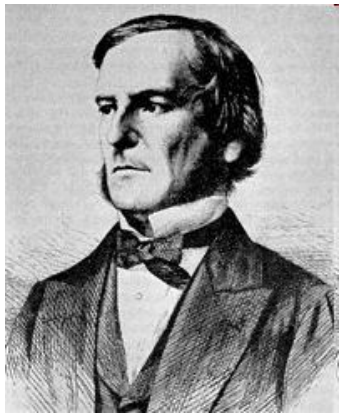
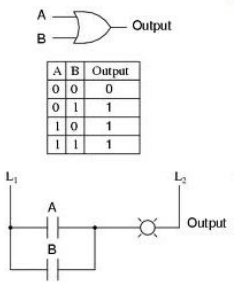
Lection №9: Цифровой логический уровень

A	B	C	A+B	¬(A+B)	¬(A+B)↔C	AB	(¬(A+B)↔C)→AB
0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	0
0	1	0	1	0	1	0	0
0	1	1	1	0	0	0	1
1	0	0	1	0	1	0	0
1	0	1	1	0	0	0	1
1	1	0	1	0	1	1	1
1	1	1	1	0	0	1	1

UCC
Coláiste na hOllscoile Corcaigh, Éire
University College Cork, Ireland



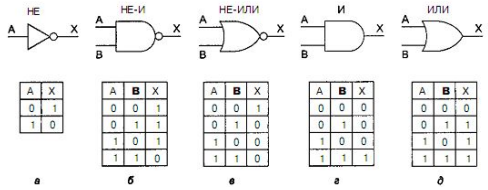


Джордж Буль (George Boole; 02.11.1815, Линкольн — 08.12.1864, Баллинтемпл, графство Корк, Ирландия) — английский математик и логик.
Профессор математики Королевского колледжа Корка (ныне Университетский колледж Корк) с 1849.

Один из основоположников математической логики

Буль показал, как из любого числа высказываний, включающих любое число терминов, вывести любое заключение, следующее из этих высказываний, путём чисто символических манипуляций..

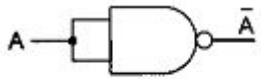
Четыре их дочери снискали известность как учёные (геометр Алисия, химик Люси), или члены учёных семей (Мэри, жена математика и писателя Ч. Г. Хинтона, и Маргарет, мать математика Дж. И. Тейлора), а пятая — Этель Лилиан Войнич — прославилась как писатель.

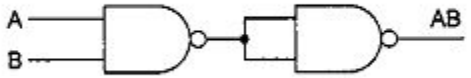


Буль умер на пятидесятом году жизни от воспаления лёгких.

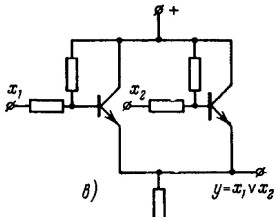
- Логические элементы и таблицы истинности
- Логические операции

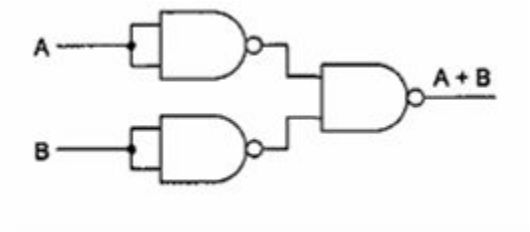
- **Законы алгебры логики:**
  - закон одинарных элементов;
  - законы отрицания;
  - комбинационные законы;
  - правило поглощения;
  - правило склеивания.

