

# A dynamic DFI-compatible strobe qualification system for Double Data Rate (DDR) Physical Interfaces

Alexios Alexandropoulos, Fotis Plessas, Michael Birbas

Analogies S.A.

Patras Science Park

Rio, 26504, Greece

e-mail: alexis.alexandropoulos@analogies.eu

**Abstract**—In this work a dynamic strobe masking system (DSMS) for DDR memory interfaces is presented which works with existing DFI signals to provide dynamic masking and produces a clean strobe suitable for data capture, making it the first dynamic DFI-compatible strobe qualification system. This DSMS scheme produces a masking signal to qualify the expected pulse stream on the DQS line, thus masking out other spurious activity. Post layout simulation results in TSMC 90nm process validate the proposed masking system in the 200-533MHz range, meeting the mask shut-off specification in best, typical, and worst case corners.

**Keywords**—Double Data Rate (DDR); Strobe Marking; DDR-PHY Interface (DFI);

## I. INTRODUCTION

In a real-world Double Data Rate (DDR) memory interface, voltage glitches are frequently observed on the DQS strobe during a memory read operation. These spikes, caused by signal reflections on the bi-directional strobe, appear on highZ-to-0 and 0-to-highZ transitions, during the preamble and postamble phases of the strobe [1]. If unfiltered, these can be misinterpreted by the physical interface (PHY) as actual strobe edges and lead to unexpected system behaviour or latching of erroneous data. Thus, an efficient way of eliminating these spurious edges is essential for correct operation of any PHY.

One common DQS strobe qualification methodology has been presented by L. Amarilio *et al.* in [2]. In this implementation, a calibration data pattern is written to the memory and then read back while shifting a masking pulse, until the read data match the written ones. The shifting continues until the read data no longer match the written. As a final step, the masking signal is placed somewhere within the measured valid window.

Depending on the granularity of their delay lines, such shift-and-read techniques may require a significant number of read operations and can therefore become quite time consuming. Moreover, setting the masking signal to a fixed position in time [3, 4] does not account for variations on the DQS line, thus a re-calibration procedure may be required to re-align the masking signal at the optimum position. In order to prevent the re-calibration procedure from corrupting possible existing data in the memory, additional functionality is required to temporarily save the contents of

the memory location that the calibration engine writes its data pattern to, and then restore it after the recalibration procedure is complete. The extra logic required to support this functionality increases design complexity, area and power consumption. Thus, a dynamic method for strobe qualification appears quite advantageous.

Additional design complexity in Memory Controller-PHY integration is added by the fact that different strobe masking implementations demand special input signals from the Memory Controller, with timing requirements that serve their specific architecture [5, 6]. This hinders the porting of the PHY to different Memory Controllers and adds the extra cost of altering the design of the Memory Controller in order to provide the specific signals required by the strobe masking engine. With the emergence and establishment of the Denali's DFI protocol for Memory Controller-PHY interfaces [7], the advantage of a strobe masking system that utilizes only DFI signals becomes apparent.

The proposed strobe qualification system works with existing DFI signals to provide dynamic masking of the incoming read-DQS strobe without the need for any calibration sequence. By monitoring the length of the *dfi\_rddata\_en* pulse of the DFI bus, the dynamic strobe masking system (DSMS) is able to determine the expected number of pulses on the DQS strobe. The mask becomes high at a programmable point around the middle of the preamble region and de-asserts right after the last negative edge of the DQS strobe is counted. This method provides accurate dynamic masking and eliminates any unwanted glitches or spurious pulses. The only requirement is for the user to provide the system with a fixed, one-time programmable binary value. This application-specific value correlates to the signal time-of-flight from the SDRAM to the PHY and determines the point within the preamble that the mask will become high. The mask will shut-off automatically after reception of the last expected DQS falling edge. A variety of values will work for a wide range of trip delays, as long as the selected value causes the mask to assert well inside the preamble region which is typically one clock cycle wide.

For demonstration purposes, we have used a JEDEC compliant [8] DDR2 memory interface. However, the DSMS scheme could be readily applicable to other DDR interfaces. The rest of the paper is organized as follows.

