USER MANUAL FOR ATMEL DATAFLASH® VHDL MODEL

Important :

The VHDL implementation has two setups(2 folders).

1.flash_vhd_w_hold -- Setup which has the HOLD feature implemented and tested.

2.flash_vhd_wo_hold – Setup which does not posses the HOLD signal at the port level.

Use of DataFlash Model in Synopsys VCS and ModelSim :

The setups(both folders) have the following files in them

a. AT26DFXXX.vhd --> VHDL DataFlash model.

b. tb_dataflash.vhd --> Test Bench.

c. flash.csh --> Run script for Synopsys VCS tool

d. flash_modsim.do --> Command lines used when running with ModelSim.

e. run.sim --> Include file which takes care of dump and simulation

f. synopsys_sim.setup --> Required when using VCS. Setup for 'work' directory and 'Timescale' specification are done here.

Executing the script :

The entity name for AT26DFXXX.vhd (DUT – Flash Model) is 'dataflash26x' and the entity name for test bench file tb_dataflash.vhd is 'dataflashtestbench'. The instantiation is done inside the testbench with portmap handle(Instance name) as 'flash'.

Synopsys VCS :

The script 'flash.csh' takes care of compilation and simulation as well. Executing the script at the console starts the process.

ModelSim :

The script flash_modsim.do is for ModelSim. At the Modelsim> prompt one needs to enter

'do flash_modsim.do' .

Signal Description :

This briefly describes the signals used by the model AT26DFXXX.vhd

Signal	***Description***	***Default Value***	***Type***
CSB	Chip Select	'1'. From Testbench	Input. Active low
SCK	System Clock	'1'. from Testbench	Input
SI	Serial In	'X' from testbench	Input
WPB	Write protect	'1' from testbench	Input. Active low
SO	Serial Out		Output
HOLDB(for flash_vhd_w_hold setup only)	Hold Pin	'1' from testbench	Input. Active low.

Generics and their descriptions

Generic	Purpose
DEVICE	The component name that is passed as generic
PRELOAD	Signal which determines if the memory in the model is to be preloaded with values from a text file or not. If '1' the memory is preloaded
MEMORY_FILE	The text file that contains the 8 bit Hex values that is to be preloaded inside the memory

The ***PRELOAD*** signal is a feature that has been implemented to preloaded the memory with values from a text file passed via a generic string ***MEMORY_FILE*** whenever required.

Apart from these there are other signals which have been used as switches to select between different opcodes.

TRIGGERING SWITCHES FOR VARIOUS TASKS

rd_array	When '1' triggers Read Array Hi Freq task
rd_array_low	When '1' triggers Read Array Lo Freq task
Blke4	When '1' triggers 4KB block erase
Blke32	When '1' triggers 32K block erase
Blke64	When '1' triggers 64K block erase
byte_prog	When '1' triggers Byte prog task
seq_byte_prog	When '1' triggers Sequential byte prog task
wr_en	When '1' triggers Write Enable task
wr_dis	When '1' triggers Write Disable task
prot_sec	When '1' triggers Protect Sector task
unprot_sec	When '1' triggers Unprotect sector task
rd_sec_prot_reg	When '1' triggers Read Sector protection reg task
rd_sts_reg	When '1' triggers Read Status register task
wr_sts_reg	When '1' triggers Write status register task
rd_manf_id	When '1' triggers Read Manufacturer ID task
EDPD	When '1' triggers Enter Deep power down mode task
RDPD	When '1' triggers Resume from deep power down task

Signals used in **STATUS REGISTER**

SPRL	Sector Protection register locked
SPM	Sequential Program mode
EPE	Erase/Program error
WPP	Write protect pin status
SWP	Software protection status
WEL	Write enable latch status
RDYBSY	Ready/busy status

Apart from these there are other internal signals. Some of those with prime importance are

addr_reg --> Used to store 24 bit address . Used in 'getaddr' procedure

opcode_temp --> Gets opcode sent via SI

so_en --> Enable signal for SO bus. If it is '1' then SO gets the value that is present in 'so_reg', else if so_en is '0', SO gets 'Z'.

so_reg --> Signal which puts data onto SO.

Lock,Unclock,

global_protect,

global_unprotect --> SPRL

Protect, Unprotect --> Signals which act as switches in starting Protect/Unprotect tasks.

Protected_sig --> Tells if a particular sector is protected or not.

