www.ietdl.org

Published in IET Computers & Digital Techniques
Received on 5th October 2010
Revised on 10th January 2011
doi: 10.1049/iet-cdt.2010.0143

ISSN 1751-8601

# Advanced calibration techniques for high-speed source–synchronous interfaces

*F. Plessas A. Alexandropoulos S. Koutsomitsos E. Davrazos M. Birbas*

*Analogies S.A., Patras Science Park, Stadiou Str., 26504 Patras, Greece*
*E-mail: fotis.plessas@analogies.eu*

**Abstract:** Advanced and dynamic calibration techniques for maximising the link performance of parallel source–synchronous interfaces are introduced and demonstrated in this study, using as a case study a 533 MHz DDR2 SDRAM memory interface implemented in 90 nm standard complementary metal-oxide-semiconductor (CMOS), whereas most of them have been validated at 800 MHz too. A novel dynamic strobe masking system (DSMS) has also been employed which, in contrast to traditional techniques, adjusts dynamically the length of the masking signal in real time, based on the incoming strobe. Furthermore, optimal data capture is achieved by employing a fast bit-deskew calibration engine, while also a novel I/O calibration scheme is included. Post-layout simulation results demonstrate that the dynamic calibration and skew compensation techniques employed improve the timing margin while providing advanced robustness over process, voltage and temperature variations.

## 1 Introduction

DDR3/2 SDRAMs use a source–synchronous interface where the data strobe is aligned with the data and system clock while achieving data rates up to 1.6 Gbps per pin. A wide parallel bidirectional data bus is employed, using Stub Series Terminated Logic (SSTL) Input/Output (I/O)s, a single bidirectional strobe signal and a data masking (DM) signal for each group of data bits (DQ). The strobe signal is not a free running clock, but is transmitted along with the relevant active data. Joint Electron Devices Engineering Council (JEDEC) has defined the DDR3/2 SDRAM system in such a way so as to shift design complexity into the MC and PHY in order to keep DRAMs as inexpensive as possible. This mandate left complexity in the development of the DDR3/2 PHY, resulting in significant design challenges. Furthermore, as the operation frequency increases, timing uncertainty issues, duty cycles of clocks, phase differences of multiple clocks, clock deskew with respect to data or strobe have become serious problems.

When data rates increase beyond 1066 Mbps the data valid window becomes shorter and the available margin is not sufficient to account for voltage and temperature variations. Hence there is a need to dynamically calibrate the placement of the strobe, compensate for any skew between the paths of individual DQ bits, as well as to remove glitches on the DQS line and adjust the output impedance of the I/Os.

Most of the calibration techniques for up to 1.6 Gbps memory interfaces and SDRAMs deal mainly with output impedance and on-die termination adjustment [1–3] techniques, whereas there are only few reported works addressing Data Strobe (DQS) masking and bit deskew [4–9] issues. In this work, we present a combined implementation of three different calibration mechanisms (DQS strobe, bit deskew and I/O calibration) in the same memory physical interface using novel, advanced and dynamic techniques for each separate calibration scheme. More particularly, regarding DQS strobe calibration a dynamic strobe masking system (DSMS) is proposed which works with existing DFI signals to provide dynamic masking and produces a clean strobe suitable for data capture. These characteristics make it the first dynamic DFI-compatible strobe qualification system to our knowledge, thus advantageously differentiating from most existing strobe qualification techniques, like the ones reported in [4–8], which operate in a static fashion and require special signals from the memory controller (MC), whereas also, in [9] a statistical random sampling technique is used to measure and correct the duty cycle of a clock to produce source–synchronous signals and to adjust the phase of the incoming strobe to correctly capture data.

Furthermore, in contrast to most DQ deskew methods proposed in literature [9] which are time consuming, a very fast per-bit deskew algorithm for time alignment of the DQ bits is employed here, which detects the worst-case edges of the data-valid window and appropriately places the DQS strobe thus ensuring optimised data capture (with the whole procedure needing just 3 μs to complete).

In addition, the proposed scheme includes a DDR3-like SSTL I/O calibration scheme that has been presented by the authors in a previous work [10] which except for the use of a DDR3-like architecture, it can be also calibrated both at VDDQ and midpoint voltage VDDQ/2, while furthermore, more accurate calibration and slew rate control algorithms compared to existing approaches [11] have been employed.

Finally, start-up calibration is performed at power-up, while tracking calibration performance periodically, is employed to maintain optimum functionality against voltage and temperature variations in a transparent (to the user)

way, in the sense that, these mechanisms are enabled whenever needed without affecting the operation of the interface. It must be noted that this is true for all the three different calibration schemes proposed.

For demonstration purposes we have used a DDR2 memory interface implementation, fully compliant with the corresponding JEDEC standard [12].

It should be mentioned that although this work demonstrates the proposed techniques using a DDR2 memory interface, these can be readily applied in any source–synchronous interface. The rest of the paper is organised as follows. Section 2 describes the proposed architecture of the calibration blocks together the memory PHY, followed by implementation and post-layout simulation results in Section 3. The paper's conclusions are given in Section 4.

## 2 Memory PHY and calibration blocks

In general, a memory interface PHY sits between a MC and an SDRAM memory module, ensuring proper communication between different interface schemes (DFI/JEDEC), and converting single-rate data to double-rate data and vice versa. The memory interface (Fig. 1), used to validate the proposed calibration schemes, consists mainly of two design blocks, namely the HARD PHY (read/write datapath, register-controlled delay-lock loop – RCDLL, DQS masking and the SSTL I/Os) and the SOFT PHY (DQ bit deskew, configuration register, update handler – UH).

### 2.1 Register-controlled delay-locked loop

A RCDLL subsystem has been designed which [10]: (a) generates two clock phases from the external clock (dfi_clk), namely 0° (dfi_clk0) and 90° (dfi_clk90) to be used by the PHY, (b) measures the period of the dfi_clk and provides this measurement in terms of delay units (taps) and (c) removes any skew between dfi_clk and dfi_clk0. The measurement is performed upon reset but can be repeated at any time using a measurement request signal. This feature can be used by the update/recalibration system to check periodically for variations in the internal delays of the PHY because of temperature gradients by measuring the clock period and comparing the new measurement with previous ones. The de-skewing relaxes the timing requirements of the DFI interface since it balances the timing budget for interoperability between the PHY and the MC. In this sense, we may consider that the clock alignment procedure is a kind of calibration of the DFI interface.

