CACTI-IO Technical Report

CS2012-0986, UC San Diego CSE Department

Norman P. Jouppi[3], Andrew B. Kahng[1,2],
Naveen Muralimanohar[3] and Vaishnav Srinivas[1]

UC San Diego [1]ECE and [2]CSE Departments, La Jolla, CA. {abk, vaishnav}@ucsd.edu

[3]HP Labs, Palo Alto, CA. {norm.jouppi, naveen.muralimanohar}@hp.com

August, 2012

Abstract

We describe CACTI-IO, an extension to CACTI that includes power, area and timing models for the IO and PHY of the off-chip memory interface for various server and mobile configurations. CACTI-IO enables quick design space exploration of the off-chip IO along with the DRAM and cache parameters. We describe the models added to CACTI-IO that help include the off-chip impact to the tradeoffs between memory capacity, bandwidth and power.

This technical report also provides three standard configurations for the input parameters (DDR3, LPDDR2 and Wide-IO) and illustrates how the models can be modified for a custom configuration. The models are validated against SPICE simulations and show that we are within 0-15% error for different configurations. We also compare with measured results.

Contents

Chapter 1

Introduction

The interface to the DRAM, including the PHY (physical layer front-end block that typically exists between the controller and the I/O circuit), I/O circuit (IO) and interconnect, is becoming increasingly important for the performance and power of the memory subsystem [14, 15, 16, 23, 28, 34]. As capacities scale faster than memory densities [6], there is an ever-increasing need to support a larger number of memory dies, especially for large-capacity server systems [26]. Mobile systems can afford to use multi-chip package (MCP) or stacked-die point-to-point memory configurations; by contrast, servers have traditionally relied on a dual-inline memory module (DIMM) to support larger capacities. With modern server memory sizes exceeding 1 TB, the contribution of memory power can reach 30-57% of total server power [34], with a sizable fraction (up to 50% in some systems) coming from the off-chip interconnect. The memory interface incurs performance bottlenecks due to challenges with interface bandwidth and latency. The bandwidth of the interface is limited by (i) the data rate, owing to the DRAM interface timing closure, signal integrity over the interconnect, and limitations of source-synchronous signaling [3, 38], and (ii) the width of the bus, which is often limited by size and the cost of package pins.

CACTI [4] is an analytical memory modeling tool which can calculate delay, power, area and cycle time for various memory technologies. For a given set of input parameters, the tool performs a detailed design space exploration across different array organizations and on-chip interconnects, and outputs a design that meets the input constraints. CACTI-D [17] is an extension of CACTI that models the on-chip portion of the DRAM (Dynamic Random Access Memory).

In this paper we describe CACTI-IO, an extension to CACTI, illustrated in Figure 1.1. CACTI-IO allows the user to describe the configuration(s) of interest, including the capacity and organization of the memory dies, target bandwidth, and interconnect parameters. CACTI-IO includes analytical models for the interface power, including suitable lookup tables for some of the analog components in the PHY. It also includes voltage and timing uncertainty models that help relate parameters that affect power and timing. Voltage and timing budgets are traditionally used by interface designers to begin building components of the interface [1, 3, 31, 39] and budget the eye diagram between the DRAM, interconnect, and the controller as shown in Figure 1.2. The *Eye Mask* represents the portion of the eye budgeted for the *Rx* (receiver). The setup/hold slacks and noise margins represent the budgets for the interconnect and the Tx (transmitter).

Final optimization of the IO circuit, off-chip configuration and signaling parameters requires detailed design of circuits along with SPICE analysis, including detailed signal integrity and power integrity analyses; this can take months for a new design [3]. CACTI-IO is not a substitute for

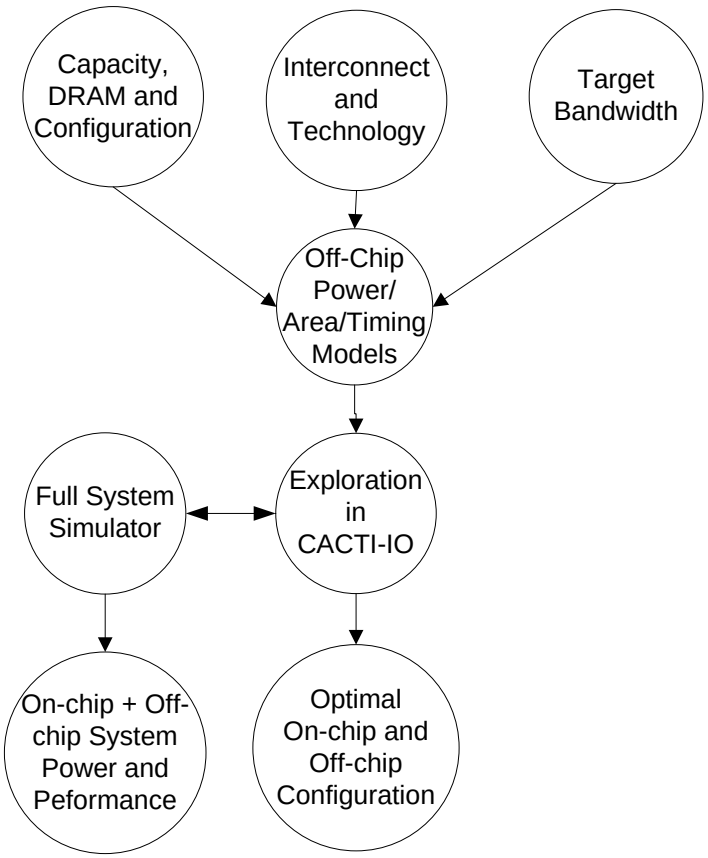


Figure 1.1: CACTI-IO: Off-chip modeling and exploration within CACTI.

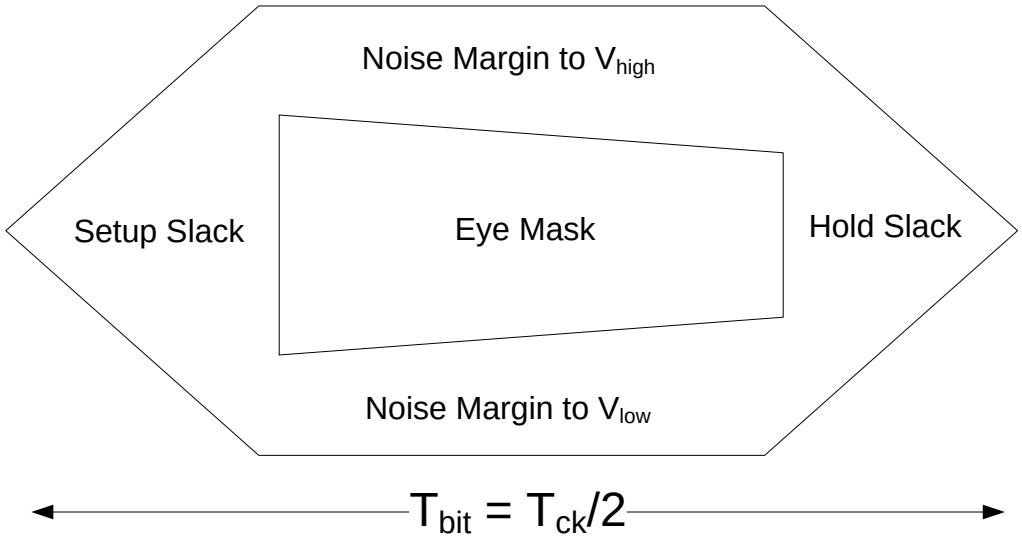


Figure 1.2: Memory interface eye diagram for voltage and noise budgets.

detailed analyses, but rather serves as a quick estimate for the system architect to get within 20% of the final power and performance numbers. This will enable the right tradeoffs between the large number of non-trivial IO and off-chip parameters on the one hand and system metrics on the other hand. Up-front identification of the off-chip design space at an architectural level is crucial for driving next-generation memory interface design.

