

Схемотехника

Пилипко Михаил Михайлович (а. 448, 2 уч. корп)

2018 год, весенний семестр

0. Литература

- Бунтов, Макаров: Микропроцессорные системы. Ч. 1 Цифровые устройства.
- Угрюмов: Цифровая схемотехника.
- Морозов, Пилипко: Схемотехника цифровых устройств.

1. Введение

Будем изучать радиоэлектронные схемы: аналоговые и цифровые.

Аналоговый сигнал идёт между E и землёй (на самом деле в каких-то внутренних пределах в данном отрезке) по амплитуде, как входной, так и выходной (в одних и тех же пределах).

Цель аналоговой схемы: обеспечение рабочей точки внутри заданного диапазона.

В цифровой схеме используется 2 уровня сигнала (логические 1 и 0 - земля), переключения между ними как можно более резкие. Между 0 и 1 - зона неопределённости.

Требуется обеспечить хорошее различение двух дискретных уровней сигнала.

Для преобразования этих схем одну в другую существуют преобразователи: ЦАП и АЦП.

АЦП делит временную ось аналогового сигнала на одинаковые отрезки (T), а ось напряжения делится на некоторые дискретные уровни, кодируемые двоичным кодом (=квантование). На каждом пересечении уровней времени и напряжения аналоговому сигналу сопоставляется как можно более точный цифровой. Чем выше частота дискретизации, тем точнее преобразование.

АЦП делает:	ЦАП делает:
1. Дискретизация по времени 2. Квантование по уровню 3. Кодирование цифровым кодом	То же самое, только в обратном порядке

2. Системы счисления

Делятся на позиционные (значение зависит от порядка расположения знаков) или ПСС, и непозиционные (соответственно, не зависит).

Разложение числа по основанию q :

$$E = e_{n-1}q^{n-1} + \dots + e_1q^1 + e_0q^0, \quad n - \text{разрядность числа по основанию } q.$$

Наиболее популярные основания: 2, 3, 8, 10, 12, 16, 60.

Один из критериев сравнения ПСС - удельная информационная плотность (УИП) представления данных.

$$\text{УИП} = q * n$$

$$N = q^n - \text{количество представимых чисел.}$$

При УИП=30:

(график на листочке)

По графику видно, что максимальной эффективности можно добиться для систем счисления по основанию e . В троичной системе представлять данные удобнее и компактнее, она понятнее, чем двоичная. Естественным образом можно ввести отрицательные числа: 1, 0, -1. Недостатки: нет надёжных и компактных логических, пригодных для троичной логики (с туннельным диодом всё тоже не очень гладко). Оказалось, что проще всего реализовать двоичную логику на

транзисторах.

3. Алгебра логики (булева алгебра)

Переменные имеют только два значения: 0 и 1 (ложь и истина соответственно).

Свойства переменных:

1. рефлексивность: $x = x$
2. симметричность: $x = y \rightarrow y = x$
3. транзитивность: $x = y, y = z \rightarrow x = z$

Функции задаются с помощью таблиц истинности: слева выписываются входные данные и их возможные значения, и им сопоставляются все возможные выходные значения.

В рамках булевой логики существует 3 базовые операции:

1. Конъюнкция (логическое **И**, логическое умножение). $y = x_1 \cap x_2$

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

2. Дизъюнкция (логическое **ИЛИ**, логическое сложение). $y = x_1 \cup x_2$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

3. Инверсия (логическое **НЕ**, логическое отрицание). $y = \bar{x}$

x	y
0	1
1	0

Операции 1, 2 возможны для большего числа переменных, причём порядок выполнения операции не повлияет на её итог (свойство ассоциативности для сложения и умножения).

$$x_1 + x_2 + \dots + x_n = y$$

$$x_1 * x_2 + \dots * x_n = y$$

4. Логические функции и выражения

Логическая функция - функция логического(-их) аргумента(-ов), которая сама принимает только два значения - 0 или 1. Также называется переключательной.

