Università degli Studi di Napoli "Federico II"

Facoltà di Scienze MM.FF.NN.

Tesi di dottorato di ricerca in
Tecnologie Innovative per Materiali, Sensori ed Imaging
XVII ciclo aggregato al XIX

Development of a FPGA based test system for Double Data Rate high speed memory devices.

Coordinatore:
Prof. Ruggero Vaglio

Candidato:
Adele Di Cicco

Contents

Introduction

Semiconductor memories are widely considered to be one of the most important types of microelectronic components in modern digital systems [1]. It is reported that memories represent about 30% of the worldwide semiconductor market [2]. The growing need for storage in computer, communications, consumer, and network applications is driving the continuous innovation of various semiconductor memory technologies. Bigger and faster memories are always desirable due to our insatiable appetites for voluminous transmission and storage of data in these applications, i.e., the continuing technology innovation is likely to increase the market share of commodity and embedded memories in the future. The Double Data Rate II (DDR II) SDRAM memory, which is the last generation of Dynamic Ram, is the most widely used type of memory in the market today and its data rate is expected to move from currently 400MHz to 667MHz and 800MHz in the near future. It is an appropriate memory solution for system ranging from workstations and servers to embedded communication system such as graphics, cache and main memory for Personal Computer. The JEDEC (Joint Electron Device Engineering Council)[1] organization has helped the memory industry by creating memory specifications in the form of JEDEC standards. This standards cover physical characteristics, electrical signal, register definitions, functional operations, memory protocol, etc. The exponential increase in the integration density and complex manufacturing steps have made the memory more vulnerable to physical defects than the other logic circuits. The fast development of memory devices and the strong market competition have increased the standards of these produced devices. The increased demand on reliability has, in turn, stressed the importance of memory testing technique. The chapter I starts with an overview of semiconductor memory devices, showing their evolution, and describing the improvements of each memory devices over its predecessor.

[1]JEDEC is a no-profit organization with members from memory, computer and test equipment manufacturers.

In this chapter is stressed the need to introduce an efficient testing process to solve the problems related to the even increasing density. The traditional ways to test semiconductor memories, realized to detect physical defects related to the high density are introduced. On the other hand, at the higher frequency, such as that of the DDR II, the *signal integrity* becomes an important issue to be considered for reliability of the memory operations. An introduction to the contribution to this problem of this Ph.D project is presented in the chapter I.

The chapter II describes in more details signal integrity issues related to high frequency systems such as high speed memory devices, starting from the Jitter definition and the description of the major sources of jitter, through the classification of the jitter, and ending with the method used to observe and study it.

The testing process is responsible for a large portion of the cost of these memories, standing now at 40% and gradually rising with each new generation. Companies usually develop the required memory tests in an ad hoc, relying heavily on an expensive combination of experience and statistics to construct the best test approach, the price of which is ultimately paid by the end consumer. Memory testing is expensive because of the high cost of the test equipment (a production memory tester costs more than 500.000 Dollars), a cost that has to be distributed over all the produced chips. However the equipments available on the market are not able to perform both functional tests and signal integrity analysis. The study of the signal integrity, further increases the production time and costs. In this thesis project, a new alternative approach to the development of an industrial memory testing is proposed. It is more systematic and less expensive than the currently prevalent test approaches. The new approach makes it possible to enhance memory tests in many different manufacturing stages, starting from the initial test application stage where silicon is manufactured, and ending the memory ramp-up stage where products are shipped to the customer. The tester realized reduce the test time and costs being based on the use of a last generation Field Programmable Gate Arrays (*FPGA*), which implements 90% of the needed functionalities. Chapter III deals with the description of the overall characteristic of the realized tester, whose design, production and testing represent the innovative issues of this Ph.D project. The realized tester, is able to perform:

- the *functional test* allowing the fast detection of physical defects,
- the *signal integrity analysis*, implementing on each memory channel, the

function of a sampling oscilloscope. The advantage of this feature is to be able to perform complex analysis without the use of an external and expensive equipment. The complex analysis include the eye diagram opening, rise and fall time, setup and hold time measurements,

- the *parametric test* which is aimed at the validation of electrical parameters (leakage currents, voltage level, rise and fall time, etc.) according to the specifications.

Chapter IV deals with the description of the debug of the functional tester and the debug of the signal integrity analyzer. The functional tester capability to detect physical defects has been verified comparing the test results with those obtained the traditional equipment. The signal integrity analyzer performances have been evaluated comparing the eye diagrams (rising edge, falling edge, eye opening, etc.) obtained with the customized-tester with those obtained by the use of last generation high bandwidth oscilloscope.

Chapter 1

High Data Rate Memory Devices

1.1 Introduction

Technical advancements such as high-speed processors, multimedia application with high resolution moving pictures, and serial data links with Gb/s data rates are driving the need for higher bandwidth memory system. The growing need for storage in computer, communications, consumers and network applications is the driving force in the advancements of DRAM technology. Mainstream DRAMs have evolved over the years through several technology enhancements such as SDRAM (Synchronous Dynamic Random Access Memory), DDR (Double Data Rate) SDRAM, DDR II (Double Data Rate II) SDRAM, DDR III (Double Data Rate III) SDRAM. Fig.1.1 shows the data transfer rate and DRAM architecture trends of main memory for server and high-end PC. Currently, Double Data Rate II is dominating the market for main memory and its data rate is expected to move from currently 400MHz to 667MHz and 800MHz in the near future.

The exponential increase in the integration density and the complex manufacturing steps have made the memory more vulnerable to physical defects than other logic circuits. This attitude have made memory testing a significantly important task to perform. Due to the increasing complexity of the memory chips, their testing is quickly becoming a difficult issue, which represent the real bottleneck of the entire production process.

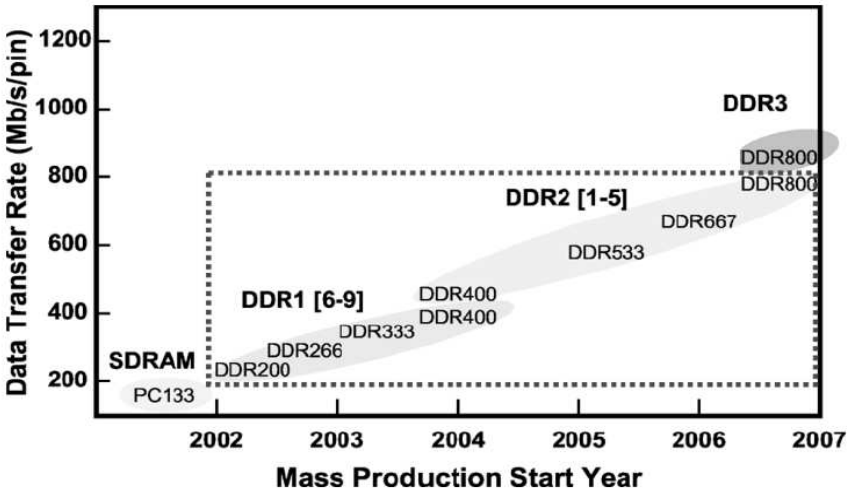

Figure 1.1: Data transfer rate and DRAM architecture trends of high-end PC and servers.

This chapter describes the traditional ways to test semiconductor memories, realized to detect physical defects and anomaly in functional behavior.

At the higher frequency, such as that of DDR II, signal integrity becomes an important issue for reliable memory operation. In this chapter there is an introduction to the work done in this thesis, which is aimed at the analysis of the signal integrity with faster, lower cost and more compact solution than the testing environments today available on the market.

1.2 Random Access Memory

The name RAM stands for a memory device in which cells may be accessed at random to perform a read or a write operation. Depending on the internal architecture and the actual memory cell structure, RAMs may be divided into:

- dynamic RAMs (DRAMs),
- static RAMs (SRAMs).

A simple block diagram of a RAM is given in fig.1.2. Three main input are shown: a read/write (R/W) switch to discriminate the type of operation

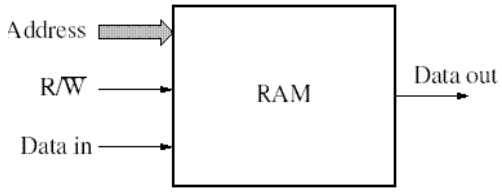

Figure 1.2: Block diagrams of a RAM .

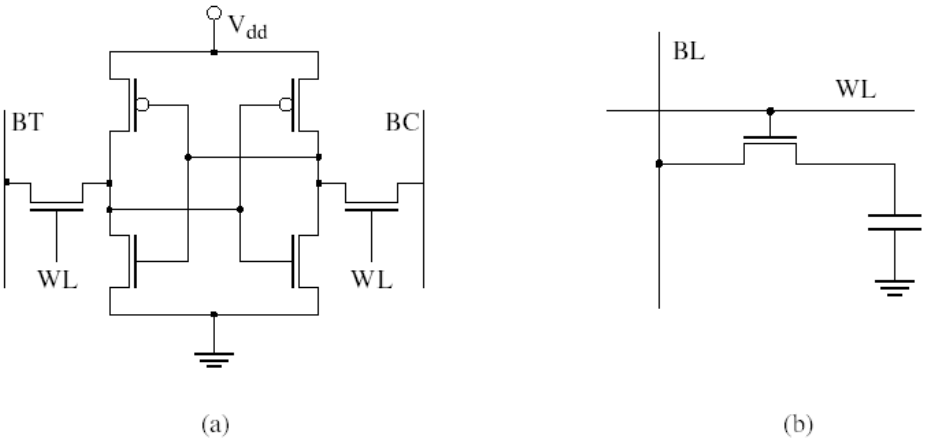

Figure 1.3: Electrical structure of (a) SRAM and (b) DRAM core cells.

performed, an address input which identifies the cell to be accessed and a data input bus that supplies the data to be written in case of write operation. A RAM also has a data output bus to be used on a read operation to forward data from the addressed cell to the outside world. In principle, both DRAMs a SRAMs share this same general interface, but the specific implementation is different and mainly depends on the targeted application of the memory.

A typical structure of an SRAM cell is shown in fig.1.3(a). The cell is constructed using six transistors, four of which are of one transistor type (NMOS) while the other two are of another type (PMOS). The *word line* (WL) performs the address selection function for the cell. The *true bit line* (BT) and *complement bit line* (BC) serve both as the data input line and the data output line for the cell at the same time. The selection between

performing a read or a write operation is fulfilled by other memory parts external to the cell. The operation of the cell is based on the fact that SRAM cells are bistable electrical elements (i.e., circuits that have two stable states). Each state is used to represent a given logical level. Once a cell is forced into one of the two states, it will stay in it as long as the memory is connected to the power supply; the name "static RAM" refers to this property.

The electrical structure of a DRAM cell is shown in fig.1.3(b). The cell is constructed using one transistor and one capacitor. The *WL* performs the address selection, while the bit line (BLs) are used as both the data input and data output lines. The selection between read and write operations is performed by other parts of the memory. The DRAMs are constructed of simple capacitive elements that store electrical charges to represents a given logical level. Inherently, DRAM cells suffer from gradual charge loss, as a result of phenomenon known as *transistor leakage currents*, which causes a cell to lose its charge gradually. In order to help cells keep their state, it is necessary for DRAMs to rewrite, or *refresh*, the already stored data bits from time to time before the cells lose their charge completely. The "dynamic RAM" refers to the fact that the data stored in the DRAM cell may change spontaneously after a given period of time.

Both DRAMs and SRAMs are called volatile memories because they can only keep their data content if they stay connected to the power supply. A closer look at the two RAM structures reveals that SRAMs store their data actively by pulling their nodes to high or low voltage levels, while DRAMs store their data in capacitive elements that take time to charge up and discharge. Therefore, SRAMs have a much higher performance than DRAMs, and this is the reason they are used as the first level memory (or *cache memory*) directly supporting the central processing unit (CPU) in a microprocessor. The main advantage that DRAMs have over SRAMs is in their density. Fig.1.3 clearly shows that DRAM cells are simple, compact elements that achieve much higher chip densities than their SRAM counterparts, which also makes them much cheaper. This cost difference is so important that it outweighs all other aspects in most applications.

DRAMs core architecture consists of memory cells organized into a two-dimensional array of row and columns (fig.1.4).The address bus is multiplexed between row and column components. The multiplexed address bus uses two control signals -the row and column address strobe signals, $\overline{RAS}$ and $\overline{CAS}$, respectively- which cause the DRAM to latch the address components. Additional DRAM control signals include $\overline{WE}$ (Write enable) for selecting write

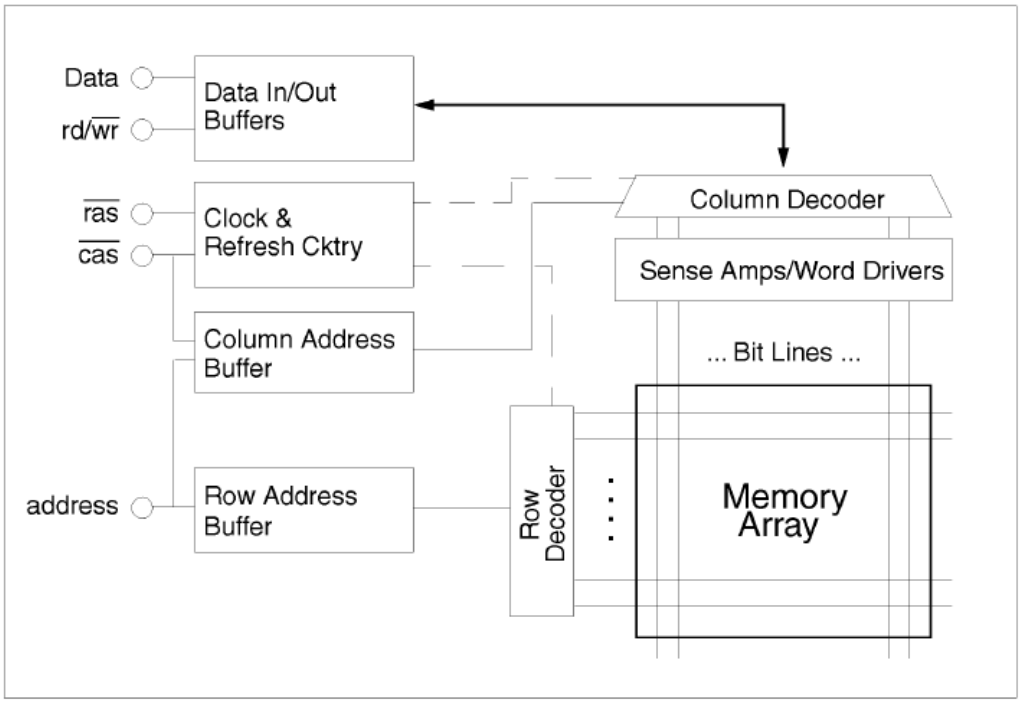

Figure 1.4: Conventional DRAM block diagram.

or read operation, $\overline{CS}$ for selecting the DRAM, and $\overline{OE}$ (Output Enable). The row address causes a complete row in the memory array to propagate down the bit lines to sense amps. The column address selects the appropriate data subset from the sense amps and causes it to be driven to the output pins. DRAM reads are destructive, meaning the data in the row of memory cells are destroyed in the read operation. Therefore, the row data need to be written back into the same row after the completion of a read or write operation on a row. This operation is called *Precharge* and is the last operation on a row. It must be done before accessing a new row and is referred to as closing an open row. A DRAM row is called a memory page and once the row is opened it is possible to access multiple sequential or different column addresses in the row. This increases memory access speed and reduces memory latency by not having to resend the row address to the DRAM when accessing memory cells in the same memory page.

The early DRAMs read cycle had four step. First, $\overline{RAS}$ goes low with a row address on the address bus. Secondly, $\overline{CAS}$ goes low with a column address on the address bus. Third , $\overline{OE}$ goes low and read data appears on the data bus. The time from the first step to the third step when the data is available on data bus is called latency. The last step is $\overline{RAS}$, $\overline{CAS}$, $\overline{OE}$ going high (inactive) and waiting for the internal precharge operation to complete restoration of the row data after the destructive read. Signal timing of the above signals is related to the edges and is asynchronous. There is no synchronous clock operation.

1.3 Synchronous DRAM

In the Synchronous DRAM the memory operations are synchronized with the functionality of the system by using a single system clock. The type of memory command is determined by the state of the control signal at the rising edge of the SDRAM clock. The clock is used to drive an internal finite state machine that pipelines incoming commands. This allows the chip to have a more complex pattern of operation than DRAM, which does not have synchronizing control circuits. Pipelining means that the chip can accepts a new command before it has finished processing the previous one. In pipelined write, the write command can be immediately followed by another command without waiting for the data to be written to the memory array. In a pipelined read, the requested data appears a fixed number of clock pulses after the read command. It is no necessary to wait for the data to appear before the next command. This delay is called the *latency*, and is an important parameter which will be described in the following paragraph.

1.4 DDR I SDRAM

The next stage of evolutionary migration from standard DRAMs is DDR I (Double Data Rate I) SDRAM which dramatically increases DRAM bus speed by activating data output on both the rising and falling edges (*double pumped*) of the system clock rather than on just the rising edge.

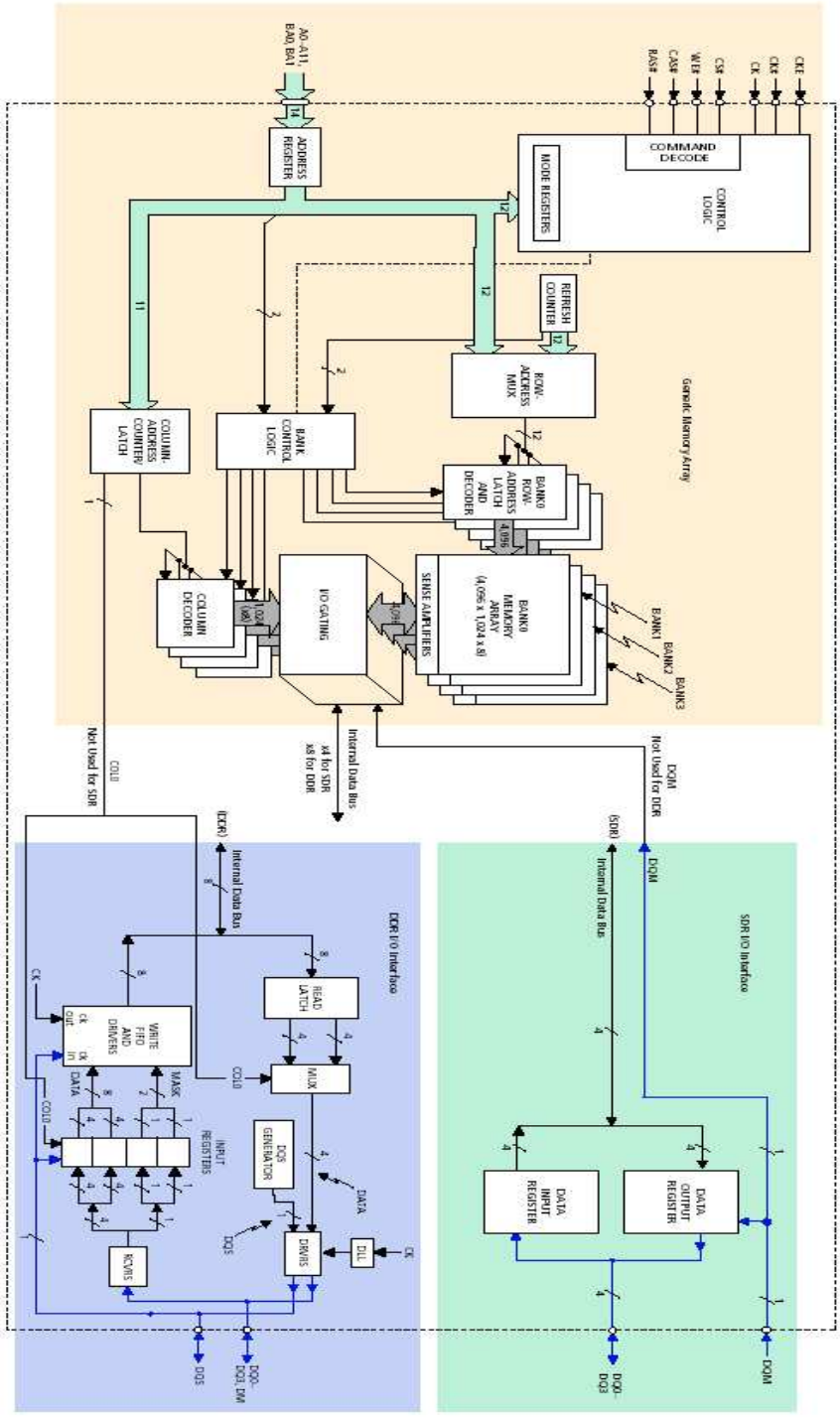

Figure 1.5: Functional block diagram with SDR and DDR interface.

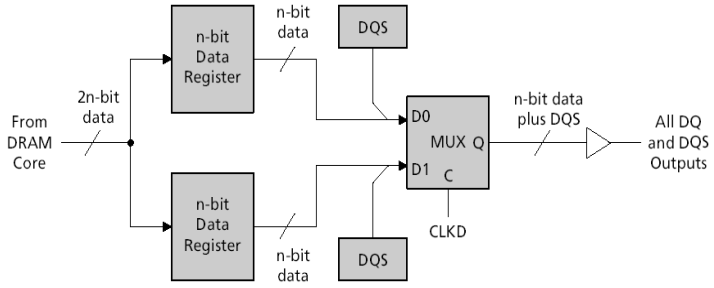

Figure 1.6: Simplified block diagram of 2n-prefetch read.

The main functional differences between DDR and Single Data Rate (SDR) are:

- the double data rate/2n-prefetch architecture designed to transfer two data words per clock cycle at I/O pins.
- the strobe based data bus to provide high speed signal integrity.
- Stub Series Terminated Logic 2 interface with differential inputs and clocks.

Fig.1.5 shows the difference between a DDR I and SDR (Single Data Rate) SDRAM functional block [3] . It reveals that the memory core is essentially the same. Both have an identical addressing and command control interface, a four-bank memory array (to provide multiple interleaved memory access) and both incorporate the same refresh requirements. The double data rate memory utilizes a differential pair for the system clock and therefore will have both a true and complementary clock signals. The main differences are found in the data interface.

First, DDR adds phase-lock-loop circuitry to existing SDRAM designs. PLL technology enables tighter synchronization of data output to the device's clock, which eliminates wasted clock cycles, makes the process more efficient, and improves performance.

