Схемотехническое проектирование

Работы 6-8

Работа 6. Исследование регистров

§ 6.1. Параллельный регистр

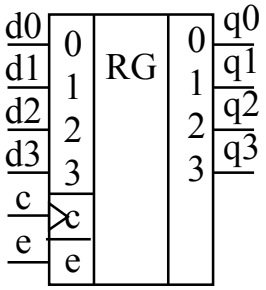

Рис. 6.1. Символ параллельного регистра

Регистром называется устройство из нескольких триггеров, предназначенное для записи, хранения и выдачи данных. Каждый разряд двоичного числа записывается в отдельном триггере. Количество триггеров в регистре определяет разрядность записываемого числа. Ввод и вывод информации могут выполняться параллельным или последовательным кодом, поэтому существуют различные типы регистров - параллельные, параллельно-последовательные, регистры сдвига.

Параллельный регистр, построенный на базе D-триггеров с динамическим управлением с входом разрешения (рис. 6.1), выполняет запись данных по фронту синхросигнала c при наличии разрешения e = 1. Этот регистр не имеет прозрачного режима. Изображение треугольника в обозначении входа сигнала соответствует регистру с прямым динамическим входом.

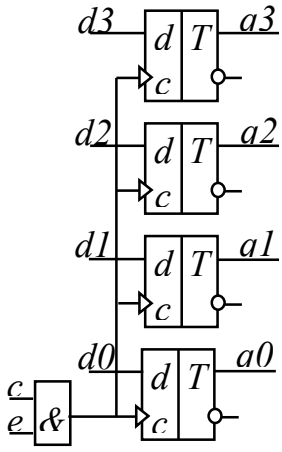

Рис. 6.2. Схема параллельного регистра

При построении схемы на основе D-триггеров (рис. 6.2) к d-входам триггеров подключается шина входных данных, а к выходам триггеров – шина выходных данных. В параллельных регистрах ввод и вывод двоичных слов производится одновременно для всех разрядов, поэтому входы синхронизации всех триггеров (c) соединяют параллельно.

Описания параллельных регистров на языке Verilog подобно описаниям D – триггеров. Сигналы d и q указаны как векторы. Для выходного сигнала указан тип reg. Работу динамического регистра и его срабатывание по фронту синхросигнала *c* при наличии разрешения e описывает строка *always @ (posedge c) if (e) q = d;* .

Другой вариант параллельного регистра - это регистр – защелка, построенный на D-триггеров со статическим управлением. Этот регистр имеет прозрачный режим, его используют в некоторых схемах где требуется передача всех изменений входного сигнала на выход при c = 1. Для описания регистра - защелки используется другая запись последовательного оператора: *always if (c) q = d;*, которая имеет смысл: «Всегда, если c=1 выполнять присваивание q=d». Заметим, в операторе «always» символ «@» и список чувствительности отсутствуют, поэтому срабатывание оператора будет происходить при любых изменениях сигнала d при условии c = 1. При этом обеспечивается прозрачный режим.

```
//Параллельный регистр
module v61_ din_reg
(c, e, d, q);
input c,e;  input [3:0] d;
output [3:0] q;  reg  [3:0] q;
always @ (posedge c)
if (e)   q = d;   endmodule
```

Задание 6.1. Иерархический проект параллельного регистра.

1) Создайте проект и символ по схеме (рис. 6.2) с именем s61_*din_reg* (этапы 1-5). При вводе схемы в графическом редакторе необходимо с=использовать триггеры dff и добавить тетминалы

2) Создайте проект и символ по описанию с именем *v*61_*din_reg*. (этапы 1-5)

3) Создайте иерархический проект с именем s*v*61_*din_reg*. Выполните исследование данного регистра в режиме симулятора.(этапы 1-9). Входной код, подаваемый на шину d[3..0] необходимо подать от счетчика (кнопка с буквой С)и предусмотреть его изменение, например, инкремент

4) Исследуйте временные задержки, опишите методику их измерения.

5) Опишите работу устройства, разработайте теоретические временные диаграммы, поясняющие

его работу.

6) Подключите печатную плату. Разработайте методику исследования устройства и использованием элементов платы – кнопок и светодиодов.

§ 6.2. Сдвигающий регистр

Сдвигающий регистр - цифровая схема, состоящая из последовательно включенных динамических D - триггеров, содержимое которых сдвигается при подаче синхроимпульсов. Каждый импульс вызывает сдвиг на один разряд. Число, код которого записан в сдвигающий регистр, при сдвиге влево удваивается, а при сдвиге вправо уменьшается вдвое. Сдвигающие регистры служит для умножения или деления коды чисел на 2.

Сдвигающий регистр используют для приема информации в виде последовательного кода, для формирования и выдачи параллельного кода, а также для преобразования параллельного кода в последовательный. При параллельной выдаче информация снимается одновременно с выходов всех триггеров. Последовательная выдача осуществляется с выхода so в моменты действия тактовых импульсов.

Схема сдвигающего регистра (рис. 6.3) содержит 4 триггера для записи параллельный выход q3 – q0. На вход триггера старшего разряда подается сигнал последовательного кода, или входной перенос si (Serial Input). С выхода триггера младшего разряда снимается выходной сигнал последовательного кода, или выходной перенос so (Serial Output), который запоминается в дополнительном триггере. Синхросигнал *c* подается параллельно на все триггеры.