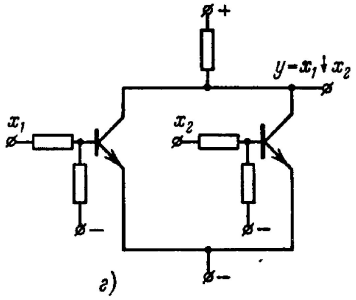


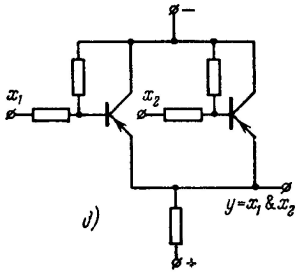


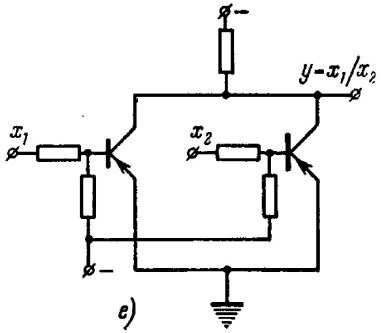
- **Свойства логических операций**



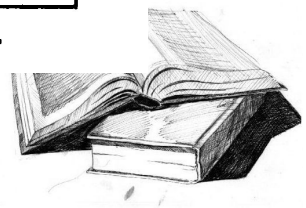








- Резюме к лекции и список используемой литературы



Логические элементы и таблицы истинности

Абсолютно все цифровые микросхемы состоят из одних и тех же логических элементов – «кирпичиков» любого цифрового узла

Логический элемент – это такая схема, у которой несколько входов и один выход.
Каждому состоянию сигналов на входах, соответствует определенный сигнал на выходе.

Таблица истинности это табличное представление логической схемы (операции), в котором перечислены все возможные сочетания значений истинности входных сигналов (операндов) вместе со значением истинности выходного сигнала (результата операции) для каждого из этих сочетаний.

Обычно сигнал от 0 до 1 В представляет одно значение (н-р, 0), а сигнал от 2 до 5 В — другое значение (н-р, 1).

Электронные устройства, которые называются **вентилями**, могут вычислять различные функции от этих двузначных сигналов.

Входы сравнения данных				Входы управления			Выходы		
A_0, B_0	A_1, B_1	A_2, B_2	A_3, B_3	I(A>B)	I(A=B)	I(A<B)	A>B	A=B	A<B
$A_1 > B_1$	*	*	*	*	*	*	1	0	0
$A_1 < B_1$	*	*	*	*	*	*	0	0	1
$A_1 = B_1$	$A_2 > B_2$	*	*	*	*	*	1	0	0
$A_1 = B_1$	$A_2 < B_2$	*	*	*	*	*	0	0	1
$A_1 = B_1$	$A_2 = B_2$	$A_3 > B_3$	*	*	*	*	1	0	0
$A_1 = B_1$	$A_2 = B_2$	$A_3 < B_3$	*	*	*	*	0	0	1
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 > B_0$	*	*	*	1	0	0
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 < B_0$	*	*	*	0	0	1
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 = B_0$	*	1	*	0	1	0
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 = B_0$	1	0	0	1	0	0
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 = B_0$	0	0	1	0	0	1
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 = B_0$	0	0	0	1	0	1
$A_1 = B_1$	$A_2 = B_2$	$A_3 = B_3$	$A_0 = B_0$	1	0	1	0	0	0

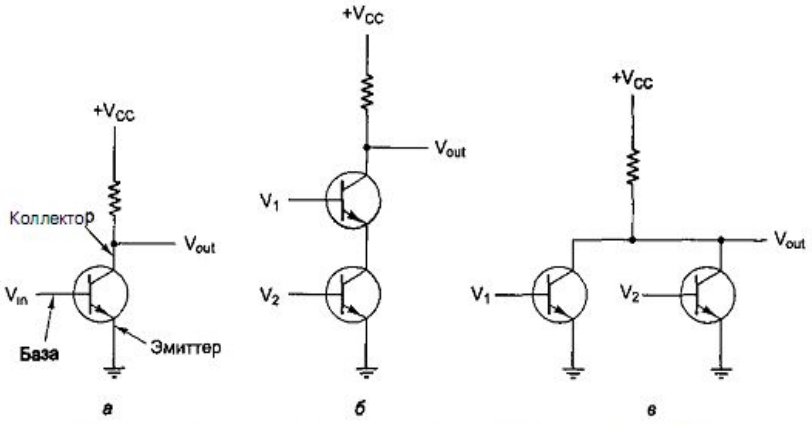
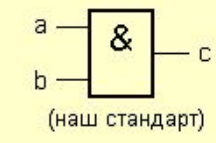


Рис. 3 . 1 . Транзисторный инвертор (а); вентиль НЕ-И (б); вентиль НЕ-ИЛИ (в)

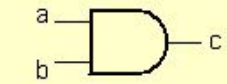
Логические элементы и таблицы истинности

Элемент «И» (AND)

Иначе его называют «конъюнктор».