The SDR data interface is a fully synchronous design where the data is only captured on the positive clock edge. The internal bus is the same width as the

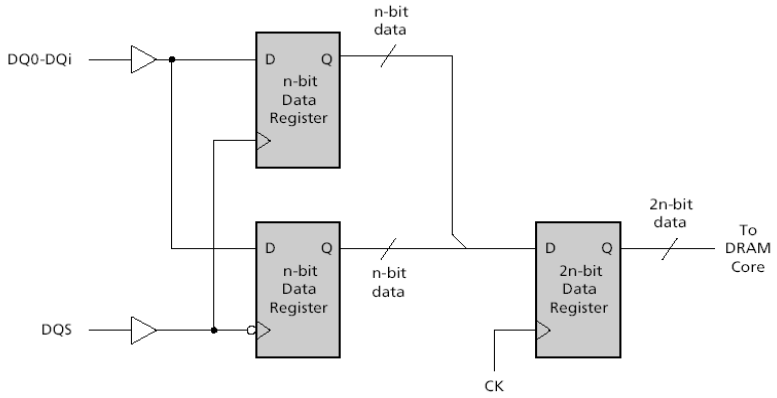

Figure 1.7: Simplified block diagram of 2n-prefetch write.

external data bus and the data latches into the internal memory sequentially as it passes through the I/O buffers.

The DDR I memory data is a true synchronous design, where the data is captured twice per clock cycle with a bidirectional data strobe. This architecture employs a *2n prefetch architecture*, where the internal data bus is twice the width of the external bus. This allows the internal memory cell to pass data to I/O buffer pairs.

Fig.1.6 show a simplified block diagram of 2n-prefetch READ. For each single read access cycle internal to the device, two external data words are provided. Similarly, two external data words written to the device are internally combined and written in one internal access (as shown in fig.1.7). The DDR command bus consist of a clock enable, chip select, row and column addresses, bank address and write enable. Commands are entered on the positive edges of clock and data occurs on both positive and negative edges of the clock.

In a SDR system, data output are referenced to a common, free-running system clock. Instead, the DDR memory bus has one clock for the control signals and 18 strobes for the data groups. The data strobes are non-free-running signal driven by the device which is driving the data signals (the controller for WRITEs, the DRAMs for READs). At the DRAM device level, for READs, the data strobe (DQS) signal are effectively additional data output (DQ) with a predetermined pattern; for WRITEs, the strobe signals are used

as clocks to capture the corresponding input data. At the board level, the strobe signals have identical loading to data signals and should be routed similarly.

For READs, the data strobe signals are edge-aligned with the data signals, meaning that all data and data strobes are clocked out of the device by the same internal clock signal, and all will be presented at the outputs at nominally the same time. The controller will internally delay the received strobe to the center of the received data eye.

For WRITEs, the controller must provide the data strobes center-aligned relative to data. That is, strobe transitions occur nominally 90 degrees (relative to the clock frequency) out of phase with data transitions.

READs and WRITEs use a different alignment in order to avoid replicating the delay circuitry throughout the DRAM.

Previous SDR memory technology used LVTTL and a fixed voltage level for signal interface. DDR I SDRAM utilizes differential inputs as a reference voltage for all interface signals. This interface is called SSTL2, which stand for stub series terminated logic for 2.5volts. SSTL2 is an industry standard defined by JEDEC document. Although some DRAMs will support a reduced drive output, most will comply with the SSTL2 Class II drive levels.

Benefits to the SSTL2 interface include symmetrical low and high levels, improved signal integrity, and better noise immunity, as the input levels track minor variations in the supply voltage.

1.5 DDR II SDRAM

In order to avoid the problem associated with increased data rate, JEDEC has standardized DDR-II SDRAM, targeting 800Mb/s at much lower supply voltage than DDR-SDRAM. DDR-II is specified to operate with 1.8V supply voltage while DDR SDRAM use 2.5V supply voltage. This voltage scaling enhancement has the potential to reduce overall power requirements for the memory system. Another benefit of the lower operating voltages is the lower logic voltage swings. For the same slew rate, the reduced voltage swings increase logic transition speeds to support faster clock rates. In addition, the data strobe can be programmed to be a differential signal. Using differential data strobe signal reduces noise, crosstalk, dynamic power consumption and EMI (Electromagnet Interference) and increases noise margin.

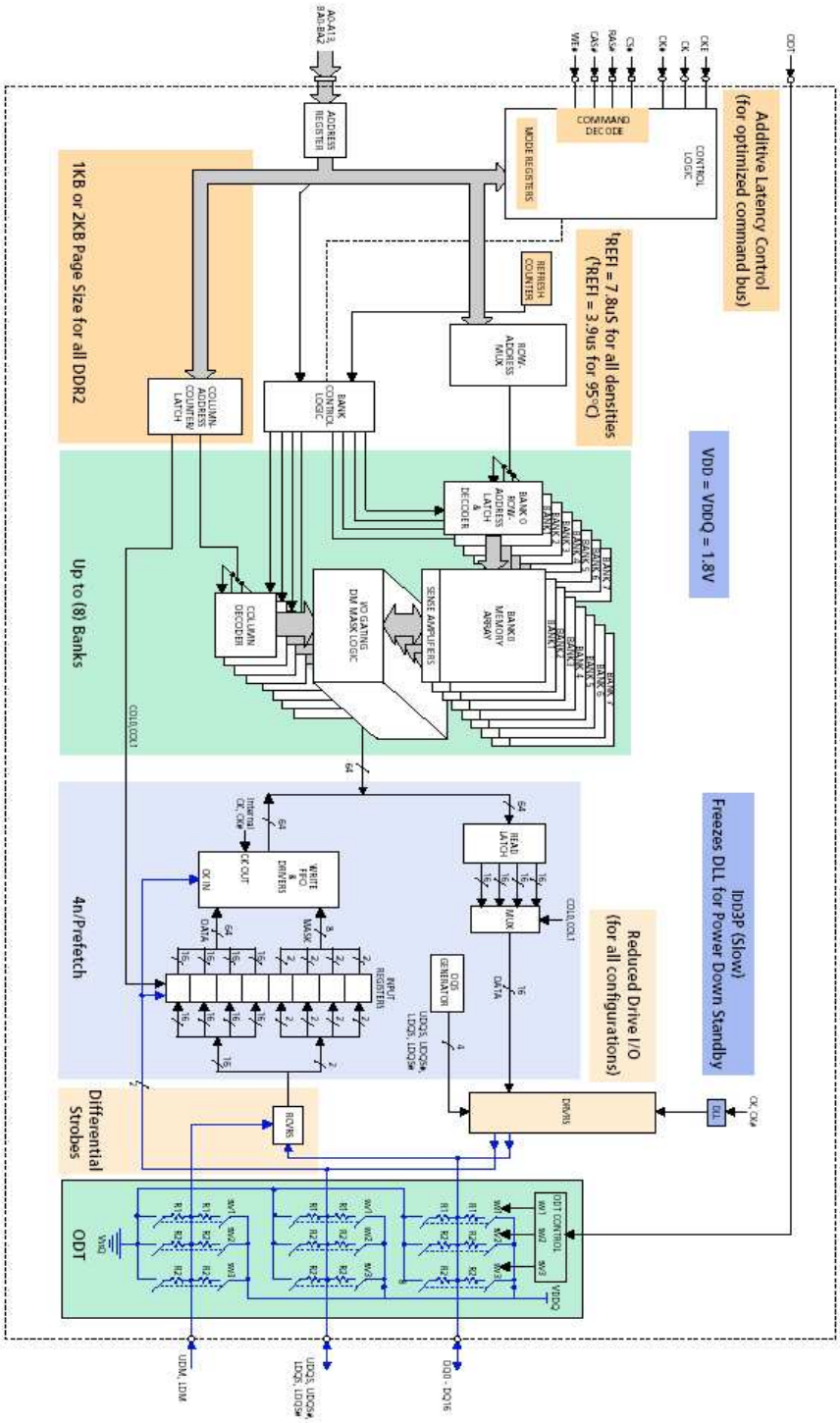

Figure 1.8: DDR II block diagram.

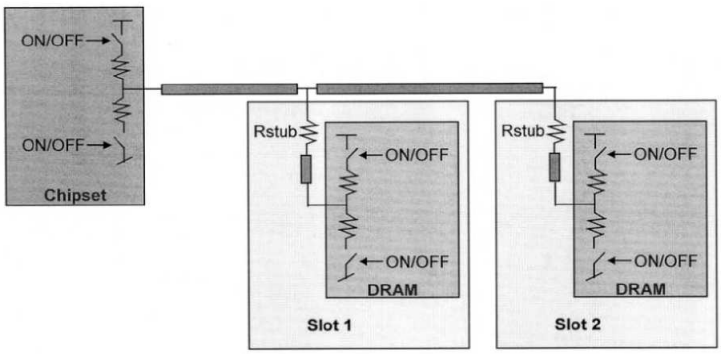

Figure 1.9: Two slot memory system with SSTL bus terminated by ODT.

The channel structure of a DDR-II SDRAM (fig.1.8) memory system is basically the same as that of a DDR-I SDRAM memory system [4].

DDR II improves data bandwidth by using eight banks. The eight banks increase the flexibility of accessing large memory DDR II by interleaving different memory bank operations. The DDR II architecture employs 4-bit prefetch architecture, i.e. the internal bus width has been made four times wider than the external bus width, so that data bus transfers can be accelerated by a factor of four without having to change the operating speed of the internal bus (memory cell array).

The *On-Die Termination* is the most significant feature that has been added to the DDR II SDRAM. This function greatly improve the signal integrity at over 533Mb/s operation. Unlike the conventional SSTL bus of DDR SDRAM, termination resistors are located inside the DRAMs in the DDR-II memory system as shown in fig.1.9, and they can be switched ON and OFF. Embedding the termination resistors inside the DRAM, the DDR-II memory system has much better signal integrity than conventional SSTL bus. In the usual way, i.e. with motherboard termination, a resistor with a suitable resistance value is connected at the end of transmission path (fig.1.10). This method doesn't reduce signal reflection adequately in the operating range used by DDR II SDRAM. In fact, if there are several DRAMs on the same bus, such as is shown in fig.1.10, DRAM that are in active mode are affected by reflected signal from DRAM that are in stand by mode. In other words, reflections caused at the I/O interface to the memory chips will have to re-enter the bus first before they can be terminated. There is a constant noise level on

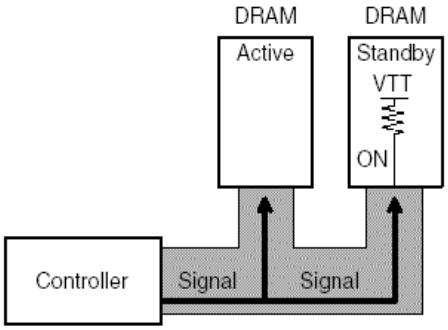

Figure 1.10: Signal Reflection when Using Motherboard Termination.

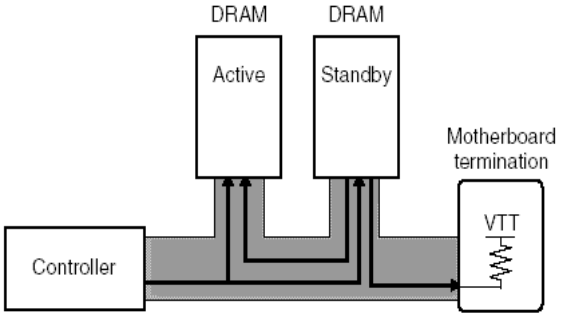

Figure 1.11: ODT and Reflected Signals.

the bus that will interfere with the data signals and cause quality degradation. Using ODT, all chips that are not selected for data I/O, that is, those that are in standby mode will have ODT switched On to eliminate reflection where it originates, that is, at the bus to die interface. As a result, there is no possibility for the reflections to reenter the bus and contaminate the other signals (fig.1.11).

Tab.1.1 [5] compares the simulated signal integrity between the conventional motherboard terminated (MBT) bus and ODT bus of DDR-II SDRAM. The inter-symbol interference (ISI), aperture size, and slew rate are all better in the ODT bus than in MBT bus. The ODT can have either 50Ω, 75Ω or 150Ω depending on the system configuration.

	Write		**Read**	
	MBT	ODT	MBT	ODT
Inter-symbol interference [ps]	493	432	503	373
Aperture [V]	0.43	0.85	0.67	1.14
Slew [V/ns]	0.42	0.62	0.59	0.88

Table 1.1: Signal Integrity Comparison.

Another function that has been added to the DDR II is the *Off-Chip Driver* [6]. As just discussed above, DDR II introduce a bidirectional, differential I/O data strobe consisting of DQS and $\overline{DQS}$ as complementary signals. Differential means that the two signals are measured against each other instead of using a simple strobe signal and a reference point. In theory the pull-up and pull-down signals should be mirror-symmetric to each other but reality shows otherwise. That means that there will be skew-induced delays to reaching the output high and low voltages (V_{OH} and V_{OL}) and the cross points between DQS and $\overline{DQS}$ used for clock forwarding will not necessarily coincide with the DQ crossing the reference voltage (Vref) or even be consistent from one clock to the next. The mismatch between clock and data reference points is referred to as the DQ-DQS skew. One way to solve the problem is to use Off-Chip Driver calibration where both parts of the differential strobes are calibrated against each other and against the DQ signal. Through this sort of calibration, the ramping voltages are optimized for the buffer impedances to reduce over and undershooting at the rising and falling edges. More importantly, DQS and $\overline{DQS}$ are matched so that their cross point coincides with the DQ signal crossing the reference voltage to eliminate DQ-DQS skew. In summary, the entire scheme results in better compatibility between different designs, higher signal integrity through minimization of DQ-DQS skew and reduced overshoot/undershoot for better signal quality.

Another feature is that signal reflections can be controlled through selectable output drive levels. Two mode are selectable, full and reduced drive. DDR II configured as full drive has a target output impedance of approximately 18Ω. When the device is configured as reduced drive the target output impedance is approximately 40Ω.

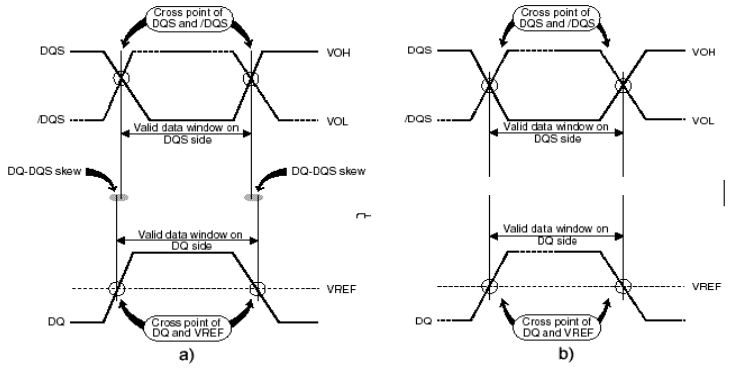

Figure 1.12: a)Before OCD calibration; b)After OCD calibration.

1.6 Functional Memory Testing

The fast development of memory devices and the strong market competition have increased the standards of these produced devices. The increased demand on reliability has, in turn, stressed the importance of failure analysis and memory testing techniques. Tests for semiconductors' memories have experienced a long development process. Before 1980, tests required long test times for a given fault coverage (FC) (i.e. the number of detected faults divided by the number of total faults), typically of order $O(n^2)$, where , n is the size of the memory. In order to reduce the test time and improve the FC, during the early 1980s, the *Functional Faults Models*, which are abstract fault models, were introduced. A functional fault model is defined as the deviation of the observed memory behavior from the functionally specified one, under a given sequence of performed memory operation. The advantages of these models was that the FC could be proven while the test time was usually of order $O(n)$, that is linear with the size of the memory. The first functional model were not based on real memory design. The next stage was the introduction of the inductive fault analysis (IFA). IFA is a systematic procedure to predict the faults in an integrated circuit by injecting spot defects in the simulated geometrical representation of the circuit. The result was that new functional fault models where introduced. In the late 1990s, the experimental results of applying a large number of tests to a large number of chips indicated that many functional tests detect faults which cannot be explained with the existing fault models at that time. This led to the introduction of the framework of all possible fault models for memories based on the *fault primitive*(FP)

concept. Any difference between the observed and expected memory behavior can be denoted by the following notation $< S/F/R >$, referred to as a *fault primitive*(FP). S describes the operation sequence that sensitizes the fault, F describes the value of the faulty cell and R describes the logic output level of a read operation. For example, in the FP $< 0_c w1_c/0/- >$, $S = 0_c w1_c$ means that cell c is assumed to have the initial value 0, after which a 1 is written into c. The fault effect F=0 indicates that after performing a $w1$ to c, as indicates by S, c remains in state 0.The concept of FP also allowed for the classification of the memory faults framework in different classes:static against dynamic (depending on the number of operation required to sensitize the the fault), simple against linked (depending on the way the FPs manifest themselves) and so on. The faults can be classified into different class depending on S. If only a single cell is involved in S the fault is called *single cell fault*, if more cells are involved in S the FP sensitized by the corresponding S is called *coupling fault*. In the coupling fault a cell or group of cells (aggressor) influences the behavior of another cell (victim).

1.6.1 Single-Cell Static Fault

Single cell faults are restricted to a single cell of the memory array. This class consists of the following fault types:

1. *State Fault (SF_x)*— A cell is said to have a *State Fault* if the logic value of the cell changes before it is accessed (read or written), even if no operation is performed on it. No operation is needed to sensitize it and, therefore , it only depends on the initial stored value in the cell. There are two types of state faults: State-0 fault (SF_0) and State-1 fault (SF_1).

   - State-0-fault (SF_0) =$\{< 0/1/- >\}$,
   - State-1-fault (SF_1) =$\{< 1/0/- >\}$

2. *Transition Fault (TF_x)*— A cell is said to have a *transition fault* if it fails to undergo a transition ($0 \rightarrow 1$ or $1 \rightarrow 0$) when it is written. This fault is sensitized by a write operation and depends on both the initial stored logic level and the type of the write operation. There are two types of transition faults: Up transition fault (TF_1) and Down transition fault (TF_0)

- Down transition fault (TF_0) ={$< 1w0/1/- >$},
- Up transition fault (TF_1) ={$< 0w1/1/- >$}

3. *Read Destructive Fault (RDF_x)*— A cell is said to have a *read destructive fault* if a read operation performed on the cell changes the data in the cell and returns an incorrect value on the output. There are two types of read destructive faults: Read-0 destructive fault (RDF_0), Read-1 destructive fault (RDF_1).

   - Read-0 destructive fault (RDF_0) ={$< 0r0/1/1 >$},
   - Read-1 destructive fault (RDF_1) ={$< 1r1/0/0 >$}

4. *Write Destructive Fault (WDF_x)*— A cell is said to have a *destructive fault* if a non-transition write operation causes a transition in the cell.There are two types of write destructive faults: Write-0 destructive fault (WDF_0) and Write-1 destructive fault (WDF_1).

   - Write-0 destructive fault (WDF_0) ={$< 0w0/1/- >$},
   - Write-1 destructive fault (WDF_1) ={$< 1w1/0/- >$}

5. *Incorrect Read Fault (IRF_x)*— A cell is said to have an *incorrect read fault* if a read operation performed on the cell returns the incorrect logic value, while keeping the correct stored value in the cell. There are two types of incorrect read faults:Incorrect read-0 fault (IRF_0) and Incorrect read-1 fault (IRF_1).

   - Incorrect read-0 fault (IRF_0) ={$< 0r0/0/1 >$},
   - Incorrect read-1 fault (IRF_1) ={$< 1r1/1/0 >$}

6. *Deceptive Read Destructive Fault ($DRDF_x$)*— A cell is said to have a *deceptive read destructive fault* if a read operation performed on the cell return the correct logic value, while changing the contents of the cell. There are two types of deceptive read disturb faults: Deceptive r0 destructive fault ($DRDF_0$) and Deceptive r1 destructive fault ($DRDF_1$).

   - Deceptive r0 destructive fault ($DRDF_0$) ={$< 0r0/1/0 >$},
   - Deceptive r1 destructive fault ($DRDF_1$) ={$< 1r1/0/1 >$}

1.6.2 Two-cell static faults

In the coupling fault a cell or group of cells (aggressor) influences the behavior of another cell (victim). This class consists of the following fault types:

1. *State coupling fault (CF_{st})*— Two cells are said to have a state coupling fault if the victim is forced into a given logic state only if the aggressor is in a given state, without performing any operation on the victim.

2. *Disturb coupling fault (CF_{ds})*— Two cells are said to have a disturb coupling fault if an operation (write or read) performed on the aggressor forces the victim into a given logic state. Here, any operation performed on the aggressor is accepted as a sensitizing operation for the fault, be it a read, a transition write or a non-transition write operation.

3. *Transition coupling fault (CF_{tr})*— Two cells are said to have a transition coupling fault if the state of the aggressor results in the failure of a transition write operation performed on the victim. This fault is sensitized by a write operation on the victim, while the aggressor is in a given state.

4. *Write destructive coupling fault (CF_{wd})*— A cell is said to have a write destructive coupling fault if a non-transition write operation performed on the victim results in a transition, while the aggressor is in a given logic state.

5. *Read destructive coupling fault (CF_{rd})*—Two cells are said to have a read destructive coupling fault if a read operation performed on the victim destroys the data stored in the victim, while a given state is present in the aggressor.

6. *Incorrect read coupling fault (CF_{ir})*— Two cells are said to have an incorrect read coupling fault if a read operation performed on the victim returns the incorrect logic value, while the aggressor is in a given state.

7. *Deceptive read destructive coupling fault (CF_{dr})*—A cell is said to have a deceptive read destructive coupling fault if a read operation performed on the victim returns the correct logic value and changes the contents of the victim, while the aggressor is in a given logic state.

1.7 March Test

In order to inspect memory devices for possible faulty behavior, memory testing is performed on all produced memory components. Many types of memory tests are being used today, each with its own advantages and disadvantages. *March tests* are among the most popular memory tests, due to their low complexity and high fault coverage.

A March test consists of a sequence of march elements; a march element consists of a sequence of operations which are all applied to a given cell, before proceeding to the next cell. The way one proceeds to the next cell is determined by the address order which can be an increasing address order (denoted by $\Uparrow$), or a decreasing address order (denoted by $\Downarrow$). For some march elements the address order can be chosen arbitrarily, this will be indicated by the $\Updownarrow$ symbol. An operation, applied to a cell, can be a $w0$ (write '0'), a $w1$ (write '1'), a $r0$ (read '0'), or a $r1$ (read '1') operation. A complete march test is delimited by the '...' bracket pair, while a march element is delimited by '(...)' bracket pair. In the following the most used march test are listed:

MATS: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Updownarrow (r1)\}$

MATS+: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0)\}$

MATS++: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0, r0)\}$

March X: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0)\}$

March Y: $\{\Updownarrow (w0); \Uparrow (r0, w1, r1); \Downarrow (r1, w0, r0); \Updownarrow (r0)\}$

March C: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Updownarrow (r0) \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0); \}$

March C-: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); \Updownarrow (r0); \}$

MOVI: $\{\Downarrow (w0); \Uparrow (r0, w1, r1); \Uparrow (r1, w0, r0); \Downarrow (r0, w1, r1); \Downarrow (r1, w0, r0); \}$

IFA-9: $\{\Updownarrow (w0); \Uparrow (r0, w1); \Uparrow (r1, w0); \Downarrow (r0, w1); \Downarrow (r1, w0); Delay; \Updownarrow (r0, w1); Delay; \Updownarrow (r1); \}$

IFA-13: $\{\Updownarrow (w0); \Uparrow (r0, w1, r1); \Uparrow (r1, w0, r0); \Downarrow (r0, w1, r1); \Downarrow (r1, w0, r0); Delay; \Updownarrow (r0, w1); Delay; \Updownarrow (r1); \}$

1.8 Signal Integrity analysis for High Data Rate Memory

The data rate of the DDR II is expected to move from currently 400MHz to 667Mhz and 800MHz in the near future. Faster clock speeds requires smaller voltage swings and shorter setup and hold time. As result, data valid window means that jitter induces noise, cross talk, and intersymbol-Interference further reduce its size, creating errors. The signal integrity analysis is still a challenging task, that could not be studied with traditional tools cited in previous paragraph. The easiest way to study signal integrity is to plot the eye diagrams using an ultrawideband sequential oscilloscope.