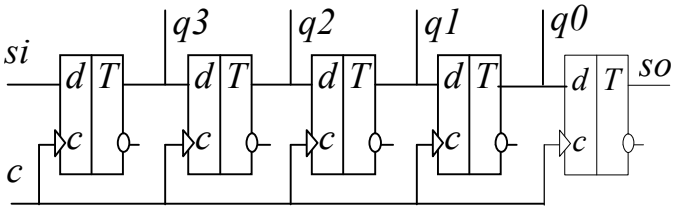

Рис. 6.3. Схема сдвигающего регистра

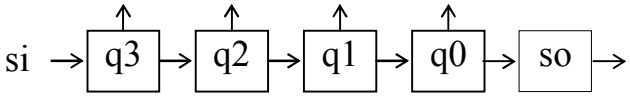

Рис. 6.4. Сдвиг и выдача кода

Работа схемы выполняется в соответствии с известным правилом работы динамического D – триггера, который по фронту синхроимпульса запоминает входной сигнал. В схеме по фронту синхросигнала одновременно выполняются присваивания: so = q0, q0 = q1, q1 = q2, q2 = q3. q3 = si (рис. 6.4). Первым слева на схеме изображен триггер старшего разряда с выходом q3, затем триггеры остальных разрядов с выходами q2, q1, q0. Каждый синхроимпульс в соответствии с рис. 6.4 выполняет сдвиг кода q[3..0] вправо и уменьшает его значение в два раза. Бит переноса запоминается в триггере so. Входной сигнал – последовательный код si должен подаваться младшими разрядами вперед.

```
module v62_shift_reg
(c, si,so, q);
 input c, si;
output so;
output [7:0] q;reg [7:0] q;
 always  @ (posedge c)
 { q,so} = {si,q};
 endmodule
```

Другой вариант построения схемы сдвигающего регистра – это подача si на вход младшего разряда q0 и соединение триггеров, когда выходы младших разрядов соединяются с входами старших разрядов. В этом случае регистр озволяет получить сдвиг влево и умножение кодда в два раза в каждом такте синхросигнала. Последовательный код, для его преобразования в параллельный, необходимо подавать старшими разрядами вперед..

В описании 8-разрядного сдвигающего регистра на Verilog учитывается поведение схемы. Каждый синхроимпульс записывает бит входного переноса si и 8-разрядный код числа q со сдвигом в другом виде: вначале 8-разрядный код числа q, затем бит переноса so.

Задание 6.2. Иерархический проект сдвигающего регистра.

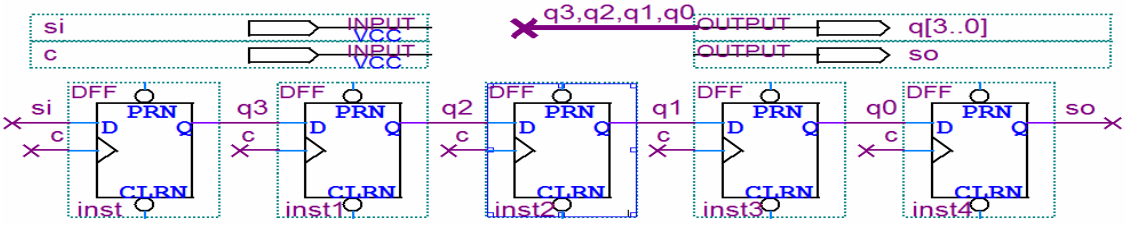

Рис. 6.5. Схема проекта s65_shift_rtg

1)Создайте проект и символ по схеме (рис. 6.5) с именем s62_*shift_reg*

2) Опишите работу регистра по временным диаграммам при вводе заданного последовательного кода например, 1011_2. Определите временные задержки, опишите методику их измерения.

3) Создайте проект 8-разрядного сдвигающего регистра по описанию с именем *v62_shift_reg*.. Выполните моделирование, опишите временные диаграммы для заданного в таблице последовательного кода.

Вариант	1	2	3	4	5	6
Код (hex)	8f	9d	a3	b5	c7	d3
Вариант	7	8	9	10	11	12
код	e2	1f	97	3d	85	93

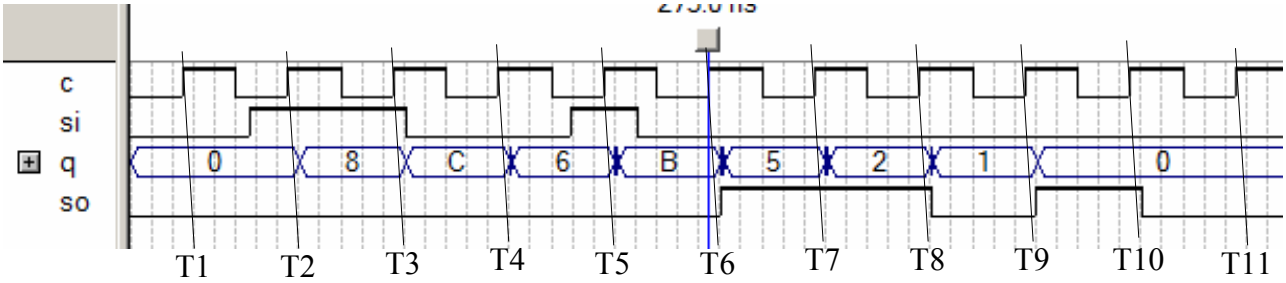

Рис. 6.6. временные диаграммы проекта s65_shift_rtg

Пояснение. Описание работы схемы по временным диаграммам (рис. 6.6)..

При разработке и вводе тестовых сигналов в сигнальном редакторе Quartus вначале необходимо изобразить синхросигнал c (Period 50 ns), а затем, используя курсор и кнопки, изобразить импульсы si. Срабатывание регистра (сдвиги) происходят в моменты действия фронтов синхроимпульсов, поэтому значения сигнала si должны быть заданы в моменты, предшествующие фронтам синхросигнала, младшими разрядами вперед. Например, если задан код $b_{16} = 1011_2$, то необходимо изобразить входной сигнал si в виде последовательности 1101_2 .

Анализ временных диаграмм. Моменту времени T1 соответствует исходное состояние схемы: q = 0, si = 0. сигнал q = const, so = 0.

