2010 IEEE Annual Symposium on VLSI

A novel 1.8 V, 1066 Mbps, DDR2, DFI-compatible, Memory Interface

A. Alexandropoulos[1], E. Davrazos[1], F. Plessas[1,2], M. Birbas[1,2]
[1]Analogies S.A., Patras, Greece
[2]Department of Telecommunication Systems and Networks, Technological Education Institute of Messolonghi,
Nafpaktos, Greece
alexis.alexandropoulos@analogies.eu

Abstract - **An innovative design of a 533 MHz DDR2 SDRAM PHY based on a common standard bus interface (DFI) and implemented in 90 nm standard CMOS process, is presented in this paper. Off-chip driver with calibrated strength, slew rate control, and on-die termination mechanism are utilized to provide improved signal integrity. Furthermore a DDR3-like I/O architecture and an appropriate calibration mechanism has been employed in order to reduce input capacitance. A Register-Controlled Delay Locked Loop (RCDLL) is included that measures the period of the external DFI clock to generate two stable clock phases (0°, 90°) and aligns it with the internal PHY clock. A novel Dynamic Strobe Masking System (DSMS) has also been employed which, in contrast to traditional techniques, dynamically adjusts the length of the masking signal in real-time, based on the incoming strobe. Finally, the PHY provides the necessary hooks for data capture training by an external calibration engine. Post layout simulation results demonstrate its robustness over process, voltage, and temperature variations.**

Keywords-DDR2; off-chip driver; on-die termination; DLL; memory interface; SDRAM

I. Introduction

With the emergence of DDR2 SDRAM the industry has experienced a shift from all soft memory controllers plus I/Os to memory controllers plus a hard physical interface (PHY). The hard PHY is physically designed for a specific process and foundry in order to achieve the performance required by today's DDR2 SDRAMs and has become a standard component of a DDR2 interface solution.

The DDR2 interface is specified by JEDEC to operate at rate of 400–800 Mbps, where few suppliers support even higher rates of up to 1066 Mbps. A wide parallel bidirectional data bus is employed, using SSTL I/Os, a single bidirectional strobe signal (DQS) and a masking (DM) signal for each group of 8 data bits (DQ). The strobe signal is not a free running clock, but is transmitted along with the relevant active data. JEDEC has defined the DDR2 SDRAM system in such a way so as to shift design complexity into the memory controller and PHY, in order to keep DRAMs as inexpensive as possible. This mandate left complexity in the development of the DDR2 PHY, resulting in significant design challenges. Furthermore, as the operation frequency increases, excessive skew and jitter have become serious problems. Several skew and jitter reduction techniques for up to 800 Mbps interfaces and SDRAMs have been proposed [1-5]. Another critical problem that has not yet been addressed in DDR2 interfaces or SDRAM implementations is the PHY's high input capacitance that tends to suppress the high frequency signals, thus reducing its frequency operating capability. In this work, a DDR2 memory interface implementation is presented operating at 1.8 V and fully complying with the corresponding JEDEC standards [8], In contrast to existing schemes, the proposed PHY's SSTL I/O drivers have been designed to be compatible with the DDR3 SSTL architecture [6-7]. Slew rate control and automatic impedance calibration have been furthermore employed. A novel dynamic masking system ensures that any glitches on the DQS input signals will not affect the specified operation. Section II, describes the proposed architecture of the DDR2 compatible memory interface, followed by implementation and post-layout simulation results in Section III. The paper's conclusions are given in section IV.

II. Architecture and Data Path

In general, a memory interface PHY sits between a Memory Controller (MC) and an SDRAM memory module, ensuring proper communication between different interface schemes, and converting single-rate data to double-rate data and vice-versa. The proposed memory interface (DDR2 PHY) consists mainly of three design blocks, namely the Read/Write datapath, the Register-Controlled Delay Lock Loop (RCDLL), and the SSTL I/Os (Fig. 1).