This thesis work provides a memory test system which performs the traditional tests in order to detect functional faults and supports up to 64 channels simultaneous signal integrity testing.

The memory test is designed to provide, on each memory channel, the functions of a sampling oscilloscope. For each channel performs eye diagram analysis, rise/fall time, logic level and eye opening measurements.

Chapter 2

Measuring Jitter in digital system

2.1 Introduction

The obvious technology trend today is to have for-ever increasing bit rates, higher clock speeds and technology advances that keep lowering operating voltages, reducing package sizes and ball pitch, and forcing more components into a smaller amount of board area. As frequency increases, the trace on a circuit board become more than simple conductors. At lower frequencies (such as the clock rate of an older digital system) the trace exhibits mostly resistive characteristics. At higher frequencies, the trace begins to act like an impedance not purely resistive. A the highest frequencies, the trace's inductance plays a larger role. All of this characteristics can adversely affect the integrity of the transmitted signals.

A signal integrity problem is defined as any phenomenon that can compromise a signal's ability to convey binary information. Having good signal integrity means controlling unwanted noise on logic signals. Noise falls into one of two main domains:

- *Level related noise* affects the logic level of the signal. If the noise is large enough, the signal may cross the threshold from a desired logic state to undesired state and propagate into other logic.
- *Time related noise*, affect the position of a signal transition and causes

setup/hold window for data sampling to be violated, thereby allowing incorrect data to be sampled and propagated through the system.

The combination of amplitude and timing jitters contribute to reduce signal margins in both the time and frequency domains, effectively reducing the window in which good data is available.

2.2 Jitter Definition

Jitter is usually defined as the deviation of a timing event of a signal from its ideal position [7]. Jitter describes a timing uncertainty, and therefore has to be considered when we look a timing budget of a design. Jitter is just another component that makes part of the bit period unusable for sampling just like setup and hold time. However unlike setup and hold times that are usually thoroughly specified for logic families and can be taken from data sheets, jitter is a function of the design and has to be measured. There are two definitions of jitter, an analog and a digital one. In the analog world jitter is also known as *phase noise*, and defined as a phase offset that continually changes the timing of a signal

$$S(t) = P(t + \Phi(t)) \tag{2.1}$$

where S(t) is the jittered signal waveform, P(t) is the undistorted waveform, and ϕ(t) is the phase offset, or phase noise. This definition is most useful in the analysis of analog waveform like clock signals, and frequently used to express the quality of oscillators. In the digital world, we are looking only at the 1-0 and 0-1 transitions of the signal, and jitter is therefore only defined when such a transition occurs (crossover). The jittered digital signal can be written as

$$t_n = T_n + \Phi_n \tag{2.2}$$

where t_n is the time when the nth transition occurred, T_n is the ideal timing value for the nth transition, and Φ_n is the time offset of this transition (advance or delay), also known as the time jitter.

2.3 Jitter Classification

Jitter on a signal will have different characteristics depending on its causes, so categorizing the sources of jitter becomes important for measuring and

analyzing jitter [8].

The random noise mechanisms fall in the first category, processes that randomly introduce noise to a system. These sources include:

- *Thermal Noise* which can be represented by broadband "white" noise, and has flat spectral density. It is generated by the transfer of energy between "free" electrons and ions in a conductor. The amount of energy transfer and, therefore, the amount of noise, are related to temperature. It is associated with electron flow in conductors and increases with bandwidth, temperature and noise resistance.

- *Shot Noise* is broadband "white" noise generated when electrons and holes move in a semiconductor. Shot noise amplitude is a function of average current flow. The current fluctuations about the average value give rise to noise.

- *Flicker Noise* has a spectral distribution that is proportional to $1/f^{\alpha}$ where α is generally close to unity. Because flicker noise is proportional to $1/f$, its contribution is most dominant at lower frequencies. The origin of flicker noise is a surface effect due to fluctuations in the carrier density as electrons are randomly captured and emitted from oxide interface traps.

The second category is ruled by system mechanisms, effects on a signal that result from characteristics of its digital system. Examples of system related noise sources include:

- *Crosstalk* which occurs when the magnetic or electric fields of a signal on a conductor are inadvertently coupled to an adjacent signal-carrying conductor. The coupled signal components algebraically add to the desired signal, and can slightly alter its bias depending on the amount of coupling and the frequency content of the interfering signal. The altered bias translates into jitter as the signal crosses the receiver's threshold.

- *Electromagnetic Interference or EMI* which is the result of unwanted radiated or conducted emissions from a local device or system. Switching-type power supplies are common sources of EMI. These devices can radiate strong, high frequency electric and magnetic fields, and they can conduct a large amount of electrical noise into a system if they lack adequate shielding and output filtering.

- *Reflections* which occur when impedance mismatches are present in a data channel. With copper technology, the max power transfer occurs when the transmitter and receiver have the same characteristic impedance as the medium. If an impedance mismatch is present at the receiver a portion of the energy is reflected through the medium to the transmitter. Reflections typically come from uncontrolled stubbing and incorrect terminations. Reflected energy, or energy not available to the receiver, reduces the signal to noise ratio at the receiver and increase jitter. If the transmitter is also mismatched, the transmitter absorbs a portion of the reflected signal energy while the remainder is reflected toward the receiver.

The so-called data category includes *Data dependent* mechanisms, in which the patterns or other characteristics of the data being transferred affect the net jitter seen at the receiver. Data-dependent jitter sources include:

- *Duty Cycle Distortion* which occurs when long strings of ones or zeros cause the voltage level to drift and consequently delay the edge transition. It is observed when the durations of logic 1 pulses are different than the duration of logic 0 pulses. More details will be shown in the par.2.5.1.

- *Intersymbol Interference* (ISI) which is caused by attenuation and bandwidth limitations of transmission structure. It is a function of the data rate, board layout and material, and the data pattern sent over the link. The ISI results in reduced edge speeds, and in both retarded or advanced edges relative to their ideal positions. More details on ISI will be shown in the par.2.5.2

The mentioned sources of jitter are categorized into two types, *unbounded* and *bounded*. Unbounded jitter does not achieve a maximum or minimum phase deviation within any time interval, and jitter amplitude from these sources theoretically approaches infinity. This type of jitter is referred to as *random jitter* and result from random noise sources identified in the first group above. It is usually described by a Gaussian distribution which is quantified by the standard deviation, J^R_{rms}, and mean. Bounded Jitter sources reach maximum and minimum phase deviation value within an identifiable time interval. This type of jitter is also called *deterministic jitter*, and result from system and data-dependent jitter producing mechanism (the second and third group identified above). In particular, the jitter caused by the EMI

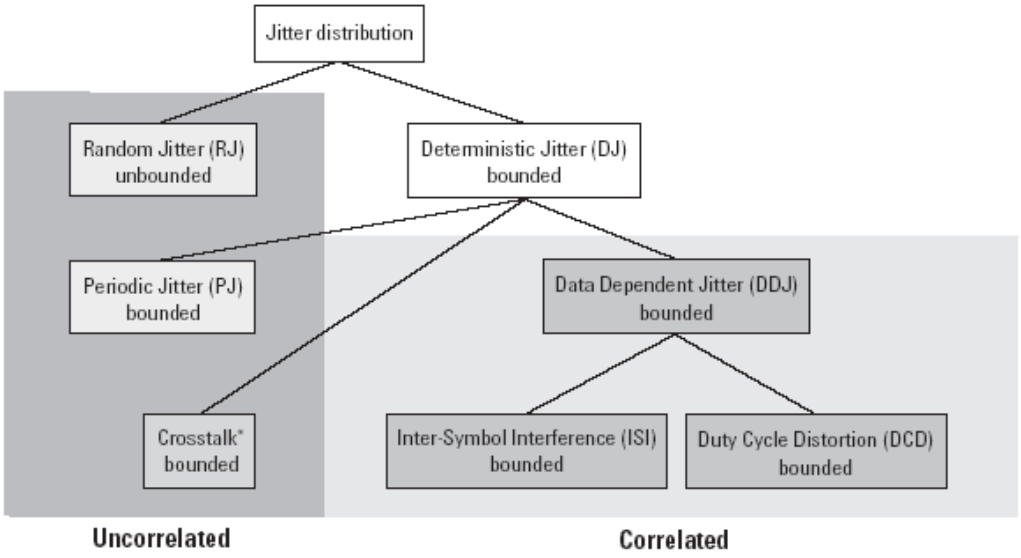

Figure 2.1: Jitter segmentation.

mechanism is called *Periodic Jitter*(par.2.6). The Deterministic Jitter doesn't follow any a-priori predictable distribution and is characterized by peak-to-peak amplitude, J_{pp}^{D}. Fig.2.1 shows the typical jitter segmentation where the DCD and ISI are classified as *bounded correlated jitter*, PJ and crosstalk as *bounded uncorrelated jitter* and RJ as *unbounded uncorrelated jitter*.

2.4 Jitter Measurements

The traditional way to measure jitter is with an *eye diagram* [9]. An eye diagram is a composite view of the sampled bit periods of a waveform. Fig.2.2 shows an idealized eye diagram, very straight and symmetrical with smooth transitions (left and right crossing points), and a large, wide open eye to provide an ideal location to sample a bit. The Y-axis represents signal amplitude, the X-axis represents time *relative* to the occurrence of a specific event, commonly called a *trigger event*. The eye diagram is obtained supplying a data stream to an ultrawideband sequential-sampling oscilloscope. When the oscilloscope responds to a trigger event, it measures a portion of the bit stream.

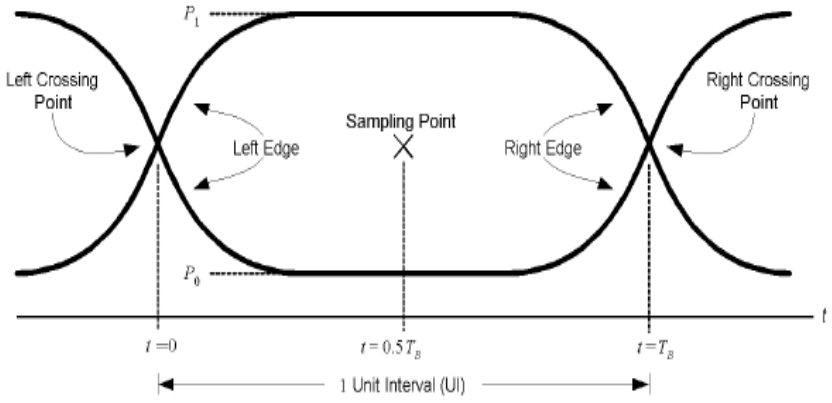

Figure 2.2: An idealized eye diagram.

As other triggers arrive, it measures different portions of the data. With the persistence, the display build up with several waveforms. The various combinations of signal overlay on each other and create the eye diagram. The *Unit Interval* is the time period equivalent to one bit time in a serial data stream. It is the reciprocal of the baud rate. The *Sampling Point* is the specific instant where the data must be sampled. Ideally, the sampling instant would always occur at the center of a data bit time, equidistant between two adjacent edge transition points. The presence of jitter changes the edge positions with respect to the sampling point. An error will then occur when a data edge falls on the wrong side of a sampling instant.

To develop an intuition for RJ and DJ [10], consider fig.2.3 where there are several distinct pulse shapes. The different pulse shapes are caused by DJ. The width of the lines is determined by RJ. That is, DJ determines which line a given bit will follow and RJ determines how much that bit oscillates about the DJ determined average. J_{pp}^{DJ} is given by the distance between the most widely separated DJ determined edges, and J_{rms}^{RJ} is the rms deviation about a given edge.

Another jitter measurement viewpoint is the *histogram*. It plots the range of value exhibited by the analyzed parameter on the x-axis versus the frequency of occurrence on the y-axis (*Probability Density Function* or *PDF*). The histogram, is of great importance for separating random from deterministic jitter. Fig.2.4 shows an eye diagram and its associated histogram. The eye

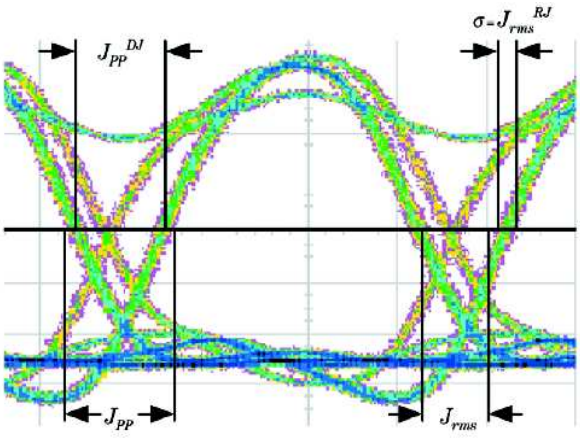

Figure 2.3: An eye diagram with both Random Jitter (RJ) and Deterministic Jitter (DJ) distinguishing the genuine rms jitter, from the rms jitter caused by RJ, J_{rms}^{RJ}, and the total peak-to-peak jitter,J_{pp}, from the peak-peak jitter caused by DJ,J_{pp}^{DJ}.

diagram is shifted so that it centers on the transition or crossing points between two eyes. This waveform appears to have two distinct rising and falling edges, indicating the presence of deterministic jitter, but all trajectories are spread out or fuzzy, so a large random jitter is also present. The histogram derived from this eye diagram's transition point has a bimodal characteristic rather than a single Gaussian curve. This is indicative of a signal with both random and deterministic jitter.

2.5 Data Dependent Jitter (DDJ)

DDJ is a part of deterministic Jitter, and consist of *Duty Cycle Distortion* (DCD) and *Inter-Symbol Interference*. This is referred to as data correlated jitter, and its response is determined by both current state and previous state of data.

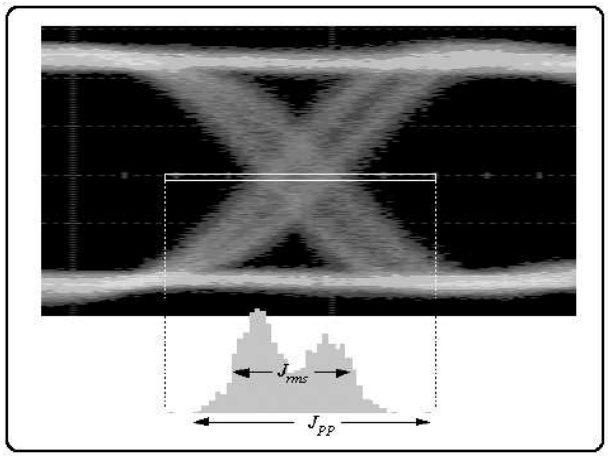

Figure 2.4: Histogram.

2.5.1 Duty Cycle Distortion

The *Duty Cycle Distortion*[1] (DCD) is both the variance in timing away the 50% duty cycle, and also the variance in average voltage offset. DCD occurs when the transmitter threshold is drifted from its ideal level and consequently the edge transition shifted. It is easily observed in the eye diagram as the nominal eye crossing (where rising edges intersect falling edge) occurring somewhere other than 50% amplitude point. Fig.2.5 shows a typical Duty Cycle distortion. The dotted line shows the ideal output of a transmitter with an accurate threshold level set at 50% and with a duty cycle of 50%. The solid line waveform represents a distorted output of a transmitter due to a positive shift in the threshold level. With a positive shift in threshold level, the resultant output signal of the transmitter will have less than 50% duty cycle. If the threshold level is shifted negatively, then the output of the transmitter will have greater than 50% duty cycle.

Another cause of DCD is asymmetry in rising and falling edge speeds. A slower falling speed relative to the rising edge will result in greater than 50% duty cycle for a repeating 1-0-1-0...pattern, and slower rising edge speeds relative to the falling edge will result in less than 50% duty cycle. The corresponding results are similar to the previous example, illustrated in Fig.2.5.

[1]Duty Cycle is the ratio of the pulse duration to the pulse period. Ideally the duty cycle is 50%.

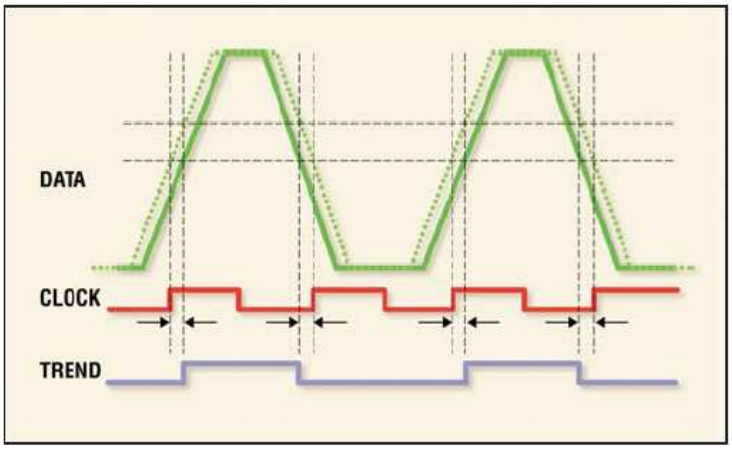

Figure 2.5: Duty Cycle Distortion.

2.5.2 Inter Symbol Interference

Another cause of jitter is *Inter-Symbol Interference* (ISI) [11]. Its main effect is the reduction of the edge speed and so of the amplitudes of the data bits which is dependent on repeating-bit lengths and preceding bit states. Obviously this will result in vertical eye closure and in both retarded and advanced edges relative to their ideal positions. There are two primary causes of ISI: the bandwidth limitation in either the transmitter or physical media and improper impedance termination and physical media discontinuities that cause reflections.

Fig.2.6 shows an example of ISI due to bandwidth limitation problems. Limited bandwidth produces limited edge speeds will result in varying pulse amplitudes at high speed data rates. Varying pulse amplitudes will then result in transition timing errors.

With a long series of repeating "1's", the amplitude of the data signal will eventually rise to a full steady state high level as illustrated by the long-high pulse at point A in fig.2.6. When the state of the data change to a "0", the signal will have a relatively long transition time to reach the threshold level, resulting in a positive timing error. This will be manifested as a positive peak of timing error in the jitter trend waveform at point B. The negative

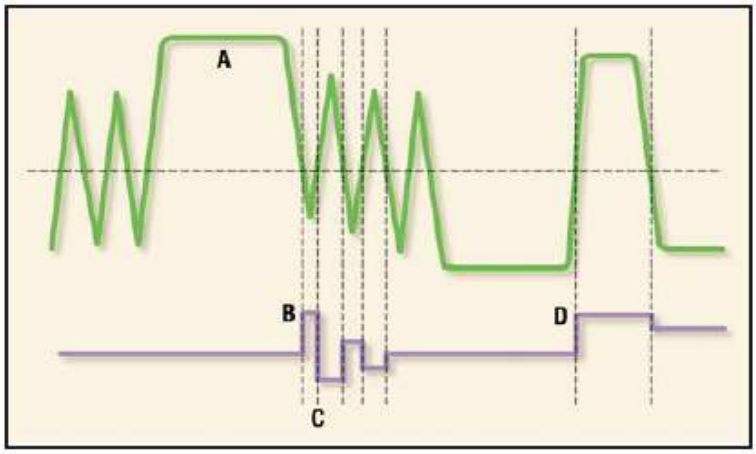

Figure 2.6: Inter-Symbol Interference due to bandwidth problems.

peak amplitude af the next "0" bit preceded by a long string of "1's" will be attenuated for two reasons. First, the preceding long string of "1's" means that the signal will take longer to transition to a true low level since the data signal starts from a higher initial level. Secondly, the following "1" bit causes the signal to reverse direction before it even reaches a solid low level. This reduction in signal amplitude will produce a negative timing error on the next transition to a "1" since the signal has a very short distant to travel to reach the threshold level. This is illustrated at point "C" on the jitter trend waveform. The positive timing error illustrated at point D on the jitter trend waveform follows the same logic as the positive timing error at point B previously discussed. Another cause of inter-symbol interference is signal reflections which occur when impedance mismatches are present in a data channel. Signal reflections will produce distortions in the amplitude of the data signal as shown in fig.2.7. If the amplitude of the signal becomes distorted on or near a data transition edge due to reflections, then a timing error may occur.

2.6 Periodic Jitter

Periodic Jitter (PJ) is usually the result of Electro Magnetic Interference (EMI) problems in the system. It provides sinusoidal varying jitter, where a sinusoidal signal modulates the phase of the ideal clock or data signal.

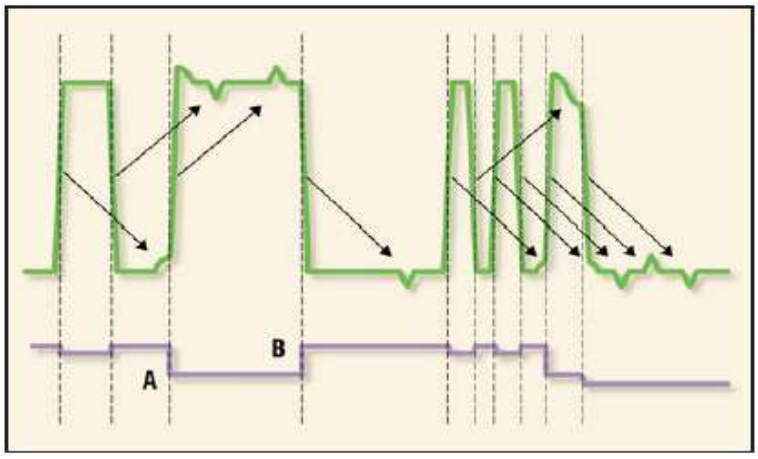

Figure 2.7: Inter-Symbol Interference due to reflections.

Referring to fig.2.8, in the top is shown a regular clock signal with a 50% duty cycle, in the middle the same clock signal with a sinusoidal phase perturbation at 1/10 the original signal, and with an amplitude of $4/3\pi$, and in the bottom a phase versus time plot of the phase perturbation sinusoid. An example of PJ would be signals from a switching power supply coupling into the data or system clock signals. It is not time-correlated with either the clock or data signal since it would be based on a different clock source.

Fig.2.9 shows an example of a "corrupter" signal (upper trace) capacitively coupling into our serial data signal (middle trace). This coupling will result in amplitude distortion on the data signal. If the amplitude distortions occur at or near a data signal transition, a timing error occurs.

