Mixed Signal Verification of an FPGA-Embedded DDR3 SDRAM Memory Controller using ADMS

Arch Zaliznyak[1], Malik Kabani[1], John Lam[1], Chong Lee[1], Jay Madiraju[2]

1. **Altera Corporation**
2. **Mentor Graphics Corporation**

Abstract

System-level verification of a high-end FPGA with an embedded DDR3 SDRAM external memory controller presents challenges to designers that exceed those encountered in ASIC-like verification flows due to the mixed-signal nature of the design.

The Stratix III FPGA with a DDR3 Hard IP is fabricated on a 65nm process. The memory interface is comprised of programmable-width data groups running at 800-Mbps per data bit. The memory controller consists of custom/analog Hard IP (Read, Write, Command/Address paths and the Clock Management) and the Soft IP (Auto-calibration IP and Memory Controller) and with the External Memory device model comprise a mixed-signal system. See Figure 1.

Traditional Verilog verification methodology, while guaranteeing functional aspect of the verification flow, is not capable of verifying the adherence to the system-level timing specifications, nor in accurately predicting system timing budget. This paper describes a strategy devised and successfully utilized for the verification of the embedded DDR3 SDRAM external memory controller in the FPGA. The mixed-signal verification methodology is based on the use of Mentor Graphics ADMS mixed-mode simulator with the Mentor Graphics Fast SPICE engine ADiT enabled. It presents steps taken in the verification flow dealing with the simulation data preparation, the usage of the specific features of ADMS, and the verification process automation.

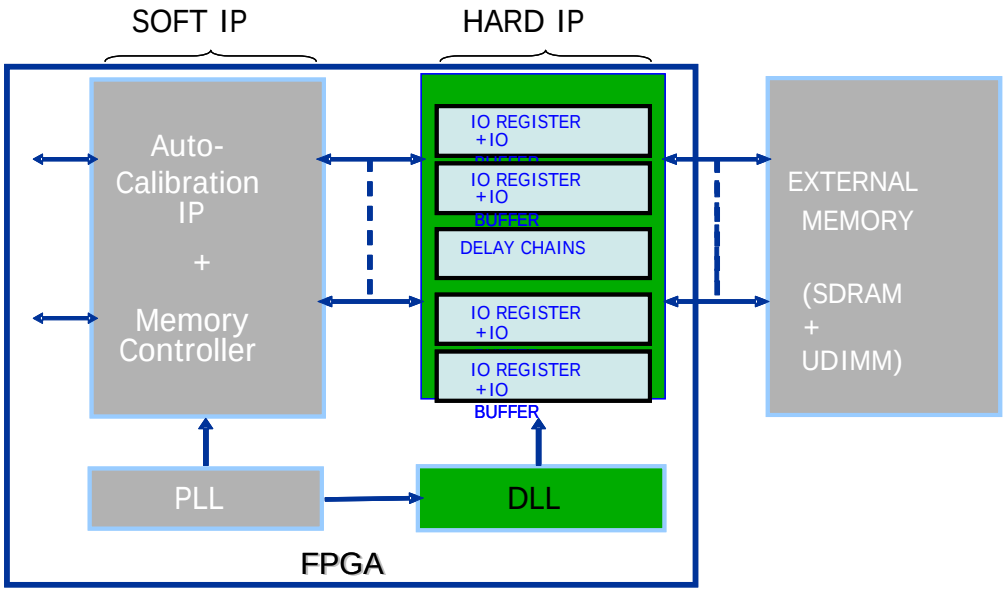


Figure 1. Memory Interface System

CP-01028-1.0
March 2007

Motivation for Mixed Signal Verification

Traditionally, ASIC design verification flows have always been content with using Digital Simulators to do their system-level verification. The entire system was described in a Hardware description language like Verilog and simulated using digital simulators like Mentor Graphics Modelsim. This type of verification is termed "functional verification" since it mainly targeted verifying the high level functionality of the chip. Hence it only guaranteed the high-level functionality of the chip matched its specifications. With this approach the designers were not verifying if the overall system timing met the specifications. The individual circuit blocks were verified using SPICE/Fast SPICE tools in a block level verification environment. However, for designs mixed-signal in nature this approach is not sufficient for various reasons. Using only purely digital simulations, it is very hard to verify the timing specifications, or accurately predict system timing budget that would be exhibited in silicon-package-board environment.

