



# Архитектура ЭВМ

Лекции

# Логические основы ЭВМ

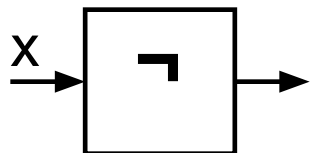
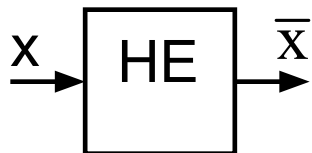
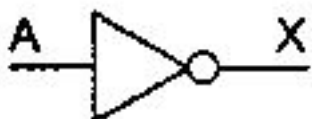
- Базовые функциональные элементы ЭВМ (вентили)
- Основные логические схемы

# Вентили

- *Логический элемент* - электронное устройство, на входах и выходах которого сигнал может иметь только один из двух дискретных уровней напряжения: низкий ( $0-2V$ ) или высокий ( $3-5V$ ).
- *Базовым* считают элемент с наиболее простой структурой, на основе которого легче всего создавать другие электронные схемы.
- И, ИЛИ, НЕ
- И-НЕ

# Инвертор (вентиль НЕ)

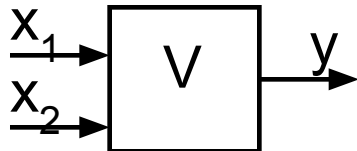
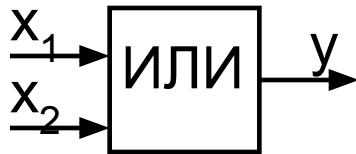
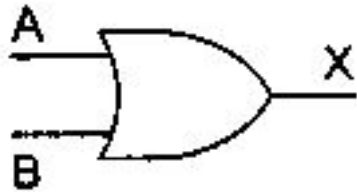
- Обозначения Таблица истинности



$x$	$\bar{x}$
0	1
1	0

# Дизъюнктор (вентиль ИЛИ)

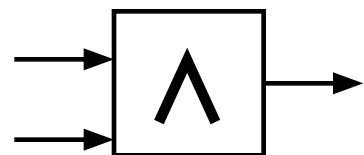
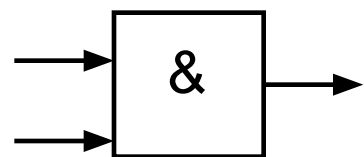
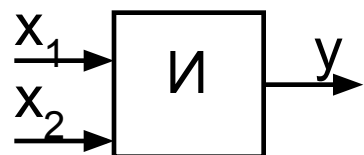
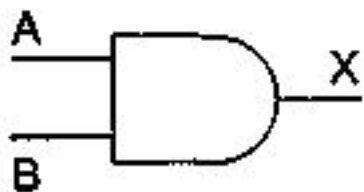
- Обозначения Таблица истинности