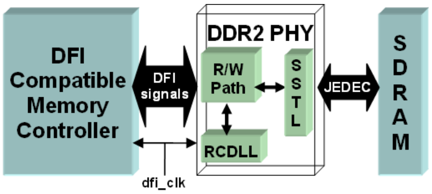


Figure 1. Architecture of the DDR2 interface system

A. Register-Controlled Delay-Locked Loop

The RCDLL subsystem is an all-digital design that performs the following functions: (i) Generates two clock phases from the incoming *dfi_clk*, namely 0° (*dfi_clk0*) and 90° (*dfi_clk90*) which are the clocks used by the rest of the PHY. (ii) Measures the period of *dfi_clk* and provides this measurement in terms of delay units (henceforth termed as 'taps'). The measurement is performed upon reset but can be repeated at

978-0-7695-4076-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ISVLSI.2010.49

any time using the *measure_req* signal. This feature can be used by an update/recalibration system to periodically check for variations in the internal delays of the PHY due to temperature gradients. (iii) Removes any skew between *dfi_clk* and *dfi_clk0***.** This relaxes the timing requirements of the DFI interface since it improves the timing budget for the interoperability between the PHY and the MC. In addition, skew removal ensures that all signals launched by the PHY are synchronous to *dfi_clk*, as specified by the DFI protocol [9].

The tap unit used in all delay lines inside the PHY is set as the propagation delay of two balanced NAND gates connected in series as inverters. The balanced rise and fall times of these gates ensure that minimum duty cycle distortion is induced on clock and strobe signals. The simulated post-layout tap delay in various PVT conditions is summarized in the following Table I.

TABLE I. POST-LAYOUT TAP DELAY

Process Corner /	Tap Delay (ps)			
Voltage	- 40 °C	27 °C	60 °C	120 °C
TT / 1V	43.9	47.3	49.2	52.6
SS / 0.9V	68.7	72.1	74.1	77.7
FF / 1.1V	30.8	33.7	35.0	39.6

It can be observed that the tap delay variation with temperature is ~10ps in the [-40°C, 120°C] temperature range, across all process corners.

The architecture of the RCDLL (Fig. 2) consists of the following sub-blocks: (i) Customized Time-to-Digital Converter (TDC) which generates a decoded 256-bit measurement of the *dfi_clk* period, in tap units. (ii) Encoder which converts the 256-bit period measurement into a 9-bit binary number (*period_taps<8:0>*). It also provides a 6-bit binary number (*deg90_taps<5:0>*) which corresponds to a quarter of the *dfi_clk* period (in taps). (iii) Bang-Bang Phase Detector (BBPD) that compares *dfi_clk* and the feedback clock (*dfi_clk0_buff*)**,** generating a shift-left/right decision, based on their phase relation, (iv) Shift Register: Depending on the shift decision from the BBPD, the Shift Register will shift its 192-bit output, (v) Digital-to-Time Converter (DTC_192) which is a programmable delay line of 192 taps, controlled by the Shift Register, (vi) Slave Delay Line (SDL) consisting of a 64-tap delay line (DTC_64) and a 6-to-64 Decoder that selects the amount of delay to be applied, in units of taps. This block is loaded with *deg90_taps* to produce an accurate 90^o phase shift of the input signal, (vii) dummy DTC_64: This is another 64-bit programmable delay line that is used as a dummy cell to compensate for the inherent delay of the DTC_64 inside the SDL. (viii) Control Finite State Machine (FSM): Times and sequences the various micro-operations of the DLL.

Upon reset, the Control FSM induces the measurement of the period of *dfi_clk*. On measurement completion, the FSM loads the T/4 tap number (*deg90_taps<5:0>*) to the SDL in order to produce *dfi_clk90*. After *dfi_clk0* and *dfi_clk90* have been stabilized and passed to the PHY, the RCDLL attempts to deskew the internal PHY clock, as it arrives at the clock-pins of the PHY flip-flops. In order to do this, the RCDLL provides a feedback input (*dfi_clk0_buff*) which connects to the clock pin of a PHY flip-flop at the leaf of the *dfi_clk0* clock tree, forming a feedback loop. Inside the DLL, the *dfi_clk* and *dfi_clk0_buff* phases are compared by the BBPD which decides on the addition or subtraction of delay on the *dfi_clk0_buff* signal through the DTC_192, until the two clocks align in time domain with ±1 tap accuracy. When this is detected, the RCDLL is considered locked and the FSM produces a *done* flag to indicate that all operations are complete. The RCDLL continues to monitor the two clocks and performs correctional shifts whenever their time difference exceeds the ±1 tap range.

