A
DISSERTATION
ON

Designing a NAND Flash Interface I/O which meets all the specifications of ONFI 2.3

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING
(ELECRONICS AND COMMUNICATION)

SUBMITTED BY:

PANKAJ SINGH

COLLEGE ROLL NO: 10/E&C/08
UNIVERSITY ROLL NO. 13873

Under the esteemed Guidance of:

Dr. Rajiv Kapoor

(PROFESSOR & HOD)
DEPARTMEN OF ELECTRONICS AND COMMUNICATION

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
DELHI COLLEGE OF ENGINEERING
UNIVERSITY OF DELHI

2008-2012

CERTIFICATE

DELHI COLLEGE OF ENGINEERING
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

Date: _____

This is certified that the dissertation entitled **"Designing a NAND Flash Interface I/O which meets all the specifications of ONFI 2.3"** is a work of **Mr. Pankaj Singh** (University Roll No- 13873) a student of Delhi College of Engineering. This work is completed under my direct supervision and guidance and forms a part of master of engineering (Electronics and communication) course and curriculum. He has completed his work with utmost sincerity and diligence.

The work embodied in this major project has not been submitted to any other Institute/University for the award of any other degree to the best of my knowledge.

Rajeev Kapoor
(Project Guide)
HOD & Professor
Department of Electronics and Communication.
Delhi College of Engineering,
University of Delhi, India

ACKNOWLEDGEMENT

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my supervisor **Prof. Rajiv Kapoor** HOD & Professor, Department of Electronics and communication, Delhi College of Engineering, for his invaluable guidance, encouragement and patient reviews. His continuous inspiration only has made me complete this dissertation. Without his help and guidance, this dissertation would have been impossible. He remained a pillar of help throughout the project.

I am extremely thankful to **Prof. Rajiv Kapoor**, Head of the Department, Electronics and communication, Delhi College of Engineering Delhi. I would also like to take this opportunity to present my sincere regards to all my teachers those who came to my help in some way for marking the project successful. I am also thankful to all non-teaching staff of Electronics and communication Department for providing me unconditional and any time access to the resources.

I am grateful to my parents for their moral support all the time; they have been always around to cheer me up, in the odd times of this work. I am also thankful to my classmates for their unconditional support and motivation during this work. Last but not least, I special thanks to the crowd who are active in the field of Audio Watermarking issues from whom I got the knowledge and inspiration to take up this field for my project work.

PANKAJ SINGH

Master of Engineering
(Electronics and Communication)
College Roll No. - 10/E&C/08
University Roll No. - 13873
Department of Electronics and Communication
Delhi College of Engineering, Delhi-110042

ABSTRACT

An interface is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware and software. This allows a component, whether a piece of hardware such as a graphic card or a piece of software such as an internet browser, to function independently while using interfaces to communicate with other components via an input /output system and an associated protocol.

The Open NAND Flash Interface Working Group (ONFI) is a consortium of technology companies working to develop open standards for NAND flash memory chips and devices that communicate with them. The formation of ONFI was announced at the Intel Developer Forum in March 2006. The ONFI Working Group is dedicated to simplifying integration of NAND Flash memory into consumer electronics (CE) applications and computing platforms.

ONFI 2.3 standard specifications were ratified in 16 AUG 2010 which includes the EZ-NAND protocol. EZ-NAND, which stands for error correction code (ECC) Zero NAND, was designed to remove the burden of the host controller to keep pace with the fast changing ECC requirements of NAND technology.

Here In this thesis work, I design a NAND Flash Interface I/O which meets all the specifications of ONFI 2.3. This I/O design is working for data rate of 200 MHz with I/O supply voltage of 1.8V. I tested my design in ADE using SPECTRE as simulator.

LIST OF ACRONYMS

ADE	Analog Design Environment
ASIC	Application Specific Integrated Circuit
BGA	Ball Grid Array
CE	Chip Enable
DDR	Double Data Rate
DIMM	Dual in Line Memory Module
DRAM	Dynamic Random Access Memory
ECC	Error Correction Code
EZ NAND	ECC Zero NAND
IC	Integrated Circuits
LGA	Land Grid Array
LUN	Logical Unit Number
MLC	Multi Level Cell
MUX	Multiplexer
ONFI	Open NAND Flash Interface
PC	Personal Computer
ROM	Read Only Memory
SLC	Single Level Cell
SP	Signal Processing
SSDs	Solid State Drives

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Chapter 1: Introduction

1.1 Introduction to Interface

An interface is a tool and concept that refers to a point of interaction between components, and is applicable at the level of both hardware and software. This allows a component, whether a piece of hardware such as a graphic card or a piece of software such as an internet browser, to function independently while using interfaces to communicate with other components via an input /output system and an associated protocol.

In addition to hardware and software interfaces, a computing interface may refer to the means of communication between the computer and the user by means of peripheral devices such as monitor or a keyboard, an interface with the Internet via internet protocol, and any other point of communication involving a computer.

A software interface may refer to a range of different types of interface at different "levels": an operating system may interface with pieces of hardware, application or programs running on the operating system may need to interact via streams, and in object oriented programs, objects within an application may need to interact via methods.

A Hardware interfaces exist in computing systems between many of the components such as the various buses, storage devices, other I/O devices, etc. A hardware interface is described by the mechanical, electrical and logical signals at the interface and the protocol for sequencing them.

1.2 Background:

The Open NAND Flash Interface Working Group (ONFI) is a consortium of technology companies working to develop open standards for NAND flash memory chips and devices that communicate with them. The formation of ONFI was announced at the Intel

Developer Forum in March 2006. The ONFI Working Group is dedicated to simplifying integration of NAND Flash memory into consumer electronics (CE) applications and computing platforms.

Before the advancements made by the working group, use of NAND Flash in these end-use applications was hampered by the lack of sufficient standardization. As of 2006, NAND flash

memory chips from most vendors use similar packaging, have similar pin-outs, and accept similar sets of low-level commands. As a result, when more capable and inexpensive models of NAND flash become available, product designers can incorporate them without major design changes. However, "similar" operation is not optimal: subtle differences in timing and command set mean that products must be thoroughly debugged and tested when a new model of flash chip is used in them. When a flash controller is expected to operate with various NAND flash chips, it must store a table of them in its firmware so that it knows how to deal with differences in their interfaces.

When new NAND Flash is released (from the same or a different vendor), the controller vendor needs to do numerous update:

- Look for new Flash Device ID
- Update timing to comprehend new part
- Update ECC to comprehend bit rate
- Other modifications based on device parameter

Another challenge: these changes must be made to a Mask ROM, with small storage area (64KB typical) and updates requiring a 1 month wait for a new IC

Result of These Challenges:

- Inventory control is difficult and array of parts confusing to customers.
- Mask ROMs are too small to support all Flash behavior, must choose subset.
- Several firmware versions must be maintained for every controller to support most NAND in the Market
- Lost revenue opportunity that could be rectified with standard interface.

This increases the complexity and time-to-market of flash-based devices, and means they are likely to be incompatible with future models of NAND flash, unless and until their

firmware is updated. Implementing these changes was extremely costly due to the new testing cycle required, which led to slower rates of adoption for new NAND Flash components. ONFI aims to remedy that problem and speed time to market for NAND Flash based applications. ONFI ensure no pre-association with NAND Flash at host design is required. It Adds mechanism for a device to self

-describe its features, capabilities, etc to the host via a parameter page. Features that cannot be self -described in a parameter page (like number of CE#) is host discoverable.

The **ONFI 1.0** specification was developed in **December 2006** to enable NAND Flash devices to self-describe their capabilities to host systems. This facilitates:

- Faster integration into host platforms
- The ability to add a new NAND device to an existing solution without firmware or software modifications

The specification also standardizes the NAND command set and establishes infrastructure for future evolution of NAND Flash capabilities, providing flexibility for supplier-specific optimizations.

ONFI 2.0 defines a high-speed NAND Flash interface that can deliver speeds greater than 133 MB/s, whereas the legacy NAND interface was limited to 50 MB/s. The full ONFI 2.0 specification was released in **February of 2008**.

The **NAND Connector** Specification was ratified in **April of 2008**. It specifies a standardized connection for NAND modules (similar to DRAM DIMMs) for use in applications like caching and SSDs in PC platforms.

ONFI 2.1 was ratified in **January of 2009** and contains a plethora of new features that deliver speeds of 166 MB/s and 200 MB/s, plus other enhancements to increase power, performance, and ECC capabilities

In **July 2009** ONFI created the **Block Abstracted NAND** addendum specification to simplify host controller design by relieving the host of the complexities of ECC, bad block management, and other low-level NAND management tasks. The ONFI Block Abstracted NAND revision 1.1 specifications adds the high speed source synchronous interface, which provides up to a 5X improvement in bandwidth compared with the traditional asynchronous NAND interface.

The ONFI Workgroup continues to evolve the ONFI specifications to meet the needs of a rapidly growing and changing industry.

Ratified in **October of 2009**, **ONFI 2.2** provides several useful new features:

- Individual LUN reset
- Enhanced program page register clear

LUN reset and page register clear enable more efficient operation in larger systems with many NAND devices, while the standardized Icc testing and definitions will provide simplified vendor testing and improved data consistency

---

1.2.1 ONFI 2.3

Aug. 16, 2010 The Open NAND Flash Interface (ONFI) Working Group, the organization dedicated to simplifying integration of NAND Flash memory into consumer electronic devices, computing platforms, and industrial systems, introduced its new **ONFI 2.3** specification, which includes the EZ-NAND protocol. EZ-NAND, which stands for error correction code (ECC) Zero NAND, was designed to remove the burden of the host controller to keep pace with the fast changing ECC requirements of NAND technology.

Before ONFI 2.3 in NAND implementations, the host controller must assume responsibility for all NAND management functions, including block management, wear leveling, and ECC. With ECC being almost solely dependent on the NAND technology and not the system, EZ-NAND will alleviate the issue of controller manufacturers having to implement and keep up with the rapidly changing NAND ECC requirements.

"ONFI's new 2.3 specification with its EZ-NAND protocol provides the ultimate win-win scenario. "NAND continues to scale at its own technology cadence while host controllers can be optimized for a variety of applications, free of ECC management issues. This can enable longer platform cycles for customers and potentially lower system cost."

"EZ-NAND enables faster adoption of new technologies by having NAND providers develop standard solutions to vendor and lithography specific challenges like ECC,"

NAND 2.3 specification offers maximum performance while preserving the external host controller's ability to offer its own unique capabilities.

The EZ-NAND packages and electrical interfaces are the same as defined in the ONFI 2.2 specification. ONFI 2.3 includes some minor command set changes to execute the optional ECC off-load protocol, with the goal being to keep ONFI 2.3 as compatible as possible with today's implementations.

ONFI supports two different data interface types: asynchronous and source synchronous. The asynchronous data interface is the traditional NAND interface that uses RE_n to latch data read, WE_n to latch data written, and does not include a clock. The source synchronous data interface includes a clock that indicates where commands and addresses should be latched and a data strobe that indicates where data should be latched. On power-up, the device shall operate in asynchronous data interface timing mode 0. After the host determines that the source synchronous data interface is supported in the parameter page, the host may select a source synchronous timing mode by using Set Features with a Feature Address of 01.

The source synchronous data interface uses a DDR protocol. Thus, an even number of bytes is always transferred. The least significant bit of the column address shall always be zero in the source synchronous data interface. If the least significant bit of the column address is set to one in the source synchronous data interface then the results are indeterminate.

1.3 Thesis Objective

The objective of the present thesis work is

- Designing a NAND Flash Interface I/O that can support a minimum of 200MHz data rate on 1.8V I/O supply, while meeting all the specifications of ONFI 2.3 standard.

1.4 Organization of Thesis

Chapter two give brief introduction about the objective and specification which I am going to use for my design verification. Chapter 3 is Literature Survey, which describe the requirements and need for the Interface I/O design. Chapter four covers the

Architecture of the Design and its components. Chapter four also describe which concepts and techniques I used for designing ONFI 2.3 Interface I/O. Chapter five explains the Simulaion results and various observations, observed during my thesis work. This thesis work is concluded in chapter 6, which also discusses the future scope of this work.

Chapter2: Signal Description and Specifications

2.1 Signal Description:

Table 1 provides the signal descriptions.

Where the pin function has an inverted logic sense, that is, the function is true or invoked for a low signal, the overbar, trailing slash (\), #, or _n symbol is appended to the symbol. The _t and _c symbols represent true and complementary logic states for differential pairs

Signal name	**Input/ Output**	**Description**
R/B_x_n	O	**Ready/Busy** The Ready/Busy signal indicates the target status. When low, the signal indicates that one or more LUN operations are in progress. This signal is an open drain output and requires an external pull-up.