Момент T2 и сигнал si = 1. Это младший разряд заданного последовательного кода числа b_{16}.В результате сдвига $q = 8_{16}$, so = o

Момент T3 сигнал si = 1. В результате сдвига $q = c_{16}$, so = o

В моменты T4 и T5 вводятся значения si = 0 и si = 1. В результате сдвига $q = b_{16}$, so = o

На интервале T2 – T5 выполняется преобразование последовательного кода, подаваемого на si, равного 1011_2, в параллельный q = b. Также исходное значение параллельного кода q = 0 преобразуется в последовательный код so (0000_2).

На интервале T5 – T11 выполняется преобразование параллельного кода b_{16} в последовательный код.

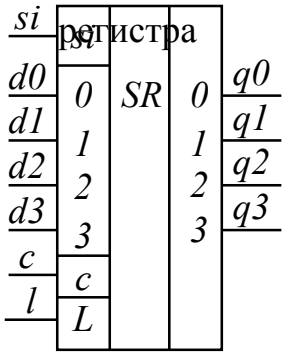

Рис. 6.7. Обозначение универсального регистра сдвига

§ 6.3. Универсальный сдвигающий регистр

Универсальные регистры сдвига имеют параллельные и последовательные входы и выходы, что позволяет выполнять ввод и вывод данных в виде параллельного или последовательного кода, а также преобразовывать не только последовательный код в параллельный, но и выполнять обратное преобразование параллельного кода в последовательный. Последовательно - параллельные регистры используются также при реализации последовательных интерфейсов в микропроцессорной системе.

Универсальный сдвигающий 4-разрядный регистр (рис. 6.7) имеет отдельные выводы для ввода данных в виде параллельного кода (d3..d0) и в виде последовательного кода (si - Serial Input). Выбор входных сигналов определяет сигнал разрешения параллельной загрузки (l - Load - загрузка). Выходные сигналы регистра формируются на шине q3 – q0.

Структура регистра (рис. 6.8) зависит рт заданного направления сдвига. Если задан сдвиг влево, то сигнал si подается на вход триггера младшего разряда, выход которого подается на вход триггера первого разряда и т. д. Каждый синхроимпульс умножает код регистра на 2. Входной сигнал – последовательный код (si) должен подаваться старшими разрядами вперед.

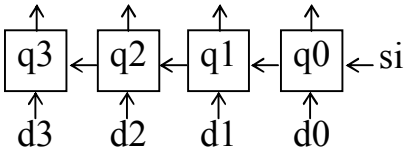

Рис. 6.8. Структура универсального регистра

В схеме универсального регистра (рис. 6.9) входной сигнал каждого d-триггера формируется мультиплексором «2 в 1». При разрешении параллельной загрузки (l=1), входы триггеров подключаются к проводникам входной шины данных. В этом режиме по положительному фронту синхроимпульса с выполняется запись входного кода в регистр.

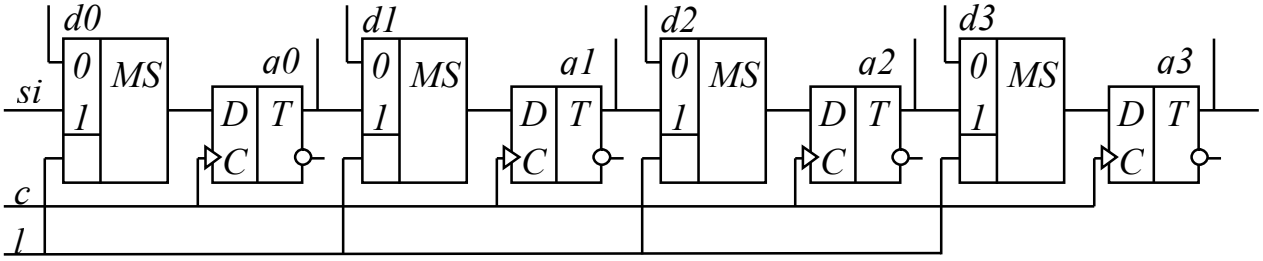

Рис. 6.. Схема универсального сдвигающего регистра

```
module v63_univ_reg (c, si, d, l, q)
input c, si, l;  input [3:0] d;
 output [3:0] q;  reg  [3:0] q;
always @ (posedge c)
if (l) q=d;
 else  {q} = {q[2:0], si };
endmodule
```

При запрещении параллельной загрузки (l=0) выполняется последовательная загрузка. По положительному фронту синхроимпульса «с» код, хранящийся в регистре, переместится на один разряд в сторону старших разрядов. Порядок размещения триггеров на схеме (рис. 6.9) определяется направлением передачи сигналов, он не соответствует структуре (рис. 6.8). в которой учитывается расположение разрядов в записи чимла.

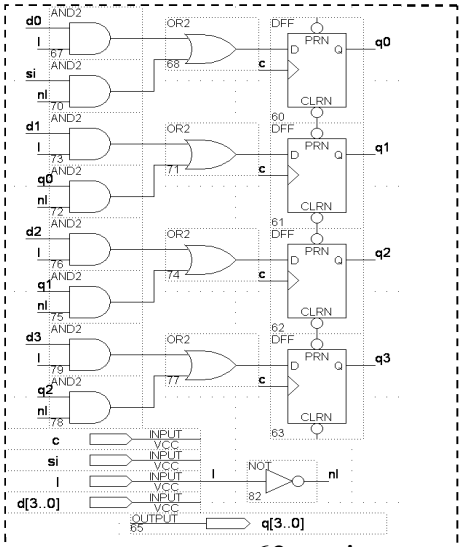

Рис. 6.10. Схема s63_univ_reg

Описание универсального сдвигающего регистра содержит определение входных и выходных сигналов в соответствии с обозначением (рис. 6.7). Описание работы выполняет последовательный оператор, срабатываний по фронту сигнала *c*, который содержит оператор «if», в качестве условия которого записан сигнал *l*. Если *l* = 1, то выполняется параллельная загрузка, описанная присваиванием {q}={d}.

Если l=0, то выполняется сдвиг влево и заполнение младшего бита сигналом si.

