

Глава 2 :: Темы

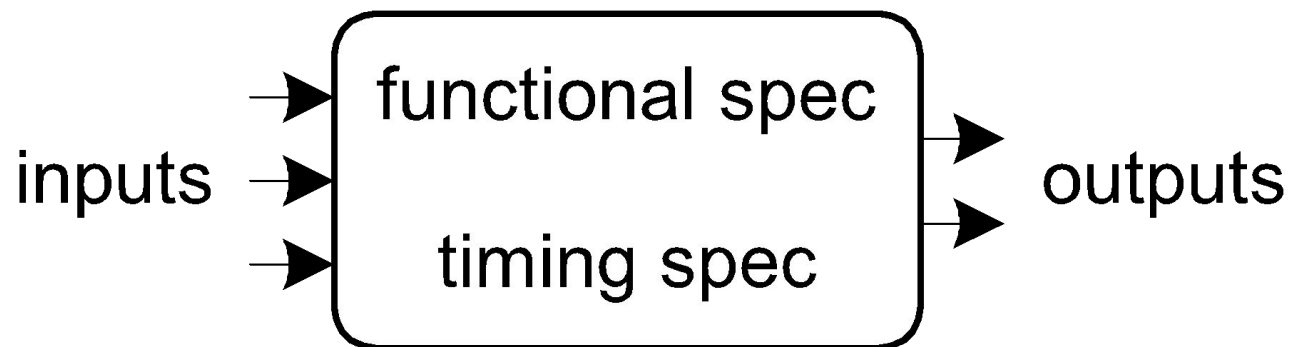
- **Введение**
- **Булевы выражения**
- **Булева алгебра**
- **От логики к логическим элементам**
- **Многоуровневая комбинационная логика**
- **Что за X и Z?**
- **Карты Карно**
- **Базовые комбинационные блоки**
- **Временные характеристик**

Application Software	
Operating Systems	
Architecture	
Micro-architecture	
Logic	
Digital Circuits	
Analog Circuits	
Devices	
Physics	

Введение

Логическая схема состоит из:

- Входов
- Выходов
- Функциональной спецификации
- Временной спецификации



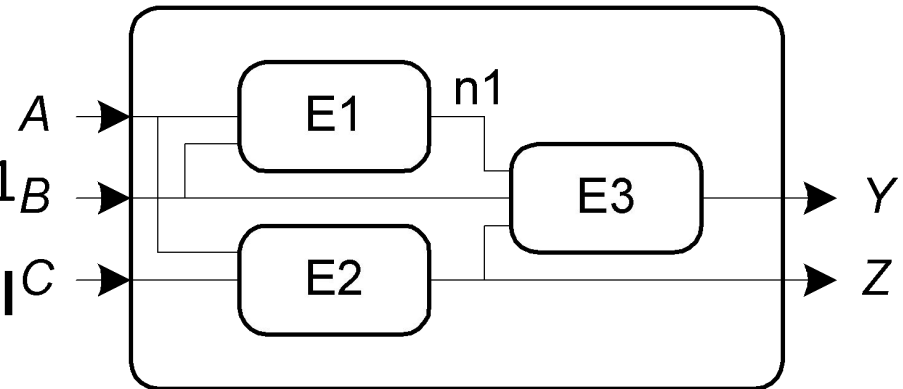
Схемы

- Узлы

- Входы: A, B, C
- Выходы: Y, Z
- Внутренний узел: $n1$

- Элементы схемы

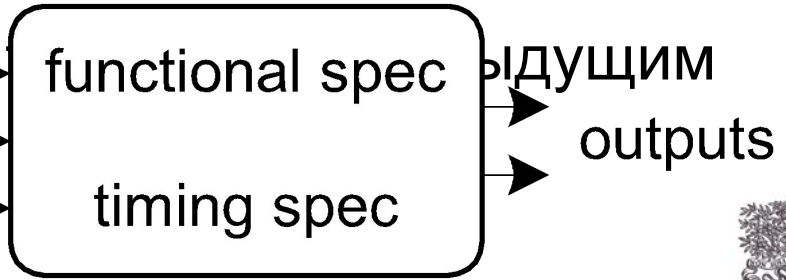
- $E1, E2, E3$
- Каждый из них, в свою очередь, является схемой



Типы цифровых схем

- **Комбинационные цифровые схемы**
 - Не имеют памяти
 - Выход определяется текущим состоянием входов

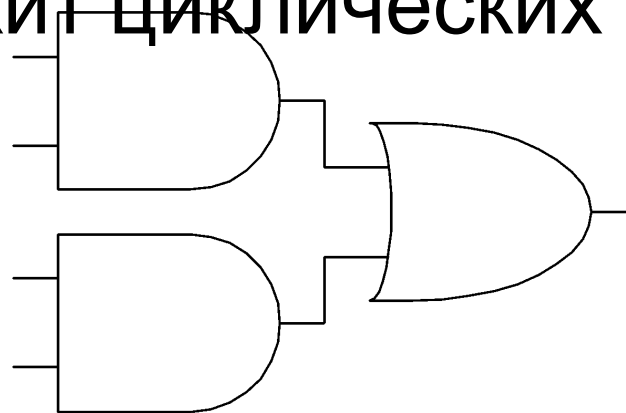
- **Последовательностные цифровые схемы**

- Имеют память
 - Выход определяется состоянием входов
- 
- The diagram shows a rounded rectangular block representing a sequential circuit. On the left side, there are three arrows pointing into the block, labeled 'inputs'. On the right side, there are two arrows pointing out of the block, labeled 'outputs'. Inside the block, the text 'functional spec' is positioned above 'timing spec'. To the right of the block, the text 'ыдущим' is positioned above 'outputs'.

Правила комбинационной

композиции

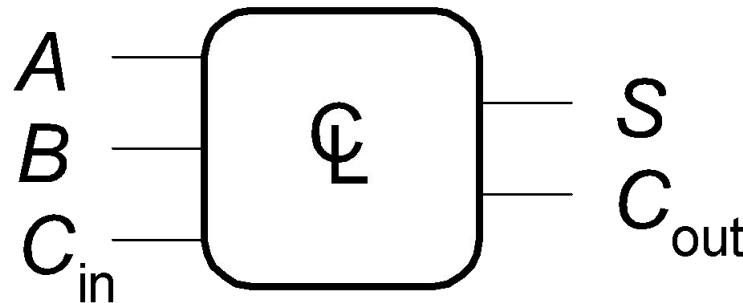
- Каждый элемент сам является комбинационным
- Каждое узел схемы является или входом, или подсоединен к одному-единственному выходу другого элемента
- Схема не содержит циклических путей
- **Пример:**



Булевы выражения

- Функциональная спецификация выходов по значениям входов
- **Пример:** $S = F(A, B, C_{in})$

$$C_{out} = F(A, B, C_{in})$$



$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + AC_{in} + BC_{in}$$

Некоторые определения

- Дополнение: переменная с чертой над именем
 A, B, C
- Литерал: переменная или ее дополнение
 $A, \bar{A}, B, \bar{B}, C, \bar{C}$
- Импликанта: произведение литералов
 ABC, AC, BC
- Минтерм: произведение, в которое входят литералы всех входных переменных
 $ABC, \bar{A}BC, A\bar{B}C$
- Макстерм: сумма, в которую входят литералы всех входных переменных
 $(A+B+C), (A+\bar{B}+\bar{C}), (\bar{A}+B+C)$

Дизъюнктивная форма

- Все выражения могут быть записаны в дизъюнктивной форме
- Каждой строке соответствует **минтерм**
- Минтерм является произведением (И, AND) литералов
- Каждый минтерм становится **ИСТИННЫМ** только для своей строки
- Функция записывается путем суммирования минтермов тех строк, для которых выход равен **ИСТИНЕ**
- Таким образом, формируется сумма (ИЛИ, OR) произведений (И, AND)

A	B	Y	minterm	minterm name
0	0	0	$\bar{A} \bar{B}$	m_0
0	1	1	$\bar{A} B$	m_1
1	0	0	$A \bar{B}$	m_2
1	1	1	$A B$	m_3

$$Y = F(A, B) =$$

Дизъюнктивная форма

- Все выражения могут быть записаны в дизъюнктивной форме
- Каждой строке соответствует **минтерм**
- Минтерм является произведением (AND) литералов
- Каждый минтерм становится **ИСТИННЫМ** только для своей строки
- Функция является суммой минтермов тех строк, для которых выход равен **ИСТИНЕ**
- Таким образом, формируется сумма (ИЛИ, OR) произведений (И, AND)

A	B	Y	minterm	minterm name
0	0	0	$\bar{A} \bar{B}$	m_0
0	1	1	$\bar{A} B$	m_1
1	0	0	$A \bar{B}$	m_2
1	1	1	$A B$	m_3

$$Y = F(A, B) =$$

Дизъюнктивная форма

- Все выражения могут быть записаны в дизъюнктивной форме
- Каждой строке соответствует минтерм
- Минтерм является произведением (И, AND) литералов
- Каждый минтерм становится ИСТИННЫМ только для своей строки
- Функция записывается путем суммирования минтермов тех строк, для которых выход равен ИСТИНЕ
- Таким образом, формируется сумма (ИЛИ, OR) произведений (И, AND)