$X_1$	$X_2$	$Y$
0	0	0
0	1	1
1	0	1
1	1	1

# Конъюнктор (вентиль И)

- Обозначения Таблица истинности

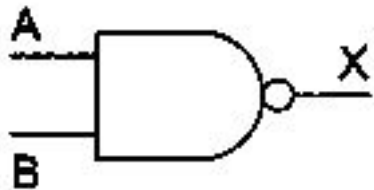


$X_1$	$X_2$	$Y$
0	0	0
0	1	0
1	0	0
1	1	1

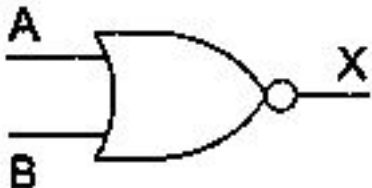
# Вентили НЕ-И и НЕ-ИЛИ

- Обозначение Таблица истинности

НЕ-И



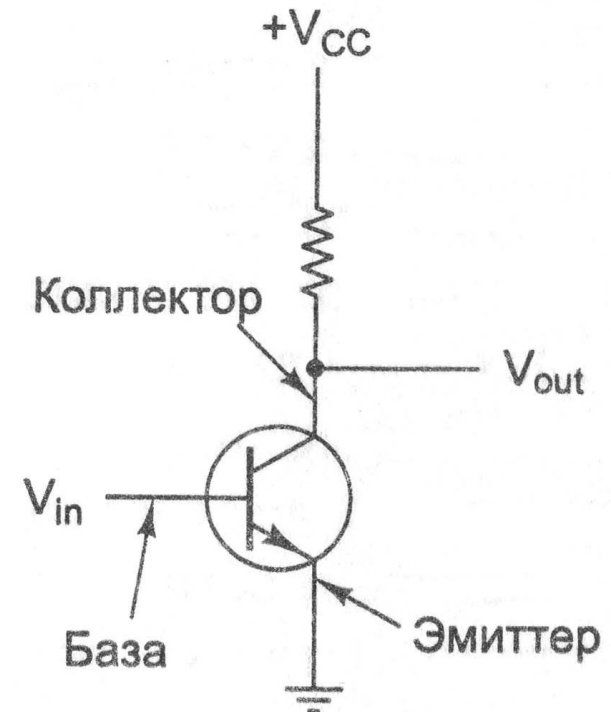
НЕ-ИЛИ



$X_1$	$X_2$	НЕ-И	НЕ-ИЛИ
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

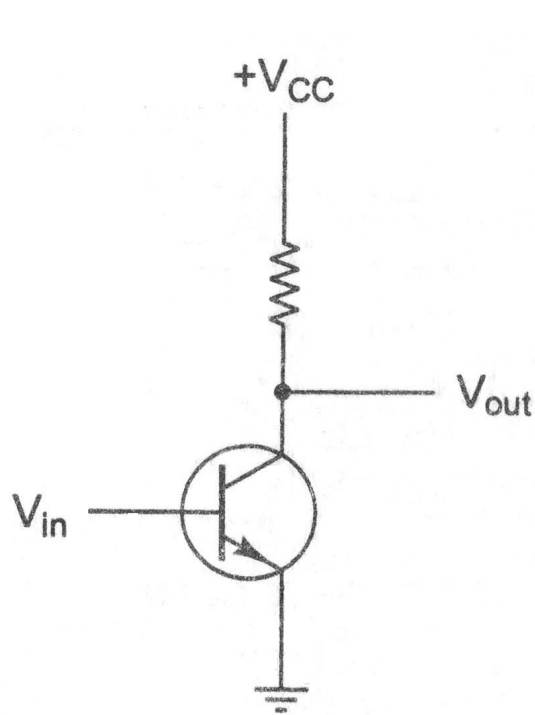
# Уровень физических устройств

- **Транзистор** имеет три соединения с внешним миром: коллектор, базу и эмиттер.
- Если входное напряжение  $V_{in}$  низкое, то транзистор выключается и действует как очень большое сопротивление. Это приводит к выходному сигналу  $V_{out}$ , близкому к  $V_{cc}$  (обычно +5В).
- Если  $V_{in}$  высокое, то транзистор включается и действует как провод, вызывая заземление сигнала  $V_{out}$  (по соглашению 0 В).
- На переключение с одного состояния на другое обычно требуется несколько наносекунд.