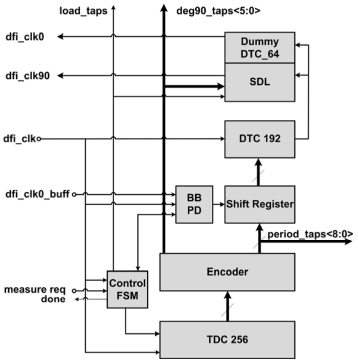


Figure 2. The block diagramm of the RCDLL subsystem

B. *DQS Strobe Masking*

In a real-world DDR2 memory interface, glitches are frequently observed on the DQS strobe during a memory read operation. These glitches occur during the preamble and postamble phases of the strobe and, if unfiltered, can be misinterpreted by the PHY as actual strobe edges and lead to unexpected system behaviour or latching of erroneous data. In contrast to traditional strobe masking techniques that rely on long calibration sequences, the proposed Dynamic Strobe Masking System (DSMS) works with existing DFI signals to provide dynamic masking and produce a clean strobe suitable for data capture (patent pending).

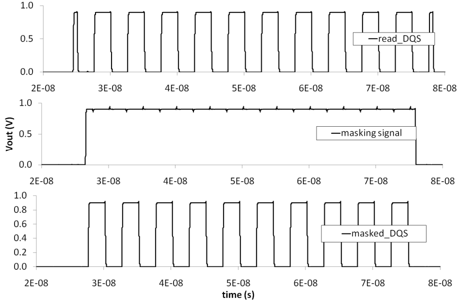


Figure 3. Post-layout simulation waveforms showing a strobe with glitches (top), masking signal from DSMS (middle), and resulting clean strobe (bottom).

C. *Digital Read/Write Path*

The digital part of the 8-bit Read/Write Path can be seen in Fig. 4. It is structured in a bit-sliced manner and consists of eight 1-bit DQ bit-slices along with one DM bit-slice and one DQS bit-slice. Each DQ slice includes the necessary modules for transmitting/receiving one DQ bit of data to/from the

SSTL I/Os. It also contains a Programmable Delay Line that allows an external (third party) data capture training engine to add or remove delay to each DQ bit individually so as to maximize the data valid window during memory-read operations. During memory writes, the DQS slice includes a Finite State Machine that generates a JEDEC-compliant strobe to the SDRAM, while in memory-reads the slice receives an incoming strobe, eliminates any spurious edges on the line and passes the glitch-free strobe to the DQ slices for data capture. A Programmable Delay Line inside the DQS slice allows an external (third party) data capture calibration engine to shift the read-DQS strobe to the optimum sampling position of the DQ data eye. The DM slice transmits the data masking signal during a memory write operation. The DQ and DM slices use the 0^o and 90^o phases of the clock (*dfi_clk0* and *dfi_clk90* respectively) while the DQS slice uses only *dfi_clk0*. Finally, an ADDR/CTRL slice is added which contains D flip flops (clocked by *dfi_clk0*) that latch the address and control signals from the DFI interface and pass them to the SSTL I/Os for transmission to the SDRAM.