W/R_x_n	I	**Write/Read Direction** The Write/Read Direction signal indicates the owner of the DQ bus and DQS signal in the source synchronous data interface. This signal shares the same pin as RE_x_n in the asynchronous data interface
CE_x_n	I	**Chip Enable** 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.
Vcc	I	**Power** The Vcc signal is the power supply to the device
VccQ	I	**I/O Power** The VccQ signal is the power supply for input and/or output signals
Vss	I	**Ground** The Vss signal is the power supply ground
VssQ	I	**I/O Ground** The VssQ signal is the ground for input and/or output signals
VREFQ_x	I	**Voltage Reference** This signal is reserved for future use
VDDi	NA	**ASIC Voltage Control** This signal is used to assist in stabilizing the internal power supply to a NAND controller ASIC (e.g. EZ NAND) by connecting to an external capacitor
CLE_x	I	**Command Latch Enable** The Command Latch Enable signal is one of the signals used by

		the host to indicate the type of bus cycle (command, address, data).
ALE_x	I	**Address Latch Enable** The Address Latch Enable signal is one of the signals used by the host to indicate the type of bus cycle (command, address, data).
CLK_x_t	I	**Clock** The Clock signal is used as the clock in the source synchronous data interface. This signal shares the same pin as WE_x_n in the asynchronous data interface.
CLK_x_c	I	**Clock Complement** This signal is reserved for future use
WP_x_n	I	**Write Protect** The Write Protect signal disables Flash array program and erase operations
DQ0_0 – DQ7_0	I/O	**DQ Port 0, bits 0-7** The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device.
DQS_x_t	I/O	**Data Strobe** The data strobe signal that indicates the data valid window for the source synchronous data interface
DQS_x_c	I/O	**Data Strobe Complement** This signal is reserved for future use
DQ0_1 – DQ7_1	I/O	**I/O Port 1, bits 0-7** The I/O port is an 8-bit wide bidirectional port for transferring address, command, and data to and from the device. These pins may be used as an additional 8-bit wide bidirectional port for devices that support two independent data buses.

VSP_x		**Vendor Specific** The function of these signals is defined and specified by the NAND vendor. Devices shall have an internal pull-up or pull-down resistor on these signals to yield ONFI compliant behavior when a signal is not connected by the host. Any VSP signal not used by the NAND vendor shall not be connected internal to the device.
R		**Reserved** These pins shall not be connected by the host
RFT		**Reserved for Test** These pins shall not be connected by the host

Table 1 Signal descriptions

2.2 Specifications:

These specifications defines a standardized NAND Flash device interface that provides the means for a system to be designed that supports a range of NAND Flash devices without direct design pre-association. The solution also provides the means for a system to seamlessly make use of new NAND devices that may not have existed at the time that the system was designed.

Some of the goals and requirements for the specifications include:

 Support range of device capabilities and new unforeseen innovation

 Consistent with existing NAND Flash designs providing orderly transition to ONFI

 Capabilities and features are self‐described in a parameter page such that hard-coded chip ID tables in the host are not necessary

 Flash devices are interoperable and do not require host changes to support a new Flash device

 Define a higher speed NAND interface that is compatible with existing NAND Flash interface

 Allow for separate core(Vcc) and I/O (VccQ) power rails

 Support for off loading NAND lithography specific functionality to a controller stacked in the NAND package (EZ NAND)

The complete sets of specification given for the current design are shown below:

PARAMETER	**SYMBOL**	**TYP**	**MAX**	**MIN**	**UNITS**
Supply voltage for 1.8 volt Devices	VCC	1.8	1.95	1.7	V
Supply voltage for 1.8 V I/O Signaling	VCCQ	1.8	1.95	1.7	V
Ground Voltage Supply	VSS	0	0	0	V
Ground Voltage Supply for I/O Signaling	VSSQ	0	0	0	V

Table 2 Recommended DC operating condition

Parameter	**Symbol**	**Test Condition**	**Min**	**Typ**	**Max**	**Units**
DC Input High Voltage	VIH(DC)	-	0.7*VCCQ	-	VCCQ+0.3	V
AC Input High	VIH(AC)	-	0.8*VCCQ	-	VCCQ+0.3	V

Voltage						
DC Input Low Voltage	VIL(DC)	-	-0.3	-	VCCQ*0.3	V
AC Input Low Voltage	VIL(AC)	-	-0.3	-	VCCQ*0.2	V
Standby current, CMOS	ISBQ	CE_n= VccQ-0.2V, WP_n=0V/VccQ	-	-	25	uA
Input leakage current	ILI	VIN=0V to VccQ	-	-	±10	uA
Output leakage current	ILO	VOUT=0V to VccQ	-	-	±10	uA

Table 3 DC and Operating Conditions for VccQ of 1.8V, measured on VccQ

Setting	Driver Strength	VCCQ
18 Ohm	2.0x = 18 Ohm	1.8 V
25 Ohm	1.4x = 25 Ohm	1.8 V
35 Ohm	1.0x = 35 Ohm	1.8 V

50 Ohm	0.7x = 50 Ohm	1.8 V

Table 4 I/O Drive Strength Settings

Condition	Temperature	VccQ(1.8 V)	Process
Min Impedance	-40 degree celsius	1.95V	Fast- Fast
Nominal Impedance	25 degree celsius	1.8V	Typical
Max Impedance	110 degree celsius	1.7V	Slow-Slow

Table 5 Testing Conditions for Impedance Value

Description	Timing Mode						Units
	0	1	2	3	4	5	
Input Slew Rate (max)	4.5	4.5	4.5	4.5	4.5	4.5	V / n-sec
Input Slew Rate (min)	0.5	0.5	0.5	0.5	0.5	0.5	V / n-sec

Table 6 Input Slew Rate Requirements

Description	VOUT To VssQ	Maximum	Nominal	Minimum	Units
18 Ohm Pulldown	0.2 x VccQ	34.0	13.5	7.5	Ohms
18 Ohm Pullup	0.2 x VccQ	44.0	23.5	11.0	Ohms

25 Ohm Pulldown	0.2 x VccQ	47.0	19.0	10.5	Ohms
25 Ohm Pullup	0.2 x VccQ	61.5	32.5	16.0	Ohms
35 Ohm Pulldown	0.2 x VccQ	66.5	27.0	15.0	Ohms
35 Ohm Pullup	0.2 x VccQ	88.0	52.0	22.0	Ohms
50 Ohm Pulldown	0.2 x VccQ	95.0	39.0	21.5	Ohms
50 Ohm Pullup	0.2 x VccQ	126.5	66.5	31.5	Ohms

Table 7 Impedance Values for 1.8V VccQ

Output Impedance	Maximum	Minimum	Units
18 Ohms	6.3	0	Ohms
25 Ohms	8.8	0	Ohms
35 Ohms	12.3	0	Ohms
50 Ohms	17.7	0	Ohms

Table 8 Pull-up and Pull-down Impedance Mismatch

Parameter	**Value**
Positive Input Transition	VIL(DC) TO VIH(AC)

Negative Input Transition	VIH(DC) TO VIL(AC)

Table 9 Testing Conditions for Input Slew Rate

Description	Output Slew Rate		Units	Normative Or Recommended
	Min	Max		
18 Ohm	1.0	5.5	V / n-sec	Normative
25 Ohm	0.85	5.0	V / n-sec	Normative
35 Ohm	0.75	4.0	V / n – sec	Normative
50 Ohm	0.6	4.0	V / n-sec	Recommended

Table 10 Output Slew Rate Requirements for 1.8V VccQ

Parameter	Value
VOL(DC)	0.3*VccQ
VOH(DC)	0.7*VccQ
VOL(AC)	0.2*VccQ
VOH(AC)	0.8*VccQ
Positive Output Transition	VOL(DC) to VOH(AC)
Negative Output Transition	VOH(DC) to VOL(AC)
tRISE	Time During Rising Edge From VOL(DC) to VOH(AC)
tFALL	Time During Falling Edge From VOH(DC) to VOL(AC)

Output Slew Rate Rising Edge	[VOH(AC) – VOL(DC)] / Trise
Output Slew Rate Falling Edge	[VOH(DC) - VOL(AC)] / tFALL
Output Load	50 Ohm to Vtt

Table 11 Testing Conditions for Output Slew Rate

Parameter	**Description**	**Mode 5 (100 MHz or 10 n-sec)**		**Units**
		Min	**Max**	
tAC	Access window of DQ[7:0] from CLK	3	20	Ns
tADL	Address cycle to data loading time	70	-	Ns
tCADf	Command, Address, Data delay	25	-	Ns
tCADs	Command, Address, Data delay	45	-	Ns
tCAH	Command/address DQ hold time	2	-	ns
tCALH	W/R_n, CLE and ALE hold time	2	-	Ns
tCALS	W/R_n, CLE and ALE setup time	2	-	Ns

tCAS	Command/address DQ setup time	2	-	Ns
tCEH	CE_n high hold time	20	-	ns
tCH	CE_n hold time	2	-	Ns
tCK(avg) or tCK	Average clock cycle time, also known as tCK	10	-	Ns
tCK(abs)	Absolute clock period, measured from rising edge to the next consecutive rising edge	Minimum:tCK(avg) + tJIT(per) min Maximum: tCK(avg)+ tJIT(per) max		Ns
tCKH(abs)	Clock cycle high	0.43	0.57	tCK
tCKL(abs)	Clock cycle low	0.43	0.57	tCK
tCKWR	Data output end to W/R_n high	Minimum: RoundUp{[tDQSCK(max) + tCK] / tCK} Maximum : ---------		tCK
Tcs	CE_n setup time	15	-	Ns

tDH	Data DQ hold time	0.9	-	Ns
tDPZ	Data input pause setup time	1.5	-	tDSC
tDQSCK	Access window of DQS from CLK	---	20	Ns
tDQSD	W/R_n low to DQS/DQ driven by device	0	18	Ns
tDQSH	DQS input high pulse width	0.4	0.6	tCKor tDSC
tDQSHZ	W/R_n high to DQS/DQ tri-state by device	----	20	Ns
tDQSL	DQS input low pulse width	0.4	0.6	tCKor tDSC
tDQSQ	DQS-DQ skew, DQS to last DQ valid, per access	--	0.85	Ns
tDQSS	Data input to first DQS latching transition	0.75	1.25	tCK
tDS	Data DQ setup time	0.9	-	Ns
tDSC	DQS cycle time	10	-	Ns
tDSH	DQS falling edge to CLK rising – hold time	0.2	-	tCK

tDSS	DQS falling edge to CLK rising – setup time	0.2	-	Tck
tDVW	Output data valid window	tDVW = tQH – tDQSQ		Ns
tFEAT	Busy time for Set Features and Get Features	---	**1**	Us
tHP	Half-clock period	tHP = min(tCKL, tCKH)		Ns
tITC	Interface and Timing Mode Change time	--	1	Us
tJIT(per)	The deviation of a given tCK(abs) from tCK(avg	0.5	0.5	Ns
tQH	DQ-DQS hold, DQS to first DQ to go non-valid, per access	tQH = tHP – tQHS		Ns
tQHS	Data hold skew factor	--	**1**	Ns
tRHW	Data output cycle to command, address, or data input cycl	100	-	Ns
tRR	Ready to data output cycle (data only	20	-	ns

tRST (raw NAND)	Device reset time, measured from the falling edge of R/B_n to the rising edge of R/B_n.	-	5/10/500	Us
tRST (EZ NAND)	Device reset time, measured from the falling edge of R/B_n to the rising edge of R/B_n.	-	150/150/500	us
tWB	CLK rising edge to SR[6] low	-	-	Ns
tWHR	Command, address or data input cycle to data output cycle	60	-	Ns
tWPRE	DQS write preamble	1.5	-	tCK
tWPST	DQS write postamble	1.5	-	tCK
tWRCK	W/R_n low to data output cycle	20	-	Ns
tWW	WP_n transition to command cycle	100	-	Ns

Table 12 Source Synchronous Timing Mode 5 Specs

All source synchronous timing parameters are referenced to the rising edge of CLK or the latching edge of DQS. Note that R/B_n and WP_n are always asynchronous signals. If CLK is a different frequency than those described in the source synchronous timing modes, then the host shall meet the setup and hold requirements for the next fastest timing mode.

Chapter 3: Literature Survey

3.1 CE_n Signal Requirements

If one or more LUNs are active and the host sets CE_n to one, then those operations continue executing until completion at which point the target enters standby. After the CE_n signal is transitioned to one, the host may drive a different CE_n signal to zero and begin operations on another target. Note that if using a dual x8 package (e.g. BGA-100), then operations may execute in parallel on two different CE_n‘s if they are connected to different 8-bit data buses.

When SR[6] for a particular LUN is cleared to zero and the CE_n signal for the corresponding target is cleared to zero, the host may only issue the Reset, Synchronous Reset, Read Status, or Read Status Enhanced commands to that LUN.

3.2 Source Synchronous Data Interface Requirements

When using the source synchronous data interface, the following requirements shall be met if the device does not support CLK being stopped during data input:

1. CLK shall only stop or start when CE_n is high.

When using the source synchronous data interface, the following requirements shall be met if the device supports CLK being stopped during data input:

1. CLK shall only stop or start when either:
    a. CE_n is high, or
    b. CE_n is low and the bus state is data input

When using the source synchronous data interface, the following requirements shall always be met:

1. CLK shall only change frequency when CE_n is high.
2. When CE_n is low, CLK shall maintain the same frequency.
3. CE_n shall only transition from one to zero when the CLK is stable and has a valid period based on the timing mode selected.
4. The interface shall be in an idle state when CE_n changes value.
   CE_n shall only transition when the following are true:
    a. ALE and CLE are both cleared to zero

3.3 Calculating Pin Capacitance

To calculate the pin capacitance for all loads on the I/O bus, the host should utilize the reported pin capacitance per target in Read Parameter Page . The maximum capacitance may be used, or the typical capacitance if provided by the device may be used. The algorithm to use is:

```
PinCapacitance = 0;
for (target = 0; target < TotalTargets; target++)
        PinCapacitance += GetCapacitanceFromRPP(target);
```

This methodology will calculate an accurate maximum or typical pin capacitance, respectively, accounting for all targets present.