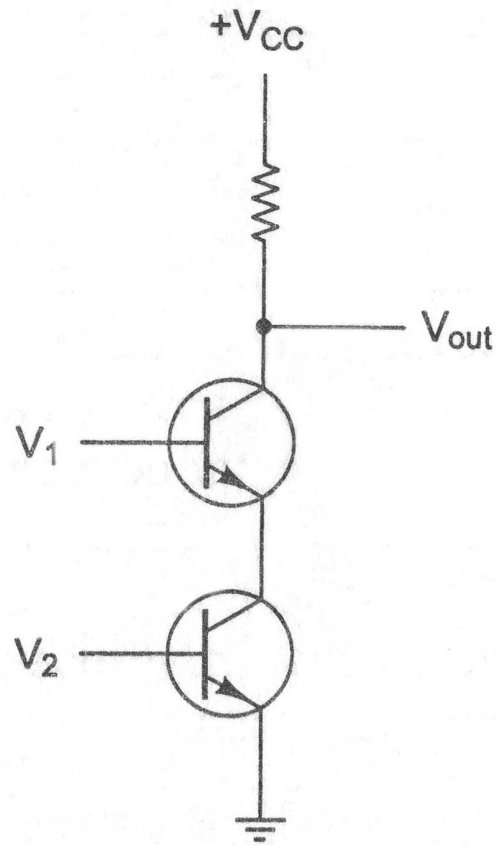




# Техническая схема вентиляей

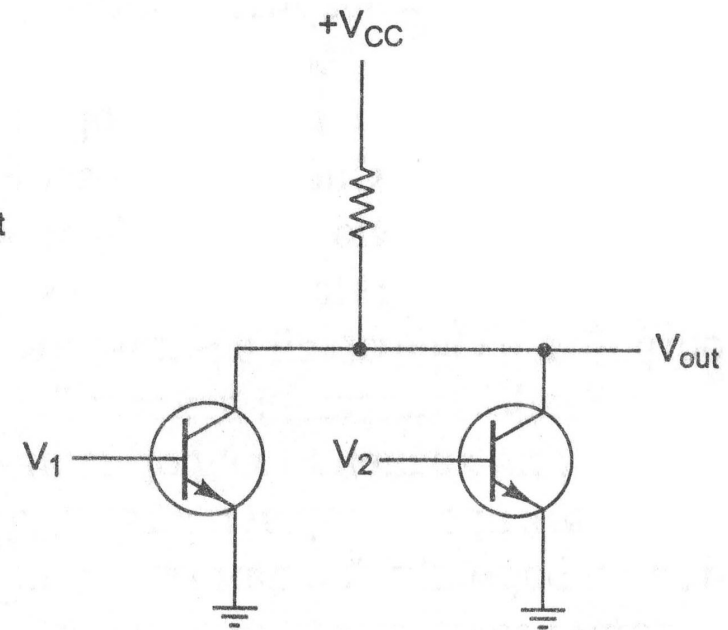


HE



HE-И

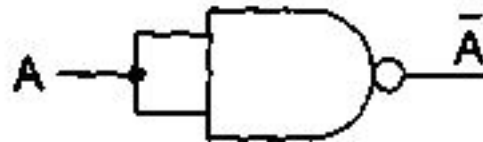
**И, ИЛИ - ?**



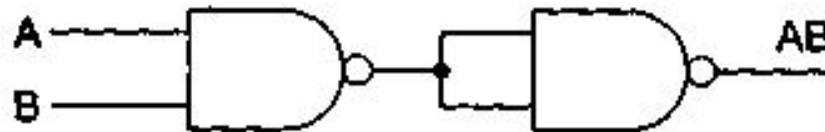
HE-ИЛИ

# Конструирование вентиляей НЕ, И, ИЛИ с использованием только НЕ-И

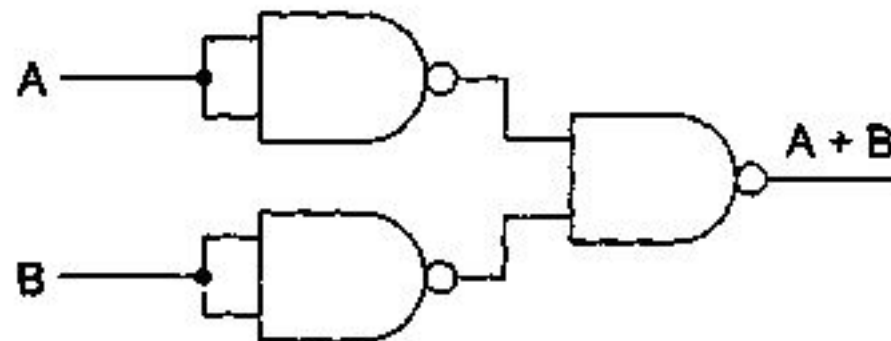
■ НЕ



■ И



■ ИЛИ

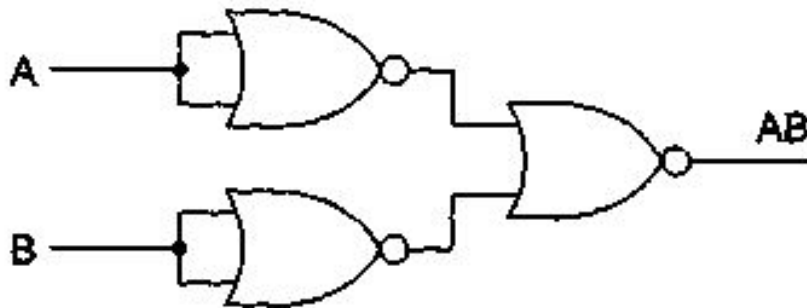


# Конструирование вентиляей НЕ, И, ИЛИ с использованием только НЕ-ИЛИ

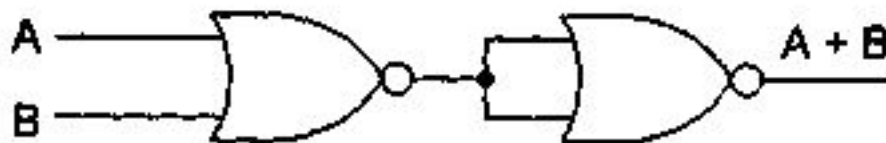
■ НЕ



■ И



■ ИЛИ



# Логический синтез схем

- Одноразрядный двоичный полусумматор
- Полный одноразрядный сумматор
- Полный многоразрядный сумматор
- Дешифратор
- Триггер

# Как реализовать схему

- Составить таблицу истинности для данной функции
- Обеспечить инверторы, чтобы породить инверсии для каждого входного сигнала.
- Нарисовать вентиль И для каждой строки таблицы истинности с результатом 1.
- Соединить вентили И с соответствующими входными сигналами.
- Вывести выходы всех вентилях И в вентиль ИЛИ.