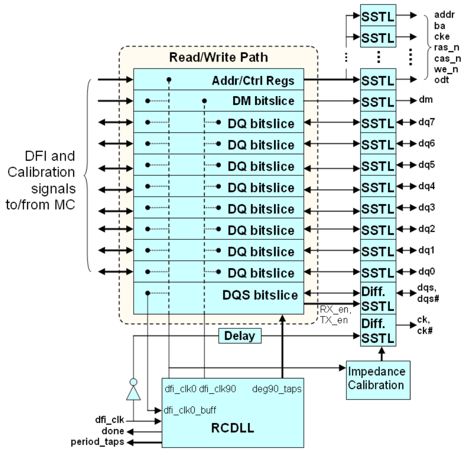


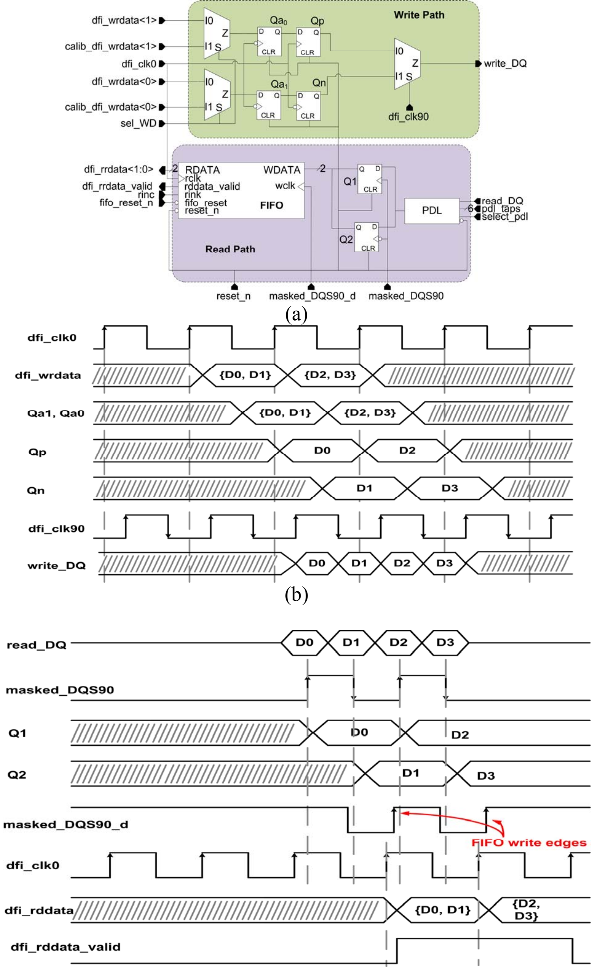
Figure 4. Internal architecture of the DDR2 PHY

C1. *The DQ bit-slice*

The architecture of the DQ bitslice is illustrated in Fig. 5a. In the Write Path, two multiplexers select between SDR (Single Data Rate) DFI data coming from the Memory Controller or some calibration engine during training of the PHY. The incoming SDR data transmitted on the rising edge of the *dfi_clk* clock are latched on the negative edge of *dfi_clk0*. The latched data are then passed to a positive-edge and a negative-edge register that feed the inputs of the serializing multiplexer. This pipeline ensures that the pulse width of the serialized data (*write_DQ*) is equal to that of the *dfi_clk90* clock (Fig. 5b), meeting the t_{DIPW} specification of the JEDEC standard.

In the Read Path, the incoming DDR DQ bit passes through a 64-tap Programmable Delay Line (PDL) which can be controlled by an external calibration engine during data capture training. In other words, the DQ data bit can be shifted to achieve the optimum sampling position with respect to the incoming DQS strobe. A typical memory read operation dataflow is illustrated in Fig. 5c. The aligned DQ data are latched on both edges of *masked_DQS90* which is optimally placed in the center of the data eye, after calibration is complete. The sampled data are then written into the FIFO on the rising edges of *masked_DQS90_d* and read out of the FIFO on the *dfi_clk0* domain, along with the *dfi_rddata_valid* flag. The FIFO can also be controlled (namely reset or allowed to be read) externally through the *rinc* and *fifo_reset_n* signals. This feature is commonly utilized by third-party data-capture calibration engines.

The DM slice consists of only the Write Path since this is a unidirectional signal, used to mask data transmitted by the PHY during a memory-write operation.



(a)

(b)

(c)

Figure 5. Architecture of the DQ bitslice (a), example of dataflow during a write operation (b), example dataflow during a read operation (c).

C2. *The DQS bit-slice*

During memory-read operations, the DQS slice eliminates any glitches on the incoming DQS line and from this generates the strobe phases necessary for capturing and synchronizing the incoming data. During memory-write operations, the DQS slice utilizes an FSM that generates a JEDEC-compliant DQS strobe with its edges positioned around the centre of the serialized DQ data-eye.

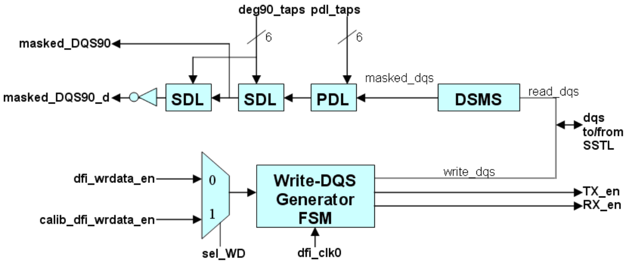


Figure 6. Block diagram of the DQS slice

Inside the DQS bit-slice (Fig. 6), the incoming read-DQS strobe passes through the DSMS which eliminates any spurious edges on the line. Following that, the clean strobe goes through a PDL and a SDL. The SDL produces a 90^{o} phase shift of the strobe with respect to data. At low frequencies this accurate shift could be adequate for safe data capture. At higher frequencies, the strobe sampling point can be finely adjusted by an external data capture calibration engine using the PDL module. The resulting *masked_DQS90* strobe is passed to the DQ slices for DDR data capture. Since the incoming DQ bits are captured on both rising and falling edges of the *masked_DQS90* signal, the clock that stores the captured data into the FIFO must ideally be a clone of the *masked_DQS90* signal but 270^{o} phase shifted (Fig. 5c). However this would require a long delay line, especially at low frequencies. An easier solution is to invert the *masked_DQS90* signal and shift its phase by 90^{o}. This is achieved by passing the *masked_DQS90* signal through another SDL and inverting its output to produce *masked_DQS90_d*.