A	B	Y	minterm	minterm name
0	0	0	$\bar{A} \bar{B}$	m_0
0	1	1	$\bar{A} B$	m_1
1	0	0	$A \bar{B}$	m_2
1	1	1	$A B$	m_3

$$Y = F(A, B) = \bar{A}B + AB = \Sigma(1, 3)$$

Конъюнктивная форма

- Все булевы выражения могут быть записаны в конъюнктивной форме
- Каждой строке соответствует **макстерм**
- Макстерм является суммой (ИЛИ, OR) литералов
- Каждый макстерм становится ЛОЖНЫМ только для своей строки
- Функция является произведением макстермов тех строк, для которых выход равен ЛОЖЬ
- Таким образом, формируется произведение (И, AND) сумм (ИЛИ, OR)

A	B	Y	maxterm	maxterm name
0	0	0	$A + B$	M_0
0	1	1	$A + \overline{B}$	M_1
1	0	0	$\overline{A} + B$	M_2
1	1	1	$\overline{A} + \overline{B}$	M_3

$$Y = F(A, B) = (A + B)(A + \overline{B}) = \Pi(0, 2)$$

Примеры булевых

- Вы собираетесь в кафетерий пообедать
 - Вы не пообедаете (\bar{E})
 - Там закрыто (не открыто, \bar{O}) или
 - Они предлагают только корн-доги (C)
- Запишите таблицу истинности, по которой можно определить пообедаете ли вы (E)

O	C	E
0	0	
0	1	
1	0	
1	1	

Примеры булевых

- Вы собираетесь в кафетерий пообедать
 - Вы не пообедаете (\bar{E})
 - Там закрыто (не открыто, \bar{O}) или
 - Они предлагают только корн-доги (C)
- Запишите таблицу истинности, по которой можно определить пообедаете ли вы (E)

O	C	E
0	0	0
0	1	0
1	0	1
1	1	0

Дизъюнктивная и конъюнктивная формы

- Дизъюнктивная форма (SOP, sum-of-products) сумма (ИЛИ) произведений (И)

O	C	E	minterm
0	0		$\overline{O} \overline{C}$
0	1		$\overline{O} C$
1	0		$O \overline{C}$
1	1		$O C$

- Конъюнктивная форма (POS, product-of-sums) - произведение (И) сумм (ИЛИ)

O	C	Y	maxterm
0	0		$O + C$
0	1		$O + \overline{C}$
1	0		$\overline{O} + C$
1	1		$\overline{O} + \overline{C}$

Дизъюнктивная и конъюнктивная формы

- Дизъюнктивная форма (SOP, sum-of-products) сумма (ИЛИ) произведений (И)

O	C	E	minterm
0	0	0	$\overline{O} \overline{C}$
0	1	0	$\overline{O} C$
1	0	1	$O \overline{C}$
1	1	0	$O C$

$$Y = O\overline{C}$$
$$= \Sigma(2)$$

- Конъюнктивная форма (POS, product-of-sums) - произведение (И) сумм (ИЛИ)

O	C	E	maxterm
0	0	0	$O + C$
0	1	0	$O + \overline{C}$
1	0	1	$\overline{O} + C$
1	1	0	$\overline{O} + \overline{C}$

$$Y = (O + C)(O + \overline{C})(\overline{O} + \overline{C})$$
$$= \Pi(0, 1, 3)$$

Булева алгебра

- Аксиомы и теоремы позволяют **упрощать** булевы выражения
- Подобно обычной алгебре, но проще: переменные принимают только два значения (0 или 1)
- **Двойственность** аксиом и теорем:
 - Можно взаимно заменить И и ИЛИ, 0 и 1

Булевы аксиомы

	Axiom		Dual		Name
A1	$B = 0 \text{ if } B \neq 1$	A1'	$B = 1 \text{ if } B \neq 0$		Binary field
A2	$\bar{0} = 1$	A2'	$\bar{1} = 0$		NOT
A3	$0 \bullet 0 = 0$	A3'	$1 + 1 = 1$		AND/OR
A4	$1 \bullet 1 = 1$	A4'	$0 + 0 = 0$		AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	A5'	$1 + 0 = 0 + 1 = 1$		AND/OR

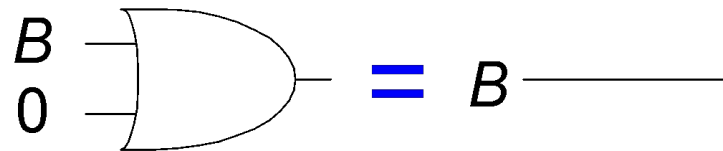
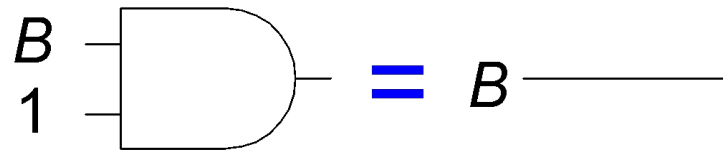
	Theorem		Dual		Name
T1	$B \bullet 1 = B$	T1'	$B + 0 = B$		Identity
T2	$B \bullet 0 = 0$	T2'	$B + 1 = 1$		Null Element
T3	$B \bullet B = B$	T3'	$B + B = B$		Idempotency
T4		$\bar{\bar{B}} = B$			Involution
T5	$B \bullet \bar{B} = 0$	T5'	$B + \bar{B} = 1$		Complements

T1: Теорема идентичности

- $V \cdot 1 = V$
- $V + 0 = V$

T1: Теорема идентичности

- $B \cdot 1 = B$
- $B + 0 = B$

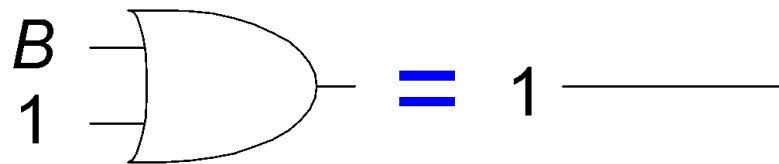
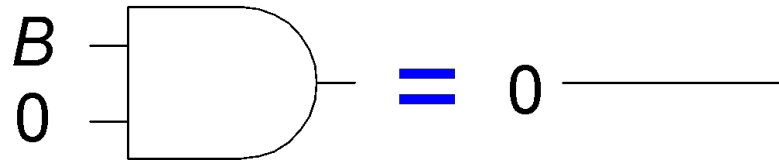


T2: Теорема о нулевом элементе

- $B \cdot 0 = 0$
- $B + 1 = 1$

T2: Теорема о нулевом элементе

- $B \cdot 0 = 0$
- $B + 1 = 1$

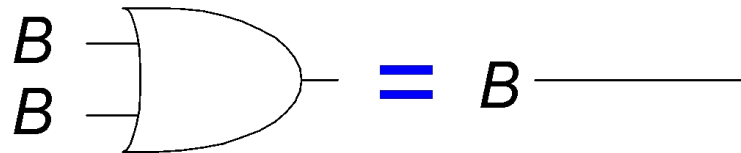
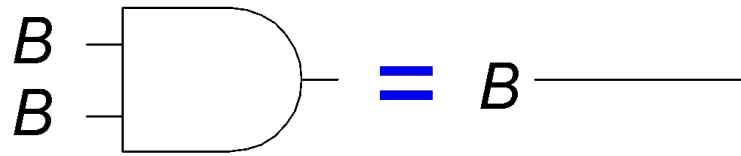


Т3: Теорема об идемпотентности

- $V \cdot V = V$
- $V + V = V$

Т3: Теорема об идемпотентности

- $B \cdot B = B$
- $B + B = B$

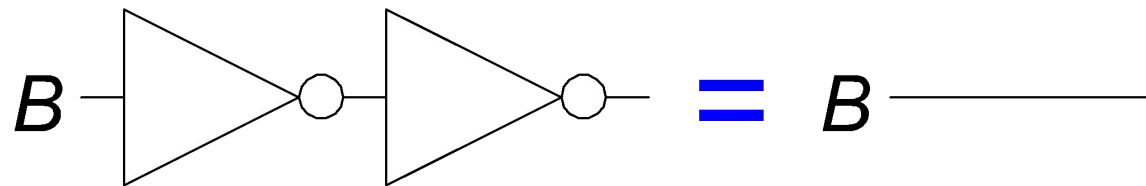


T4: Теорема идентичности

- $\overline{\overline{B}} = B$

T4: Теорема идентичности

- $\overline{\overline{B}} = B$

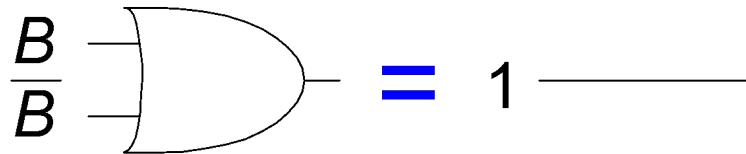
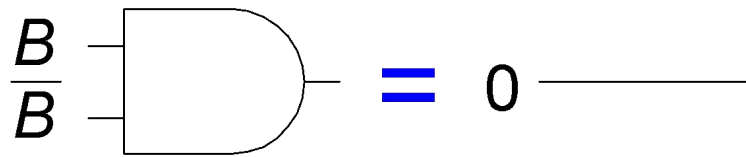