For our DDR3 solution, the system timing budget depends on proper configuration of the dynamically programmed delays which is accomplished by the Auto-calibration IP. The Auto-calibration accuracy depends on accurate modeling of silicon-package-board, preferably in SPICE. The functional robustness of the Auto-calibration IP needs to be checked with the realistic duty-cycle distortions of the data and strobe signals as they travel through the circuits of the Hard IP. Also, the drive strength, termination scheme, data pattern effects, process voltage and temperature (PVT) effects can be investigated with the SPICE models to find the areas critical to the system's bandwidth. To accomplish that, one needs to incorporate transistor-level models into these simulations. However running the entire system at the transistor-level is impossible due to the capacity/speed limitation of current Analog simulators. Hence a mixed-Signal simulation flow with the ability to run various test benches with the capability to simulate both Verilog and SPICE is required to address the above needs. Also, the flow needs to be scalable with increasing SPICE content in the simulations.

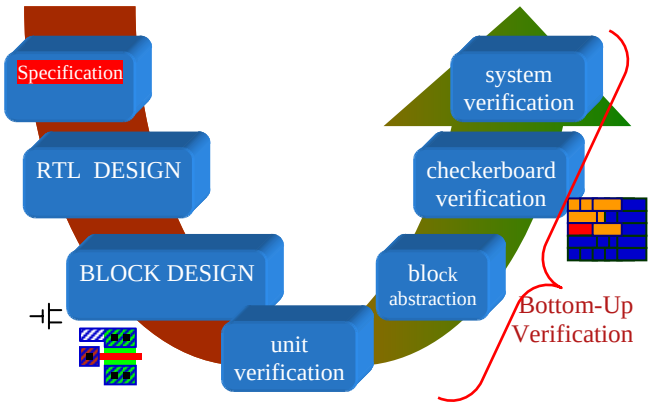


Figure 2. Bottom up Verification Flow

Bottom Up Verification Strategy using Advance MS

In order to efficiently and exhaustively validate the design, verification flow using Mentor Graphics mixed signal simulation tool Advance MS (ADMS) simulator was adopted. In this flow, each block of the Hard IP has been verified at transistor-level on its own. The system-level validation is targeted to check the adherence to the system-level timing specifications, and to accurately predict system timing budget, taking into account duty-cycle distortions, which will guarantee that the design is functionally and timing-wise correct in the first silicon run.

"Simulation configuration" Concept

Here the concept of a "Simulation configuration" is introduced. Each Simulation configuration runs the same design except that different blocks in the design are at different levels of abstraction, i.e. "checkerboard" pattern. For example, in configuration # 1 a Write Path design block (see Figure. 3) may be in SPICE, whereas in configuration # 2 the same design block may

be in Verilog and the Read Path design block may be in SPICE, depending on the nature of the test and a particular timing parameter being tested. It is the verification team's job then to come up with a series of simulation configurations to test various system characteristics.

One advantage of using such a flow is the ability to use the same set of test benches used in digital simulations, since the simulation is run on the complete design. This alleviates the need to create separate test benches for stand alone portions of the design. Also, now the designer can verify each SPICE block in the context of the full design. Critical system characteristics can be measured by keeping only the required portions of the chip in SPICE and the rest in Verilog. Important constraints such as data path timing, critical path delay can be verified with the SPICE implementation of the design blocks rather than HDL models as with digital verification flow. This enables designers to accurately predict the system budget and evaluate the risk accordingly.