Задание 6.3. Иерархический проект универсального сдвигающий регистра

1) Создайте проект и символ по схеме (рис. 6.10) с именем s*63_univ_reg* (этапы 1-5).

2) Создайте проект и символ по описанию с именем *v63_univ_reg* (этапы 1-5).

3) Создайте иерархический проект с именем s*v63_univ_reg*. Выполните исследование данного регистра в режиме симулятора.(этапы 1-9).

4) Проведите моделирование работы схемы в режиме последовательной загрузки. Установите l = 0. Входной сигнал si введите подобно предыдущему заданию.

5) Проведите моделирование работы схемы в режиме параллельной загрузки. Входной код, подаваемый на шину d[3..0] должен периодически загружаться в регистр. Для отображения процесса загрузки целесообразно предусмотреть изменение (инкремент) загружаемого кода.

5) Исследуйте временные задержки, опишите методику их измерения.

§ 6.4. Генератор псевдослучайной последовательности.

В случайной последовательности чередование нулей и единиц не имеет никакой

закономерности. Статистические характеристики случайной последовательности и соответствуют сигналу типа «белый шум».

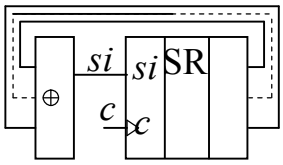

Рис. 6.11. Полинрмиальный счетчик

Псевдослучайная последовательность формируется цифровыми методами, характеризуется наличием повторения значений через определенный интервал – период псевдослучайной последовательности. Статистические параметры случайной и псевдослучайной последовательностей близки. Они должна иметь одинаковые количества одиночных нулей (..101.) и одиночных единиц (..010..), а также двойных и тройных единиц (..0110..и ..01110..) и нулей (..1001..и ..10001..). Псевдослучайные последовательности применяют в системах шифрования, контроля и тестирования.

Генераторы псевдослучайной последовательности строят, используя кольцевые полиномиальные счетчики, содержащие сдвигающие регистры с линейными обратными связями которые содержат составлена только из сумматоров по модулю два (Рис. 6.11). Выходной сигнал такой цепи описывает линейный полином:

$$si(q_0, q_1, \dots, q_{N-1}) = a_0 q_0 \oplus a_1 q_1 \oplus \dots \oplus a_{N-1}\, q_{N-1}.$$

Коэффициенты полинома $a_0,\ a_1,\ \dots a_{N-1}$ принимают значения 1 для тех разрядов, выходы которых необходимо подключить к входам сумматора по модулю два. Для получения последовательности максимальной длины (2^n -1) необходимо подать обратную связь с определенных разрядов регистра, номера которых для регистров различной разрядности указаны в приведенной таблице.

Разряд	Точки подкл.
4	1,0
5	**2,0**
6	1,0
7	3,1,0
9	4,0
17	3,0
20	3,0

Состояние, когда в регистре содержатся все нули, и , и сигнал si = 0, является для данной схемы «тупиковым», в регистре непрерывно выполняется сдвиг нулей. Для выхода в рабочий режим необходимо установить начальное значение кода регистра

Схема генератора псевдослучайной последовательности s64_rnd содержит 5-разрядный сдвигающий регистр с элементом xor (сумма по модулю 2) в цепи обратной связи, на входы которого подключены разряды 0 и 2 (Рис. 6.12), поэтому псевдослучайная последовательность будет иметь максимальную длину.

Цепь начальной установки кода регистра на элементах nor6 и or2, выдает сигнал 1 при коде в регистре 00000, обеспечивает выход из тупиковой ситуации.

Битовые псевдослучайные последовательности формируются на всех выводах сдвигающего регистра, и на входе si, различаются сдвигом во времени.

Последовательность кодов регистра не является псевдослучайной, даже начинающий хакер обнаружит связь (корреляцию) между соседними отсчетами кода. Которые получены в результате сдвига.

Для формирования псевдослучайной последовательности многоразрядных кодов используют вычисления рекурсивных функций с ограничением разрядности результата.

Задание 6.4. Проект генератора псевдослучайной последовательности.

1. Проведите исследование 5-разрядного генератора псевдослучайной последовательности s64_rnd (рис. 6.12).

