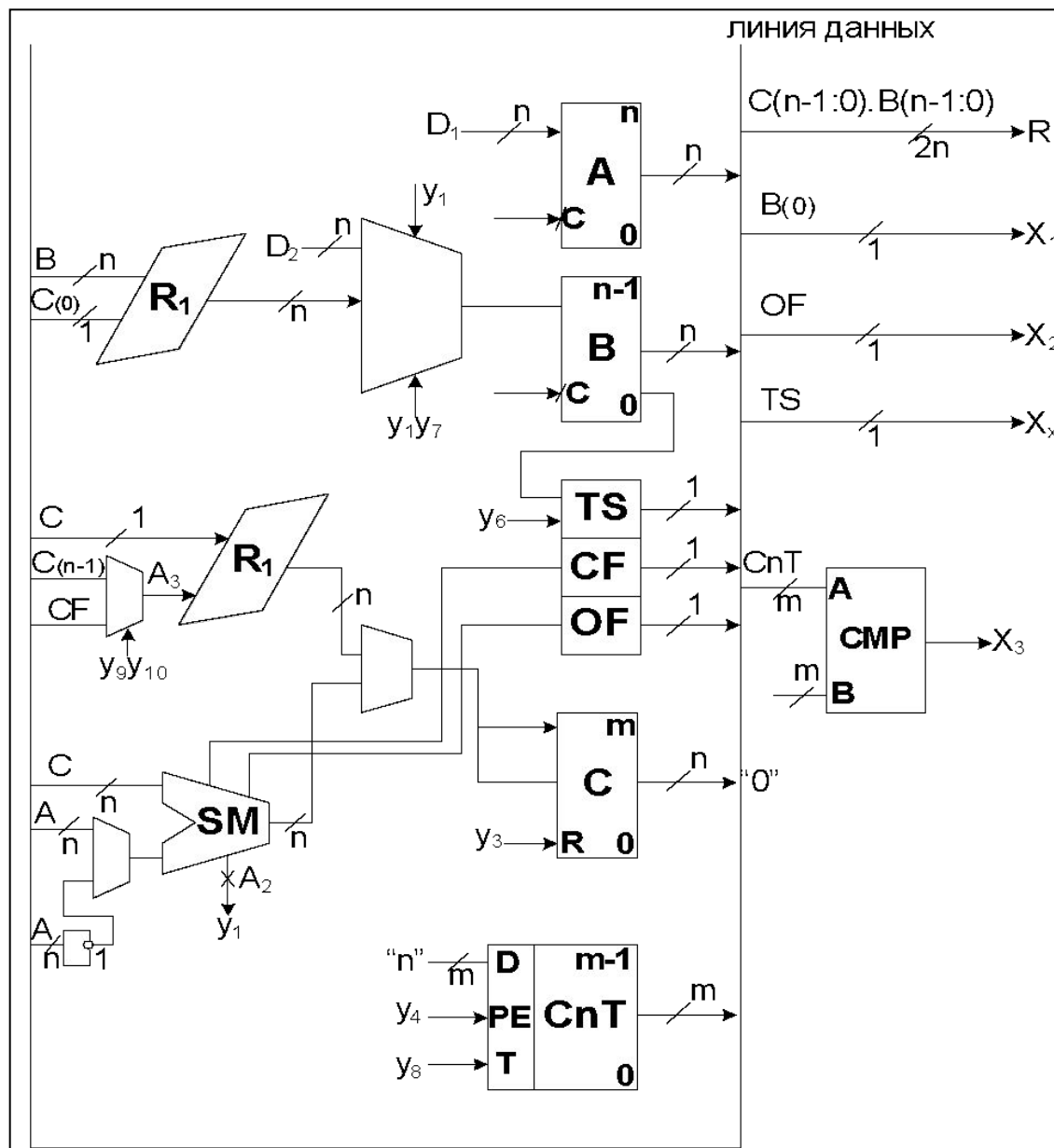
$$K_{C_2} = \begin{cases} y_5: C := C + A \\ y_{11} C: C + \bar{A} + 1 \end{cases} \quad C := C + A_1 + A_2$$

$$A_1 = \begin{cases} A_1 y_5 = 1 \\ A_1 y_{11} = 1 \end{cases}, \quad A_2 = \begin{cases} 1, y_{11} = 1 \\ 0, \hat{a} \hat{i} \hat{n} \hat{o} \hat{a} \hat{e} \hat{u} \hat{i} \hat{u} \hat{o} \hat{n} \hat{e} \hat{o} \hat{a} \hat{y} \hat{o} \end{cases}$$

$$K_{C_3} = \begin{cases} y_9: C := R_1(C(n-1).C) \\ y_{10}: C := R_1(CF.C) \end{cases} \Rightarrow C = R_1(A_3.C),$$

$$A_3 = \begin{cases} C(n-1).y_9 = 1 \\ CF, y_{10} = 1 \end{cases}$$

6. Строится структурно-функциональная схема ОА.



Контрольные вопросы

1. Эквивалентные микрооперации и форма их представления.
2. Структурная организация и порядок проектирования операционного автомата типа I.
3. Основные характеристики операционного автомата типа I.