2.7 Bounded Uncorrelated Jitter

Bounded Uncorrelated Jitter [12] is commonly caused by crosstalk coupling from adjacent interconnects on printed circuit board. It is bounded due to finite coupling strength, and uncorrelated because there is no correlation to the channel'own data pattern. In other words, the correlation is with the adjacent traces, not with the trace under study. By definition, crosstalk is the coupling of energy from one trace to another. This coupling is due to the electromagnetic field generated by the propagating signals, and its strength

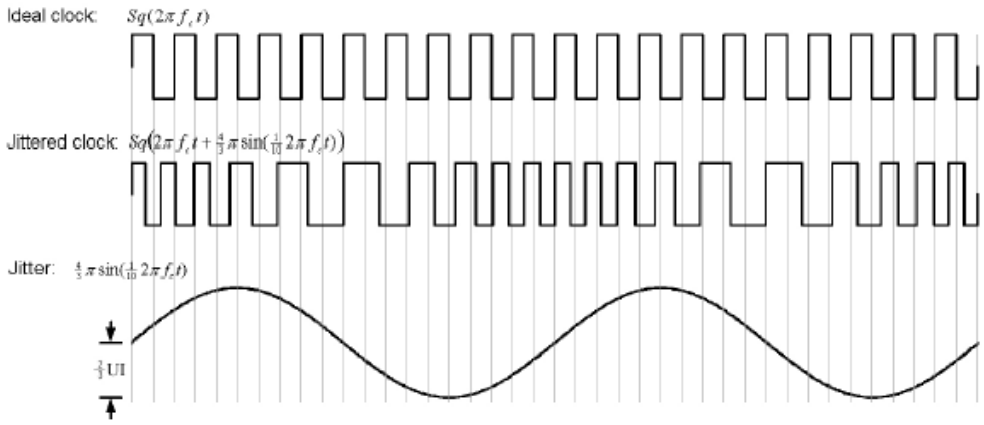

Figure 2.8: Periodic Jitter (PJ).

is dependent on the physical layout and properties of the traces. Crosstalk is caused by two main effects: capacitive and inductive coupling.

The crosstalk-induced pulse trace travels both backward toward the near end, and forward toward the far end of the victim trace, as shown in fig.2.10. The currents due to mutual capacitance propagate toward both the near and far end of the victim. On the other hand, mutual inductance only drives the current from the far end toward the near end in the victim. The total crosstalk is the result of the subtraction between inductive and capacitive induced pulses. The far end voltage amplitude of the crosstalk can be calculated from

$$V_{pfar} = \frac{\Delta V_a l \sqrt{LC}}{2T_{edge}}(\frac{L_m}{L} - \frac{C_m}{C}) \tag{2.3}$$

where ΔV_a is the total amplitude change of the aggressor signal, L_m is the mutual inductance per unit length, L is the self-inductance per unit length, C_m is the mutual capacitance per unit length , C is the self-capacitance per unit length, l is the length of the trace, and T_{edge} is the edge transition time of the aggressor signal. ΔV_a has a positive polarity for a rising edge transition and a negative polarity for a falling edge transition. Fig.2.11 illustrates how the distorted victim edge transition can occur earlier or later than the distortion-free victim edge transition. If the amplitude of the cross-talk induced pulse is negative, the distorted victim rising edge will occur later than the distortion-free edge transition, as illustrated in fig.2.11a. On the other hand, if the amplitude of the crosstalk-induced pulse is positive, the distorted victim rising

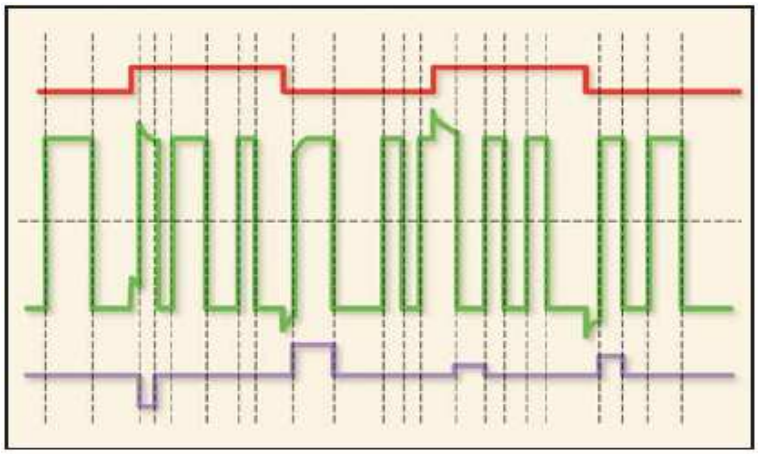

Figure 2.9: Periodic Jitter (PJ) caused by capacitive coupling.

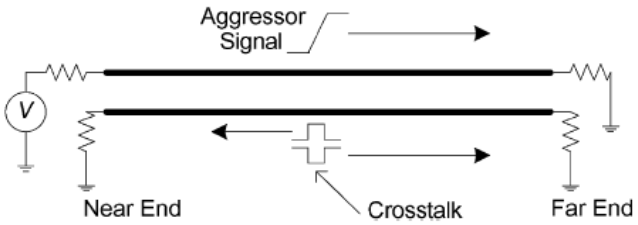

Figure 2.10: Forward and background propagation

edge will occur earlier, as illustrated in fig.2.11b. The time difference can be calculated from subtracting the edge crossings of the distortion-free and the distorted victim edge transition.

2.8 Total jitter

The quality of a digital transmission system can be expressed in terms of how many bits out of transmitted sequence were received in error. The BER is defined as the number of bits received in error divided by the number of bits

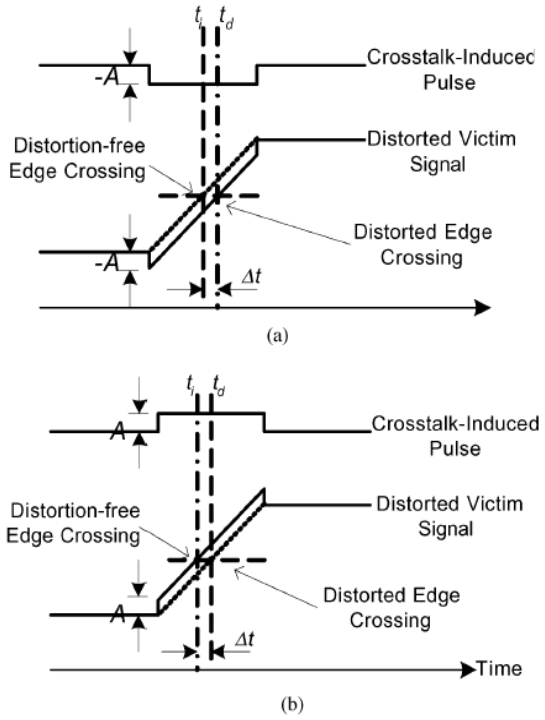

Figure 2.11: Timing difference due to crosstalk.(a)Distorted edge crossing occurs later then the original edge crossing. (b) Distorted edge crossing occurs earlier than the original edge crossing

transmitted

$$BER = \frac{N_{Err}}{N_{Bits}} \tag{2.4}$$

To calculate the total jitter on a signal, the components of the Jitter must be combined to obtain the expected BER performance. If the individual components are independent it is possible to use a convolution operation to calculate the distribution which represent the Total jitter

$$J(x) = PJ(x) \otimes DDJ(x) \otimes RJ(x) \tag{2.5}$$

The total jitter distribution is divided in three regions: at the crossing point the distribution is dominated by DJ, at time-delays farther from the crossing point the distribution is increasingly dominated by RJ until, far from the crossing point, in the asymptotic limit, the tails follow the Gaussian RJ distribution. The ideal way to determine the behavior of the tails, and hence the

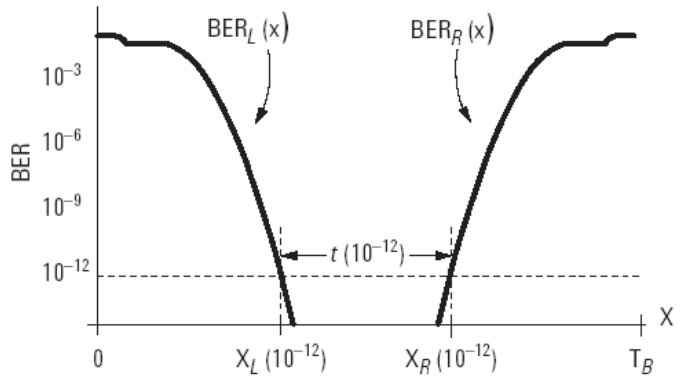

Figure 2.12: A bathtube plot: the bit error ratio as a function of sampling point delay, x

TJ(BER) would be to deconvolve RJ and DJ. But without knowing the DJ distribution, there is no practical way to deconvolve the distribution.

2.8.1 Calculating Total Jitter from Bathtub Curves

The goal of the jitter analysis is to determine the effect of the jitter on the BER and ensure that the system BER is less than some maximum value, usually 10^{-12} for many applications. This test is usually done using an error performance analyzer, commonly referred to as a *Bit Error Rate tester* or BERT. A binary sequence (it could be a pseudorandom sequence PRBS) from a pattern generator is used to modulate the transmission system's source, while an error detector compares the received signal with the original transmitted pattern and counts for received bits and errors, and calculates the BER. Fig.2.12 shows the *bathtube plot* that is a graph of BER versus sampling point throughout the unit interval.

BER(x), can also be derived from the jitter distribution, J(x). Since BER(x) is given by the probability for a logic transition fluctuating across the sampling point time position, x, if we consider the left edge of the eye diagram the probability of a transition fluctuating across the point x is given by the product between the *Cumulative Probability Density Function*(CDF)[2] and

[2]The CDF provides for each time value the probability that the transition happened

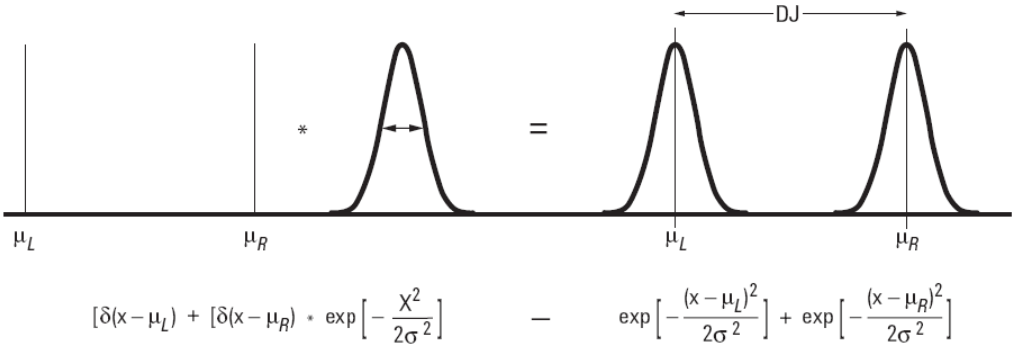

Figure 2.13: The convolution of the sum of two delta functions separated by DJ and a Gaussian RJ distribution of width σ. The underlying assumption of the dual-Dirac approximation is that any jitter distribution can be modeled in this way.

ρ_T , the *transition density*, which is the ratio of the number of logic transitions to the total number of bits:

$$BER_L(x) = \rho_T \cdot \int_x^{+\infty} J(x')dx' \tag{2.6}$$

The transition density measures 0.500 for a repetitive pattern 0-1-0-1..., and 0.4961 for a PBRS7 sequence (pseudorandom bit sequence of length $2^7 - 1$).

Similarly, on the right side of the eye diagram, near $x = T_B$,

$$BER_R(x) = \rho_T \cdot \int_x^{-\infty} J(x')dx' \tag{2.7}$$

so that

$$BER(x) = BER_L(x) + BER_R(x) \tag{2.8}$$

The eye opening at a given BER, t(BER), is given by the separation of the left and right BER curves a given BER. For example, in fig.2.12, the eye opening at $BER = 10^{-12}$, is given by the difference of x_L and x_R-those points where $BER = 10^{-12}$. If we invert $BER(x)$, then

$$t(BER) = x_R(BER) - x_L(BER) \tag{2.9}$$

earlier.

TJ(BER) is the difference in the bit period and the eye opening:

$$TJ(BER) = T_B - t(BER) \tag{2.10}$$

2.8.2 The Dual-Dirac Model

To calculate the total jitter, DJ could be approximated by a dual-Dirac function [13] [14] that assumes the density function consist of only a pair of delta functions. The *dual-Dirac Model* is an useful tools to estimating the total jitter at a bit error ratio. It is based on the following assumptions:

- Jitter can be separated into two categories, random jitter (RJ) and deterministic jitter (DJ).
- RJ follows a Gaussian distribution and can be fully described in terms of a single relevant parameter, the rms value of the RJ distribution or, equivalently, the width of the Gaussian distribution, σ.
- DJ follows a finite, bounded distribution.
- DJ follows a distribution formed by two Dirac-delta functions. The time-delay separation of the two delta functions gives the dual-Dirac model-dependent DJ, as shown in fig.2.13.

The first three assumptions have been described in the previous sections. The fourth one provides a model which is universally accepted for its utility to quickly estimate the total jitter defined at a bit error ratio. The dual-Dirac models, provides the simplest possible distribution and it is described by the following equation:

$$PDF_{(DJ\delta\delta)} = \frac{1}{2}[\delta(x - \mu_L) - \delta(x - \mu_R)] \tag{2.11}$$

The crossing point is separated into two Dirac delta functions positioned at μ_L and μ_R, the DJ dominated region, followed by an an artificially abrupt transition to the RJ dominated tails.

To calculate the total jitter on a signal, the components of the jitter must be combined to obtain the expected BER performance.

$$J(x) = RJ(x) \otimes DJ(x) \tag{2.12}$$

Q	BER
6.4	10^{-10}
6.7	10^{-11}
7.0	10^{-12}
7.3	10^{-13}
7.6	10^{-14}

Table 2.1: Values of Q_{BER}, the multiplicative constant for determining eye closure due to RJ, for different BER values.

where x is the time delay. But the PDF of the Random Jitter is described by a Gaussian whose width is σ.

$$RJ(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot e^{-\frac{x^2}{2\sigma^2}} \tag{2.13}$$

Replacing eq.2.11 and eq.2.13 in the eq.2.12

$$J(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \otimes \frac{1}{2}[\delta(x-\mu_L) - \delta(x-\mu_R)] \tag{2.14}$$

The convolution of the sum of two delta functions separated by DJ and a Gaussian RJ distribution of width σ is shown in fig.2.13. The results is composed of two Gaussian of width σ separated by a fixed amount $DJ(\delta\delta) \equiv |\mu_L - \mu_R|$.

$$J(x) = \frac{1}{\sqrt{2\pi\sigma}} \cdot (e^{\frac{(x-\mu_L)^2}{2\sigma^2}} + e^{\frac{(x-\mu_R)^2}{2\sigma^2}}) \tag{2.15}$$

The eye closure is composed of a fixed amount, $DJ(\delta\delta)$, and an amount that depends on the bit error ratio of interest. Once σ and $DJ(\delta\delta)$ are measured the eye closure at any BER can be estimated with:

$$TJ(BER) = 2Q_{BER} \times \sigma + DJ(\delta\delta), \tag{2.16}$$

where Q_{BER} is calculated from the complementary error function and $DJ(\delta\delta) < J_{pp}^{DJ}$. Since σ is multiplied by $2Q_{BER}$, the accuracy of the TJ(BER) depends first on the accuracy of the RJ measurement, σ, and second on the accuracy of the $DJ(\delta\delta)$ measurements. For bounded DJ(x) the asymptotic behavior of J(x) is the same as that of a Gaussian,

$$lim_{x\to-\infty} J(x) \to Ae^{-\frac{(x-\mu_L)^2}{2\sigma^2}} \tag{2.17}$$

$$lim_{x\to+\infty} J(x) \to Ae^{-\frac{(x-\mu_R)^2}{2\sigma^2}} \tag{2.18}$$

so it is possible to estimate TJ(BER) using eq.2.6 and eq.2.7 through eq.2.10.

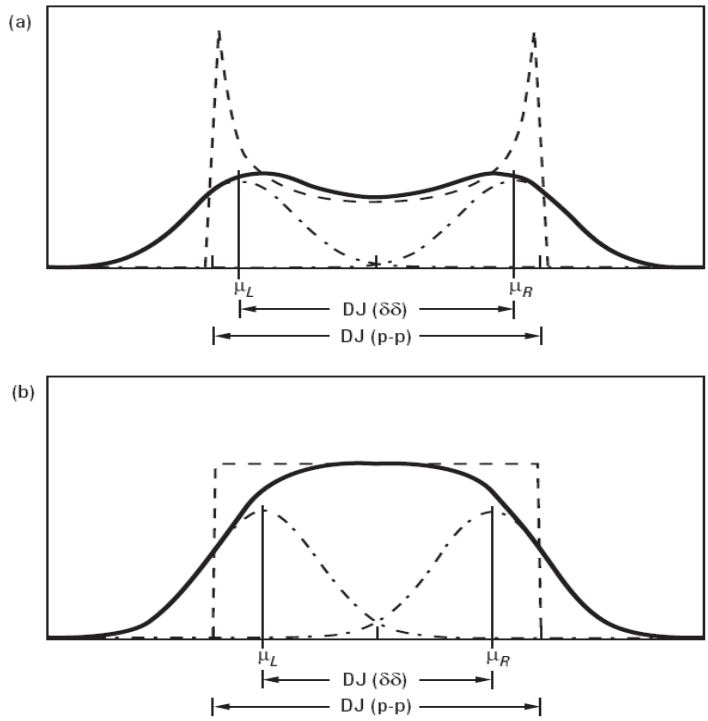

Figure 2.14: Application of the dual Dirac model to two ideal cases. In (a) the dashed DJ distribution is caused by a single frequency of a sinusoidal jitter and in (b) the dashed bounded, constant (square wave) DJ distribution could be caused by a triangle wave phase modulation. The solid curve is the convolved jitter distribution and the dash-dot lines are the dual Dirac approximation. The vertical line indicate where the dual-Dirac model sets the means of Gaussian, μ_R and μ_L.

In fig.2.14a the dashed DJ distribution is given by a single frequency of sinusoidal jitter and, in fig.2.14b, by a flat, bounded DJ distribution. The DJ distributions are convolved with a Gaussian resulting in the smooth solid curves. The effect of the convolution is to smooth the sharp edge of the DJ distribution and J(x) obtains its Gaussian tails. The vertical lines in fig.2.14 are set μ_R and μ_L, demonstrating the inequality, $DJ(\delta\delta) = |\mu_R - \mu_L| < J_{pp}^{DJ}$. The two Gaussian curves (dash-dot) give the dual -Dirac approximation to

the solid curve. Its central part doesn't match the actual distribution but the important features is that Gaussian tails match the tails of the true jitter distribution as in eq.2.17 and eq.2.18 so that TJ(BER) can be estimated using eq.2.6 and eq.2.7 through eq.2.10.

Chapter 3

The DDR II tester

3.1 Introduction

As discussed in the first chapter, DDR II is the highest-bandwidth memory available today, supporting the threaded applications, improved graphics and multitasking often found on the most advanced PCs and workstations. These memories are elaborately tested by their manufacturers to ensure high quality product for the consumer.

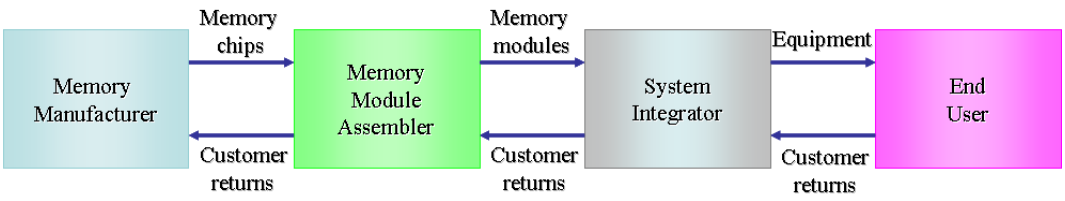

Figure 3.1: Supply chain of memory devices.

There are four major parties involved in the supply chain of memory devices, and each one needs to test the memory in a different way:

- the memory chips manufacturers,
- the memory modules assemblers,
- the system integrators,

- the end user.

Fig.3.1 graphically represents these four parties involved and the way they interact. A *memory chip manufacturer* is the party involved in defining the specifications of memory devices and subsequently designing and manufacturing row memory device of the form shown in fig.3.2. The *memory modules*

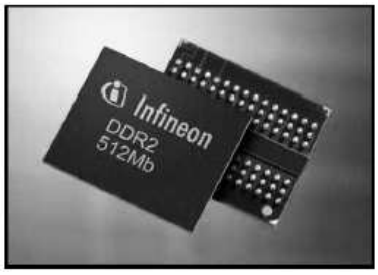

Figure 3.2: Memory chip in TBGA package (source: Infineon Technologies).

assembler is the party that buys memory devices from a memory manufacturer and implements them into memory modules (fig.3.3) that will be bought by the *system integrator* to realize more complex system such as PCs, laptops, workstations or networking equipment.

Figure 3.3: Memory module.

The *end user* is the party that acquires the equipment provided by the system integrator in order to deploy it to solve a specific problem. The big burden of extensive testing and qualification of memory devices rests on the shoulder of the memory chip manufacturer, the first party in the memory supply chain. The test done in this phase can be divided in two categories:

- *frontend* (or wafer level) testing and,

- *backend* (or component level) testing.

Frontend testing is performed before chip packaging so that only functional chips get to the packaging process, in order to reduce the costs of packaging. Backend testing ensures that packaged chips function properly before delivery to the customer. The backend test can be further subdivided in more phases:

- *Burn-in Test* which is a well-known method to check the reliability of manufactured components by applying highly stressful operational conditions, called *test stresses* (such as high voltages, possibly combined with high temperature), which accelerate the aging process of the memory,
- *At Speed Test* which has as main objective to check chip functionalities at the nominal frequency,
- *Parametric Test* which is aimed at the validation of electrical parameters (leakage currents, voltage level, rise and fall time, etc.) according to the specifications.

The memory module assembler performs both the test of the memory device, to verify the specifications declared by the manufacturer, and the test of assembled system (checking of the connections, verifying of the terminations etc.).

The system integrator tests the whole system extensively before delivery to the end user, because it is expected, in this point of the chain, a very high quality level. Referring to the chip, after the testing process, if the memory device are found to cause equipment failure, the defective memories are sent back to the manufacturer in the form of *customer returns*. The manufacturer is then expected to investigate these returns and to try to screen them out before they are sold to the memory module assembler.

Finally the end user is not expected to perform any specific testing on acquired systems other than setting them up for regular operation. Here too, the end user sends back defective systems to the integrator in the form of customer returns for reparation or replacement.