List of parameters:

Parameter	*Purpose*
fRDLF	SCK Frequency for read Array (Low freq - 03h opcode)
fSCK	Serial clock (SCK) Frequency in MHz
tSCKH	SCK High time
tSCKL	SCK Low time
tCSH	Chip Select high time
tCSLS	Chip Select Low Setup time
tCSLH	Chip Select Low hold time
tCSHS	Chip Select high Setup time
tCSHH	Chip Select high hold time
tDS	Data in Setup time
tDH	Data in Hold time
tHLS	HOLD! Low Setup Time
tHHS	HOLD! High Setup Time
tHLH	HOLD! Low Hold Time
tHHH	HOLD! High Hold Time
tWPS	Write Protect Setup Time (only when SPRL=1)
tWPH	Write Protect Hold Time (only when SPRL=1)
tWRSR	Write Status Register Time
tSECP	Sector Protect Time
tSECUP	Sector Unprotect Time
tEDPD	Chip Select high to Deep Power-down (3 us)
tRDPD	Chip Select high to Stand-by Mode
tPP	Page Program Time
tBP	Byte Program Time
tBLKE4	Block Erase Time 4-kB (0.350 sec)
tBLKE32	Block Erase Time 32-kB
tBLKE64	Block Erase Time 64-kB
tCHPE	chip erase time

Functions used in the DataFlash model(AT26DFxxx.vhd)

Function	*Purpose*	*Return Type*
Blocks	Returns no of 64K Bocks for	Integer

Function	*Purpose*	*Return Type*
	each device	
Sectors	Returns no of sectors for each device	Integer
Memsize	Returns memory size	Integer

Procedures Used in the DataFlash model(AT26DFxxx.vhd)

Procedure	*Purpose*
glbl	Used for Global Protect/Unprotect
getdata	Used for obtaining data during Sequential Byte Programming
data_wr_sts	Used for obtaining data for Write Status Reg command
Read_array	Read Array procedure(Both Hi and Lo freq)
Read_status	Read Status Register Task
Byte_prg	Byte Programming procedure
Protect_sec	Protect Sector Procedure
Unprotect_sec	Unprotect sector procedure
read_sec_prot	Read Sector Protection registers task
erase_4k	4KB block erase task
erase_32k	32KB block erase
Erase_64k	64KB block erase task
chip_erase	Chip Erase task
getaddr_hifreq	Used for read array Hi freq only. Gets24 bit address and 8 don't care bits
getaddr	Gets 24 bit address.

Procedures used in the testbench(tb_dataflash.vhd) :

Procedure	*Purpose*
Opcode	Sends in the opcode on SI
Addr	Send in the 24 bit address on SI
Data	Sends in the 8 bit data on SI
data_x	Used only for read array Hi Freq. Sends in the 8 don't cares.

Error Display statements :

The following explains the display messages that get displayed when each procedure or their corresponding triggering switches get triggered.

Frequency check:

This is done in all procedure calls. The maximum frequency is 70 Mhz. If fSCK is more than 70 Mhz a message gets displayed saying

" WARNING : Frequency should be less than 70 Mhz "

EDPD :

On the EDPD getting triggered it comes inside the EDPD task. If the RDYBSY_reg is not deasserted by then, a message gets displayed saying

" Device is busy. Cannot enter into Deep-Power down "

RDPD :

On the RDPD getting triggered it comes inside the RDPD task. If the RDYBSY_reg is not deasserted by then, a message gets displayed saying

" Device is busy. Cannot enter into Deep-Power down "

rd_array :

On entering the read array procedure call if the RDYBSY_reg is not asserted it continues with the procedure executions, else it reports

" Device is busy. READ ARRAY Hi Freq is not allowed "

rd_array_low :

On entering the read array procedure call if the RDYBSY_reg is not asserted it continues on with the procedure execution else it reports

Device is busy. READ ARRAY Lo Freq is not allowed

byte_prog :

On entering the byte programming task, it enters the page programming functionality. If the programming stops at an index other than '0' the statement gets displayed saying

" Page Programming stopped at wrong index value. CSB deasserted"

Else if RDYBSY_reg is '1', then the report gets displayed as

" PAGE/BYTE PROG -- DEVICE BUSY "

seq_byte_prog :

Before entering sequential byte programming the RDYBSY_reg is checked. If it is '1' then the display is

" SEQ BYTE PROG -- DEVICE BUSY "

If RDYBSY_reg is '0' and enters sequential byte programming but fails at sector protection then the statement is

" SEQ BYTE PROGRAMING CANNOT BE EXECUTED. "

rd_manf_id:

Before entering Read manufacture ID if RDYBSY_reg is '1' then the report statement gets displayed

" READ MANUF ID -- DEVICE BUSY "