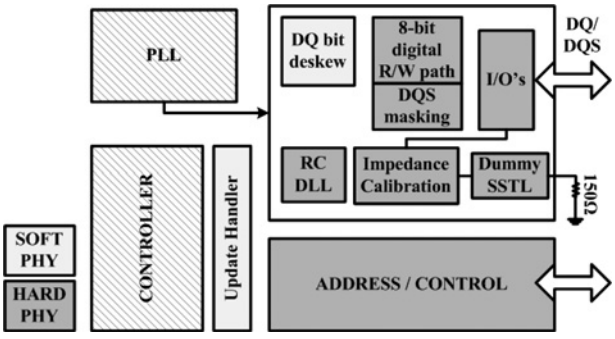


**Fig. 1** *Blocks included in the memory interface*

### 2.2 DQS strobe calibration (masking)

In a real-world DDR2 memory interface, glitches are frequently observed on the DQS strobe during a memory read operation [4–8, 13, 14]. A voltage bump is produced when the DDR module attempts to tri-state the DQS strobe whereas the DQ lines have not reached a well-defined voltage level. When this happens, an oscillation is induced at the DQ lines which propagates to the power/ground circuit and in turn produces the DQS glitch. In the preamble, the DQ lines are stably held at high-impedance state and thus a glitch on the DQS line during the transition from high-$Z$ to '0' is unlikely. Therefore designers are mainly concerned with glitches in the postamble region, when the DDR module attempts to tri-state the DQS and DQ lines simultaneously and a voltage bump is possible. Nevertheless, an occasional preamble glitch can be a problem and must also be prevented reliably. If unfiltered, these glitches can be misinterpreted by the PHY as actual edges and lead to unexpected system behaviour or false data capture.

This issue is typically addressed by means of a masking pulse which is AND-ed with the strobe and properly timed to remove any possible glitches. The resulting strobe is then used for data capture within the PHY. Ideally, a masking pulse will assert in the middle of the preamble region and de-assert immediately after reception of the last DQS falling edge. According to JEDEC, the typical READ preamble time, $t_{RPRE}$, is one clock cycle, measured from the point the DQS strobe leaves the high-impedance state to the first rising edge of the strobe. This is illustrated in the Fig. 2.

The minimum specified preamble time is $t_{RPRE,min} = 0.9t_{CK}$ whereas the minimum specified postamble time is only $t_{RPST,min} = 0.4t_{CK}$. Therefore the major design challenge is to guarantee the de-assertion of the masking pulse within the short postamble time under all conditions. One common DQS strobe qualification methodology has been presented by Amarilio *et al.* [8]. 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. Finally, the masking signal is placed somewhere within the measured valid window.

Depending on the granularity of the mask shifting, this technique 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 does not account for variations on the DQS line, thus a re-calibration procedure may be required to re-position the masking signal at the optimum position. In order to prevent the recalibration 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 restores it after the recalibration procedure is complete. The extra logic needed 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 MC–PHY integration is added by the fact that different strobe masking implementations require special input signals from the MC, with time specifications that serve their specific architecture. This hinders the porting of the PHY to different MCs, since this might require modification of the MC design to provide the specific signals required by the strobe masking engine.

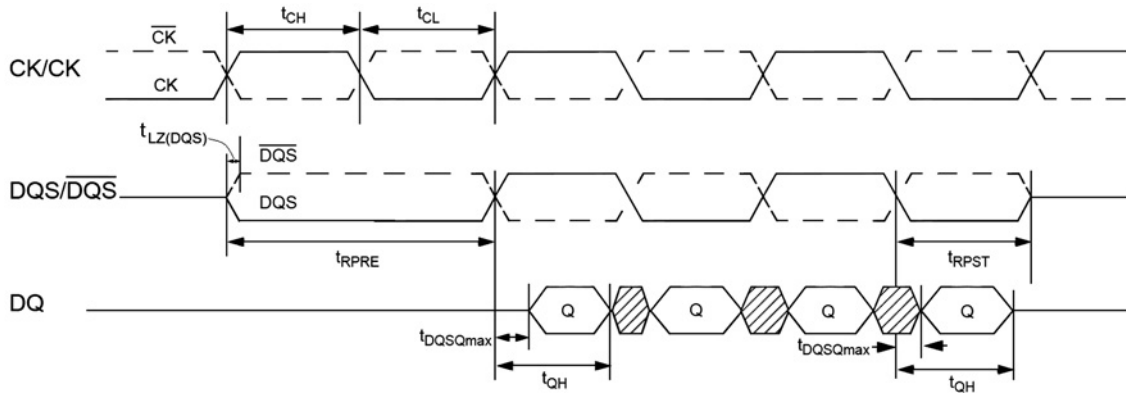


**Fig. 2** *Data output (read) timing according to JEDEC [12]*

With the emergence and establishment of the DFI protocol for MC–PHY interfaces [15], the advantage of a strobe masking system that utilises only DFI signals becomes obvious.

The proposed DSMS uses a masking signal to qualify the expected pulse stream on the DQS line, while masking out all other activity on it. It utilises two counters: one uses the DFI signal dfi_rddata_en to calculate the number of expected pulses on the strobe, whereas the other counts the actual number of DQS pulses received. The assertion of the masking signal is programmed to occur around the middle of the preamble time of the read-DQS based on the Printed Circuit Board (PCB), packaging and IO delay of a specific application. In the case of a 1066 Mbps interface ($t_{CK} = 1876$ ps) the minimum preamble time is $t_{RPRE,min} = 1688$ ps. Taking into account the DQS low-impedance timing specification of JEDEC, $t_{LZ(DQS)}$, and derating for jitter gives an estimated worst-case preamble time of about 1.2 ns, in the middle of which the mask assertion point must be set. Typical DDR PHY programmable delay lines (PDLs) provide a resolution of 40–50 ps, thus the mask assertion point can be easily placed in the middle of the comparably long 1.2 ns worst-case preamble region, while allowing adequate margin for variation. The de-assertion of the masking signal is induced immediately after the two counters' counts become equal. This ensures that the masking signal is deactivated within the postamble region. This dynamic method of strobe qualification creates a masking signal with variable pulse length that adapts to each incoming strobe, avoiding truncation of DQS pulses because of early mask de-assertion which is a possible issue with other static strobe masking architectures.