T5: Теорема о дополнительности

- $V \cdot \bar{V} = 0$
- $V + \bar{V} = 1$

T5: Теорема о дополнительности

- $B \cdot \bar{B} = 0$
- $B + \bar{B} = 1$



Булевы теоремы, обзор

	Theorem		Dual	Name
T1	$B \bullet 1 = B$	T1'	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	T2'	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	T3'	$B + B = B$	Idempotency
T4		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5'	$B + \overline{B} = 1$	Complements

Булевы теоремы нескольких переменных

	Theorem		Dual		Name
T6	$B \cdot C = C \cdot B$	T6'	$B + C = C + B$		Commutativity
T7	$(B \cdot C) \cdot D = B \cdot (C \cdot D)$	T7'	$(B + C) + D = B + (C + D)$		Associativity
T8	$(B \cdot C) + B \cdot D = B \cdot (C + D)$	T8'	$(B + C) \cdot (B + D) = B + (C \cdot D)$		Distributivity
T9	$B \cdot (B + C) = B$	T9'	$B + (B \cdot C) = B$		Covering
T10	$(B \cdot C) + (B \cdot \overline{C}) = B$	T10'	$(B + C) \cdot (B + \overline{C}) = B$		Combining
T11	$(B \cdot C) + (\overline{B} \cdot D) + (C \cdot D)$ $= B \cdot C + \overline{B} \cdot D$	T11'	$(B + C) \cdot (\overline{B} + D) \cdot (C + D)$ $= (B + C) \cdot (\overline{B} + D)$		Consensus
T12	$\overline{B_0 \cdot B_1 \cdot B_2 \dots}$ $= (\overline{B_0} + \overline{B_1} + \overline{B_2} \dots)$	T12'	$\overline{B_0 + B_1 + B_2 \dots}$ $= (\overline{B_0} \cdot \overline{B_1} \cdot \overline{B_2})$		De Morgan's Theorem

Упрощение булевых

Пример 1:

- $Y = AB + \overline{A}B$

Упрощение булевых

Пример 1:

- $Y = AB + \bar{A}B$
 $= B(A + \bar{A})$ T8
 $= B(1)$ T5'
 $= B$ T1

Упрощение булевых

Пример 2:

- $Y = A(AB + ABC)$

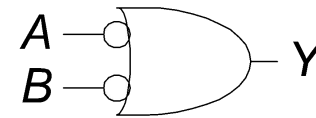
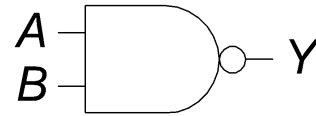
Упрощение булевых

Пример 2:

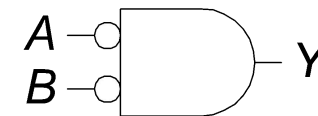
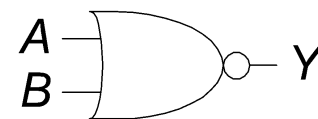
- $Y = A(AB + ABC)$
 - $= A(AB(1 + C))$ T8
 - $= A(AB(1))$ T2'
 - $= A(AB)$ T1
 - $= (AA)V$ T7
 - $= AB$ T3

Теорема де Моргана

- $Y = \overline{AB} = \overline{A} + \overline{B}$



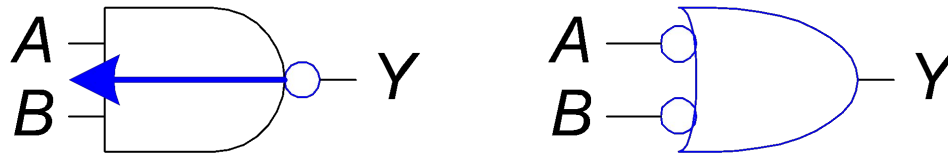
- $Y = \overline{A + B} = \overline{A} \cdot \overline{B}$



Перемещение инверсии

• Назад:

- Изменить тип элемента
- Добавить инверсию на входы



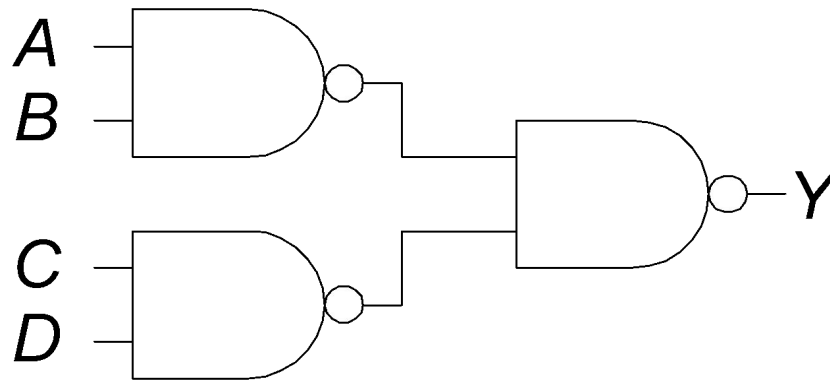
• Вперед:

- Изменить тип элемента
- Добавить инверсию на выход



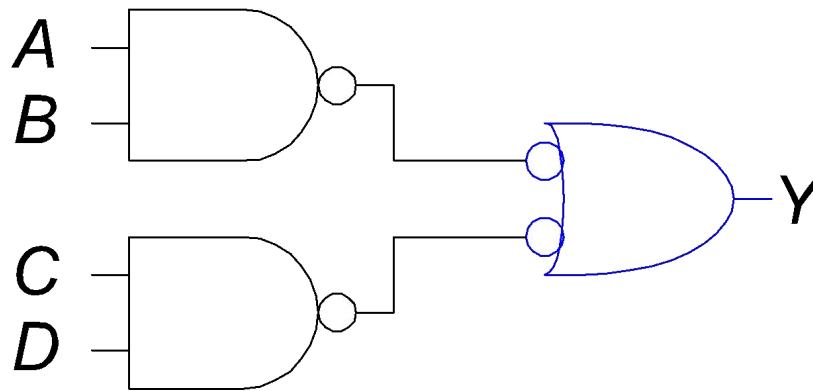
Перемещение инверсии

- Запишите булево выражение для этой схемы



Перемещение инверсии

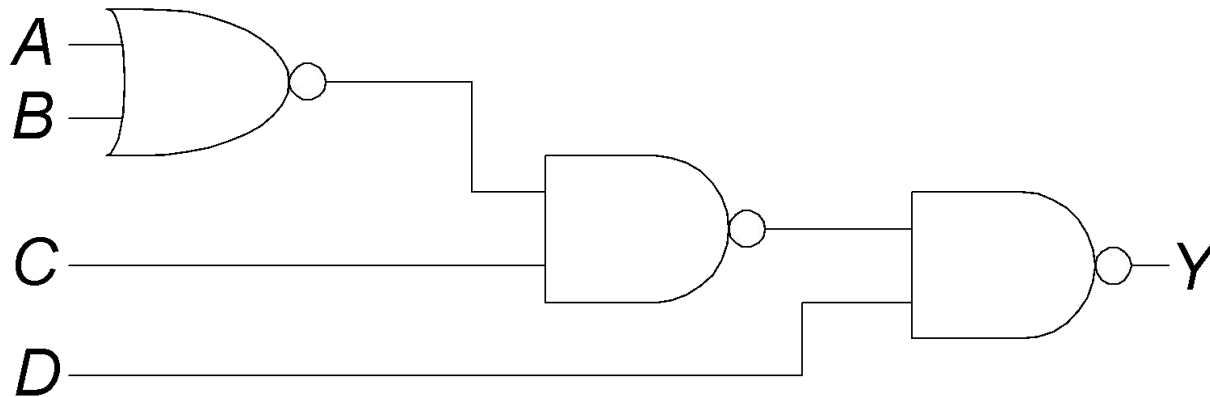
- Запишите булево выражение для этой схемы