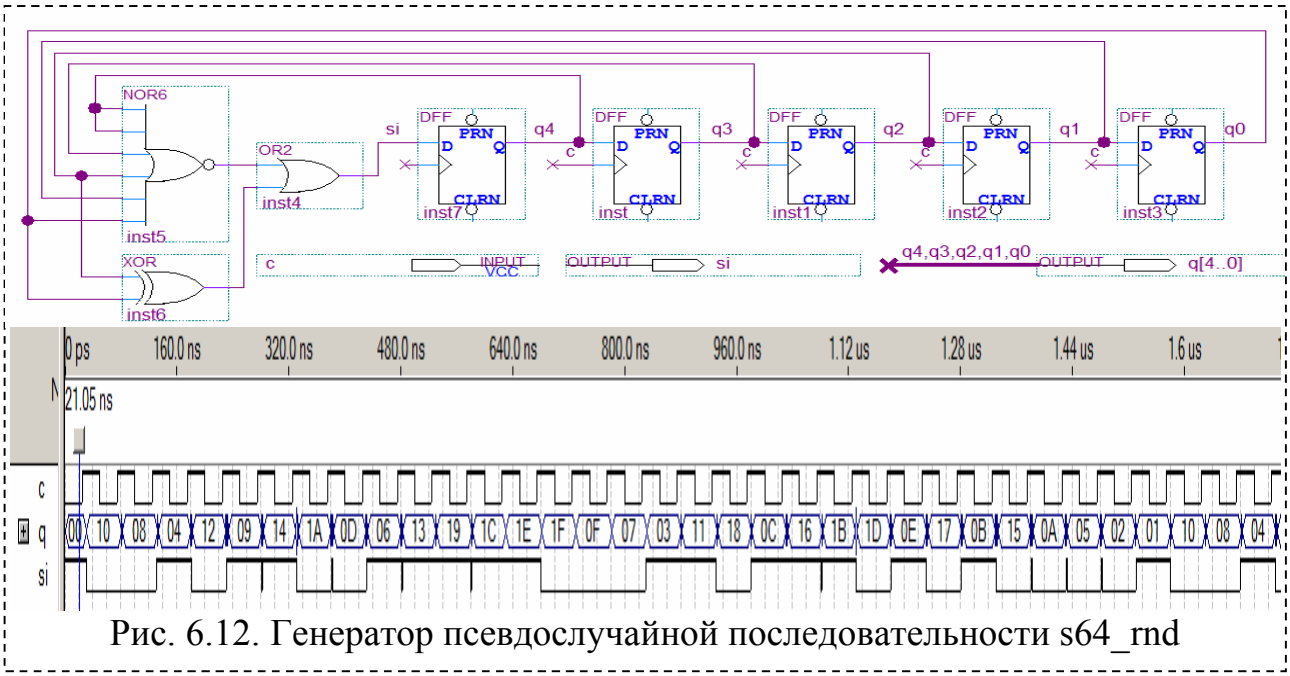

Рис. 6.12. Генератор псевдослучайной последовательности s64_rnd

При моделировании необходимо увеличить время моделирования Edit\End Nime\2u.

Определите период битовой последовательности для сигнала si. Подсчитайте за период количество одиночных, двойных и тройных единиц и нулей.

2. Измените схему, подключите входы элемента xor к выходам разрядов регистра 0 и 1. Определите период битовой последовательности для сигнала si.

Объясните результат.

§ 6.5. АЛУ аккумуляторного типа

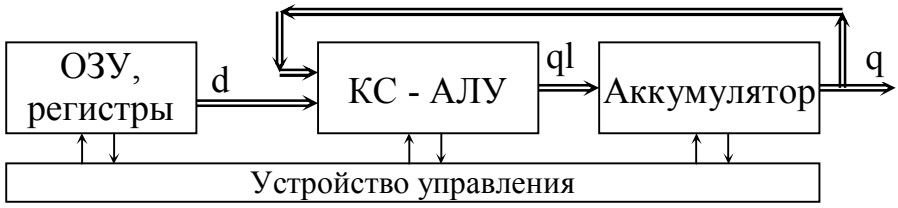

Рис. 6.13. Структура процессора аккумуляторного типа

Процессор – устройство для выполнения математической обработки данных и для управления процессом обработки. Простейшим является процессор с одним аккумулятором, используемый в микрокалькуляторах и некоторых микроконтроллерах, а также в накапливающих сумматорах, которые позволяют прибавлять к ранее полеченной сумме новые слагаемые. (рис. 6.13).

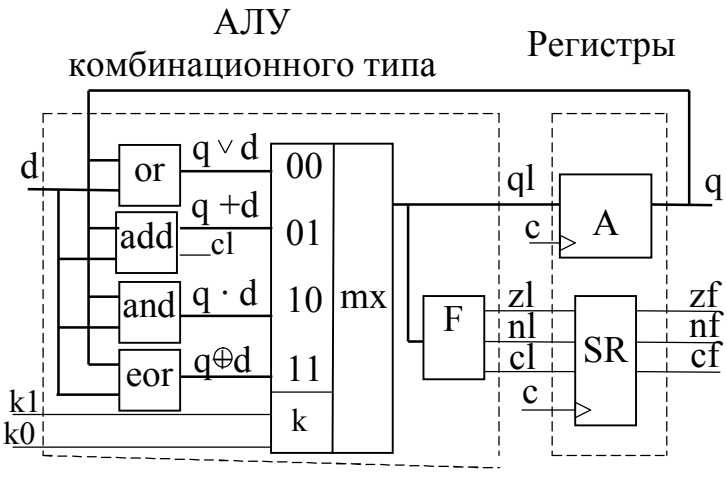

Рис. 6.14. АЛУ аккумуляторного типа

Обработка данных в указанных процессорах выполняется посредством АЛУ аккумуляторного типа, в котором используется комбинационное АЛУ (КС АЛУ) и аккумулятор.

Обрабатываемые данные хранятся в оперативном запоминающем устройстве (ОЗУ) и регистрах процессора и подаются на один из входов АЛУ через шину данных d. На другой вход АЛУ подается выход аккумулятора. На выходе АЛУ формируется логический сигнал ql. Это результат выполнения операции, аргументами которой являются текущее содержимое аккумулятора и данные d, который записывается в аккумулятор. Таким образом, процессоры данного типа выполняют одноадресные команды, которые используют в качестве одного из своих операндов исходное

содержимое аккумулятора, а после выполнения операции сохраняют результат в том же аккумуляторе.

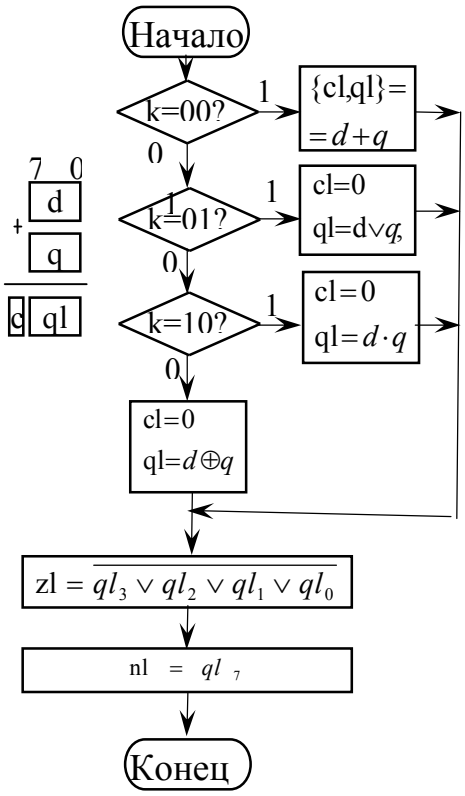

Рис. 6.14. Алгоритм работы АЛУ

Недостатком процессора аккумуляторного типа является низкое быстродействие, объясняемое тем, что каждый раз перед выполнением операции необходимо переслать в аккумулятор один из операндов, а после выполнения операции переслать результат из аккумулятора в ячейку памяти.