3.4 Staggered Power-up

Subsystems that support multiple Flash devices may experience power system design issues related to the current load presented during the power-on condition. To limit the current load presented to the host at power-on, all devices shall support power-up in a low-power condition.

Until a Reset (FFh) command is received by the target after power-on, the target shall not draw more than IST of current per LUN and ISTQ (if present for devices that support EZ NAND). For example, a target that contains 4 LUNs may draw up to 40 mA of current until a Reset (FFh) command is received after power-on.

This value is measured with a nominal rise time (tRise) of 1 millisecond and a line capacitance (cLine) of 0.1 µF. The measurement shall be taken with 1 millisecond averaging intervals and shall begin after Vcc reaches Vcc_min and VccQ reaches VccQ_min.

3.5 Independent Data Buses

There may be two independent 8-bit data buses in some ONFI packages (i.e. the LGA and the 100-ball BGA package). If the device supports two independent data buses, then CE1_n and CE3_n (if connected) shall use the second data bus, which on these packages are marked CE0_1_n and CE1_1_n. CE0_n and CE2_n shall always use the first data bus pins, which on these packages are marked CE0_0_n and CE1_0_n. Note that CE0_n, CE1_n, CE2_n, and CE3_n may all use the first data bus and the first set of control

signals (RE0_n, CLE0_n, ALE0_n, WE0_n, and WP0_n) if the device does not support independent data buses.

Implementations may tie the data lines and control signals (RE_n, CLE, ALE, WE_n, WP_n, and DQS) together for the two independent 8-bit data buses externally to the device.

3.6 Ready/Busy (R/B_n) Requirements

3.6.1 Power-On Requirements

Once VCC and VccQ reach the VCC minimum and VccQ minimum values, respectively, listed in Table 2 and power is stable, the R/B_n signal shall be valid after RB_valid_Vcc and shall be set to one (Ready) within RB_device_ready, as listed in Table16 . R/B_n is undefined until 50 µs has elapsed after VCC has started to ramp. The R/B_n signal is not valid until both of these conditions are met.

Parameter	Raw NAND	EZ NAND
RB_valid_VCC	10us	250us
RB_device_ready	1 ms	2ms

Table 13 R/B_n Power-on Requirements

During power-on, VccQ shall be less than or equal to Vcc at all times. Figure 3.1 shows VccQ ramping after Vcc, however, they may ramp at the same time.

3.6.2 R/B_n and SR[6] Relationship

R/B_n shall reflect the logical AND of the SR[6] (Status Register bit 6) values for all LUNs on the corresponding target. For example, R/B3_n is the logical AND of the SR[6] values for all LUNs on CE3_n. Thus, R/B_n reflects whether any LUN is busy on a particular target.

3.7 Bus Width Requirements

All targets per device shall use the same data bus width. All targets in case of source synchronous interface shall have an 8-bit bus width.

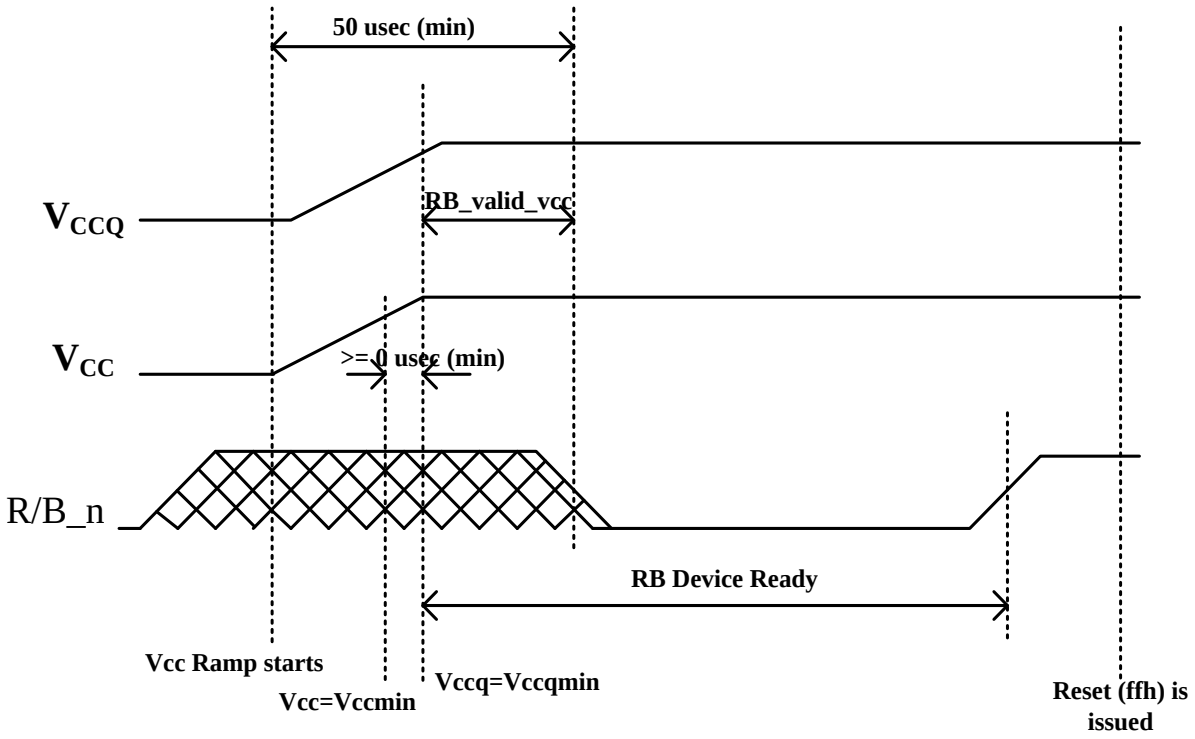


Fig 3.1 R/B_n Power on Behaviour

3.8 Write Protect

When cleared to zero, the WP_n signal disables Flash array program and erase operations. This signal shall only be transitioned while there are no commands executing on the device. After modifying the value of WP_n, the host shall not issue a new command to the device for at least tWW delay time.

Figure 3.2 describes the tWW timing requirement, shown with the start of a Program command. The transition of the WP_n signal is asynchronous and unrelated to any CLK transition in the source synchronous data interface. The bus shall be idle for tWW time

after WP_n transitions from zero to one before a new command is issued by the host, including Program. The bus shall be idle for tWW time after WP_n transitions from one to zero before a new command is issued by the host.

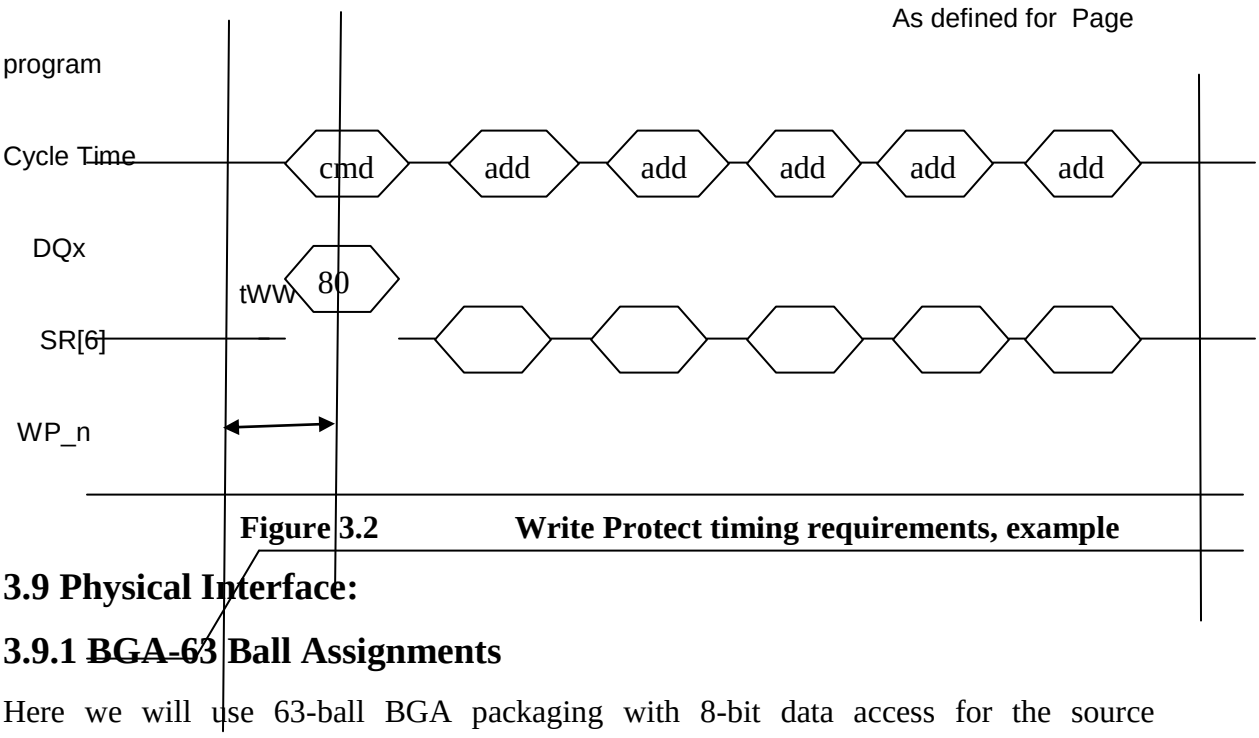


Figure 3.2 Write Protect timing requirements, example

3.9 Physical Interface:

3.9.1 BGA-63 Ball Assignments

Here we will use 63-ball BGA packaging with 8-bit data access for the source synchronous data interface which defines the ball assignments for devices. Figure 3.3 defines the ball assignments for devices using 63-ball BGA packaging with 8-bit data access for the source synchronous data interface. The ball spacing for the 63-ball BGA package is 0.8mm between each. The solder ball diameter is 0.45 mm post reflow.

0.8mm

R	R							R	R
R								R	R
		WP_n	ALE	VSS	CE0_n	R	R/B0_n		
		VCC	W/R_n	CLE	CE1_n	CE2_n	R/B1_n		

		R	R	R	R	CE3_n	R/B2_n		
		R	R	VREFQ	R	VSS	R/B3_n		
		VSP3	VCC	VSP1	R	R	VSP2		
		R	DQ0	DQS_c	CLK_c	CLK_t	VCCQ		
		R	DQ1	DQS_t	VCCQ	DQ5	DQ7		
		VSSQ	DQ2	DQ3	DQ4	DQ6	VSSQ		
R	R							R	R
R	R							R	R

Figure 3.3 BGA-63 ball assignments for 8-bit data access

3.10 Existing ONFI Standards before ONFI 2.3

Firstly when in 2006 ONFI released its first specifications ONFI 1.0, at that time the NAND array reads are parallel and very fast

• Read array bandwidth is greater than 330MB/s (8KB read in 25us)

Interface speed is the limiting factor

• Read bus bandwidth is 40MB/s (25ns clock)

Total throughput after array read and I/O transfer is 34MB/s in case of SLC (single level cell) 4KB Page size and 30 MB/S in case of MLC (multi level cell) 4 KB Page size. So read performance is I/O limited because tIO ≫ tR. But if we compare with the programming performance Interface speed is not the limiting factor as Program performance is not so impressive. Array program bandwidth is 33MB/s. So total throughput after I/O transfer and array programming is 17MB/s for SLC 4KB Page size

and 7MB/s for MLC 4KB Page size. So program performance is array limited because tPROG ≫tIO, especially for MLC devices.

System performance is improved by introducing the source synchronous interface defined in ONFI 2.0 increases the bandwidth of each I/O channel while adding only one pin, DQS. ONFI 2.1 improves the I/O channel performance up to 200MB/s. It is designed for up to 16 die per I/O channel through use of output impedance control. The protocol is backwards compatible to asynchronous NAND reducing or eliminating firmware changes for command set. Because ONFI 2 devices are backwards compatible with the asynchronous NAND interface, controllers that only support the asynchronous interface can still use these devices.

ONFI 2 provides a significant increase to throughput per I/O channel

- Allows better utilization of bus bandwidth
- Requires less I/O channels overall
- For read operations, a single die can immediately take advantage of higher bandwidth
- For program operations, multiple die take advantage of higher bandwidth and reduce number of I/O channels required to achieve target bandwidths.

ONFi 2.1 was ratified in mid-January and contains a plethora of new features. New features include:

• 166 MB/s and 200 MB/s speeds
• Power management enhancements
• Enhanced ECC information
• New commands for increased performance and functionality

ECC is required to correct for bit errors that naturally occur with NAND

• As the raw bit error (RBER) of NAND increases, the amount of ECC applied goes up
• Communicating error correction needs effectively, is critical

NAND Scaling continues at a relentless pace. In fact it is approaching the atomic level where storage levels are separated by a countable number of electrons. But NAND Scaling increases complexity in terms of ECC and Signal management. ECC requirements have been increasing for years as shown in Figure 3.4. Signal management has long been key to Flash reliability, but no longer contained at the interface boundary.

