Контроллер памяти DDR3 SDRAM

Алексей Гребенников (г. Актау, Казахстан)

Статья содержит описание контроллера памяти DDR3 SDRAM на языке Verilog. Рассмотрены принципы действия блоков контроллера и схемы конечных автоматов основных модулей. Подробно описана схема ввода/вывода ПЛИС Spartan 6 с использованием блоков ISERDES2/OSERDES2 и линий задержки IODELAY2.

Введение

Динамическая память получила очень широкое распространение в современных электронных устройствах. Низкая стоимость хранения байта информации по сравнению с другими типами энергозависимой памяти позволяет включать значительные объёмы динамической памяти даже в недорогие устройства. По мере развития технологии производства динамической памяти, появляются новые семейства – DDR , DDR2 и DDR3 SDRAM. Каждое последующее семейство работает на более высокой частоте и, соответственно, имеет более высокую скорость передачи данных.

Элементом хранения информации в динамической памяти является конденсатор, поэтому требуется периодическая регенерация всех элементов массива для предотвращения потери информации. Управление массивом осуществляется при помощи контроллера, который регенерирует память через определённые промежутки времени и выполняет операции чтения/записи. Минимальная рабочая частота памяти DDR3 SDRAM при включенном блоке DLL составляет 303 МГц, что предъявляет повышенные требования к быстродействию контроллера памяти при его реализации на недорогих ПЛИС, таких как Xilinx Spartan 6.

В статье рассмотрена модель контроллера памяти DDR3 SDRAM на языке Verilog, реализованная для ПЛИС Spartan 6, которая входит в состав отладочной платы SP605. Проект был разработан в среде PlanAhead v13.3, симуляция проводилась с использованием программы ModelSim 6.4a. Исходные файлы проекта находятся в архиве DDR3_sources.zip на интернет-странице журнала.

На плате SP605 установлена микросхема DDR3 SDRAM MT41J64M16LA-187E фирмы Micron. Эта память объёмом 1 Гбит организована по схеме 8M×16×8, т.е. содержит восемь банков; ширина шины данных составляет 16 бит. Более подробная информация об этой микросхеме памяти и принципах работы памяти DDR3 SDRAM содержится в [1]. В статье [2] приведено описание контроллера более медленной динамической памяти – DDR SDRAM. Этот материал также может быть полезен при построении контроллера памяти DDR3 SDRAM.

Система на кристалле

Файл верхнего уровня проекта sopc.v, блок-схема которого изображена на рисунке 1, содержит в своём составе контроллеры UART, шины Wishbone и памяти DDR3 SDRAM, а также блоки синтеза частоты (PLL, ФАПЧ) и управления портами ввода/вывода.

Контроллер UART [3] формирует команды чтения/записи данных на шине Wishbone. Контроллер памяти декодирует эти команды и генерирует управляющие сигналы для памяти, которые вместе с данными передаются через блок контроля портов ввода/вывода. Частоты, необходимые для работы памяти DDR3 SDRAM и контроллера, формируются блоком синтеза частот. В данной версии контроллера используются следующие частоты:

- 33 МГц – рабочая частота шины Wishbone и UART; также используется в контроллере памяти в цепях интерфейса с вышеуказанными блоками;
- 83 МГц – основная рабочая частота контроллера;
- 167 МГц – частота ввода/вывода, на которой конечные автоматы контроллера памяти взаимодействуют с блоком контроля ввода/вывода;
- 667 МГц – высокая частота блока ввода/вывода, из которой формируется рабочая частота памяти 333,5 МГц.

Обозначения сигналов на языке Verilog отвечают следующим правилам:

- все сигналы, имеющие в своём названии _n_, активны при лог. 0, в противном случае – при лог. 1. Например, сигнал reset_n_o активен при лог. 0, а сигнал cke_o – при лог. 1;
- если сигнал оканчивается на _o, значит, это выходной сигнал модуля (output), если на _i – значит, сигнал является входом модуля (input), _io – двунаправленный сигнал. Примеры: adr_o – выходной сигнал, rst_i – входной сигнал, dq_io – двунаправленный сигнал;
- если название сигнала оканчивается на _w, значит, он относится к типу «провод» (wire); если на _r – сигнал является регистром. Например, rd_w, rd_done_r.

Рассмотрим более подробно работу контроллера памяти.

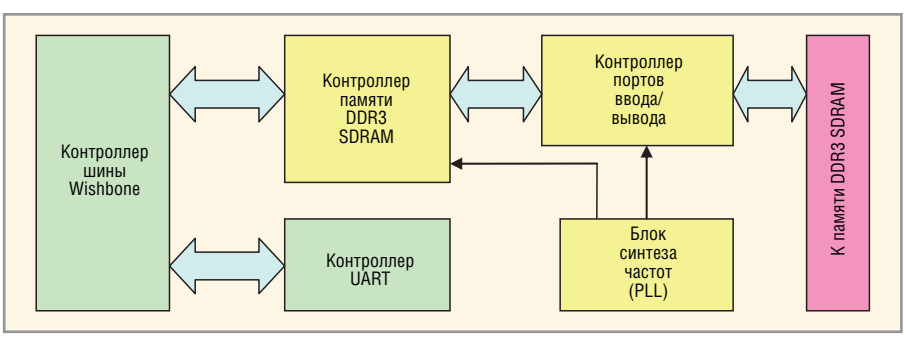

Рис. 1. Блок-схема системы на кристалле

Контроллер памяти

Файл верхнего уровня контроллера памяти расположен в файле ddr3_top.v, а его блок-схема изображена на рисунке 2. Контроллер шины Wishbone работает только в режиме ведомого устройства (slave). Этот контроллер декодирует команды и обеспечивает приём/передачу данных; 32-битный адрес на шине Wishbone передаётся модулю трансляции адреса. В этом модуле 32-битный адрес разбивается на адреса банка, строки и столбца по следующему правилу:

```
assign c_adr_o =
{adr_i[10:4],3'b000};
assign r_adr_o = adr_i[23:11];
assign ba_o = adr_i[26:24];
```