Арифметико-логическое устройство аккумуляторного типа содержит АЛУ комбинационного типа, рассмотренное в § 4.5, и регистры – аккумулятор A и регистр состояния SR – Status Register (рис. 6.14).

АЛУ комбинационного типа содержит набор комбинационных схем, для выполнения заданных функций, выходы которых подаются на мультиплексор mx, формирующий логический сигнал ql в соответствии с заданным кодом операции k. В качестве примера заданы коды операции и функции. Результат операции ql подается на формирователь F, выходы которого – логические сигналы признаков нуля - zl, отрицательного числа - nl, переноса – cl.

$$Если k = 00 то\ ql = q + d;$$
$$если k = 01 то ql = q \vee d;$$
$$если k = 10 то ql = q \cdot d;$$
$$если k = 11 то ql = q \oplus d.$$

Аккумулятор A и регистр признаков SR имеют динамическое управление, они предназначены для записи новых значений выходного кода и признаков результата выполняемой операции, полученных на выходах комбинационных схем в виде логических сигналов. . По фронту синхросигнала выполняется присваивание: q = ql; cf = cl; zf = zl; nf = nl. Таким образом, устройство выполняет операцию присваивания вида: q := q + d, что означает: «новое значение q равно предыдущему значению плюс текущее значение входного сигнала d».

В соответствии с заданными формулами новое значение сигнала q получается как результат присваивания результата заданной операции над текущим значением сигнала q и данными d. Например, если k=00, то сигнал ql является результатом суммирования n-разрядных кодов - содержимого аккумулятора q (это текущее значение) и входных данных d: ql = q + d. Сигнал ql подается на формирователь признаков результата F, который вырабатывает логические сигналы zl, nl, cl. В результате на выходах АЛУ комбинационного типа подготовлены новые значения выходного сигнала и признаков в виде логических сигналов, предназначенных для последующей записи в регистры A (аккумулятор) и SR (Status Register – регистр состояния или признаков).

Описание АЛУ аккумуляторного типа на Verilog составляется на основе анализа поведения схемы (рис. 6.14). Для описания комбинационных схем и элементов памяти используются отдельные операторы. Комбинационные схемы описывают, в основном, параллельными (assign) операторами, а элементы памяти – последовательными. Это было использовано в предыдущих работах.

Описание сложных комбинационных схем, например, АЛУ, выполняют последовательными операторы, которые позволяют использовать операторы if, case, скобки begin – end. Для такого описания необходимо указать тип выходного сигнала reg, а также из оператора исключить список чувствительности (не записывать «@ posedge c»), после слова always ничего не писать. В этом случае срабатывание будет происходить при изменении любого сигнала, подобно параллельному оператору.

В заголовке (строка 2) перечислены все входные и выходные сигналы, для которых требуются терминалы, они описаны в строках 3 – 9 с указанием разрядности. Внутренние сигналы устройства, к которым не предполагается подключение терминалов, должны быть

описаны: указано имя и тип wire, или reg (строки 10, 11). Для выходных сигналов, формируемых с помощью последовательных операторов, указан тип rtg.

```
// //Описание АЛУ 1
module v65_alu (c,d,k,q,ql,cf,zf,nf);//2
input c;                              //3
input [3:0]d;                         //4
input [1:0]k ;                        //5
output [3:0] q, ql;                   //6
reg [3:0] q, ql;                      //7
output cf, zf, nf;                    //8
reg cf, zf, nf;                       //9
wire   zl, nl; //11                   //10
reg cl;                               //11
always        case (k)                //12
2'b00 : {cl,ql} = q + d;              //13
2'b01 : {cl,ql} = {1'b0,(q | d)};//14
2'b10 : {cl,ql} = {1'b0,(q & d)};//15
default :{cl,ql} = {1'b0,(q ^ d)};//16
endcase                               //17
assign zl=~(ql[3]|ql[2]|ql[1]|ql[0]);//18
assign nl = ql[3];                    //19
always @ (posedge c)                  //20
begin                                 //21
q=ql; zf=zl; nf=nl; cf=cl;            //22
end                                   //23
endmodule                             //24
```

Для описания комбинационной схемы АЛУ использован последовательный оператор без списка чувствительности и оператор case, селектором которого является код операции k (строка 12). При суммировании может возникать перенос, при этом должен формироваться признак переноса cl=1. Логические операции (дизъюнкция, конъюнкция сумма по модулю два) выполняются поразрядно, при этом перенос cl =0. Для всех команд принят одинаковый формат, содержащий бит переноса cl и n – разрядный выходной код, записанный в фигурных скобках, содержащий признак переноса cl и код ql. Для логических операции cl тождественно равен 0.

АЛУ выполняет операцию, определяемую кодом k, в соответствии с заданием. При k = 00 выполняется суммирование, при k = 01 – операция ИЛИ, при k = 10 – И, при k = 11 – Исключающее ИЛИ. Список вариантов (строки 13 - 16) содержит записанные через двоеточие значения селектора и выполняемые функции. Последняя строка списка вариантов выполняет вариант с любым номером, который не был использован ранее. Завершение оператора варианта – закрывающая скобка endcase (строка 17). Для всех констант использован полный формат записи с указанием разрядности и двоичной системы счисления (первый элемент-количество разрядов, второй элемент - апостроф, третий – буква b для двоичной системы, четвертый - двоичное число).