A well-known industrial rule of thumb (sometimes referred to as the *rule of tens*) on the relative cost of system testing states that *at each successive stage in production, it cost ten times as much to identify and repair a defective component than it would have cost at a previous production stage*:

$$Cost_{stage(i+1)} = 10 \times Cost_{stage(i)} \tag{3.1}$$

Fig. 3.4 gives a graphical representation of this rule. For this reason, the

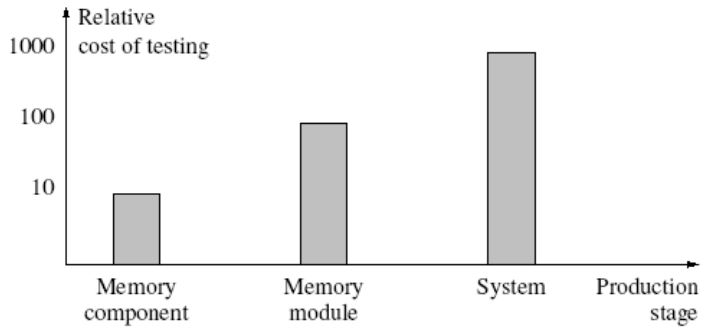

Figure 3.4: Relative cost of component testing at each stage in production.

manufacturer have to elaborately test their memory chips in order to ensure the very high quality requirements expected from memories. However, this testing process is responsible for a large portion of the cost of these memories, standing now at 40% and gradually rising with each new generation. Memory testing is expensive because of the high cost of the test equipment (a production memory tester costs more than 500.000 Dollars), a cost that has to be distributed over all the produced chips.

In this thesis work, a new alternative approach to the development of industrial memory testing is proposed. It is more complete, more flexible and less expensive than the currently prevalent test approach. This tester can be used in some phase of manufacturer testing and in all phases of memory module assembler testing.

As shown in the second chapter at high frequency, such as those of DDR II memory, the signal integrity becomes an important issue. However the equipments available on the market are not able to perform both functional test and signal integrity analysis. The study of the signal integrity, further increase the production time and costs. On the contrary, the main features of the realized system are:

- *reduced test time*, because it support up to 64 channels simultaneous signal integrity testing,
- *reduced cost*, because it is based on a programmable device which implements 90% of needed functionalities.

The board, developed in the XS-consulting laboratory, is able to:

- perform the functional test allowing the implementation of any algorithm, such as that described in par. 1.7, with high flexibility,
- perform the signal integrity analysis, implementing on each memory channel, the function of a sampling oscilloscope. This feature performing complex analysis without the use of an external and expensive equipment. The complex analysis include the eye opening, rise and fall time, setup and hold time measurements. Moreover it is possible, without additional circuitry, to perform the OCD calibration.
- The voltage, the frequency and the refresh rate can be varied over specified ranges. This feature allows the reconstruction of the *Shmoo plot* which is the graphical display of the response of a component over a range of conditions and inputs. Often it is used to represent the results of testing of complex electronic system such as ASICs or microprocessors. The plot shows the range of conditions in which the device under test will operate.

  The plot is the two dimensional graphical representation of the pass/fail behavior of the memory under the applied test. Fig.3.5 shows an example of a Shmoo plot, where the x-axis represents the clock cycle time and the y-axis represents the supply voltage V_{dd}. The figure shows, for example, that a lower voltage and a shorter cycle time are the most stressful conditions for the applied test.

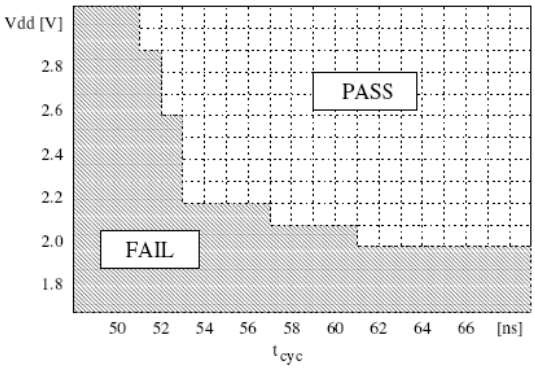

Figure 3.5: Shmoo plot showing the cycle time and supply voltage.

3.2 Board description

Fig.3.6 and fig.3.7 show respectively the top and bottom of the board. As

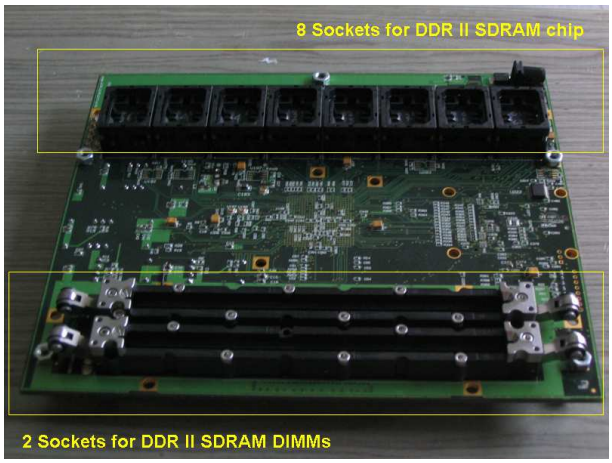

Figure 3.6: Tester board:top view.

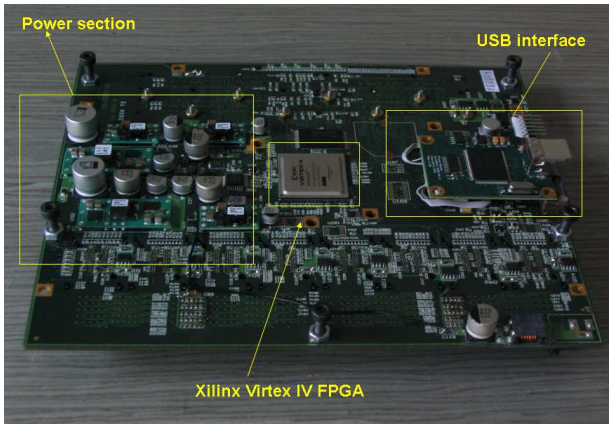

Figure 3.7: Tester board:bottom view.

shown it can supports:

- *DDR II SDRAM memory chips:* it is able to test up to eight 8 pins chips or up to four 16 pins chips,

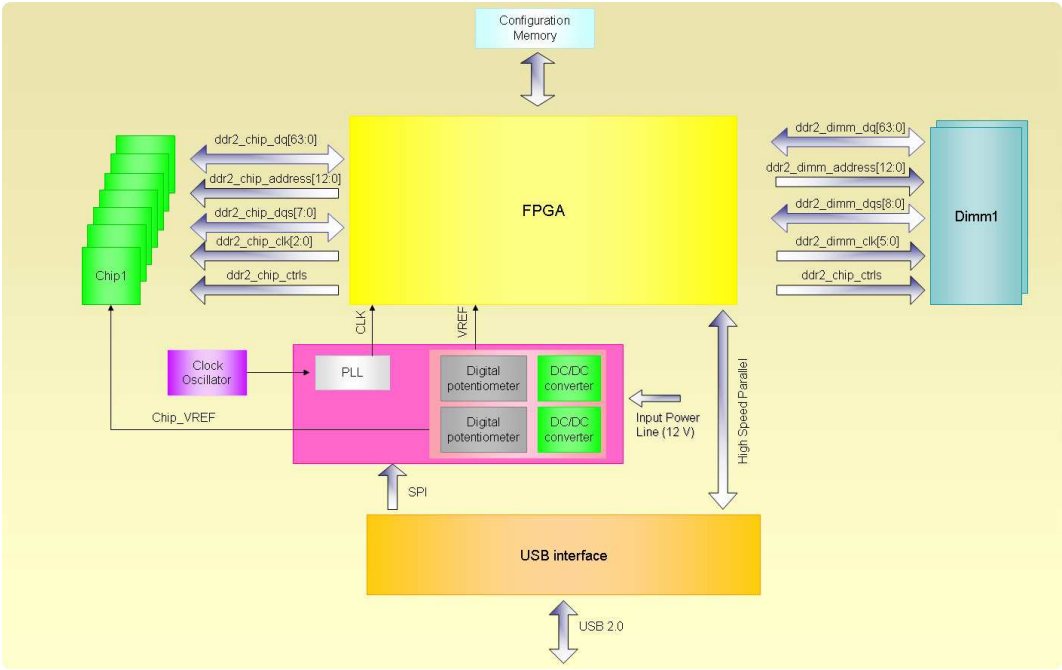

Figure 3.8: Block diagram of the DDR II tester board.

- *DDR II SDRAM memory module:* it is able to test up to two 72 pins memory modules.

Fig.3.8 details the architecture of the board. A complete system-on-chip approach has been followed in the design of tester. All the functions are implemented into a *Field Programmable Gate Array* (FPGA). This allowed us to reduce the hardware resources and the board dimension and to achieve very high frequency. The FPGA, are quite complex matrices of logical gates, memories and registers with user programmable interconnections. This technology not only promises new levels of system integration onto a single chip, but it also allows for more features and capabilities in a reprogrammable technology.

The FPGA supplies data, data strobes, controls, addresses and clocks to the memory chips and memory modules. The memory clocks are synchronous to the FPGA clock.

The board is controlled through a commercial USB interface (QuickUSB QUSB2 plug in module). The QuickUSB QUSB2 Plug-In Module is a 2" $\times$ $1^{1/2}$" circuit board that implements a bus-powered Hi-speed USB 2.0 endpoint

terminating in a single 80-pin target interface connector. The target interface is shown in the fig.3.9.

As just discussed the board is able to perform any algorithm such as that described in the par. 1.7. It receives the March element through the USB interface. The data are moved, through the High Speed Parallel bus, to the FPGA where they are elaborated. The test result is sent back to the USB interface.

The voltage and frequency of the DDR II chip and DDR II modules can be varied on a specific range. This feature allows the parametrical analysis as discussed above. The voltage can be varied by the use of a digital potentiometer controlled through the SPI port (fig.3.8). The frequency of the chips can be changed, varying the FPGA clock by the use of an external PLL. All the board function are synchronized with respect to a local clock signal generated by a programmable PLL driven by a 16MHz oscillator. The PLL is programmable through the SPI port.

Also the Vref of the FPGA is programmable by the use of a digital potentiometer through the SPI port. This feature allows the innovative implementation of the sampling oscilloscope that will be described in detail in the following.

The tester board is powered through a 12 V power input line; five non-isolated DC/DC converters derive the voltage source required by the electronic components mounted on the board.

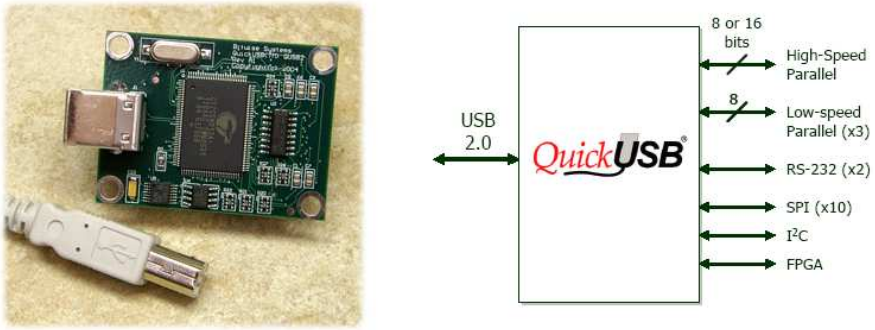

Figure 3.9: Picture of the QuickUSB QUSB2 Plug-In Module and it block diagram.

The following table reports the tester specifications from the electrical and mechanical point of view (area, weight, power consumption):

- 200×170 mm^2,
- 150 g,
- 1 W.

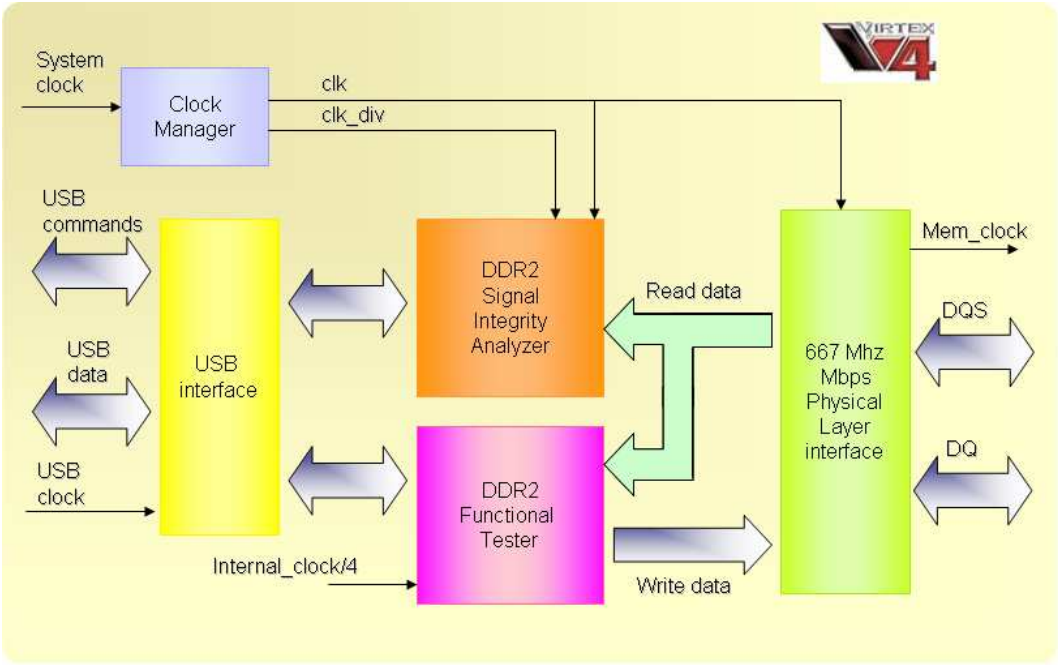

Figure 3.10: Block diagram of the DDR II tester fpga.

The FPGA used is a VIRTEX 4 device. Manufactured in 1.2v, 90nm triple oxide technology, VIRTEX-4 FPGA offers twice the performance, twice the density (it contains up to 200.000 gates) and less than one-half the power consumption of previous generation device. It has up to 80 independent clocks and 20 digital clock managers. Differential global clocking minimizes skew and jitter. It has up to 10 Mbits Block RAM with built-in FIFO control logic. The VIRTEX 4 makes easy to build robust high speed memory interfaces by the use of the ChipSync^{TM} technology. This technology provides dedicated high precision circuits for simplified source synchronous interface. More details about this technology will be done in the following paragraphs.

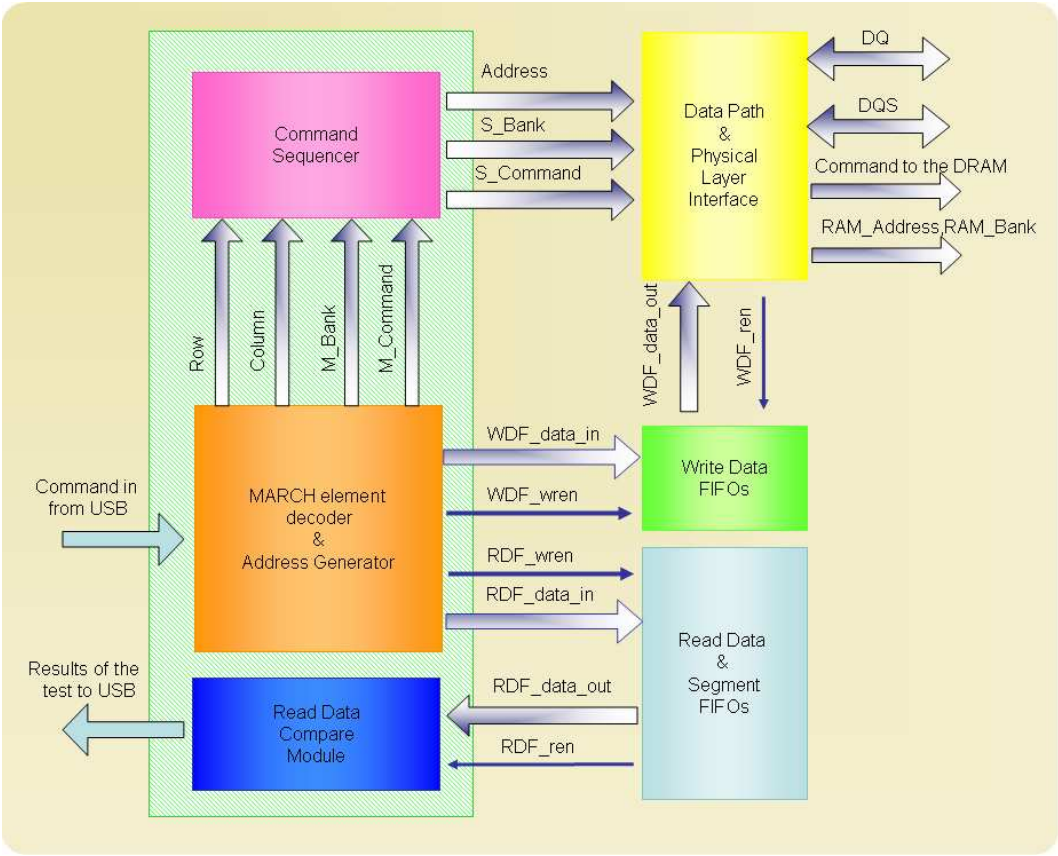

Figure 3.11: Block diagram of the DDR II tester fpga.

3.3 FPGA description

Fig.3.10 shows details about the internal architecture of the FPGA. The DDR II functional tester receives in input from the USB interface the Macro instruction which represent the March element, and gives as output the result of the test. The Signal integrity analyzer receives in input the data and dqs from the DDR II, samples the signals and reconstructs the eye diagram. The Physical layer is responsible for transmitting and receiving all the signals to and from the memories. The clocking scheme for this design includes one DCM (Data Clock Manager) and one PMCD (Phase-Matched Clock Divider). The internal logic is clocked at half the frequency of the interface. This feature has allowed us to test chip up to 400MHz (800Mb/s).

The design of the FPGA has been driven by the following guidelines:

- reduce the resources consumed into the FPGA,
- work at the high frequency.

This results in high pipelined and parallel architecture. The overall design require 2450 flip-flop and 3745 four input look up table; these correspond to the 12% and 18% of the available resources. The architecture runs internally up to 200 MHz clock frequency.

3.4 The functional tester

Fig.3.11 shows details about the functional tester. It receives in input the sequence of commands (WRs and RDs) from USB interface and gives back the result of the test.Moreover, it provides commands, data and addresses to the SDRAM. Tab.3.1 lists the commands issued by the controller.

Command	**Row Address Select**	**Column Address Select**	**Write Enable Signals**
Load Mode	L	L	L
Auto Refresh	L	L	H
Precharge [1]	L	H	L
Bank Activate	L	H	H
Write	H	L	L
Read	H	L	H
No Operation/IDLE	H	H	H

Table 3.1: Commands issued by controller

The commands are detected by the memory using the following control signals:

- Row Address Select ($\overline{RAS}$)
- Column Address Select ($\overline{CAS}$)

- Write Enable ($\overline{WE}$) signals
- Clock Enable (CKE) held High throughout and after device configuration
- Chip Select ($\overline{CS}$)

Moreover the tester initializes the memory because the DDR2 SDRAMs must be powered up and initialized in a predefined manner. During this procedure four internal register of the DDR II are set:the *Mode register*, the *Extended Mode Register*, the *Extended Mode Register (2)*, the *Extended Mode Register (3)*. The Mode Register, shown in fig3.12, stores the data for controlling the various operating mode of DDR II SDRAM. It controls *CAS latency*, which is the delay in clock cycles between the registration of read command and the availability of the first bit of output data, *Burst length*, which is the maximum number of locations accessed for a given read, and various vendor specific option to make DDR II SDRAM useful for various applications. The Extended Mode Register, shown in fig.3.13, stores the data for enabling or disabling the DLL, the output drive strength, the ODT, the differential DQS, the OCD. The Extended Mode Register(2) controls refresh related features. The Extended Mode Register (3) is reserved for future use.

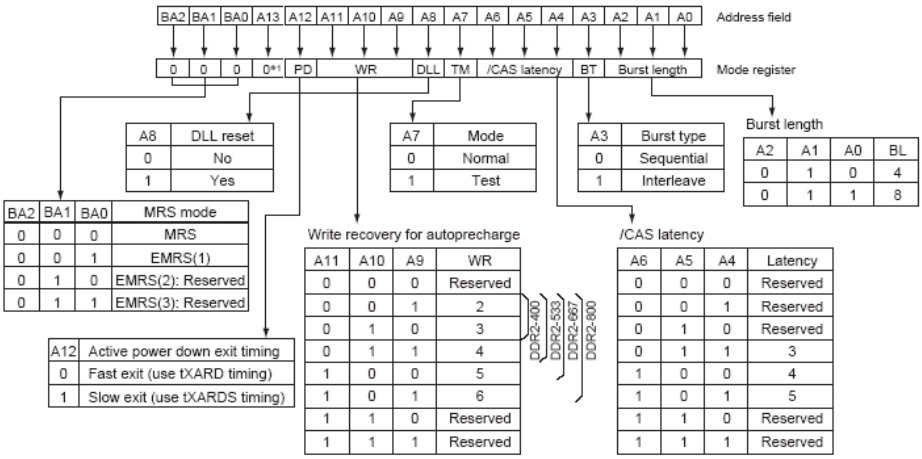

Figure 3.12: Mode register.

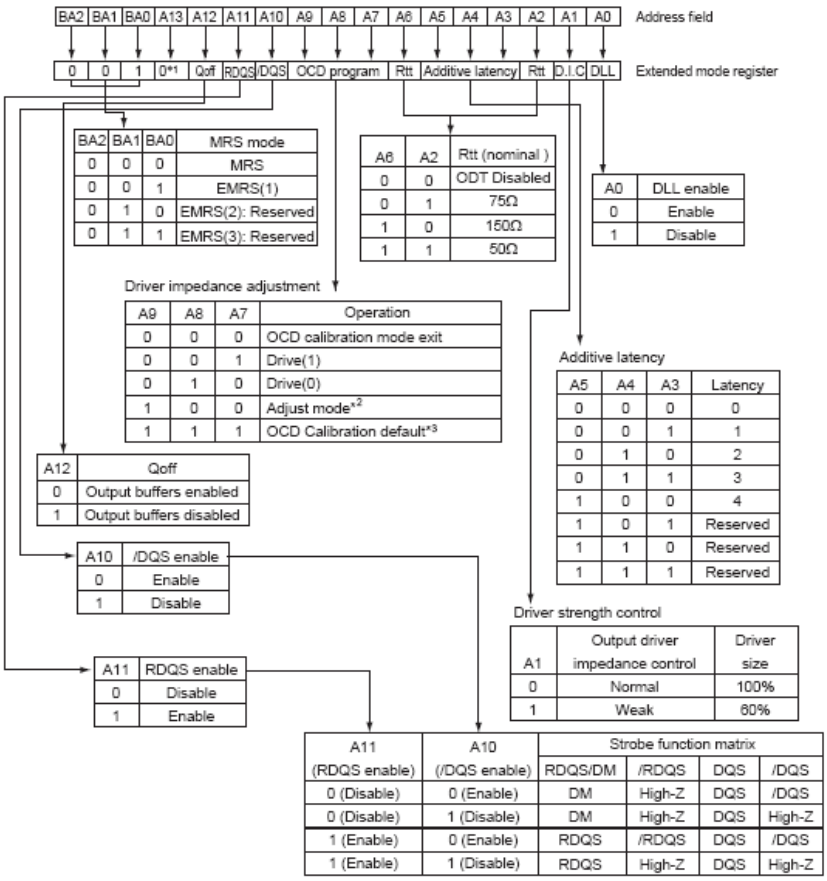

Figure 3.13: Extended Mode register.