Поскольку память имеет объём 1 Гбит = 128 Мб, для её полной побайтной адресации требуется 27 линий адреса. Три старших бита, 26 – 24, адресуют банк памяти. Следующие тринадцать бит, 23 – 11, адресуют строки. Для адресации столбцов используются десять адресных линий. Однако линии шины Wishbone 3 – 1 заменяются нулями, так как обмен с памятью производится фиксированными пакетами длиной по восемь 16-битных слов. В связи с тем, что обмен данными производится 16-битными словами, нулевой бит адреса не используется для адресации столбцов.

После включения питания контроль шины адреса и команд памяти передаётся конечному автомату инициализации, при этом основной конечный автомат и счётчик находятся в режимах ожидания. После выполнения процедуры инициализации памяти и калибровки блока ввода/вывода сигнал завершения инициализации init_done_w принимает значение лог. 1, и основной конечный автомат и счётчик начинают работать в основном режиме. Конечный автомат ddr_fsm обрабатывает команды от основного счётчика и контроллера шины Wishbone. Задача основного счётчика – отсчёт времени с момента последней регенерации памяти. Когда значение счётчика достигает T_REFI (константа в файле ddr3_parms.v), сигнал необходимости регенерации ref_req принимает значение лог. 1:

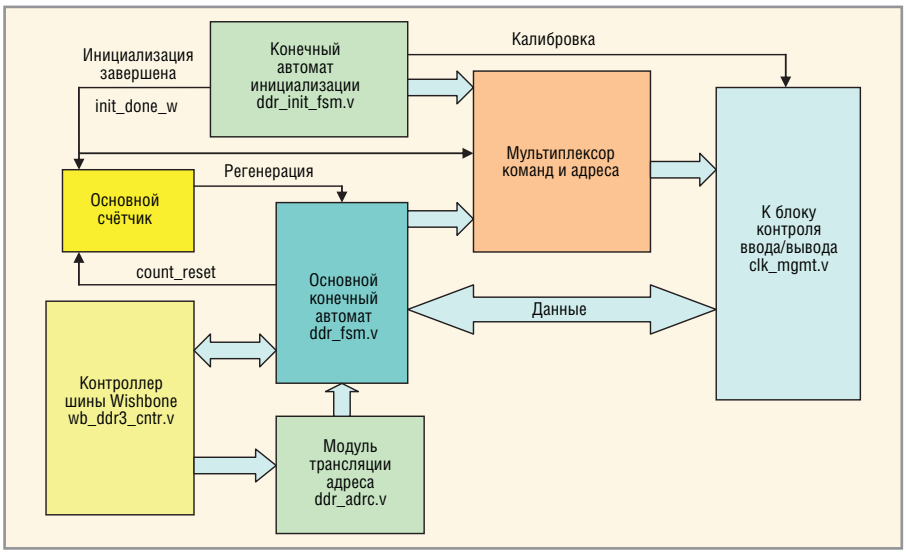

Рис. 2. Блок-схема контроллера памяти

```
if ((ddr_count == `T_REFI) &&
init_done_w)
ref_req <= 1'b1;
```

При получении этого сигнала конечный автомат начинает процедуру регенерации памяти. После завершения регенерации основной счётчик обнуляется сигналом count_reset и начинает отсчёт времени снова. Команда счётчика имеет более высокий приоритет, чем команды контроллера шины Wishbone, так как своевременная регенерация памяти необходима для сохранения целостности данных, а операции чтения/записи можно отложить.

Рассмотрим более подробно работу отдельных блоков контроллера памяти.

Конечный автомат инициализации

Для правильной работы динамической памяти необходима её первоначальная инициализация. Эту процедуру выполняет специальный конечный автомат, исходный код которого находится в файле ddr_init_fsm.v, а блок-схема изображена на рисунке 3.

После подачи питания или после системного сброса включается счётчик времени init_count, при этом конечный автомат находится в состоянии DDR_I_IDLE. В этом состоянии активируется сигнал сброса памяти reset_n_o = 1'b0 и отключается сигнал разрешения тактового сигнала cke_o = = 1'b0. Через промежуток времени I_WAIT_T автомат переходит в состояние DDR_I_WAIT. При этом деактивируется сигнал сброса памяти reset_n_o,