During a memory-write operation, the DQS bit-slice uses an internal FSM (*Write-DQS Generator*) that generates a write-strobe to be transmitted to the SDRAM together with the write-data. The strobe is produced from *dfi_clk0* so that t_{DQSS}, t_{DQSH}, and t_{DQSL} JEDEC specifications are easily met. Moreover, the signal timing of the signals inside the FSM is such that $t_{WPRE} = t_{WPST} = 0.5 \times T_{CK}$ (where T_{CK} is the clock period), meeting the corresponding JEDEC specification. Finally, the FSM also generates the necessary receive/transmit enable signals (*RX_en*, *TX_en*) used by the SSTL I/Os. For the generation of the write-DQS strobe, only balanced gates were used, to ensure minimal duty cycle distortion (DCD) of the generated strobe.

D. Off-Chip Driver and On-Die Termination

The main building blocks include the output driver/termination driver (Transmitter/Termination), the input buffer (receiver), and the Impedance Calibration Mechanism. The Transmitter/Termination provides 1.8V for data-in = "1" and 0V for data-in = "0" and also the appropriate termination values which are 50, 75 and 150 Ohm for DDR2 operation. The input buffer (receiver) translates the SSTL input voltage levels to 0-1 V CMOS levels, and also ensures that the output signal has sufficiently fast rise and fall times when applied to the CMOS core. In Fig. 7 the proposed architecture for the merged output driver/termination driver is shown. There are 8 p- and 8 n-type legs with a total resistance of 150 Ohm. In DDR2 operation, the full strength driver has an output impedance of 18.75 Ohm, represented by enabling eight 150 Ohm pull-up and pull-down legs. An extra couple of p- and n-type legs of 300 Ohm resistance each are used to achieve the 150 Ohm termination value in DDR2 operation. Furthermore, in order to minimize the crowbar current through the output transistors, a "break-before-make" predriver circuitry (BbM) is employed.

Figures 8 and 9 show the schematics of the n- and p-type legs respectively. The pull-up leg has a passive resistor (180 Ohm) in parallel with five p-channel devices so as to control the total resistance of the leg and provide a tuning range sufficiently large to compensate for any process, voltage and temperature (PVT) variations.