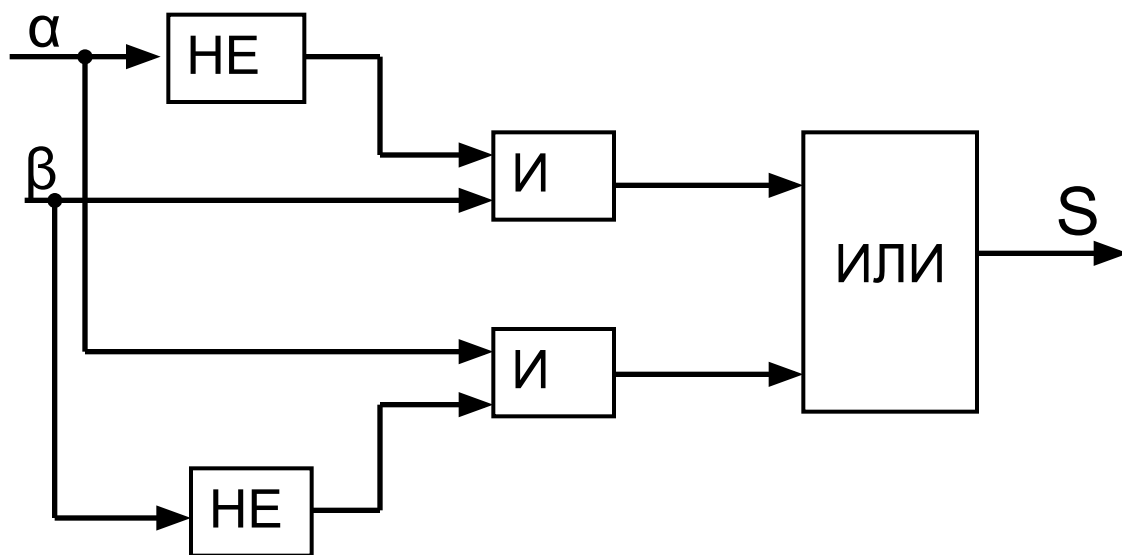
# Одноразрядный двоичный полусумматор

- Пусть  $\alpha$  и  $\beta$  - одноразрядные числа.
- $S = \alpha + \beta$
- Таблица истинности функции  $S$

$\alpha$	$\beta$	$S$
0	0	0
0	1	1
1	0	1
1	1	0

$$S = \bar{\alpha}\beta \vee \alpha\bar{\beta}$$

- Логическая схема



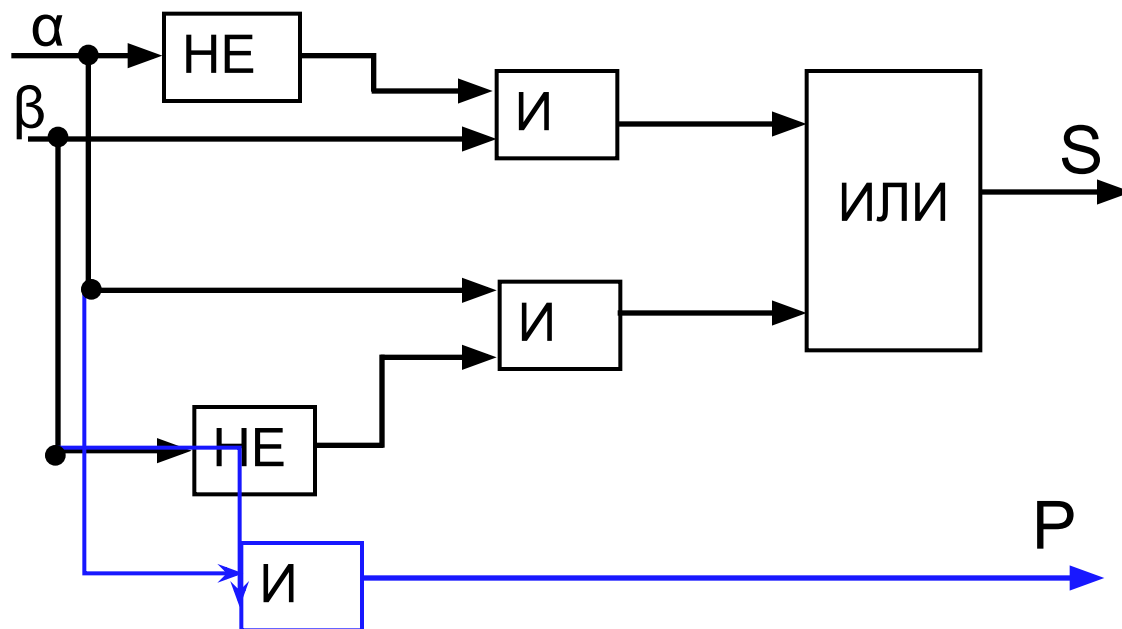
# Одноразрядный двоичный полусумматор (продолжение)

- Добавим функцию  $P$  – цифра переноса в следующий (старший) разряд.
- Таблица истинности функции  $P$

$\alpha$	$\beta$	$P$
0	0	0
0	1	0
1	0	0
1	1	1

$$P = \alpha\beta$$

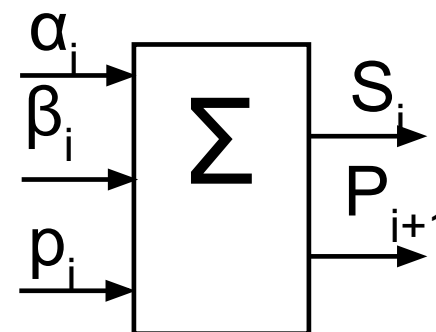
- Логическая схема



# Полный одноразрядный сумматор

- — полусумматор, дополненный третьим входом – значением разряда переноса из соседнего младшего разряда.
- Таблица истинности