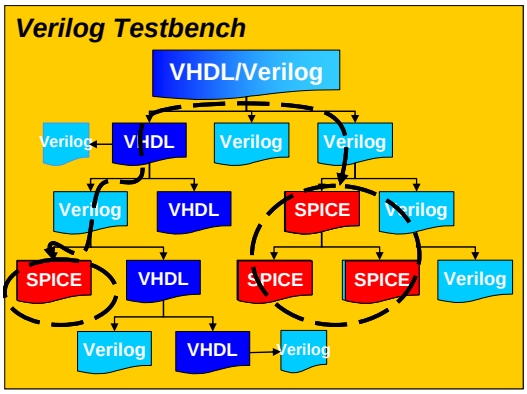
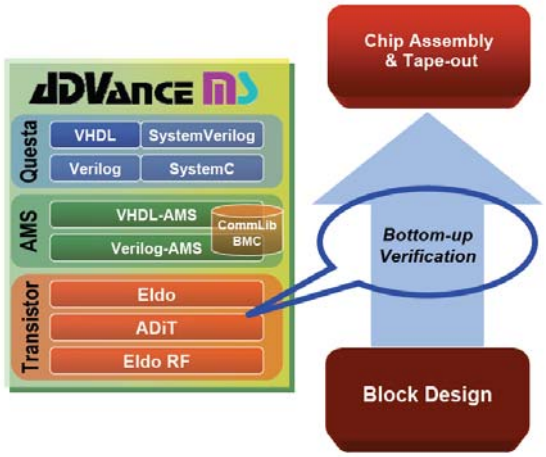


Figure 3. Analog inside Digital Centric Flow

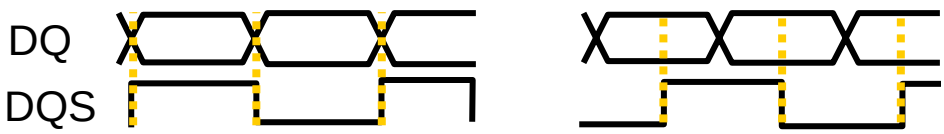
However, when trying to run these system-level simulations using a fairly large amount of SPICE, the simulations take too long to be useful. Compounded with the requirement of running these simulations at multiple process corners, that makes the turn-around time even longer. This renders the flow difficult to use in situations where it is absolutely a must to run such simulations to verify certain system characteristics. Hence using ADVance MS by itself was not scalable with the increasing amount of SPICE in these simulations. The solution was to use Mentor Graphics ADMS-ADiT Simulator which integrates the Mentor Graphics Fast SPICE engine ADiT into the ADMS mixed signal simulator. The difference between ADMS and ADMS-ADiT is that the former uses the highly accurate but slow Eldo SPICE engine whereas the latter uses the slightly less accurate but high speed ADiT fast SPICE engine. Incorporating ADMS-ADiT enables running of larger amount of SPICE with reasonable turn around times making the flow scalable and more efficient.



Description of the DDR3 SDRAM external memory controller

System-level Verilog test bench contains both the Soft IP RTL modules as well as the mixture of SPICE and Verilog netlist representing the custom/analog block, along with the necessary Verilog and VHDL-AMS models embedded in both SPICE and Verilog netlists.

DDR3 memory interface is source-synchronous, where a bidirectional data strobe DQS is used for both Read and Write operations. However, handling timing is



Read: Edge-aligned DQS sent by memory Write: Center-aligned DQS expected by memory

Figure 4. Data and DQS Relationship in DDR Memories

different in each case. The data strobe is edge aligned with the data during a Read operation and center aligned during a Write operation (see Figure 4). One data strobe signal is associated with a number of data bits, usually eight. Within the memory, the data strobe path closely matches the data path; hence the access times with respect to data strobe are significantly better than the access times with respect to clock.If data strobe is used from the memory to capture the data sent by memory, maximum performance is realized. The challenge for the Memory controller is to transfer the data captured in the memory domain to the system domain.

Need for Auto-calibration Techniques

As applications move to higher frequencies and newer technology generations, memory and board uncertainties are increasing as a percentage of the bit period. See Figure 5. The need to place the strobe in the center of the data valid window is most important, so there is a need to reduce some of these uncertainties and improve the margin on the controller side to achieve higher performance. With Auto-calibration, the internal timing is set up in real time in a real system environment to achieve optimum timing, highest performance, and a robust design.