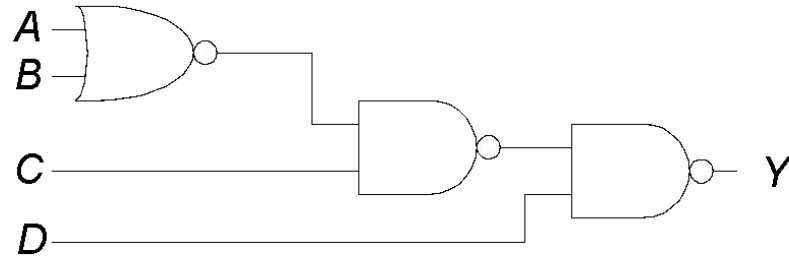
$$Y = AB + CD$$

Правила перемещения инверсии

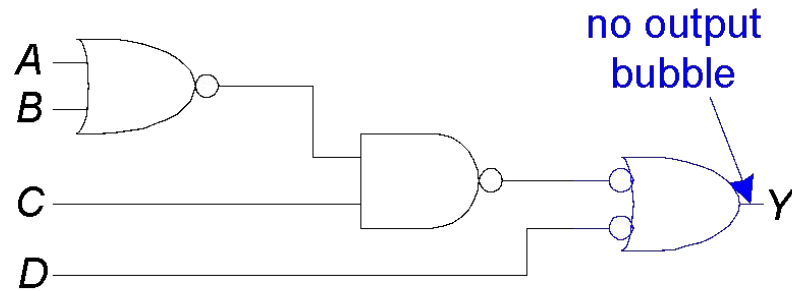
- Начать с выхода и двигаться в направлении входов
- Перемещать инверсию от выходов ко входам
- Нарисовать элементы так, чтобы увидеть, что инверсии взаимно уничтожаются



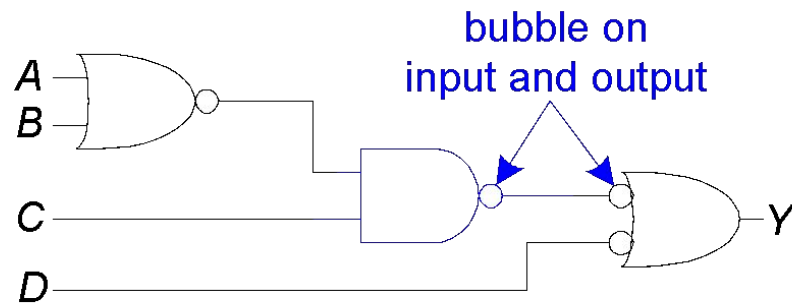
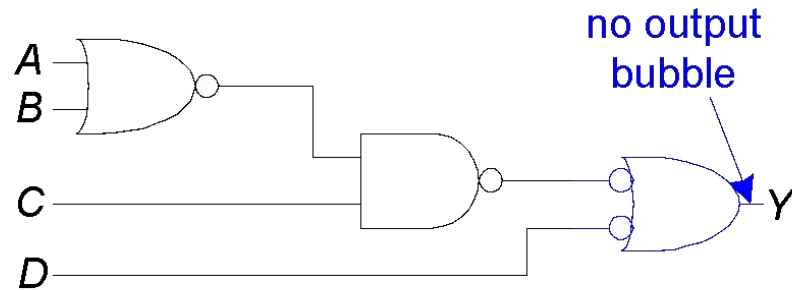
Пример перемещения инверсии



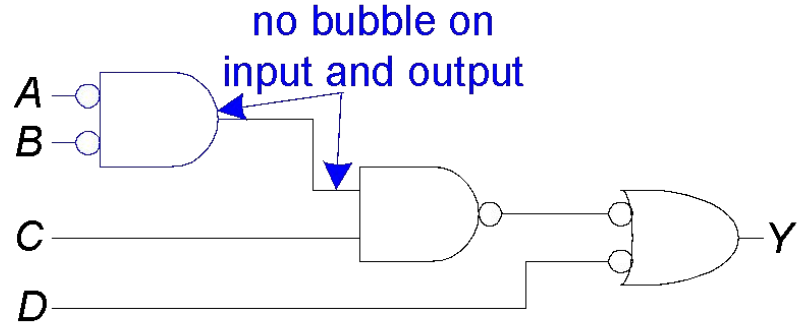
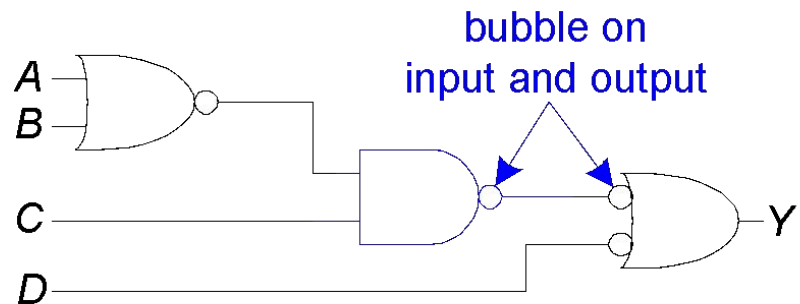
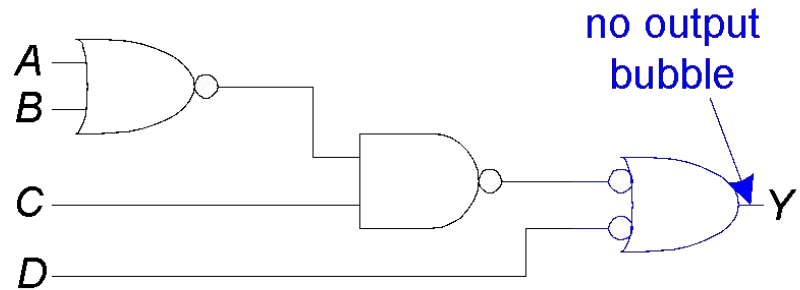
Пример перемещения инверсии



Пример перемещения инверсии



Пример перемещения инверсии

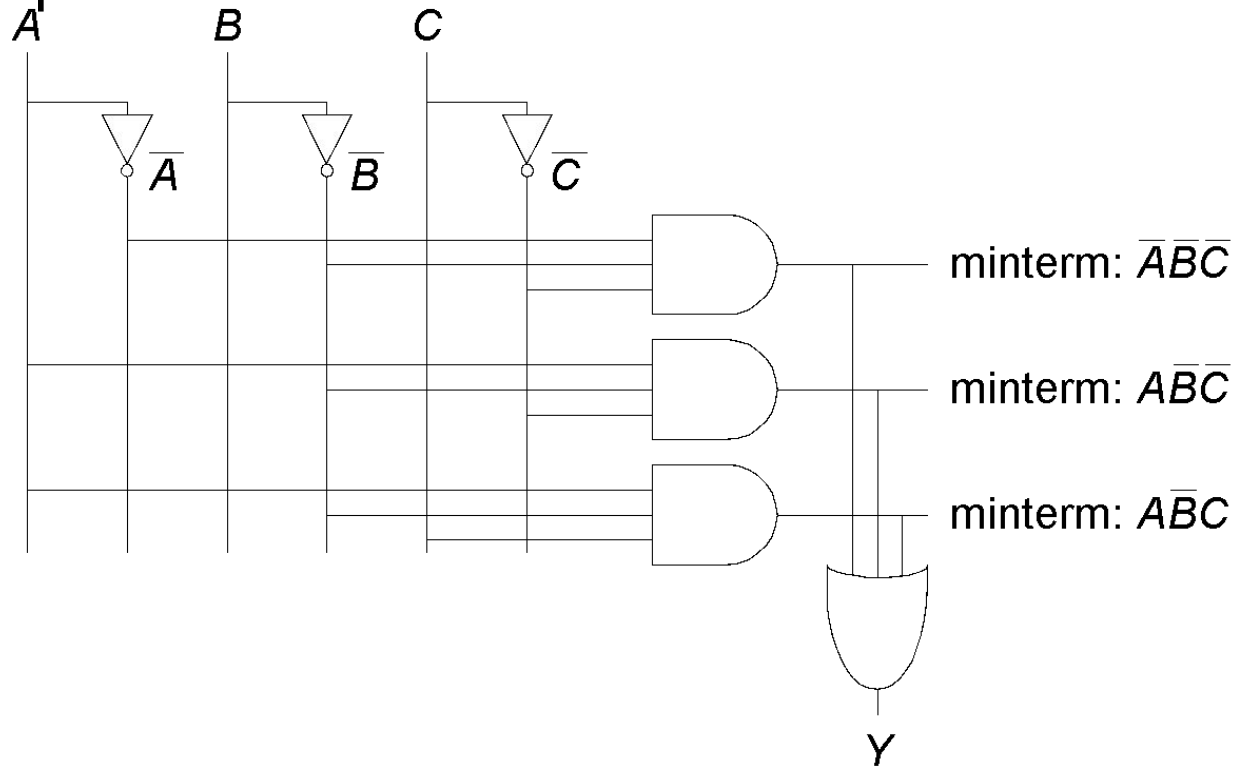


$$Y = \overline{A}BC + \overline{D}$$



От логики к логическим элементам

- Двухуровневая цифровая схема Сначала элементы И, затем ИЛИ
- Пример: $Y = \overline{A}B\overline{C} + A\overline{B}C + A\overline{B}C$



