# Мультиплексоры, демультиплексоры и коммутаторы

Мультиплексоры и демультиплексоры решают задачи направления потоков данных в программируемой логике. В зависимости от состояния управляющих сигналов мультиплексоры и демультиплексоры выбирают один из нескольких возможных путей прохождения данных.

Также мультиплексоры применяются для стробирования сигналов тактирования в целях снижения энергопотребления устройства.

## Мультиплексор

Мультиплексор – это комбинационное цифровое устройство, имеющее несколько входов данных, один или более управляющих входов и один выход. Мультиплексор позволяет передавать сигнал с одного из входов данных на выход; при этом выбор желаемого входа осуществляется подачей соответствующей комбинации управляющих сигналов. У мультиплексора только один выход (см. рис. 1).

Мультиплексор имеет два типа входов: информационные (входы данных) и адресные. Каждому информационному входу присваивается уникальный номер (адрес). В процессе работы мультиплексор обеспечивает подключение одного из нескольких информационных входов к единственному выходу устройства, выбор подключаемого к выходу информационного входа определяется состоянием сигналов на управляющих (адресных) входах. Другими словами в каждый момент времени в мультиплексоре только один вход данных соединен с выходом.

Сигналы на адресных входах определяют, какой конкретно информационный канал подключен к выходу. Если между числом информационных входов $N$ и числом адресных входов $M$ действует соотношение $N = 2^M$, то такой мультиплексор называют полным. Если $N < 2^M$ , то мультиплексор называют неполным.

Пример таблицы истинности для полного мультиплексора с тремя адресными входами представлен в таблице 1. У такого мультиплексора $2^3 = 8$ входов данных.

Адресные входы могут использоваться для расширения функциональных возможностей мультиплексора. Они используются для наращивания разрядности мультиплексора, синхронизации его работы с работой других узлов. Сигналы на разрешающих входах могут разрешать, а могут и запрещать подключение определенного входа к выходу, то есть могут блокировать действие всего устройства.

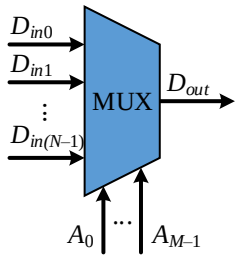


Рис. 1. Мультиплексор

*Таблица 1. Таблица истинности для полного мультиплексора с тремя адресными входами. $D_{in\,i}$ – i-й информационный вход, $A_i$ – i-й адресный вход*

| $A_2$ | $A_1$ | $A_0$ | $D_{out}$ |
|---|---|---|---|
| 0 | 0 | 0 | $D_{in0}$ |
| 0 | 0 | 1 | $D_{in1}$ |
| 0 | 1 | 0 | $D_{in2}$ |
| 0 | 1 | 1 | $D_{in3}$ |
| 1 | 0 | 0 | $D_{in4}$ |
| 1 | 0 | 1 | $D_{in5}$ |
| 1 | 1 | 0 | $D_{in6}$ |
| 1 | 1 | 1 | $D_{in7}$ |

Основное назначение мультиплексоров – управление потоками данных. Мультиплексоры также могут использоваться в делителях частоты (в современных ПЛИС так делать категорически не рекомендуется), триггерных устройствах, сдвигающих устройствах и др. Мультиплексоры могут использоваться для реализации логических функций многих переменных.

## Демультиплексор

Демультиплексор – это комбинационное цифровое устройство, предназначенное для переключения сигнала с одного информационного входа на один из информационных выходов, при этом выбор желаемого выхода осуществляется подачей соответствующей комбинации управляющих (адресных) сигналов. Демультиплексор в функциональном отношении противоположен мультиплексору (см. рис. 2). В каждый момент времени только один из выходов демультиплексора соединен с входом данных.

Если между числом выходов $N$ и числом адресных входов $M$ действует соотношение $N = 2^M$ для двоичных демультиплексоров, то такой демультиплексор называют полным. Если $N < 2^M$ для двоичных демультиплексоров, то демультиплексор называют неполным. Функции демультиплексоров сходны с функциями дешифраторов. Дешифратор можно рассматривать как демультиплексор, у которого информационный вход поддерживает напряжение выходов в активном состоянии, а адресные входы выполняют роль входов дешифратора.

В таблице 2 приведена таблица истинности для полного демультиплексора с тремя адресными входами. У такого демультиплексора $2^3 = 8$ выходов данных.

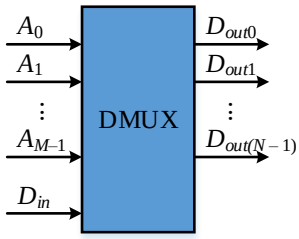


*Рис. 2. Демультиплексор*

*Таблица 2. Таблица истинности для полного демультиплексора с тремя адресными входами и восемью информационными выходами*

| $A_2$ | $A_1$ | $A_0$ | $D_{out0}$ | $D_{out1}$ | $D_{out2}$ | $D_{out3}$ | $D_{out4}$ | $D_{out5}$ | $D_{out6}$ | $D_{out7}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | $D_{in}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | $D_{in}$ | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | $D_{in}$ | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | $D_{in}$ | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | $D_{in}$ | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | $D_{in}$ | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $D_{in}$ | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $D_{in}$ |

Также, как и шифраторы с дешифраторами, мультиплексоры и демультиплексоры могут дополняться входом разрешения работы **ce** (chip enable). Работа устройства выполняется только, если на этот вход поступает активный логический сигнал. Иначе все выходы неактивны вне зависимости от состояния входов.