3.4.1 March Element Decoder

The *March Element Decoder*(MED) receives in input the march element, decodes it and provides the commands, the addresses and data to the next logic stage. Its block diagram is shown in fig.3.14. Referring to the picture, the Configuration Block is composed of five registers:

- the *Operation Code Register*, which defines the operations belonging to the March element. The bit 0 represents the first operation to be done, the bit 1 the second and so on. If Operation Code Register(i)= '1' the i-th operation must be done is a WRITE , else it is a READ.

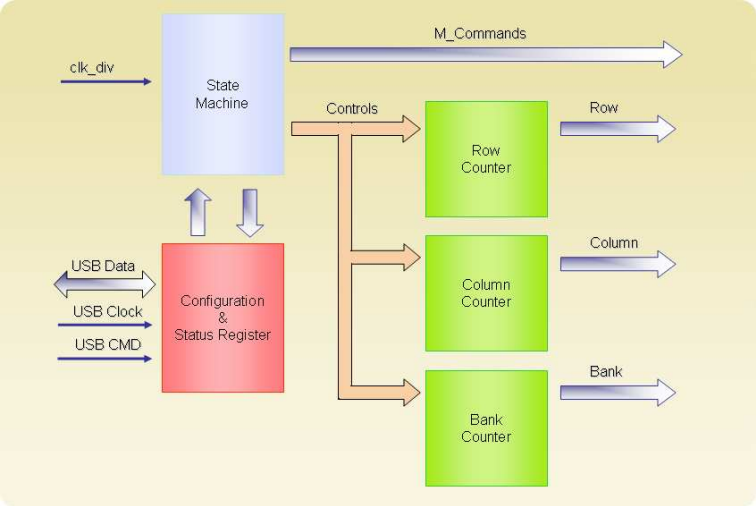

Figure 3.14: Block diagram of the March Element Dedoder.

- the *N Operation Register*, which specifies the number of operations belonging to the March element,
- the *Pattern Register*, which contains the pattern must be write or expected in a READ operation.
- the *Pattern Code Register*, which defines the polarity of the pattern related to each operation in the ME. The bit0 represents the polarity of the pattern related to the first operation to be done, the bit1 represents the polarity of the pattern related to the second operation and so on. If Pattern Code Register(i)='1' the pattern related to the i-th operation is $\overline{pattern}$, else it is *pattern*.
- the *Mode Register*, which specifies the *Functional Mode* and the address order (UP or DOWN).

The five registers contribute to define the March element. The State Machine sends to the next stage the operations specified in the operation code, then it increases or decreases the counters according to the functional mode. There are two functional modes:

- *Column Fast access*: for each row address the bank address increases up to the maximum value, for each bank address the column address increases up to the maximum value.

- *Row Fast access*: for each column address the bank address increases up to the maximum value, for each bank address the row address increases up to the maximum value.

3.4.2 Command Sequencer

The *Command Sequencer*(CS) (fig.3.15) receives in input the row address, the column address, the bank address and the commands from MED. As just

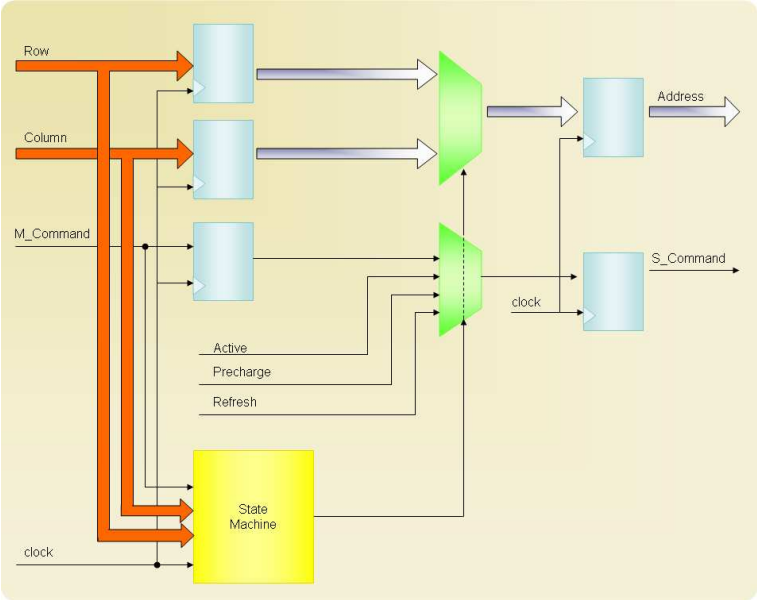

Figure 3.15: Block diagram of the Command Sequencer.

discussed in the previous chapter, Read and Write accesses to the DDR II SDRAM device are burst oriented. The burst length which determines the maximum number of locations accessed for a given read is programmable and its possible values are four and eight. The address bits registered with the ACTIVE command are used to select the bank and row to be accessed. The address bits registered with the read or write command are used to select the bank and the starting column location for the burst access. According to these specifications the CS activates a row in the corresponding bank if all banks have been precharged, or it compares the bank and row addresses to the already open row address and bank address.

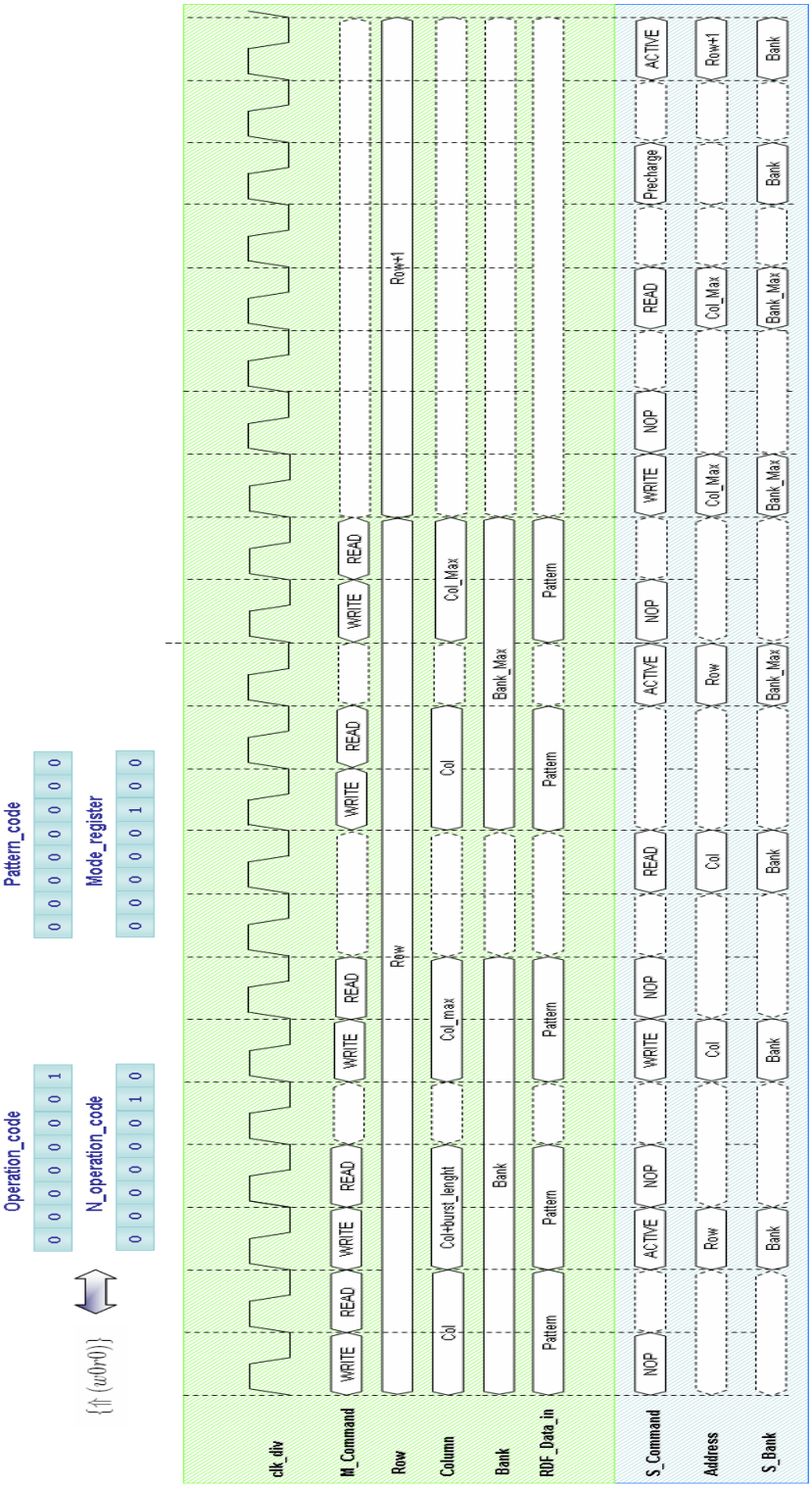

Figure 3.16: Timing Diagram of the March Element Decoder and Command Sequencer operations.

Parameter	Description	t	Units
T_{RP}	PRECHARGE Command Period	15	ns
T_{RFC}	REFRESH to ACTIVE or REFRESH to REFRESH Command Interval	75	ns
T_{RCD}	ACTIVE to READ or WRITE Delay	15	ns
T_{RAS}	ACTIVE to PRECHARGE Command	40	ns
T_{RC}	ACTIVE to ACTIVE (Same Bank) Command	55	ns
T_{RTP}	READ to PRECHARGE Command Delay	7.5	ns
T_{WTR}	WRITE to READ Command Delay	7.5	ns
T_{WR}	WRITE Recovery Time	15	ns

Table 3.2: Timing Requirements

If there is a conflict, it introduce a PRECHARGE command to precharge the open bank and then issue an ACTIVE command. Before to issue any commands the Command Sequencer checks timing requirements according to the tab.3.2. These parameters are fully programmable through tester. They are strictly dependent to the choose component and to its speed grade. The tab.3.2 refers to a 267 MHz DDR II.

Referring to the Fig.3.16, in the green box are shown the timing of the signal related to the MED, in the blue box those related to the CS. The functional mode chosen is the Column Fast Access with increasing address order. The March element that must be done is $\{\Uparrow (w0r0)\}$. It corresponds to the the registers settings shown in the picture. The MED has to issue a command to write the predefined pattern to a given memory location. After the write operation is complete, data has to be read back from the same location and compared with the written data. An ACTIVE command is issued to activate the row in the specified bank. After the row has been opened, the WRITE command has been issued to that row, subject to the t_{RCD} specification. When the CS detects an incoming address referring to a row in a bank other than the currently opened row, the controller issues a PRECHARGE command to deactivate the open row. It also also issue another ACTIVE command to the new row (fig.3.16).

The PRECHARGE command is used to deactivate the open row in a

particular bank. The bank is available for a subsequent row activation for a specified time (t_{RP}) after the PRECHARGE command is issued. The $bit10$ of the bus address determines whether one or all banks are to be precharged.

DDR II devices are required to be refreshed every $7.8\mu s$. The circuit to flag the refresh command is built into the CS. The controller issues the refresh command after it completes the transactions in the current open bank.

3.4.3 The Data Path and Physical Layer Interface

The Data path module is responsible for transmitting to and receiving data from the memories. Major functions include:

- Writing data to the memory,
- Reading data from the memory,
- Transferring the read data from the memory clock domain to the FPGA clock domain,
- Providing all the necessary control signals to the memory.

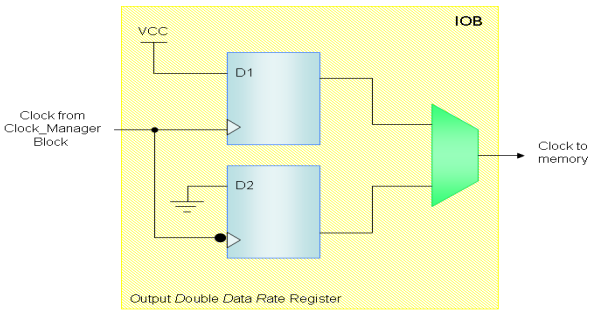

Figure 3.17: Clock generation for memory device.

The FPGA generates all the clocks and control signals for reads and writes to memory. As shown in the fig.3.17 the memory clocks are generated using a Double Data Rate (DDR) register, there is only one clock input to the ODDR. Falling edge data is clocked by a locally inverted version of the input clock. In the fig.3.18 are shown the timing related to the control signals generation.

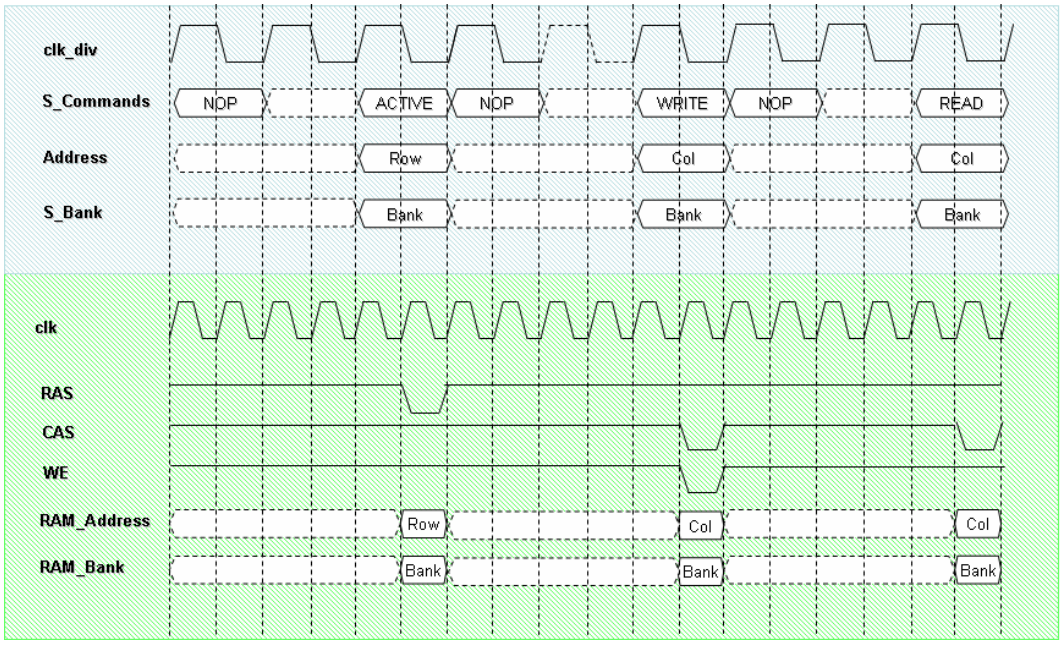

Figure 3.18: Timing Diagram related to control signals generation.

The Write Data and strobe are clocked out of the FPGA. The strobe must be center-aligned with respect to the data (3.19). For DDR II memories, it could be differential and non-free running, i.e. it is required only when valid data is being written into the memory. The design supports differential and non-differential data strobe. To meet the requirements specified above the write data is clocked out using a clock that is shifted 90° from the primary clock going to the memory and the data strobes are generated out of primary clocks going to the memory.

The read and write data path uses the Input Serializer/Deserializer (ISERDES) and Output Serializer/Deserializer (OSERDES) features available in every Virtex-4 I/O. The write data and strobe transmitted by the FPGA use the OSERDES (fig.3.20). The OSERDES transmits the data (DQ) and data strobe (DQS) signals. The OSERDES is a dedicated parallel to serial converter with specific clocking and logic resources designed to facilitate the implementation of high-speed source synchronous interface.The OSERDEDS converts 4-bit parallel data at half the frequency of the interface to DDR data at the interface frequency.

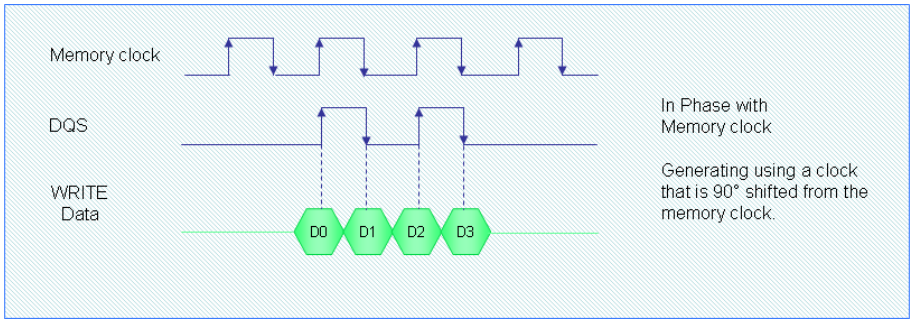

Figure 3.19: Write Timing Diagram.

A DDR2 SDRAM interface is source-synchronous, where the read data and read data strobe are transmitted edge-aligned (fig.3.21). To capture this transmitted data using Virtex-4 FPGAs, either the strobe or the data can be delayed. In this design, the read data is captured in the delayed strobe domain and recaptured in the FPGA clock domain in the ISERDES. The received signal, double data rate (DDR) read data, is converted to 4-bit parallel single data rate (SDR) data at half the frequency of the interface using the ISERDES. The read data is captured in the delayed strobe domain and recaptured in the FPGA domain. To capture read data without errors the dqs must be delayed to center in the read data window. Both read and data are further delayed to align to the fpga clock domain. To delay the data strobe and read data the FPGA uses a 64 taps absolute delay elements built into each IOB, called IDELAY block. The resolution of each tap ia approximately $75ps$ over process, voltage and temperature. This feature provide flexibility and makes read capture easy. Fig.3.22 shows the block diagram of read data path module.

To ensure reliable data capture in the DQS domain and recapture in the FPGA clock domain in the ISERDES, a training sequence is required. This sequence is performed by the calibration block. The controller issues a WRITE command to write a data pattern to a specified memory location. The controller then issues a back to back read commands to read back the written data from this specified location. The outputs of the ISERDES are compared with the known data pattern. If they don't match, DQS is delayed by one tap, and then the comparison is performed again. The tap increments until there is a match. With the first detected match, the DQS count is incremented to 1. DQS continues to be delayed in unit tap increments until a mismatch is

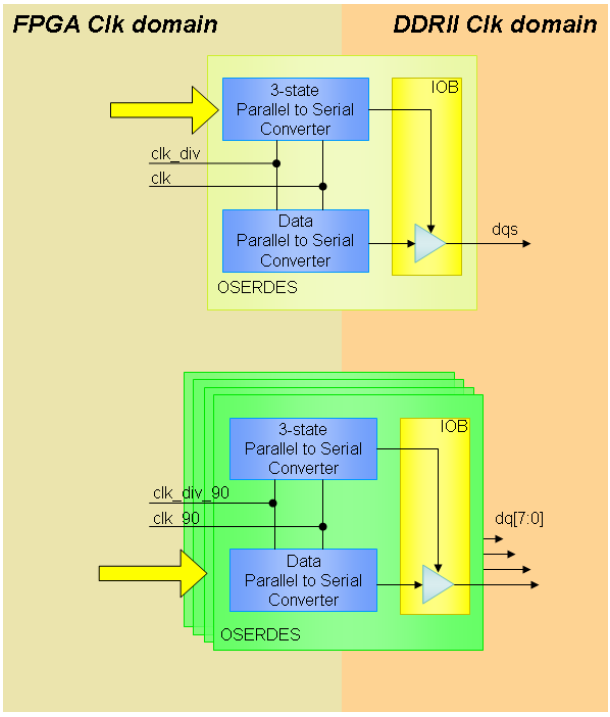

Figure 3.20: Write data path block diagram.

found. The DQS count is also incremented along with the tap increments to record the width of the data valid windows. DQS is then decremented by half the window count to center DQS edge to the center of the data valid window. Fixed the position of the DQS to respect to the DQ valid window, both data and strobe are delayed following the same procedure described above. This last step fix the phase of the DQ-DQS to respect the FPGA clock. The calibration block also computes the delay in FPGA clock cycles between the registration of the read command and the first bit of data available. This parameter is used in the following read commands.

3.4.4 The Read Data Compare Module

The block diagram of the Read Data Compare Module is shown in fig.3.23. This block receives in input the data from the Physical Layer Interface and compares it to the expected pattern stored in the Read Data Fifo. Beyond the Read Data Fifo there is the Segment data fifo. The two fifo are read si-

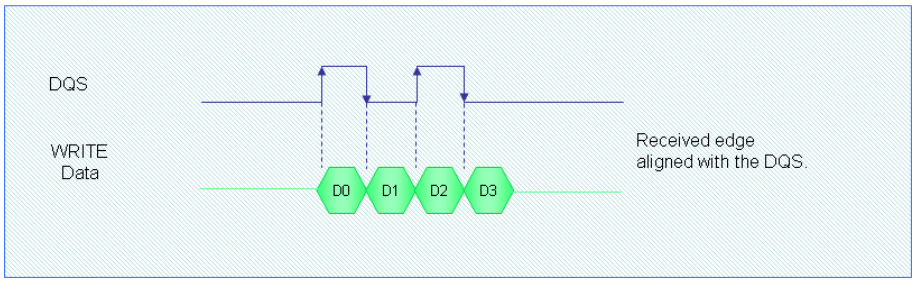

Figure 3.21: Read Timing Diagram.

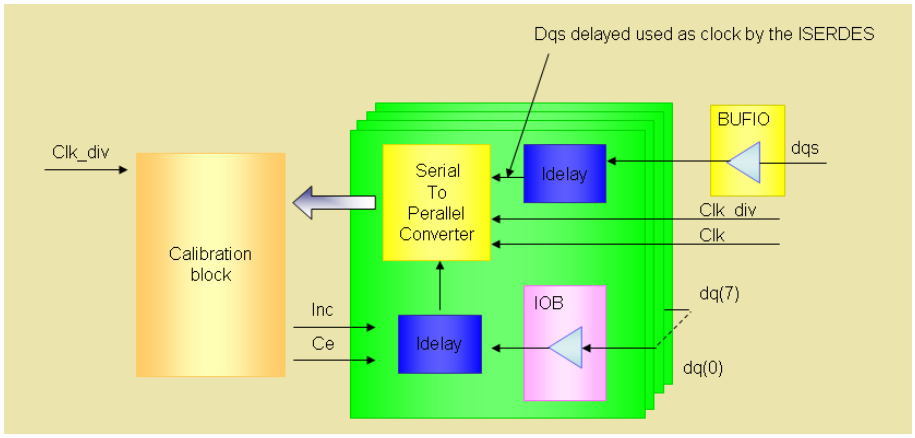

Figure 3.22: Read data path block diagram.

multaneously. The segment data identifies the segment whose the data belong to. The segment is also the address of an internal RAM which contains the accumulated errors.