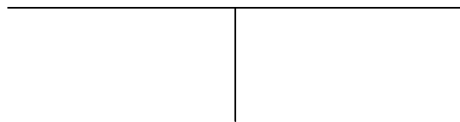
Правила изображения принципиальных схем

- Входы слева (или сверху)
- Выходы справа (или внизу)
- Информация передается от элементов, расположенных слева, к элементам, расположенным справа
- Для проводников стараться использовать прямые линии

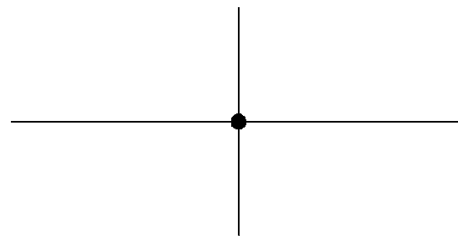
Правила изображения принципиальных схем (продолжение)

- Проводники всегда должны соединяться в виде буквы Т
- Точка в месте пересечения проводников обозначает их соединение
- Проводники, пересекающиеся без точки, не имеют соединения друг с другом

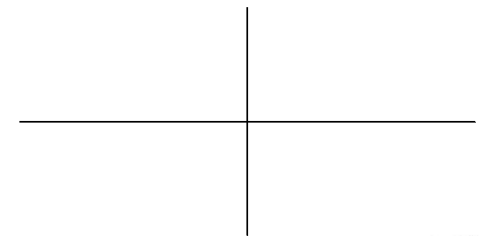
wires connect
at a T junction



wires connect
at a dot



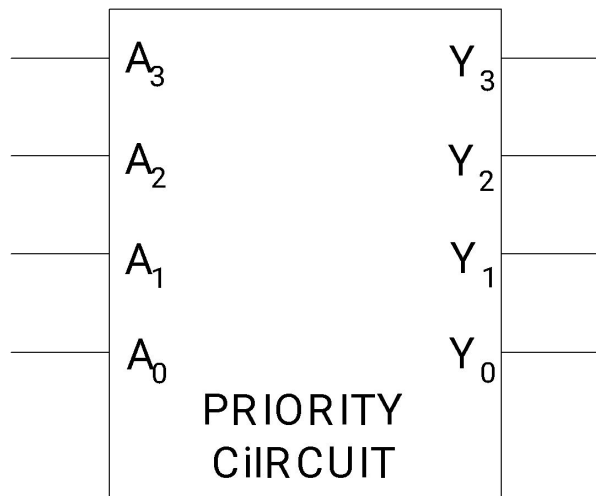
wires crossing
without a dot do
not connect



Схемы с несколькими

- Пример: Схема приоритета**

Выход устанавливается
в соответствии
со старшим разрядом
который равен ИСТИНЕ

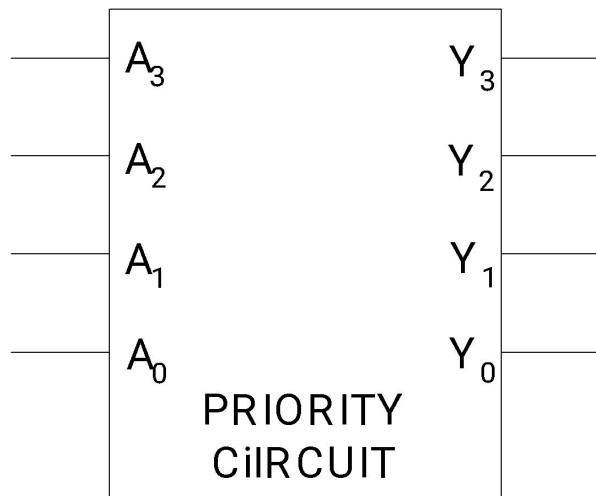


A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0				
0	0	0	1				1
0	0	1	0			1	
0	0	1	1			1	1
0	1	0	0		1		
0	1	0	1		1		1
0	1	1	0		1	1	
0	1	1	1		1	1	1
1	0	0	0	1			
1	0	0	1	1			
1	0	1	0	1			
1	0	1	1	1			
1	1	0	0	1	1		
1	1	0	1	1	1		
1	1	1	0	1	1	1	
1	1	1	1	1	1	1	1

Схемы с несколькими

- **Пример: Схема приоритета**

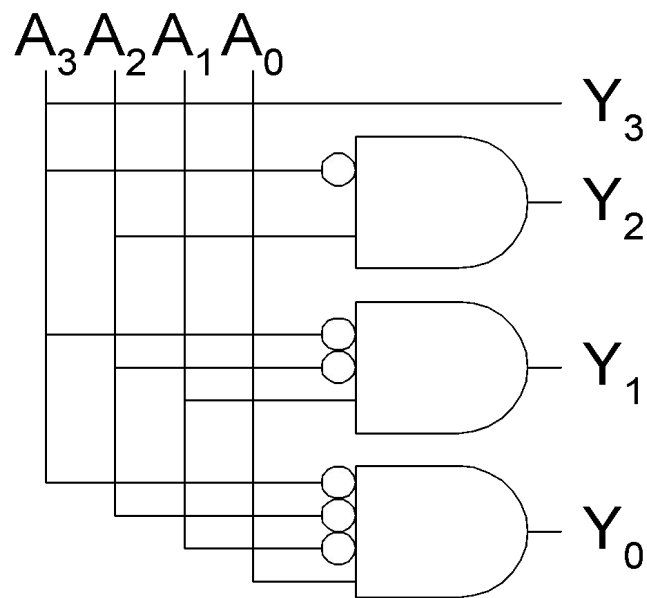
Выход устанавливается
в соответствии
со старшим входом
который равен ИСТИНЕ



A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

Реализация схемы

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0



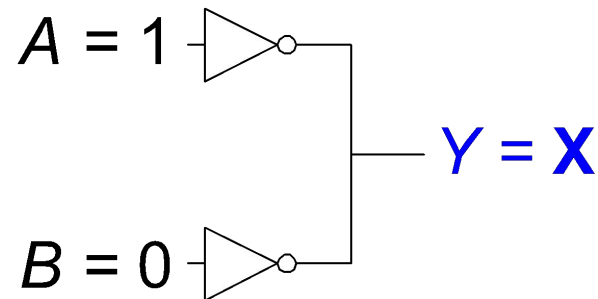
Безразличное значение

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	0
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	0	0
1	1	1	1	1	0	0	0

A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	X	0	0	1	0
0	1	X	X	0	1	0	0
1	X	X	X	1	0	0	0

Конфликт: X

- Конфликтом: схема пытается установить на выходе одновременно 0 и 1
 - Действительное значение находится где-то между
 - Может быть 0, 1 или в запрещенной диапозоне
 - Может зависеть от напряжения, температуры, времени, шума
 - Часто вызывает большое энергопотребление

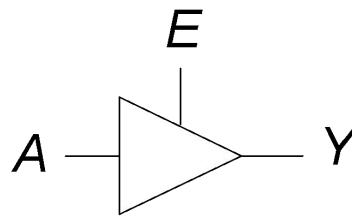


- **Предупреждение:**
 - Конфликт обычно является **ошибкой**
 - В зависимости от контекста **X** обозначает **безразличное значение или конфликт**

Третье состояние: Z

- Третье состояние, состояние высокого импеданса, неподключенная или плавающая цепь
- Состояние неподключенного выхода может быть 0, 1 или быть промежуточным
 - Вольтметр не показывает что узел находится в плавающем состоянии

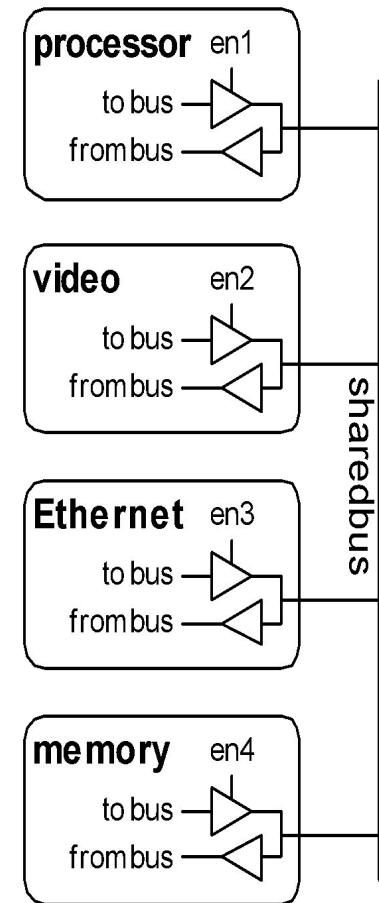
Буфер с тремя состояниями



<i>E</i>	<i>A</i>	<i>Y</i>
0	0	Z
0	1	Z
1	0	0
1	1	1

Шины с тремя состояниями

- Состояние высокого импеданса используется для создания шин
 - Несколько микросхем подключены к шине
 - Но активной в некоторый момент может быть одна и только одна