The DSMS is included as a subsystem within the DQS bit-slice (Fig. 3) which was designed to enable JEDEC-compliant memory read/write operations so that the effectiveness of these calibration techniques could be tested. During memory-read operations, the glitch-free strobe from the DSMS passes through a series of PDLs to generate the strobe phases required by the DQ bit-slices to capture and synchronise the incoming data. One of these PDLs is controlled by the DQ-bit deskew calibration system, which shifts the DQS strobe appropriately until optimal data sampling is achieved. The remaining slave delay lines (SDLs) are controlled by the RCDLL and produce accurate 90° phase shifts to generate the strobes used by the DQ bit slices. Fig. 4 illustrates post-layout transient waveforms of DSMS operation at worst-case conditions.

### 2.3 DQ-bit deskew calibration

The architecture of the DQ bit slice is illustrated in Fig. 5. In the write path, two multiplexers select between SDR (single data rate) DFI data coming from the MC or some

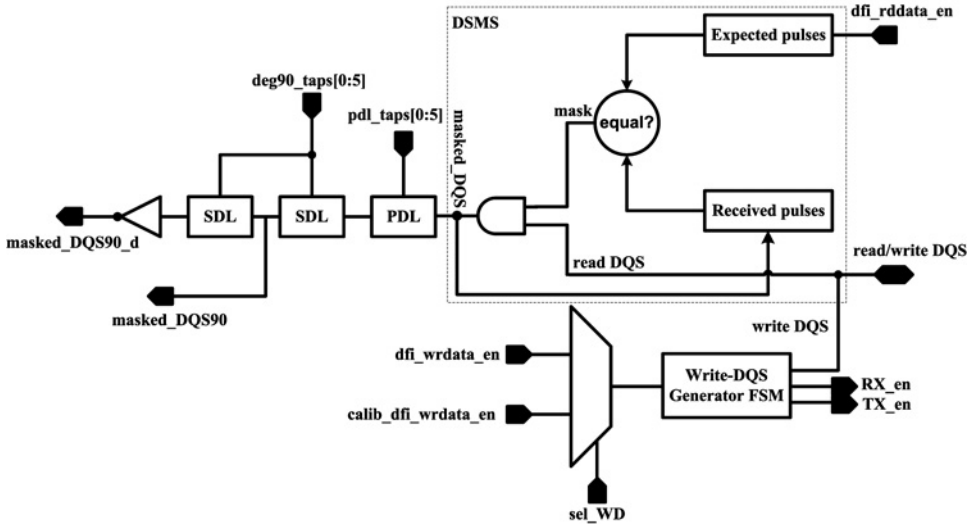


**Fig. 3** *Block diagram of the DQS bit slice*




doi: 10.1049/iet-cdt.2010.0143

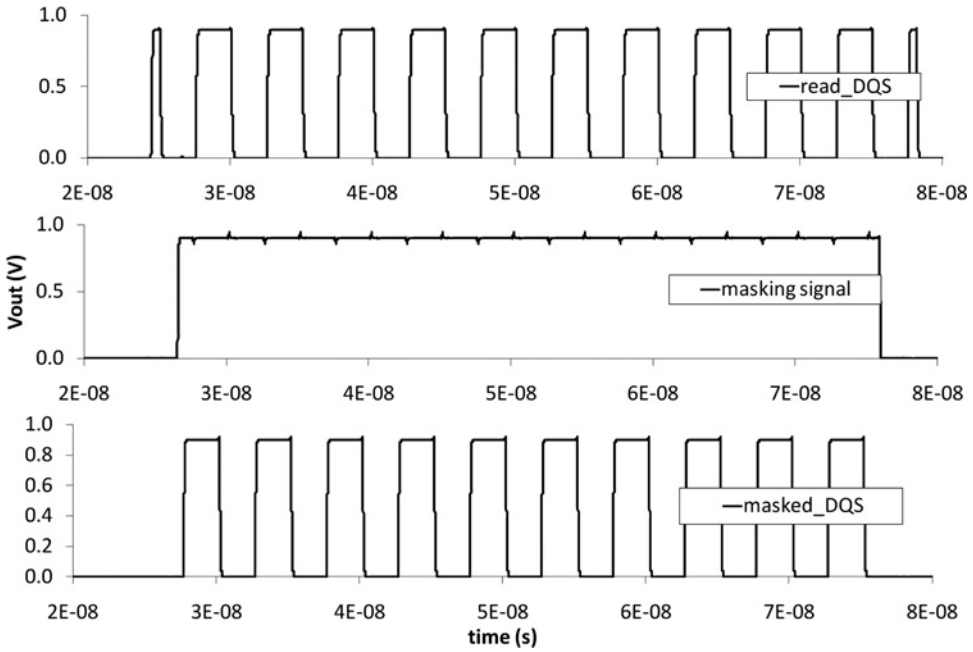


**Fig. 4** *Post-layout simulation waveforms showing a strobe with glitches (top), masking signal from DSMS (middle) and resulting clean strobe (bottom)*

