Key Design Features

- Synthesizable, technology independent VHDL IP Core
- 8-bit / 16-bit Flash memory controller with synchronous user interface
- Provides the physical interface between your FPGA / ASIC and the external Flash memory component
- JEDEC® standard Flash EEPROM pinouts and commands
- Configurable timing parameters to suit different Flash memory components
- Configurable command FIFO compensates for System-to-Flash speed differences
- Wide range of Flash memories supported
- Vendors such as Microchip®, Atmel®, AMD®, EON® and STmicroelectronics®
- Examples include the SST39*, AT49*, AM29*, EN29* and M29* series Flash memory ICs

Applications

- Any application where non-volatile storage is required
- Offline storage of parameters and data for FPGA / ASIC

Generic Parameters

Generic name	*Description*	*Type*	*Valid range*
t_as	Address setup time in system clock cycles	integer	$0 < t\_as < 2^{32}$
t_ah	Address hold time in system clock cycles	integer	$0 < t\_ah < 2^{32}$
t_wp	Write strobe pulse width in system clock cycles	integer	$0 < t\_wp < 2^{32}$
t_bp	Byte program time in system clock cycles	integer	$0 < t\_bp < 2^{32}$
t_se	Sector erase time in system clock cycles	integer	$0 < t\_se < 2^{32}$
t_ce	Chip erase time in system clock cycles	integer	$0 < t\_ce < 2^{32}$
t_aa	Address access time in system clock cycles	integer	$0 < t\_aa < 2^{32}$
dw	Flash data width	integer	8 or 16
depth	Flash command FIFO depth	integer	≥ 2
log2d	Flash command FIFO depth log2	integer	log2 (depth)

Block Diagram

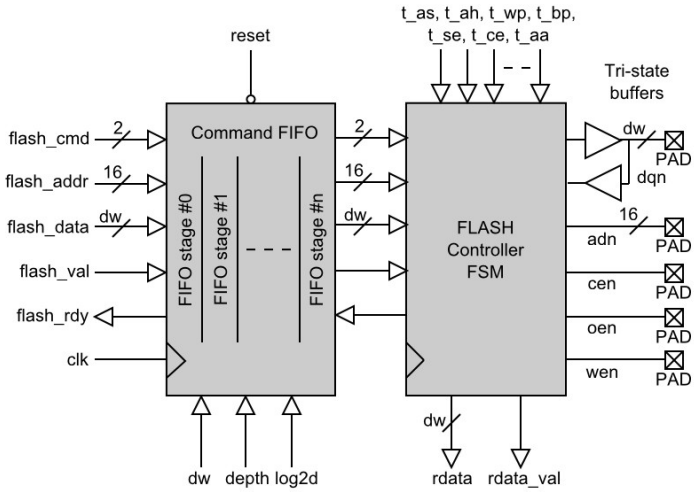

Figure 1: Flash memory controller architecture

Pin-out Description

Pin name	*I/O*	*Description*	*Active state*
clk	in	Synchronous clock	rising edge
reset	in	Asynchronous reset	low
flash_cmd[1:0]	in	Flash command	bus
flash_addr[15:0]	in	Flash address	bus
flash_data[dw-1:0]	in	Flash write data	bus
flash_val	in	Flash command valid	high
flash_rdy	out	Flash command ready (handshake signal)	high
rdata [7:0]	out	Flash read data in	bus
rdata_val	out	Flash read data valid	high
adn[15:0]	out	Flash address	bus
dqn[dw-1:0]	io	Bi-directional Flash read/write data	bus
cen	out	Chip enable	low
oen	out	Transmit data	low
wen	out	Transmit data valid	low

General Description

FLASH_CONT is a JEDEC® compliant Flash controller IP Core that provides a convenient way of interfacing your FPGA or ASIC to an external FLASH memory component. The IP Core features a simple-to use command interface and is fully synchronous with the system clock.

The Flash controller is comprised of two main blocks as described by Figure 1. These blocks are the command FIFO and the main controller state machine that generates the correct signalling to the Flash memory.