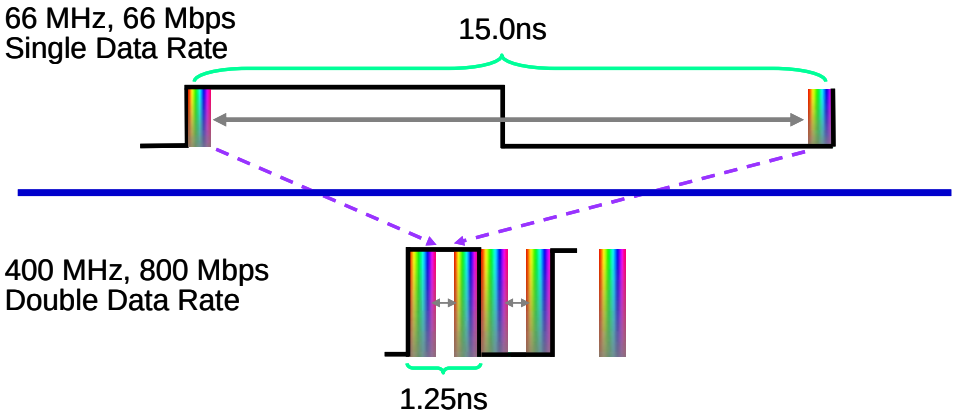


Figure 5. Memory and board uncertainties increase as a percentage of the bit period

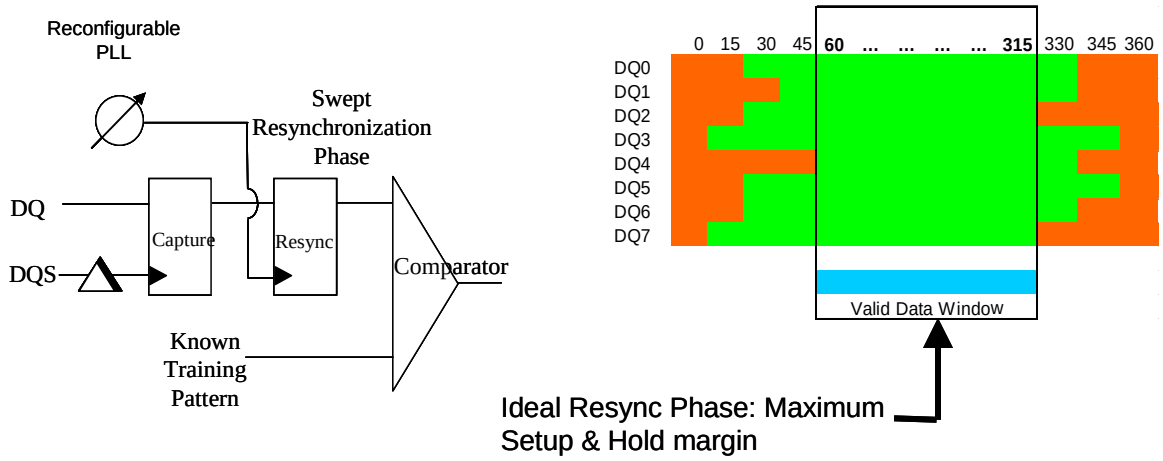


Figure 6. Auto-calibration System Block Diagram

The aim of data Auto-calibration is to determine and set up the optimum sampling phase of the capture and resynchronization clock. The capture register shown in Figure 6 captures the data from external memory. The resync register is used to transfer the data from the memory domain to the FPGA system domain.

In such a mixed-signal system, with the need to account for Vt, temperature, load and other variations, the simulation approach chosen is a must. To verify the Auto-calibration is working correctly to produce the correct timing, the system-level testbench goes through the Auto-calibration process on the read path, write path on groups of pins and measures various delay parameters. From these measured delays, the Auto-calibration algorithm decides the best timing fit. If this entire system was simulated in Verilog using a Digital Simulator it would necessitate the manual back-annotating of the various delay parameters measured by the Auto-calibration process which can be time-consuming and error-prone. In order to make sure the timing requirements on the Data Strobe signal are within the specs, it is necessary to simulate the entire Memory subsystem with the address path, data path, strobe, clock, etc in transistor-level implementations.