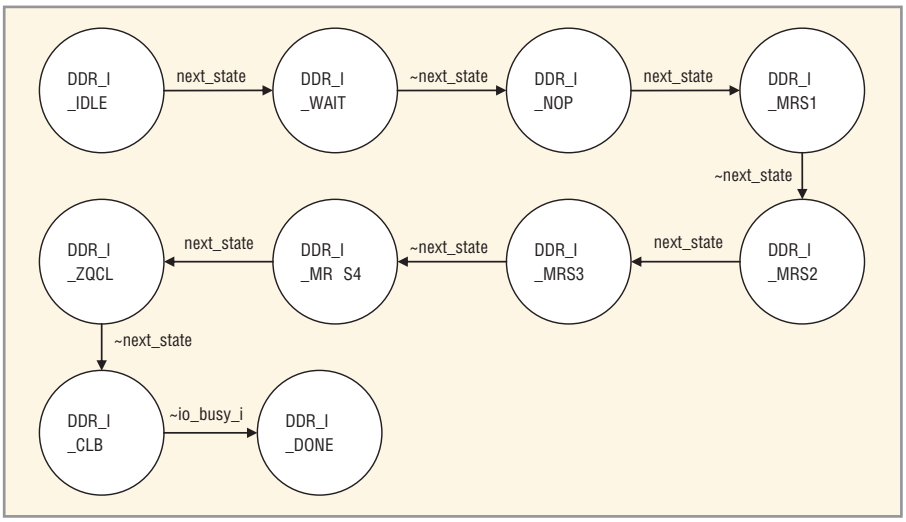

Рис. 3. Конечный автомат инициализации

а сигнал cke_o сохраняет значение лог. 0. Когда счётчик времени достигает значения I_NOP_T, автомат переходит в состояние DDR_I_NOP. В этом состоянии сигнал cke_o принимает значение лог. 1, т.е. входной тактовый сигнал начинает обрабатываться памятью.

В момент времени I_MRS1_T начинается инициализация контрольных регистров памяти MR0–MR3. Все команды задаются 3-битным регистром opcode_o, который подсоединяется к управляющим линиям памяти ras_n_o (Raw Access Strobe), cas_n_o (Column Access Strobe), we_n_o (Write Enable). При этом команда должна быть активна в течение одного периода тактовой частоты памяти; всё остальное время регистр opcode_o должен иметь значение OP_NOP (NO Operation) – нет активных команд. Список всех команд памяти приведён в файле параметров ddr3_parms.v. Конечный автомат работает на частоте 83 МГц, а команды для памяти выдаются с частотой 167 МГц. Блок ввода/вывода, работа которого будет рассмотрена ниже, преобразует частоту выдачи команд в необходимую для DDR3 частоту 333,5 МГц. Параметры для команд инициализации передаются через адресные линии adr_o и линии адресации банков ba_o.

Динамическая память DDR3 SDRAM имеет четыре контрольных регистра MR (Mode Register), задающих важнейшие параметры работы памяти. Регистры инициализируются в последовательности MR2–MR3–MR1–MR0. Рассмотрим некоторые параметры, задаваемые этими регистрами.

Регистр MR0. Burst length – длина пакета данных. Динамическая память может обмениваться только пакетами данных. Для DDR3 – это восемь единиц данных, четыре единицы, а также возможность устанавливать размерность четыре или восемь перед каждой транзакцией (on the fly mode). Под единицей данных понимается размерность шины данных памяти. В данном примере размерность шины равна 16, длина пакета 8, т.е. каждая операция чтения или записи содержит пакет данных размером 16 бит × 8 = 16 байт данных. CAS Latency – задержка для операций чтения между выдачей команды и данных памяти; для данного примера равна 7.

Регистр MR2. CWL – CAS Write Latency. Задержка для операций записи между выдачей команды и данных для записи. В рассматриваемом контроллере CWL = 6.

Регистр MR3. MPR – Multi Purpose Register. При активации этого регистра память вместо данных из массива для всех операций чтения выдаёт значение регистра MPR, т.е. заранее заданную последовательность данных. Этот регистр удобно использовать во время настройки контроллера для проверки корректности операций чтения. В нормальном режиме работы регистр MPR деактивирован.

Более подробно функции регистров, а также все временные задержки памяти DDR3 SDRAM описаны в [1].

Для всех команд время активации контролируется счётчиком hspd_count, работающим на частоте 167 МГц. Этот счётчик в заданные промежутки времени устанавливает в лог. 1 линию cmd_out на время одного периода. Именно в этот период времени выдаётся активная команда:

```
if (cmd_out)
begin
ba_o = 3'b000;
adr_o = `IA_MRS0;
opcode_o = `OP_MRS;
end
```

ba_o задаёт номер регистра MR (при ba_o = 3'b000 программируется регистр MR0), adr_o – значение регистра, opcode_o – команда конфигурации регистров.

При cmd_out = 1'b0 все выходные сигналы, указанные в примере, принимают значения по умолчанию для конечного автомата:

```
ba_o = 3'b000;
        adr_o = 13'h0;
opcode_o = `OP_NOP;
```

После инициализации всех четырёх регистров автомат переходит в состояние DDR_I_ZQCL. В этом состоянии память выполняет операции калибровки. От контроллера не требуется каких-либо активных действий, кроме выдачи команды OP_ZQCL. В момент времени I_ZQCL_T, т.е. после завершения калибровки памяти, автомат переходит в следующее состояние – DDR_I_CLB. В этом состоянии автомат ждёт завершения процесса калибровки блоков IODELAY линий данных памяти dq (эти блоки находятся в файле sopc.v).

Весь процесс калибровки блоков IODELAY проходит в несколько этапов. В состоянии автомата DDR_I_MRS4 активируется линия io_clb_o. Это – сигнал начала калибровки для линий IODELAY. В семействе ПЛИС Spartan 6 калибруется только входная задержка, а выходная должна быть задана. В данной реализации контроллера входная задержка данных блока IODELAY относительно тактового сигнала сконфигурирована как VARIABLE FROM ZERO. При таком значении параметра задержка данных относительно тактового сигнала равна нулю. Процесс калибровки занимает 12 – 20 циклов тактового сигнала. После завершения калибровки IODELAY блок запоминает значение внутреннего 8-битного регистра, при котором получается нулевая задержка. Однако это новое значение вступает в силу только после активации сигнала сброса блока. Именно это и происходит в момент перехода конечного автомата из состояния DDR_I_ZQCL в состояние DDR_I_CLB. Сигнал io_rst_o активируется на время одного периода тактового сигнала (частотой 83 МГц).