The command FIFO uses the standard Zipcores valid-ready streaming protocol. This is a synchronous interface in which commands, addresses and data are written to the FIFO on the rising-edge of *clk* when *flash_val* and *flash_rdy* are both high[1].

The command FIFO may be used to 'queue up' a sequence of commands to the Flash memory, and in doing so, can help to alleviate the speed differences between the system and the Flash device. The depth of the command FIFO may be configured with the generic parameters *depth* and *log2d*. Increasing the depth of the FIFO will allow more commands to be queued up. The 2-bit *flash_cmd* signal is decoded as follows:

00 : Write byte command
01 : Sector erase command
10 : Chip erase command
11 : Byte read command

The output ports to the Flash use a standard JEDEC pinout for an 8 or 16-bit EEPROM device. These pins may be connected directly to the external Flash memory component. The controller is compatible with most standard COTS devices such as those from Microchip®, Atmel®, AMD®, Winbond®, EON® and STmicroelectronics®.

Functional Timing

Figure 2 shows the user interface with Flash controller IP Core. The *flash_cmd*, *flash_addr* and *flash_data* signals are sampled on the rising edge of the system *clk* when *flash_val* and *flash_rdy* are both high. If the *flash_rdy* signal is low then this indicates that the command FIFO is full and no further commands should be sent until the FIFO has emptied.

Read data from the Flash controller appears on the *rdata* port with *rdata_val* signifying valid data. Read data is also synchronous with the system clock.

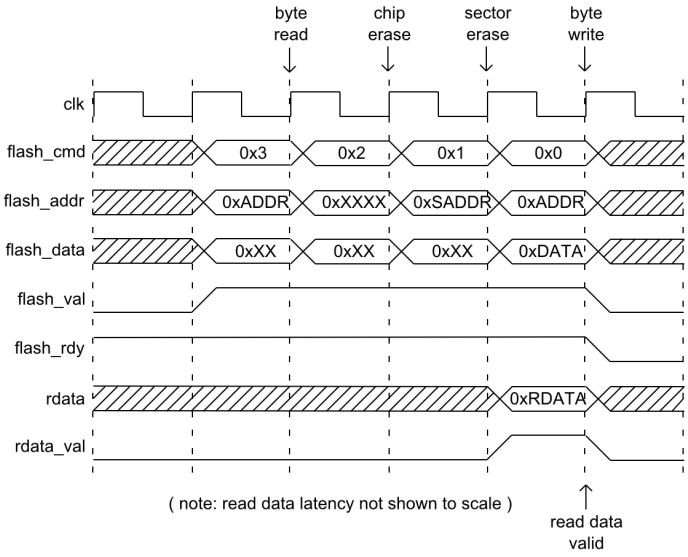

Figure 2: Flash controller user-interface timing

Figures 3, 4, 5 and 6 on the following pages show the timing waveforms with the physical Flash component. These waveforms and pinouts follow the standard JEDEC specification for Flash EEPROM devices. All timing parameters shown can be configured using the generic IP Core settings.

