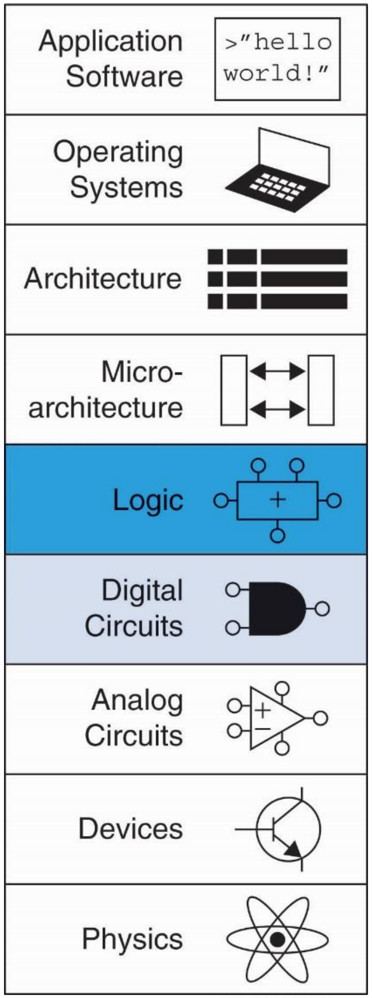
Глава 2 :: Темы

- **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**



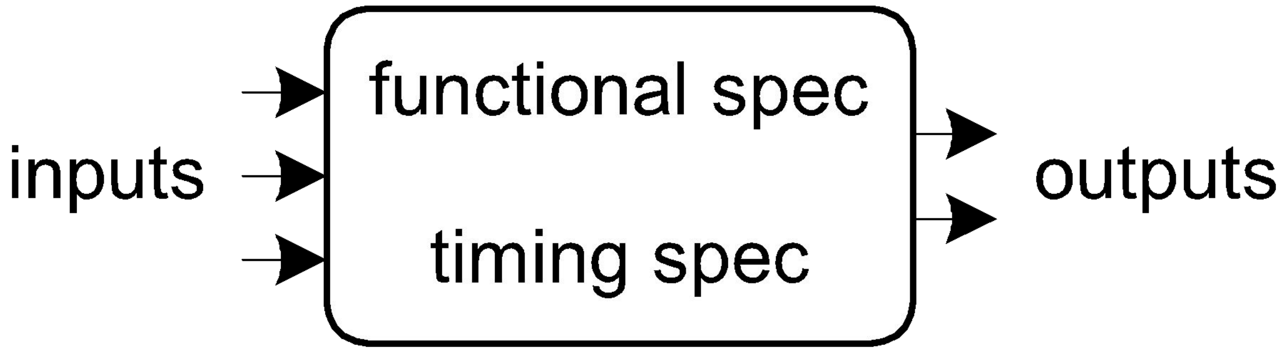
© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

Введение

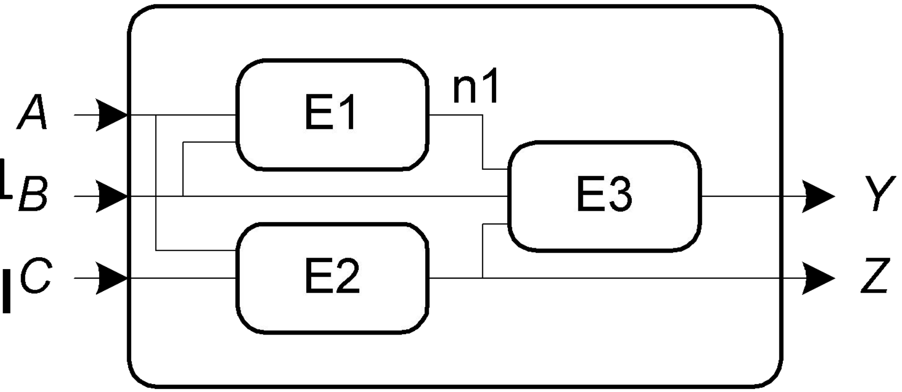
Логическая схема состоит из:

- Входов
- Выходов
- Функциональной спецификации
- Временной спецификации



Схемы

- Узлы
  - Входы: *A*, *B*, *C*
  - Выходы: *Y*, *Z*
  - Внутренний узел: n1
- Элементы схемы
  - E1, E2, E3
  - Каждый их них, в свою очередь, является схемой

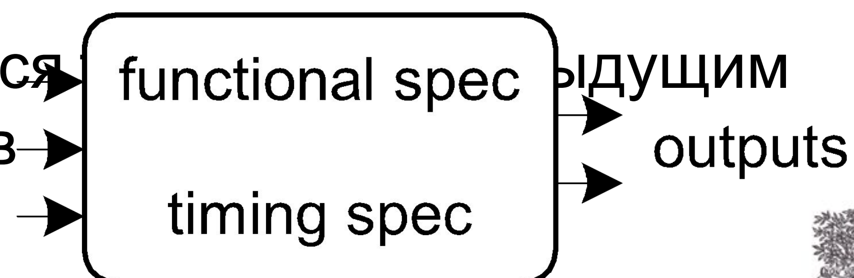


© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

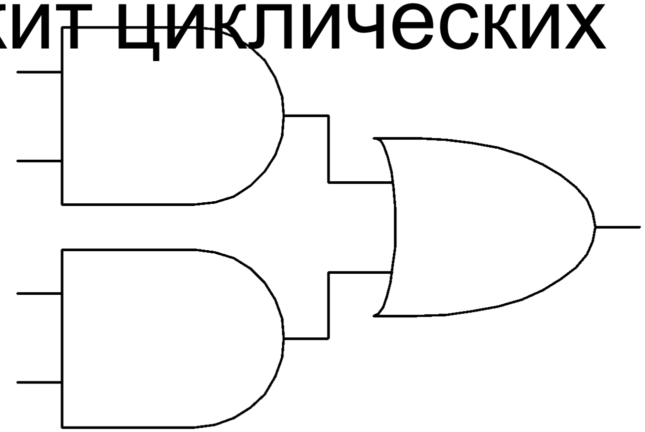
Типы цифровых схем

- **Комбинационные цифровые схемы**
  - Не имеют памяти
  - Выход определяется текущим состоянием входов
- **Последовательностные цифровые схемы**
  - Имеют память
  - Выход определяется … ыдущим состоянием в…



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:**



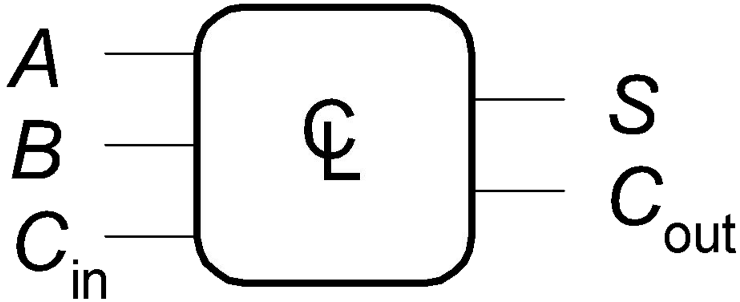
© Digital Design and Computer Architecture, 2nd Edition, 2012

ELSEVIER

Булевы выражения

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

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



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

© Digital Design and Computer Architecture, 2nd Edition, 2012

Некоторые определения

- Дополнение: переменная с чертой над именем

  $\bar{A}, \bar{B}, \bar{C}$

- Литерал: переменная или ее дополнение

  $A, \bar{A}, B, \bar{B}, C, \bar{C}$

- Импликанта: произведение литералов

  $AB\bar{C}, \bar{A}C, BC$

- Минтерм: произведение, в которое входят литералы всех входных переменных

  $AB\bar{C}, \bar{A}\bar{B}C, ABC$

- Макстерм: сумма, в которую входят литералы всех входных переменных

  $(A+\bar{B}+C), (\bar{A}+B+\bar{C}), (A+\bar{B}+\bar{C})$

© Digital Design and Computer Architecture, 2nd Edition, 2012

ELSEVIER

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	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	1	$A\ B$	m_3

$$Y = \mathbf{F}(A, B) =$$

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	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	1	$A\ B$	m_3

$$Y = \mathbf{F}(A, B) =$$

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	$\overline{A}\ \overline{B}$	m_0
0	1	1	$\overline{A}\ B$	m_1
1	0	0	$A\ \overline{B}$	m_2
1	1	1	$A\ B$	m_3

