

АЛУ (Де)мультиплексирование

Д. В. Луцив

Кафедра системного программирования СПбГУ



CS220 (231000)

Устройство и действие сумматора

Складываем $A + B = S$.

При суммировании двоичных чисел у i -го разряда на вход поступают: a_i, b_i, c_{i-1}

на выходе имеем: s_i, c_i .

При этом:

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$

$$c_i = \begin{cases} 0 & a_i + b_i + c_{i-1} \leq 1 \\ 1 & a_i + b_i + c_{i-1} > 1 \end{cases}$$

Отождествляя 1 и Истину, можем написать

$$c_i = (a_i \wedge b_i) \vee (b_i \wedge c_{i-1}) \vee (c_{i-1} \wedge a_i).$$

Таким образом, сумматор — логическая схема.

Устройство и действие сумматора

Складываем $A + B = S$.

При суммировании двоичных чисел у i -го разряда на вход поступают: a_i, b_i, c_{i-1}

на выходе имеем: s_i, c_i .

При этом:

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$
$$c_i = \begin{cases} 0 & a_i + b_i + c_{i-1} \leq 1 \\ 1 & a_i + b_i + c_{i-1} > 1 \end{cases}$$

Отождествляя 1 и Истину, можем написать

$$c_i = (a_i \wedge b_i) \vee (b_i \wedge c_{i-1}) \vee (c_{i-1} \wedge a_i).$$

Таким образом, сумматор — логическая схема.

Устройство и действие сумматора

Складываем $A + B = S$.

При суммировании двоичных чисел у i -го разряда на вход поступают: a_i, b_i, c_{i-1}

на выходе имеем: s_i, c_i .

При этом:

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$
$$c_i = \begin{cases} 0 & a_i + b_i + c_{i-1} \leq 1 \\ 1 & a_i + b_i + c_{i-1} > 1 \end{cases}$$

Отождествляя 1 и Истину, можем написать

$$c_i = (a_i \wedge b_i) \vee (b_i \wedge c_{i-1}) \vee (c_{i-1} \wedge a_i).$$

Таким образом, сумматор — логическая схема.

Устройство и действие сумматора

Складываем $A + B = S$.

При суммировании двоичных чисел у i -го разряда на вход поступают: a_i, b_i, c_{i-1}

на выходе имеем: s_i, c_i .

При этом:

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$
$$c_i = \begin{cases} 0 & | \ a_i + b_i + c_{i-1} \leq 1 \\ 1 & | \ a_i + b_i + c_{i-1} > 1 \end{cases}$$

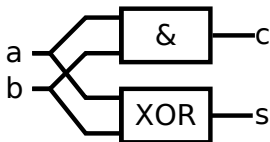
Отождествляя 1 и Истину, можем написать

$$c_i = (a_i \wedge b_i) \vee (b_i \wedge c_{i-1}) \vee (c_{i-1} \wedge a_i).$$

Таким образом, сумматор — логическая схема.

Схемотехнически

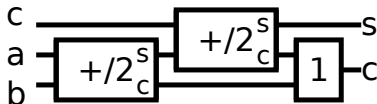
Полусумматор



Оптимизация формулы:

$$c_i = (a_i \wedge b_i) \vee (c_{i-1} \wedge (a_i \oplus b_i))$$

1-битный сумматор (на самом деле входы равноправны)



Устройство и действие инвертора (для вычитания)

$$x - y = x + (-y)$$

Проблема — вычислить $-y$, зная y .

$$[-y] = [P - y] = [P - 1 + 1 - y] = [1] + [(P - 1) - y]$$

уже легче, надо вычислить $(P - 1) - y$.

Для $P = 2^N$ справедливо $P - 1 = \sum_{i=0}^{N-1} 2^i$.

Т.е. это число со всеми двоичными единицами. Чтобы вычесть из него y , достаточно инвертировать все биты y при помощи вентиля НЕ. Пример для $P = 2^8$:

$$\begin{array}{r} 11111111 \\ - 11001010 \\ \hline = 00110101 \end{array}$$

Устройство и действие инвертора (для вычитания)

$$x - y = x + (-y)$$

Проблема — вычислить $-y$, зная y .