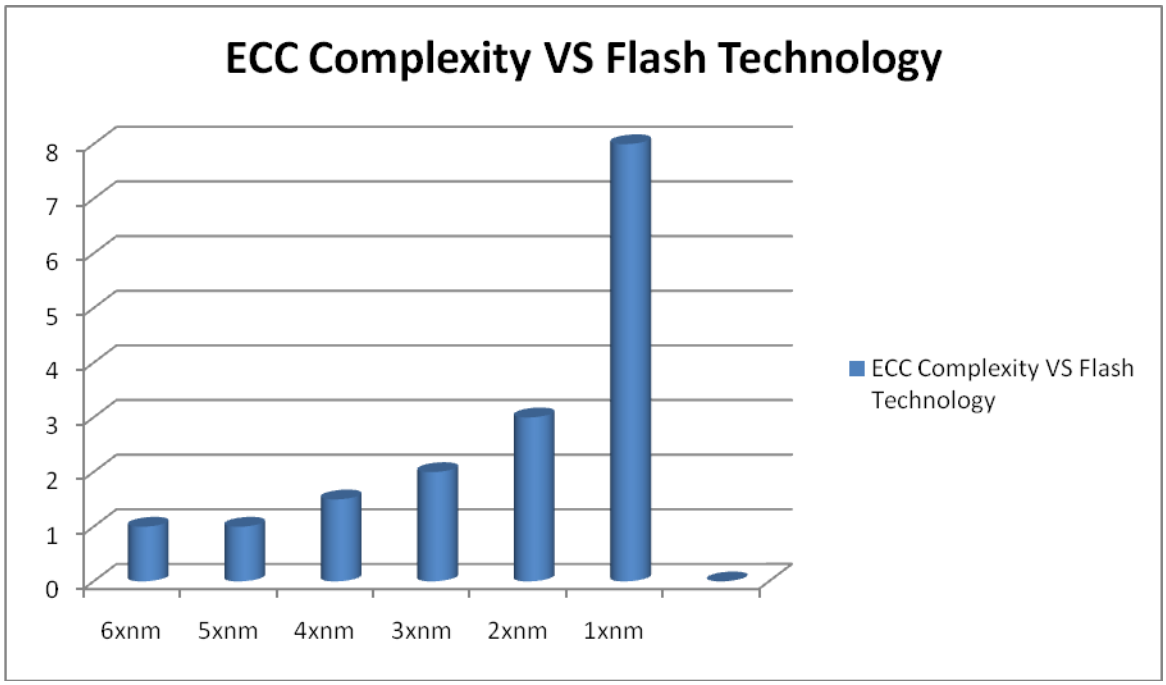
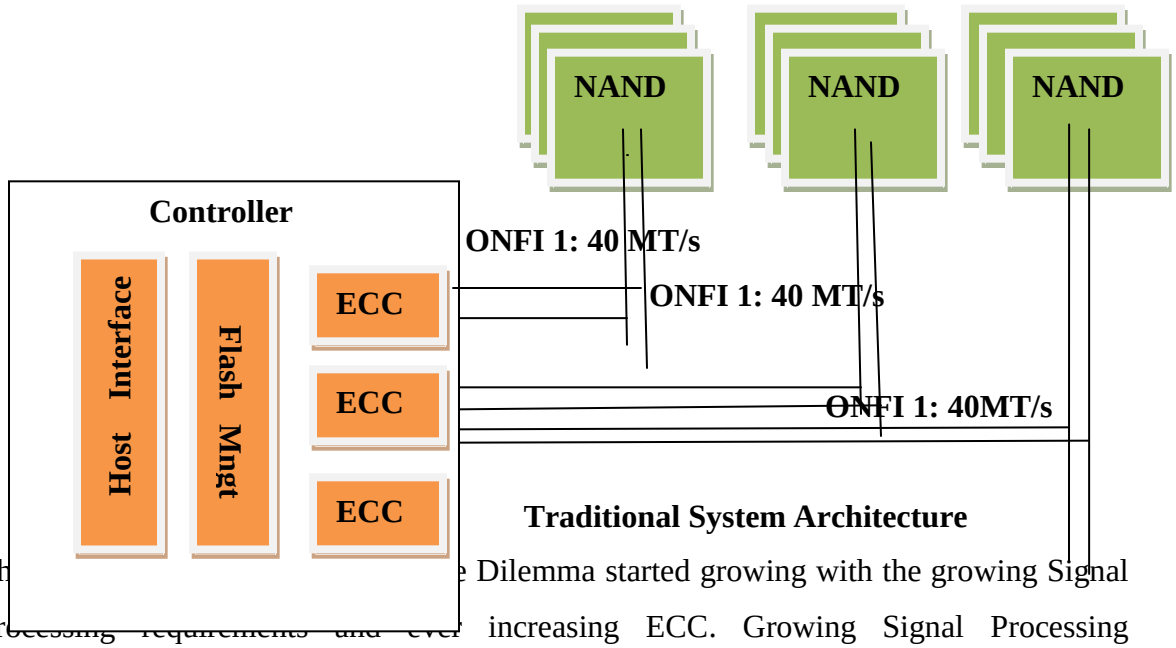


Figure 3.4 Showing Increasing ECC Complexity with scaling technology

Traditional Systems using NAND have architectures similar to as shown in figure 3.5. ECC sized to support target suppliers across a range of technology nodes. Controller must be sized for maximum capacity and performance.



Traditional System Architecture

Th e Dilemma started growing with the growing Signal Processing requirements and ever increasing ECC. Growing Signal Processing

requirements and ever increasing ECC creates controller implementation more tied to Flash Technology than ever. Because ECC+SP can dominate the controller gate count and the pin requirements can impact the die size and package selection.

Than Emerging Architecture – Abstracted Storage using EZ-NAND came into picture. Tightly coupling of ECC and SP ensures error management is precisely tuned for the NAND technology. Controller ASIC becomes the independent of flash technology as shown in figure 3.7

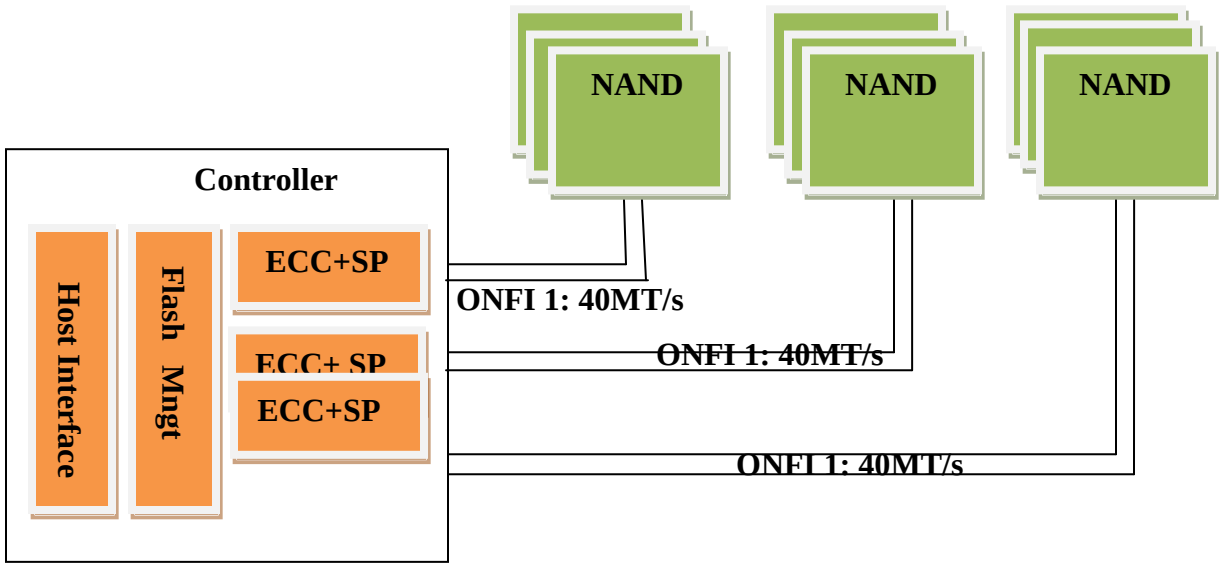
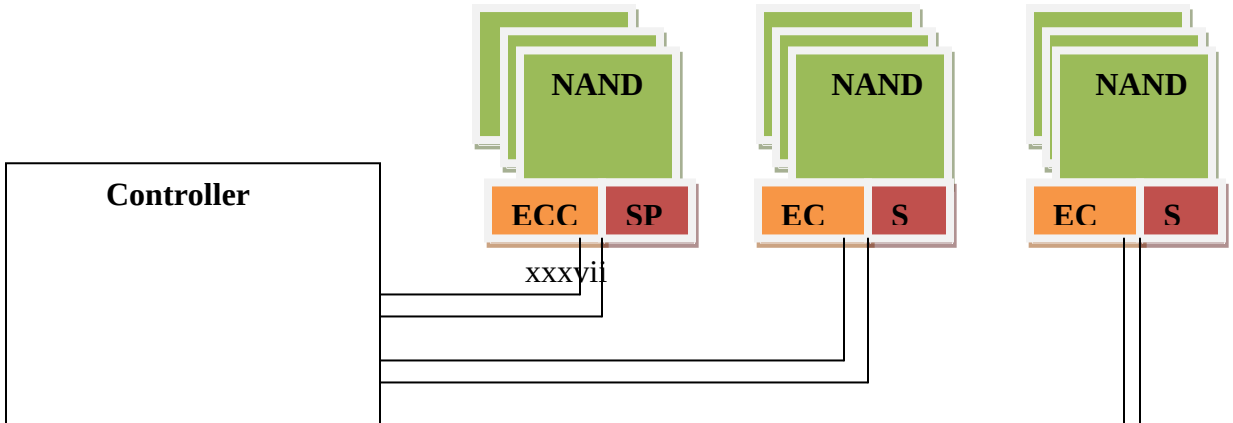
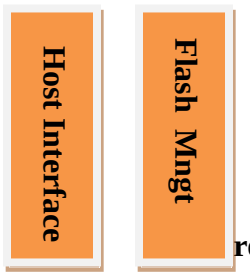


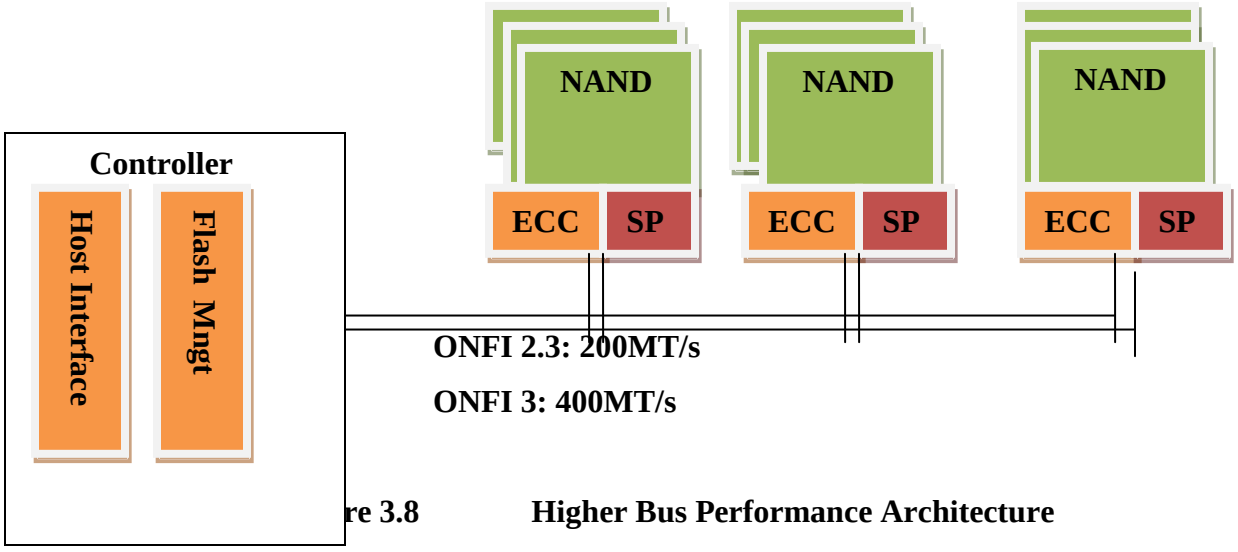
Figure 3.6 Traditional System Architecture –Growing Dilemma





re 3.7 Abstracted Storage using EZ-NAND

Than when ONFI 2 came into picture a architecture was designed for high BUS Performance. High bus speeds and the EZ – NAND Controller combine to increase the capacity and performance per pin on the Controller. Controller cost are minimized. This architecture provide Controller with reduced pin count by factor 1/3 and 3x performance increase.



re 3.8 Higher Bus Performance Architecture

3.11 Data Output Cycle Timing Diagram for ONFI 2.3

Data output cycle timing describes timing for data transfers from the device to the host (i.e. data reads). The host shall not start data output (i.e.transition ALE/CLE to 11b) until the tDQSD time has elapsed. Data output cycle are shown in fig 3.9

3.12 Data Input Cycle Timing Diagram for ONFI 2.3

Data input cycle timing describes timing for data transfers from the host to the device (i.e. data writes).Data input cycle are shown in fig 3.10