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

ELSEVIER

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

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

© Digital Design and Computer Architecture, 2nd Edition, 2012

ELSEVIER

Примеры булевых

- Вы собираетесь в кафетерий пообедать
  - Вы не пообедаете ($\overline{E}$)
  - Там закрыто (не открыто, $\overline{O}$) 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 ($\overline{E}$)
  - Там закрыто (не открыто, $\overline{O}$) 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

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

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	$\overline{O} + \overline{C}$

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

ELSEVIER

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	$B = 0$ if $B \neq 1$	A1′	$B = 1$ if $B \neq 0$	Binary field
A2	$\overline{0} = 1$	A2′	$\overline{1} = 0$	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		$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	T5′	$B + \overline{B} = 1$	Complements

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

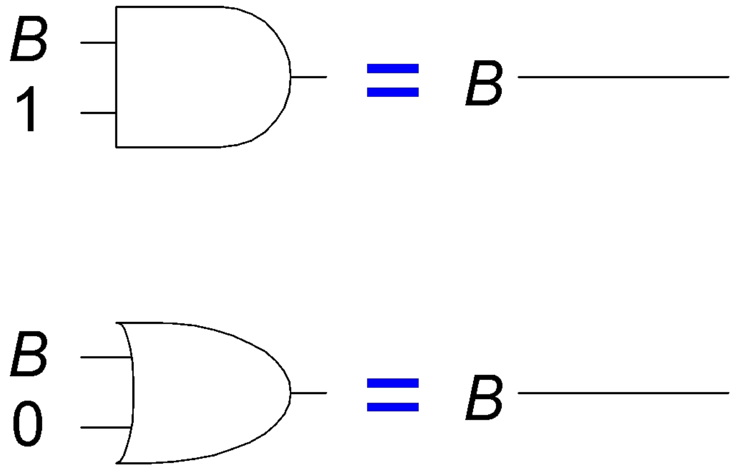
ELSEVIER

T1: Теорема идентичности

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

T1: Теорема идентичности

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



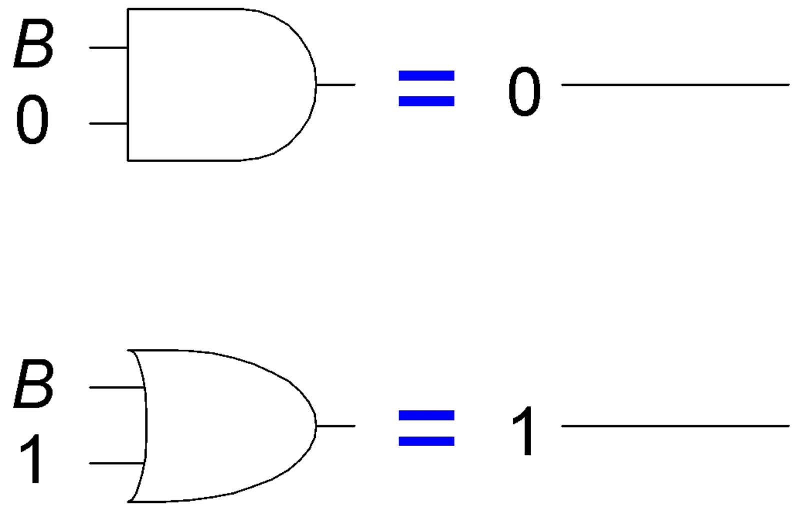
T2: Теорема о нулевом элементе

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

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

T2: Теорема о нулевом элементе

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



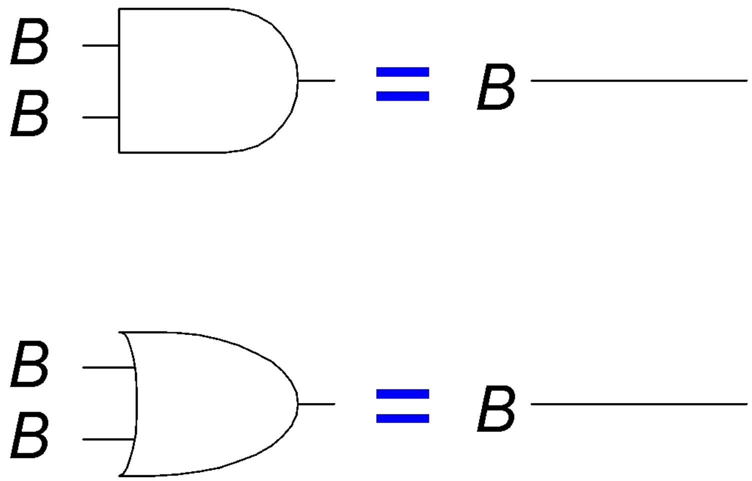
T3: Теорема об идемпотентности

- $B \bullet B = B$
- $B + B = B$

ELSEVIER

T3: Теорема об идемпотентности

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

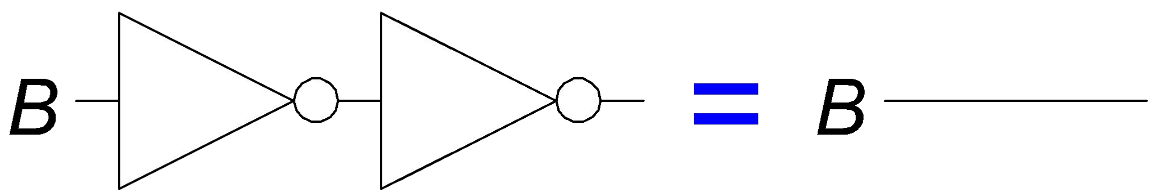


T4: Теорема идентичности

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

T4: Теорема идентичности

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



T5: Теорема о дополнительности

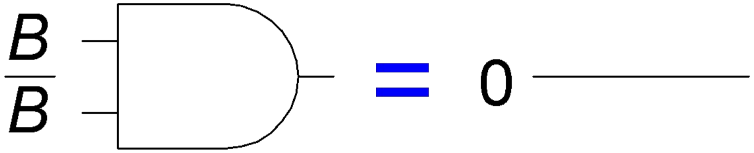
- $B \bullet \overline{B} = 0$
- $B + \overline{B} = 1$

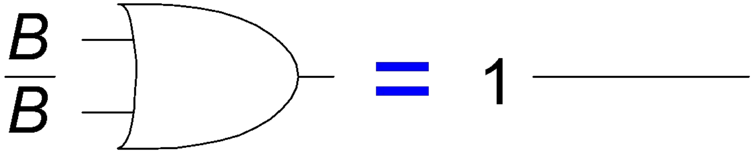
© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

T5: Теорема о дополнительности

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





© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

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		$\bar{\bar{B}} = B$		Involution
T5	$B \bullet \bar{B} = 0$	T5′	$B + \bar{B} = 1$	Complements

ELSEVIER

Булевы теоремы нескольких переменных

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

Упрощение булевых

Пример 1:

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

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

Упрощение булевых

Пример 1:

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

  $= B(A + \overline{A})$ T8

  $= B(1)$ T5'

  $= B$ T1

ELSEVIER

Упрощение булевых

Пример 2:

- $Y = A(AB + ABC)$

ELSEVIER

Упрощение булевых

Пример 2:

- $Y = A(AB + ABC)$

$= A(AB(1 + C))$ T8

$= A(AB(1))$ T2'

$= A(AB)$ T1

$= (AA)B$ T7

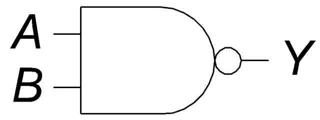
$= AB$ T3

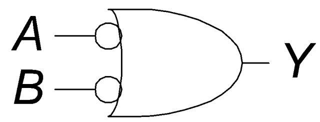
© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