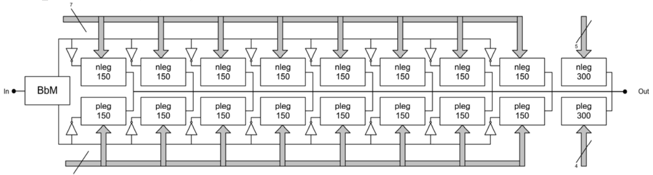


Figure 7. Output driver architecture

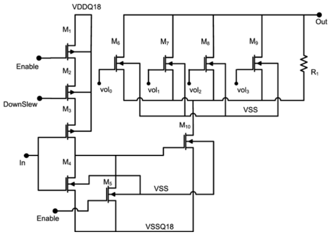


Figure 8. The n-leg of the proposed output driver / termination driver

Thus, the total resistance of the leg can be always tuned to the desired 150 Ohm value. The pull-down leg is similar to the pull-up leg except for using four n-channel devices due to the different R_{ON} resistance of the P- and N-MOS devices. The 300 Ohm p- and the 300 Ohm n-legs are designed accordingly using a 320 Ohm passive resistor. The slew rate can be controlled by coordinating the slope of the gate signal of the main driver (i.e. M_{11} for the p-leg or M_{10} for the n-leg). The slew-rate control signals *UpSlew* and *DnSlew* are generated simply by using a voltage divider formed by an internal and an external resistor. The value of the external resistor is chosen such that the resultant *UpSlew* or *DnSlew* signals give the desired slew rate. The schematic of the proposed input buffer is shown in Fig. 10. It consists of three stages: the rail-to-rail preamplifier, the decision stage, and the output buffer [10]. The preamplifier consists of both a PMOS and NMOS differential amplifier whereas the decision circuit uses positive feedback from the cross-gate connection of M_{21} and M_{23} to increase the gain of the decision element. The output buffer converts the output of the decision circuit into a logic signal, and includes an inverter as an additional gain stage in order to isolate any load capacitance. The proposed calibration scheme using two external precision resistors (R_{ZQ}) is shown in Fig. 11. The main concept is that the resistance of a dummy n-leg and a dummy p-leg of the output driver including the passive resistor and the combination of transistors is compared to the external resistor R_{ZQ} (150 Ohm). This comparison is performed by comparing the actual voltage at the ZQ_1 or ZQ_2 point to an internally generated reference voltage, VDDQ/2 (or VDDQ/4), through the use of a simple voltage comparator. R_{ZQ} is placed between the ZQ_1 pin and power and between the ZQ_2 pin and ground, respectively.

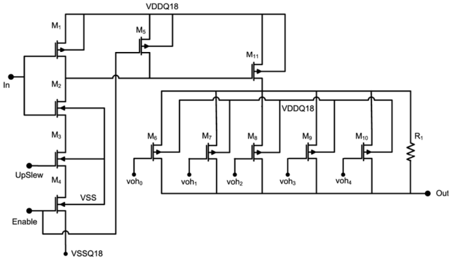


Figure 9. The p-leg of the proposed output driver / termination driver

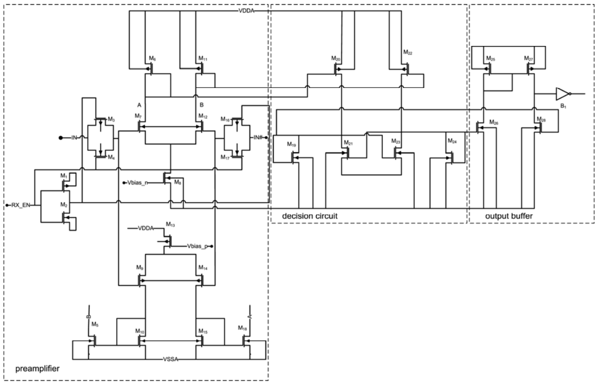


Figure 10. The proposed input driver

The transistors are binary weighted to provide a resolution of 2% of R_{ZQ}. A 5-bit (Voh) and a 4-bit (Vol) digital code is used to select or deselect the p-channel and the n-channel devices respectively. The impedance of each leg is adjusted automatically to 150 Ohm to account for process, voltage and temperature variation through the Impedance Calibration Mechanism. The calibration process of the SSTL I/O involves the execution of two separate binary search algorithms, one for the pull-up and one for the pull-down resistor networks of the SSTL. The successive approximation procedure followed by these algorithms is performed by two finite state machines (FSMs) which run sequentially, with the pull-down FSM (pdFSM) executing first. As seen from the outside world, the two FSMs operate as a single machine with the *sstl_calib_act* input of the pdFSM being the actual 'calibration activation' signal. Upon system reset, the pdFSM is at state IDLE which corresponds to Vol=0000 and remains there until a pulse on the *sstl_calib_act* signal is received. After activation, it jumps to the READY state which is followed by the first state (START, Vol=1000) of the pull-down binary search algorithm. At each following rising edge of the clock, the pdFSM samples the output of the Comparator and branches to the corresponding next value of the Vol bus, namely the next FSM state. These successive jumps form an approximation path through the algorithm tree and continue until the pdFSM reaches the final Vol binary setting. This value is the current optimum configuration for the pull-down resistor network of the SSTL. When the pdFSM reaches its final state, the *calib_done* flag is asserted to indicate that the pull-down resistor network has been calibrated. Then the pull-up FSM (puFSM) is activated, following a similar binary search algorithm until the pull-up resistor network is calibrated. It is evident that the pull-up and pull-down calibration algorithms are complementary to each other, as is the design of the resistor networks. It should be noted that both FSMs operate using a clock frequency that is a sub-multiple of the *dfi_clk0* frequency, i.e. the clock signal of the FSMs is the system clock divided by 2 (in the case that the system clock has a frequency between 200-400 MHz) or by 4 (in the case that the system clock has a frequency of 533 MHz). This provides adequate time margin for the stabilization of the comparator decision and eliminates the need for a majority filter at its output. The final optimal Voh and Vol binary values are stored into registers so that they remain stable and available to all SSTL IOs until they are updated by the next re-calibration procedure (initiated by assertion of the *sstl_calib_act* signal).