Пусть есть n переменных: x_1, x_2, \dots, x_n

Тогда $N = 2^n$ - количество наборов переменных, а количество возможных функций - 2^N

Этот факт можно проиллюстрировать следующей таблицей:

x	y_1	y_2	y_3	y_4
0	0	0	1	1
1	0	1	0	1

При этом $y_1 = 0, y_2 = x, y_3 = \bar{x}, y_4 = 1$.

Если для некоторых наборов аргументов значение не задано, функция называется недоопределённой.

Принцип двойственности логических функций:

1. $\overline{x_1 + x_2} = y \rightarrow \overline{x_1} * \overline{x_2} = y$
2. $\overline{x_1 * x_2} = y \rightarrow \overline{x_1} + \overline{x_2} = y$

Это можно доказать, составив соответствующие таблицы истинности. Например, для первого утверждения:

x_1	x_2	$x_1 + x_2$	$\overline{x_1 + x_2}$	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} * \overline{x_2}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Эквивалентные преобразования логических функций.

Делятся на тождества, законы и операции.

1. Тождества:

- 1) $x + 0 = x \quad x * 1 = x$
- 2) $x + 1 = 1 \quad x * 0 = 0$
- 3) $x + x = x \quad x * x = x$
- 4) $x + \overline{x} = 1$
- 5) $\overline{\overline{x}} = x$

2. Законы:

- 6) $x_2 + x_1 = x_1 + x_2 \quad x_1 x_2 = x_2 x_1$
- 7) $(x_1 + x_2) + x_3 = x_1 + (x_2 + x_3) \quad (x_1 x_2) x_3 = x_1 (x_2 x_3)$
- 8) $x_1(x_2 + x_3) = x_1 x_2 + x_1 x_3 \quad x_1 + (x_2 * x_3) = (x_1 + x_2)(x_1 + x_3)$

Можно доказать последнее:

$$(x_1 + x_2)(x_1 + x_3) = x_1 x_1 + x_1 x_2 + x_1 x_3 + x_2 x_3 = x_1(1 + x_2 + x_3) + x_2 x_3 = x_1 + x_2 x_3$$

3. Операции:

- 9) $\overline{x_1 + x_2} = \overline{x_1} * \overline{x_2} \quad \overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$ (правила де Моргана)
- 10) $x_1 * x_2 + x_1 * \overline{x_2} = x_1 \quad (x_1 + x_2)(x_1 + \overline{x_2}) = x_1$ (склеивание)

Однако!

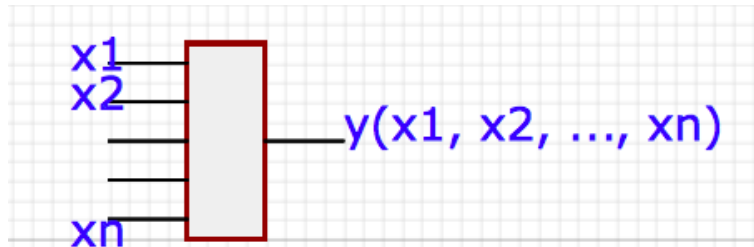
$$x_1 * \overline{x_2 x_3} + x_1 * x_2 x_3 = x_1$$

$$x_1 * \overline{x_2} \overline{x_3} + x_1 * x_2 x_3 \neq x_1$$

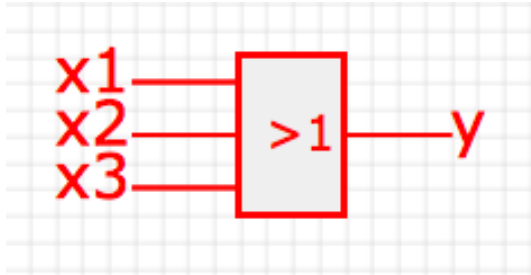
- 11) $x_1 + x_1 x_2 = x_1 \quad x_1(x_1 + x_2) = x_1$ (поглощение)
- 12) $x_1 * \overline{x_2} + x_2 = x_1 + x_2 \quad (x_1 + \overline{x_2})x_2 = x_1 x_2$

5. Комбинационная логическая схема

Входные сигналы (x_1, x_2, \dots, x_n) будем изображать слева, выход функции - справа от элемента, реализующего логическую функцию. Также элементу требуется питание и земля, обычно не изображаемые на логических диаграммах, но обязательные на реальных схемах для сборки (на картинке).

