INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

Implementation Of Single Data Rate Module In ONFI

Chethan L
Dayananda Sagar College of Engineering, Bangalore, india

Rajeswari P
Dayananda Sagar College of Engineering, Bangalore, India

Praveen Y S
WHIZCHIP Design Technologies, Bangalore, India

Abstract:

Huge amount of Data is generated by Different Modules by different Processes ,as a requirement of operation data or information has to sent from one module to another module which are usually dissimilar ,this can be achieved with a help of interface between two modules .

The Interface plays an important role in transferring a data between the host (Memory controller) and the device (Memory Unit). Different Devices have Different Interfaces and for NAND Flash ONFI is standard interface which helps in transferring the data between the host (Controller) and the device (Card) in a most accurate way and at higher speed than compared to any other interfaces. ONFI provides three most efficient way of data transferring methods i.e., Single Data rate (SDR),NV-Double Data Rate (NV-DDR), NV-Double Data Rate -2 (NV-DDR2).

Key words: *NAND FLASH, ONFI, SDR, STATE MACHINES*

1.Introduction

ONFI is a standard set of specification for those who design NAND FLASH devices. An interface will be required for communication between the host (controller) and the device (NAND Flash). This ONFI specification is used for interface design between any Host and a NAND memory.

As we know many companies manufactures NAND Flash device. so when a third party want to make use of the NAND device designed by other manufacturer , it might support with their application or not is not known because they don't know the way the NAND Flash device is designed.

So a group of companies set a unique standard which will be followed by all other companies when designing NAND Flash Device. This is called as OPEN NAND FLASH INTERFACE Group (ONFI), This ONFI will support all the NAND Flash devices designed by any company. This ONFI is a standard that we must follow for designing a NAND Flash memory device. A set of rules in other words.

ONFI is a standard specification used in designing NAND FLASH memory devices. There is a memory called as NAND Flash, Which stores data. To store data as in any device we send signals such as CLK and others to load the Data onto the I/O bus. In this, there are methods to load data into memory mainly 3 types SDR (Single data rate), NV-DDR (Nonvolatile-Double data rate), NV-DDR2 (Nonvolatile-Double data rate 2).For these 3 methods we have different signals based on the methods we use different signals to communicate. Mainly we will observe the timing difference and the data rate by which we can load data into memory. SDR being the slowest and NV_DDR2 is the fastest.

In the earliest ONFI versions the SDR was called by Asynchronous, NV-DDR by Source Synchronous and NV-DDR2 was not introduced, where as in ONFI 3.0 version NV-DDR2 data loading method is introduced and is the fastest among the three methods.

SDR data interface is the Asynchronous single data rate NAND interface with minimum data transferring speed and does not include a clock, data transfer is achieved with toggling Write Enable pin (WE_n) and Data Read is achieved by toggling Read Enable pin (RE_n).

2.Memory Organization

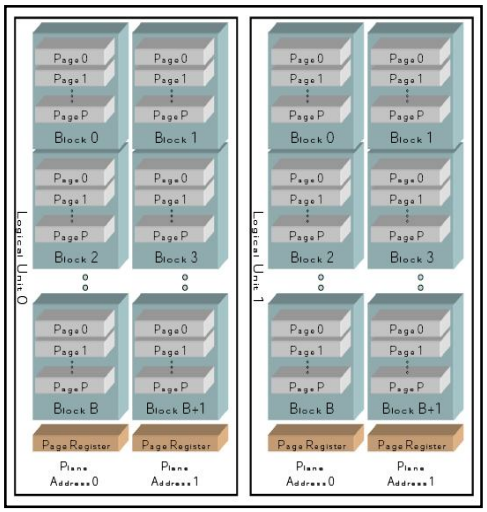

Figure 1: Target Memory Organization

The figure describes the Memory Organization in an NAND Flash device. A device contains one or more targets and it is controlled by one CE_n(chip enable) signal. A target is organized into one or more logical units (LUNs) ie.,Logical unit 0 and Logical unit 1.