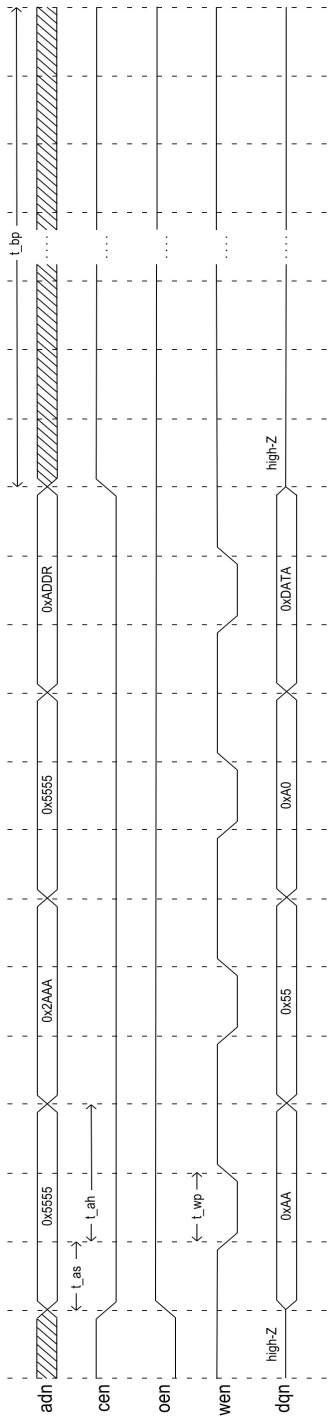

Figure 3: Write byte timing sequence

1 See Zipcores application note: app_note_zc001.pdf for more examples of the valid-ready streaming protocol.

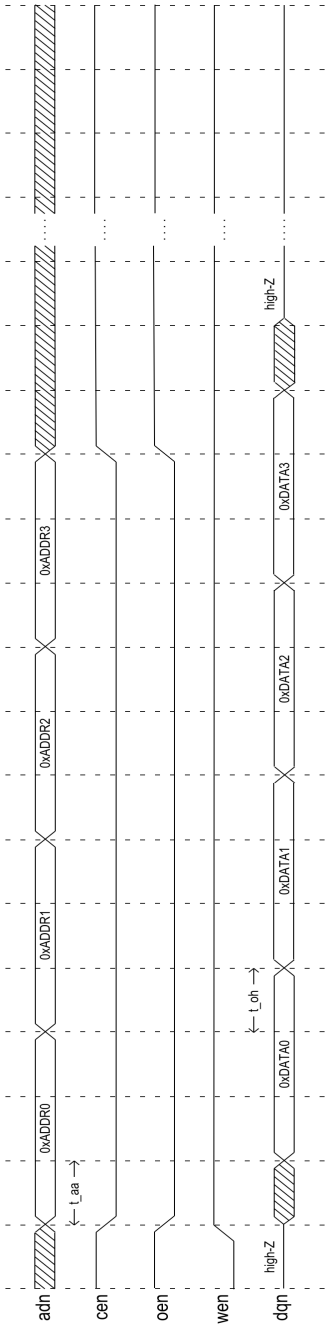

Figure 4: Read byte timing sequence

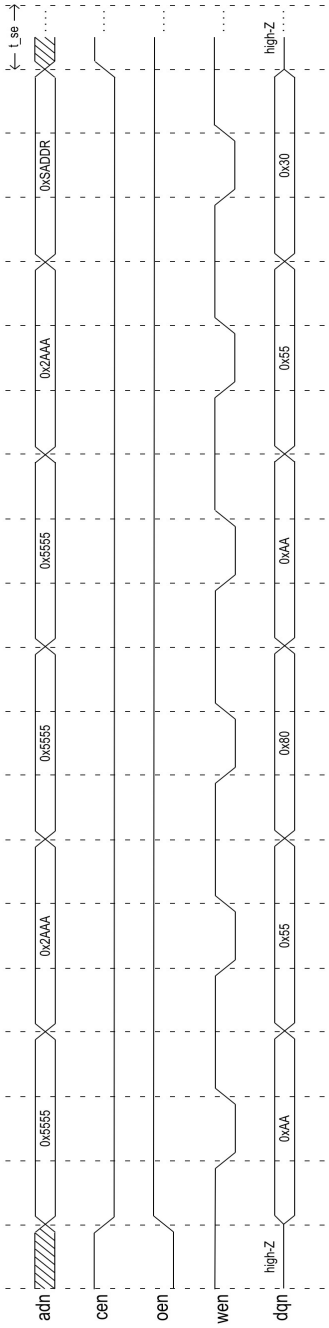

Figure 5: Sector erase timing sequence

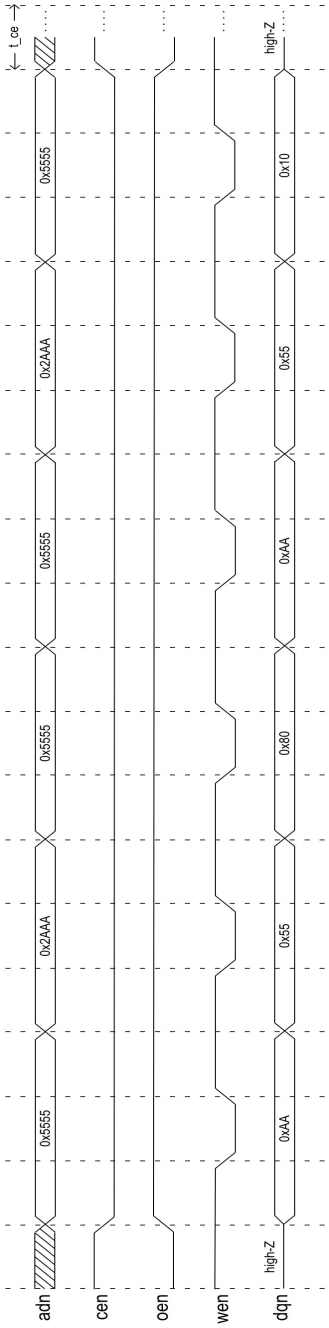

Figure 6: Chip erase timing sequence

Source File Description

All source files are provided as text files coded in VHDL. The following table gives a brief description of each file.

Source file	***Description***
flash_io_buffer.vhd	Bi-directional tristate buffer
flash_io_buffer_n.vhd	N-deep bi-directional tristate buffer
flash_fifo.vhd	Flash command FIFO
flash_fsm8.vhd	8-bit controller state machine
flash_fsm16.vhd	16-bit controller state machine
flash_cont.vhd	Top-level component
flash_cont_bench.vhd	Top-level test bench

Functional Testing

An example VHDL test bench is provided for use in a suitable VHDL simulator. The compilation order of the source code is as follows:

1. flash_io_buffer.vhd
2. flash_io_buffer_n.vhd
3. flash_fifo.vhd
4. flash_fsm8.vhd
5. flash_fsm16.vhd
6. flash_cont.vhd
7. flash_cont_bench.vhd

The VHDL test bench instantiates the Flash controller component in an 8-bit configuration and drives user interface with a basic sequence of commands. These commands are: byte write, sector erase, byte write, chip erase, byte read, byte read. The output waveforms may be viewed in the simulator to observe the behaviour.