Карты Карно

- Булевы выражения можно упростить путем комбинирования термов
- Карты Карно позволяют наглядно минимизировать выражение
- $PA + P\bar{A} = P$

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Y		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	0	0	0

Y		AB			
		00	01	11	10
C	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
	1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Карты Карно

- Обведите овалом 1 в соседних квадратах
- В булево выражение включаются только те литералы, прямая и инверсная форма которых не попадает в овал

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	0	0	0

$$Y = \overline{A}B$$

Карты Карно - три входа

		AB			
		00	01	11	10
C	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$A\bar{B}\bar{C}$	$A\bar{B}C$
	1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

K-Map

		AB			
		00	01	11	10
C	0				
	1				

Карты Карно - три входа

		AB			
		00	01	11	10
C	0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$A\bar{B}\bar{C}$	$AB\bar{C}$
	1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

K-Map

		AB			
		00	01	11	10
C	0	0	1	1	0
	1	0	1	0	0

$$Y = \bar{A}B + B\bar{C}$$

Построение карты Карно

- **Дополнение:** переменная с чертой над именем
A, B, C
- **Литерал:** Литерал: переменная или ее дополнение
A, A, B, B, C, C
- **Импликанта:** произведение литералов
ABC, AC, BC
- **Первичная импликанта:** импликанта, соответствующая наибольшему овалу на карте Карно

Правила карт Карно

- Каждая 1 должна входить хотя бы в один овал
- Каждый овал должен охватывать блок, число клеток которого в каждом направлении равно степени двойки (то есть 1, 2 или 4)
- Каждый овал должен настолько большим, насколько это возможно
- Овал может связывать края карты Карно
- Безразличные значения (X) могут входить в овал, если это помогает минимизировать выражение

Карты Карно - четыре входа

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

<i>Y</i>	<i>AB</i>			
<i>CD</i>	00	01	11	10
00				
01				
11				
10				

Карты Карно - четыре входа

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

<i>Y</i>	<i>AB</i>			
<i>CD</i>	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

Карты Карно - четыре входа

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

<i>Y</i>	<i>AB</i>			
<i>CD</i>	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

$$Y = \bar{A}C + \bar{A}BD + A\bar{B}\bar{C} + \bar{B}\bar{D}$$

Карты Карно и безразличные значения

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>Y</i>
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

<i>Y</i>	<i>AB</i>			
<i>CD</i>	00	01	11	10
00				
01				
11				
10				

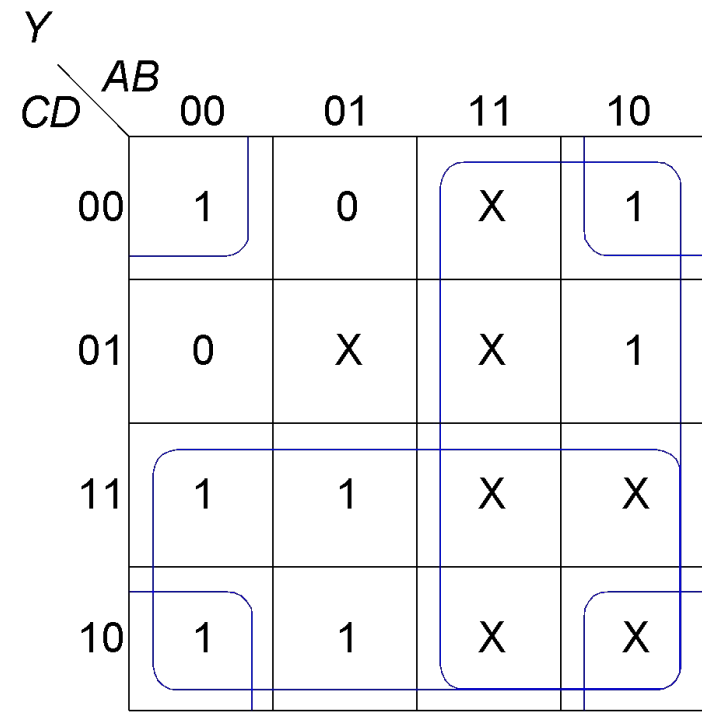
Карты Карно и безразличные значения

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y	AB			
CD	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

Карты Карно и безразличные значения

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X



$$Y = A + \bar{B}\bar{D} + C$$



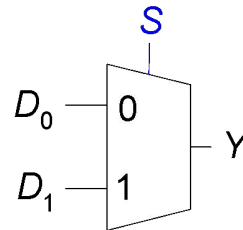
Базовые комбинационные блоки

- Мультиплексоры
- Дешифраторы

Мультиплексор (Мух)

- Выбирает один из N входов и соединяет его с выходом
- $\log_2 N$ -бит для выбора входа – вход управления (выбора)

- Пример:



2:1 Мух

S	D ₁	D ₀	Y	S	Y
0	0	0	0	0	D ₀
0	0	1	1	1	D ₁
0	1	0	0		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	1		
1	1	1	1		

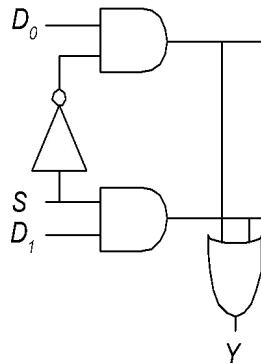
Реализация

- **Используя логические элементы**

- Дизъюнктивная форма

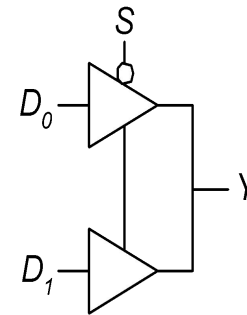
S	$D_0 D_1$			
	00	01	11	10
0	0	0	1	1
1	0	1	1	0

$$Y = D_0 \bar{S} + D_1 S$$



- **Используя буферы с тремя состояниями**

- Для N-входового мультиплексора используется N буферов с тремя состояниями
- Один и только один из них включается для выбора соответствующего входа

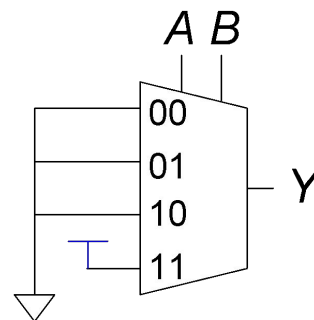


Цифровые схемы на основе мультиплексоров

- Использование мультиплексоров как таблиц преобразования

<i>A</i>	<i>B</i>	<i>Y</i>
0	0	0
0	1	0
1	0	0
1	1	1

$$Y = AB$$

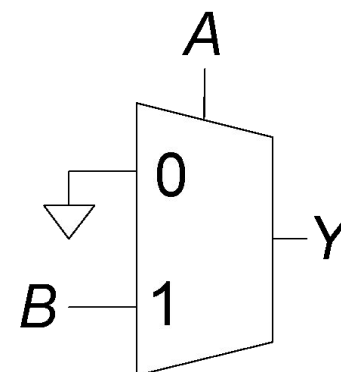


- Уменьшение размера мультиплексора

$$Y = AB$$

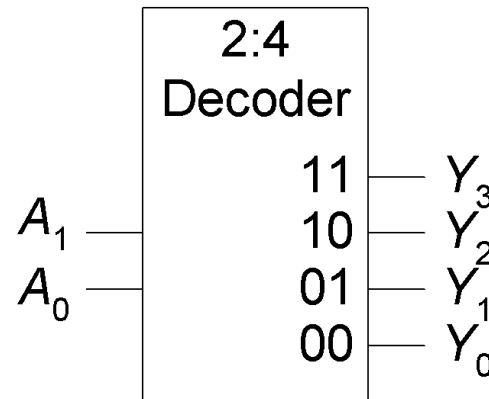
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A	Y
0	0
1	B