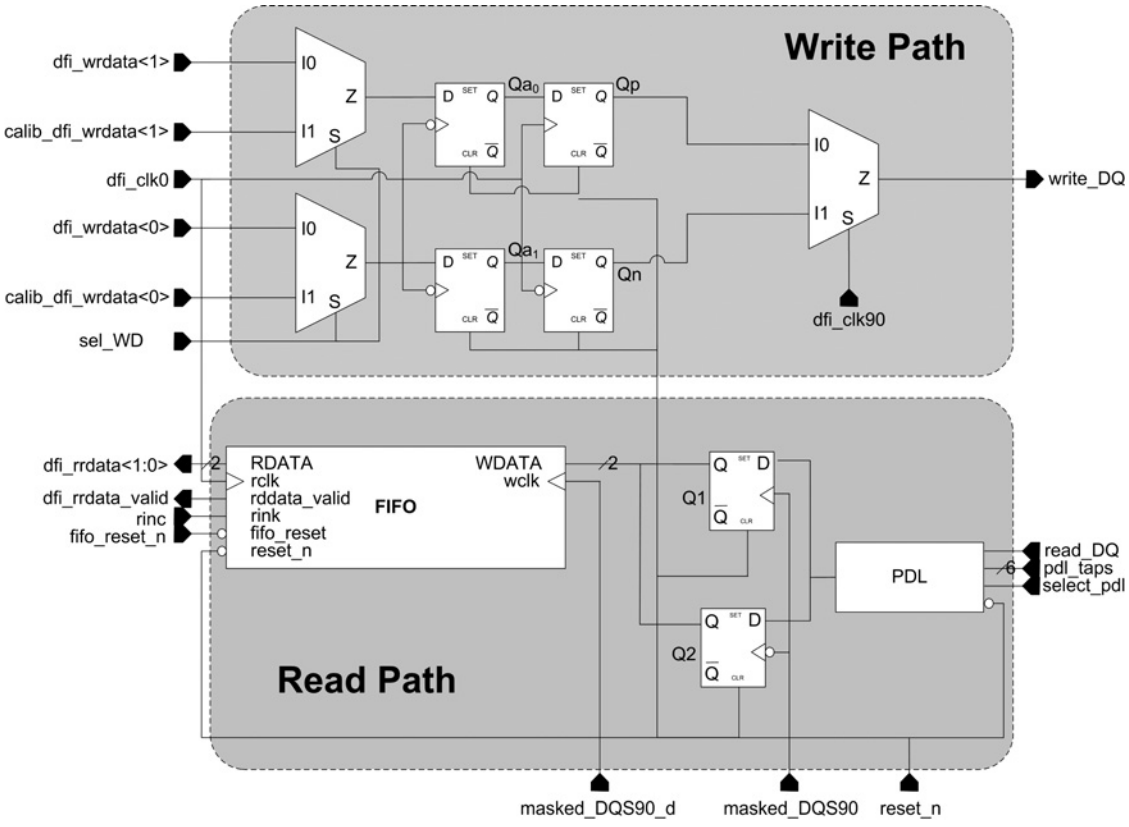


**Fig. 5** *Architecture of the DQ bit slice*

calibration engine during the training mode of operation 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 serialising multiplexer. This pipeline ensures that the pulse width of the serialised data (write_DQ) is equal to that of the dfi_clk90 clock, meeting the $t_{DIPW}$ specification of the JEDEC standard.

In the read path, the incoming DDR DQ bit passes through a 64-tap 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. The aligned DQ data are latched on both edges of masked_DQS90 which is optimally placed in the centre of the data eye, after calibration is complete. The sampled data are then synchronised to the DFI clock domain through a First In, First Out (FIFO).

Although the SDRAM issues 8-bit data aligned with the respective strobe (DQS) within certain timing ($t_{DQSQ}$), shifting the strobe by 90° is not enough to ensure proper data capture. PCB trace mismatches among the data lines and strobe line, noise coming from simultaneous switching of DQ lines as well as power supply noise and, finally, wire bonds and delay mismatches inside the PHY package result in the shrinking of the data valid window (time window within which the values of all eight DQs have been settled). Given that the interface should operate at high speeds, the need for time alignment of the DQs relative to DQS is crucial.

The proposed algorithm for time alignment of the DQs detects the worst-case edges of the data-valid window and places DQS in the middle of the window thus ensuring proper data capture.

The algorithm initially writes a data pattern of 'FFh' and '00h' in the first 64 bits of the SDRAM array, shifts DQs by 32 PDL taps and DQS by 16 PDL taps, then issues consecutive eight-word memory-read operations in order to detect the presence of the pattern in the data captured. A conceptual state diagram of the algorithm is shown in Fig. 6. In each memory-read, the data read are compared to the written pattern and the data lines (DQs) found to have invalid data are shifted back by a single PDL tap. A valid window edge is detected when valid data are captured in all data lines within all data words read from memory inside a single read burst (since SDRAM delays on DQs vary over time, the algorithm tries to read as many words as possible to cover as many as possible delay variations). After detecting the first valid window edge, the algorithm continues to shift all DQ lines in the same direction until it reads a single invalid datum on any of the lines. The tap settings of the PDLs during the previous read operation are considered as the last valid edge of the data valid window. Based on the tap settings upon which the data valid window edges where found, the DQS PDL is set (in terms of number of tap delays) in such a way so as to place the strobe in the middle of the data valid window. Note that prior to writing the pattern and upon programming the DQS PDL, the algorithm reads (from the SDRAM and stores into the PHY) and restores (from the PHY to the SDRAM) respectively, the data originally stored in the first 64 bits of the SDRAM so as to avoid memory data corruption. This enables the UH to call the calibration engine whenever it is considered that data calibration is needed.

It also should be noted that the above-mentioned data pattern triggers simultaneous switching noise on the SDRAM DQ outputs. The algorithm can be setup with a different data pattern to allow the user to trigger other effects (e.g. crosstalk).

Architecturewise, the algorithm employs an FSM which controls the nine PDLs placed inside the DQ-bit and DQS slices. The timing resolution regarding both the detection of the valid window edges and the strobe placement depends entirely on the tap delay of the PDLs. This makes the algorithm appropriate for memory interfaces with tighter timing (such as DDR3), just by redesigning the PDLs with