The main objectives for the CACTI-IO tool are as follows.

(1) Obtain IO power numbers for different topologies and modes of operation that can be fed into a full-system simulator. The tradeoffs between performance, power and capacity in the memory subsystem are non-trivial [13, 17], but previous studies often do not explore alternatives for the memory interface to a standard DDR3 configuration. Furthermore, most simulators do not model the off-chip interface power and timing, and have no visibility into the details of the PHY and IO. CACTI-IO provides IO power numbers for Read, Write, Idle (only clock active) and Sleep modes that can easily be integrated into a system simulator. This enables architects to see the most significant on-chip and off-chip sources of power across modes.

(2) Enable co-optimization of off-chip and on-chip power and performance, especially for new off-chip topologies. Historically, off-chip parameters (i.e., signaling properties and circuit parameters) have been limited to standardized configurations including DIMMs, with operating voltage, frequency, data rates and IO parameters strictly governed by standards. A major drawback and design limiter – especially when operating at high frequencies – in this simplistic design context is the number of DIMMs that can be connected to a channel. This often limits memory capacity, creating a *memory-wall*. Recent large enterprise servers and multicore processors instead use one or more intermediate buffers to expand capacity and alleviate signal integrity issues. Such a design still adheres to DRAM standards but has more flexibility with respect to the interconnect architecture that connects memory and compute modules, including serial interfaces between the buffer and the CPU. While current and future memory system capacity and performance greatly depend on various IO choices, to date there is no systematic way to identify the optimal off-chip topology

that meets a specific design goal, including capacity and bandwidth. CACTI-IO provides a way for architects to systematically optimize IO choices in conjunction with the rest of the memory architecture. Below, we illustrate how CACTI-IO can help optimize a number of off-chip parameters – number of ranks (fanout on the data bus), memory data width, bus frequency, supply voltage, address bus fanout and bus width, – for given capacity and bandwidth requirements. CACTI-IO can also be used to evaluate the number of buffers needed in complex, large-capacity memory configurations, along with their associated overheads.
(3) Enable exploration of emerging memory technologies. With the advent of new interconnect and memory technologies, including 3-D TSS (through-silicon stacking) based interconnect being proposed for DRAM as well as new memory technologies such as MRAM (magnetic RAM) and PCRAM (phase-change RAM) [33], architects are exploring novel memory architectures involving special off-chip caches and write buffers to filter writes or reduce write overhead. Note that most emerging alternatives to DRAM suffer from high write energy or low write endurance. The use of additional buffers plays a critical role in such off-chip caches, and there is a need to explore the changing on-chip and off-chip design space. When designing new off-chip configurations, many new tradeoffs arise based on the choice of off-chip interconnect, termination type, number of fanouts, operating frequency and interface type (serial vs. parallel). CACTI-IO provides flexible baseline IO models that can be easily tailored to new technologies and used to explore tradeoffs at a system level.

The technical report is organized as follows.

- Chapter 2 describes the interface models in detail, including those for power, voltage margins, timing margins and area.
- Chapter 3 provides a brief user guide for CACTI-IO, including a list of new IO and off-chip parameters added. It also contains three standard configurations that are included in the CACTI-IO code. A DDR3 configuration, an LPDDR2 configuration and a Wide-IO configuration.
- Chapter 4 includes guidance on how the IO and off-chip models can be ported to a new technology.
- Chapter 5 validates CACTI-IO against SPICE simulations and measurements.

Chapter 2

IO, PHY and Interconnect Models

Complete details of the IO, PHY and interconnect models included in CACTI-IO are provided here. Power and timing models for interconnect and terminations have been well documented and validated over the years [1, 2, 5]. Our goal here is to show the framework of the baseline models, which can then be adapted and validated to any customized configuration needed, including new interconnect technologies.

Shown in Figure 2.1 is a typical memory interface datapath. This shows a PHY block that interfaces the memory controller to the IO circuit (Transmitter (Tx) and Receiver (Rx)). The Tx and Rx are connected through an off-chip interconnect channel.

2.1 Power Models

Power is calculated for four different modes: WRITE (peak activity during WRITE), READ (peak activity during READ) , Idle (no data activity, but clock is enabled and terminations are on), and Sleep (clock and terminations are disabled, in addition to no data activity). The mode of the off-chip interconnect can be chosen by setting the *iostate* input parameter to W (WRITE), R (READ), I (IDLE) or S (SLEEP).

CACTI-IO off-chip models include the following.

(1) Dynamic IO Power. The switching power at the load capacitances is described in Equation

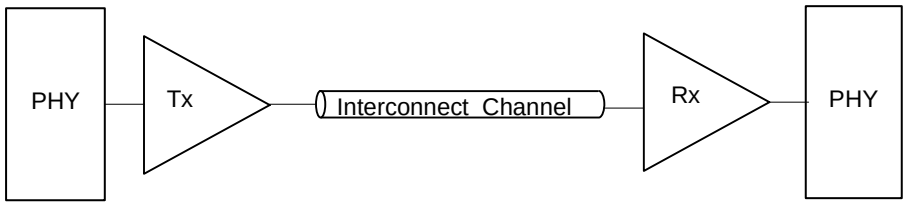


Figure 2.1: Typical Memory Interface Datapath.

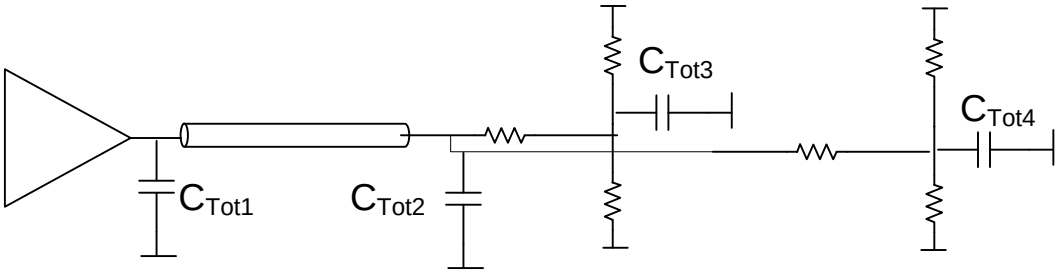


Figure 2.2: Capacitive Loads Off-chip.

(2.1), where N_{pins} is the number of signal pins; D_c is the duty cycle of activity; α is the activity factor for the signal switching (number of 0 to 1 transitions per clock period, i.e. $\alpha = 1$ for a clock signal); i denotes various nodes along the interconnect, with possibly different swings in a terminated or low-swing scheme; C_{Total_i} is the capacitance at node i as shown in the example in Figure 2.2; V_{sw_i} is the swing of the signal at node i; V_{dd} is the supply voltage; and f is the frequency of operation.

$$P_{dyn} = N_{pins} D_c \alpha (\sum_i C_{Total_i} V_{sw_i}) V_{dd} f \tag{2.1}$$

(2) Interconnect power. The power dissipated on the interconnect ($P_{dyn\_interconnect}$) is given by Equation (2.2). The energy/bit consumed on the interconnect ($E_{bit}^{interconnect}$) is described in Equation (2.3), where Z_0 is the characteristic impedance of the line, t_L is the flight time (time taken for the signal to traverse the line length) and t_b is the bit period. For large-capacity servers, generally $2t_L > t_b$ since the interconnect is long, while for mobile configurations, generally $2t_L < t_b$. For an FR-4 based interconnect used on printed circuit boards, t_L is approximately 180 ps/inch. The interconnect is generally modeled as a transmission line unlike an on-die RC network [2] when $t_L > t_r/3$, where t_r is the rise-time of the signal.

$$P_{dyn\_interconnect} = N_{pins} D_c \alpha E_{bit}^{interconnect} f \tag{2.2}$$

$$E_{bit}^{interconnect} = \begin{cases} \frac{t_L V_{sw} V_{dd}}{Z_0} & \text{if } 2t_L \leq t_b \\ \frac{t_b V_{sw} V_{dd}}{Z_0} & \text{if } 2t_L > t_b \end{cases} \tag{2.3}$$

(3) Termination Power. Terminations are used to improve signal integrity and achieve higher speeds, and the values depend on the interconnect length as well as the frequency or timing requirements. Terminations on the DQ (data) bus typically use an ODT (on-die termination) scheme, while those on the CA (command-address) bus use a fly-by termination scheme to the multiple loads. Figures 2.3 and 2.4 show the DDR3 DQ and CA termination schemes along with the static current consumed by them as used in [18].