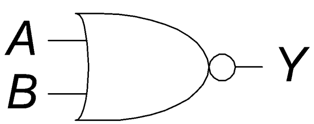
Теорема де Моргана

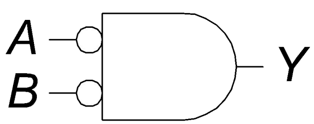
- $Y = \overline{AB} = \overline{A} + \overline{B}$





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



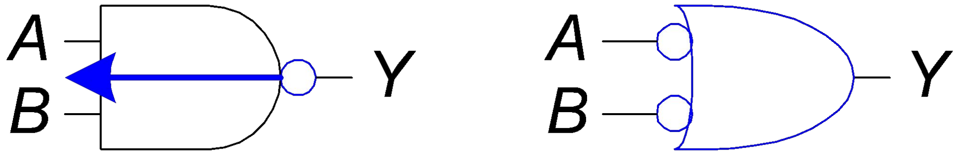


© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

Перемещение инверсии

- **Назад:**
  - Изменить тип элемента
  - Добавить инверсию на входы



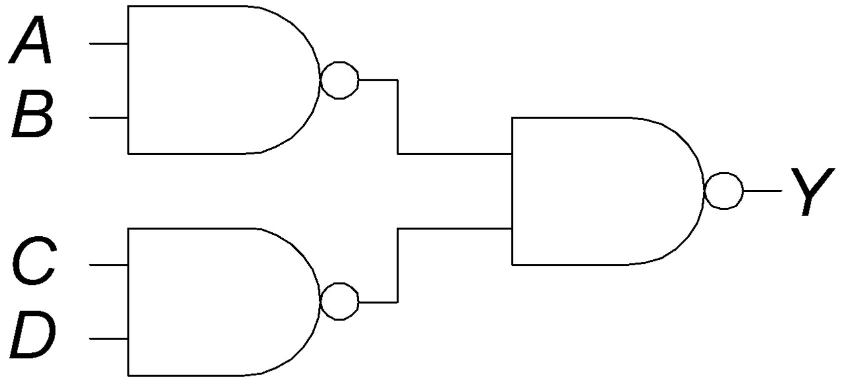
- **Вперед:**
  - Изменить тип элемента
  - Добавить инверсию на выход

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

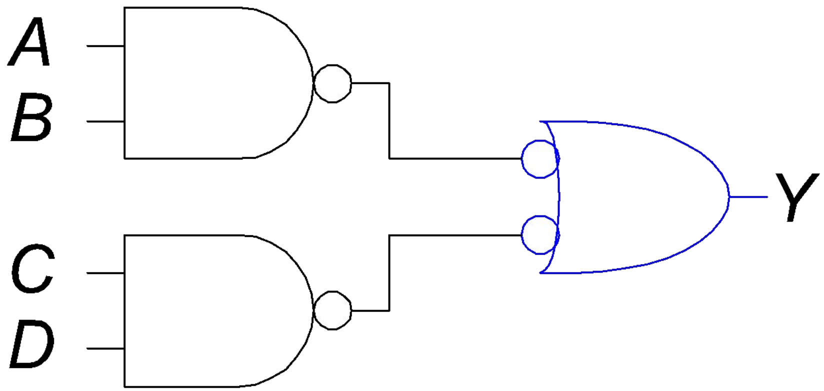
Перемещение инверсии

- Запишите булево выражение для этой схемы



Перемещение инверсии

- Запишите булево выражение для этой схемы



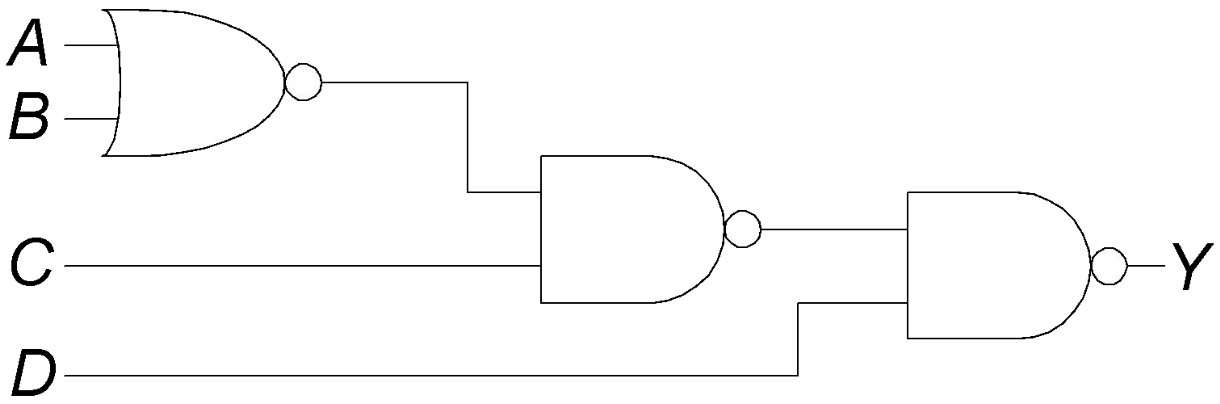
$$Y = AB + CD$$

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

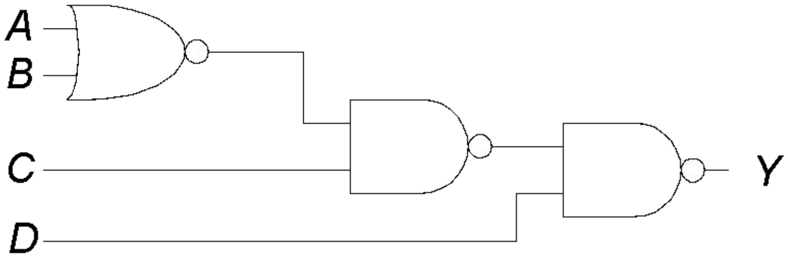
ELSEVIER

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

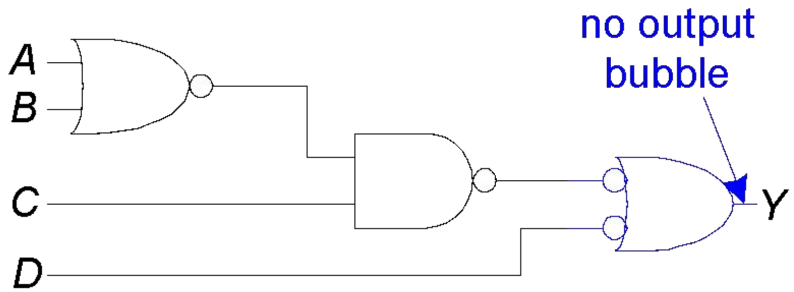


Пример перемещения инверсии



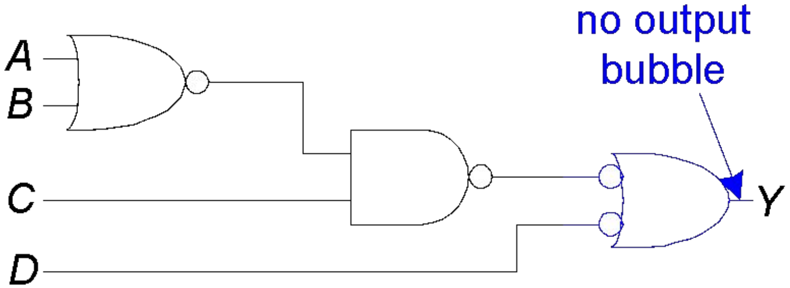
© Digital Design and Computer Architecture, 2nd Edition, 2012