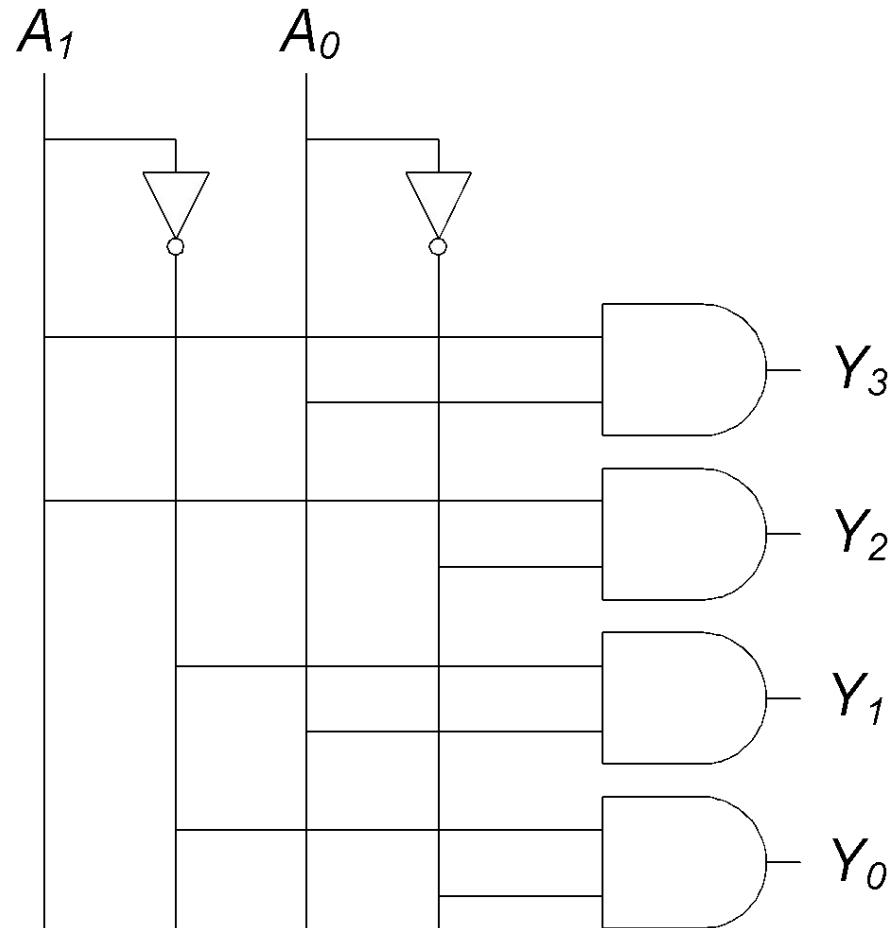
Дешифраторы

- N входов, 2^N выходов
- Прямой унитарный код: только один выход принимает значение ИСТИНА

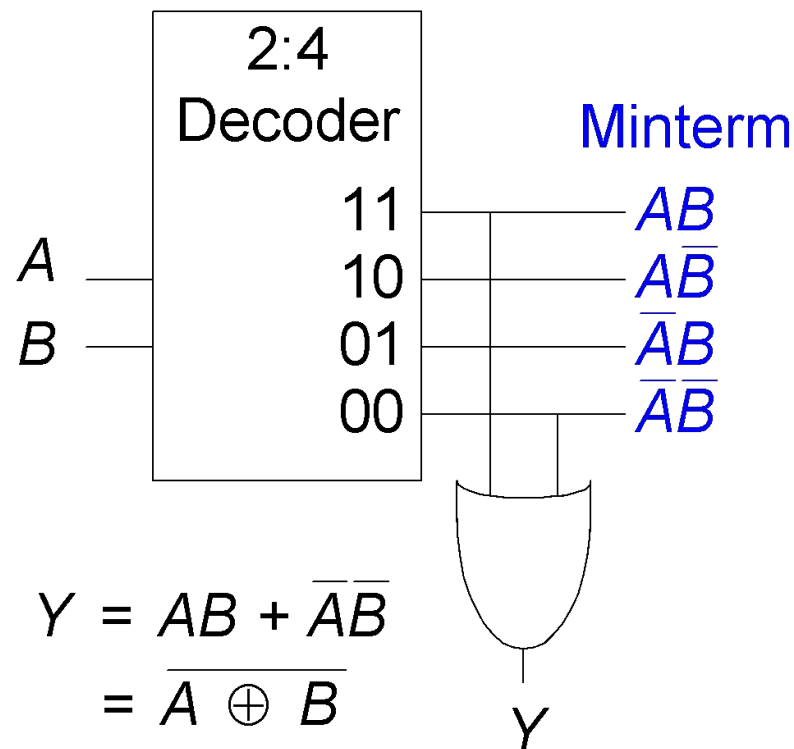


A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Реализация дешифраторов

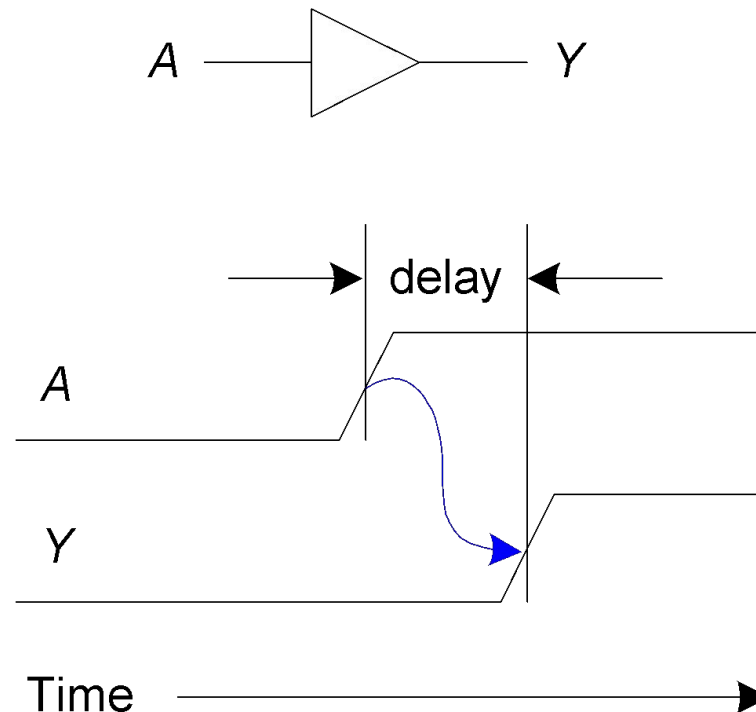


- Функция ИЛИ от минтермов



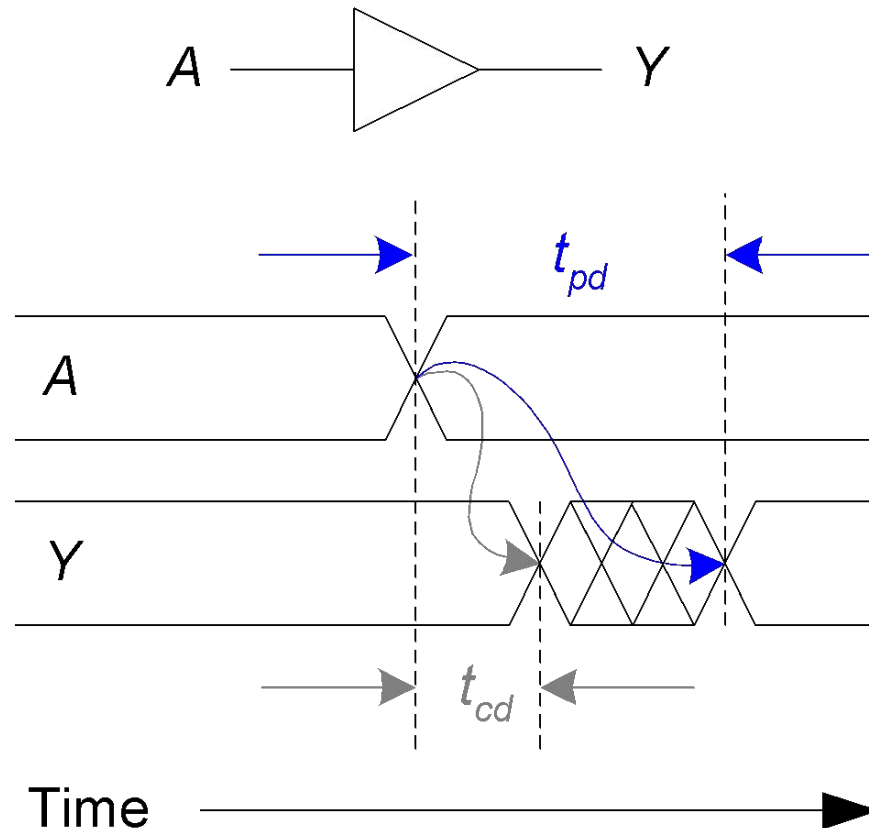
Временные характеристики

- Задержка между изменением входа и соответствующим изменением выхода
- Как проектировать быстрые схемы?