До момента вступления в силу новой величины задержки, обеспечивающей

нулевой сдвиг данных относительно тактового сигнала, блоки IODELAY удерживают сигнал BUSY в активном состоянии. По количеству линий данных – 16, – этих сигналов также шестнадцать. Логическое ИЛИ всех сигналов BUSY из модуля sopc передаётся в модуль ddr_init_fsm в виде сигнала io_busy_i. Конечный автомат в состоянии DDR_I_CLB ожидает деактивации сигнала io_busy_i, т.е. момента времени, когда все 16 линий данных установят нулевую задержку данных. После того как сигнал io_busy_i становится равным нулю, процесс инициализации памяти считается законченным, и конечный автомат переходит в состояние DDR_I_DONE, в котором он остаётся до следующего системного сброса. При этом активируется основной конечный автомат ddr_fsm, запускается основной счётчик времени ddr_count (в модуле ddr3_top), и мультиплексоры адреса и команд также переключаются на основной конечный автомат.

Основной конечный автомат

Этот автомат выполняет все основные операции взаимодействия с памятью. Его исходный код находится в файле ddr_fsm.v, а блок-схема состояний изображена на рисунке 4.

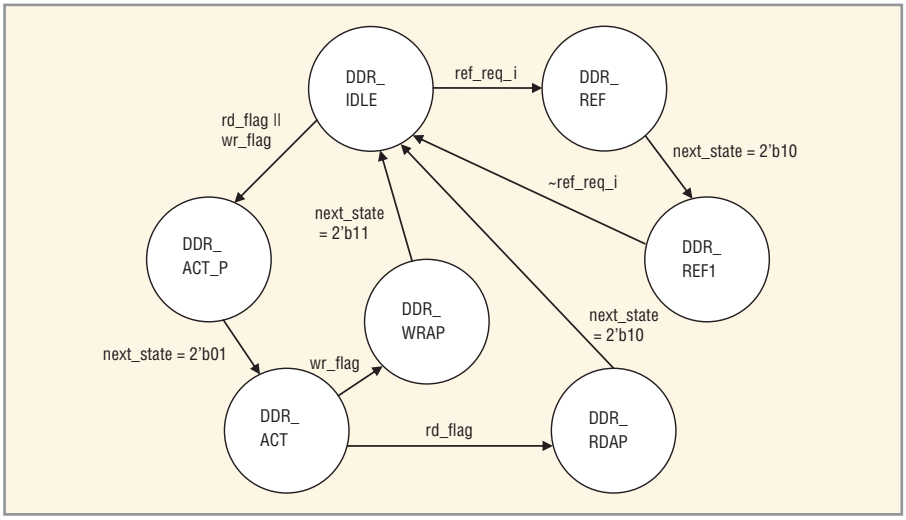

Рис. 4. Основной конечный автомат

Конечный автомат контроллера имеет те же состояния, что и автомат памяти, полная схема которого приведена в [1]. Однако автомат контроллера содержит только необходимый минимум, достаточный для функционирования памяти – из операций чтения и записи реализованы варианты с предзарядом (WRAP – Write with Auto Precharge, RDAP – ReaD with Auto Precharge).

После завершения инициализации основной конечный автомат начинает свою работу в состоянии DDR_IDLE, ожидая дальнейших команд от контроллера шины Wishbone или от основного счётчика. Запросы на регенерацию памяти поступают от счётчика с периодичностью T_REFI. При этом автомат переходит сначала в состояние DDR_REF. В этом состоянии памяти выдаётся команда OP_PRE (precharge) – закрытие и предзаряд активных банков.