Prot_sec:

On entering protect sector is RDYBSY_reg is '1' then

" PROTECT SECTOR -- DEVICE BUSY "

unprot_sec:

On entering protect sector is RDYBSY_reg is '1' then

"UROTECT SECTOR -- DEVICE BUSY "

wr_sts_reg:

On checking SPRL, if SPRL is '1' then the message that gets displayed is as follows

" SPRL LOCKED. GLOBAL PROTECT/UNPROTECT CANNOT BE PERFORMED "

If WEL bit is set then

" WEL bit not set. Write Status Register is not allowed "

Else if RDYBSY_reg is '1' then

" Device is busy. WRITE STATUS REG cannot be performed "

blke4, blke32, blke64 :

On checking sector protection if any of the sector is found protected then the following get

displayed

" Sector is protected. 4K Block Erase couldn't be performed "

" Sector is protected. 32K Block Erase couldn't be performed "

" Sector is protected. 64K Block Erase couldn't be performed "

chip_ers:

If WEL and SWP are not set then the statement gets displayed

" WEL bit is not set or/and SWP is not '00'. CHIP ERASE CANNOT BE PERFORMED "

Status and Error printing of AT26DFxxx.vhd module:

Based on the opcode received, any of the following will appear:
Opcode(ab) for Resume from Deep Power-down received
if the device is in Deep Power-down, and if any other opcode except "ab" appears, then the following print will appear:
Opcode XX is not allowed: device in Deep Power-down
Opcode(0b) for Read Array received
Opcode(03) for Read Array (Low Freq) received
Opcode(20) for 4 KB Block erase received
Opcode(52) for 32 KB Block erase received
Opcode(d8) for 64 KB Block erase received
Opcode(c7) for Chip erase received
Opcode(02) Byte Program received
Opcode(af) Sequential Byte Program received
if device selected is AT26DF321, then this printing will appear
Sequential Byte Program is not supported for device AT26DF321
Opcode(06) for Write Enable received
Opcode(04) for Write Disable received
Opcode(36) for Protect Sector received
Opcode(39) for Unprotect Sector received
Opcode(3c) for Read Sector Protection Register received
Opcode(05) for Status Register Read received
Opcode(01) for Write Status Register received
Opcode(9f) for Read Manufacturer and Device ID received
Opcode(b9) for Deep Power-down received
if none of the above opcodes received, then the following print will appear:
Unrecognized opcode XX

Inside Tasks:

When calling protect sector, following printing will appear:

Entered Protect Sector task

When entering unprotect sector, anyone of the following printing will appear:
Entered Unprotect sector for Device: AT25DF041A
Entered Unprotect sector for Device: AT26DF081A
Entered Unprotect sector for Device: AT26DF161A
Entered Unprotect sector for Device: AT26DF321
Unprotect Sector failed

if the check_protection process fails, following printing will appear:
Check Sector Protection failed

In erase_4kb, erase_32kb, erase_64kb and erase_chip following printing will appear respectively:
4 KB block is to be erased. Will wait for tBLKE4 time units
32 KB block is to be erased. Will wait for tBLKE32 time units
64 KB block is to be erased . Will wait for tBLKE64 time units
CHIP ERASE is going to start. Will wait for tCHPE time units

Entering into Deep Power-down execution, following printing will appear:
If device is busy(RDYnBSY==1):
“Device is busy. Cannot enter into Deep-Power down”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX Mhz ”
At the end of operation:
“Device enters into Deep Power-down mode. Send 'Resume from Deep Power-down' to resume”

Entering into Resume from Deep Power-down execution, following printing will appear:
If device is busy(RDYnBSY==1):
“Device is busy. Cannot resume from Deep-Power down”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX Mhz.”
At the end of operation:
“Device Resumes from Deep Power-down mode”

Entering into Manufacturing ID Read execution, following printing will appear:
If device is busy(RDYnBSY==1):
“READ MANUF ID -- DEVICE BUSY”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX Mhz ”
At the start of operation:
“ENTERED READ MANUFACTURER ID FUNCTION”
At the end of operation:
“EXITED READ MANUFACTURER ID FUNCTION”

Entering into Status Register Read execution, following printing will appear:
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX Mhz for SPI interface.”
“ENTERED READING STATUS REGISTER FUNCTION”