(наш стандарт)



(буржуйский стандарт)

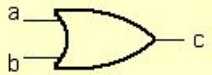
a b	c
0 0	0
0 1	0
1 0	0
1 1	1

Понять его не сложно: “1” на выходе элемента «И» возникает только тогда, когда на оба входа поданы единицы. Это объясняет название элемента: единицы должны быть И на одном, И на другом входе.

Элемент «ИЛИ» (OR)

Иначе его называют «дизъюнктор».

(наш стандарт)



(буржуйский стандарт)

a b	c
0 0	0
0 1	1
1 0	1
1 1	1

На выходе возникает “1”, когда на один ИЛИ на другой ИЛИ на оба сразу входа подана единица.

Операция, выражаемая связкой *"и"*, называется ***конъюнкцией*** (лат. conjunctio — соединение) или ***логическим умножением*** и обозначается точкой **" · "** (может также обозначаться знаками Λ или **&**).

X	Y	X*Y
0	0	0
1	0	0
0	1	0
1	1	1

Операция, выражаемая связкой *"или"* (в неисключающем смысле этого слова), называется ***дизъюнкцией*** (лат. disjunctio — разделение) или ***логическим сложением*** и обозначается знаком **v** (или **плюсом**).

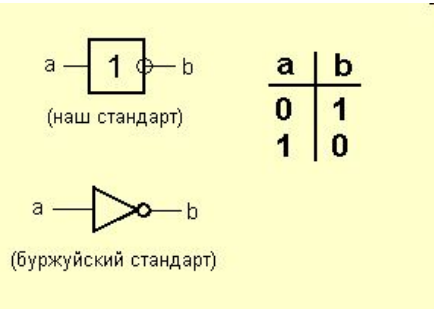
x	y	x v y
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

http://cxem.net/beginner/beginner28.php

Логические элементы и таблицы истинности

Элемент «НЕ» (NOT)

Иначе его называют «инвертор»

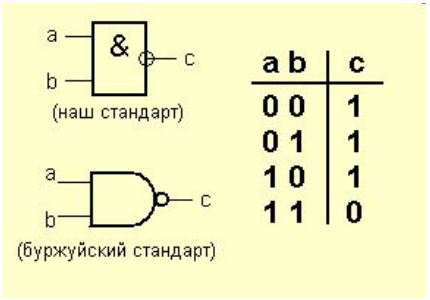


a	b
0	1
1	0

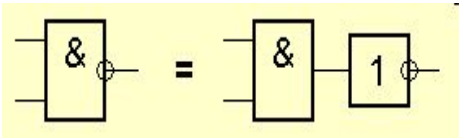
Операция, выражаемая словом "не", называется ***<u>инверсией</u>*** или ***отрицанием*** и обозначается чертой над высказыванием.

X	¬ X
0	1
1	0

Элемент «И-НЕ» (NAND)

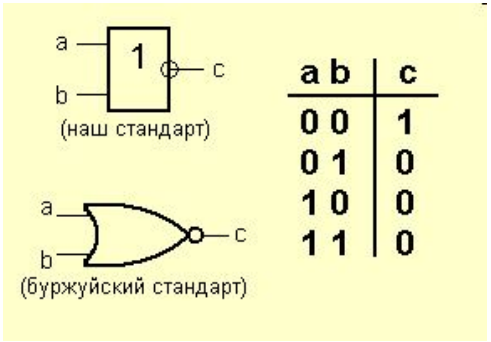


a	b	c
0	0	1
0	1	1
1	0	1
1	1	0



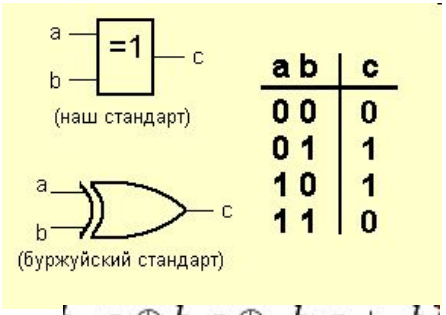
Элемент И-НЕ работает точно так же как «И», только выходной сигнал полностью противоположен. Там где у элемента «И» на выходе должен быть «0», у элемента «И-НЕ» - единица. И наоборот.