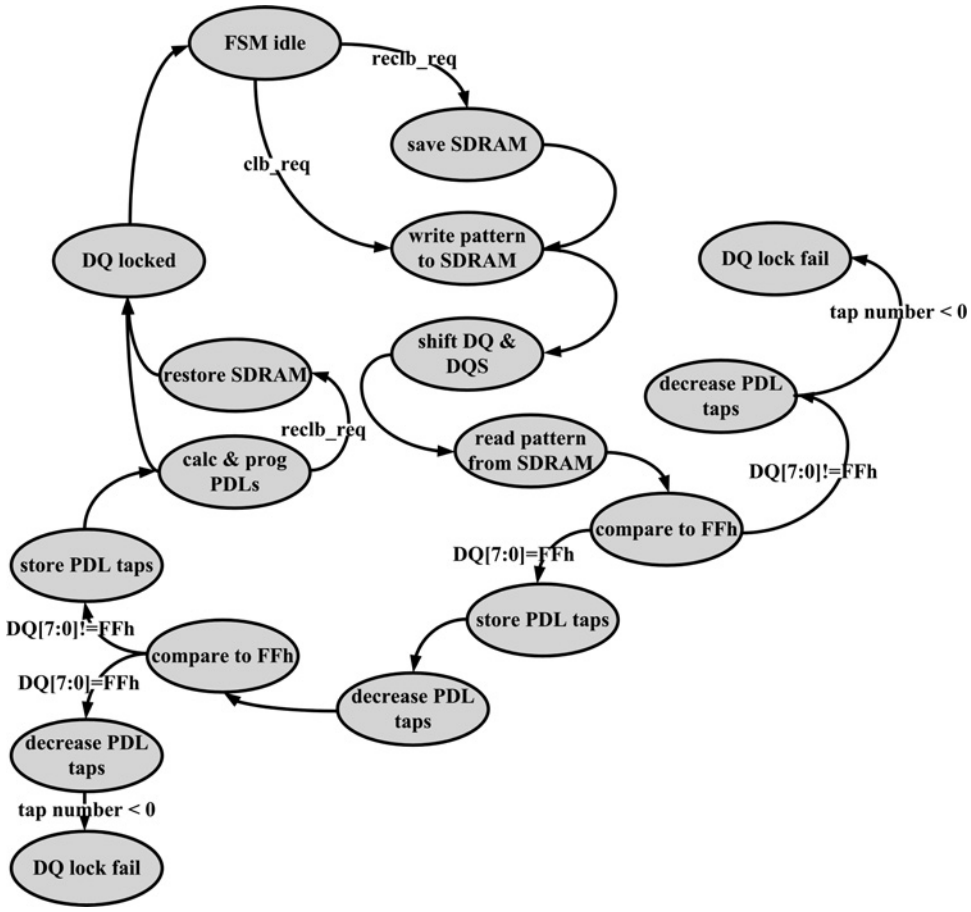


**Fig. 6** *Conceptual state diagram of the DQ bit de-skew algorithm*




doi: 10.1049/iet-cdt.2010.0143

smaller per-tap delay. Moreover, and given that real-world memory interfaces are wider than 8 bits, the algorithm can be scaled-up so as to be able to calibrate wider data buses in groups of eight data lines with one data strobe per group.

In contrast to calibration methods proposed in literature [9] which are time consuming, the hereby proposed calibration algorithm needs only about 3 μs to complete. Considering that any DDR2 SDRAM needs at least 200 μs during the power-up sequence [12], the calibration time for the proposed algorithm adds a trivial overhead to the total power-up time of the SDRAM, thus enabling it to be part of the DDR2 power-up sequence in commercial solutions and/or even to let it run as re-calibration option during DDR2 operation.

In post-layout simulations, a DFI-compliant DDR2 MC was connected to the DFI interface of the PHY and deskew calibration was carried out upon power-up of the system, then the MC issued consecutive 32-word write and read bursts on a Micron DDR2-1066 SDRAM component. Fig. 7 shows the time alignment of DQ bits with respect to DQS in the beginning of the algorithm. It is obvious that DQS either latches erroneous data or latches data during their transitions resulting in timing violations and therefore in inaccurate data capture. Fig. 8 shows a part of a read burst after calibration, where DQS latches data within the data valid window, thus ensuring correct data capture.

### 2.4 Off-chip driver and on-die termination calibration mechanism

A DDR3-like SSTL driver demonstrated in a previously reported work of the team [10] has been incorporated. Except for the use of a DDR3-like architecture, the resistance of the *n*- and *p*-leg can be calibrated at Output Supply Voltage (VDDQ) and midpoint voltage VDDQ/2. Furthermore, more accurate search algorithms compared to existing approaches [11], and slew rate control have been employed.

### 2.5 Calibration update

Another characteristic feature of the proposed architectures is the transparent method operation in the sense that these are enabled whenever needed without affecting the functionality of the interface. Thus, the three calibration