Entering into Status Register Write execution, following printing will appear:
If device is busy(RDYnBSY==1):
“Device is busy. WRITE STATUS REG cannot be performed”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz”
“Entered write status register task”
Any of the following print will appear based on the write status register operation.
“SPRL LOCKED. GLOBAL PROTECT/UNPROTECT CANNOT BE PERFORMED”
if SPRL bit is not set prior to write status register operation, following print will appear:
“SPRL LOCKED. GLOBAL PROTECT/UNPROTECT CANNOT BE PERFORMED”
if WEL bit is not set prior to write status register operation, following print will appear:
“WEL bit not set. Write Status Register is not allowed”
“Write Status Register operation completed”

Entering into Write Enable execution, following printing will appear:
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz for SPI interface. Write Enable is not allowed.”
“Write Enable Latch Set”

Entering into Write Disable execution, following printing will appear:
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than fSCK for SPI interface. Write Disable is not allowed.”
“Write Enable Latch Reset”

Entering into Read Array execution, following message will appear:
If device is busy(RDYnBSY==1), either of these two will appear on the screen :
“Device is busy. READ ARRAY Hi Freq is not allowed”
" Device is busy. READ ARRAY Lo Freq is not allowed "
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz”

Entering into Protect Sector execution, following message will appear:
If device is busy(RDYnBSY==1):
“PROTECT SECTOR -- DEVICE BUSY”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz ”

Entering into Unprotect Sector execution, following message will appear:

If device is busy(RDYnBSY==1):
“UNPROTECT SECTOR -- DEVICE BUSY”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz”
UNPROTECT SECTOR EXITED

Entering into Read Sector Protection Register execution, following message will appear:
If device is busy(RDYnBSY==1):
“Device is busy. Read Sector Protection Register is not allowed”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz”
READ SECTOR PROTECTION REGISTERS

Entering into Byte Program execution, following message will appear:
If device is busy(RDYnBSY==1):
“PAGE/BYTE PROG -- DEVICE BUSY”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz ”
if CSB deasserted in wrong point of time, following print will appear and operation will get aborted:
“Page Programming stopped at wrong index value. CSB deasserted”
If WEL bit is not set or sector protected:
“BYTE/PAGE Not possible. SECTOR IS PROTECTED”
During **Byte Program execution following message will appear:**
“ENTERED BYTE PAGE PROGRAMING FUNCTION”
“PAGE PROGRAMING BEGINS”
“PAGE PROGRAMING DONE”

Entering into Sequential Byte Program execution, following message will appear:
If device is busy(RDYnBSY==1):
“SEQ BYTE PROG -- DEVICE BUSY”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz”
if the sequential operation reaches end of memory:
“Sequential Byte Program for device reaches end of memory. No Wrapping allowed. Sequential Byte Program is not allowed”
“ENTERED SEQ BYTE PROGRAM FUNCTION”
If WEL bit is not set or sector protected:
“”SEQ BYTE PROGRAMING CANNOT BE EXECUTED

Entering into 4KB Block Erase execution, following message will appear:
If device is busy(RDYnBSY==1):
“Device is busy. 4KB Block Erase is not allowed”
if frequency exceeds specified fSCK:
“WARNING: Frequency should be less than XX MHz ”

If WEL bit is not set:

"WEL bit not set. 4KB Block Erase is not allowed"

Based on sector protection condition, following print will appear:

"Sector is protected. 4K Block Erase couldn't be performed"

"4 KB block erase completed"

Entering into 32KB Block Erase execution, following message will appear:

If device is busy(RDYnBSY==1):

"Device is busy. 32KB Block Erase is not allowed"

if frequency exceeds specified fSCK:

"WARNING: Frequency should be less than XX MHz"

If WEL bit is not set:

"WEL bit not set. 32KB Block Erase is not allowed"

Based on sector protection condition, following print will appear:

"Sector is protected. 32K Block Erase couldn't be performed"

Entering into 64KB Block Erase execution, following message will appear:

If device is busy(RDYnBSY==1):

"Device is busy. 64KB Block Erase is not allowed"

if frequency exceeds specified fSCK:

WARNING: Frequency should be less than XX MHz "

if WEL bit is not set:

"WEL bit not set. 64KB Block Erase is not allowed"

Based on sector protection condition any one of the following print will appear:

""Sector is protected. 64K Block Erase couldn't be performed

64kB Block with start address XXXXXX erased"

Entering into Chip Erase execution, following message will appear:

If device is busy(RDYnBSY==1):

"Device is busy. Chip Erase is not allowed"

if frequency exceeds specified fSCK:

"WARNING: Frequency should be less than XX MHz"

if WEL bit is not set or sectors protected:

"WEL bit is not set or/and SWP is not '00'. CHIP ERASE CANNOT BE PERFORMED"