In the default set up, the simulation must be run for around 5 ms for the test to complete. The simulation generates a pair of output text files called: *flash_in.txt* and *flash_out.txt* These files contain the input commands and output read data from the Flash controller IP Core.

Synthesis

The files required for synthesis and the design hierarchy is shown below:

- flash_cont.vhd
  - flash_fifo.vhd
  - flash_fsm8.vhd
    - flash_io_buffer_n.vhd
      - flash_io_buffer.vhd
  - flash_fsm16.vhd
    - flash_io_buffer_n.vhd
      - flash_io_buffer.vhd

The VHDL core is designed to be technology independent. However, as a benchmark, synthesis results have been provided for the Xilinx® Virtex 6 and Spartan 6 FPGA devices. Synthesis results for other FPGAs and technologies can be provided on request.

Note that in order to achieve the most area efficient designs the size of the FIFOs should be kept to a minimum.

Trial synthesis results are shown with the generic parameters set to: t_as = 10, t_ah = 20, t_wp = 10, t_bp = 2000, t_se = 2500000, t_ce = 1000000, dw = 8, depth = 15, log2d = 4.

Resource usage and timing is specified after Place and Route.

VIRTEX 6

Resource type	***Quantity used***
Slice register	75
Slice LUT	150
Block RAM	0
DSP48	0
Occupied slices	62
Clock frequency (approx)	350 MHz

SPARTAN 6

Resource type	***Quantity used***
Slice register	80
Slice LUT	148
Block RAM	0
DSP48	0
Occupied slices	54
Clock frequency (approx)	200 MHz

Revision History

Revision	***Change description***	***Date***
1.0	Initial revision	09/07/2014
1.1	Increased internal counter widths to 32-bits wide. Added testbench input file and output file capture. Included support for 16-bit memories.	13/02/2015

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Zipcores:
SKU91