Элемент «ИЛИ-НЕ» (NOR)



a	b	c
0	0	1
0	1	0
1	0	0
1	1	0

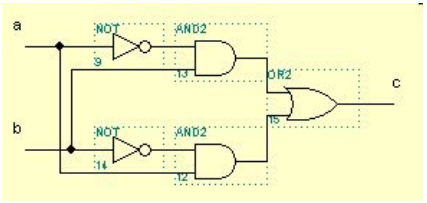
Элемент «Исключающее ИЛИ» (XOR)



a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

Операция, которую он выполняет, часто называют «сложение по модулю 2». На самом деле, на этих элементах строятся цифровые сумматоры.

$b,\ a \oplus b, a \oplus_2 b, a +_2 b,$ $a \neq b, (a,b)\oplus_2, a\ XOR\ b$



Логические операции

ЕСЛИ-ТО

Операция, выражаемая связками *"если ..., то", "из ... следует", "... влечет ..."*, называется ***импликацией*** (лат. *implico* — тесно связаны) и обозначается знаком $\longrightarrow$. Высказывание **A** $\rightarrow$ **B** ложно тогда и только тогда, когда **A** истинно, а **B** ложно.

РАВНОСИЛЬНО

Операция, выражаемая связками "*тогда и только тогда*", "*необходимо и достаточно*", "*... равносильно ...*", называется ***эквиваленцией*** или ***двойной импликацией*** и обозначается знаком $\leftrightarrow$ или $\sim$.

Высказывание **A** $\leftrightarrow$ **B** истинно тогда и только тогда, когда значения **A** и **B** совпадают.

АКСИОМЫ алгебры логики описывают действие логических функций "И" и "ИЛИ" и записываются следующими выражениями:

0 * 0 = 0; 0 * 1 = 0; 1 * 0 = 0; 1 * 1 = 1;

0 + 0 = 0; 0 + 1 = 1; 1 + 0 = 1; 1 + 1 = 1;

Также как в обычной математике в алгебре логики имеется старшинство операций.

При этом первым выполняется:

1. Действие в скобках
2. Операция с одним операндом (одноместная операция) – НЕ
3. Конъюнкция - И
4. Дизъюнкция - ИЛИ
5. Сумма по модулю два.

http://digital.sibsutis.ru/digital/AlgLog.htm

Логические операции

В математических выражениях Ложь отождествляется с логическим нулём, а Истина — с логической единицей, а операции отрицания (НЕ), конъюнкции (И) и дизъюнкции (ИЛИ) определяются в привычном нам понимании.

$$x \mid y = \overline{x * y} = \bar{x} + \bar{y}$$

Штрих Ше́ффера — бинарная логическая операция, булева функция над двумя переменными. (это логический элемент «и-не»)

X	Y	X | Y
0	0	1
0	1	1
1	0	1
1	1	0

Как и любую булеву операцию, штрих Шеффера можно выразить через отрицание и дизъюнкцию:

$$(x_1 \mid x_2) = (\neg x_1 \vee \neg x_2)$$
$$(x_1 \mid x_2) = \neg (x_1 \wedge x_2)$$

$$x_1 \mid x_1 = (\neg x_1) \vee (\neg x_1) = \neg x_1$$ ⇐ Отрицание

$$(x_1 \mid x_1) \mid (x_2 \mid x_2) = \neg (\neg x_1) \vee \neg (\neg x_2) = x_1 \vee x_2$$ ⇐ Дизъюнкция

$$(x_1 \mid x_2) \mid (x_1 \mid x_2) = \neg (\neg x_1 \vee \neg x_2) \vee \neg (\neg x_1 \vee \neg x_2) =$$

$$= (x_1 \wedge x_2) \vee (x_1 \wedge x_2) = (x_1 \wedge x_2)$$ ⇐ Конъюнкция

http://ru.wikipedia.org/wiki/Алгебра_логики

Логические операции

Стрелка Пи́рса (символ Лукашевича) — бинарная логическая операция, введена в рассмотрение Ч. Пирсом (Ch. Peirce). Стрелка Пирса, обычно обозначаемая ↓, задаётся следующей таблицей истинности:

X	Y	X↓Y
0	0	1
0	1	0
1	0	0
1	1	0

Таким образом, высказывание «A ↓ B» означает «ни A, ни B». Стрелка Пирса обладает тем свойством, что через неё одну выражаются все другие логические операции:

$\neg x \equiv x \downarrow x$

$x \,\&\, y \equiv (x \downarrow x) \downarrow (y \downarrow y)$

$x \vee y \equiv (x \downarrow y) \downarrow (x \downarrow y)$

$x \rightarrow y \equiv ((x \downarrow x) \downarrow y) \downarrow ((x \downarrow x) \downarrow y)$

От перемены мест операндов результат операции не изменяется.