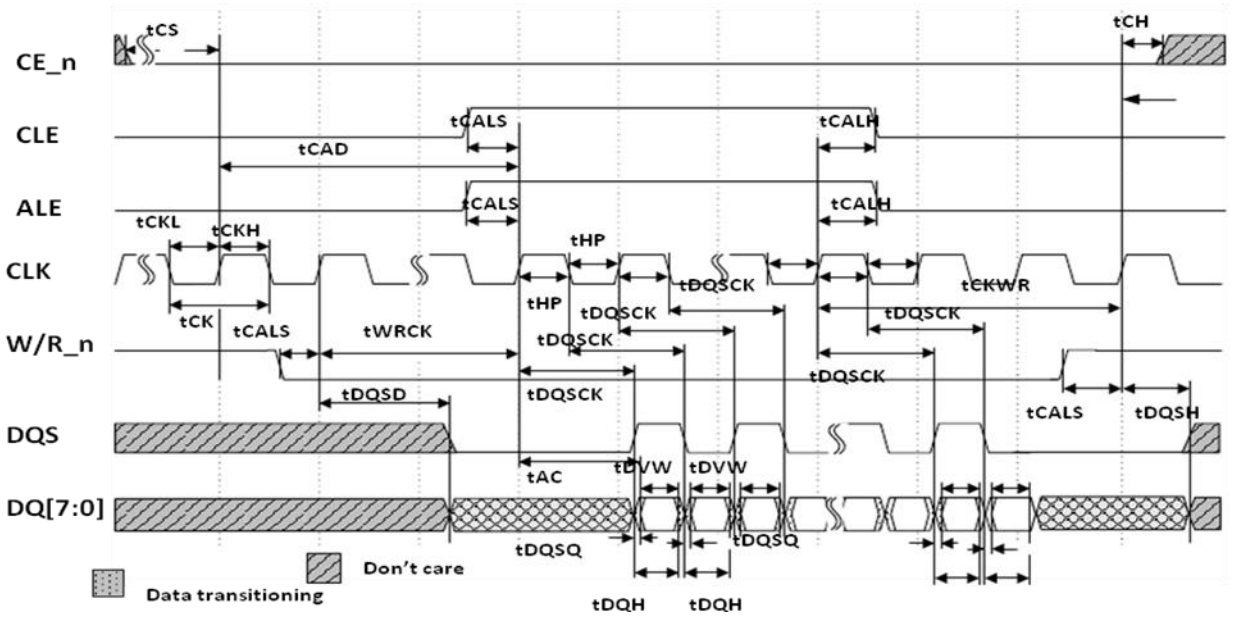


Fig 3.9 Data Output Cycle Timing

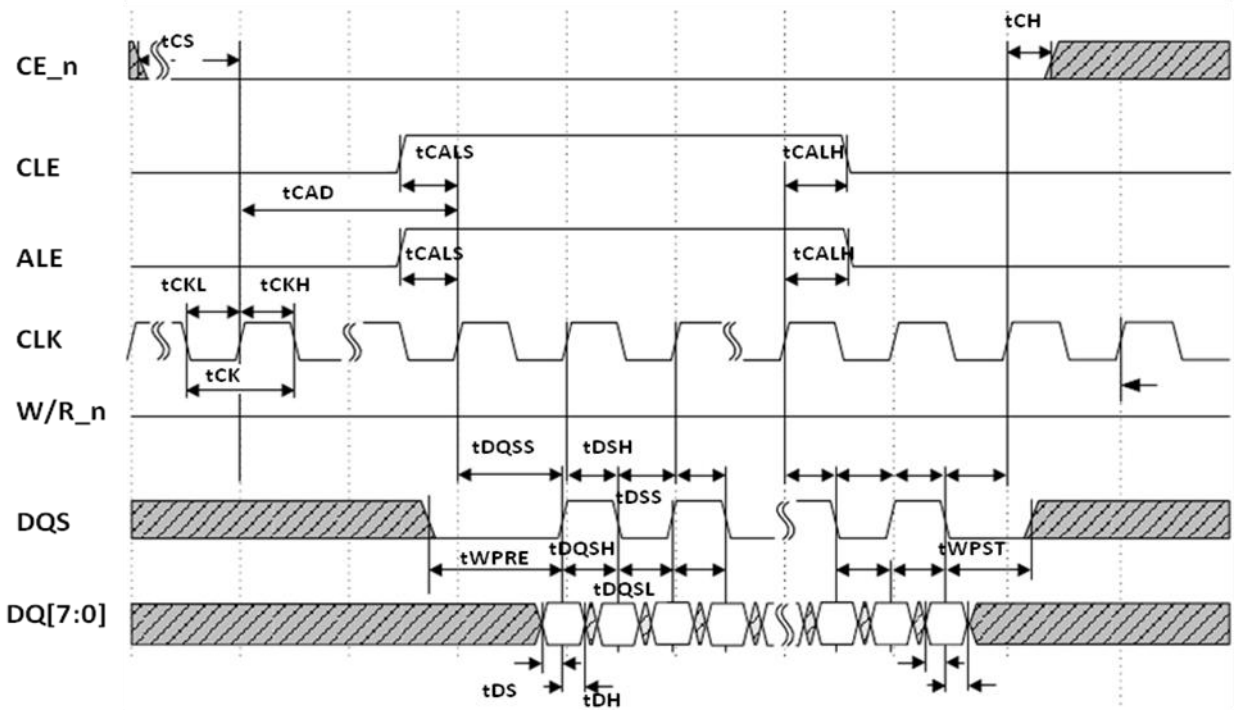


Fig 3.10 Data Input Cycle Timing

Chapter 4: Architecture

The factors discussed in the earlier chapters motivated to develop a new (at least modified in accordance with the current requirements if not completely new) ONFI I/O interface that has high speed of 200MT/s working for all deriver strength of 18ohm, 25ohm, 35ohm and 50ohm , on power supply voltage of 1.8V.

4.1 Development of ONFI 2.3 Data Transmitter For all strength

The Basic idea starts with the designing of Driver for 18ohm strength by designing the PMOS and NMOS with corresponding 0.2 VCCQ, with pull up and pull down resistances as mentioned in the Table 7 by doing dc simulations. I have used the thick gate devices for the driver design as this is working for 1.8V supply. In the driver design I have used the Ballasting resistors for ESD Protection with the concept of 50 ohm resistor for every 10 micron length. The verification of my driver designed is done by estimating its rise and fall time, output slew rate in accordance with the table 10.

Than I designed the pre-driver for the driver so that we can use for all strengths by enabling the respective PMOS and NMOS of driver part. Pre-Driver are designed according to the concept of FO4 with respect to the driver part. As I have used NOR as a pre-driver for NMOS part and NAND as pre-driver for PMOS part. Because of the enable signal present on the pre-driver I am able to design my driver for the other driving strength by enabling or disabling the respective no. of PMOS and NMOS devices. Pre-driver are also designed using thick Gate devices.

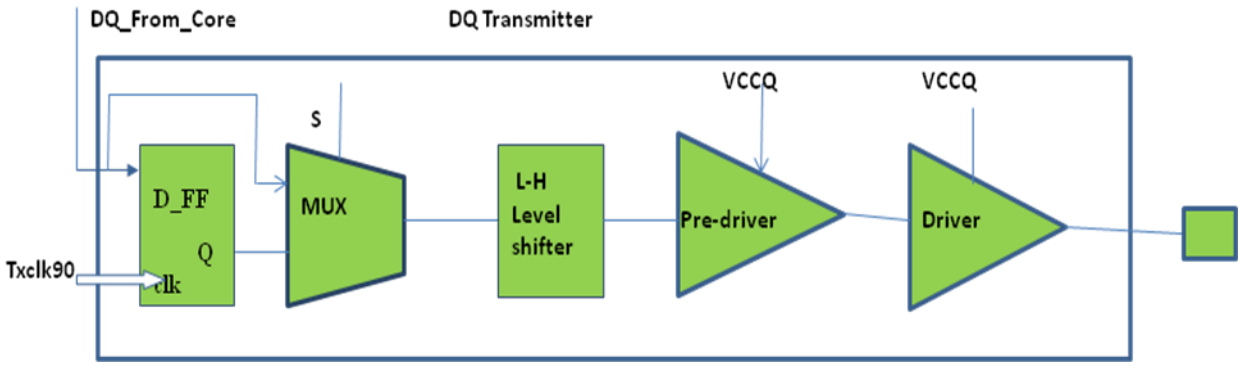


Fig 4.1 Top Level Block for Data Transmitter

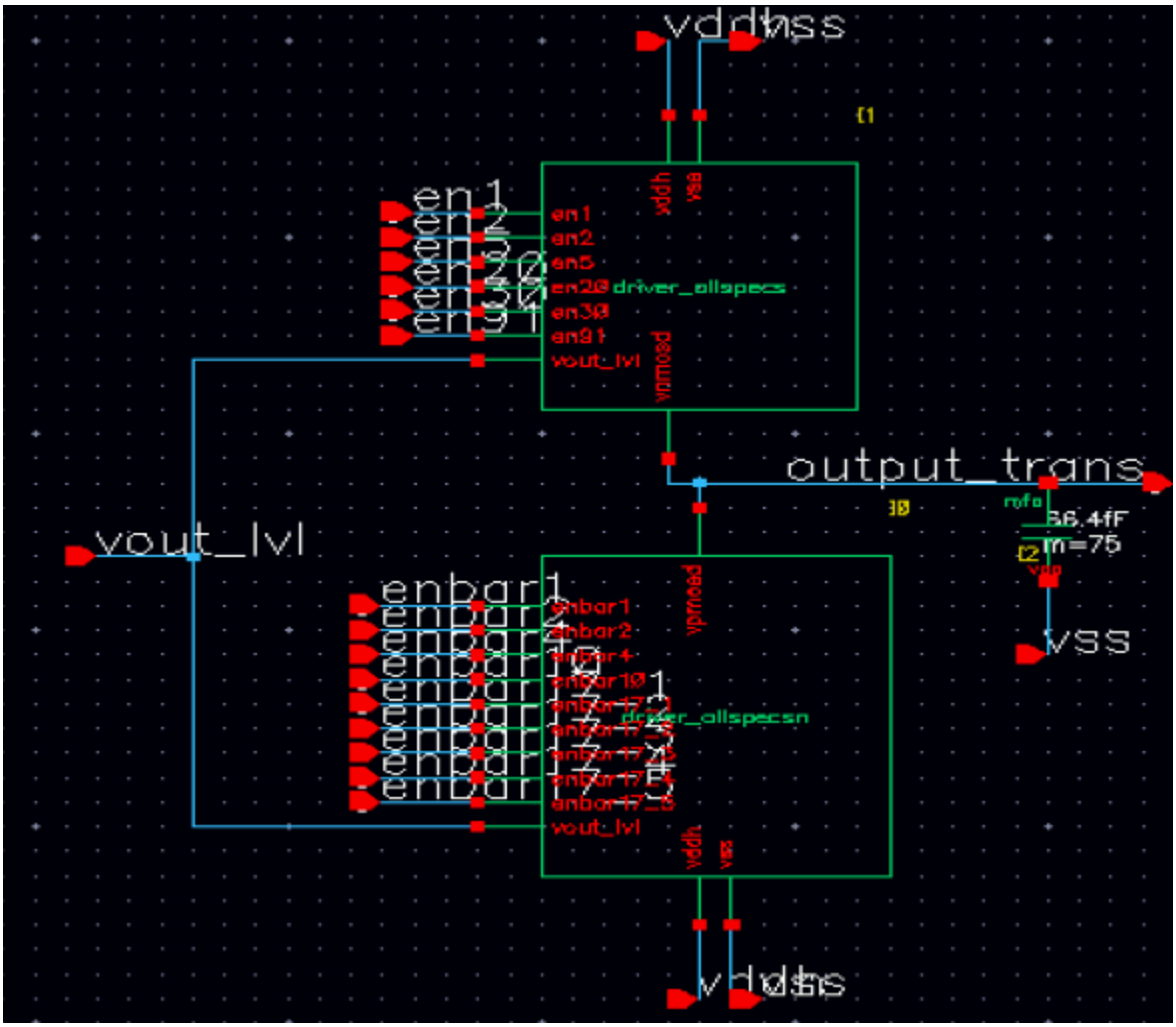


Figure 4.2 Driver Design Meeting all the Driving Strengths

Than i designed the level shifter as shown in fig 4.2 which shifts the voltage level from 1.05V to 1.8 V. The Level Shifter output is the input of the driver. So for obtaining the maximum swing I used the two buffers between level shifter and pre-driver. The Buffers are designed according to the concept of FO4 respective to pre-driver part. I have used the four terminal thick gate devices for level shifter, buffers implementation too. The inverter is driving NTG1 using FO4 concept. The sizes of PTG0 and PTG1 are same and NTG0 and NTG1 are of same sizes.