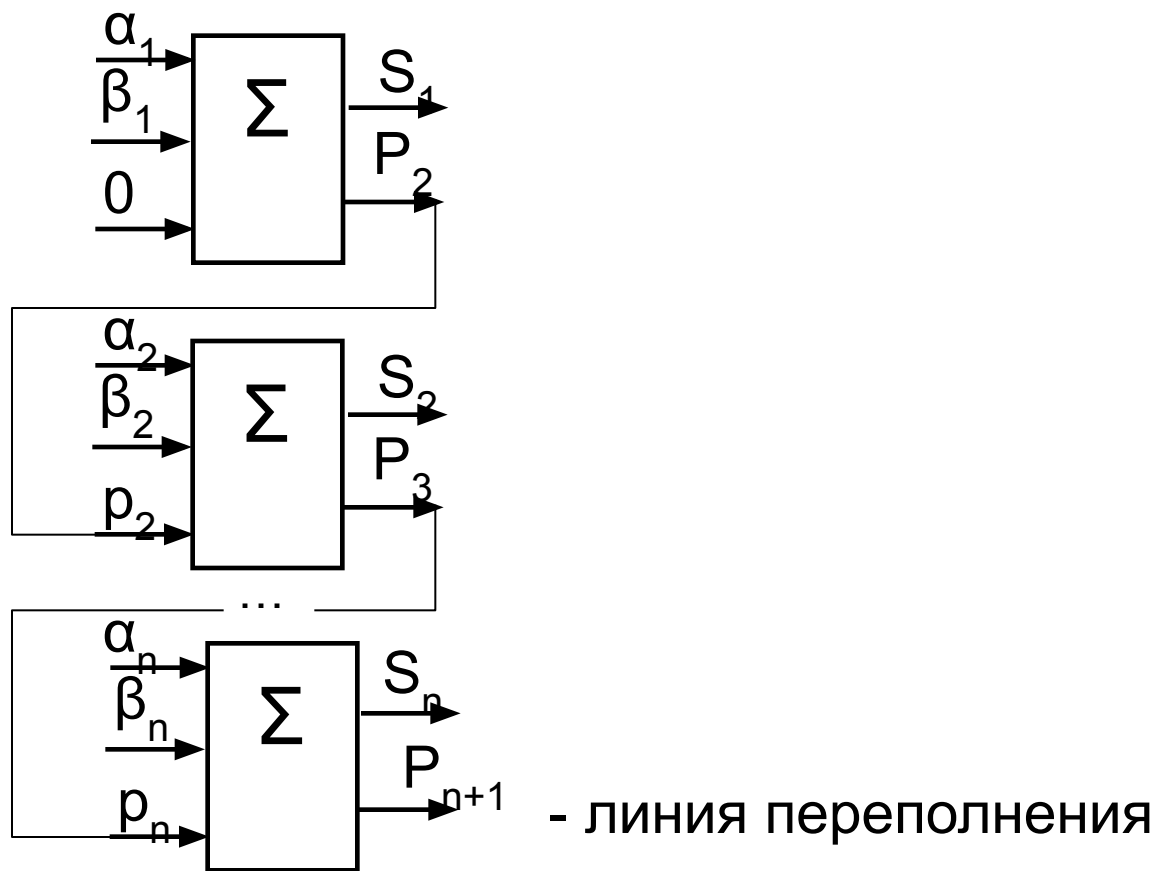
$\alpha_i$	$\beta_i$	$P_i$	$S_i$	$P_{i+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1	1	1