$[-y] = [P - y] = [P - 1 + 1 - y] = [1] + [(P - 1) - y]$ —
уже легче, надо вычислить $(P - 1) - y$.

Для $P = 2^N$ справедливо $P - 1 = \sum_{i=0}^{N-1} 2^i$.

Т.е. это число со всеми двоичными единицами. Чтобы вычесть из него y , достаточно инвертировать все биты y при помощи вентиля НЕ. Пример для $P = 2^8$:

$$\begin{array}{r} 11111111 \\ - 11001010 \\ \hline = 00110101 \end{array}$$

Устройство и действие инвертора (для вычитания)

$$x - y = x + (-y)$$

Проблема — вычислить $-y$, зная y .

$$[-y] = [P - y] = [P - 1 + 1 - y] = [1] + [(P - 1) - y]$$

уже легче, надо вычислить $(P - 1) - y$.

Для $P = 2^N$ справедливо $P - 1 = \sum_{i=0}^{N-1} 2^i$.

Т.е. это число со всеми двоичными единицами. Чтобы вычесть из него y , достаточно инвертировать все биты y при помощи вентиля НЕ. Пример для $P = 2^8$:

$$\begin{array}{r} 11111111 \\ - 11001010 \\ \hline 00110101 \end{array}$$

Замечания

Перенос через сумматор распространяется постепенно, поэтому:

- в старших разряде «ответ» есть сразу, но он неправильный;
- правильный ответ появляется через какое-то время;
- чем больше разрядность операции, тем больше время, даже для одного и того же процессора.

Последний бит переноса c_n можно запомнить и при следующем суммировании использовать в качестве c_{-1} . Это позволит соединить несколько сумматоров или несколько раз использовать один и тот же для реализации арифметики произвольной разрядности.

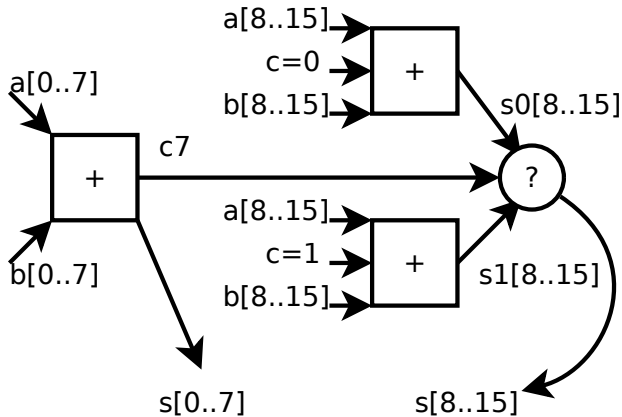
Замечания

Перенос через сумматор распространяется постепенно, поэтому:

- в старших разряде «ответ» есть сразу, но он неправильный;
- правильный ответ появляется через какое-то время;
- чем больше разрядность операции, тем больше время, даже для одного и того же процессора.

Последний бит переноса c_N можно запомнить и при следующем суммировании использовать в качестве c_{-1} . Это позволит соединить несколько сумматоров или несколько раз использовать один и тот же для реализации арифметики произвольной разрядности.

Оптимизация: упреждающий сумматор I



Оптимизация: упреждающий сумматор II

- (?) — мультиплексор (ниже)
- Вынашивание полутора детей за 4,5 месяца. Да, можно!
- Можно сделать 3, 4 или больше сегментов
- Важна золотая середина: сегменты выбираются последовательно \Rightarrow грубая оценка — порядка \sqrt{N} (на самом деле меньше)

Мультипликатор

При умножении на 1 бит числа x мы можем пользоваться логической схемой:

$$x \times b = x \wedge b.$$

Тогда для многоразрядных чисел справедливо (\ll — операция сдвига):

$$x \times y = \sum_{i=0}^{N-1} (y_i \wedge (x \ll i))$$

Т.е. выразили умножение через известные операции.

Разрядность произведения — сумма разрядностей множителей.

Мультипликатор

При умножении на 1 бит числа x мы можем пользоваться логической схемой:

$$x \times b = x \wedge b.$$

Тогда для многоразрядных чисел справедливо (\ll — операция сдвига):

$$x \times y = \sum_{i=0}^{N-1} (y_i \wedge (x \ll i))$$

Т.е. выразили умножение через известные операции.