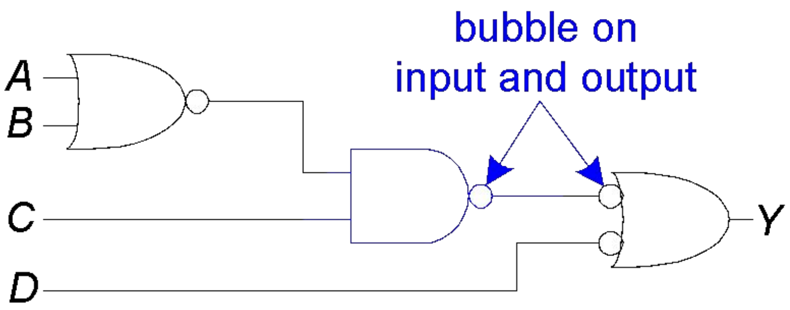
Пример перемещения инверсии



ELSEVIER

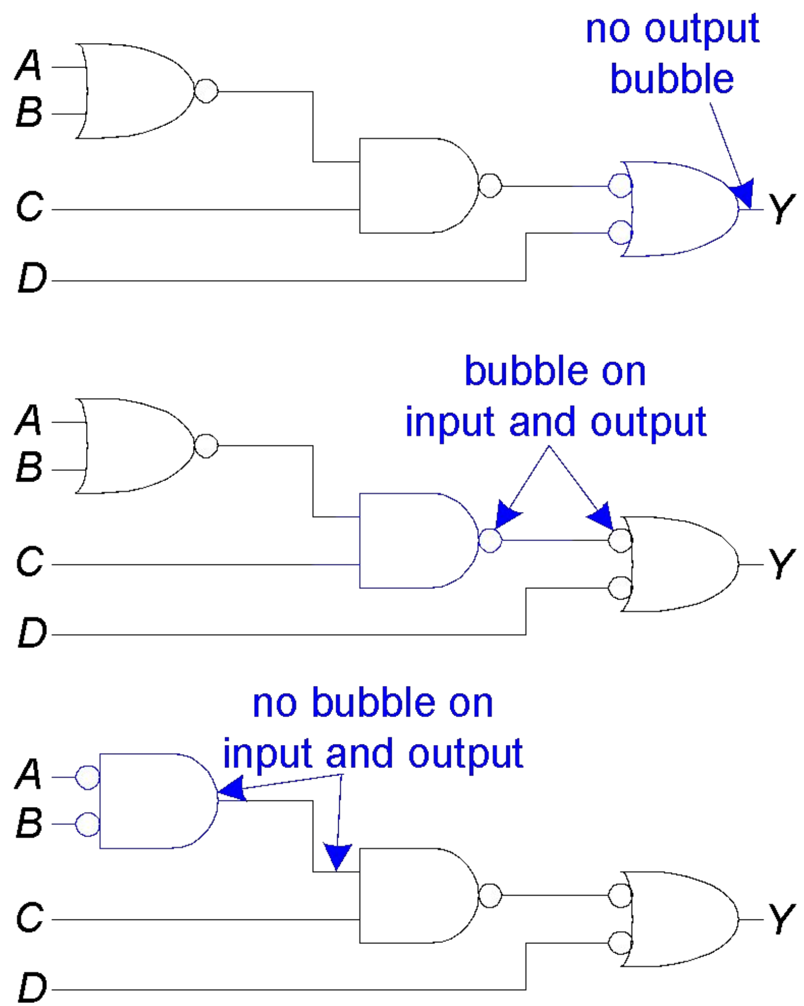
Пример перемещения инверсии





© *Digital Design and Computer Architecture*, 2nd Edition, 2012

Пример перемещения инверсии

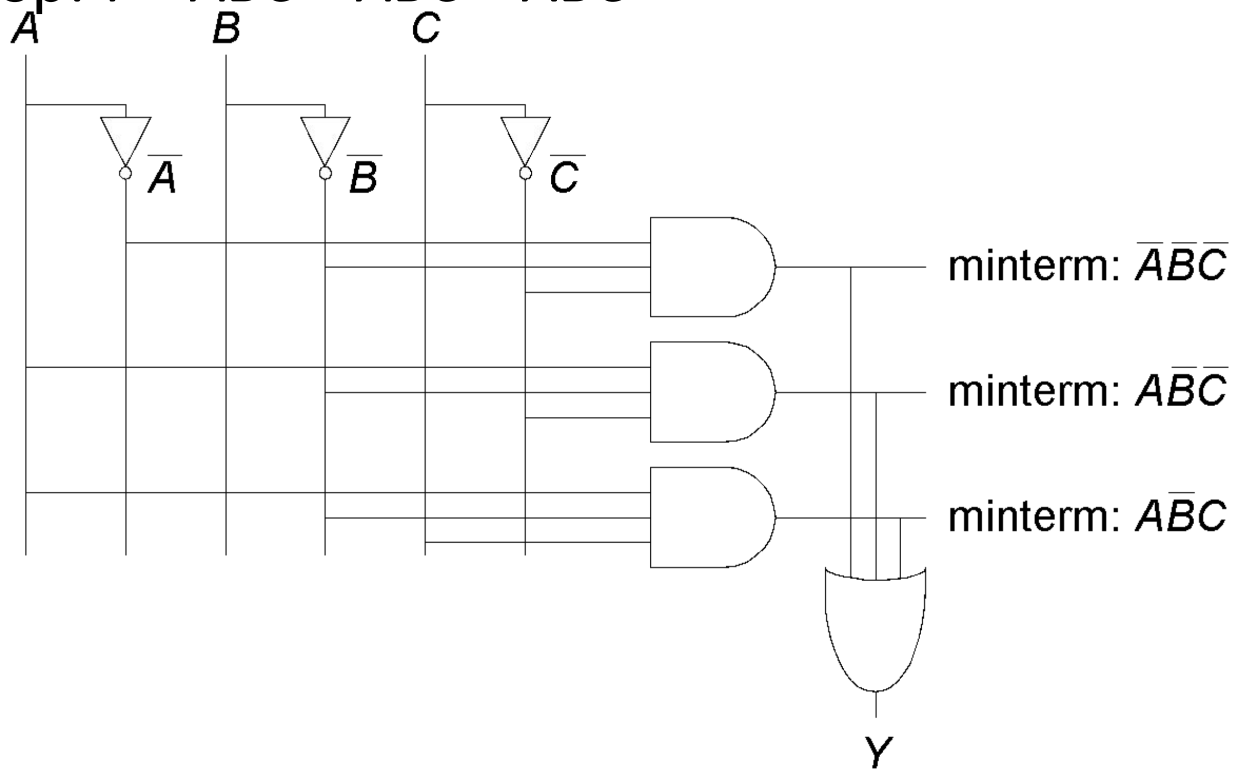


$$Y = \overline{A}\,\overline{B}C + \overline{D}$$

© Digital Design and Computer Architecture, 2nd Edition, 2012

От логики к логическим элементам

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



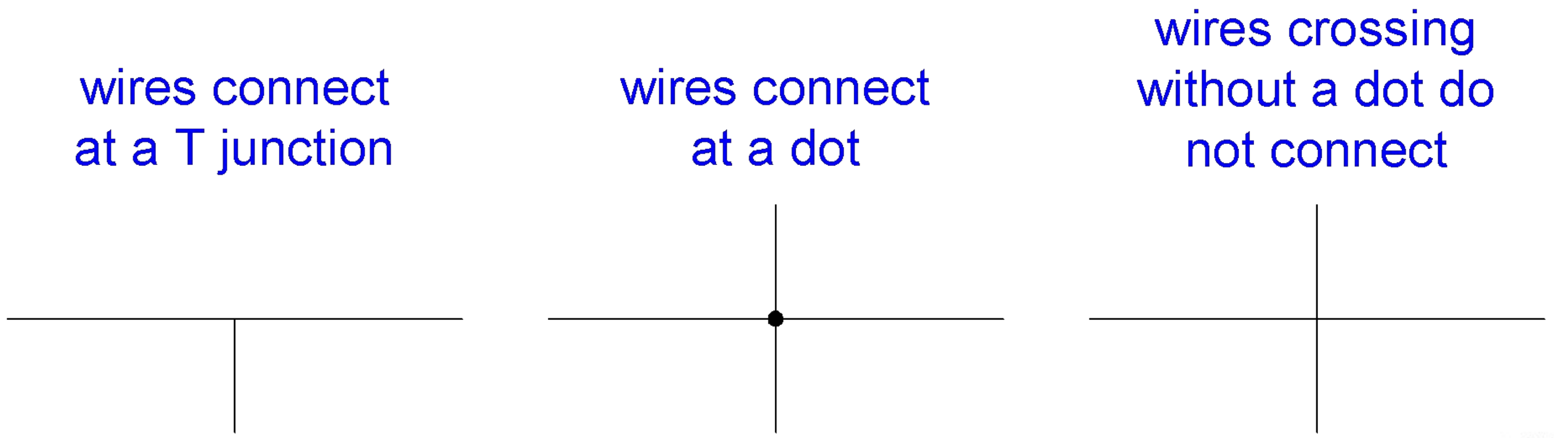
© Digital Design and Computer Architecture, 2nd Edition, 2012