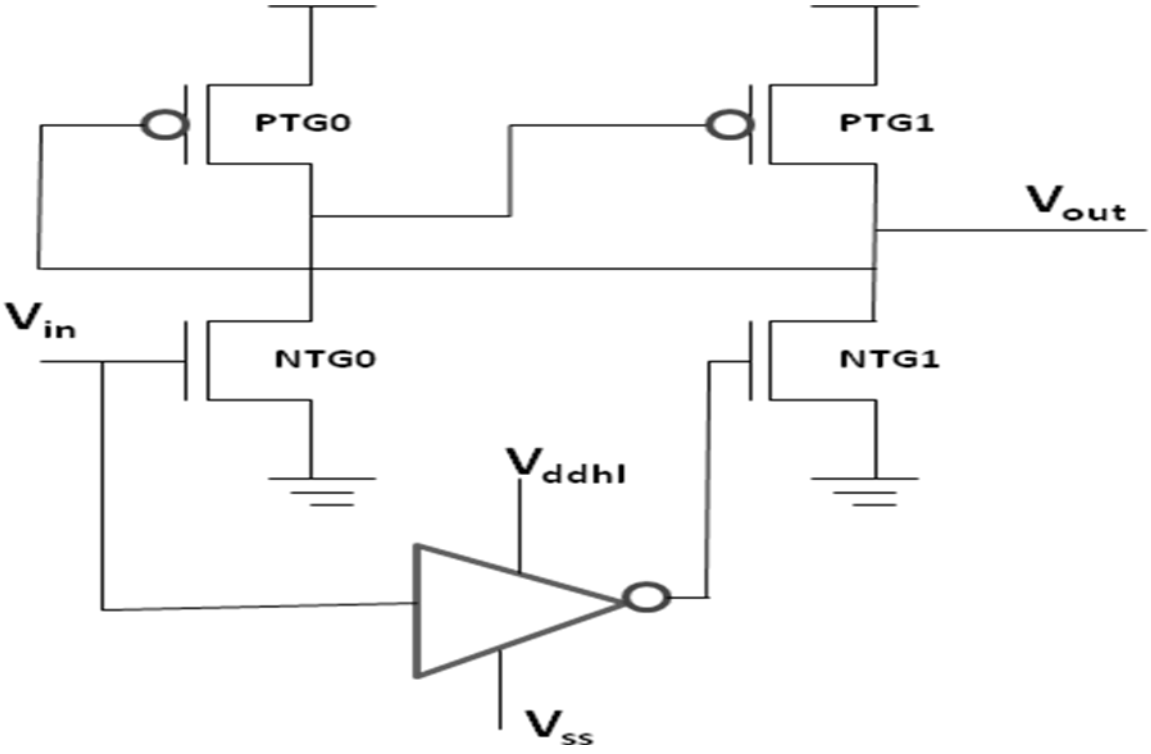


Figure 4.3 Level Shifter

The level shifter input is coming from the MUX. So than we design the MUX which is implemented using the transmission gates as shown in fig 4.4

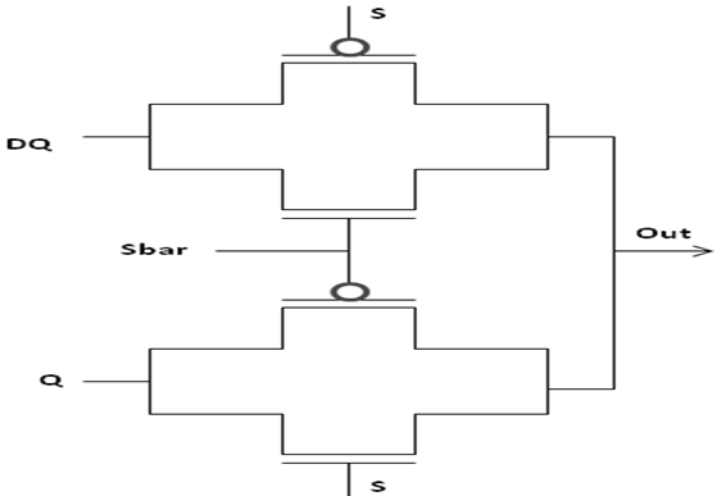


Figure 4.4 Multiplexer

We are using MUX into our design for selecting the particular mode of transmitter i.e. whether it is working in Synchronous mode or Asynchronous mode. Here I am meeting the specification of

Synchronous one but it can also be used for Asynchronous mode. Multiplexer Synchronous input is coming from the D_FF. The multiplexer is operating at 1.05V so nominal four terminal devices are preferred.

As Synchronous mode is following DDR Protocols i.e. data is coming at both the edges of clock or we can say that the data is coming at double rate 200 MHz while clock is coming at 100MHz. Data from the Core is coming as input to the D_FF so for capturing the data at both the edges we are using Tri-state logic for D_FF implementation. D_FF is also operating at 1.05V so this is also implemented using four terminal nominal devices. The Tri-state D_FF is shown in figure 4.5

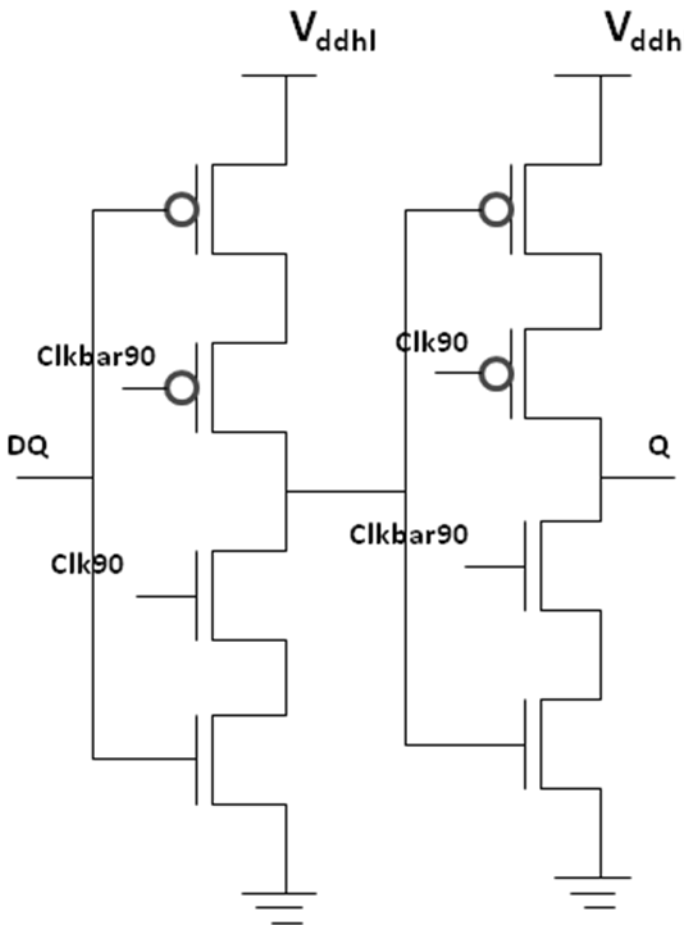


Figure 4.5 D FLIP FLOP

Here clock coming to the flip flop is 90 degree phase shifted as compared to the clock which is coming from the core i.e. Txclk90.The clock is coming from the master DLL which clock divider by a factor of two for the clock coming from the core, or we can say we have Txclk2x from the core, than DLL divide it by a factor of two and give Txclk.

4.2 Development of ONFI 2.3 DQS Transmitter

The structure of the DQS Transmitter (as shown in fig 4.6) is same as that of the Data Transmitter with the same basic components. Only difference is that Clock is coming in same phase as coming from core after master DLL and DQS is coming from the core in place of data. Rest is same.

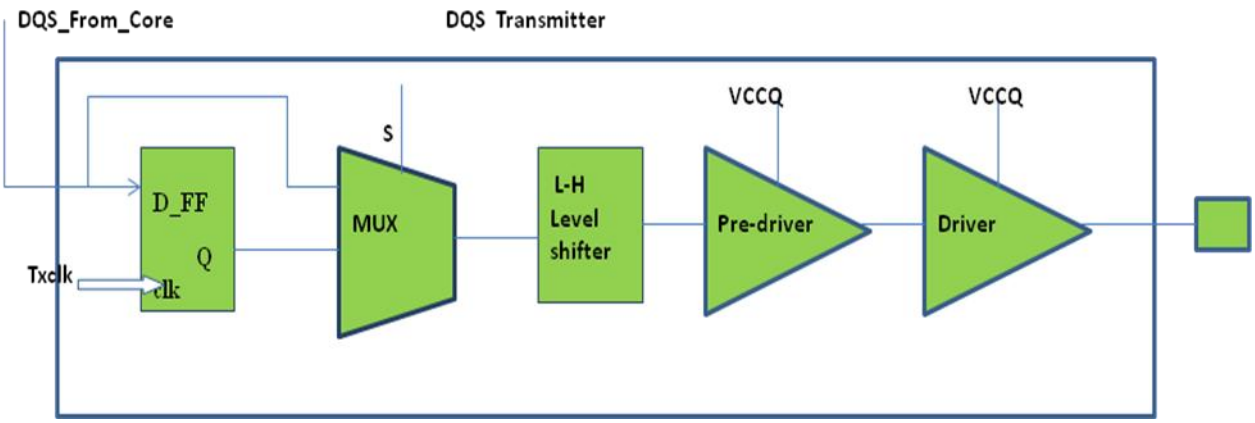


Figure 4.6 Top Level Block for DQS Transmitter

4.3 Development of ONFI 2.3 Data Receiver

The basic idea for designing of the data receiver (as shown in fig 4.7) stars from the designing of the differential pair with active current mirror and a realistic current source.

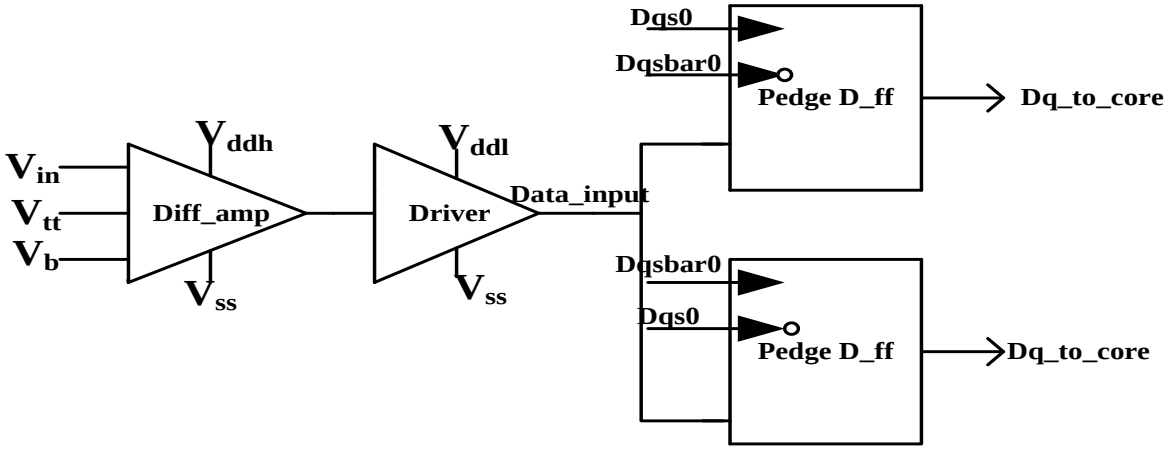


Figure 4.7 Top Level Block for Data Receiver

The one input (Vin as shown in fig 4.8) to the differential pair is taken from VIL(DC) TO VIH(DC) and other (Vtt as shown in schematic in fig 4.8) is taken VCCQ/2. As this pair is operating at 1.8V so we are using thick gate devices for its designing. All the devices are sized such that these are operating into saturation region. The differentia pair is shown below in Fig 4.8

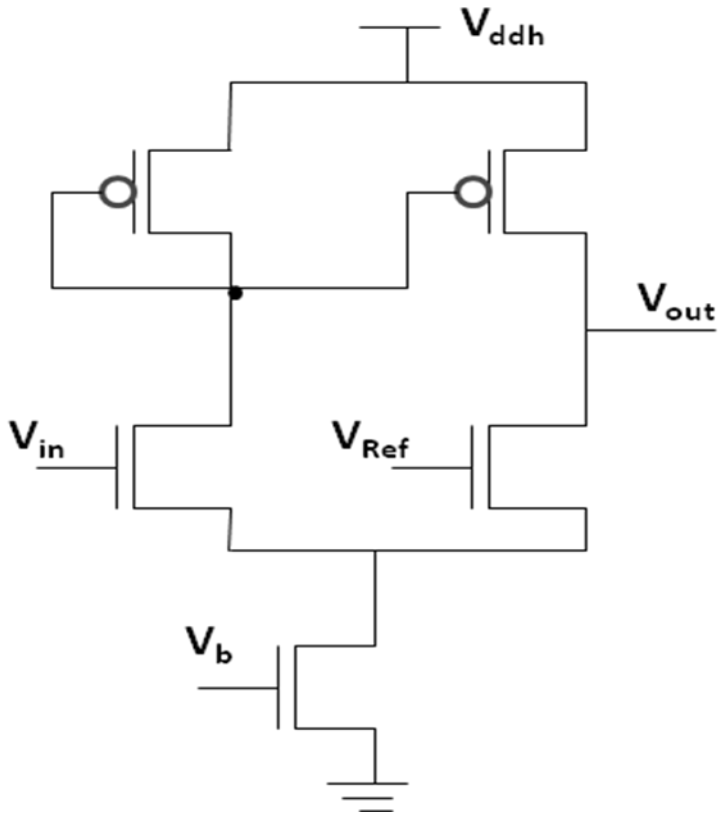


Figure 4.8 Differential Amplifier

The second stage of the data receiver is the Driver (as shown in fig 4.7) which shifts the output of differential amplifier to low level or we can say that we are using the driver into our design for shifting the voltage 1.8V to 1.05V.As this driver is converting a high voltage to low voltage so we are using thick gate devices for its implementation. The sizing of the transistor should be done such that it won't cause any capacitive coupling with the following stages.

The third stage contains two D flip flops which are capturing the data coming from the driver at double rate i.e. at 2x rate and give output at 1x rate. Both of these flip flops are implemented using transmission gates as shown in fig 4.9. Clock signal is coming from the DQS receiver or

we can say that DQS is acting as a clock signal in case of the data receiver. Flip flop is capturing the data at both the edges of the DQS. As it is clear from the data input cycle timing or from the signal description in chapter 1 that 8 bit data is coming so we need to distribute the DQS coming from the DQS receiver so that we can get all the 8 bits latched

together at one edge i.e. at rising edge and next 8 bit data gets latch at the falling edge of DQS. Here upper D_FF latching the data at rising edge of DQS and other at falling edge of data. So we are getting the data at 1x rate. Than we will combine these rates using multiplexer and provide data to core at 2x rate.