Формирование заданных признаков описывают очевидные логические функции. Признак нуля zl = 1 , если все разряды данных равны нулю. Признак отрицательного числа nl – это старший разряд кода числа, nl = 1 для отрицательного числа. Формирование логических сигналов признаков выполняют параллельные операторы с ключевым словом assign. (строки 18 - 19)

Для описания аккумулятора и регистра состояния использован один последовательный оператор (always), содержащий операторные скобки begin – end, в которых перечислены все присваивания, выполняемые по фронту синхросигнала c. Операторы, записанные в данных скобках, будут выполняться по фронту синхросигнала последовательно, друг за другом в порядке их записи(строки 20 - 23).

Приведенный вариант описания комбинационных схем посредством последовательных операторов и скобок begin – end является мощным средством описания сложных функции. Выполняемые функции, записываемые в списке вариантов после двоеточия, могут быть весьма сложными и содержать операторы if и скобки begin – end.

Вместо строк 12 – 17 может быть использован упрощенный вариант описания с параллельным оператором условного присваивания: assign {cl,ql} = (k == 2'b00) ? q + d: (k == 2'b01) ? {1'b0,(q | d)}: (k == 2'b10) ? {1'b0,(q & d)}: {1'b0,(q ^ d)}; //(12 – 17). В этом слечае сигналы cl,ql должны быть типа wire.

Задание 6,5. АЛУ аккумуляторного типа

1) Разработайте проект 4-разрядного процессорного ядра аккумуляторного типа по приведенному описанию с именем v65_alu. Выполните моделирование (этапы 1 – 9)..
2) Опишите работу устройства,
3) Опишите методику измерения временных задержек, определите максимальную задержку.
4) Опишите предельное значение периода синхросигнала.

5) Выполните моделирование при различных значениях периода синхросигнала (равен предельному и меньше предельного).

Пояснение.

Моделирование. Подаите на входы устройства следующие сигналы:

- синхросигнал c (кнопка с циферблатом), установите Period 50 ns);
- 2-разрядный код команды k от счетчика (кнопка C), установите Count every 50 ns);
- 4-разрядный код входных данных d, установите Start Value e_{16}.

Получите временные диаграммы (рис. рис. 6.15).

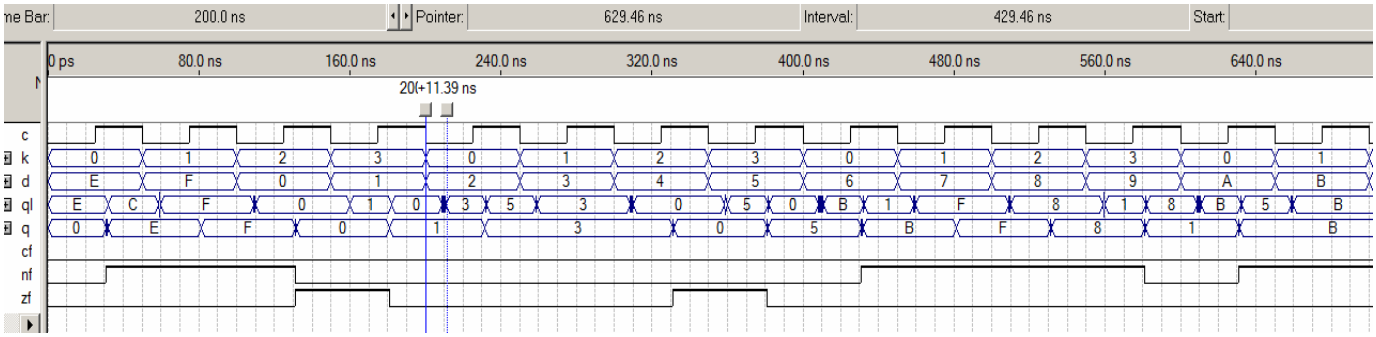

Рис. 6.15. Временные диаграммы АЛУ аккумуляторного типа

Описание работы устройства по временным диаграммам.

Вначале необходимо отметить моменты времени, соответствующие фронтам импульсов сплошными линиями, а спадам – пунктирными линиями. Спады синхроимпульсов определяет моменты изменения данных и команд и периоды выполнения команд. За время периода выполнения команды синхросигнал C принимает значения: вначале 0, а затем 1.

Интервал k = 00, выполняется суммирование: *если* $k = 00$ *то* $ql = q + d$. При C = 0 используются аргументы: q = 0 и d = e. Теоретическое значение ql определяется суммированием: $ql = 0000 + 1110 = 1\ 1110_2 = e_{16}$; $cl = 1$. при этом появляется признак отрицательного числа nl = 1. Полученные значение ql и nl записывается в аккумулятор и регистр признаков по фронту синхросигнала, формируются сигналы q и nf.

После записи нового значения q (на интервале k=00, c=1) появляется сигнал ql = c. Это результат суммирования данных d и нового значения q, он не будет использоваться, его называют «мусор».

Интервал k = 01, выполняется операция ИЛИ: *если* $k = 00$ *то* $ql = q \vee d$. При выполнении операции будут использованы аргументы, соответствующие c = 0: q = e и d = f, при этом теоретическое значение $ql = 11110 \vee 1111 = 1111_2 = f_{16}$. Полученное значение ql записывается в аккумулятор по фронту синхросигнала. В результате сигнал ql появляется на выходе q.. По результату операции ql сформирован признак nf=1, который записывается в регистр состояния по фронту синхросигнала.

Интервал k = 10, выполняется операция И: *если* $k = 10$ *то* $ql = q \cdot d$. При c = 0 используются аргументы: q = f и d = 0. Теоретическое значение ql определяется поразрядной операцией: $ql = 1111 \cdot 0000 = 0000_2 = 0_{16}$. По фронту синхросигнала в аккумулятор записывается полученное значение ql. Появляется q = 0 и zf = 1.