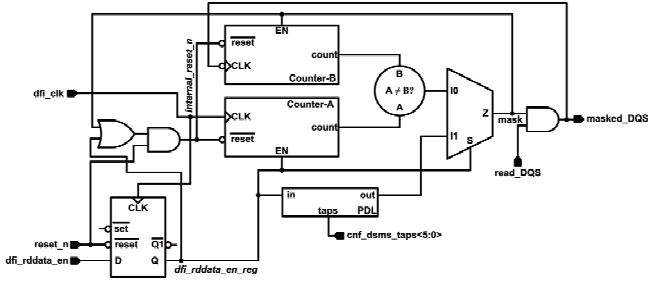


Figure 1. DSMS architecture

Section II, describes the proposed architecture of the DSMS block, followed by implementation and post-layout simulation results in Sections III and IV respectively. The paper’s conclusions are given in Section V.

## II. DSMS ARCHITECTURE

The DSMS scheme contains two synchronous counters and becomes activated during a memory-read transaction, when the *dfi\_rddata\_en* signal is asserted on the DFI bus by the Memory Controller. According to the Denali® DFI protocol, the *dfi\_rddata\_en* signal must be asserted for the number of Single Data Rate (SDR) words that the DFI is expecting. While this signal is asserted, the first counter (Counter-A) counts up. Each SDR unit of data corresponds to two units of DDR data, which are accompanied by a pulse on the DQS strobe. Therefore the count of Counter-A is equal to the number of DQS negative edges (or, equivalently, pulses) expected from the SDRAM. When the DQS pulses arrive at the PHY, a second counter (Counter-B) inside the DSMS counts the strobe negative edges and causes the de-assertion of the mask when the number of negative edges equals the number of expected edges, as provided by the Counter-A. The internal architecture of the DSMS, presented in Figure 1, consists of the following: (a) A masking gate that performs an AND operation between the incoming DQS strobe (*read\_DQS*) and the masking signal. (b) A 3-bit positive edge-triggered up-counter (Counter-A) that counts the number of clock cycles that *dfi\_rddata\_en* remained high. (c) A 3-bit negative edge-triggered up-counter (Counter-B) that counts the number of negative edges received on the DQS strobe. (d) An Inequality Monitor that continuously checks whether or not the received number of strobe negative edges matches the expected. If not equal, the output becomes high, otherwise remains low. (e) Positive edge-triggered D Flip-Flop that registers the incoming *dfi\_rddata\_en* signal. (f) A 64-tap Programmable Delay Line (PDL) used for applying a fixed delay to the registered *dfi\_rddata\_en* signal. This delay controls the point within the preamble that the mask will assert and is provided by the system user, as explained above. (g) A multiplexer that outputs the masking signal. (h) Counter Reset Logic (2-input OR gate driving a 2-input AND gate) for resetting both counters when the masking procedure is complete.

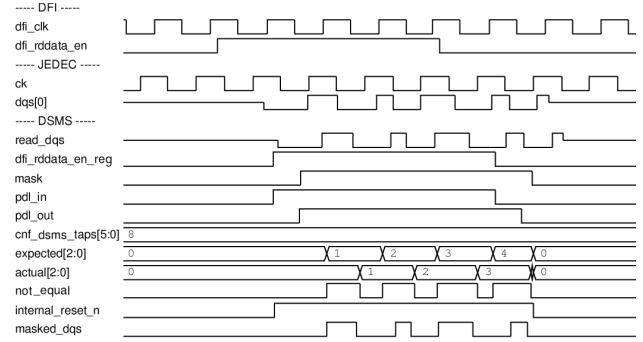


Figure 2. Example of DSMS operation with Micron DDR2 memory model at 533MHz.

Upon reset, both of the counters are equal to 000, the Q output of the D-FF is low, and the masking signal (i.e. Inequality Monitor output propagated through the multiplexer) is also driven low. When the registered *dfi\_rddata\_en* signal (*dfi\_rddata\_en\_reg*) asserts, the counters’ reset signal is released and the multiplexer selects the PDL output as the masking signal. While *dfi\_rddata\_en\_reg* is high, Counter-A is enabled and starts to increment with every rising edge of *dfi\_clk*, counting the number of expected negative edges. During the high-time of *dfi\_rddata\_en\_reg*, the first DQS edges begin to arrive and Counter-B starts to count the strobe falling edges. To ensure the necessary condition that the first strobe edges arrive while *dfi\_rddata\_en\_reg* is high, the  $t_{rddata\_en}$  DFI parameter which corresponds to the delay between the issuance of a READ command on the DFI interface and the assertion of *dfi\_rddata\_en*, is set equal to the SDRAM’s read latency (RL). Other values for this parameter can be employed by appropriately delaying the *dfi\_rddata\_en* signal inside the DSMS so that the aforementioned condition is met.