The IO termination power is provided for various termination options, including unterminated (as used in LPDDR2 and Wide-IO), center-tap (as used in DDR3), VDDQ (as in DDR4) and differential terminations (as used in M-XDR). The voltage swing set by the terminations is fed into the dynamic power equation described above in Equation (2.1).

The termination power is then calculated for source and far-end terminations when the line is driven to 0 (V_{ol}) and driven to 1 (V_{oh}), and the average power is reported assuming that the probability of a 0 or 1 is equal during peak activity.

$$P_{term\_oh} = (V_{dd} - V_{TT})(V_{oh} - V_{TT})/R_{TT} \tag{2.4}$$
$$P_{term\_ol} = V_{TT}(V_{TT} - V_{ol})/R_{TT} \tag{2.5}$$
$$P_{avg} = (P_{term\_oh} + P_{term\_ol})/2 \tag{2.6}$$
$$P_{Totavg\_term} = \sum P_{avg} \tag{2.7}$$

(i) Unterminated. No termination power.

(ii) Center-tap termination, as in DDR3. The DQ WRITE, DQ READ and CA powers are described in Equations (2.8)-(2.10) respectively. R_{ON} is the driver impedance, R_{TT1} and R_{TT2} are the effective termination impedance of the used and unused rank respectively. $R_{||}$ is the effective impedance of both the ranks seen together. For the CA case, R_{TT} is the effective fly-by termination.

$$P_{DQ\_Term} = 0.25 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{TT1}} + \frac{1}{R_{TT2}} + \frac{1}{R_{ON} + R_{||}} \right) \tag{2.8}$$

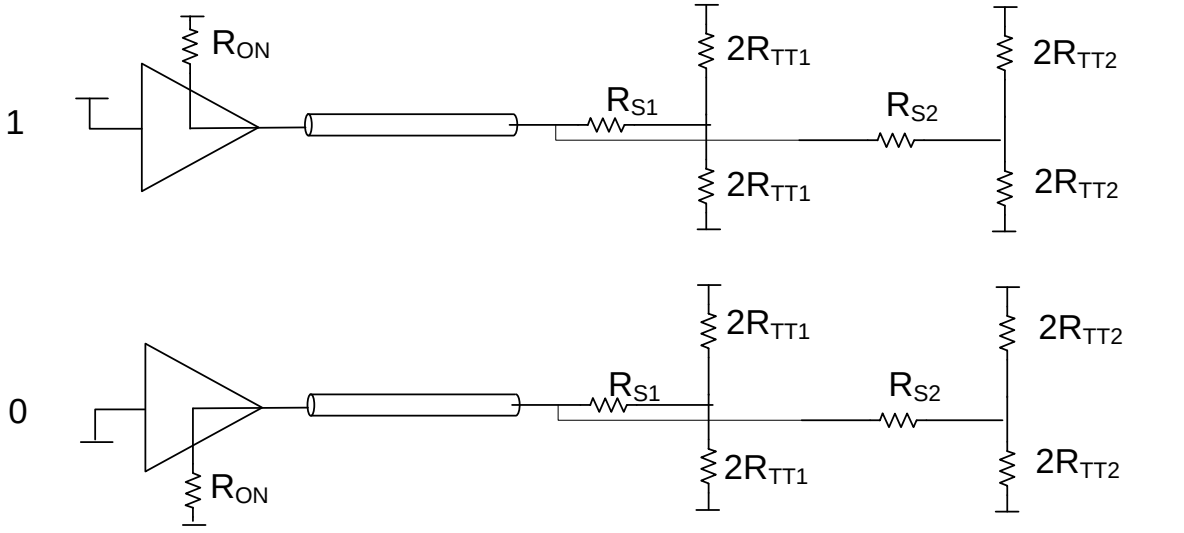
$$P_{DQ\_Term} = 0.25 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{TT1}} + \frac{1}{R_{TT2}} + \frac{1}{R_{ON} + R_{S1} + R_{||}^{read}} \right) \tag{2.9}$$

$$P_{CA\_Term} = 0.25 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{ON} + R_{TT}} \right) \tag{2.10}$$

(iii) Differential termination for low-swing differential interfaces. The power for a typical differential termination scheme is as follows.

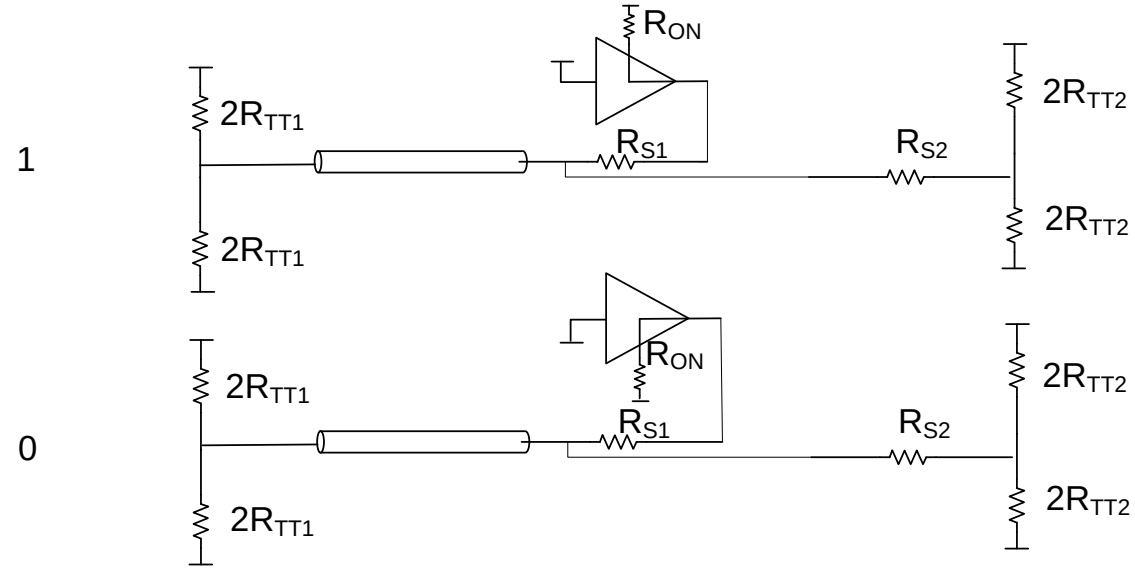
$$P_{diff\_term} = 2 \cdot V_{dd} V_{sw} / R_{TT} \tag{2.11}$$

The differential termination scheme is shown in Figure 2.5.
In some cases, differential low-swing transmitter circuits could use a small voltage regulated supply to generate a voltage-mode output [28]. In such a situation, the termination power would be one half of the value above in Equation (2.11).



$$I_{Totavg} = \frac{V_{dd}}{4(R_{TT1} \,||\, R_{TT2} \,||\, (R_{ON}+R_{||}))} \qquad R_{||}=(R_{TT1}+R_{S1})||(R_{TT2}+R_{S2})$$