Определение временных задержек. Временные диаграммы (Рис. 6.15) показывают, что

Каждая операция в устройстве (рис. 6.14) выполняется в два этапа. Это формирование логического сигнала в комбинационной схеме и запись его в память. Выполнение данных этапов сопровождается временными задержками сигналов.

1 этап. Формирование на выходах комбинационных схем (АЛУ и формирователей признаков) логических сигналов ql, nl, cl, zl при c = 0. Необходимо измерить временную задержку в АЛУ при выполнении операции, имеющей максимальные задержки (например, суммирование, k = 0). Интервалом выполнения этой операции (Ткс) является участок диаграмм, для которого c = 0 и k = 0. Необходимо установить маркер на спад синхросигнала, а указатель мыши (или курсор Time Bar) на начало стабильного состояния ql. Результат - Ткс = 10,5 нс (рис. 6.16).

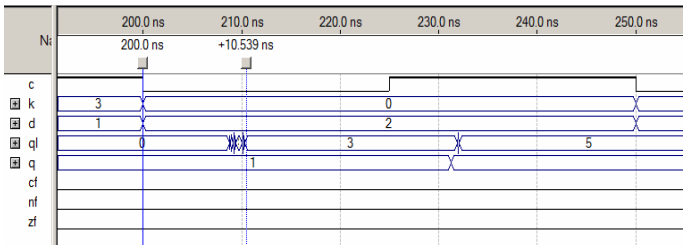

Рис. 6.16. Задержка в КС

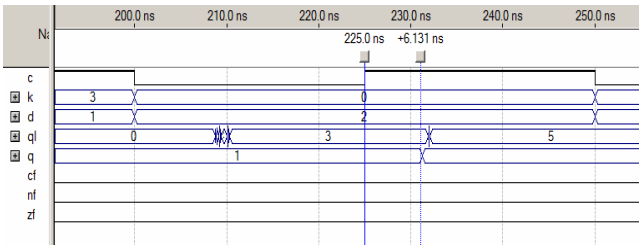

Рис. 5.17. Задержка записи в память

2 этап. Запись выхода КС в память. Начало этапа – фронт синхросигнала, окончание – получение q = ql. Результат: время записи Тзап = 6,1 нс (рис. 6.17). Максимальное значение имеет задержка в КС: Tmax = Ткс = 10,5 нс.

Определение минимального периода синхронизации и максимальной рабочей частоты. Работоспособность устройства будет обеспечена, если длительность интервала, соответствующего значению с = 0, превышает задержку в АЛУ: T_0 > Т кс, а также . длительность интервала, соответствующего значению с = 1, превышает задержку записи в память: T_1 > Т зап.

Импульсы синхронизации имеют вид меандра, для которого $T_0=T_1=Tc/2$. Для того, чтобы обеспечить работоспособность устройства, необходимо выбрать период синхросигнала Тс так, чтобы длительность интервала Тс/2 превышала максимальную из задержек: Тс/2> Tmax, или Тс > 2* Tmax. При этом Fc = 1/Тс.

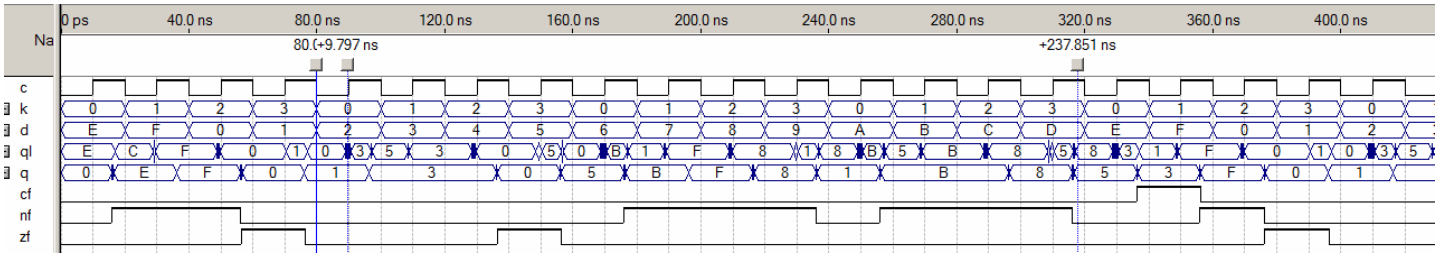

Рис. 6.18. Период синхросигнала 20 нс.

В данном случае максимальную задержку создает комбинационная схема Ткс = 10,5нс., выбираем предельное значение периода синхросигнала Тс = 20нс.

Экспериментальная проверка при установке параметров Period 20 ns и Count every 20 ns показывает, что работоспособность устройства при сохраняется. Сравнение диаграмм (рис. 6.15 и 6.18) показывает совпадение последовательностей кодов на шине q и признаков. Максимальная частота синхронизации составит: $Fc = 1/Tc = 1/20\cdot1^{0-9}=50\cdot10^6 = 50$ МГц.

Контрольные вопросы

1. Классификация, области применения регистров
2. Классификация типов сдвигов. Какие сдвиги реализуются в регистрах?
3. Опишите работу универсального регистра в режиме параллельной загрузки.
4. Опишите работу универсального регистра в режиме сдвига.
6. Изобразите алгоритм преобразования параллельного кода в последовательный.
6. Изобразите алгоритм преобразования последовательного кода в параллельный.
7. Поясните способы описания регистров на языке Verilog.
8. Какими свойствами обладает псевдослучайная последовательность кодов?
9. Какими свойствами обладает операция сумма по модулю два?
10 Изобразите схему АЛУ комбинационного типа, поясните его работу.
11. Составьте описание на Verilog АЛУ комбинационного типа.
12. Изобразите теоретические временные диаграммы для АЛУ комбинационного типа.

Работа 7. Счетчики

§ 7.1. Функциональное назначение и типы счетчиков

Счетчик – цепочка последовательно включенных счетных триггеров, каждый из которых делит частоту повторения входных импульсов на два.