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



© *Digital Design and Computer Architecture*, 2nd Edition, 2012

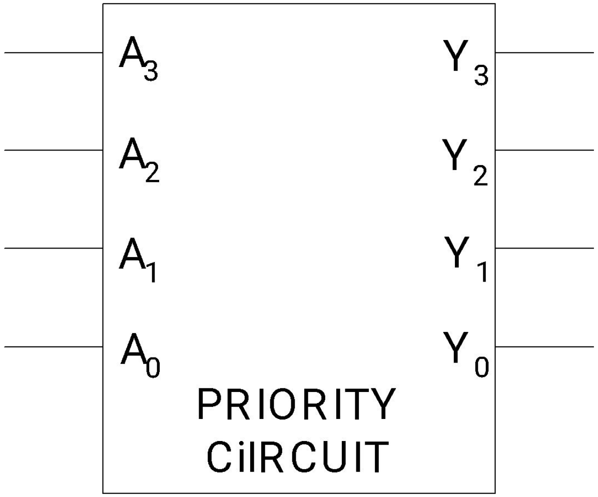
ELSEVIER

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				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				



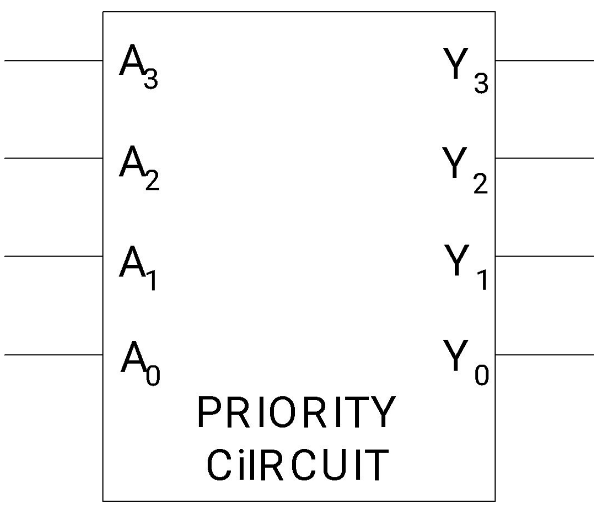
ELSEVIER

Схемы с несколькими

- **Пример: Схема приоритета**

Выход устанавливается в соответствии со старшим входом который равен ИСТИНЕ

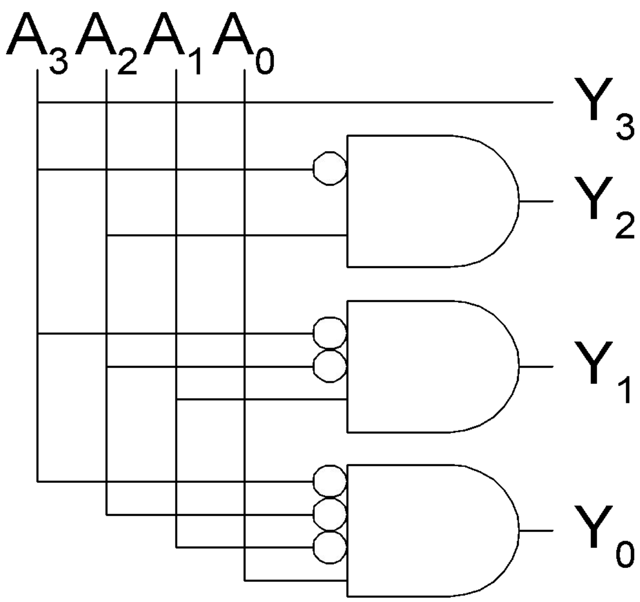
A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	1	**0**	**0**	**0**	**1**
0	0	1	0	**0**	**0**	**1**	**0**
0	0	1	1	**0**	**0**	**1**	**0**
0	1	0	0	**0**	**1**	**0**	**0**
0	1	0	1	**0**	**1**	**0**	**0**
0	1	1	0	**0**	**1**	**0**	**0**
0	1	1	1	**0**	**1**	**0**	**0**
1	0	0	0	**1**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



© Digital Design and Computer Architecture, 2nd Edition, 2012

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**



© Digital Design and Computer Architecture, 2nd Edition, 2012

Безразличное значение

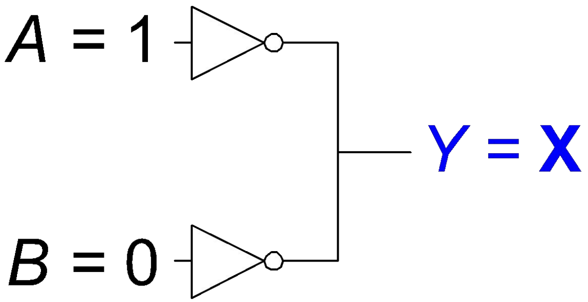
A_3	A_2	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	1	**0**	**0**	**0**	**1**
0	0	1	0	**0**	**0**	**1**	**0**
0	0	1	1	**0**	**0**	**1**	**0**
0	1	0	0	**0**	**1**	**0**	**0**
0	1	0	1	**0**	**1**	**0**	**0**
0	1	1	0	**0**	**1**	**0**	**0**
0	1	1	1	**0**	**1**	**0**	**0**
1	0	0	0	**1**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

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

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 диапазоне
  - Может зависеть от напряжения, температуры, времени, шума
  - Часто вызывает большое энергопотребление



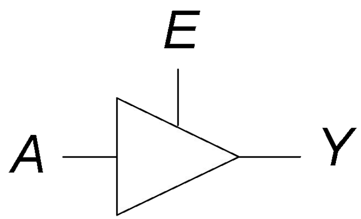
- **Предупреждение:**
  - Конфликт обычно является **ошибкой**
  - В зависимости от контекста **X обозначает безразличное значение или конфликт**

© Digital Design and Computer Architecture, 2nd Edition, 2012

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



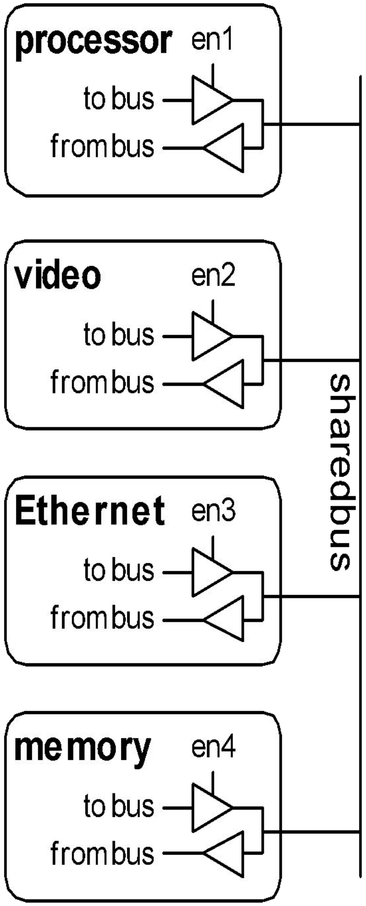
E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

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
- $PA + P\overline{A} = P$

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

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

© Digital Design and Computer Architecture, 2nd Edition, 2012

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

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

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

© Digital Design and Computer Architecture, 2nd Edition, 2012

Карты Карно - три входа

Y: C \ AB	00	01	11	10
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	$A\bar{B}C$