Since both counters operate simultaneously but asynchronously with respect to each other, they are expected to become equal at times. However, this does not cause the mask to de-assert early, since *dfi\_rddata\_en\_reg* is high during these times and the multiplexer selects the PDL output as the masking signal. When *dfi\_rddata\_en\_reg* de-asserts, the positive edge counter holds the total number of expected negative strobe edges. The multiplexer now outputs the Inequality Monitor result. When the strobe negative edge count becomes equal to the expected, the Inequality Monitor generates a ‘0’ at its output which causes the masking signal to de-assert. This completes the masking operation and resets both counters. The immediate resetting of the system allows it to operate flawlessly with consecutive READ operations.

Figure 2 illustrates a Verilog simulation of the DSMS scheme with a Micron DDR2 memory model at 533MHz, showing an example of operation. In the DFI interface, *dfi\_rddata\_en* is asserted for four *dfi\_clk* clock cycles, indicating that four negative DQS strobe edges are expected. In the JEDEC interface, the memory begins driving the DQS line (*dqs[0]*), and the signal arrives at the PHY

delayed by the time-of-flight through the PCB (*read\_dqs*). A glitch can be seen at the end of the postamble region, when the SDRAM drives the DQS strobe from low level to high-Z.

Inside the DSMS, *dfi\_rddata\_en* is registered (*dfi\_rddata\_en\_reg*) and releases the counters' reset (*internal\_reset\_n*). The *dfi\_rddata\_en\_reg* signal is also passed to the PDL (*pdl\_in*) which is programmed to produce a delay of 8 delay units (*cnf\_dsms\_taps*). The rising edge at the output of the PDL (*pdl\_out*) causes the masking signal (*mask*) to go high around the middle of the preamble region. During the high time of *dfi\_rddata\_en\_reg*, Counter-A increments with every *dfi\_clk* positive edge and measures the expected number of strobe negative edges (*expected[2:0]*). When the first falling edge of the strobe arrives, Counter-B increments (*actual[2:0]*). Every time the two counts become equal, the *not\_equal* flag de-asserts. When *dfi\_rddata\_en\_reg* de-asserts and the last strobe falling edge is received, the actual count equals the expected count and the *not\_equal* flag is selected as the masking signal, causing *mask* to drop and reset the counters. The resulting masked strobe (*masked\_dqs*) is now clean and can be used for data capture.

### III. IMPLEMENTATION

The two DSMS counters were developed as individual blocks, since they operate at different clock domains. The block of Counter-A includes the D flip-flop that registers the *dfi\_rddata\_en* signal, since they belong to the *dfi\_clk* domain and should share the same clock tree. The block of Counter-B includes the logic for the Inequality Monitor. The *expected* input driven by Counter-A is continuously compared to the internal count of Counter-B and the *not\_equal* flag is asserted if the two numbers are unequal or de-asserts if they are equal. The masking AND gate was replaced by a balanced NAND gate followed by another balanced NAND gate in inverter configuration. The choice of balanced gates ensures that minimum duty cycle distortion is induced on the DQS strobe.

As explained earlier, the de-assertion of the *dfi\_rddata\_en\_reg* signal forces the multiplexer to output the Inequality Monitor result. However, the relative delays of the DSMS components may cause the result of the Inequality Monitor to switch at that same moment. This race condition can lead to a glitch on the output of the multiplexer (the masking signal). This situation is illustrated in Figure 3.

To resolve this issue, a small delay (in the form of two buffers connected in series) is induced on the *dfi\_rddata\_en\_reg* line before it reaches the multiplexer, so that enough time is provided to the Inequality Monitor to stabilize its output. Thus, the final schematic of the DSMS is illustrated in Figure 4. The top-level DSMS layout in TSMC 90nm technology was built by assembly of the individual components and is illustrated in Figure 5.