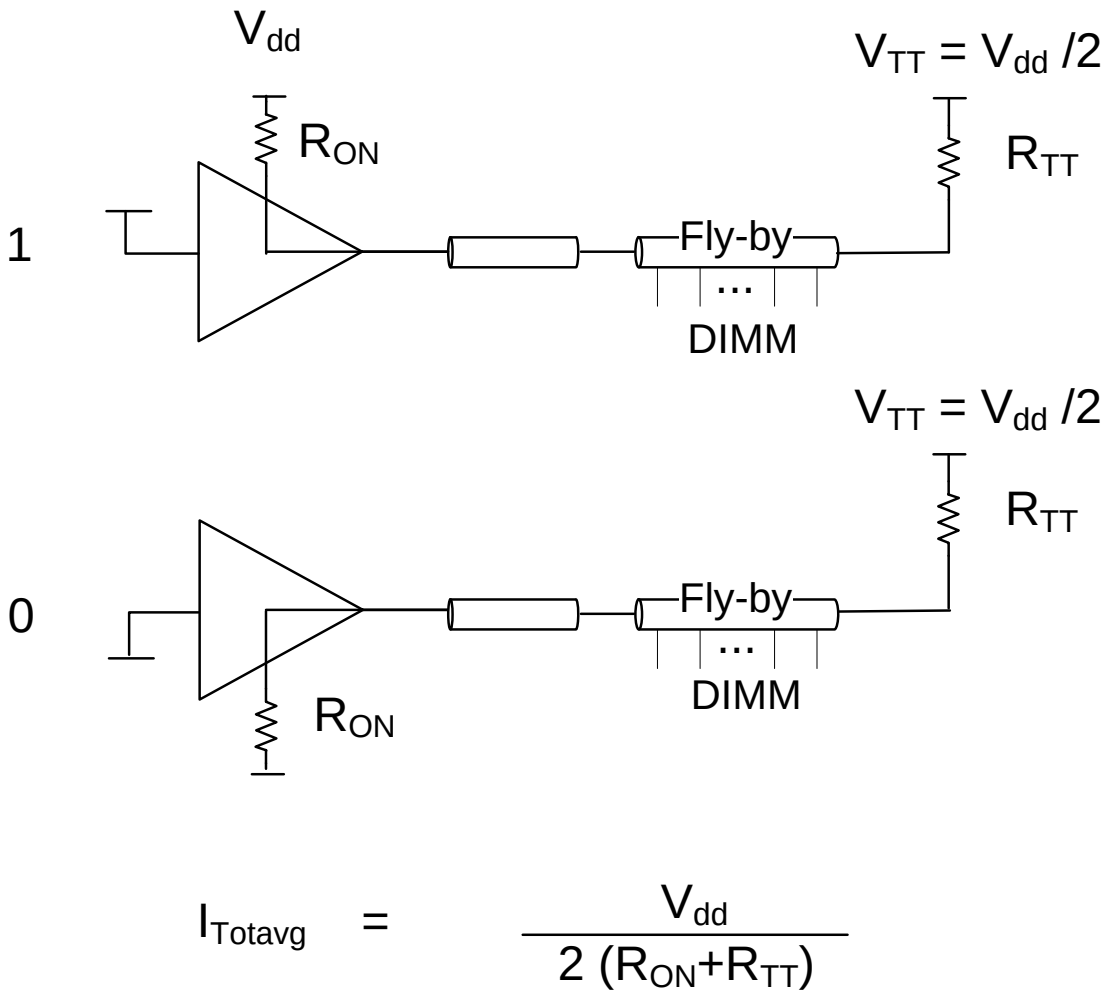
(a) WRITE



$$I_{Totavg} = \frac{V_{dd}}{4(R_{TT1} \,||\, R_{TT2} \,||\, (R_{ON}+R_{S1}+R_{||}))} \qquad R_{||}=(R_{TT1})||(R_{TT2}+R_{S2})$$

(b) READ

Figure 2.3: DDR3 DQ dual rank termination.



$$I_{Totavg} = \frac{V_{dd}}{2\,(R_{ON}+R_{TT})}$$

Figure 2.4: DDR3 CA termination.

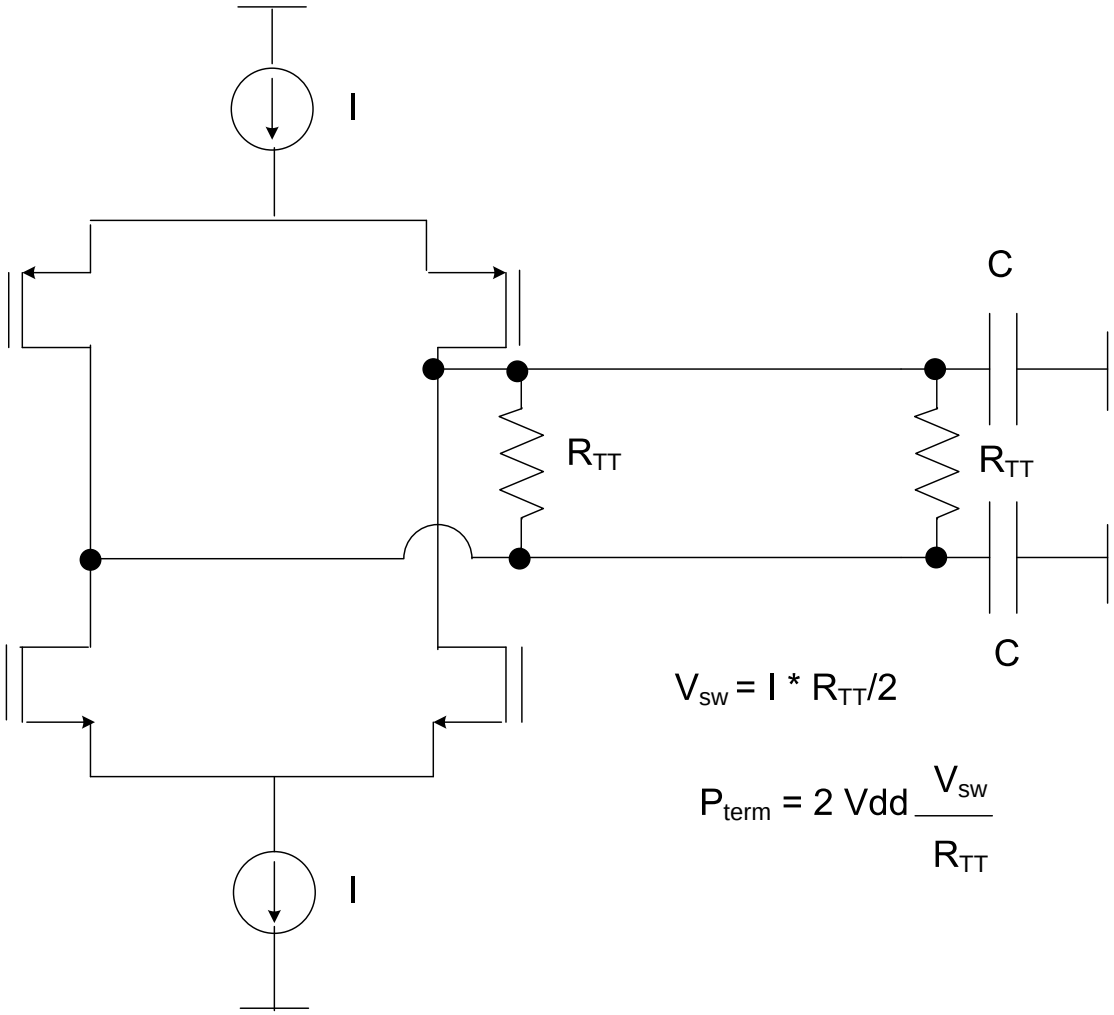


Figure 2.5: Differential Termination.