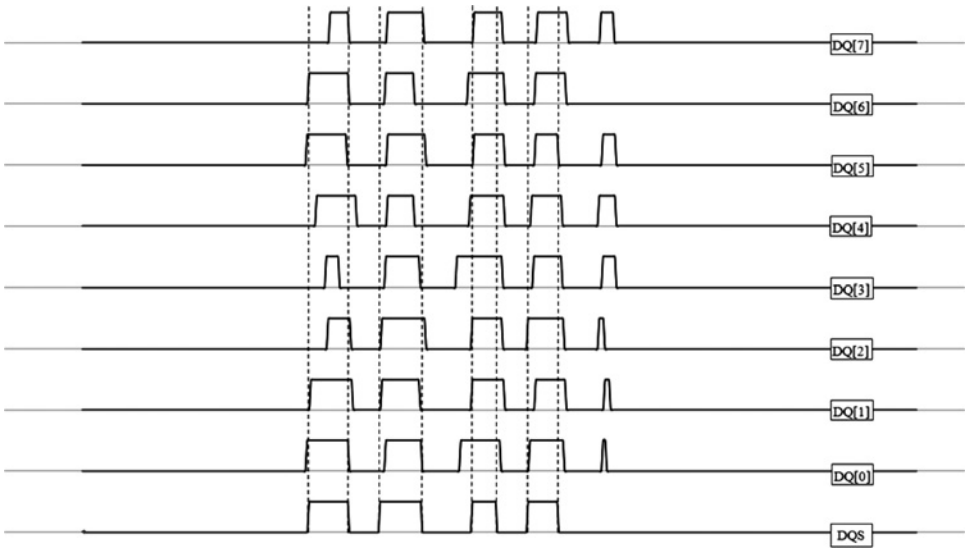


**Fig. 7** *Time alignment of DQ bits to DQS in the beginning of the calibration algorithm*

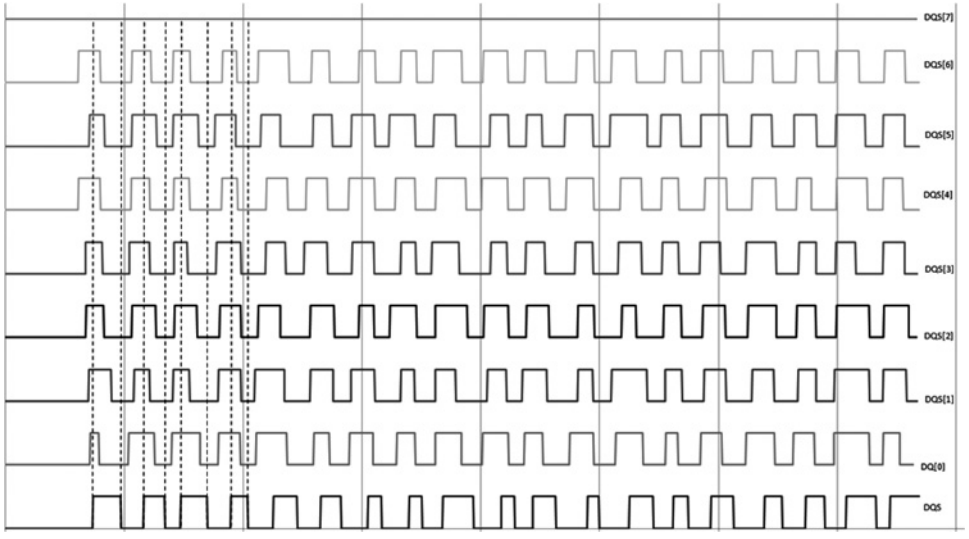


**Fig. 8** *Part of a read burst after calibration*

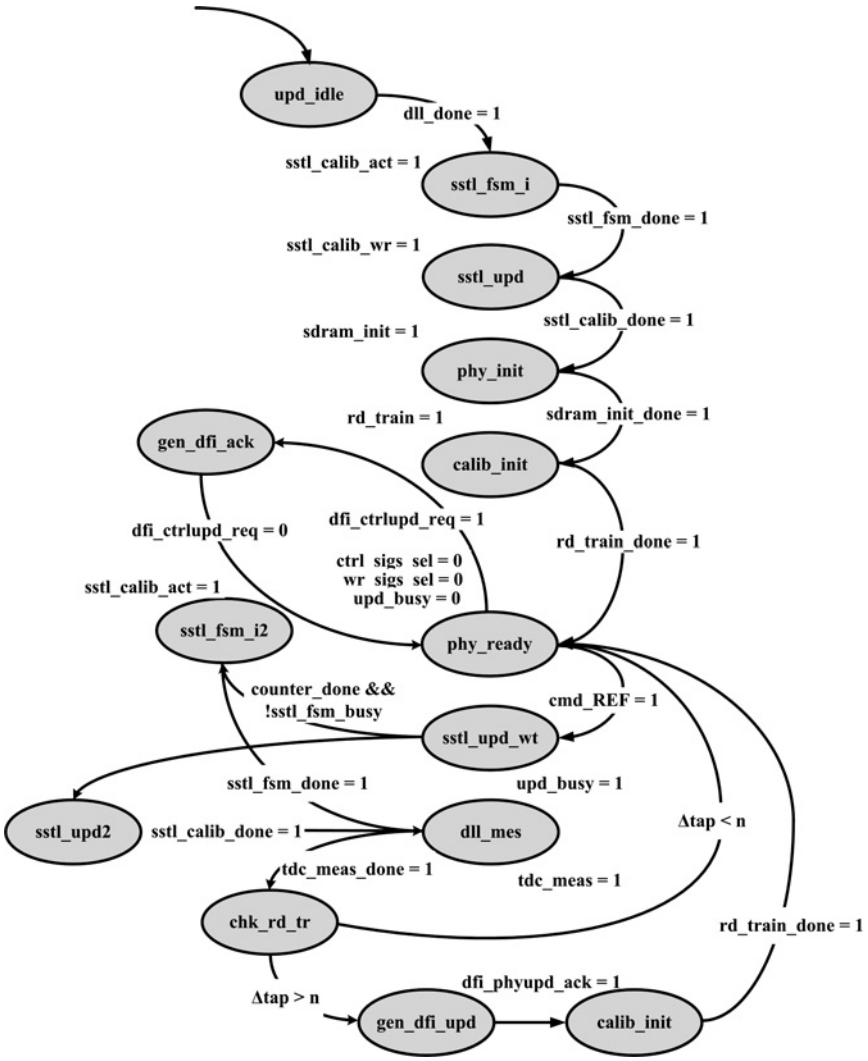


**Fig. 9** *State diagram of the UH*

mechanisms are controlled by the UH. The UH ensures that dynamic calibration is performed whenever is needed. Specifically, it employs an FSM which detects Refresh commands issued on the DFI and, within Refresh intervals, it requests a period measurement from the RCDLL and then compares it to the previous (or initial) measurement. Any difference will indicate that the tap delays within PDLs and SDLs have been altered, assuming that temperature variations along the PDLs of the DQ/DQS slices and the PDLs of the RCDLL are similar. The allowed difference in taps, $n$, before performing a recalibration is programmable to allow flexibility. Owing to the correlation between the period ($T$) measurement inside the RCDLL and the values ($T/4$) loaded to the SDLs, a value of $n = 4$ is considered adequate.

When the measurement difference exceeds the specified $n$ value, the UH initiates an update request on the DFI (through the DFI update interface [15]) and the calibration FSM is called to deskew the DQ lines again. At the same time, the SSTL impedance calibration is initiated during a Refresh interval and updates the drivers with the calculated values within the next refresh interval, since a single interval does not provide enough time for it to complete calculations. This two-interval calibration update is performed periodically, with the period being a programmable parameter. The state diagram of the UH is shown in Fig. 9.

## 3 Performance

The main objective of the physical design was to verify the key properties of the proposed calibration techniques. For this purpose, the choice of a 90-nm CMOS technology was considered to be adequate, and specifically the 90 nm, general purpose, 1.0/1.8 V process of TSMC was employed. The simulation environment of our test chip included a memory interface verification IP with real memory models. Fig. 10 shows the layout of the implemented DDR2 PHY incorporating the three proposed calibration modules.