3.5 The Signal Integrity Analyzer

As just discussed in the first chapter the digital errors often have their roots in analog signal integrity problems. To track down the cause of the digital fault, it is often necessary to turn to an oscilloscope, which can display waveform details, edges and noise; can detect and display transients and can help to precisely measure timing relationships such as setup and hold time.

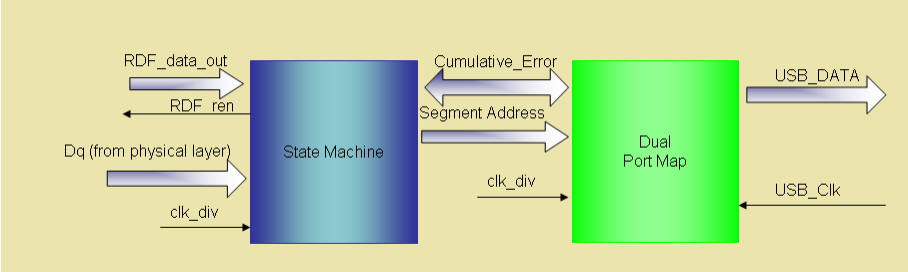

Figure 3.23: Read Data Compare Module.

The signal integrity analyzer (SIA) is designed to provide, on each memory channel, the functions of a sampling oscilloscope. The oscilloscope is basically a graph-displaying device, it draws a graph of an electrical signal showing how the signal change over time: the vertical (y) axis represents voltage and the horizontal (x) axis represents the time. As a traditional oscilloscope it creates a waveform point, which is the digital value stored and displayed to construct the waveform, by saving one sample point during each waveform interval. The magnitude of each sample points is equal to the amplitude of the input signal at the time in which the signal is sampled. In the following the basic principle used in this application will be described. The SIA constructs a picture of a signal by making the scanning both in time domain and in voltage range. Referring to the fig.3.24, the SIA, fixed the threshold , performs a time scanning.

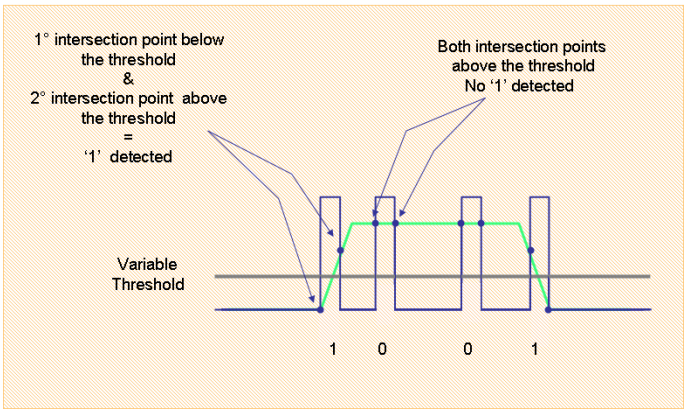

Figure 3.24: Timing Diagram: basic principle used in the Signal Integrity Analyzer.

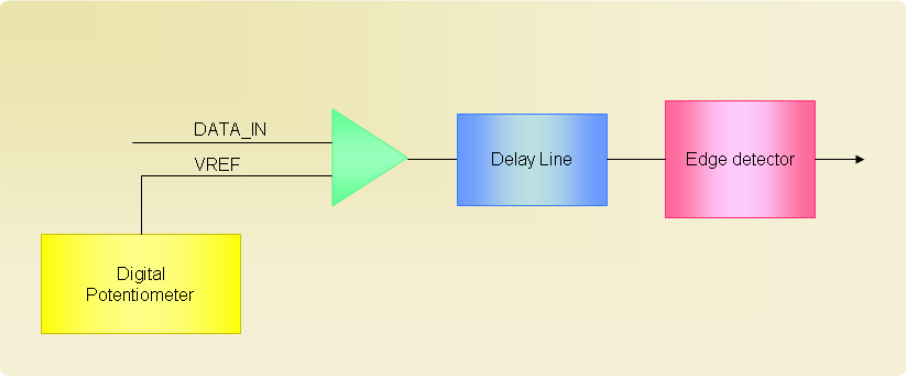

Figure 3.25: Block diagram: basic principle used in the Signal Integrity Analyzer.

A window runs along the time axis.A logic one is detected when the right (left) extreme of the window is below the threshold and the left (right) extreme is above it. When the scanning is complete, the threshold is increased and a new timing scanning performed. The waveform reconstruction is obtained connecting the detected '1s'. However, when measuring high frequency signals, the SIA may not be able to collect enough samples in one sweep. This can be resolved by taking advantage of the fact that the DQ and DQS, which are the signals under study, are repetitive signals. This method constructs a picture of a repetitive signal by capturing a sample of the information from each repetition. Summarizing the basic element to realize an oscilloscope are (fig.3.25):

- a *comparator* to compare the input signal with the threshold,
- a *digital potentiometer* to change the threshold to perform the vertical scanning,
- a *programmable delay line* to perform the horizontal scanning,
- an *edge detector* to identify the '1s' and '0s'.

The Virtex-4 device embodies all the components listed above. Fig.3.26 shows the block diagram of the Signal Integrity analyzer.

The time shifting and the edge detection are performed by the modules in the blue box. The Data Clock Manager allows a fine grained phase shifting, i.e., the DCM output clocks can be phase shifted with respect to the input

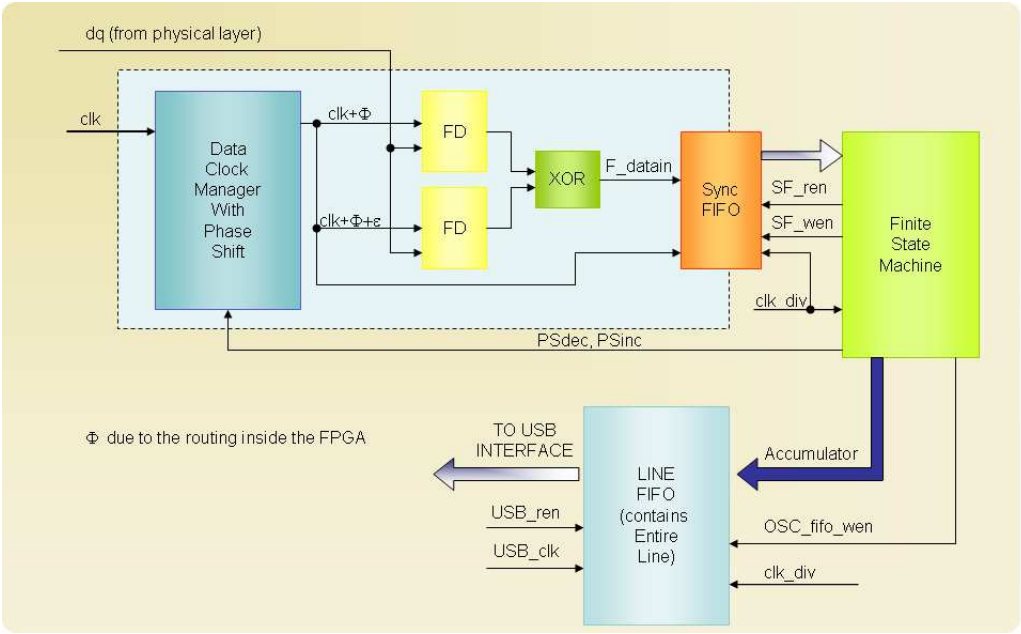

Figure 3.26: Block diagram of the signal integrity analyzer.

clock. The phase shift is programmed by the state machine through a control signal named PSINC. A further delay is introduced by the routing of the two flip flops inside the FPGA. As result one flip flop is clocked with $clock + \Phi$ and the other one is clocked with $clock + \Phi + \epsilon$, ϵ due to their different position inside the FPGA. To obtain a known ϵ the two flip flops are manually routed.

The voltage scanning is obtained varying the Vref of the FPGA. Fig.3.27 shows the timing diagram related to the edge detection. Referring to the top side of the picture, the rising edge of the $clock + \Phi$ corresponds to a voltage level under threshold and thus the logic level of the flip flop output is '0', the rising edge of the $clock + \Phi + \epsilon$ corresponds to a voltage level over the threshold resulting in a logic '1' at flip flop output. In this case an edge is detected as shown in the picture. Referring to bottom of the picture, where the Φ is unchanged, but the threshold has been moved to the top, both rising edges detect a voltage level under threshold resulting in a logic '0' in both outputs flip flops. In this case no '1' is detected. Referring to the fig.3.26, the step followed by the internal logic are:

- Fixed the phase Φ, more acquisitions are performed. The detected logic

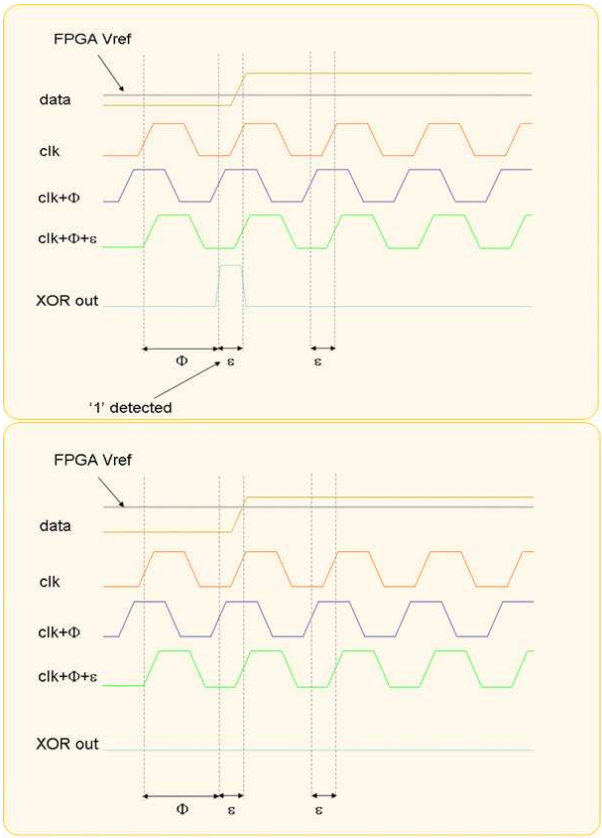

Figure 3.27: Timing Diagram of the edge detection.

'1s' are stored in the Sync Fifo, which is clocked with the clock of the incoming data. The depth of the Fifo is correlated to the number of sample (acquisitions) to be acquired.

- The logic '1s' from the Sync Fifo are read by the state machine and then accumulated. The result is stored in the Line Fifo.

- Φ is incremented and the procedure described above replicated. When Φ achieves its maximum value, i.e., the entire line has been scanned, the state machine goes in an IDLE state and wait for a next acquisition command.

To obtain the eye diagram reconstruction a pseudorandom sequence must be provided to the SIA. A sequence of pseudorandom patterns are written in

all the memory locations of the DDR II SDRAM under test (fig.3.28). Then the controller issues a free-running sequence of READ commands providing a pseudorandom sequence in input of the SIA. The steps followed to obtain the eye diagram reconstruction are:

1. The Vref of the FPGA is initialized to 0 programming the external digital potentiometer through the USB interface and the SPI port,
2. the time scanning is performed. At the end of the scanning the sampling of an entire line is stored inside the internal FIFO,
3. the FIFO is read through the USB interface and the High Speed Parallel port. Then the Vref is incremented and new timing scanning performed.

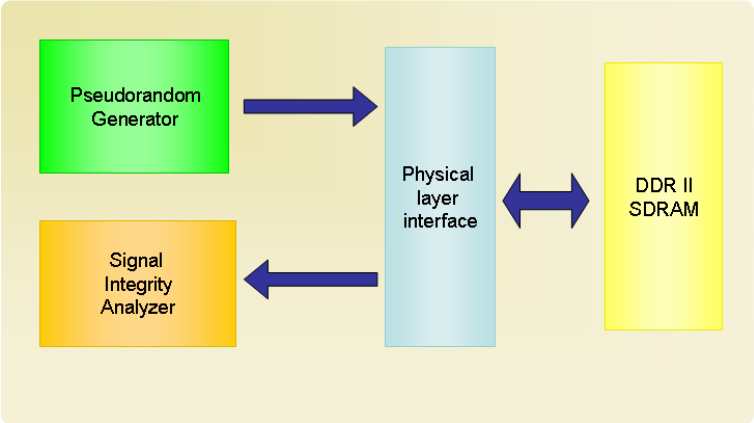

Figure 3.28: Experimental setup to measure the eye diagram of the dq data during read operation.

Fig.3.29 shows the eye diagram obtained with the Signal Integrity Analyzer at 266 MHz.

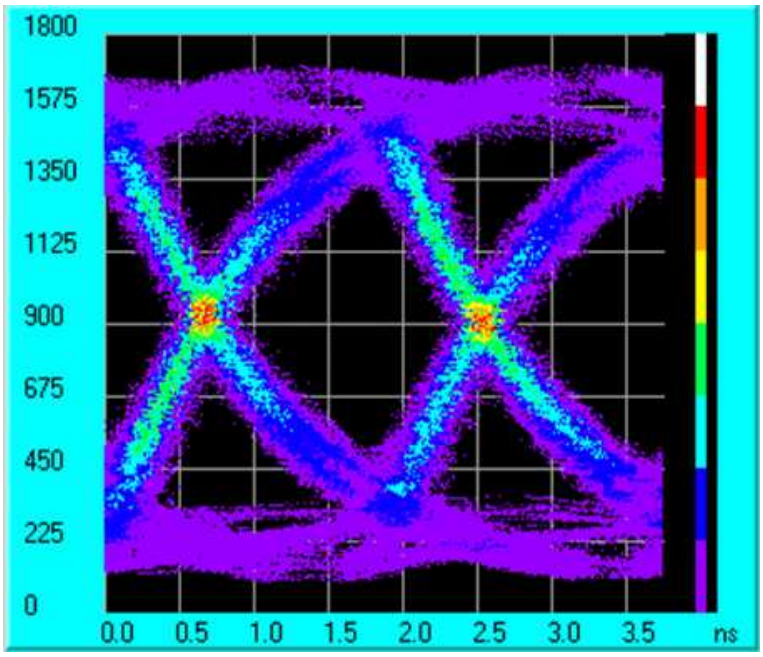

Figure 3.29: Eye diagram obtained with the Signal Integrity analyzer at 266 MHz.

Chapter 4

Experimental results

4.1 Introduction

The tester has been successfully tested in the laboratories of the XS-consulting laboratory in order to verify the foreseen performances. The test was aimed at:

- verifying the tester capability to detect physical defects, first observing the signals on the interface and after comparing the result of the testing with those obtained with the traditional tester,
- comparing of the obtained eye diagram with those obtained by the use of a traditional last generation oscilloscope.

4.2 Functional Tester Debug

In order to verify that the tester works according to the specifications, a march test has been applied and the dq, dqs, clock and command have been monitored on the pins of the chip. The test has been performed using a 64×8 DDR II SDRAM chip. Custom and low impedance probes has been used to observe the signals. Fig.4.1 shows the measured waveforms. The clock is running at $333MHz$.

In the write command (on the left-side) the DQS (selected as single ended) appears at 3 clock cycles after the write command (CAS low) is asserted. In

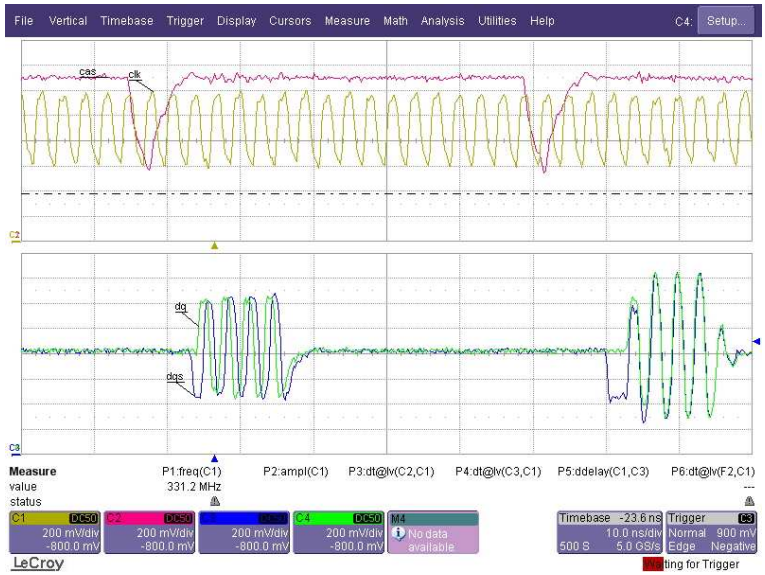

Figure 4.1: Write and Read Operation: dq, dqs, clk and CAS, monitored on the pins of the chip, are shown.

this case the DQS is provided by the controller. The data (sequence 10101010 has been chosen), appears center aligned to the dqs, as expected.

In the read command the DQS, provided by the chip, appears at 4 clock cycles after the read command (CAS low) is asserted. The data is edge aligned to the dqs. In both cases CAS latency has been programmed at 4.

In order to debug the tester, a comparison between the functional faults found by a traditional tester and those found by the realized equipment has been done. Fig.4.2 (in the bottom side) shows the results obtained testing a chip with known physical defects. The eight green rectangles represents the eight chips under test. For each rectangle, the rows represent the *segments* and the columns the byte in each segment. The segment is obtained combining some bits of the row address and all bits of the bank address. In this implementation the segment is (row13, row12, bank1, bank0). The red squares identify the defects. Several tests have been done on good chips resulting in no errors detections. Fig.4.2 shows also the software interface that has been used to test the board. Through the software interface is possible to select between different algorithms (right side). It is possible to observe the eye diagram of the signal under test (left side). The software interface allows

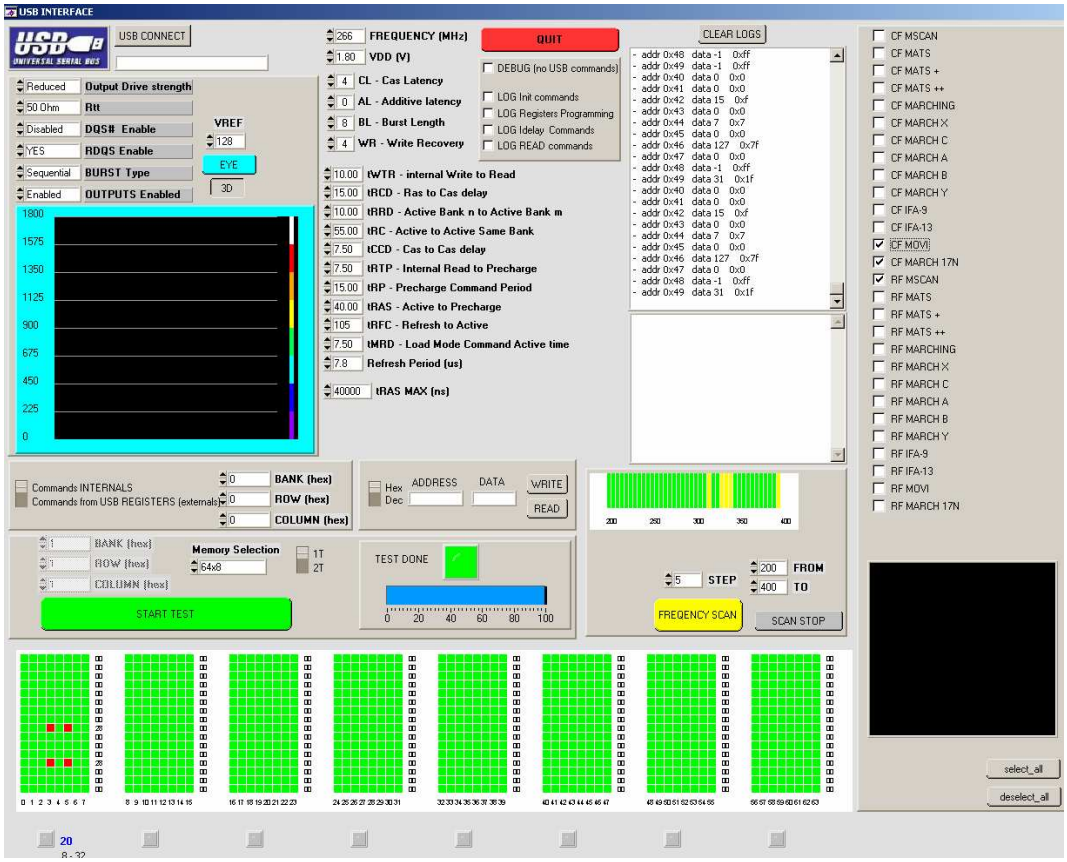

Figure 4.2: Software Interface.

to program:

- the memory frequency,
- the chip supply voltage,
- the internal memory parameters (CAS latency, Burst length, etc.),
- the timing parameter (t_{WTR}, t_{RCD}, t_{RRD}, etc.),
- the ODT (disable, 50Ω, 75Ω, 150Ω),
- the differential DQS.

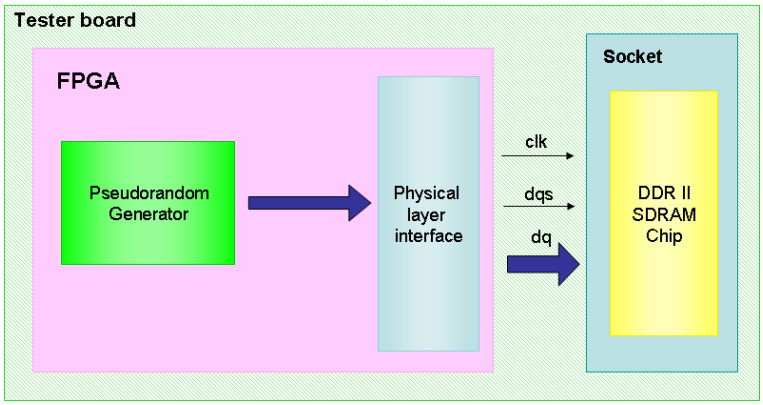

Figure 4.3: Experimental setup to measure the eye diagram of the dq data during write operation: the data is observed on special test point positioned near the chip

In order to evaluate the performances in terms of frequency of the board the eye diagram of the data at different frequencies and different value of ODT has been generated. To evaluate the performances is very interesting to observe the signals on the receivers: in the case of write command the receiver is the chip, in the case of the read command the receiver is the controller. In the first case the measurements have been done with the use of traditional oscilloscope, LeCroy WAVERUNNER with 1 GHz of bandwidth, in the second case with the use of the realized tester.

Fig.4.3 shows the setup used for the observation of the signals on the chip (write command). As shown, to obtain the eye diagram a pseudorandom sequence has been used. To evaluate the integrity of the signal the JEDEC specifications defines an area, or mask, inside the eye diagram. This area is a trapezium with the corners positioned in the point $Vref + 250V$,$Vref - 250V$,$Vref + 125V$,$Vref - 125V$. To have a good signal integrity the distance between the two parallel sides must be at least equal to $t_{setup} + t_{hold}$ defined in the JEDEC specification. This value is $275ps$. Fig.4.4 shows a comparison of a DDR II DQ bus for a typical SSTL-termination using an ODT termination versus a non ODT system writing data at $533Mb/s$. The most obvious difference between the two buses is the reduction of the over-shoot on the data bus (in the case of the ODT termination). The over-shoot reduction significantly improves the voltage margin and reduces Inter-symbol interference (ISI). The second major improvement using DDR II ODT is the increase in the slew rate. Both of these improvements increase the size of the eye diagram. The

same measurements have been done with different value of frequencies. The results have been the same.