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

Y: C \ AB	00	01	11	10
0				
1				

Карты Карно - три входа

Y: C \ AB	00	01	11	10
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	$A\bar{B}C$

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

Y: C \ AB	00	01	11	10
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

  $\bar{A}, \bar{B}, \bar{C}$

- **Литерал:** Литерал: переменная или ее дополнение

  $\bar{A}, A, \bar{B}, B, C, \bar{C}$

- **Импликанта:** произведение литералов

  $A\bar{B}C, A\bar{C}, BC$

- **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

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	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

Y

CD \ AB	00	01	11	10
00				
01				
11				
10				

Карты Карно - четыре входа

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	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

Y

CD \ *AB*	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

ELSEVIER

Карты Карно - четыре входа

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	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

Y

$CD \backslash AB$	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

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

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

CD \ *AB*	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

CD \ *AB*	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

© *Digital Design and Computer Architecture*, 2nd Edition, 2012

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

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

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

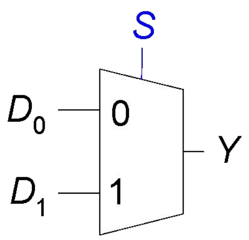
Базовые комбинационные блоки

- Мультиплексоры
- Дешифраторы

Мультиплексор (Mux)

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

2:1 Mux

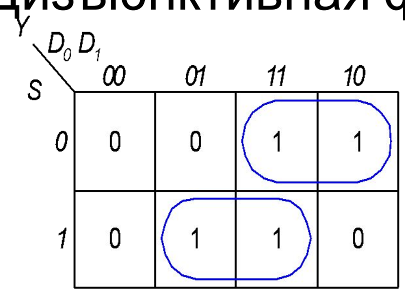


S	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

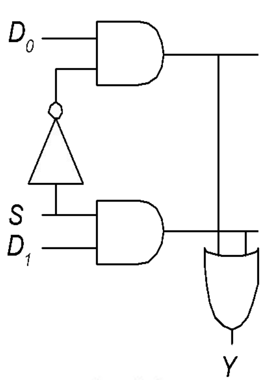
S	Y
0	D_0
1	D_1

Реализация

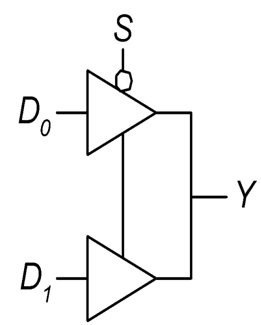
- **Используя логические элементы**
  - Дизъюнктивная форма



$Y = D_0\overline{S} + D_1S$



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



© Digital Design and Computer Architecture, 2nd Edition, 2012

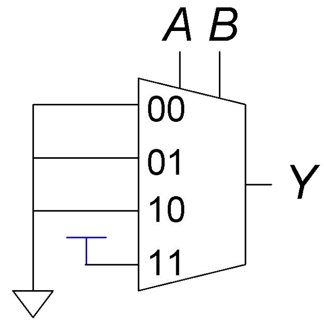
ELSEVIER

Цифровые схемы на основе мультиплексоров

- Использование мультиплексоров как таблиц преобразования

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

$$Y = AB$$



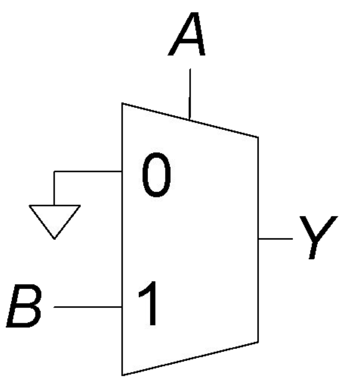
Цифровые схемы на основе мультиплексоров

- Уменьшение размера мультиплексора

$Y = AB$

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

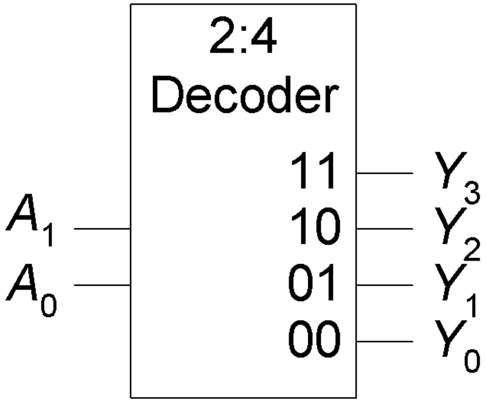
A	Y
0	0
1	B



© Digital Design and Computer Architecture, 2nd Edition, 2012

Дешифраторы

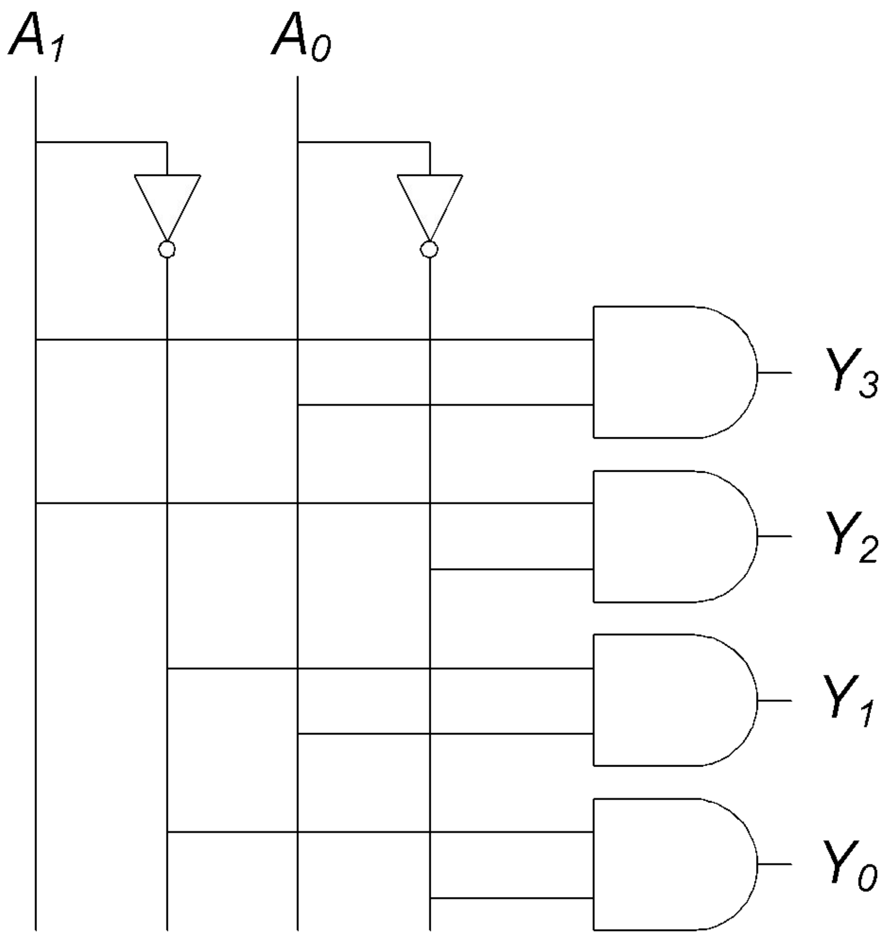
- N входов, 2^N выходов
- Прямой унитарный код: только один выход принимает значение ИСТИНА