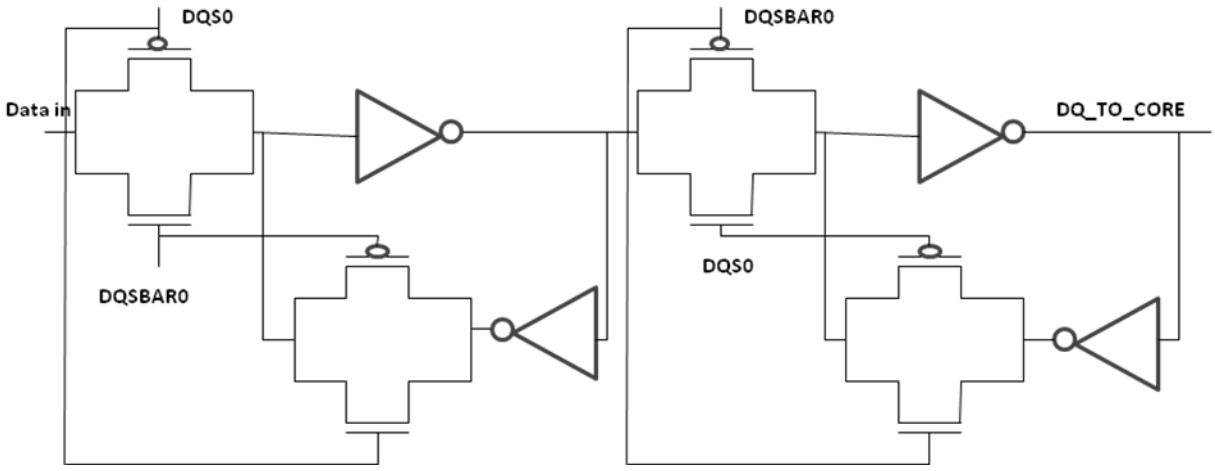


Figure 4.9 D Flip-Flop Diagram

I have design the DQS tree as shown in Fig 4.10 using topology Balanced Binary Tree of four stages. Firstly I estimated the area of my data transmitter and data receiver by drawing their layout. Through my estimation I came to know both transmitter and receiver together reside in area of 33.751 um x 29.139um. As this is mentioned in section 3.3 of chapter 3 that the spacing between two balls is 0.8mm so we have 0.8mm x0.8mm area available for placing our transmitter and receiver together. So from this we have estimated the wire length, so that we can estimate the capacitance or say loading for estimating the sizes of the drivers at each stage. I have used two repeaters between the DQS and the first stage of DQS tree for reducing delay.

4.4 Development of ONFI 2.3 DQS Receiver

DQS Receiver architecture have some elements same as that of the Data Receiver. The design of the DQS Receiver is shown in figure 4.11. From the figure it is clear that DQS Receiver have

differential amplifier for getting full swing, a driver for level shifting, than some buffers.

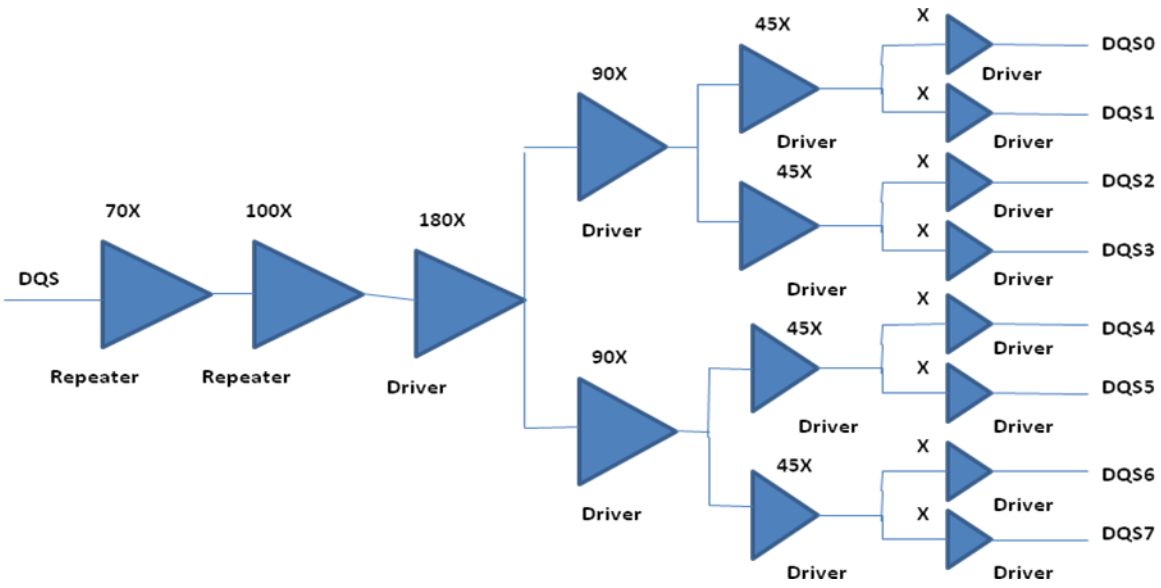


Figure 4.10 DQS Tree

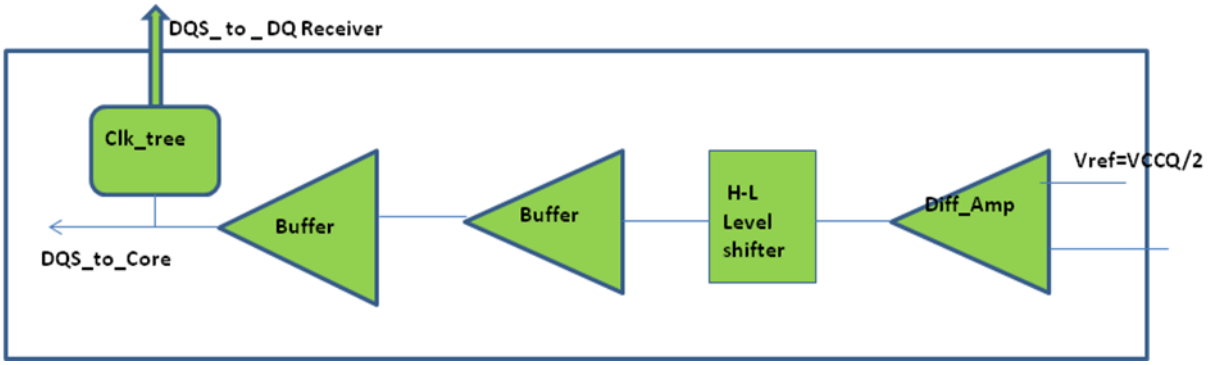


Figure 4.11 Top Level Block for DQS Receiver

The differential pair used for DQS Receiver has same specification as that of differential pair used for Data Receiver shown in figure 4.8. Driver is used for level shifting from 1.8V to 1.05V.This is also same as that which is used for Data Receiver.

Dqs_out coming from the DQS Receiver is acting as an input for the DQS Tree. So we need to insert some buffers between DQS Level shifter and DQS Tree input. I have inserted two buffers based on the FO4 concept

Chapter 5: Observation and Simulation Results

The complete I/O design is simulated in ADE (Analog Design Environment) using Spectre as a simulator. Two specification output slew rate and input slew rate are used for the verification of my design.

Output Slew Rate (Rising) = ((VOH (AC) – VOL (DC)) / tRISE

Output Slew Rate (Falling) = ((VOH (DC) – VOL (AC)) / Tfall

5.1 Data Transmitter Simulation Results and Observation

Here Clock to the Data transmitter is in 90 degree phase with respect to the clock coming from the core and after the division by a factor 2 i.e. if T2xCLK IS coming from the core TxCLK is going inside the data transmitter. Data is coming in the same phase as it is coming from the core. The data transmitter output (dq_out) results are shown in the following tables with comparison to what was required and what is obtained, of respective to the waveforms shown in figure 5.1

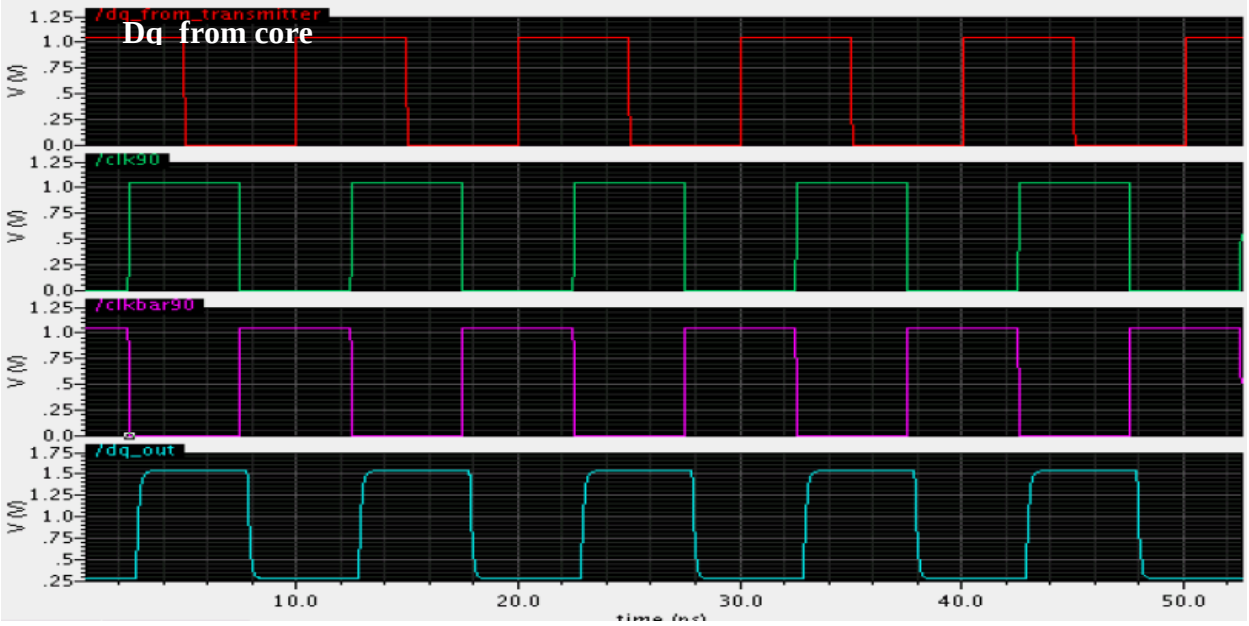


Figure 5.1 Data Transmitter output Waveforms

Description		Rise Time Obtained	Fall time Obtained	O/P Obtained	
				Min	Max
18 Ohm	Typ	254.8ps	253.9ps	298.5mV	1.551V
	Fast	181.5ps	199.3ps	315mV	1.79V
	Slow	370.4ps	331.31ps	299.5mV	1.39V

25 Ohm	Typ	342.5ps	340.3ps	330.17mV	1.5045V
	Fast	233ps	253.9ps	335.76mV	1.6885V
	Slow	631.2ps	508.05ps	331.35mV	1.3705V
35 Ohm	Typ	360.9ps	363.2ps	336.52mV	1.498V
	Fast	244.1ps	268.52ps	342.67mV	1.6824V
	Slow	778.2ps	634.12ps	337.34mV	1.3635V
50 Ohm	Typ	391.3ps	394.5ps	342.05mV	1.4865V
	Fast	259.8ps	283.18ps	348.49mV	1.6715V
	Slow	909.2ps	627.927ps	337.34mV	1.3612V

TABLE 14 Transmitter Output Specification Result

Description	**O/P Slew Rate Required**		**O/P Slew Rate Obtained**			**Units**
	Min	**max**	**TYP**	**Fast**	**Slow**	
18 ohm	1	5.5	Rising=3.53	Rising=5.37	Rising=2.29	V/ns
			Falling=3.54	Falling=4.89	Falling=2.56	
25 ohm	0.85	5.0	Rising=2.62	Rising=4.29	Rising=1.34	V/ns
			Falling=2.64	Falling=3.84	Falling=1.67	
35 ohm	0.75	4.0	Rising=2.49	Rising=3.48	Rising=1.09	V/ns
			Falling=2.47	Falling=3.16	Falling=1.34	
50 ohm	0.65	4.0	Rising=2.3	Rising=3.75	Rising=0.93	V/ns
			Falling=2.28	Falling=3.44	Falling=1.35	

Table 15 Comparison between O/P Slew Rate Required and Obtained

Observation: For tuning of Data Transmitter for 50 Ohm driving strength one extra finger is used for pull down network and 3 extra fingers are used for pull-up network.

Description	Typical
Duty Cycle of CLK w.r.t. Vth 0.525 V	49.51%
Duty Cycle of Dqs w.r.t. Vth 0.525 V	49.50%
Duty Cycle of Dqs_out w.r.t. vth 0.9 V	49.47%
Overshoot of Dqs_out (0 to 1.8 V)	0

Table 16 DQ Transmitter Output Specification Results for 18 Ohm

5.2 DQS Transmitter Results

The waveforms for DQS transmitter output is shown below in figure 5.2. Here clock is coming in the same phase as was coming from the core but this clock is obtained by dividing the clock coming from the core by a factor 2 i.e. clock coming from the core is T2Xclk but it is going as Txclk to the DQS transmitter.