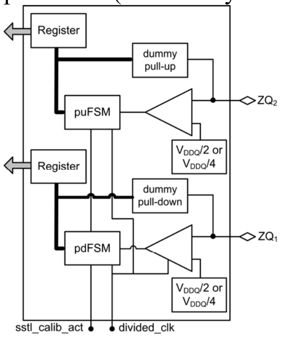


Figure 11. Resistance calibration mechanism

The resistance of the n- and p-leg can be calibrated at VDDQ or at a midpoint voltage VDDQ/2. For the p-leg the resistance remains within the nominal ±3 Ohm spec when calculated at either VDDQ (VSS) or VDDQ/2. For the n-leg the voltage comparison level should be lowered to VDDQ/4 when the resistance is calculated at VDDQ (VSS). In this case the resistor R_{ZQ} is placed between the ZQ_1 pin and VDDQ/2.

It has been found through simulation experiments that the extra 300 Ohm couple of p- and n-type legs can be calibrated using the shifted (right by 2 bits) values of the Voh and Vol signals respectively. Thus, two shift registers have been employed in the impedance calibration scheme.

III. PERFORMANCE

The main objective of the physical design was to verify the functionality and performance of the circuit and the validity of the proposed architecture. For this purpose the choice of a 90 nm CMOS technology was deemed to be adequate, and specifically the 90 nm, General Purpose, 1.0/1.8 V process of TSMC was employed. Figure 12 shows the layout of the proposed DDR2 PHY. The die area is 1 mm^2 with a large amount of decoupling capacitors built around it. The resistance value of the 150 Ohm p- and n-leg as a function of the temperature and voltage supply variations at a typical corner is shown in Table II. It is obvious that all the output and the termination resistances can be achieved using the appropriate combination of the 8 available p- and n-legs or the extra 300 Ohm p- and n- leg. The resistance value of the merged driver (when operating as termination driver) has been also simulated and can be controlled to be 150, 75 or 50 Ohm.

TABLE II. SIMULATION RESULTS FOR THE PULL-UP AND PULL-DOWN LEGS RESISTANCE

Conditions	Pull-up	Pull-down
27°C/1.8V	151.0	145.7
100°C/1.7V	151.4	152.4
0°C/1.9V	151.5	149.0

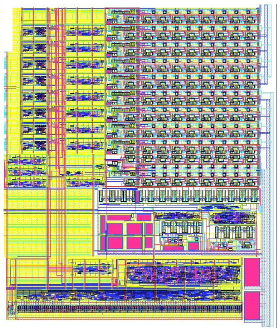


Figure 12. The layout of the proposed DDR2 PHY

The slew rate can be controlled from 0.5 V/ns to 15 V/ns for a 4 pF load using the control signals *UpSlew* and *DnSlew*. The *UpSlew* is generated by using a voltage divider formed by a 1 kOhm internal resistor connected to VDDQ and a variable external resistor which should be connected to VSS. The *DnSlew* is generated by using a voltage divider formed by a 1 kOhm internal resistor connected to VSS and a variable external resistor which should be connected to VDDQ. In Fig. 13, a post-layout simulation of a memory-read operation with burst length=8, is shown. The rising and falling edges of *masked_DQS90* sample the sequence 1, 0, 0, 1, 0, 0, 1, 0 on the *read_DQ* line**.** Thus, the expected data words are 10, 01, 00, and 10 which are indeed the values on the *dfi_rddata* bus under the *dfi_rddata_valid* flag, proving correct functionality (1066 Mbps, worst case conditions). In Fig.14, a post-layout simulation of a memory-write operation is presented, showing the centering of DQS/DQS# crossings around the middle of the DQ data-eye, thus maximizing setup/hold timing margins (1066 Mbps, worst case conditions).

IV. CONCLUSION

An innovative DDR2 memory PHY has been described. Post layout simulation results have demonstrated high performance in all DDR2 modes of operation up to 1066 Mbps. Furthermore, the proposed architecture addreses reduction of input capacitance issues and thus the high frequency signal is not suppresed. A novel dynamic masking system successfully removes the glitches appearing on the DQS line ensuring reliable data capture while the existence of appropriate slew rate and impedance controlled mechanisms offer reduced output skew.