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

ELSEVIER

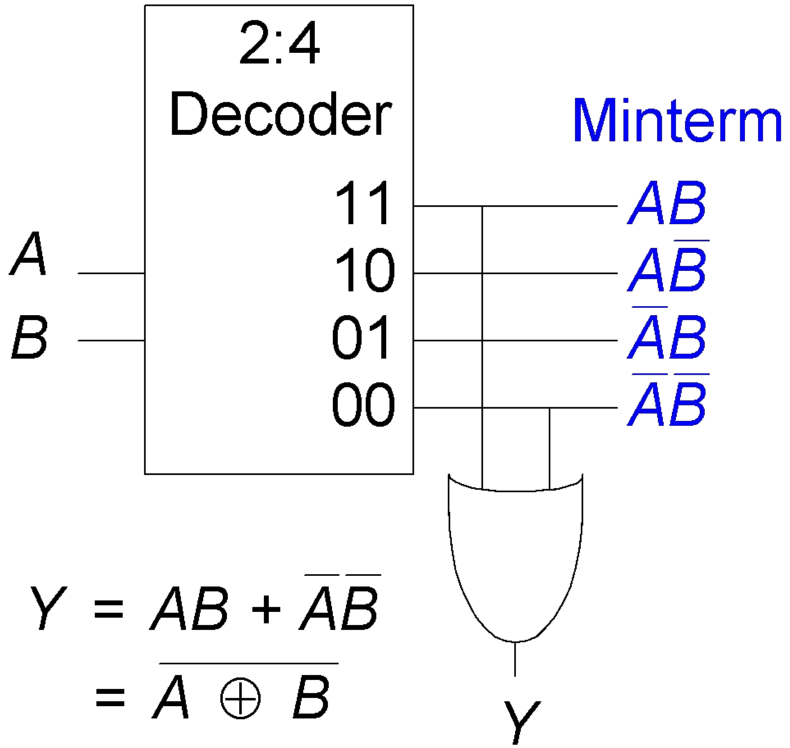
Реализация дешифраторов



© *Digital Design and Computer Architecture*, 2nd Edition, 2012

Цифровые схемы на основе дешифраторов

- Функция ИЛИ от минтермов

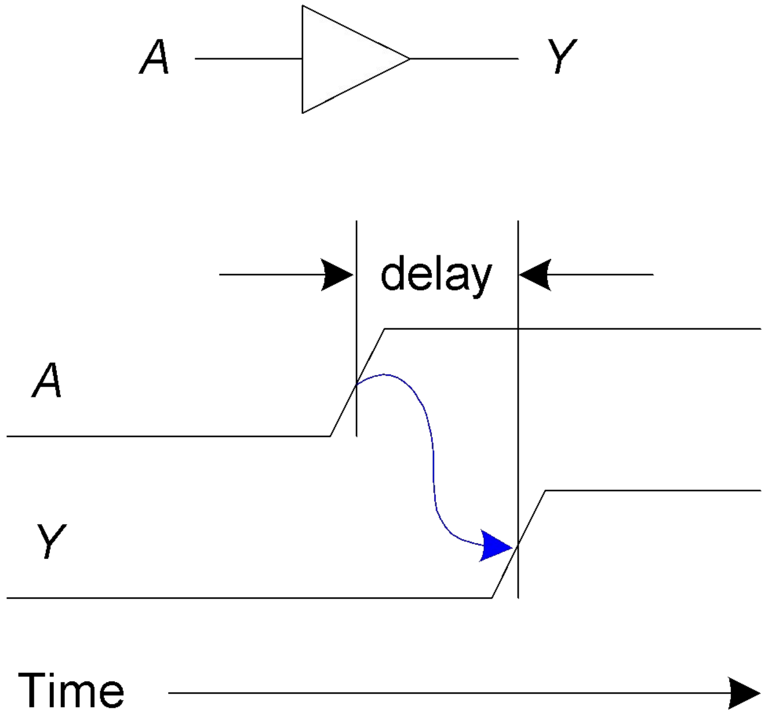


$$Y = AB + \bar{A}\bar{B}$$
$$= \overline{A \oplus B}$$

ELSEVIER

Временные характеристик

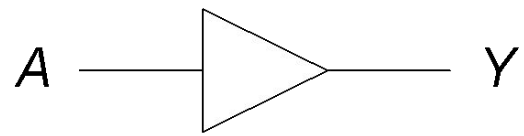
- Задержка между изменением входа и соответствующим изменением выхода
- Как проектировать быстрые схемы?

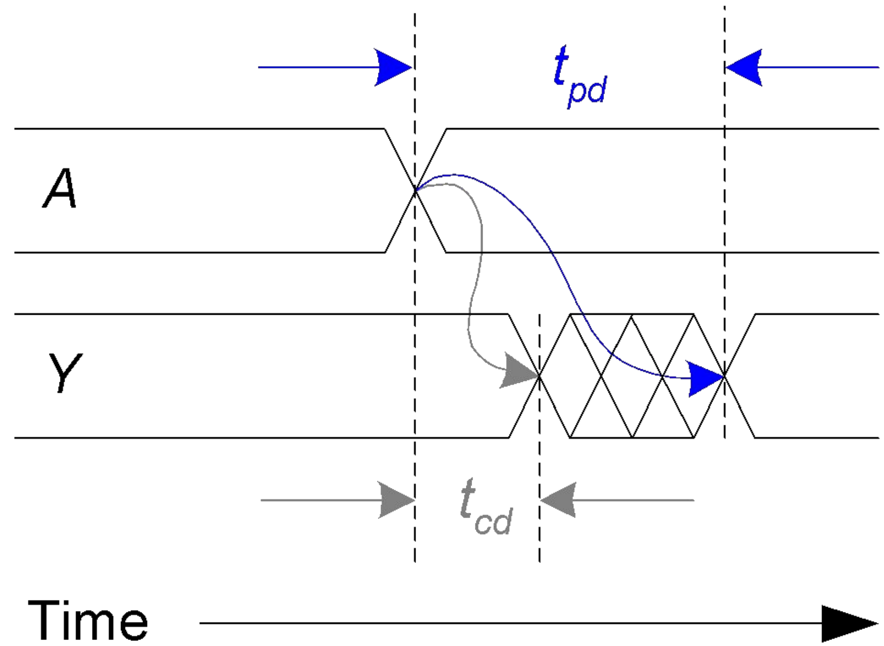


ELSEVIER

Задержки распространения и реакции

- **Задержка распространения** t_{pd} = максимальная задержка тракта вход-выход
- **Задержка реакции** t_{cd} = минимальная задержка тракта вход-выход