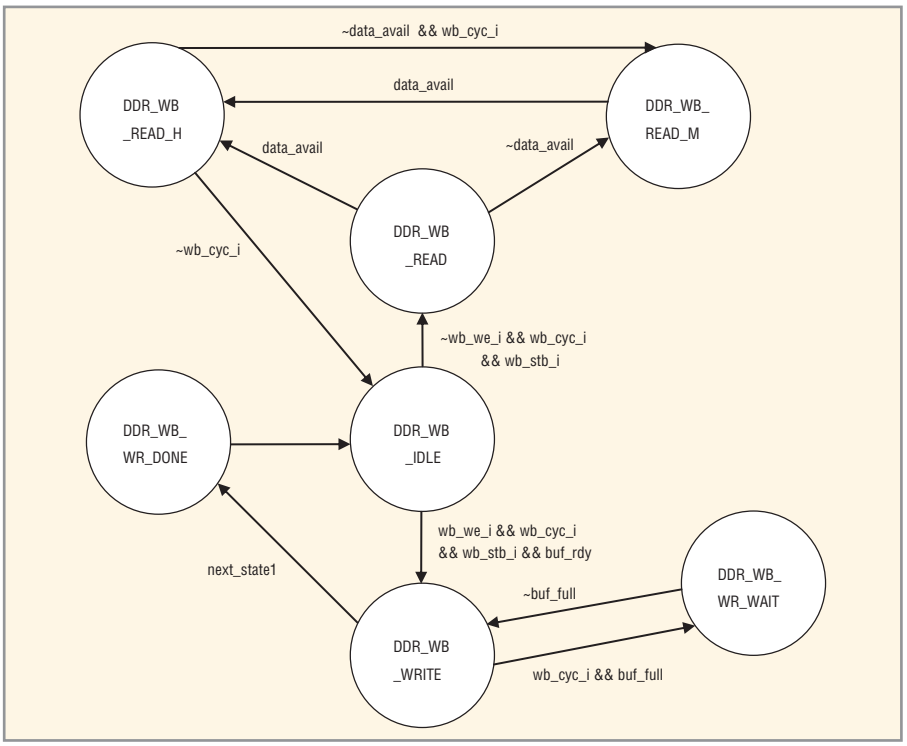

Рис. 5. Конечный автомат блока интерфейса с шиной Wishbone

Одновременно с этой командой устанавливается в лог. 1 линия адреса adr_o[10] – сигнал того, что необходима подзарядка всех банков. Затем через промежуток времени T_RP автомат переходит в состояние DDR_REF1 – вторая стадия процесса регенерации. В этом состоянии памяти выдаётся команда OP_REF (refresh) – регенерация строки. Адрес для регенерации формируется памятью автоматически, поэтому от контроллера требуется только выдача команды. Через промежуток времени T_RP + T_RFC процесс регенерации завершается, и автомат вновь переходит в состояние ожидания.

При запросе чтения/записи первоначальные действия конечного автомата одинаковы как для операции чтения, так и для операции записи – необходимо активировать строку. Для этого автомат переходит в состояние DDR_ACT_P и выдаёт памяти команду OP_ACT. В этот момент на адресных линиях банка и строки должны быть достоверные данные, полученные в результате преобразования адреса модулем ddr_adrc. Через промежуток времени T_RCD конечный автомат переходит в состояние DDR_ACT, что сигнализирует об окончании процесса открытия строки.

Далее определяется, что необходимо выполнить – запись или чтение. При равенстве единице регистра rd_flag автомат переходит в состояние DDR_RDAP, т.е. выполняет операцию чтения. Во время инициализации память была сконфигурирована на 8-пакетный приём данных – каждая операция считывает пакет размером 8×16 бит. Блок, непосредственно считывающий данные памяти во внутренний буфер, находится в модуле ddr3_top. В зависимости от латентности памяти для операций чтения, также задаваемой во время инициализации, устанавливается параметр RD_START. Память типа DDR (Double Data Rate) выдаёт данные как на положительном, так и на отрицательном фронтах синхросигнала.

В данной версии контроллера частота памяти равна 333,5 МГц, однако блок записи данных памяти работает на частоте вдвое ниже – примерно 167 МГц. Это происходит потому, что данные на скорости 333,5 МГц для обоих фронтов принимаются блоками ISERDES, работа которых будет рассмотрена ниже, и затем эти данные на пониженной частоте записываются во внутренний буфер. То есть за один период сигнала 167 МГц память передаёт четыре единицы данных – 4×16. Именно поэтому ширина шины dq_i в модуле ddr3_top равна 64 битам. Следовательно, для приёма полного пакета данных размером 8×16 бит требуются две операции чтения на частоте 167 МГц:

```
if (rd_count_pos == `RD_START)
{bufr[1],bufr[0]} <= dq_i;
else if (rd_count_pos ==
(`RD_START + 1))
begin
{bufr[3],bufr[2]} <= dq_i;
rd_done_r <= 1'b1;
end
```

Внутренние регистры буфера bufr имеют размерность 32 бита, счётчик rd_count_pos отсчитывает время с момента выдачи команды чтения. После завершения операции чтения конечный автомат ожидает промежуток времени, необходимый для подзарядки банков (так как операция чтения задаётся с автоматическим предзарядом), и после этого переходит в состояние DDR_IDLE.

Рассмотрим цикл записи. Для операции записи также вначале выдаётся команда активации строки. Затем конечный автомат из состояния DDR_ACT переходит в состояние DDR_WRAP и выдаёт команду OP_WRITE с установленным в лог. 1 битом адреса adr_o[10] – команда записи с автоматическим предзарядом банков. Блок, непосредственно записывающий данные в память, также находится в модуле ddr3_top. Параметр WR_START зависит от латентности памяти для записи CWL (CAS Write Latency), задаваемой во время инициализации. По аналогии с операцией чтения, данные выдаются пакетами по 64 бита на частоте 167 МГц в режиме работы SDR (Single Data Rate) только на отрицательном фронте синхросигнала. Затем эти данные преобразуются в блоки по 16 бит на частоте 333,5 МГц для режима работы DDR.

Через промежуток времени T_WRAP конечный автомат переходит в состояние DDR_IDLE – запись блока данных и предзаряд банков (precharge) завершены.

Команды записи и чтения выдаются блоком управления шиной Wishbone. Рассмотрим более подробно работу этого блока.