CACTI-IO calculates the voltage swing as follows, which is also used to feed into the dynamic power equation described in Equation (2.1). The swing is calculated at the two loads and on the line as shown in Figure 2.3 for both WRITE and READ modes. The terminations used are also shown in Figure 2.3, where R_{ON} is the driver impedance, R_{TT1} and R_{TT2} are the effective termination impedances (implemented as on-die terminations) for the two ranks in a dual-rank configuration and R_{S1} and R_{S2} are the series resistors used for better signal integrity.

WRITE:

$$V_{sw-line} = \frac{V_{dd} \cdot R_{||}}{(R_{ON} + R_{||})} \tag{2.12}$$

$$V_{sw-load1} = \frac{V_{dd} \cdot R_{TT1}(R_{S2} + R_{TT2})}{(R_{S1} + R_{TT1} + R_{S2} + R_{TT2})(R_{ON} + R_{||})} \tag{2.13}$$

$$V_{sw-load2} = \frac{V_{dd} \cdot R_{TT2}(R_{S1} + R_{TT1})}{(R_{S1} + R_{TT1} + R_{S2} + R_{TT2})(R_{ON} + R_{||})} \tag{2.14}$$

$$where\ R_{||} = (R_{TT1} + R_{S1})||(R_{TT2} + R_{S2}) \tag{2.15}$$

READ:

$$V_{sw-line} = \frac{V_{dd} \cdot R_{||}^{read}}{(R_{ON} + R_{S1} + R_{||}^{read})} \tag{2.16}$$

$$V_{sw-load1} = \frac{V_{dd} \cdot R_{TT1}(R_{S2} + R_{TT2})}{(R_{TT1} + R_{S2} + R_{TT2})(R_{ON} + R_{S1} + R_{||}^{read})} \tag{2.17}$$

$$V_{sw-load2} = \frac{V_{dd} \cdot R_{TT2} R_{TT1}}{(R_{TT1} + R_{S2} + R_{TT2})(R_{ON} + R_{S1} + R_{||}^{read})} \tag{2.18}$$

$$where\ R_{||}^{read} = (R_{TT1})||(R_{TT2} + R_{S2}) \tag{2.19}$$

For a VDDQ-termination for DDR4 and LPDDR3, the power equation is described here. The DDR4 and LPDDR3 specifications are still under development [21], but are expected to use a VDDQ termination scheme as shown in Figure 2.6. This is similar to other POD (pseudo-open-drain) schemes used by JEDEC [21]. The equations for the voltage swing for such a termination scheme are the same as for DDR3 above in Equations (2.12)-(2.19). However, the signal is referenced to VDDQ rather than VDDQ/2, and the power equation is described below in Equation (2.20), where $R_{||}$ is calculated for WRITE and READ modes similar to the DDR3 DQ case as shown in Equations (2.15) and (2.19). LPDDR3 supports the unterminated, full-swing interface as well.

$$P_{DQ\_Term} = 0.5 \cdot V_{dd}^2 \cdot \left(\frac{1}{R_{ON} + R_{||}} \right) \tag{2.20}$$

$$\tag{2.21}$$

The CA termination would be similar to the DDR3 fly-by scheme.

(4) PHY Power. The PHY includes analog and digital components used to retime the IO signals on the interface. A wide range of implementations exist for the PHY [14, 15, 16, 23, 24, 25], that vary in power and are fine-tuned to specific design requirements.

Shown in Figure 2.7 is a typical PHY datapath. This shows the components in a PHY block that interfaces the memory controller to the IO circuit (Transmitter (Tx) and Receiver (Rx)).

The building blocks listed include blocks that typically retime a source-synchronous interface using a forwarded clock scheme [1]. The *Datapath* refers to the data transmit path until the input to the IO Tx and the data receive path after the IO Rx. The *Phase Rotator* is a delay element used to generate a T/4 delay to center-align the data-strobe (DQS) with respect to the data (DQ) pins. It could be a DLL or any other delay element that meets the requirements on the edge placement error (T_{error}). The *Clock Tree* is the local clock-tree within the PHY that distributes the clock to all the bit lanes. The *Rx* refers to the IO receiver, which typically consumes some static power for DDR3 SSTL (stub-series terminated logic), owing to a pseudo-differential V_{ref} based receiver first stage. Some PHY implementations have a *Duty Cycle Correction* that corrects duty-cycle distortion, *Deskewing* that reduces static skew offsets, *Write/Read Leveling* that lines up the various data byte lanes with the fly-by clock and a *PLL* dedicated for the memory interface. Specific implementations could have other blocks not listed here, but the framework supports easy definition of dynamic and static active and idle power for each of the building blocks. Each building block in the PHY has an idle and sleep state, similar to the IO. While these are not described here, they are similar to the lookup tables shown for static power, but with suitable idle and sleep numbers included. These blocks often have wakeup times when entering the active mode, and these can be modeled within CACTI as well. We propose to extend our framework in the future to cover wakeup times between sleep, idle and active modes.

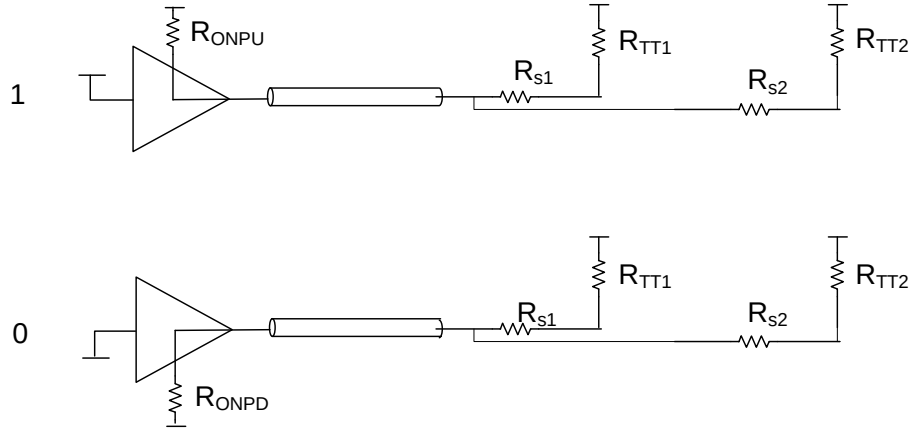
The static skew (T_{skew_setup}, T_{skew_hold}) on the interface and the duty-cycle distortion (T_{DCD}) can be reduced if the PHY implements a deskewing scheme and a duty-cycle corrector. In some cases, the DLL on the DRAM could clean up the duty-cycle distortion, which helps improve the READ margins.

Currently, the user can change the inputs for the PHY power based on a specific implementation. Tables 2.1 and 2.2 respectively show the active dynamic power per bit and static power for the entire PHY of an example PHY implementation for a x128 3-D configuration based on building blocks in the design. The building blocks are representative of typical PHY components [14, 15, 16, 23, 24, 25]. Table 2.3 shows the dynamic and static power for an example DDR3-1600 PHY. At lower data rates, certain components are not required, indicated by N/A in Tables 2.1 and 2.2.

The above 4 components of the IO and PHY power are combined in the following way depending on which mode the interface is in.