Разрядность произведения — сумма разрядностей множителей.

Делитель

Алгоритм деления «в столбик» для системы счисления с произвольным основанием:

- 1 сдвигаем делитель влево, пока он меньше делимого;
- 2 вычитаем, пока вычитается, прибавляя к очередному разряду частного;
- 3 сдвигаем делитель вправо (и меняем позицию в частном), пока нельзя вычитать, потом (2);
- 4 если делитель вернулся на исходную позицию относительно сдвига, то от делимого остался остаток.

Для двоичной:

- вычитаем, если (*a не пока*) вычитается.

Делитель

Алгоритм деления «в столбик» для системы счисления с произвольным основанием:

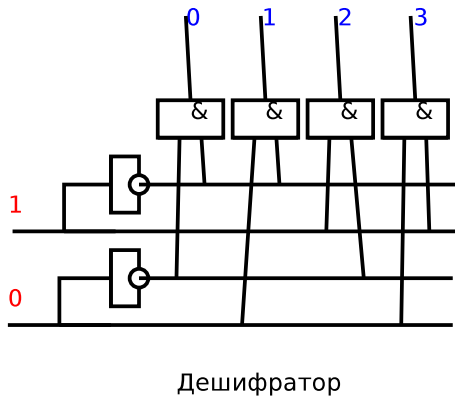
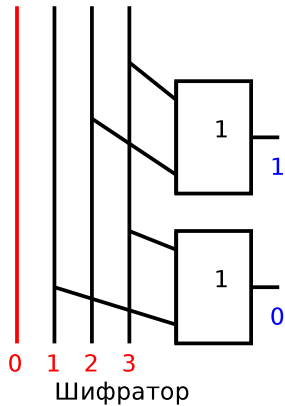
- 1 сдвигаем делитель влево, пока он меньше делимого;
- 2 вычитаем, пока вычитается, прибавляя к очередному разряду частного;
- 3 сдвигаем делитель вправо (и меняем позицию в частном), пока нельзя вычитать, потом (2);
- 4 если делитель вернулся на исходную позицию относительно сдвига, то от делимого остался остаток.

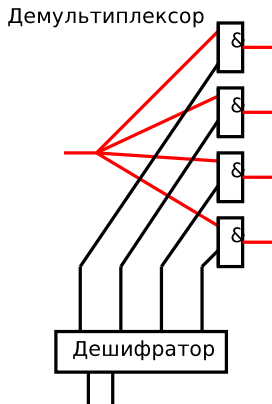
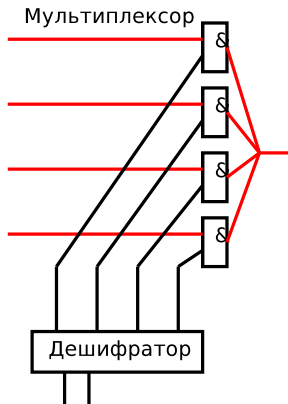
Для двоичной:

- вычитаем, если (*a не пока*) вычитается.

Замечания

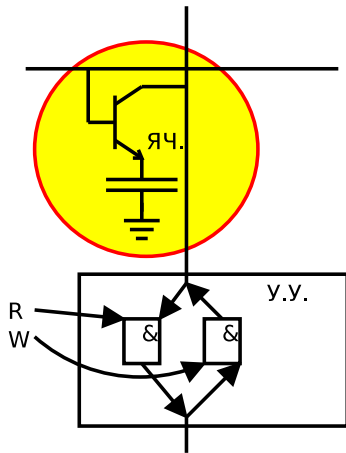
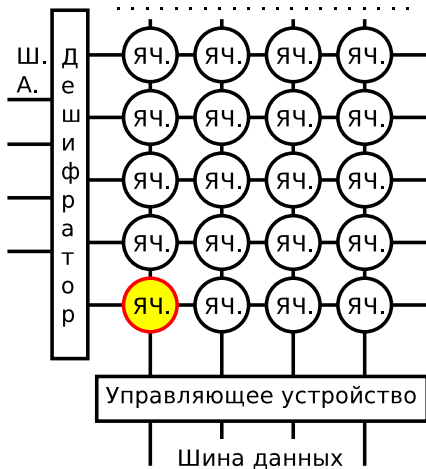
- Мультипликатор, как и сумматор, можно реализовать логической схемой, но очень громоздкой, поэтому часто его делают микропрограммой. А на простых архитектурах — программой.
- Есть аппаратные логические делители — большие и «горячие» микросхемы, используются очень редко. Делитель — почти всегда микропрограмма (или программа).
- Умножение чисел произвольной длины возможно, однако перенос эффективнее делать по слову (старшей половине произведения), а не по биту.