A logical unit (LUN) is the minimum unit that can independently execute commands and report status. A logical unit contains minimum of one plane and maximum of 16 planes depending on the operation required and a Plane contains one Page register and a Flash array as shown in the figure. The number of page registers is dependent on the number of

multi-plane operations supported for that LUN. The Flash array contains a number of blocks.

Each LUN shall have at least one page register. A page register is used for the temporary storage of data before it is moved to a page within the Flash array or after it is moved from a page within the Flash array. If EZ NAND is supported then a buffer exists in the EZ NAND controller that provides for temporary storage of data that may then be transferred to or from the page register within each LUN.

For example: It is possible to start a Page Program Operation on LUN0 and then prior to the operations completion to start a Read command on LUN1.

A block is the smallest erasable unit of data within the Flash array of a LUN. There is no restriction on the number of blocks within the LUN ie., It can have any number of blocks depending on the memory size. A block contains a number of pages.

A page is the smallest addressable unit for read and program operations. A page consists of a number of bytes or words. If 256bytes per page is the capacity of holding a data in a single page then the capacity of data area is of 0-253Bytes and remaining two Bytes 254-255Bytes is reserved for spare area to hold the error bits.

3.Addressing

There are two types used: the Column addressing and the row addressing.

4.The Column Addressing

The Column addressing is used to access bytes or words within a page, i.e. the column address is the byte/word offset into the page. The least significant bit of the column address shall always be zero in the NV-DDR and NV-DDR2 data interfaces, i.e. an even number of bytes is always transferred. The Column addressing is of 2Bytes of length.

5.The Row Addressing

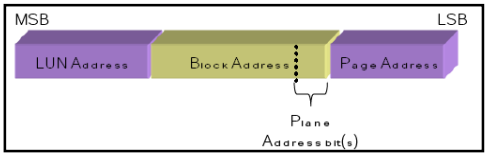

Figure 2: Row address Layout

The row address structure is shown in Figure 2 with the least significant row address bit to the right and the most significant row address bit to the left.

The Row addressing is used to address pages, blocks, and LUNs. The Row addressing is of 3 Bytes length ie., the page address always uses the least significant bits. The block address uses the middle row address bits and the LUN address uses the most significant bit(s).

When both the column and row addresses are required to be issued, the column address is always issued first in one or more 8-bit address cycles. The row addresses follow in one or more 8-bit address cycles. There are some functions that may require only row addresses, like Block Erase. In this case the column addresses are not issued.

If there are bits in the most significant bits of the column addresses and row addresses that are not used then they are required to be cleared to zero.

3.Model Architecture For SDR Interface

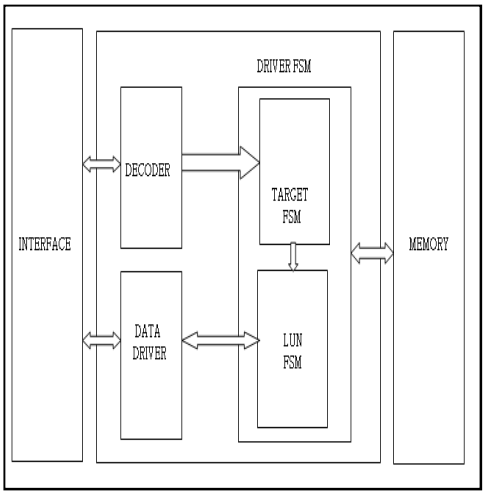

Figure 3: Model architecture for SDR Interface

This consists of Decoder, Driver FSM, Target FSM, LUN FSM, Memory and Interface.

- DECODER: It decodes the signal which comes from the interface on the IO bus and depending on the Command and Address cycles it handles the Target FSM and the LUN FSM.
- Driver FSM: It consists of Target FSM and LUN FSM.
- Target FSM: The Target FSM controls the flow of operation for Target level Commands and Target FSM task will be called from the decoder after Address decoding and based on the Command corresponding state will be selected.
- LUN FSM: The LUN FSM controls the flow of operation for LUN level Commands and LUN FSM task will be called from the decoder if the Command interpreted is LUN level Commands and based on the Command corresponding state will be selected.
- SDR INTERFACE: It contains an most efficient way of data transferring method i.e. Single Data rate (SDR).