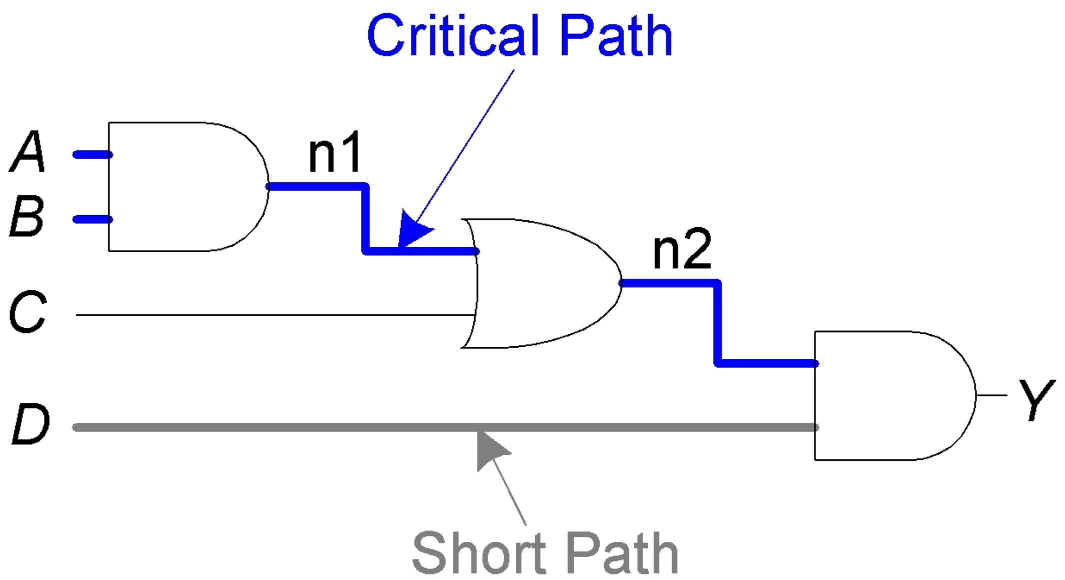


© Digital Design and Computer Architecture, 2nd Edition, 2012

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 $t_{cd} = t_{cd\_AND}$

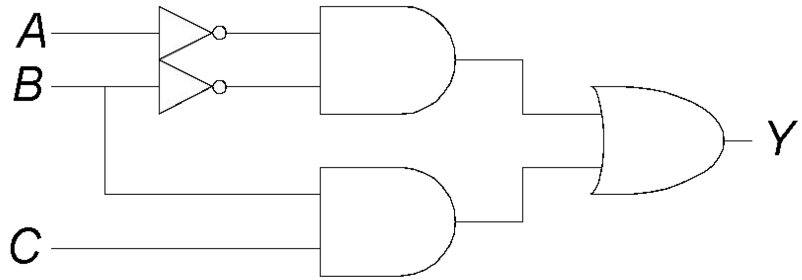
Импульсные помехи

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

ELSEVIER

Пример импульсной помехи

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



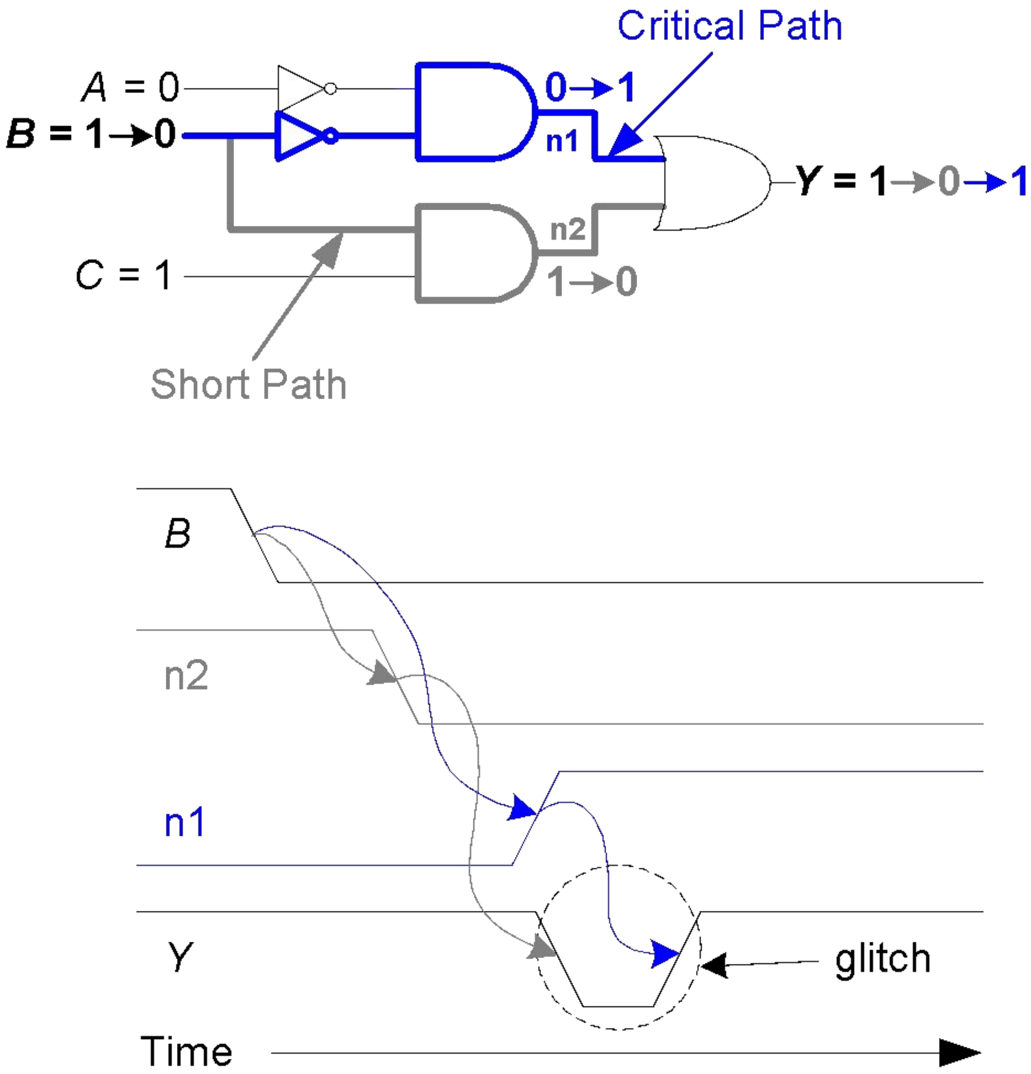
Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	1	1	0

$$Y = \overline{A}\,\overline{B} + BC$$

© Digital Design and Computer Architecture, 2nd Edition, 2012

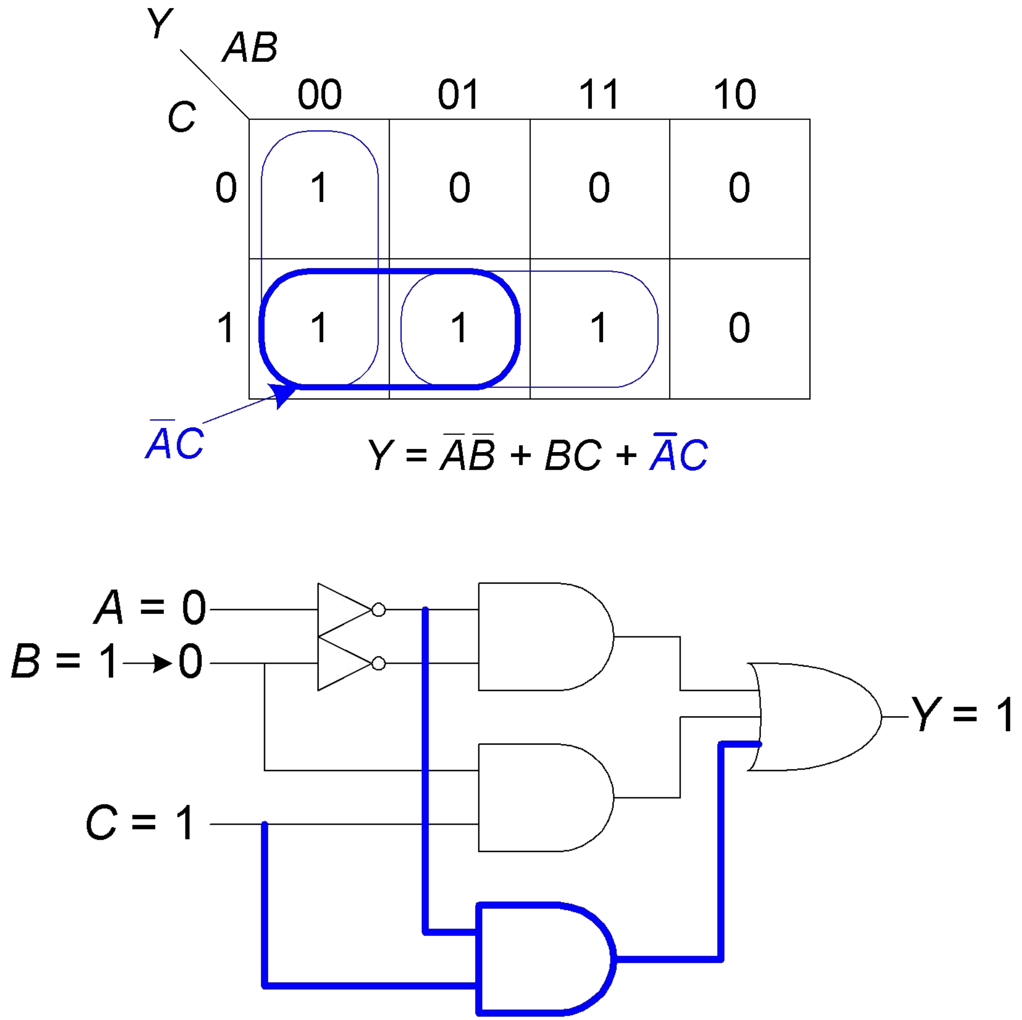
ELSEVIER

Пример импульсной помехи



ELSEVIER

Борьба с импульсными



© Digital Design and Computer Architecture, 2nd Edition, 2012

ELSEVIER

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