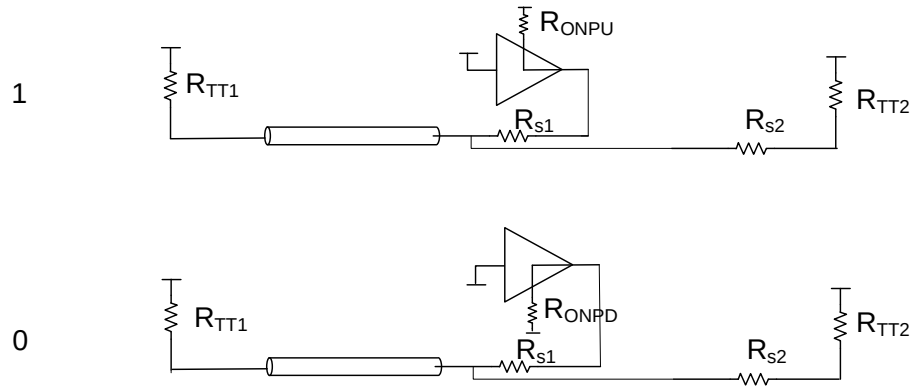
WRITE or READ:

$$P_{Total\_Active} = P_{dyn} + P_{dyn\_interconnect} + P_{term} + P_{static/bias} \tag{2.22}$$



$$I_{Totavg} = \frac{V_{dd}}{2(R_{ONPD}+R_{||})} \qquad R_{||}=(R_{TT1}+R_{S1})||(R_{TT2}+R_{S2})$$

(a) WRITE



$$I_{Totavg} = \frac{V_{dd}}{2(R_{ONPD}+R_{S1}+R_{||})} \qquad R_{||}=(R_{TT1})||(R_{TT2}+R_{S2})$$

(b) READ

Figure 2.6: DDR4 DQ Dual Rank Termination.

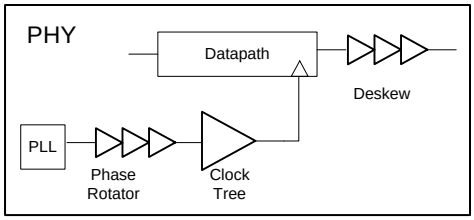


Figure 2.7: Typical PHY Datapath.

Table 2.1: PHY Active Dynamic Power /bit for 3-D configurations.

Building Block	**Dynamic Power (mW/Gbps)**		
	500 Mbps	**1 Gbps**	**2 Gbps**
Datapath	0.1	0.2	0.5
Phase Rotator	N/A	0.1	0.2
Clock Tree	0.1	0.2	0.4
Duty Cycle Correction	N/A	N/A	0.05
Deskewing	N/A	N/A	0.05
PLL	N/A	N/A	0.05

Table 2.2: PHY Static Power for a x128 3-D configuration.

Building Block	**Static Power (mW)**		
	500 Mbps	**1 Gbps**	**2 Gbps**
Phase Rotator	N/A	1	10
PLL	N/A	N/A	10

Table 2.3: PHY Dynamic /bit and Static Power for a x64 DDR3-1600.

Building Block	**Dynamic Power (mW/Gbps)**	**Static Power (mW)**
Datapath	0.5	0
Phase Rotator	0.2	10
Clock Tree	0.8	0
Rx	0.2	20
Duty Cycle Correction	0.05	0
Deskewing	0.05	0
Write/Read Leveling	0.05	0
PLL	0.1	10

IDLE:

$$P_{Total\_Idle} = P_{term} + P_{static/bias} + P_{dyn\_clock} \tag{2.23}$$

SLEEP:

$$P_{Sleep} = P_{leakage} \tag{2.24}$$

The duty cycle spent in each mode can be specified using the *duty cycle* input parameter.

2.2 Voltage and Timing Margins

The minimum achievable clock period T_{ck} depends on the voltage and timing budgets (i.e., eye diagram and/or BER (bit error rate) compliance). Traditionally, the memory interface budgets have been based on the worst-case analysis approach shown in Figure 1.2, where the budgets are divided between the DRAM, the interconnect and the controller chip or SOC. With increasing speeds there is a need for a statistical analysis approach similar to serial links [32, 36] during detailed design analysis. However, for architectural exploration and relative tradeoffs, we continue to use worst-case budgets in our initial framework, which is typically within 10% of a statistical approach [32]. The user can account for this optimism or pessimism based on measurements or prior correlation between the two approaches by using a correlation error term. This correlation error can also help address different BER requirements for server DIMM modules that include error correction (ECC) schemes [3, 26, 29].

(1) Timing budgets. The key interface timing equations are based on DRAM AC timing parameters in the JEDEC specification [19, 20]. There are nuances to the system timing based on the controller design and clocking architecture, but most rely on measuring setup and hold slacks to ensure positive margins.

It is interesting to note that while the DQ bus is DDR in almost all DRAMs today, the CA bus is mostly SDR (single data rate), except for LPDDR2 and LPDDR3 where the CA bus is DDR [19, 20]. In addition, the CA bus provides an option for 2T (two clock-cycles) and 3T (three clock-cycles) timing to relax the requirements when capacitive loading is high. This is done since the CA bus is typically shared across all memories in the DIMM.

The jitter on the interface is the true limiter of the timing budget, and optimizing the interface for low jitter is the key challenge. The common sources of jitter include Tx jitter, ISI (inter-symbol interference), crosstalk, SSO (simultaneously switching outputs), supply noise and Rx jitter [3].

Jitter can be estimated from various deterministic (DJ_i) and random (RJ_i) sources as follows [3]:

$$T_{jitter} = \sum_i DJ_i + \sqrt{\sum_i RJ_i^2} \tag{2.25}$$

$$T_{jitter}(\mathbb{F}_0) = T_{jitter\_avg} + \sum_i (T_{jitter}(F_i = F_{i0}) - T_{jitter\_avg}) \tag{2.26}$$

Here, factor F_i is a parameter that affects T_{jitter} [3]. These include parameters like R_{ON}, R_{TT} and C_{Total_i}. For tradeoffs between power and performance, only the factors that affect power need to be studied. $\mathbb{F}_0$ is the value of a set of factors $F_i = F_{i0}$ for which we calculate the jitter, $T_{jitter}(\mathbb{F}_0)$, as an estimate assuming there is no interaction between the factors F_i [3]. This is done efficiently by running a Design of Experiments (DOE) for a set of orthogonal array experiments as defined by the Taguchi method [3, 22]. T_{jitter_avg} represents the average jitter from all the experiments in the orthogonal array, while $T_{jitter}(F_i = F_{i0})$ represents the average jitter from all experiments where $F_i = F_{i0}$. For cases where F_{i0} is not part of the orthogonal array, a piecewise linear approximation is employed. Chapter 5 shows an example of such an array for a typical DDR3 channel.

The key interface timing equations are described below. $T_{jitter\_hold/setup}$ are the half-cycle jitter for hold and setup between DQ (data) and DQS (data-strobe), and T_{jitter} is the full-cycle jitter. Depending on the implementation, either T_{jitter_setup} or T_{jitter_hold} may be quite small as the DQ and DQS track each other from a common source clock in a forwarded clock scheme, but the other edge of the eye would incur the half-cycle jitter term. T_{error} is the edge placement error of the T/4 delay element, T_{skew} is the static skew in the interface, and $T_{rise/fall(V_{ref}->V_{IH/IL})}$ is the rise/fall time at the Rx input from the V_{ref} value (the switching reference voltage) to the $V_{IH/IL}$ value (the Rx thresholds). T_{SOC_hold} and T_{SOC_hold} are the hold and setup times at the SOC inputs during READ. $T_{DCD-SOC}$ is the DCD of the SOC clock output. The remaining parameters in the equations below are JEDEC DRAM parameters [19, 20].

(i) DQ-DQS WRITE:

$$\left(\frac{T_{ck}}{4}\right) - T_{DCD-SOC} - T_{error} - T_{jitter\_hold} - T_{skew\_hold} > T_{DHbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.27}$$

$$\left(\frac{T_{ck}}{4}\right) - T_{error} - T_{jitter\_setup} - T_{skew\_setup} > T_{DSbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.28}$$

(ii) DQ-DQS READ:

$$T_{QSH/QSL} - T_{DCD-SOC} - T_{error} - T_{jitter\_hold} - T_{skew\_hold} - T_{QHS} > T_{SOC\_hold} \tag{2.29}$$

$$\left(\frac{T_{ck}}{4}\right) - T_{error} - T_{jitter\_setup} - T_{skew\_setup} - T_{DQSQ} > T_{SOC\_setup} \tag{2.30}$$

(iii) CA-CLK (DDR for LPDDR2/3):

$$\left(\frac{T_{ck}}{4}\right) - T_{DCD} - T_{error} - T_{jitter\_hold} - T_{skew\_hold} > T_{IHbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.31}$$

$$\left(\frac{T_{ck}}{4}\right) - T_{error} - T_{jitter\_setup} - T_{skew\_setup} > T_{ISbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.32}$$

For DDR3 the CA interface is SDR, and the above timing is relaxed to a half-cycle as opposed to a quarter-cycle, as follows:

$$\left(\frac{T_{ck}}{2}\right) - T_{jitter\_hold} - T_{skew\_hold} > T_{IHbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.33}$$

The CA timing can be further relaxed if the 2T or 3T timing option is enabled in the DDR3 DRAM.