The die area is 1 mm$^2$ with a large amount of decoupling capacitors placed around it. In Fig. 11*a*, 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 Mb/s, worst-case conditions). In Fig. 11*b*, a post-layout simulation of a memory-write operation is presented also, showing the centring of DQS/DQS# crossings around the middle of the DQ data-eye, thus maximising setup/hold timing margins (1066 Mb/s, worst-case conditions).




doi: 10.1049/iet-cdt.2010.0143

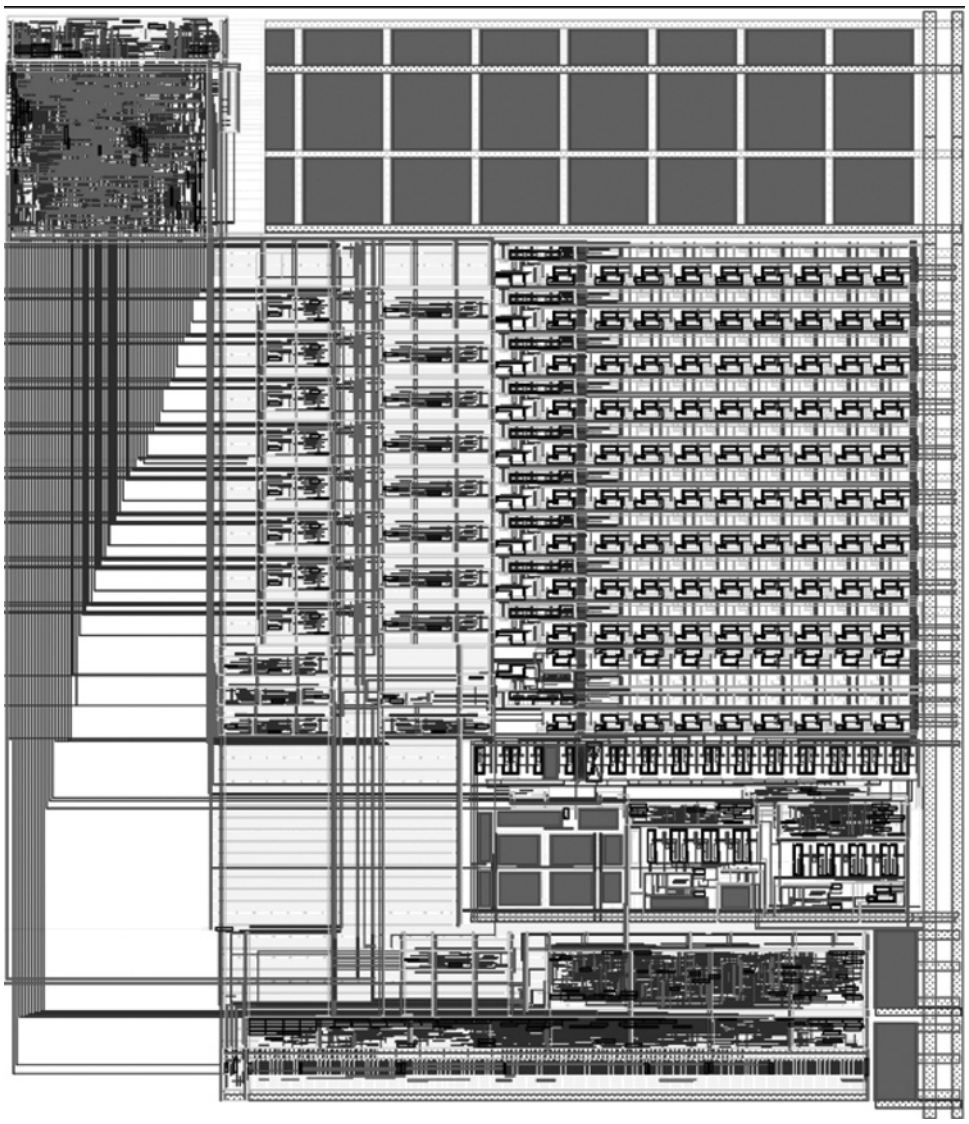

**Fig. 10** *Layout of the memory PHY including the proposed calibration mechanisms*

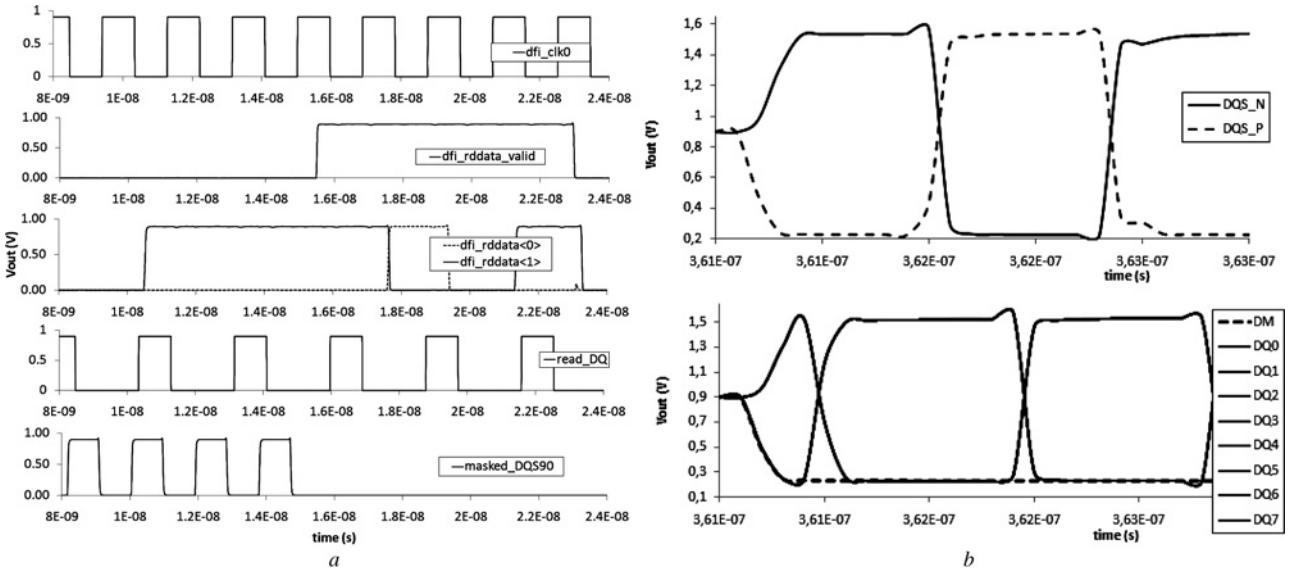


**Fig. 11** *Transient response*

*a* (Read) at 1066 Mbs
*b* (Write) at 1066 Mbs
At slow–slow corner with 0.9 V and 125°C