Блок интерфейса с шиной Wishbone

Блок интерфейса с шиной Wishbone, исходный код которого приведён в файле wb_ddr3_cntr.v, обеспечивает связь контроллера памяти с другими устройствами на шине Wishbone. Он также содержит конечный автомат, изображённый на рисунке 5. В состоянии DDR_WB_IDLE автомат ожидает запросы на чтение или запись, которые регистрируются по значениям управляющих линий шины Wishbone wb_we_i, wb_cyc_i и wb_stb_i. При получении запроса на чтение автомат пе-

реходит в состояние DDR_WB_READ. Как упоминалось выше, считанные из памяти данные хранятся во внутреннем 128-битном буфере, при этом адрес начала буфера запоминается в 28-битном регистре – указателе:

```
burst_adr <= wb_adr_i[31:4];
```

Соответственно, при поступлении запроса на чтение, сначала проверяется наличие необходимых данных во внутреннем буфере. Если сигнал data_avail равен лог. 1, значит, буфер уже содержит необходимые данные:

```
if (wb_adr_i[31:4] == burst_adr)
data_avail <= 1'b1;
```

При этом конечный автомат переходит в состояние DDR_WB_READ_H (Read Hit). Данные считываются из внутреннего буфера, и цикл чтения памяти DDR3 SDRAM не осуществляется. Из состояния DDR_WB_READ_H возможно несколько выходов. При завершении цикла чтения шины Wishbone (сигнал wb_cyc_i переходит в лог. 0) автомат переходит в состояние ожидания DDR_WB_IDLE. Если же требуется прочитать больше данных, чем содержится в буфере, автомат из состояния DDR_WB_READ_H переходит в состояние DDR_WB_READ_M (Read Miss) и генерируется запрос основному конечному автомату контроллера памяти ddr_fsm на чтение одного пакета данных в 128 бит. Затем конечный автомат контроллера шины Wishbone ожидает завершения цикла чтения и установления линии data_avail в лог. 1. После этого автомат снова возвращается в состояние DDR_WB_READ_H и считывает необходимые данные из внутреннего буфера.

Переходы из состояния DDR_WB_READ_H в DDR_WB_READ_M продолжаются до тех пор, пока не будут считаны все данные, запрашиваемые через шину Wishbone. Например, при чтении шестнадцати 32-битных слов таких переходов будет четыре – за один переход считывается четыре 32-битных слова. После завершения цикла чтения автомат переходит в режим ожидания. Если при чтении данные отсутствуют в буфере изначально, то автомат сразу переходит в состояние DDR_WB_READ_M. Затем цикл продолжается так же, как было описано выше.

Логика перехода автомата в состояние записи данных DDR_WB_WRITE аналогична – проверяются контрольные сигналы шины Wishbone wb_we_i, wb_cyc_i и wb_stb_i. Однако дополнительно проверяется сигнал готовности буфера buf_rdy. Поэтому новые данные для записи не могут быть приняты до тех пор, пока текущие не будут записаны в память. Размерность буфера записи, так же как и буфера чтения, равна длине пакета памяти (burst length) 8×16 бит, или четыре 32-битных слова. При записи более четырёх 32-битных слов за один цикл шины Wishbone автомат периодически переходит из состояния DDR_WB_WRITE в состояние DDR_WB_WR_WAIT и обратно до тех пор, пока все данные не будут записаны.

Поскольку длина пакета записи памяти DDR3_SDRAM фиксирована и равна 16 байтам, а шина Wishbone позволяет адресовать отдельные байты данных, необходим дополнительный механизм, обеспечивающий запись только необходимых данных в пределах всего пакета. Например, требуется записать только один байт по адресу 32'h80000000, а все оставшиеся данные (пакет данных будет содержать байты по адресам 32'h80000000 – 32'h8000000F) оставить без изменения. Для этой цели в памяти DDR3 SDRAM служат линии маскировки.

Для 16-битной шины данных памяти определено две линии маскировки – для верхнего байта и для нижнего байта. При лог. 0 на соответствующей линии в момент записи байт записывается в память, а при значении лог. 1 линии маскировки – не записывается, маскируется. Поэтому при записи данных во внутренний буфер, помимо заполнения 128-битного буфера данных, также заполняется 16-битный регистр маскировки.

Перед началом цикла записи внутреннего буфера регистр маскировки устанавливается равным 16'hFFFF, а по мере записи данных соответствующие биты регистра маскировки обнуляются на основании состояния линий шины Wishbone wb_sel_i и wb_adr_i[3:2]. Линии wb_adr_i[3:2] адресуют 32-битные слова в пределах внутреннего буфера, а четыре линии wb_sel_i адресуют байты в пределах 32-битного слова. Если необходимо записать один байт по адресу 32'h80000000, в момент записи этого байта во внутренний буфер wb_adr_i[3:2] будет равно 2'b00, а wb_sel_i – 4'b0001. В целом механизм формирования маскирующего регистра на языке Verilog выглядит следующим образом:

```
case (wb_adr_i[3:2])
2'b00: mask_reg <= mask_reg &
{12'hFFF,~(wb_sel_i)};
2'b01: mask_reg <= mask_reg &
{8'hFF,~(wb_sel_i),4'hF};
2'b10: mask_reg <= mask_reg &
{4'hF,~(wb_sel_i),8'hFF};
2'b11: mask_reg <= mask_reg &
{~(wb_sel_i),12'hFFF};
endcase
```

Согласно [1], во время записи пакета данных младшие 3 бита адреса столбца должны содержать достоверные данные, но эти данные игнорируются. Таким образом, пакет записываемых данных всегда выравнивается по границе 16 байт. При чтении процедура более сложная и зависит от младших битов. Однако при вышеописанном алгоритме чтения данных требуется фиксированная позиция данных во внутреннем буфере, – байт, прочитанный по нулевому адресу, должен храниться в буфере также по нулевому адресу, и т.д. Поэтому при формировании адреса столбца для операций чтения и записи младшие три бита всегда заменяются нулями в модуле трансляции адреса ddr_adrc:

```
assign c_adr_o =
{adr_i[10:4],3'b000};
```

Эти младшие биты адреса учитываются только при чтении данных из внутренних буферов контроллера устройствами на шине Wishbone.