Top-level Simulation Strategy using ADMS-ADiT

Configuration	Hard IP			Soft IP	
	Read Path	Write Path	Clock Management	Calibration Controller	External Memory Model
1	SPICE	Verilog	Verilog/VHDL	Verilog	Verilog
2	Verilog/VHDL	SPICE	Verilog/VHDL	Verilog	Verilog
3	SPICE	SPICE	Verilog/VHDL	Verilog	Verilog

ADMS-ADiT allows the Auto-calibration Soft IP to run in tandem with the transistor-level implementation of the Hard IP blocks, which results in very precise delay parameter measurements. Not only does this result in a more accurate prediction of the system timing budget, also reduces the overhead of having to do manual back-annotation of the delay parameters. The idea is to simulate portions of the hard IP in SPICE and keep the soft IP blocks (Calibration system, Memory controller, etc) in Verilog (including the RTL Memory model) and run the system-level test bench in ADMS-ADiT. Then, multiple simulations need to be run with varying process corners, temperature and some varying parameters such as the DQ-DQS trace/DIMM skew, etc. Also, since it is imperative to validate all the portions of the Hard IP to establish the required timing, keeping the entire Read/Write/Address paths in SPICE may not result in quick turnaround time. Hence multiple transistor-level simulations need to be run with different portions of Hard IP in SPICE and the rest in Verilog. The table above summarizes the simulations run on the design to validate the system timing.

Simulation Data Preparation

1. Test bench and Netlist Preparation

The top-level block of the design is the Verilog test bench which instantiates the top-level design blocks. The Hard IP is instantiated from this Verilog design hierarchy and replaced with SPICE without modifying the Verilog netlist using ADMS compilation commands. This allows the original Verilog-only flow to be extended with minimal modifications. To make the customization of netlist for each simulation configuration easy, the Verilog language feature “generate statements” is used.

Each instance inside the Verilog that could potentially be replaced with SPICE is surrounded by the Verilog construct “generate …. endgenerate”. Each such Generate statement uses a “genVar” variable to statically determine if SPICE or Verilog is to be used.

Example: The instance c6480 looks like the following in the Verilog-only Flow:

c6480 x6480b0 *(.datovr(datovrb[0]), .din(dinb[0]), .octrt(roct[0]),.oeb(oebb[0]), vccp(vccp),..vss(vss));*

Now, to extend to the Mixed-Signal flow so it can potentially be replaced with SPICE, it would now look like

generate

if (gSpice_c6480 = = 1)

***__*spicelib. c6480a x6480b0 x6480b0*__ *datovr(datovrb[0]), .din(dinb[0]), .octrt(roct[0]),.oeb(oebb[0]), vccp(vccp),..vss(vss));*

Else

c6480a x6480b0 x6480b0 *datovr(datovrb[0]), .din(dinb[0]), .octrt(roct[0]),.oeb(oebb[0]), vccp(vccp),..vss(vss));*

endgenerate

2. ADMS Simulation setup

ADMS requires all of the Verilog modules to be compiled into a Library. All the Verilog .

modules in the design are compiled into a work library and the SPICE subckts associated with each Verilog module that could potentially be replaced with SPICE is compiled into a separate Library. The ADMS command **vaspi** is used to compile the SPICE subckts. Then, when a certain simulation configuration is run, the appropriate Generate Variable is passed in from the command line to the simulator which uses the appropriate abstraction for the block (SPICE or Verilog) as specified by the value of the variable. This avoids re-compilation of modules every time the user wants to run a new configuration, because now only the value of the genvar flag being passed needs to be changed on the Simulation Command line invocation..

In the above example, the variable “gspice_c6480” is passed in from the command line with the value “1”, so the SPICE implementation for the block c6480 is used.