2T:

$$T_{ck} - T_{jitter\_hold} - T_{skew\_hold} > T_{IHbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.34}$$

3T:

$$\left(\frac{3 \cdot T_{ck}}{2}\right) - T_{jitter\_hold} - T_{skew\_hold} > T_{IHbase} + T_{rise/fall(V_{ref}->V_{IH/IL})} \tag{2.35}$$

The setup equations are similarly relaxed.

(iv) CLK and DQS:

$$\left(\frac{T_{ck}}{2}\right) - T_{DCD} - T_{jitter\_setup/hold} > T_{CH/CL\_abs} \tag{2.36}$$

$$T_{jitter} < T_{JIT} \tag{2.37}$$

$$T_{jitter\_hold} + T_{skew\_hold} + T_{DCD} < \left(\frac{T_{ck}}{2}\right) - T_{DSH} \tag{2.38}$$

$$T_{jitter\_setup} + T_{skew\_setup} < \left(\frac{T_{ck}}{2}\right) - T_{DSS} \tag{2.39}$$

(2) Voltage Budgets. A voltage budget can be developed for voltage margins as follows [1], which once again is based on a worst-case analysis, where V_N is the voltage noise, K_N is the proportionality coefficient for the proportional noise sources (that are proportional to the signal swing V_{sw}), V_{NI} is the noise due to independent noise sources and V_M is the voltage margin. Crosstalk, ISI and SSO are typical proportional noise sources [1], while the *Rx*-offset, sensitivity and independent supply noise are typical independent noise sources.

$$V_N = K_N \cdot V_{sw} + V_{NI} \tag{2.40}$$

$$K_N = K_{xtalk} + K_{ISI} + K_{SSO} \tag{2.41}$$

$$V_{NI} = V_{Rx-offset} + V_{Rx-sens} + V_{supply} \tag{2.42}$$

$$V_M = \frac{V_{sw}}{2} - V_N \tag{2.43}$$

A DOE analysis for the voltage noise coefficient, K_N, can be performed in a similar manner as described above for T_{jitter}.

2.3 Area Models

The area of the IO is modeled as shown below in Equation (2.44), where N_{IO} is the number of signals, f is the frequency, R_{ON} and R_{TT1} are the impedance of the IO driver and the on-die termination circuit respectively as shown in Figure 2.3, and A_0, k_0, k_1, k_2 and k_3 are constants for a given DRAM technology and configuration. The user can provide the design-specific fitting constants.

The area of the last stage of the driver is proportional to $1/R_{ON}$ or the drive current, and the fanout in the IO for the predriver stages is proportional to f, the frequency of the interface. In the event that the on-die termination (R_{TT1}) is smaller than R_{ON}, then the driver size is determined by $1/R_{TT1}$.

The predriver stages still drive only the legs of the output driver that provide the R_{ON} drive impedance, so the area of the predriver stages is a scaled down factor of $1/R_{ON}$.

$k_1 * f$, $k_2 * f^2$ and $k_3 * f^3$ are the scaling factors for the first, second and third stages of the predriver. Chapter 4 shows how these parameters scale with technology.

The fanout is proportional to frequency to reflect the proportional edge rates needed based on frequency.

A_0 is the area of the rest of the IO, which is assumed to be a smaller fixed portion.

$$\begin{aligned} Area_{IO} = N_{IO} \cdot \left(A_0 + \frac{k_0}{min(R_{ON}, 2 \cdot R_{TT1})} \right) + \\ N_{IO} \cdot \left(\frac{1}{R_{ON}} \right) \cdot (k_1 * f + k_2 * f^2 + k_3 * f^3) \end{aligned} \tag{2.44}$$

Chapter 3

Usage

Prior versions of CACTI use a configuration file (cache.cfg) to enable the user to describe the cache parameters in detail. CACTI-IO uses the configuration file to further enable the user to provide the IO and off-chip parameters as well. The extio.cc file contains the IO and off-chip models and can be modified if these models need to be customized.
CACTI-IO can be invoked in a similar manner to prior CACTI versions: >cacti -infile cache.cfg
Described here are the input parameters available for the user to vary as well as some in-built technology and configuration parameters that describe a few standard configurations of interest. More details on tool usage and each of these parameters is provided in the default cache.cfg file that is provided with the distribution.

3.1 Configuration Parameters

The basic off-chip configuration parameters can be entered in the cache.cfg file. They are described below.

Memory Type (D=DDR3, L=LPDDR2, W=WideIO)
-dram_type ”D”
dram_type allows the user to pick between three standard configurations described in the extio.cc file. Any choice apart from the three specified above defaults to the DDR3 configuration settings.

Memory State (R=Read, W=Write, I=Idle or S=Sleep)
-iostate ”W”
iostate picks the state the memory interface is in - READ, WRITE, IDLE or SLEEP.

ECC (Y=Yes, N=No)
-dram_ecc ”N”
dram_ecc specifies whether ECC (Error Correction Coding) is enabled on the interface. ECC usually means extra data (DQ) bits reserved for ECC functionality. The code defaults to an extra byte reserved for ECC if the dram_ecc is set to ”Y”.

Address bus timing (DDR=0.5, SDR=1.0, 2T=2.0, 3T=3.0)

-addr_timing
addr_timing specifies the timing on the address bus. The options available are DDR timing similar to DQ, SDR timing or 2T and 3T timing, which provide relaxed timing options on the CA bus owing to larger loads [19]. LPDDR2 and LPDDR3 require DDR timing as described in Equations (2.31, 2.32), while DDR3 allows for SDR, 2T and 3T timing as described in Equations (2.33, 2.34, 2.35).

Bandwidth (Gbytes per second, this is the effective bandwidth)
-bus_bw
bus_bw represents the effective bandwidth utilized on the memory bus, as opposed to the maximum bandwidth possible on the bus.

$$bus_bw = \left(\frac{2 \cdot N_{bus\_width}}{T_{ck}} \right) \tag{3.1}$$

Memory Density (Gbit per memory/DRAM die)
-mem_density
mem_density represents the memory density for one die of the DRAM. It is based on technology and availability.

IO frequency (MHz)
-bus_freq
bus_freq is the frequency of the off-chip memory bus. T_{ck} is derived as the inverse of bus_freq.

Duty Cycle
-duty_cycle
duty_cycle is fraction of time in the Memory State defined above by iostate.

Activity factor for Data
-activity_dq
activity_dq is the number of 0 to 1 transitions per cycle for the DQ signals (for DDR, need to account for the higher activity in this parameter. E.g. max. activity factor for DDR is 1.0, for SDR is 0.5). The activity_dq is used to calculate the dynamic power as α in Equation (2.1).

Activity factor for Control/Address
-activity_ca
activity_ca is the number of 0 to 1 transitions per cycle for the CA signals (for DDR, need to account for the higher activity in this parameter. E.g. max. activity factor for DDR is 1.0, for SDR is 0.5). The activity_ca is used to calculate the dynamic power as α in Equation (2.1).

Number of DQ pins
-num_dq
num_dq is the width of the memory bus per channel. Typically x64 widths are used for a DDR3 channel.