Рассмотрим более подробно работу блока управления вводом/выводом, выполняющего функции распределения тактовых сигналов и буферизации входных и выходных данных.

Блок управления вводом/выводом

Все ранее рассмотренные блоки на языке Verilog содержали универсальный код, не зависящий от типа используемой ПЛИС. Блок управления вводом/выводом содержит примитивы, специфичные для ПЛИС Xilinx Spartan 6. Рассмотрим более подробно архитектуру ввода/вывода семейства Spartan 6.

В ПЛИС Spartan 6 для высокоскоростного обмена данными можно использовать два типа примитивов: ILOGIC2/OLOGIC2 и ISERDES2/OSERDES2. Каждый из этих типов может быть встроен в исходный код на языке Verilog двумя способами: при помощи автоматизированного генератора IP-модулей (wizard) или при помощи готовых шабло-

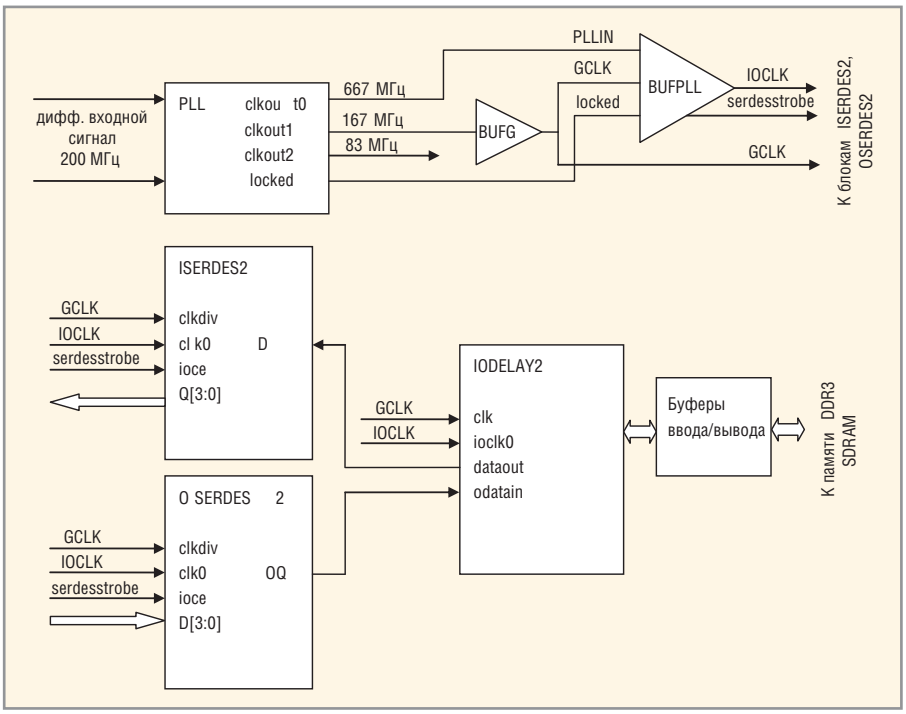

Рис. 6. Подключение блоков ISERDES2/OSERDES2

нов. В блоке управления вводом/выводом используется метод шаблонов, позволяющий более гибко регулировать внутренние соединения блоков.

Примитивы ILOGIC2/OLOGIC2 вставляются в исходный код с помощью шаблонов IDDR2/ODDR2. Эти блоки позволяют заменить обмен данными типа DDR на тип SDR. Например, блок ODDR2 можно сконфигурировать таким образом, что он будет принимать входные данные только на положительном фронте и затем автоматически разделять их на выходе на два фронта. У этого типа ввода/вывода есть несколько недостатков:

- входные данные необходимо выдавать на рабочей частоте памяти DDR3 SDRAM, в данном случае – 333,5 МГц. Это довольно близко к предельной частоте работы буфера синхросигнала (BUFG) 400 МГц;
- рабочая частота блоков IDDR2/ODDR2 также должна быть буферизована блоком BUFG, который является низкочастотным буфером по сравнению с буферами ввода/вывода.

В ПЛИС Spartan 6 существует два типа буферов синхросигнала. Блоки BUFG и BUFGMUX – это глобальные буферы с предельной рабочей частотой 400 МГц. Блоки BUFIO2 и BUFPLL относятся к типу буферов ввода/вывода с предельной рабочей частотой 1050 МГц, поэтому использование второго типа буферов предпочтительнее. Для использования этих буферов необходимы примитивы ISERDES2/OSERDES2.

Как следует из названия SERDES (SERializer/DESerializer), эти примитивы преобразуют последовательные потоки данных в параллельные для типа ISERDES2 и параллельные потоки данных в последовательные для типа OSERDES2. Параметр DATA_WIDTH задаёт ширину параллельного порта данных. При дифференциальном вводе/выводе этот параметр может принимать значения до восьми, при обычном – до четырёх. Соответственно, во столько же раз понижается частота поступления параллельных данных.

На рисунке 6 изображена упрощённая блок-схема подключения блоков ISERDES2/OSERDES2. Дифференциальный тактовый сигнал частотой 200 МГц подаётся на вход блока PLL, на выходе которого формируются три частоты – 667, 167, 83 МГц, а также вспомогательный сигнал locked (захвачен) для буфера BUFPLL. Блок PLL может одновременно генерировать до шести частот, однако только два первых выхода (clkout0 и clkout1) могут генерировать частоты выше 400 МГц, и в этом случае эти выходы должны быть подключены к буферу BUFPLL.