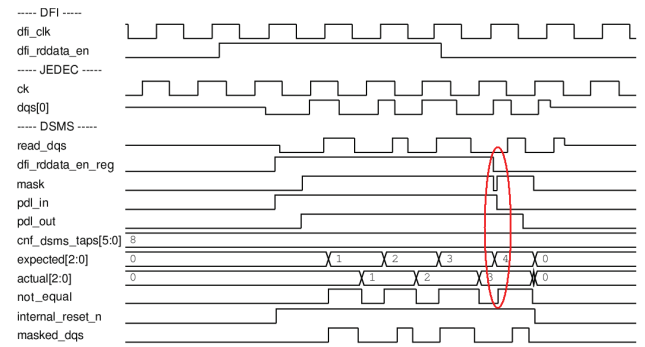


Figure 3. Example of possible glitch on the masking signal after de-assertion of *dfi\_rddata\_en\_reg*.

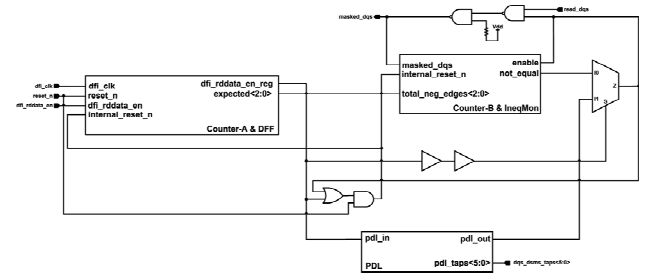


Figure 4. Top-level schematic of DSMS.

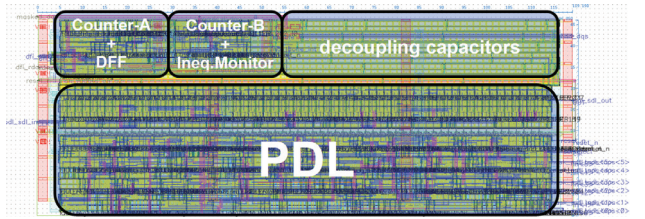


Figure 5. The layout of the implemented DSMS

The dimensions of the DSMS cell are  $H=46\mu\text{m}$ ,  $W=119\mu\text{m}$ . As expected, the majority of the space is taken up by the PDL. All empty space was filled with decoupling capacitors to reduce rail collapse during switching, assisting signal integrity on the critical DQS strobe.

### IV. POST LAYOUT SIMULATION RESULTS

Post-layout simulations of the DSMS scheme were performed including extracted parasitics. The testbench generates a DQS-like strobe with glitches in the beginning and end of the pulse stream. The preamble of the generated strobe is not as wide as the JEDEC specification requires (one clock period) but nevertheless the mask can be programmed to go high at the correct point, by proper setting of the PDL taps. The *dfi\_rddata\_en* signal is asserted for ten clock periods, designating that an equal number of negative edges are expected on the DQS strobe.

In real-world DDR2 memory modules, the postamble phase is typically half a clock period wide. This means that  $0.5 \times T_{CK}$  after the last negative strobe edge, the SDRAM begins its transition to the high-Z state and a possible glitch is generated.

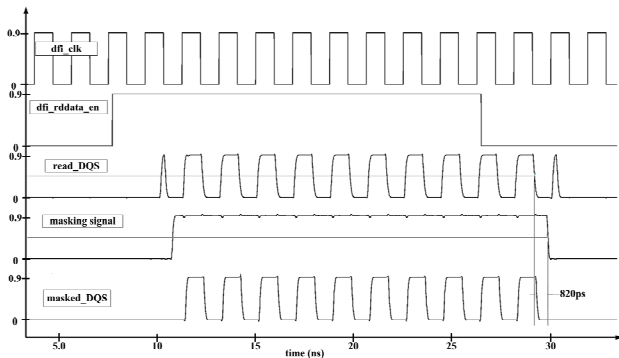


Figure 6. Worst Case transients (VDD=0.9V, models=ss, temp=125°C)

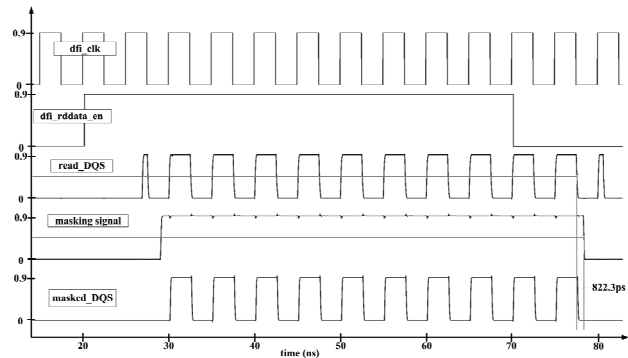


Figure 7. Worst Case transients (VDD=0.9V, models=ss, temp=125°C)