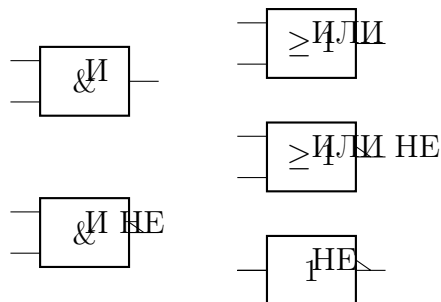


Пример: мажоритарная функция - каких значений на входных данных будет больше, таким будет выходное значение (на картинке).



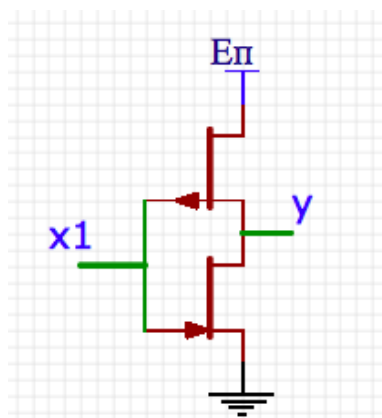
x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Наиболее распространены элементы, выполняющие базовые логические операции. Кругочком у выхода обозначается инверсия.

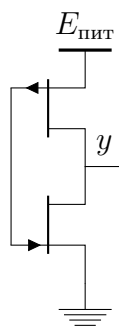
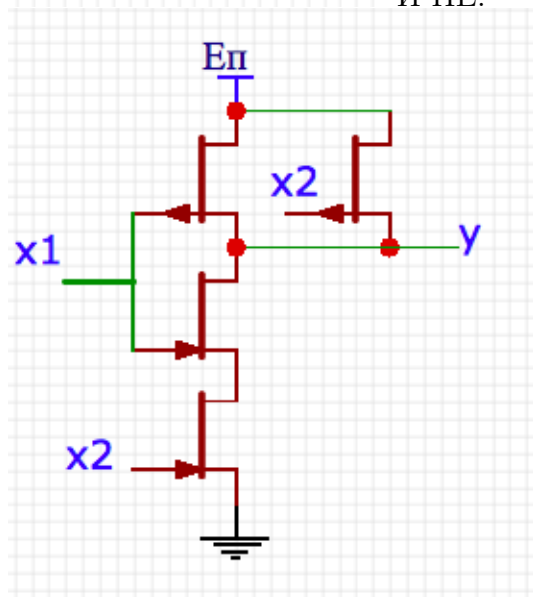


Схемотехническая реализация:

Инвертор:



И-НЕ:



6. Способы задания логических функций

1. Словесное описание:

если $x_1 = x_2$, то $y = 0$, $x_1 \neq x_2 \rightarrow y = 1$

2. Таблица истинности:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

3. Структурная формула. Способы записи:

а) Совершенная дизъюнктивная нормальная формула

	x_1	x_2	x_3	y
	0	0	0	0
	0	0	1	0
	0	1	0	0
$y = f(x_1, x_2, x_3)$	0	1	1	1
	1	0	0	0
	1	0	1	1
	1	1	0	1
	1	1	1	1

Вводятся нормирующие функции y_a, y_b, y_c, y_d (по количеству истинных значений y таблице). Их значение - значение логического **И** от переменных, равных 1 или инвертированных: $y_a = \bar{x}_1 x_2 x_3, y_b = x_1 \bar{x}_2 x_3, y_c = x_1 x_2 \bar{x}_3, y_d = x_1 x_2 x_3$. Тогда $y = y_a + y_b + y_c + y_d$.

б) Совершенная конъюнктивная нормальная формула

Для той же функции y можно ввести y_e, y_f, y_g, y_h . Берутся строки, где $y = 0$, для них производится логическая сумма входных переменных, равных 0 или инвертированных. $y_e = x_1 + x_2 + x_3, y_f = \bar{x}_1 + x_2 + x_3, y_g = x_1 + \bar{x}_2 + x_3, y_h = x_1 + x_2 + \bar{x}_3$. Тогда $y = y_e y_f y_g y_h$.