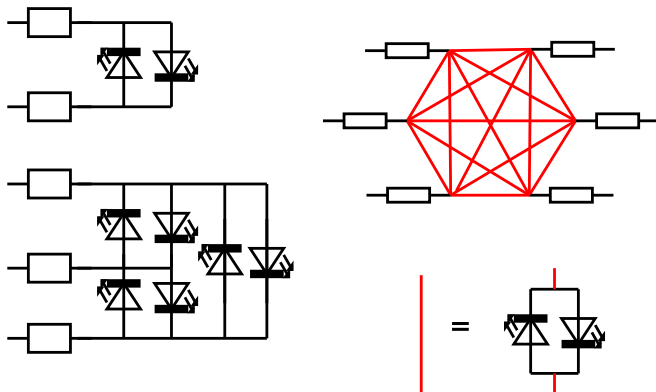


Системы упаковки и свёртки/развёртки

- Красным — сигнальные линии, м.б. аналоговые, м.б. многобитные.
- В I половине XX века для временно́го мультиплексирования телефонных линий применялись ламповые (де)мультиплексоры.
- Не в чистом виде
 - Фототелеграф (см. М. Булгаков, Мастер и Маргарита).
 - Современный (не в чистом виде) вариант: аналоговая телекамера → телевизор с ЭЛТ.



Технологически конденсатор выполняется вместе с транзистором и почти не требует дополнительного места (может быть использована ёмкость P-N перехода).



$2 \binom{n}{2} = 2C_2^n = \frac{n!}{(n-2)!2!}$ комбинаций на n входах.

Используется свойство светодиодов открывать одну самую короткую из параллельных цепей.

Элементная база

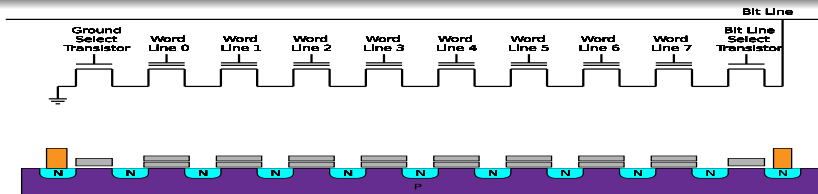
▶ Полярный транзистор (униполярный)

▶ Транзистор с плавающим затвором

Эти транзисторы могут при подаче очень высокого или очень низкого потенциалов на управляющий вход запоминать своё состояние. После нескольких миллионов срабатываний транзистор необратимо портится (поэтому у Flash ограничено количество перезаписей). Этим эффектом, так же, как и широкой петлёй гистерезиса (хотя природа этого совершенно иная), можно пользоваться для хранения данных.

▶ Flash-память (статья в Википедии, иллюстрации ниже из неё)

Память NAND



- Для чтения на все слова, кроме читаемого, подаётся небольшой «приоткрывающий» потенциал. Ток течёт с соответствующих открытым транзисторам битовых линий в землю.
- Для программирования (открытия затвора, установки в бита 0) надо небольшим потенциалом «приоткрыть» все линии слов и подать сильный сигнал на пересечения нужных слова и бита.
- Из-за особенностей изготовления танзисторов, сброс (в 1) по словам или ещё большим блокам.
- Для SSD введена операция **TRIM**, которая говорит накопителю, что блок памяти свободен и может использоваться для оптимизации с целью увеличения ресурса перезаписи.

Вопросы



▶ EDU.DLUCIV.NAME