В логических схемах носит название "операция ИЛИ-НЕ", комбинация которых позволяет заменить любой элемент схемы.

$$x \downarrow y = \overline{x + y} = \bar{x} * \bar{y}$$

http://ru.wikipedia.org/wiki/Алгебра_логики

Законы алгебры логики

1. Закон одинарных элементов:

1 * X = X

0 * X = 0

- могут быть полезны при построении коммутаторов, ведь подавая на один из входов элемента “2И” логический «0» или «1» можно либо пропускать сигнал на выход, либо формировать на выходе нулевой потенциал;
- второй вариант использования этих выражений заключается в возможности избирательного обнуления определённых разрядов многоразрядного числа. При поразрядном применении операции "И" можно либо оставлять прежнее значение разряда, либо обнулять его, подавая на соответствующие разряды единичный или нулевой потенциал.

Закон одинарных элементов: пример

1 * X = X
0 * X = 0

Например, требуется обнулить 6, 3 и 1 разряды.
Тогда:

```
  11010111
^ 10110101
----------
  10010101
```

● = * = & = ∧

Видно, что для обнуления необходимых разрядов в маске (нижнее число) на месте соответствующих разрядов записаны нули, в остальных разрядах записаны единицы.

- В исходном числе (верхнее число) на месте 6 и 1 разрядов находятся единицы.
- После выполнения операции "И" на этих местах появляются нули.
- На месте третьего разряда в исходном числе находится «0».
- В результирующем числе на этом месте тоже присутствует «0».
- Остальные разряды, как и требовалось по условию задачи, не изменены.

Закон одинарных элементов

$$1 + X = 1$$
$$0 + X = X$$

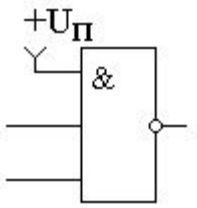
● Точно так же можно записывать единицы в нужные нам разряды. В этом случае необходимо воспользоваться нижними двумя выражениями закона одинарных элементов.

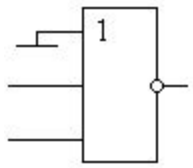
При поразрядном применении операции “ИЛИ” можно либо оставлять прежнее значение разряда, либо обнулять его, подавая на соответствующие разряды нулевой или единичный потенциал.

Пусть требуется записать единицы в 7 и 6 биты числа. Тогда:

```
 01010111
 11000000
----------
 11010111
```

Здесь в маску (нижнее число) мы записали единицы в седьмой и шестой биты.
Остальные биты содержат нули, и, =>, не могут изменить первоначальное состояние исходного числа, что мы и видим в результирующем числе под чертой.





Это позволяют использовать логические элементы с большим количеством входов в качестве элементов с меньшим количеством входов.
Для этого неиспользуемые входы в схеме “И” должны быть подключены к источнику питания

Законы алгебры логики

2. Законы отрицания:

a. Закон дополнительных элементов: $X + \neg X = 1$; $X * \neg X = 0$

Выражения этого закона широко используется для минимизации логических схем.

Если удаётся выделить из общего выражения логической функции такие подвыражения, то можно сократить необходимое количество входов элементов цифровой схемы, а иногда и вообще свести всё выражение к логической константе.

b. Двойное отрицание: $\neg 1 = 0$; $\neg 0 = 1$; $\neg \neg 1 = 1$; $\neg \neg 0 = 0$.

$$\neg x = \overline{x}$$

Законы отрицания

c. Закон отрицательной логики (Законы де Моргана):

$$\overline{a+b+c} = a \bullet b \bullet c$$

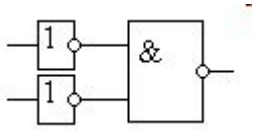
Закон отрицательной логики **справедлив для *любого* числа переменных.**

$$\overline{a \bullet b \bullet c} = a+b+c$$

Этот закон позволяет реализовывать логическую функцию "И" при помощи логических элементов “ИЛИ” и наоборот: реализовывать логическую функцию “ИЛИ” при помощи логических элементов “И”.