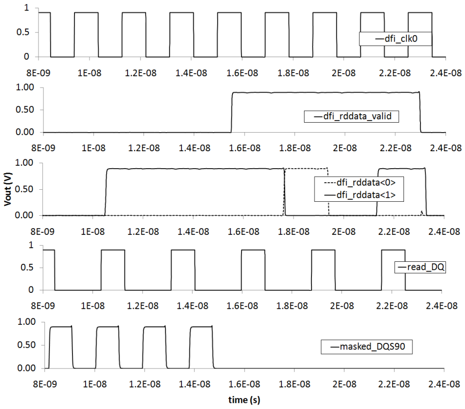


Figure 13. Transient response (Read) at 1066 Mbs, slow-slow, 0.9 V, 125 °C

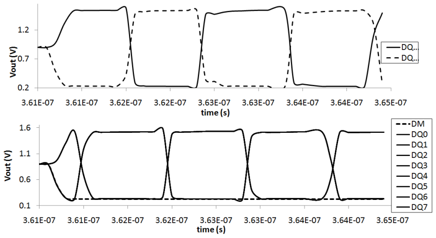


Figure 14. Transievt response (Write) at 1066 Mbs, slow-slow, 0.9 V, 125 °C

V. REFERENCES

[1] Changsik Yoo, Kye-Hyun Kyung, Kyunam Lim, Hi-Choon Lee, Joon-Wan Chai, Nak-Won Heo, Dong-Jin Lee and Chang-Hyun Kim "A 1.8V 700-Mb/s/pin 512-Mb DDR-II SDRAM With On-Die Termination and Off-Chip Driver Calibration", IEEE Journal of Solid-State Circuits, vol. 39, pp. 941 – 951, June 2004.

[2] Tatsuya Matano, Yasuhiro Takai, Tsugio Takahashi, Yuusuke Sakito, Isamu Fujii, Yoshihiro Takaishi, Hiroki Fujisawa, Shuichi Kubouchi, Seiji Narui, Koji Arai, Makoto Morino, Masayuki Nakamura, Shinichi Miyatake, Toshihiro Sekiguchi, and Kuniaki Koyama, "A 1-Gb/s/pin 512-Mb DDRII SDRAM Using a Digital DLL and a Slew-Rate-Controlled Output Buffer", IEEE Journal of Solid-State Circuits, vol. 38, pp. 762 - 768, May 2003.

[3] Hiroki Fujisawa, Masayuki Nakamura, Yasuhiro Takai, Yasuji Koshikawa, Tatsuya Matano, Seiji Narui, Narikazu Usuki, Chiaki Dono, Shinichi Miyatake, Makoto Morino, Koji Arai, Shuichi Kubouchi, Isamu Fujii, Hideyuki Yoko, and Takao Adachi, "1.8-V 800-Mb/s/pin DDR2 and 2.5-V 400-Mb/s/pin DDR1 Compatibly Designed 1-Gb SDRAM With Dual-Clock Input-Latch Scheme and Hybrid Multi-Oxide Output Buffer ", IEEE Journal of Solid-State Circuits, vol. 40, pp. 862 - 869, April 2005.

[4] Feng Lin, Jason Miller, Aaron Schoenfeld, Manny Ma, and R. Jacob Baker "A Register-Controlled Symmetrical DLL for Double-Data-Rate DRAM", IEEE Journal of Solid-State Circuits, vol. 34, pp. 565 - 568, April 1999.

[5] Jae Joon Kim, Sang-Bo Lee, Tae-Sung Jung, Chang-Hyun Kim, Soo-In Cho, Beomsup Kim, "A low-jitter mixed-mode DLL for high-speed DRAM applications", IEEE Journal of Solid-State Circuits, vol. 35, pp. 1430 -1436, Oct 2000.

[6] T. Farrell, "Core Architecture Doubles MEM Data Rate", Electronic Engineering Times Asia, December 16, 2005.

[7] Brent Keeth, R. Jacob Baker, Briant Johnson and Freng Lin, DRAM Circuit Design, Wiley Interscience, 2008.

[8] JEDEC Standard JESD208, "Specialty DDR2-1066 SDRAM", JEDEC Solid State Technology Association, 2007.

[9] DENALI SOFTWARE INC., "DDR PHY Interface (DFI) Specification", version 2.0, April 2008.

[10] R. Jacob Baker, CMOS Circuit Design, Layout, and Simulation, IEEE Press (Wiley Interscience), 2008.