4.Implementation

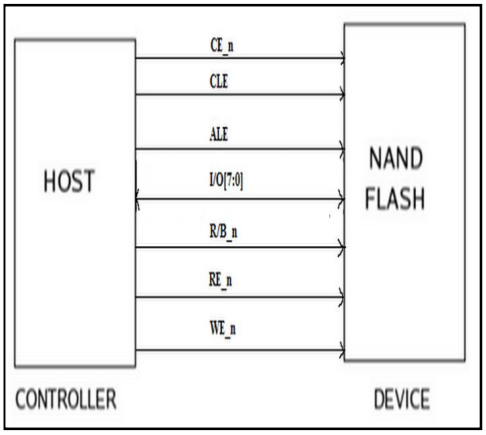

Figure 4: Design architecture of SDR INTERFACE

This consists of Host Controller, NAND Flash Device and SDR contains signals namely CE_n, CLE, ALE, I/O[7:0], R/B_n, RE_n and WE_n.

- Host Controller: It controls the NAND Flash device by sending the suitable signals depending on the operation required.
- NAND Flash Device: It decodes the signals which sent by the Host Controller and depending on the signals required, it performs the suitable operation.
- Address Latch Enable (ALE): The Address Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, and data).
- Chip Enable (CE_n): The Chip Enable signal selects the target. When Chip Enable is high and the target is in the ready state, the target goes into a low-power standby state. When Chip Enable is low, the target is selected.
- Command Latch Enable (CLE): The Command Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, and data).

- I/O Port (I/O [7:0]): The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device.
- Read Enable (RB_n): The Read Enable (True) signal enables serial data output.
- Write Enable (WE_n): The Write Enable signal controls the latching of commands, addresses, and input data in the SDR data interface.
- Write Protect (WP_n): The Write Protect signal disables Flash array program and erase operations.
- Ready/Busy (RB_n): The Ready/Busy signal indicates the target status. When low, the signal indicates that one or more LUN operations are in progress.

5.State Machines

- TARGET FSM
- LUN FSM
  - READ
  - PROGRAM
  - ERASE

5.1.Target FSM

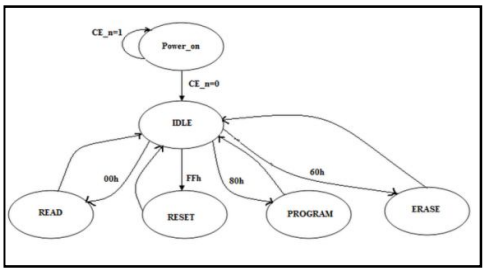

Figure 5: Target FSM

- Power Up the device and enable the CE_n(Chip enable) pin by giving logic zero ie., CE_n=0.
- After PowerOn, Check the RB_n signal Value which should be 1 and if not wait until RB_n value is 1.
- Issue a Reset Command (FFh) and by default the device should be in SDR and Timingmode 0 (TM0) and check RB_n Value which should be 1, wait until RB_n value is 1 after reset is issued.
- Issue Read Parameter Page (RPP) command (ECh).
- 4a. Issue Change Read Column command (05h) and Address, to receive first set of Parameter Page.
- Read Parameter page data and if it valid, perform the further operations ie., Read operation.

5.2.LUN FSM

5.2.1.Read Operation

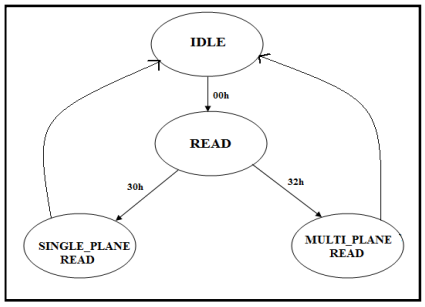

Figure 6: State Machine for Read Operation

After all the Power up operations, check the RB_n value . If RB_n is 1,then the device should be in ready state. Then depending on the operation required corresponding Opcodes is sent on the I/O bus. In this, for simple read 00h opcode value is sent in the command1 on the I/O bus and depending on the operation required ie., Single read(30h) or Multiplane read(32h) respective opcodes is going to send on the second command. After the end of the operation the device goes to Idle state only this will happen when no other operation is sent on the I/O bus immediately.