Блоки ISERDES2/OSERDES2 поддерживают только режим SDR при использовании BUFPLL. Поэтому для поддержания скорости обмена DDR на частоте 333,5 МГц высокая тактовая частота (IOCLK) равна удвоенной частоте памяти DDR3 SDRAM (667 МГц). Для работы блоков ISERDES2/OSERDES2 также требуется вторая частота GCLK, равная скорости подачи параллельных данных. При параметре DATA_WIDTH = 4 эта частота будет в четыре раза меньше частоты IOCLK, т.е. составит 167 МГц. Таким образом, частота подачи параллельных данных получается в два раза ниже по сравнению с использованием примитивов ILOGIC2/OLOGIC2. На выходе BUFPLL также генерируется сигнал serdesstrobe, управляющий загрузкой параллельных данных. Частота 83 МГц является основной рабочей частотой конечных автоматов контроллера памяти.

Блоки ISERDES2/OSERDES2, как правило, используются вместе с управляемыми линиями задержки IODELAY2. Использование линий задержки позволяет точно подобрать требуемые фазовые соотношения сигналов, например, отцентрировать выходные данные относительно тактового сигнала. Для ПЛИС семейства Spartan 6 входная задержка может быть динамически конфигурируемой во время работы, тогда как фиксированная выходная задержка должна быть задана на этапе компиляции проекта. В данной версии контроллера входная задержка устанавливается в процессе калибровки равной нулю (VARIABLE_FROM_ZERO) во время инициализации, а выходная задержка устанавливается равной 30 для линий данных и маски и равной 0 для всех остальных линий.

Для обеспечения управляемой задержки все интерфейсные сигналы памяти DDR3 SDRAM, включая линии команд и адреса, подключены через линии задержки и примитивы ISERDES2/OSERDES2. Дифференциальный тактовый сигнал частотой 333,5 МГц для памяти формируется при помощи блока OSERDES2, работающего на высокой частоте 667 МГц и имеющего на входе фиксированные данные 4'b0101. На выходе блока получается сигнал частотой 333,5 МГц, который после линии задержки и выходного буфера подаётся на входы памяти. Сигналы адреса и команд также проходят через блоки OSERDES2 и IODELAY2. Поскольку глобальная частота синхронизации равна 167 МГц, а команда для памяти должна быть активной только в течение одного периода частоты памяти (333,5 МГц), сигнал команды подаётся только на первые два входа блока OSERDES2. На оставшиеся два входа поступает постоянная величина 2'b11, что соответствует команде NOP:

```
.D1(cmd_i[i]),
.D2(cmd_i[i]),
.D3(1'b1),
.D4(1'b1),
```

Для всех выходных сигналов параметр .DELAY_SRC блока IODELAY2 должен быть равен ODATAIN, а для двунаправленных линий этот параметр должен быть равен IO. Двунаправленными в данной версии являются только линии данных dq_io. Стробирующие линии DQS также определены как двунаправленные в модуле sopc, но они используются только при записи данных и поэтому в блоках IODELAY2 заданы как выходные.

Для всех линий, использующих высокоимпедансное состояние (tristate), необходимо сигнал перехода в третье состояние также проводить через блоки OSERDES2 и IODELAY2.

Все линии памяти имеют стандарт ввода-вывода SSTL15_II или DIFF_SSTL15_II (для дифференциального тактового сигнала памяти), задаваемый в примитивах буферов ввода-вывода и в файле constr1.ucf. Более подробная информация об архитектуре блоков ввода/вывода, а также о распределении тактовых сигналов приведена в [4–6].

Хост-контроллер шины Wishbone

После описания работы всех блоков контроллера памяти рассмотрим вспомогательный модуль хост-контроллера шины Wishbone, также входящий в состав системы на кристалле. Исходный код этого модуля находится в файле wb_host.v. Задачей модуля является коммутация линий шины Wishbone в соответствии с запросами различных устройств. В данной версии системы на кристалле на шине Wishbone работают только два устройства – контроллеры UART и памяти DDR3 SDRAM, однако в хост-контроллере шины также определены линии для контроллера Ethernet, контроллера флэш-памяти и процессора (CPU). Приоритеты доступа следующие (в порядке убывания): контроллер Ethernet, UART, память DDR3 SDRAM, флэш-память, процессор. Для каждого устройства определён базовый адрес в файле wbhost_parms.v.

Заключение

В статье рассмотрена модель контроллера динамической памяти DDR3 SDRAM, разработанная на языке Verilog и реализованная для ПЛИС семейства Spartan 6, которая установлена на отладочной плате SP605. Альтернативой модели на языке Verilog является использование готового контроллера памяти фирмы Xilinx, вставляемого в проект в виде IP-модуля. Однако Verilog-модель служит хорошим учебным пособием, позволяющим детально изучить принципы функционирования памяти DDR3 SDRAM. Контроллер памяти был успешно синтезирован и показал удовлетворительные результаты при тестировании.

Литература

1. DDR3 SDRAM. 1Gb_DDR3_D1.fm, Micron Technology, Inc., 2006.
2. *Гребенников А.* Контроллер DDR SDRAM для платы DK-START-3C25N. Современная электроника. 2011. № 1.
3. *Гребенников А.* HDL-реализация асинхронного приёмопередатчика. Современная электроника. 2011. № 4.
4. Spartan-6 FPGA SelectIO Resources. UG381.
5. Spartan-6 FPGA Clocking Resources. UG382.
6. Sawyer N. Source-Synchronous Serialization and Deserialization (up to 1050 Mb/s). XAPP1064.