All of the above commands have been automated inside custom scripts written to create customized netlists, appropriate compilation and simulation commands for each configuration

Simulation Results

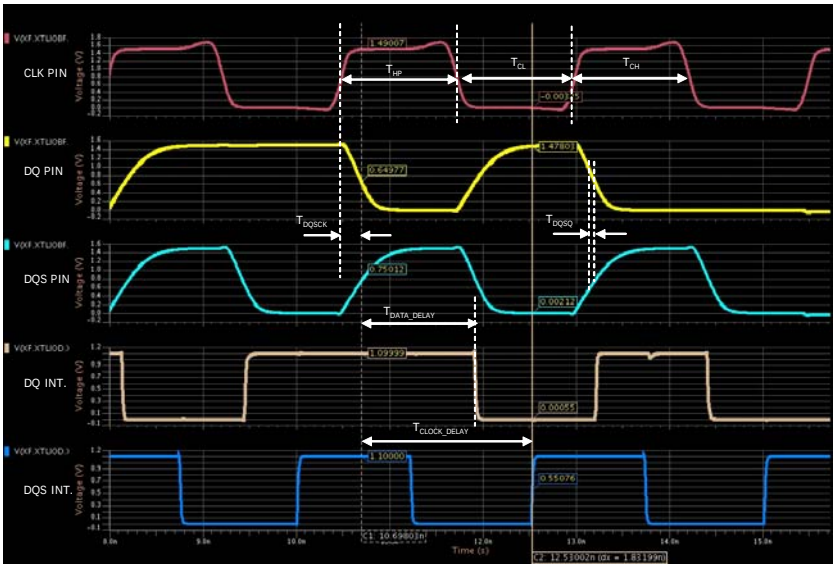


Figure 7. ADMS-ADiT Simulated Read Timing Margin

Figure 7 shows a typical ADMS waveform snapshot of the Read Path. Here, it is demonstrated the successful internal placement of the DQS strobe relative to the internal data window.

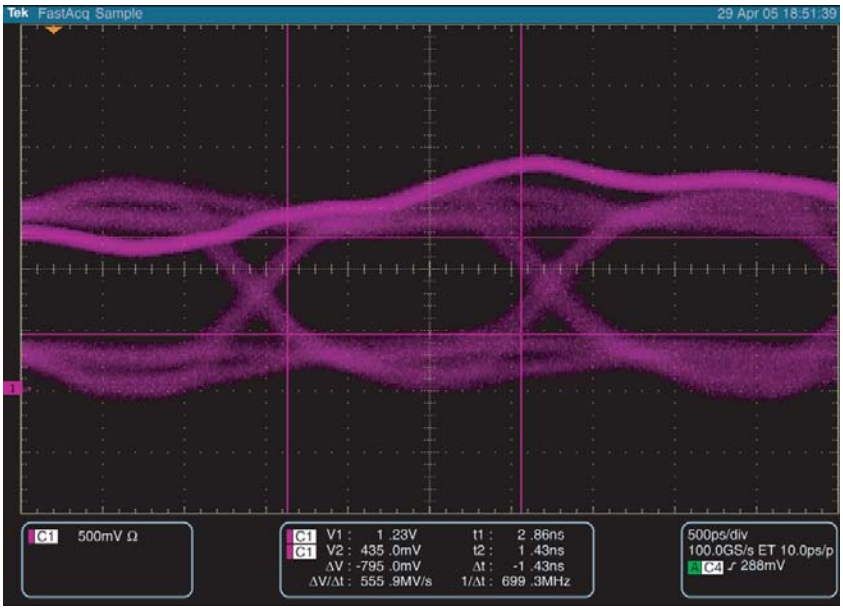


Figure 8. Measured Eye Diagram During Write to DIMM

Figure 8 shows lab measurements of the Eye during Write to DIMM.

Conclusion

Verification flow based on ADMS-ADiT Mixed signal FastSPICE simulator applied to a DDR3 Memory controller design is demonstrated. The functional robustness of the Auto-calibration IP was checked with the realistic duty-cycle distortions of the data and strobe signals as they travel through the circuits of the Hard IP. The adherence to the system-level timing specifications was verified, as well as system timing budget was accurately predicted.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com