Fig.4.5 shows that in the case of $400MHz$ the presence of the ODT dramatically increase the size of the eye diagram. In both cases ODT implementation provides improved signal integrity, data eyes, and voltage level margin for DDR II to enable the systems to attain higher data rates.

In all cases the eye opening is sufficiently large to have a good transmission of data.

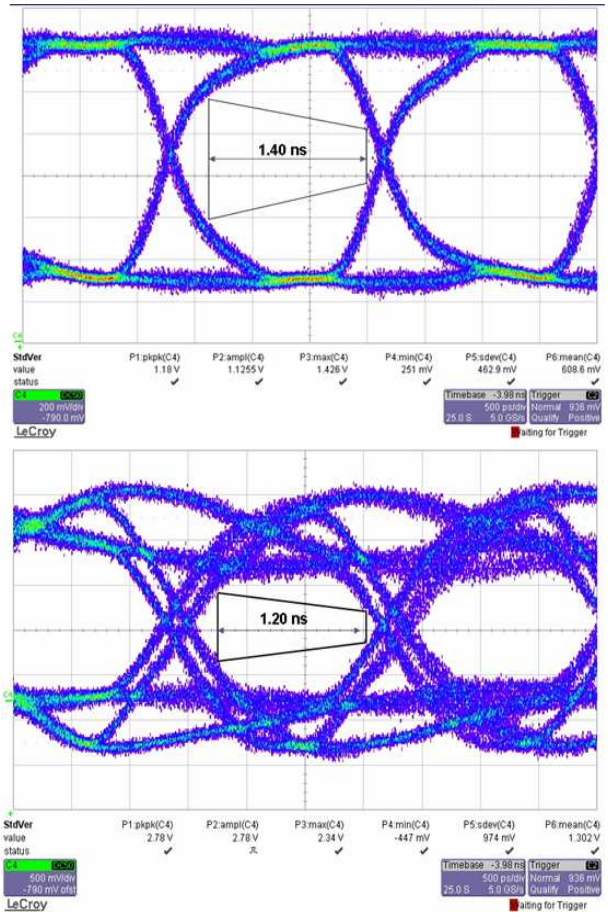

Figure 4.4: Eye diagram measured at chip using 266 MHz clock rate, 1.8-V SSTL-I/O standard, and PRBS8 pattern, and with burst lenght 8, ODT 75Ω(top side) and without ODT(bottom side)

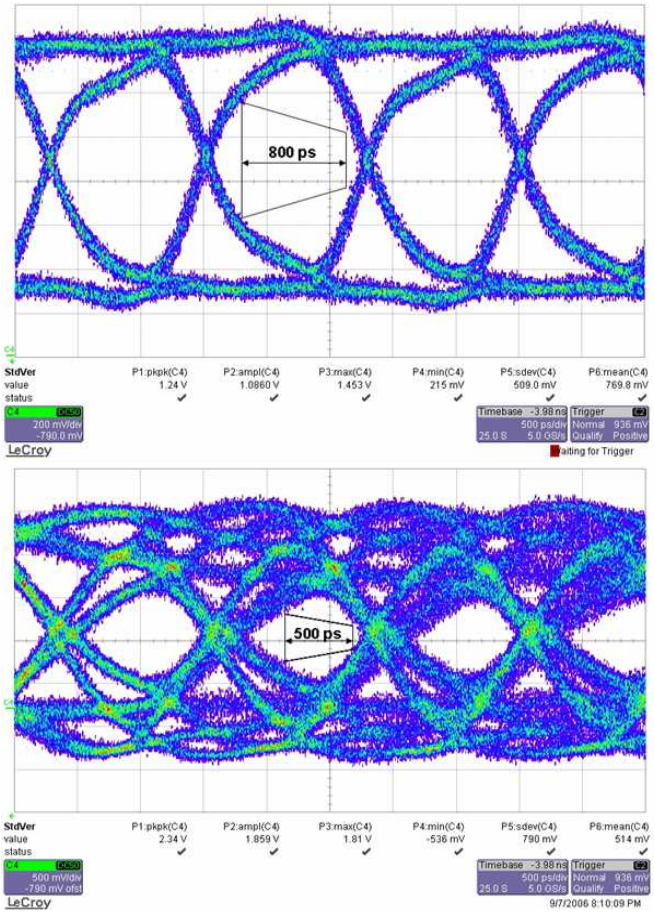

Figure 4.5: Eye diagram measured at chip using 400 MHz clock rate, 1.8-V SSTL-I/O standard, and PRBS8 pattern, and with burst lenght 8, ODT 75Ω(top side) and without ODT(bottom side)

4.3 Signal Integrity Analyzer Debug

Fig.4.6 show the setup used to observe the data with the use of the tester board. A pseudorandom pattern is written in each memory location. Then a continuous sequence of read commands are issued by the controller resulting in a pseudorandom sequence at the input of the Signal Integrity Analyzer. Fig.4.7 shows the comparison of the eye diagram obtained by the use of the tester board and the eye diagram obtained with a commercial oscilloscope.

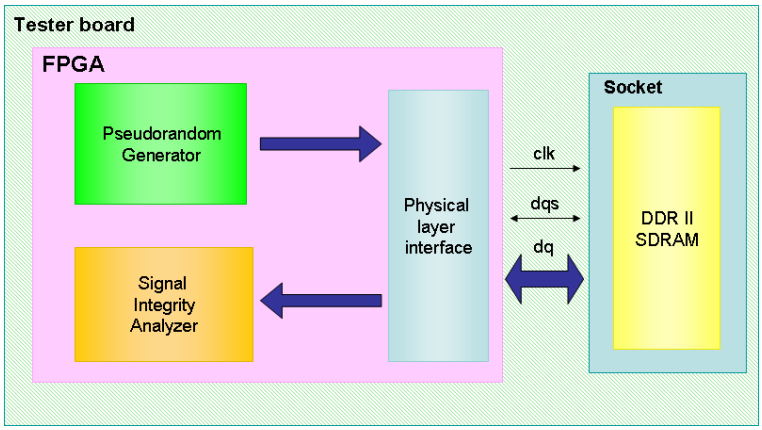

Figure 4.6: Experimental setup to measure the eye diagram of the dq data during read operation

Tab.4.1 provides a detailed comparison between the two waveforms. The slew rate for a rising edge is defined ad the slew rate between the last crossing of $Vref - 125V$ and the first crossing of $Vref$. The slew rate for a falling signal is defined as the slew rate between the last crossing of $Vref + 125V$ and the first crossing of $Vref$.

The tester can be use to quickly evaluate the configuration to be used to have a good signal integrity. Fig.4.8 and fig.4.9 show the eye diagrams of the dq signal respectively in full drive strength configuration and reduced strength configuration. DDR II configured as full drive has a target output impedance

	Custom Oscilloscope	**Commercial Oscilloscope**
horizontal eye opening	$1.4ns$	$1.4ns$
vertical eye opening	$700mV$	$700mV$
slew rate on falling edge	$700mV/ns$	$700mV/ns$
slew rate on rising edge	$700mV/ns$	$700mV/ns$
distance between parallel sides	$800ps$	$800ps$

Table 4.1: Comparison between the parameters related to the waveforms obtained by the use of the two oscilloscopes.

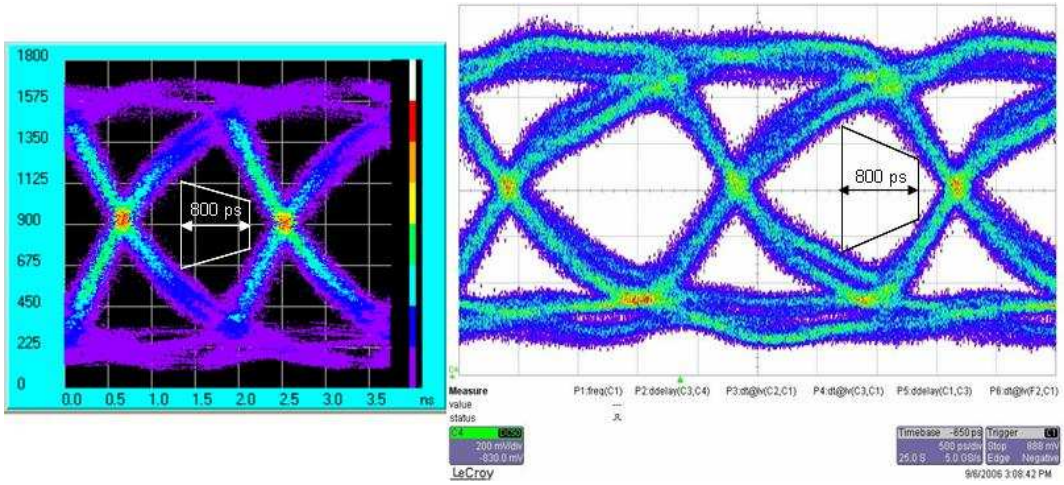

Figure 4.7: Comparison between the eye diagram obtained by the use of the board tester (left side) and those obtained by the use of a traditional oscilloscope.

of approximately 18Ω. When the device is configured as reduced drive the target output impedance is approximately 40Ω.

In the full drive configuration the chip driver provides much current than in the reduced configuration. This results in faster slew rate and higher voltage levels. In both the figures the eye diagrams have been generated with different values of ODT, respectively 50Ω, 75Ω, 150Ω and without ODT. Any configuration results in a good signal integrity. The slew rate on the rising and falling edges, referred to ODT=75Ω, has been calculate for both the configurations (full drive strength and reduced drive strength). In the case of full strength configuration it is 700mV/ns (on the rising and falling edge), in the case of reduced strength configuration it is 600mV/ns (on the rising and falling edge), slower as expected. By the fig.4.8 and the fig.4.9 is possible to observe that the crossing point occurs at 50% of the maximum voltage value. This demonstrate the absence of Duty Cycle Distortion. The result is confirmed by the measurement of the slew rate on the rising and falling edges (they have the same value).

Similar measurements have been performed for higher frequency. Fig.4.10 and Fig.4.11 show the eye diagrams of the dq signal at 400 MHz, respectively in full strength configuration and reduced strength configuration. At this higher frequency the ODT has a big impact on the signal integrity.

In both cases the signal is affected by the deterministic jitter, ISI. This results in vertical eye closure and in both delayed and advanced edges relative to their ideal positions. This jitter can be due by the bandwidth limitation of the chip, or improper impedance terminations. The measurements have been done at different value of the ODT. Referring to the fig.4.10 only in the configuration with $ODT = 75\Omega$ and $ODT = 150\Omega$ the eye opening is sufficiently large to guarantee a data transmission without errors. In the case treated in the fig.4.11 the better configurations are those related to $ODT = 150\Omega$ and ODT disable. This means that with properly chosen termination the effect of the ISI can be reduced.

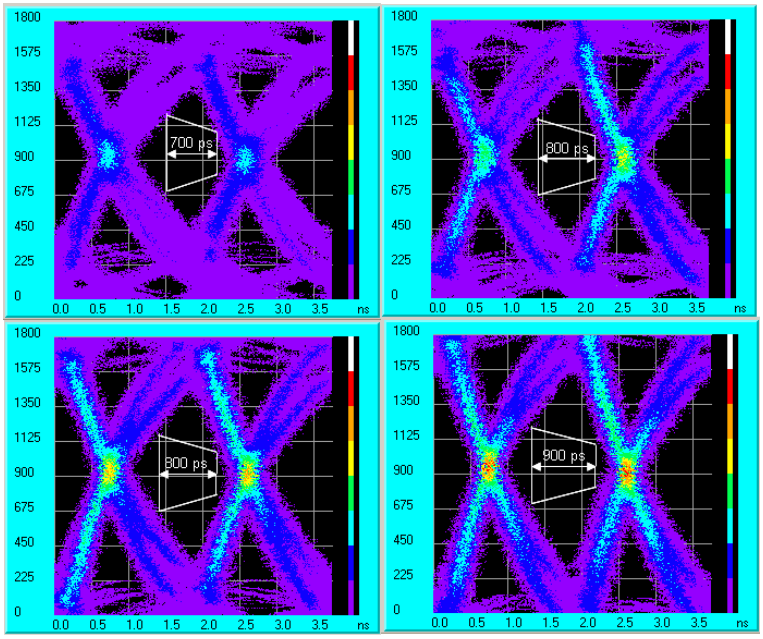

Figure 4.8: Eye diagram measured at FPGA pin using 266 MHz clock rate, 1.8-V SSTL- I/O standard, and PRBS8 pattern, and with burst lenght 8, in full drive strength configuration and ODT respectively of 50Ω, 75Ω, 150Ω and without ODT

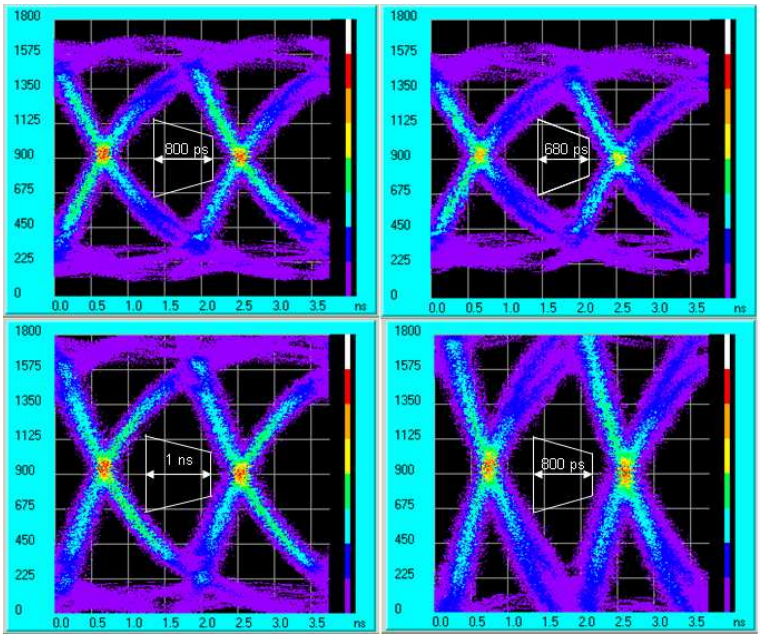

Figure 4.9: Eye diagram measured at FPGA pin using 266 MHz clock rate, 1.8-V SSTL- I/O standard, and PRBS8 pattern, and with burst lenght 8, in reduced drive strength configuration and ODT respectively of 50Ω, 75Ω, 150Ω and without ODT

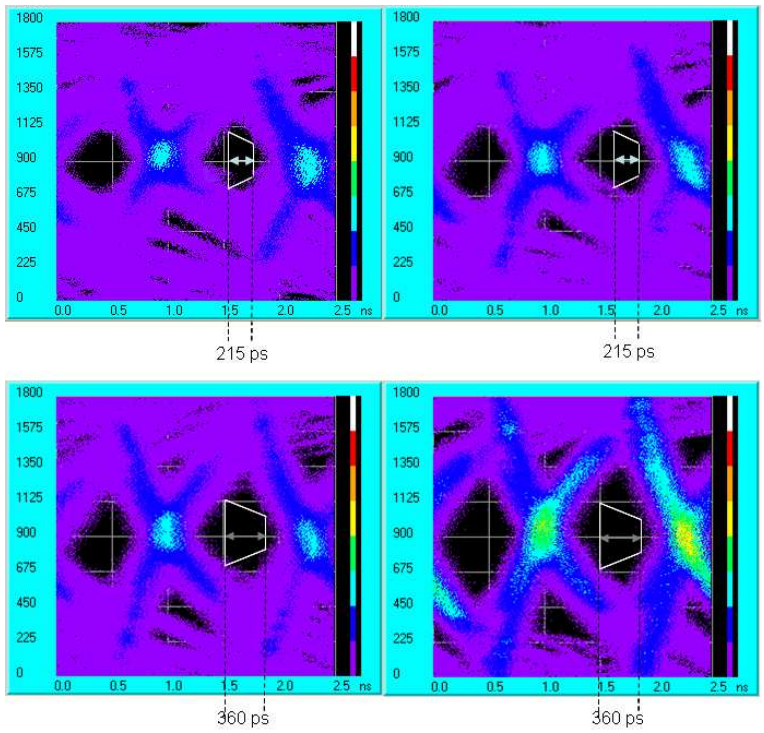

Figure 4.10: Eye diagram measured at FPGA pin using 400 MHz clock rate, 1.8-V SSTL- I/O standard, and PRBS8 pattern, and with burst lenght 8, with drive strength full and ODT respectively of 50Ω, 75Ω, 150Ω and without ODT

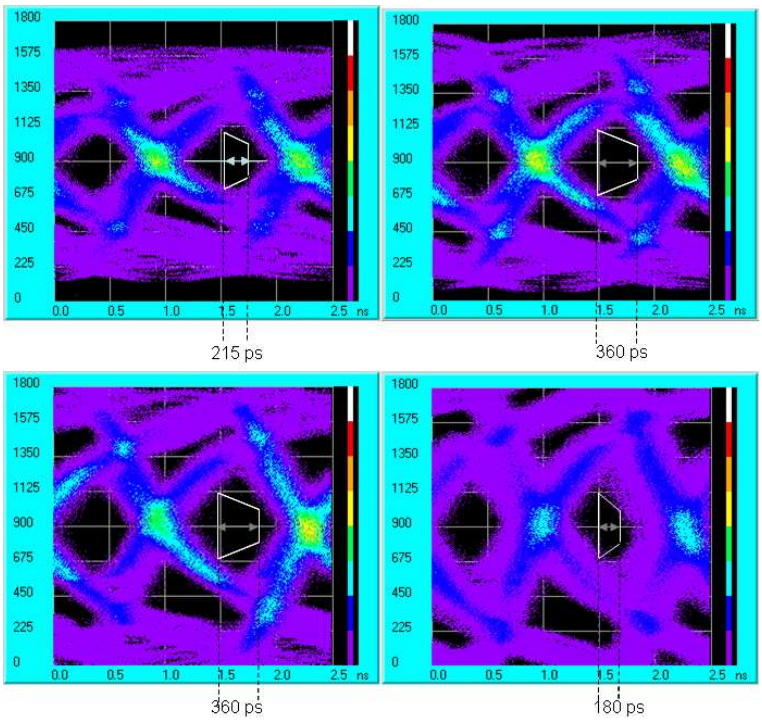

Figure 4.11: Eye diagram measured at FPGA pin using 400 MHz clock rate, 1.8-V SSTL- I/O standard, and PRBS8 pattern, and with burst lenght 8, with drive strenght reduced and ODT respectively of 50Ω, 75Ω, 150Ω and without ODT

Conclusions

DDR2 is the highest-bandwidth memory available today, supporting the improved graphics and multitasking often found on the most advanced PCs and workstations. The increasing density and higher frequencies have made the test processing an important issue to guarantee the reliability of memory operations. More and more effort is being dedicated to the study of the faulty behavior of memory devices than ever before. However the testing process is responsible for a large part of the cost of the memories, standing now at 40% and gradually rising with each new generation.

The realized tester, whose design production and testing is the innovative issue of this Ph.D thesis work, provides an attractive alternative to the traditional equipment available on the market which are the real responsible of the high cost of testing process. Moreover the tester available on the market are not able to perform a complete and exhaustive analysis without dramatically increase the production costs and time. The realized tester provides:

- fast signal diagnosis, giving the possibility to observe the signal of the chip and its eye diagram. In this way it is easy to find out anomalies on the system supporting the chip such as improper terminations, bad connections or anomaly in the chip such as bandwidth limitation. By the observation of the eye diagram is possible to perform a qualitative analysis on the Jitter of the signal and so trace the sources of it,

- fast chip diagnosis, finding out the defects on the chips, with high reliability.

The innovative issue of this Ph.D thesis work is that 90% of the functionalities implemented by the tester have been fitted into a high-speed and high-density FPGA. This feature, reducing the hardware resources, dramatically decreases the cost of the test equipment and so the production cost of

the memory devices. Thanks to a high-pipelined architecture and a special care into the place&route phase, high performances in terms of frequency have been reached. This feature also reduce the production time of the memory devices. The tester provides low mass, low power consumption and standard interfaces.

The realized tester can be introduced in many different manufacturing stages, starting from the initial test application stage performed by the manufacturers, through the test stage performed by the memory modules assemblers aimed at the verification of the specifications declared by the manufacturers and at the debugging of the assembled system, and ending the memory ramp-up stage where products are shipped to the customer. The tester has been successfully tested comparing the test results with those obtained by the use of more expensive and slower traditional equipment.

Being based on the use of programmable device it could be easily adapted to the future improvements of the DDR II SDRAM technology.

Bibliography

[1] A. K. Sharma, *Semiconductor Memories: Technology, Testing and Reliability.*, New York: IEEE, (1997).

[2] B. Prince, *Semiconductor Memories: A Handbook of Design, Manufacture and Application.*, New York: Wiley, (1991).

[3] Micron Technology Inc., *General DDR SDRAM functionality*, http://www.micron.com (2001).

[4] Micron Technology Inc., *DDR II (Point-to-Point) Features and Functionality*, http://www.micron.com (2003).

[5] Changsik Yoo et al, *A 1.8-V 700-Mb/s/pin 512-Mb DDR II SDRAM with On-Die Termination and Off-Chip Driver Calibration*, IEEE Journal of Solid State Circuit, vol.39, no.6, June 2004.

[6] Elpida Memory, Inc.,*How to use DDR2 SDRAM*,http://www.elpida.com (2004)

[7] Agilent Technologies, M. Mueller, R.Stephens, R. McHugh, *Total Jitter Measurement at Low Probability Levels, Using BERT Scan Method*, http://www.agilent.com (2005).

[8] Agilent Technologies, *Measuring Jitter in Digital System*, http://www.agilent.com (2003).

[9] Agilent Technologies,Greg LeCheminant,*How to take the evil out of eye diagrams*, http://www.agilent.com (2002).

[10] Agilent Technologies, *Jitter Analysis Techniques for High Data Rates*, http://www.agilent.com (2003).

[11] Agilent Technologies,*Finding Sources of Jitter with Real Time Jitter Analysis*, http://www.agilent.com.

[12] A.Kuo, R.Rosales, T.Farahmand, S.Tabatabaei and Andre Ivanov, *Crosstalk Bounded Uncorrelated Jitter (BUJ) for High Speed Interconnect*,IEEE Transactions on Instrumentation and measurement, vol.54,no.5,October 2005.

[13] Agilent Technologies, *Jitter Analysis: The dual-Dirac Model, RJ/DJ, and Q-scale*, http://www.agilent.com (2004).

[14] Marcel A.Kossel and Martin L.Schmatz, *Jitter Measurements of High-Speed Serial Links*, IEEE Design and Test of Computers (2004).