4. Числовой способ задания функции

Для представления функции в СДНФ под знаком \sum перечисляются номера наборов входных переменных, в которых функция равна 1.

$$y(x_1, x_2, x_3) = \sum(3, 5, 6, 7) = \sum(011, 101, 110, 111)$$

Для представления функции в СКНФ под знаком \prod перечисляются номера наборов входных переменных, в которых функция равна 0.

$$y(x_1, x_2, x_3) = \prod(0, 1, 2, 4)$$

7. Переход от структурной формулы к логической схеме и обратно

Элементы располагаются на схеме, начиная от входов, в том же порядке, в котором выполняются логические операции.

(сложная картинка с шиной)

Чтение СФ производится от выхода к входам: $y = \alpha + \beta + \gamma + \delta, \alpha = \bar{x}_1 x_2 x_3$. Поскольку для инвертора требуется 2 транзистора, для И - 8 транзисторов, а для ИЛИ - 10, всего для данной функции в КМОП технологии потребуется 48 транзисторов.

8. Минимизация логических функций

Построение логической схемы непосредственно по СФ (будь то СДНФ или СКНФ) нецелесообразно. Необходимо предварительно упростить формулу.

1. Метод Квайна (Quine) - операции склеивания ($x_1 x_2 + x_1 \bar{x}_2 = x_1$) и поглощения ($x_1 x_2 + x_1 = x_1$). Предполагает 2 этапа: склеивание соседних слагаемых и поглощение избыточных слагаемых.

Соседние слагаемые - слагаемые, отличающиеся логическими значениями только одной переменной.

x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	1	1	0
0	0	1	1	1	1
0	1	0	0	1	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

$$\begin{cases} y_1 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_3\bar{x}_2x_1 = \bar{x}_2\bar{x}_3(\bar{x}_1 + x_1) = \bar{x}_2\bar{x}_3 \\ y_2 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_2\bar{x}_3x_1 + x_1x_2x_3 = \bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_3 + x_1x_2x_3 \\ y_3 = \bar{x}_2\bar{x}_3x_1 + x_1\bar{x}_2 + x_3 + \bar{x}_1x_2x_3 + x_1x_2x_3 = \bar{x}_2x_1 + x_3x_1 + x_3x_2 = \bar{x}_2x_1 + x_3x_2 \end{cases}$$

Недостатки метода - появление лишних слагаемых и экспоненциальный рост числа проверок склеивания при увеличении числа входных переменных.

$$y_{mg} = \bar{x}_3x_2x_1 + x_3\bar{x}_2x_1 + x_3x_2\bar{x}_1 + x_3x_2x_1 = x_1x_2 + x_2x_3 + x_3x_1 - (26 \text{ транзисторов в КМОП})$$

2. Метод Карно (Karnaugh)

Обеспечивает отсутствие лишних слагаемых благодаря наглядному представлению функции в виде карты.

	x_1	x_2	y
	0	0	0
Пример:	0	1	1
	1	0	1
	1	1	1

(далее на картинке)

Принципы: если в карте есть 2^N соседствующие (по горизонтали или вертикали) единицы, то они объединяются в группы. Цель: объединить все единицы минимальным количеством максимальных контуров.

Пример (из метода Квайна): (картинка)

$$\begin{cases} y_1 = \bar{x}_3\bar{x}_2 \\ y_2 = \bar{x}_3\bar{x}_2 + \bar{x}_3\bar{x}_1 + x_1x_2x_3 \\ y_3 = \bar{x}_2x_1 + x_2x_3 = (x_1 + x_2)(x_3 + \bar{x}_2) \end{cases} \quad (1)$$

Пример: 3 контура

$$\begin{cases} y_1 = \bar{x}_3\bar{x}_1 \\ y_2 = \bar{x}_4x_3 \\ y_3 = x_1x_2x_3 = \end{cases} \quad (2)$$

	00	01	10	11	
000	0	0	1	0	Существует модифицированный
001	0	1	0	0	
011	1	1	0	0	
Пример: метод теряет наглядность.	010	1	0	0	
	110	1	0	0	
	111	1	1	0	
	101	0	1	0	
	100	0	0	0	

метод Квайна-Маккласки, его сложность также эксп. возрастает.