**Table 1** Timing parameters

| Parameter | Proposed DDR2-PHY | JEDEC specification for DDR2-1066 | Margin | Notes |
|---|---|---|---|---|
| DQ/DM setup time, $t_{DS}$ | ≃310 ps | 200 ps | 110 ps | JEDEC base value is de-rated for DQS/DQS# worst-case input slew rate |
| DQ/DM hold time, $t_{DH}$ | ≃310 ps | 240 ps | 70 ps | JEDEC base value is de-rated for DQS/DQS# worst-case input slew rate |
| DQ/DM minimum input pulse width, $t_{DIPW}$ | ≃905 ps | 657 ps | 248 ps | JEDEC values correspond to measured values at SDRAM pins, while PHY values correspond to values as transmitted by PHY (i.e. at PHY pins) |
| DQS high/low pulse width, $t_{DQSH}/t_{DQSL}$ | ≃905 ps | 657 ps | 248 ps | |
| DQS-to-CLK skew, $t_{DQSS}$ | ≃360 ps | −469 to +469 ps | 109 ps | |
| DQS write preamble, $t_{WPRE}$ | ≃$0.5t_{CK}$ | >$0.35t_{CK}$ | – | |
| DQS write postamble, $t_{WPST}$ | ≃$0.5t_{CK}$ | 0.4–$0.6t_{CK}$ | – | |

Table 1 presents post-layout simulation results of some of the most critical DDR2 timing requirements and the corresponding specifications from the JEDC standard. All values are well within the required limits, leaving adequate margin for possible signal integrity de-rating efforts through the PHY-SDRAM channel.

## 4 Conclusions

Three different advanced calibration methods, employing novel techniques for maximising the link performance of parallel source–synchronous interfaces, have been proposed and demonstrated, using a 533 MHz DDR2 SDRAM PHY implemented in 90 nm standard CMOS. The architecture employed is such that these techniques can be applied in any source–synchronous interface (DDR2/3, QDR2/3 etc.). The proposed architecture achieves reduction of input capacitance and thus the high-frequency signal is not suppressed. A novel dynamic masking system uses DFI existing signals to successfully remove 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. The proposed algorithm for time alignment of the DQs detects successfully the worst-case edges of the data-valid window and places DQS in the middle of the window thus ensuring optimum data capture.

Post-layout simulation results using a specialised verification IP and real memory models have verified the successful operation of the calibration mechanisms in data rates up to 1066 Mb/s.

## 5 References


1 Yoo, C., Kyung, K.-H., Lim, K., *et al.*: 'A 1.8 V 700-Mb/s/pin 512-Mb DDR-II SDRAM with on-die termination and off-chip driver calibration'. Proc. IEEE Int. Solid-State Circuits Conf. on Digest of Technical Papers, San Francisco, 2003, pp. 312–496

2 Matano, T., Takai, Y., Takahashi, T., *et al.*: 'A 1-Gb/s/pin 512-Mb DDRII SDRAM using a digital DLL and a slew-rate-controlled output buffer', *IEEE J. Solid-State Circuits*, 2003, **38**, (5), pp. 762–768

3 Fujisawa, H., Kubouchi, S., Kuroki, K., *et al.*: 'An 8.1-ns column-access 1.6-Gb/s/pin DDR3 SDRAM with an 8:4 multiplexed data-transfer scheme', *IEEE J. Solid-State Circuits*, 2007, **40**, (1), pp. 201–209

4 Kizer, J.M., Doraiswamy, S., Lau, B.: 'Strobe masking in a signaling system having multiple clock domains'. United States Patent 7,543,172 B2, June 2009

5 Korger, P., Moss, R.: 'Method and apparatus for calibrating DQS qualification in a memory controller'. United States Patent 6,600,681, July 2003

6 Magro, J.: 'Method and apparatus for preventing data corruption during a memory access command postamble'. United States Patent 6,918,016, July 2005

7 Telem, H., Yoeli, H., Glazer, O., *et al.*: 'Qualified data strobe signal for double data rate memory controller module'. United States Patent 7,345,933, March 2008

8 Amarilio, L., Schkolnik, D., Nadir, O.: 'Method and apparatus for DQS postamble detection and drift compensation in a double data rate (DDR) physical interface'. United States Patent 7,675,811 B2, March 2010

9 Bhatti, R., Denneau, M., Draper, J.: 'Data strobe timing of DDR2 using a statistical random sampling technique'. Proc. 50th IEEE Int. Midwest Symp. on Circuits and Systems, Montreal, August 2007, pp. 1114–1117

10 Alexandropoulos, A., Davrazos, E., Plessas, F., Birbas, M.: 'A novel 1.8 V, 1066Mbps, DDR2 DFI-compatible, memory interface'. Proc IEEE Annual Symp. on VLSI, Kefalonia, Greece, July 2010, pp. 387–392

11 Keeth, B., Baker, R.J., Johnson, B., Lin, F.: 'DRAM circuit design' (John Wiley & Sons, Inc., New Jersey, 2008)

12 JESD208: 'Specialty DDR2–1066 SDRAM' (JEDEC Solid State Technology Association, 2007)

13 Tyhach, J., Wang, B., Sung, C., *et al.*: 'A 90-nm FPGA I/O buffer design with 1.6-Gb/s data rate for source-synchronous system and 300-MHz clock rate for external memory interface', *IEEE J. Solid-State Circuits*, 2005, **40**, (9), pp. 1829–1838

14 Kurd, N., Barkatullah, J., Dizon, R., Fletcher, T., Madland, P.: 'A multigigahertz clocking scheme for the Pentium® 4 microprocessor', *IEEE J. Solid-State Circuits*, 2001, **36**, (11), pp. 1647–1653

15 Denali Software Inc.: 'DDR PHY interface (DFI) specification' version 2.0 (Denali Software Inc., 2008)





doi: 10.1049/iet-cdt.2010.0143