5.2.2.Program Operation

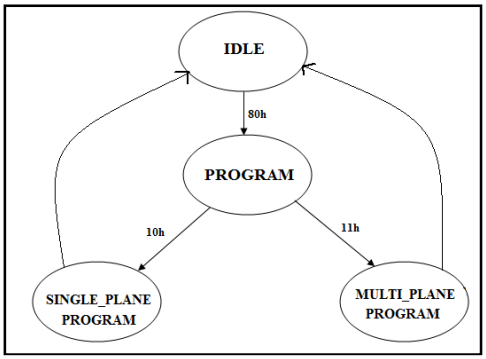

Figure 7: State Machine for Program Operation

After all the Power up operations, check the RB_n value . If RB_n is 1,then the device should be in ready state. Then depending on the operation required corresponding Opcodes is sent on the I/O bus. In this, for simple program 80h opcode value is sent in the command1 on the I/O bus and depending on the operation required ie., Single program(10h) or Multiplane program(11h) respective opcodes is going to send on the second command. After the end of the operation the device goes to Idle state, only this will happen when no other operation is sent on the I/O bus immediately.

5.2.3.Erase Operation

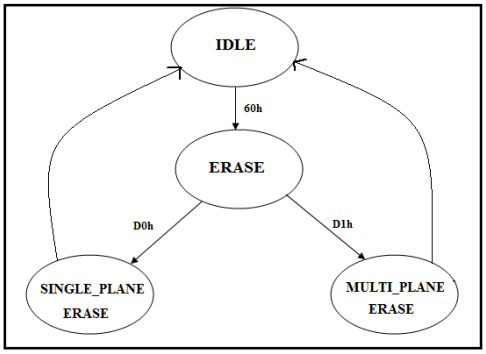

Figure 8: State Machine For Erase Operation

After all the Power up operations, check the RB_n value. If RB_n is 1, then the device should be in ready state. Then depending on the operation required corresponding Opcodes is sent on the I/O bus. In this, for simple erase 60h opcode value is sent in the command1 on the I/O bus and depending on the operation required ie., Single erase(D0h) or Multiplane erase(D1h) respective opcodes is going to send on the second command. After the end of the operation the device goes to Idle state, only this will happen when no other operation is sent on the I/O bus immediately.

5.Results

Main result of this paper gives us the transfer rates of 200 MB/s between the host (Memory controller) and the devices (Memory Unit).

6.Reference

1. An Open NAND Flash Interface Specification revision 3.1 :By ONFI Workgroup website at www.ONFI.org
2. Myoungsoo Jung, Ellis Herbert Wilson, David Donofrio, John Shalf Mahmut Taylan Kandemir "NANDFlashSim: Intrinsic Latency Variation Aware NAND Flash Memory System Modeling and Simulation at Microarchitecture Level" 978-1-4673-1747-4/12/$31.00 c 2012 IEEE
3. Seung-Ho Lim and Kyu-Ho Park "An Efficient NAND Flash File System for Flash Memory Storage" IEEE TRANSACTIONS ON COMPUTERS, NO. 7, JULY 2006, VOL. 55.
4. Yu-Hsiang Kao and Juinn-Dar Huang" High-Performance NAND Flash Controller Exploiting Parallel Out-of-Order Command Execution" 978-1-4244-5271-2/10/$26.00 ©2010 IEEE..
5. Flash Design Team" A 64Gb NAND Flash Memory with 800MB/s Synchronous DDR Interface" 978-1-4673-1081-9/12/$31.00 ©2012 IEEE.