Это особенно полезно в схемотехнике, т.к. там легко реализовать логические элементы “И”, но при этом достаточно сложно логические элементы “ИЛИ”.

Благодаря закону отрицательной логики можно реализовывать элементы “ИЛИ” на логических элементах “И”.



Логический элемент “2ИЛИ”, реализованный на элементе “2И-НЕ” и двух инверторах.

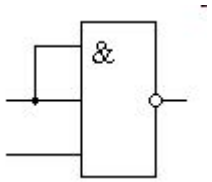
Законы алгебры логики

3. Комбинационные законы. Комбинационные законы алгебры логики во многом соответствуют комбинационным законам обычной алгебры, но есть и отличия.

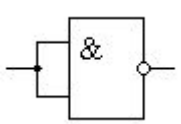
a. Закон тавтологии (многократное повторение): X+X+X+X=X; X*X*X*X=X

Этот закон позволяет использовать логические элементы с большим количеством входов в качестве элементов с меньшим количеством входов.

Например, можно реализовать двухвходовую схему 2И на элементе 3И:



или использовать схему 2И-НЕ в качестве обычного инвертора:



Однако следует предупредить, что объединение нескольких входов увеличивает входные токи логического элемента и его ёмкость, что увеличивает ток потребления предыдущих элементов и отрицательно сказывается на быстродействии цифровой схемы в целом.

http://digital.sibsutis.ru/digital/AlgLog.htm

Комбинационные законы

b. закон переместительности: $A+B+C+D=A+C+B+D$

c. закон сочетательности:

$A+B+C+D=A+(B+C)+D=A+B+(C+D)$

d. закон распределительности: $X_1(X_2+X_3)= X_1X_2 + X_1X_3$

Доказательство:

$X_1+X_2X_3=(X_1+X_2)(X_1+X_3)=$/ докажем это путём раскрытия скобок/=
$X_1X_1+ X_1X_3+ X_1X_2+ X_2X_3= X_1(1+X_3+X_2)+ X_2X_3= X_1+X_2X_3$

http://digital.sibsutis.ru/digital/AlgLog.htm

Законы алгебры логики

4. Правило поглощения
(одна переменная поглощает другие):

$$X_1+X_1X_2\,X_3=X_1(1+X_2\,X_3)=X_1$$

5. Правило склеивания
(выполняется только по одной переменной)

$$A\bar{B}C+ABC=AC(\underbrace{\bar{B}+B}_{1})=AC$$

Свойства логических операций

1. $x * \bar{x} = x * 0 = x \oplus x = 0$

2. $x + \bar{x} = x + 1 = x \sim x = x \to x = 1$

3. $x + x = x * x = x * 1 = x + 0 = x \oplus 0 = x$

4. $x \oplus 1 = x \to 0 = x \sim 0 = x \mid x = x \downarrow x = \bar{x}$

5. $x \oplus y = x * \bar{y} + \bar{x} * y = (x + y) * (\bar{x} + \bar{y})$

6. $x \sim y = \overline{x \oplus y} = 1 \oplus x \oplus y = x * y + \bar{x} * \bar{y} = (x + \bar{y}) * (\bar{x} + y)$

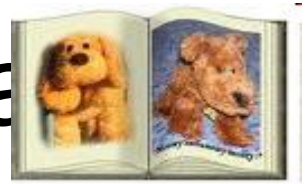
7. $x \to y = \bar{x} + y = x * y \oplus x \oplus 1$

8. $x + y = x \oplus y \oplus x * y$

Используемые Интернет-ресурсы:

http://de.ifmo.ru/--books/electron/cpu-cod.htm
http://www.soft-tlt.ru/pocessora46.html
http://www.gaw.ru/html.cgi/txt/doc/micros/arm/arh_7dtmi/interfase_process.htm
http://www.arxitektura-computerov.ru/node/261
http://xpoint.ru/know-how/Articles/FloatingPointNumbers

Используемая литература



- Книга «Архитектура ЭВМ»,автор Мюллер
- Книга «Процессоры *Pentium4, Athlon* и *Duron*», авторы Михаил Гук, Виктор Юров
- Книга «Архитектура ЭВМ», автор Танненбаум