Задержки распространения и реакции

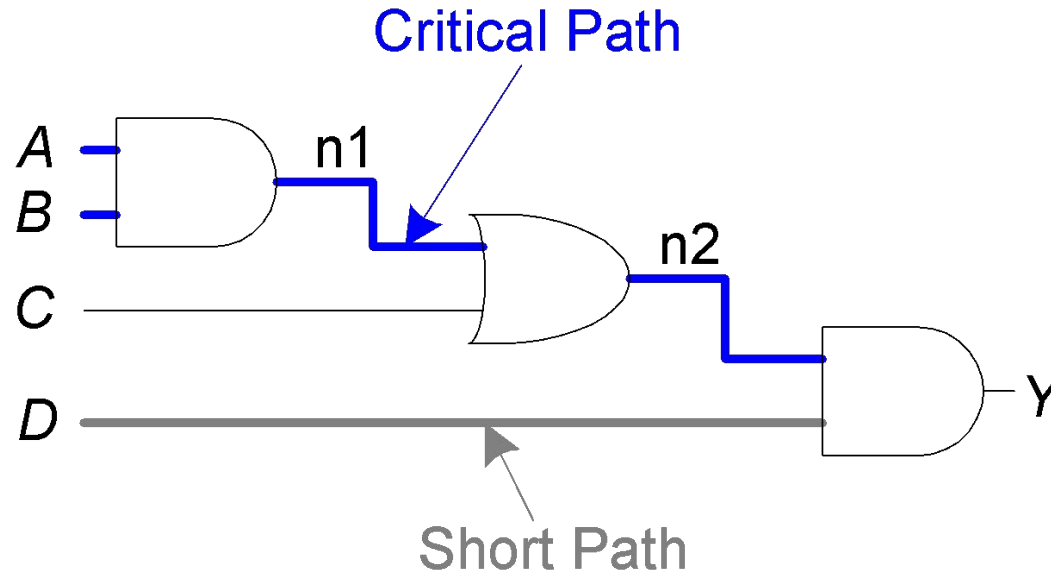
- **Задержка распространения** t_{pd} = максимальная задержка тракта ВХОД-ВЫХОД
- **Задержка реакции** t_{cd} = минимальная задержка тракта ВХОД-ВЫХОД



Задержки распространения и реакции

- Задержки обусловлены
 - Емкостями и сопротивлениями в цепях
 - Конечностью скорости света
- Причины, по которым t_{pd} и t_{cd} могут различаться
 - Разные задержки нарастания и спада сигнала
 - Несколько входов и выходов, одни из которых быстрее, чем другие
 - Замедление работы схемы при повышении температуры и ускорение при охлаждении

Критический (длинный) и кратчайший ПУТИ



Критический (длинный) путь $t_{pd} = 2t_{pd_AND} + t_{pd_OR}$

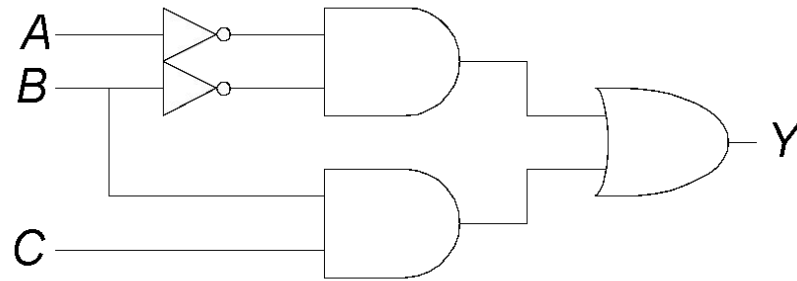
Кратчайший путь $t_{cd} = t_{cd_AND}$

Импульсные помехи

- **Одиночное изменение на входного сигнала вызывает несколько изменений сигнала на выходе**

Пример импульсной помехи

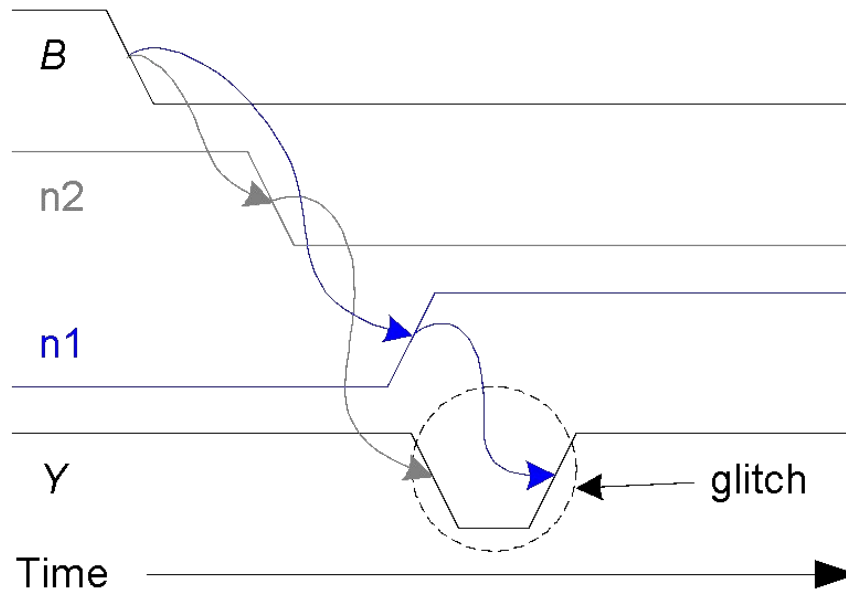
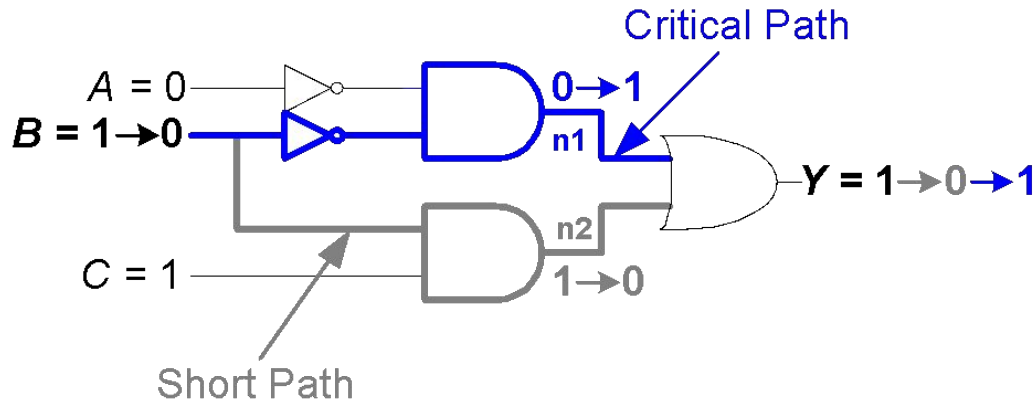
- Что происходит когда $A = 0$, $C = 1$, а B изменяется с 1 до 0?



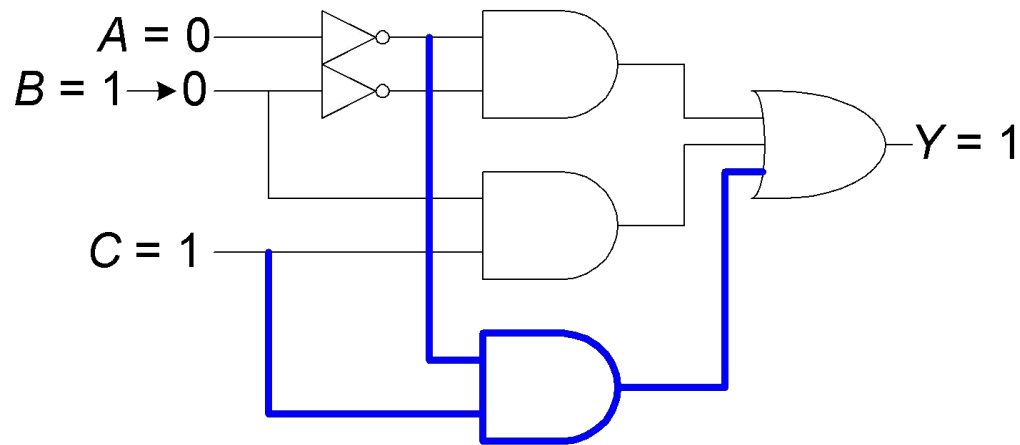
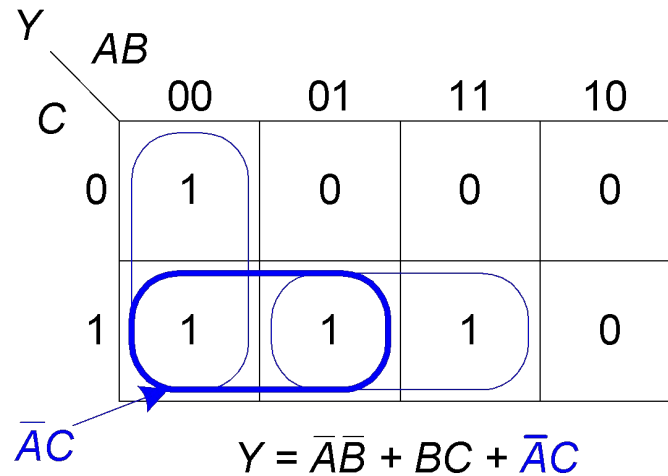
		AB			
		00	01	11	10
C	0	1	0	0	0
	1	1	1	1	0

$$Y = \bar{A}\bar{B} + BC$$

Пример импульсной помехи



Борьба с импульсными



Почему импульсные помехи важны?

- Импульсные помехи не являются серьезной проблемой при проектировании **синхронных схем** (глава 3)
- Важно уметь распознавать их при моделировании или на экране осциллографа
- От всех импульсных помех невозможно избавиться - одновременные изменения нескольких входов могут привести к появлению таких помех