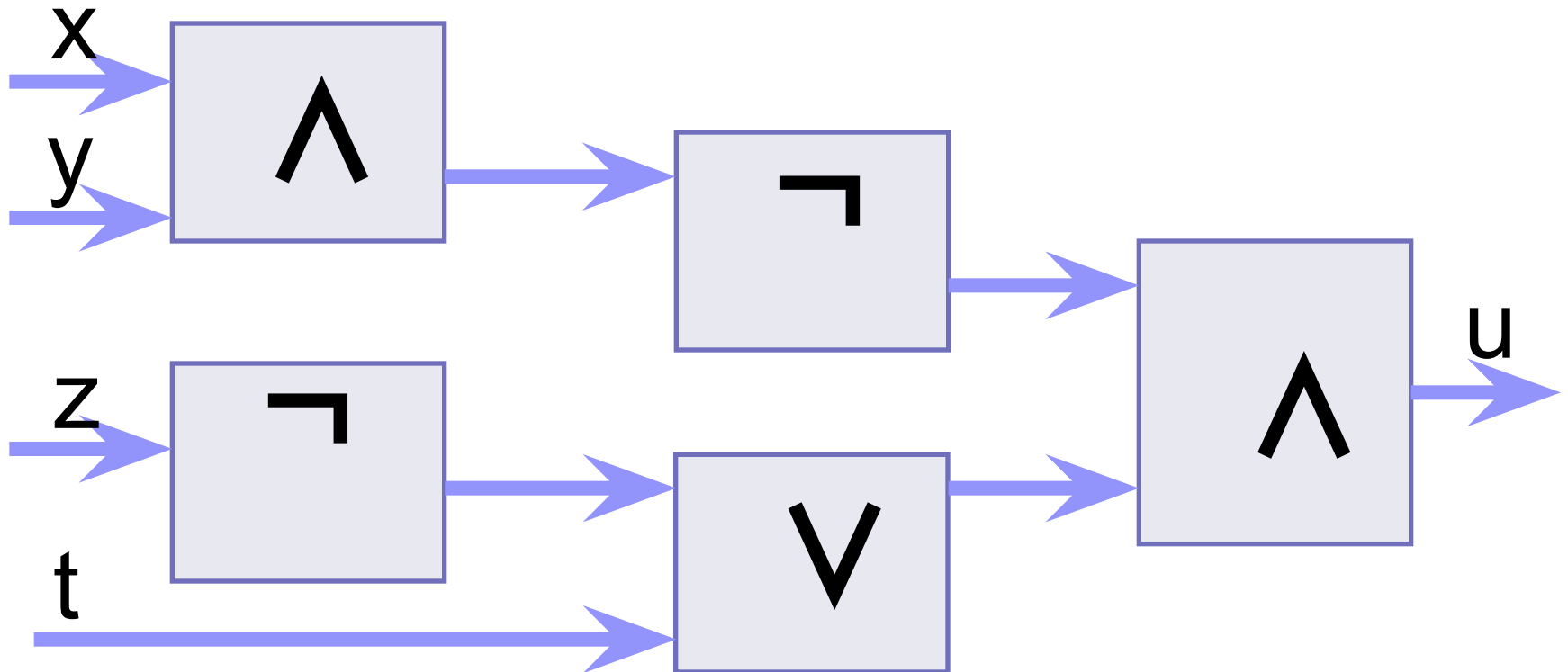
# Полный многоразрядный сумматор

$$\alpha = (\alpha_n \alpha_{n-1} \dots \alpha_2 \alpha_1) \quad \beta = (\beta_n \beta_{n-1} \dots \beta_2 \beta_1)$$



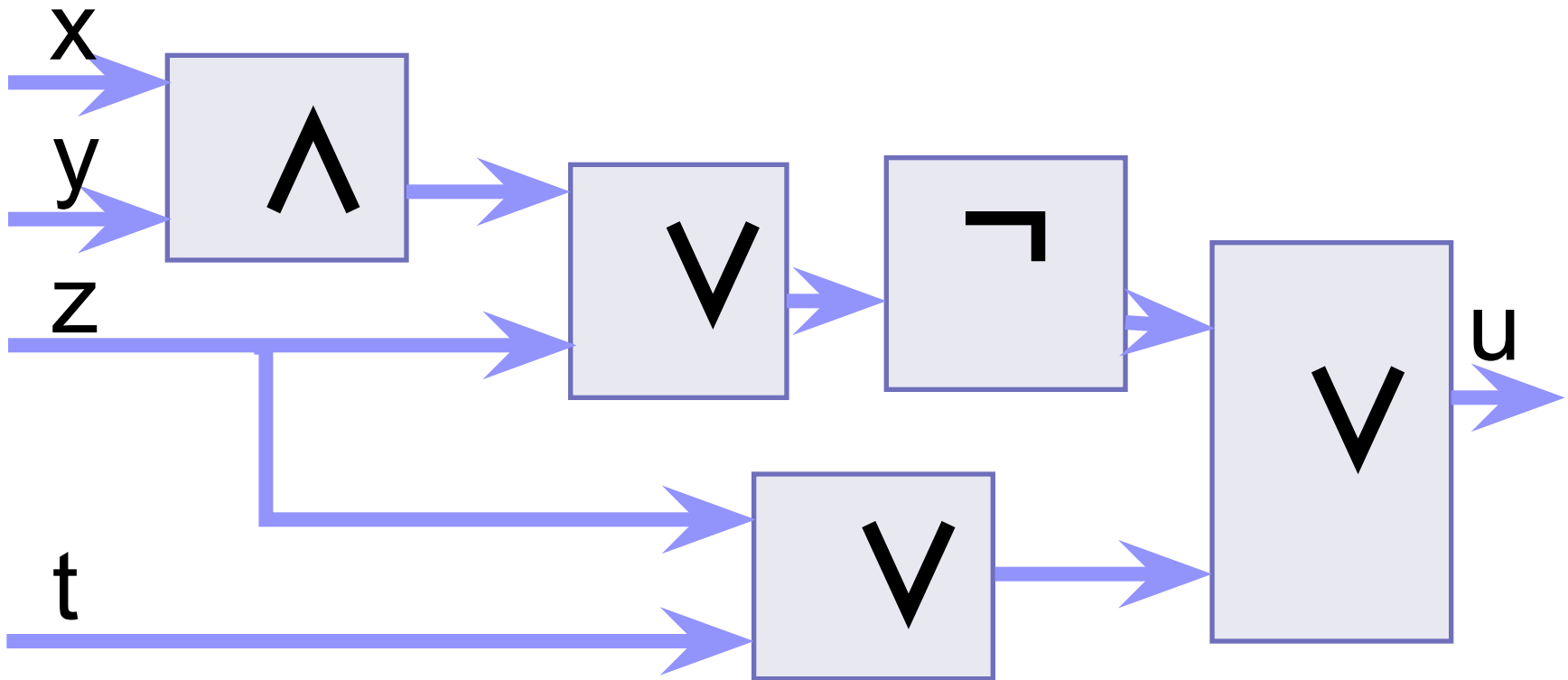
# Задача 1а

- Определить логическую функцию



# Задача 16

- Определить логическую функцию



## Задача 2

- Построить логическую схему для функции:

- а)  $z = \bar{x} \wedge \bar{y} \vee y$

- б)  $u = \overline{x \wedge y} \wedge (\bar{z} \vee t)$

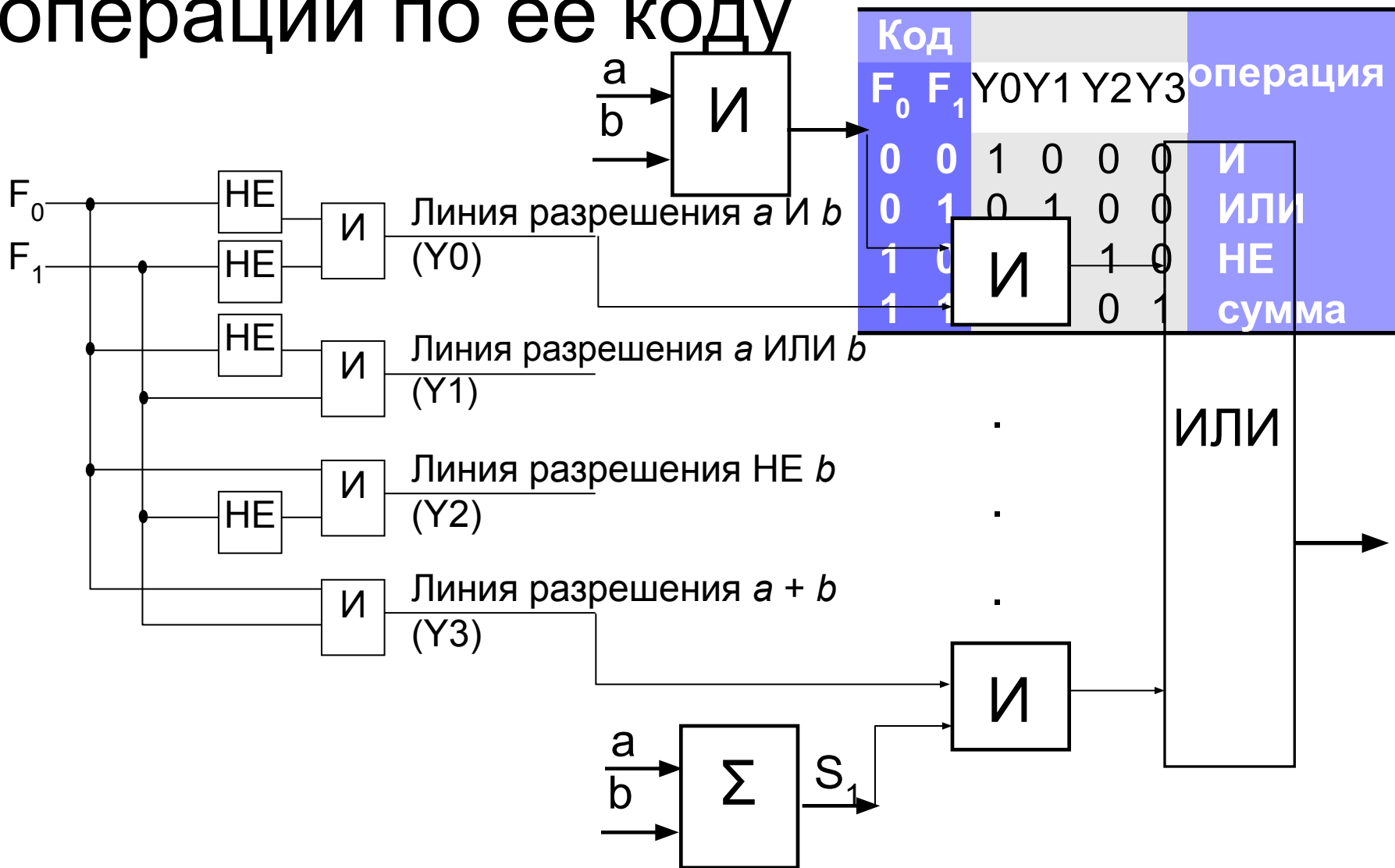
## Задача 2в

- Построить наиболее экономную схему для функции

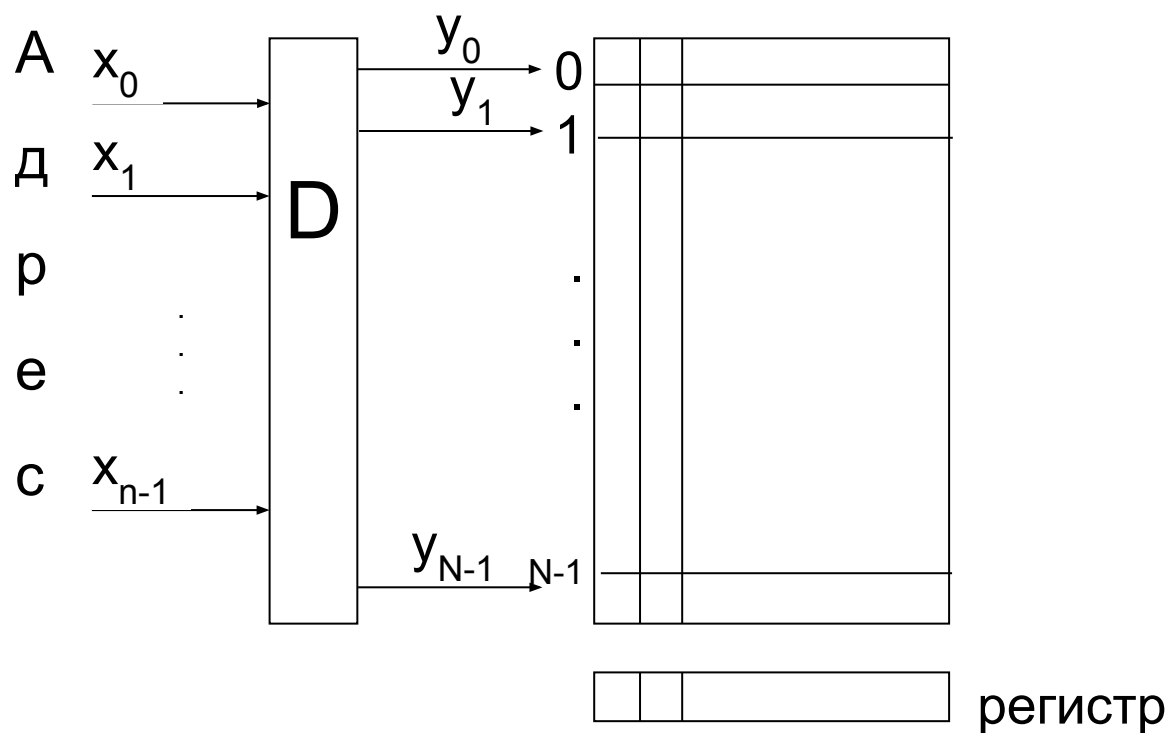
$$z = \bar{x} \wedge \bar{y} \vee \overline{x \vee y} \wedge y \vee y \vee \bar{x}$$



# Дешифратор для выбора операции по ее коду



# Дешифратор для выбора ячейки памяти по ее адресу

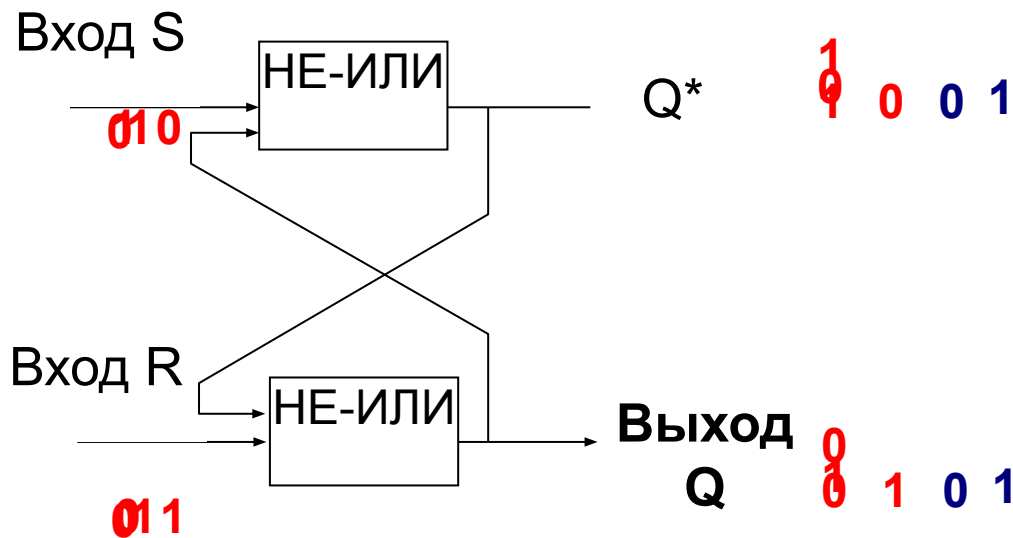




# Определения

- Схема называется **комбинационной** (схема без памяти), если значения переменных на выходе схемы однозначно зависят только от значения входных переменных.
- Схема называется **последовательностной** (схема с памятью), если значения выходов схемы зависят не только от значений на входах, но и от внутреннего состояния (внутренних переменных).
- Ч.С.
  - Схема называется **схемой с обратными связями**, если в качестве внутренних переменных используются значения выходных переменных, полученных в предыдущий момент времени.

# Триггер (SR-защелка)



$x_1$	$x_2$	$x_1$ НЕ-ИЛИ $x_2$
0	0	1
0	1	0
1	0	0
1	1	0

При  $S=0$   $R=0$

Стабильные состояния

а)  $Q=0$   $Q^*=1$

б)  $Q=1$   $Q^*=0$

Нестабильные состояния

а)  $Q=1$   $Q^*=1$

б)  $Q=0$   $Q^*=0$

При  $S=1$   $R=0$

$Q=1$

При  $S=0$   $R=1$

$Q=0$

При  $S=1$   $R=1$

$Q=Q^*=0$

# Регистр

- Регистр – схема для приема, хранения и передачи n-разрядного блока данных.
- Используются для промежуточного хранения, сдвига, преобразования и инверсии данных.
- Выполняются на триггерах и логических элементах.