## Коммутатор

При проектировании цифровых устройств часто возникает задача направления потока данных от одного из множества источников к одному из множества получателей. Т.е. необходимо устройство, которое подключает (коммутирует) определенные входные порты к определенным выходным портам. Эту задачу решает коммутатор.

В нашем курсе внимание будет уделено коммутатору, который в каждый момент времени работает только с одной парой входного и выходного портов. Такой коммутатор имеет $N_{in}$ информационных входов, $N_{out}$ информационных выходов, $M_{in}$ и $M_{out}$ адресных входов для выбора соединяемых входного и выходного информационных портов соответственно.

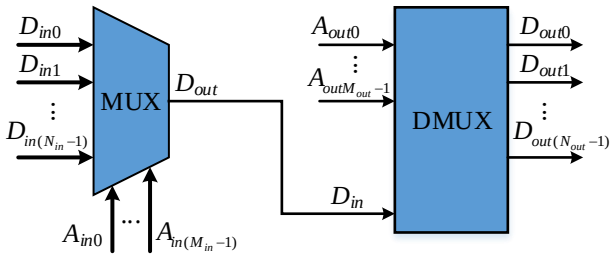


*Рис. 3. Последовательное соединение мультиплексора и*

*Таблица 3. Таблица истинности для коммутатора 4x4.*

| $A_{in1}$ | $A_{in0}$ | $A_{out1}$ | $A_{out0}$ | $D_{out0}$ | $D_{out1}$ | $D_{out2}$ | $D_{out3}$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $D_{in0}$ | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | $D_{in0}$ | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | $D_{in0}$ | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | $D_{in0}$ |
| 0 | 1 | 0 | 0 | $D_{in1}$ | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | $D_{in1}$ | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | $D_{in1}$ | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | $D_{in1}$ |

*демультиплексора для построения коммутатора*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | $D_{in2}$ | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | $D_{in2}$ | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | $D_{in2}$ | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | $D_{in2}$ |
| 1 | 1 | 0 | 0 | $D_{in3}$ | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | $D_{in3}$ | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | $D_{in3}$ | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | $D_{in3}$ |

Если соединить выходы демультиплексора со входами мультиплексора (см. рис. 4) при $N_{in} = N_{out}$, то получится устройство с одним информационным входом и одним информационным выходом. Такое устройство позволяет разделять (разуплотнять) один высокоскростной канал данных на множество низкоскоростных, что упрощает обработку данных в каждом низкоскоростном канале. После того, как данные обработаны, их можно снова объединить в один высокоскоростной канал. По такому принципу работают современные высокоскоростные последовательные трансиверы в FPGA.

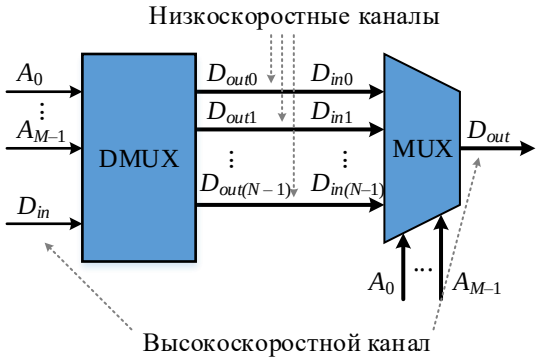


*Рис. 4. Последовательное соединение демультиплексора и мультиплексора для разуплотнения высокоскоростного канала*

## Особенности FPGA

### Моделирование цифровых устройств

В процессе разработки цифрового устройства бо́льшую часть времени обычно занимает отладка проекта. Для этого на первых этапах отладки используют программные средства моделирования на регистровом уровне, а на последних – внутрисхемные отладчики и моделирование на физическом уровне (с учетом параметров конкретной FPGA).

Корректно выполненное моделирование устройства целиком и его отдельных частей на первых этапах отладки – залог успешной и быстрой разработки. Зафиксируем внимание на том, что в больших составных проектах тщательному тестированию подлежит не только устройство целиком, но и отдельные его блоки[1]. Также хорошим стилем разработки является разделение[2] тестбенчей и синтезируемых модулей.

Для того чтобы протестировать какой-либо модуль, разрабатывается специальный тестирующий модуль – тестбенч (test bench). Тестбенч включает в себя в качестве компонента тестируемый модуль и генерирует сигналы для всех входов тестируемого модуля. Тестбенч также может дополнительно обрабатывать сигналы с выходов тестируемого модуля (например, записывать результаты в файл). В простых тестах все входные сигналы могут создаваться непосредственно в тестбенче (сигналы сброса, тактирования, счетчики и т.д.). В более сложных

[1] В идеальном случае тестируются все компоненты устройства. В реальности – времени на всеобъемлющее тестирование может не хватить, тестируются все крупные узлы устройства.

[2] Использование тестирующих вставок кода (обычно, несинтезируемых) в синтезируемом модуле не является в общем случае ошибкой. При синтезе проекта в консоль будут выданы только предупреждения, а не сообщения об ошибках.

тестах удобно тестовые последовательности генерировать в какой-либо другой среде разработки (например, в Matlab) и с помощью тестбенча подавать их на тестируемый модуль.

Разработка тестбенча может выполняться на любом языке программирования цифровых устройств (VHDL, Verilog, System Verilog и др.) или графически. В отличие от модулей, подлежащих прошивке в FPGA, в тестбенчах могут применяться несинтезируемые синтаксические конструкции языков программирования (например, работа с файлами, задержки на интервалы времени).

Для моделирования работы цифровых устройств можно использовать различные симуляторы: ISim (встроенный в ISE Design Suit), Modelsim и др.