TABLE I. SUMMARY OF RESULTS AT 533 MHz

Post-layout results f=533MHz		
Specification	Required	Simulated
Mask shut-off delay	<938ps	502.4ps (TC) 348.5ps (BC) 820.0ps(WC)

Therefore, the DSMS is required to force the mask signal low before the  $0.5 \times T_{CK}$  point. Thus, the worst case scenario occurs when the clock period is minimum,  $T_{CK,min} = 1876ps$  at 533MHz. Thus, the DSMS must close the mask in less than  $0.5 \times 1876ps = 938ps$  after the last negative edge on the DQS strobe. The following paragraphs present the results of DSMS simulation at the lowest and highest operational frequency.

#### A. Post-layout simulation at 533MHz

For this scenario, the parasitics-extracted layout was simulated at 533MHz. In best case (BC), typical case (TC), and best case (BC) corners, the mask shut-off point meets the <938ps specification and produces a clean DQS strobe with ten pulses and no glitches (*masked\_dqs*). The measurements are summarized in Table I. Figure 6 presents the transient waveforms at worst case (WC) corner.

#### B. Post-layout simulation at 200MHz

For this scenario, the parasitics-extracted layout was simulated at 200MHz. Since the clock period is quite large at this frequency, the DSMS was expected to meet the  $<0.5 \times T_{CK}$  specification quite easily. Indeed, the mask shut-off point meets the <2500ps specification in all corners and produces a glitch-free DQS strobe with ten pulses. The measurements are summarized in Table II. Figure 7 presents the transient waveforms at worst case (WC) corner.

### V. CONCLUSIONS

In this work we presented the architecture, design and operation of a Dynamic Strobe Masking System for high speed DDR interfaces. The method of operation of this strobe masking system differs from the traditional techniques in the sense that it functions dynamically, adjusting the length of the masking signal in real-time based

TABLE II. SUMMARY OF RESULTS AT 200 MHz

Post-layout results f=200MHz		
Specification	Required	Simulated
Mask shut-off delay	<2500ps	502.4ps (TC) 347.8ps (BC) 822.3ps(WC)

on the incoming strobe. The proposed system requires only existing DFI signals for its operation, making it readily portable to any DFI-compatible Memory Controller.

The DSMS measures the number of negative edges expected on the strobe and then counts the received number of negative DQS edges. When the received edges match the expected, the masking signal drops generating a glitch-free strobe that can then be used by the PHY for safe data capture. The crucial requirement that the DSMS must satisfy is to shut off the mask in less than  $0.5 \times T_{CK}$ , after reception of the last expected DQS negative edge. The parasitics-extracted layout of the system in TSMC 90nm process was simulated and proved functional in the 200-533MHz range, meeting the mask shut-off specification in BC, TC, and WC simulation corners.

### REFERENCES

- [1] Brent Keeth, R. Jacob Baker, Briant Johnson and Freng Lin, DRAM Circuit Design, Wiley Interscience, 2008
- [2] Lior Amarilio, David Schkolnik, and Ophir Nadir "Method and Apparatus for DQS Postamble Detection and Drift Compensation in a Double Data Rate (DDR) Physical Interface", United States Patent No. 7,675,811 B2, March 9, 2010.
- [3] Jade M. Kizer, Sivakumar Doraiswamy, and Benedict Lau "Strobe Masking In a Signaling System Having Multiple Clock Domains", United States Patent No. 7,543,172 B2, June 2, 2009.
- [4] Peter Korger and Robert W. Moss "Method and Apparatus for Calibrating DQS Qualification in a Memory Controller", United States Patent No. 6,600,681 July 29, 2003.
- [5] James Magro "Method and Apparatus for Preventing Data Corruption during a Memory Access Command Postamble", United States Patent No. 6,918,016 July 12, 2005.
- [6] Haggai Telem, et al. "Qualified Data Strobe Signal for Double Data Rate Memory Controller Module", United States Patent 7,345,933 March 18, 2008.
- [7] DDR PHY Interface (DFI) Specification, Denali Software Inc., version 2.0, 2008.
- [8] JEDEC Standard JESD208, Specialty DDR2-1066 SDRAM, JEDEC Solid State Technology Association, 2007.