O/P Slew Rate Required		O/P Slew Rate Obtained		
Min	**Max**	**Typ**	**Fast**	**Slow**
1 V/ns	5.5 V/ns	Rising = 3.57 V/ns	5.31V/ns	2.28 V/ns
		Falling = 3.55 V/ns	4.86 V/ns	2.60 V/ns

Table 17 Comparison between O/P Slew Rate Required and Obtained for 18 Ohm

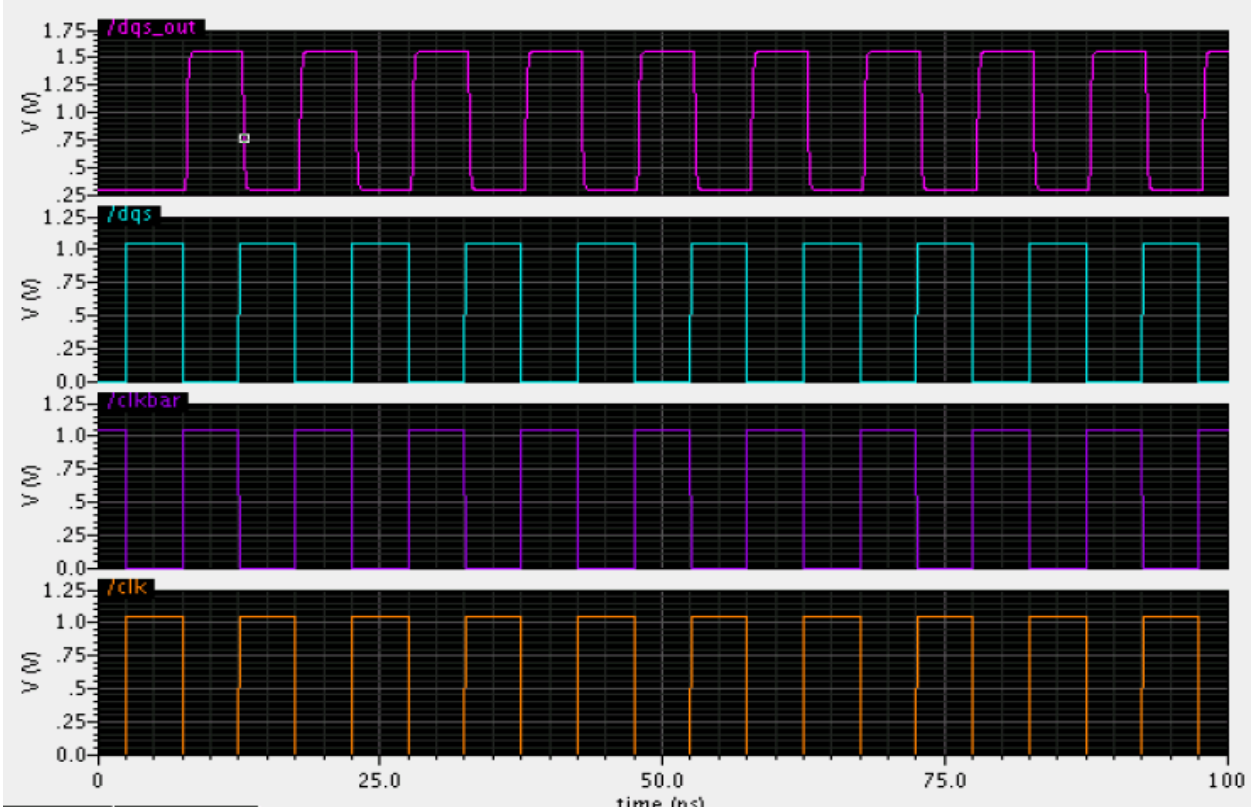


Figure 5.2 DQS Transmitter Output Waveforms

Description	Typ	Fast	Slow	Units
Rise Time	252.1	183.3	372.4	Psec
Fall Time	253.5	200.47	326.26	Psec
O/P max value	1.551	1.731	1.418	V
O/P min value	298.5	303.5	300.2	mV

Table 18 DQS Transmitter Output Specification Results for 18 Ohm

Description	Typical
Duty Cycle of CLK w.r.t. Vth 0.525 V	49.51%

Duty Cycle of Dqs w.r.t. Vth 0.525 V	49.51%
Duty Cycle of Dqs_out w.r.t. vth 0.9 V	50.57%
Overshoot of Dqs_out (0 to 1.8 V)	0

Table 19 DQS Transmitter Output Specification Results for 18 Ohm

5.2 DQS Receiver output Results

Here are the Result of the DQS Receiver in tabular form. We will view its output waveforms for the worst case in the following section i.e. Complete I/O Results.

Description	**For I/P Slew Rate of 4.5 V/ns**			**For I/P Slew Rate of 0.5 V/ns**			**Units**
	Typ	**Slow**	**Fast**	**Typ**	**Slow**	**Fast**	
Rise Time	32.64	41.02	32.08	49.31	68.63	40.3	Psec
Fall Time	16.19	17.74	15.47	51.49	61.8	49.23	Psec
O/P Max	1.04998	0.9448	1.5489	1.0496	0.94489	1.1545	V
O/P Min	13.27u	75.35u	7.08884u	2.98m	557.85u	3.45m	V

Table 20 DQS Receiver Output Specification Results

5.2 ONFI 2.3 Interface I/O output Results

Figure 5.3 shows the block diagram of ONFI 2.3 Interface I/O and figure 5.4 shows the Output waveforms of the Data receiver for the worst case input i.e. I have obtain the output results for data receiver by simulating the figure 5.3, provided that both Data

Transmitter and DQS Transmitter should be disable. These can be disabled by disabling all the enable pins shown in figure 5.3. I have taken the worst case input for both Data Receiver and DQS Receiver i.e. which satisfies all the requirements for Input Slew Rate of 0.5 V/ns. For evaluating whether Data Receiver output is correct or not, firstly I have estimated the Set Up Time(21psec) and Hold Time(24.12psec) of Data Receiver D_FF. Than I estimated the JITTER (50psec) by evaluating the delay between the DQS and DQS0,DQS1,DQS2,DQS3,DQS4,DQS5,DQS6.DQS7 by doing simulation for all the +-10% voltages. Than I just checked whether input is captured by flip flop at both the edges of DQS0 or not and as this is clear from the waveform data is being captured at both the edges. You can evaluate this by viewing the waveform results for DQS0 and DQSBAR0.

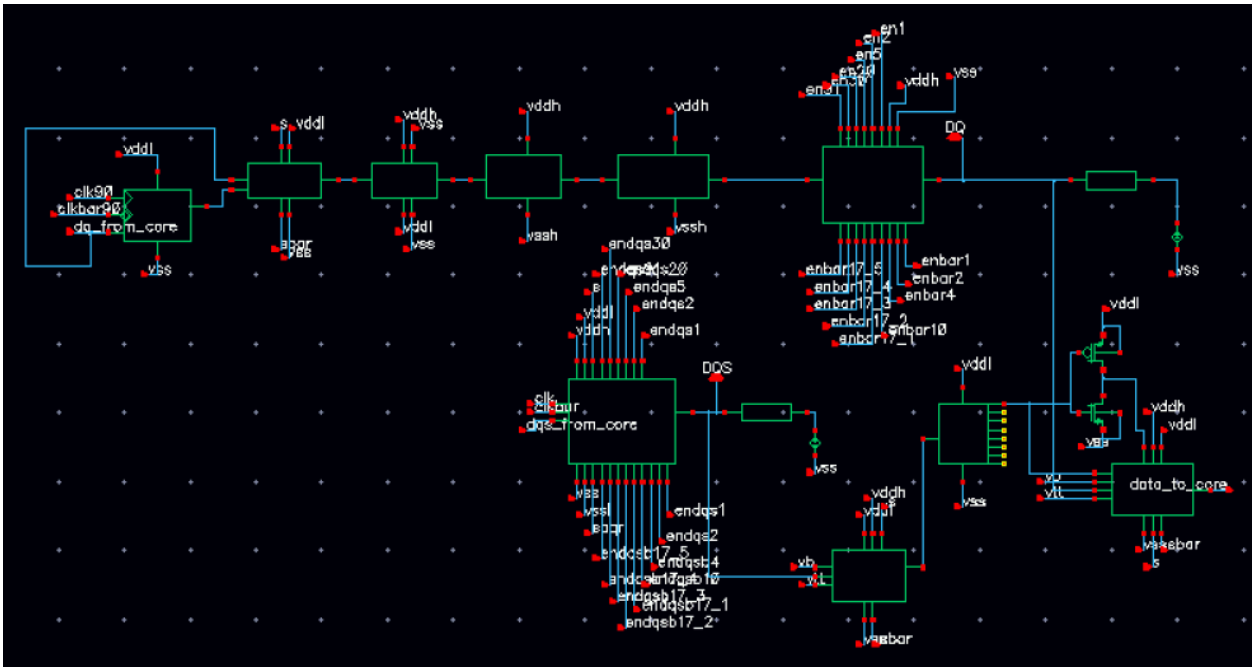


Figure 5.3 Schematic Diagram of Complete I/O

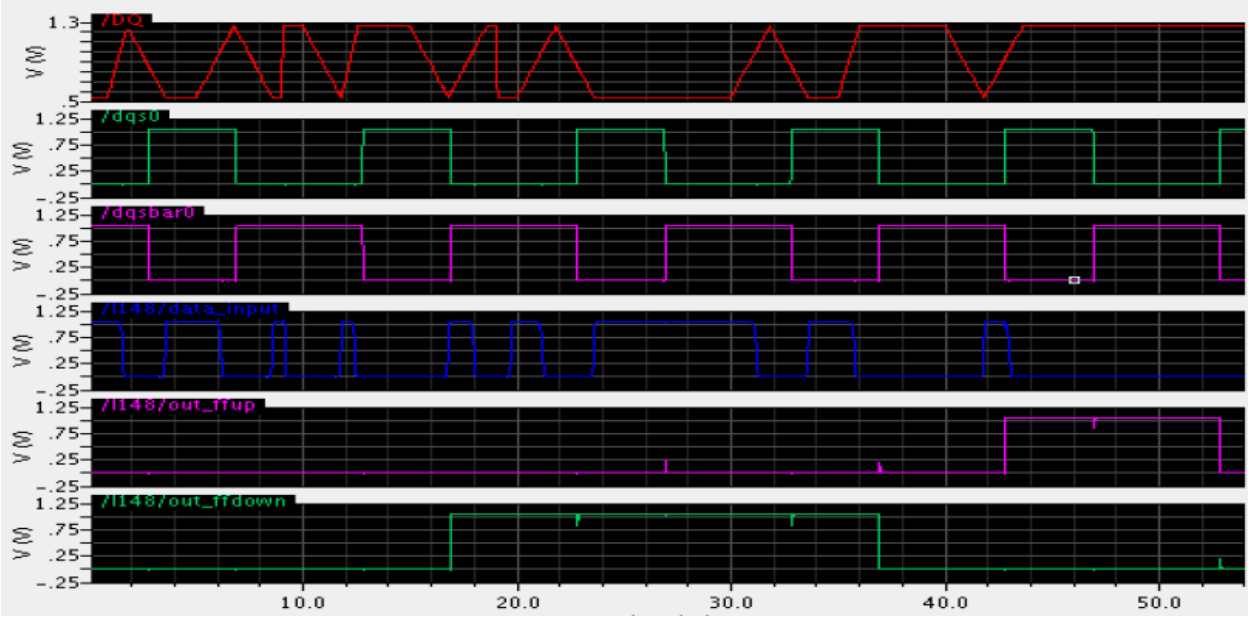


Figure 5.4 Output Waveforms of Data Receiver

Chapter 6: Conclusion & Future Scope

The Data Transmitter and DQS Transmitter Output Slew rate for rising and falling edges are falling in the range specified in the specification sheet mentioned in references. Both data transmitter and DQS transmitter are working fine. Data receiver is tested for the worst case input with the DQS coming from DQS receiver after following the given Input Slew Rate specifications. Data receiver is capturing the data for the worst case input. So my complete I/O design is working perfectly.

So all in all we can say that NAND market is driving lowest cost, highest density NAND which results in challenges for NAND performance in terms of slower NAND array timing , increase in ECC requirements. So I concluded that these NAND array challenges can be overcome by designing a EZ NAND interface working at data rate of 200 MHz. The future scope for this work is that after this, by just verifying the design using Monte Carlo Simulations the Design will be ready for fabrication.

References

[1] Dean Nobunaga et.all “A 50nm 8Gb Flash Memory with 100MB/s Program Throughput and 200MB/s DDR Interface” © 2008 IEEE International Solid-State Circuits Conference.

[2] Yongsoo Joo et.all “Energy and Performance Optimization of Demand Paging With OneNAND Flash” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions.

[3] Chuan-Sheng Lin; Kuang-Yuan Chen; Yu-Hsian Wang; Lan-Rong Dung; “A NAND Flash Memory Controller for SD/MMC Flash Memory Card” Electronics, Circuits and Systems, 2006. ICECS '06. 13th IEEE International Conference.

[4] Johns & Martin. Analog Integrated Circuit Design. Wiley & Sons, Canada. © 1997. pgs. 228-231

[5] Samsung Electronics Co., “NAND Flash Memory & Smart Media Data Book”, 2002.

[6] Behzad Razavi. Design of CMOS Integrated Circuits. McGRAW- HILL Higher Education, International edition 2001.

[7] www.onfi.org

[8] http://onfi.org/about/

[9] http://onfi.org/presentations/

[10] http://onfi.org/specifications/

[11]http://maltielconsulting.com/NAND_vs_NOR_Flash_Memory_Technology_OverviewRead Write_Erase_speed_for_SLC_MLC_semiconductor_consulting_expert.pdf