Number of DQS pins
-num_dqs
num_dqs is the number of DQS (data-strobe) pins. While most DRAMs have 1 DQS differential pair per byte of DQ, this could change based on the DRAM type and the width of the DRAM E.g. a x4 DRAM will have a DQS pair for every 4 DQ signals). Each differential pair accounts for 2 DQS signals for num_dqs.

Number of CA pins
-num_ca
num_ca is the number of command and address signals in the DRAM interface. This could vary depending on the addressable space in the design and the number of ranks.

Number of CLK pins
-num_clk
num_clk is the number of clocks. Typically there is 1 differential pair for a channel, but in some cases extra pairs could be used to improve signal integrity.

Number of Physical Ranks
-num_mem_dq
num_mem_dq is the number of loads on DQ, i.e. the number of ranks. Multiple chip selects can choose between parallel ranks connected to the CPU. Multiple ranks increase loading on all pins, except CS and CKE pins, which are unique per rank.

Width of the Memory Data Bus
-mem_data_width
mem_data_width is the width of the DRAM. x4 or x8 or x16 or x32 memories are popular. E.g. When x8 or x16 memories are used to connect to a x32 or x64 bus, we have multiple memories making up a single word. This increases loading on CA and CLK, not on DQ/DQS.

Number of loads on CA and CLK
Besides the above parameters, the number of loads on the CA bus and the CLK pin are inferred from the above parameters as follows.

num_mem_ca = num_mem_dq * (num_dq/mem_data_width)

num_mem_clk = num_mem_dq * (num_dq/mem_data_width)/(num_clk/2)

Configuration parameters chosen by dram_type

extio.cc contains three standard configurations described below that are chosen based on the value of dram_type (D=DDR3, L=LPDDR2, W=Wide-IO). The parameters defined in these standard configurations are described here.
vdd_io, the IO supply voltage (V), which is V_{dd} used in all the power equations.
v_sw_clk, the Voltage swing on CLK/CLKB (V) (swing on the CLK pin if it is differentially terminated)

The loading capacitances used in Equation (2.1) are as follows.
c_int, the Internal IO loading (pF) (loading within the IO, due to predriver nets)
c_tx, the IO TX self-load including package (pF) (loading at the CPU TX pin)

c_data, the Device loading per memory data pin (pF) (DRAM device load for DQ per die)
c_addr, the Device loading per memory address pin (pF) (DRAM device load for CA per die)

The bias and leakage currents are as follows.
i_bias, the Bias current (mA) (includes bias current for the whole memory bus due to RX Vref based receivers)
i_leak, the Active leakage current per pin (nA)

The leakage and bias power are calculated from these as shown below.

$$P_{leakage} = I_{leak} \cdot V_{dd} \tag{3.2}$$

$$P_{bias} = I_{bias} \cdot V_{dd} \tag{3.3}$$

The IO Area coefficients used in Equation (2.44) are as follows
ioarea_c is A_0, ioarea_k0 is k_0, ioarea_k1 is k_1, ioarea_k2 is k_2 and ioarea_k3 is k_3.

The timing parameters (t_ds, t_is, t_dh, t_ih, t_dcd_soc, t_dcd_dram, t_error_soc, t_skew_setup, t_skew_hold, t_dqsq, t_qhs, t_soc_setup, t_soc_hold, t_jitter_setup, t_jitter_hold, t_jitter_addr_setup, t_jitter_addr_hold) are as described in Equations (2.27) - (2.38).

The PHY power coefficients (phy_datapath_s, phy_phase_rotator_s, phy_clock_tree_s, phy_rx_s, phy_dcc_s, phy_deskew_s, phy_leveling_s, phy_pll_s, phy_datapath_d, phy_phase_rotator_d, phy_clock_tree_d, phy_rx_d, phy_dcc_d, phy_deskew_d, phy_leveling_d, phy_pll_d) are shown in Table 2.3.

The termination resistors for WRITE, READ and CA (rtt1_dq_read, rtt2_dq_read, rtt1_dq_write, rtt2_dq_write, rtt_ca, rs1_dq, rs2_dq, r_stub_ca, r_on) are used in Equations (2.8) - (2.19).

The flight time for DQ (t_flight) and CA (t_flight_ca) are used to calculate the interconnect power in Equation (2.3).

3.2 The Standard Configurations

Included in CACTI-IO are three default configurations which can be chosen by setting the dram_type variable as described in Chapter 3 above. dram_type allows the user to pick between three standard configurations (D=DDR3, L=LPDDR2, W=WideIO) described in the extio.cc file. Any choice apart from the three specified above defaults to the DDR3 configuration settings. The table below summarizes the parameter values for the three standard configurations provided.

Table 3.1: Standard Configurations.

Parameter	**Configuration**		
	DDR3	**LPDDR2**	**Wide-IO**
vdd_io (V)	1.5	1.2	1.2
v_sw_clk (V)	0.7	1.0	1.2
c_int (pF)	1.5	2.0	0.5
c_tx (pF)	1.5	1.0	0.5
c_data (pF)	1.5	2.5	0.5
c_addr (pF)	0.75	1.5	0.4
i_bias (mA)	10	2	0
i_leak (nA)	3000	1000	500
t_ds (ps)	150	250	300
r_on (Ω)	34	50	75
rtt1_dq (Ω)	60	100000	100000
rtt_ca (Ω)	50	240	100000
t_flight (ns)	1.0	1.0	0.05
t_flight_ca (ns)	2.0	2.0	0.05

Chapter 4

Portability

The models described in Chapter 2 above are dependent on on-die as well as off-chip technology. As with prior CACTI versions, the IO and off-chip parameters that scale with process technology are taken from ITRS [35]. The underlying assumption is that the DRAM technology scales to meet the speed bin that it supports [21], since if DRAM technology is scaled, the speed bin that the IO parameters belong to are suitably scaled as well, including load capacitances (DRAM DQ pin capacitance (C_{DQ}), DRAM CA pin capacitance (C_{CA})) and AC timing parameters in Equations (2.27) - (2.39). LPDDRx use different technologies compared to DDRx to save leakage power, so their capacitances and timing parameters are different from a DDRx memory of the same speed bin. Voltage also scales with DRAM technology, typically when a DRAM standard changes, e.g. DDR2 used 1.8V IO supply voltage, while DDR3 uses 1.5V IO supply voltage [21]. Sometimes a lowered voltage specification is released as an addendum to a standard, e.g. DDR3-L [21]. Shown below in Table 4.1 are a subset of DDR3 DRAM parameters based on the speed bin.

If the user is interested in studying the impact of technology on a future memory standard, or a speed bin that is yet undefined, the timing parameters can be assumed to scale down linearly with frequency to the first order. The load capacitances can be assumed to remain above 1pF for DQ and 0.75pF for CA.

Table 4.1: Technology Scaling for DDR3.

Parameter	**Data rate (Mb/s)**		
	800	**1066**	**1600**
vdd_io (V)	1.5	1.5	1.5
c_data_max (pF)	3.0	3.0	2.3
c_addr_max (pF)	1.5	1.5	1.3
t_ds_base (ps)	75	25	10
t_dh_base (ps)	150	100	45
t_dqsq (ps)	200	150	100
t_qhs (ps)	300	225	150

The SOC PHY power and timing parameters scale with the technology node of the SOC, but are far more sensitive to the circuit architecture and analog components used to implement the design. It is hard to provide simplistic scaling trends for these parameters. For a given design and architecture, it would be possible to provide scaling power and timing for different technology

nodes, but as speeds increase, the design and architecture for the PHY and IO are optimized and/or redesigned for the higher speed. Various design-specific trends for power and timing scaling with technology suggest around 20% scaling of analog power from one technology node to the next, or one speed bin to the next [15].

The area of the IO directly scales with technology, but often only with the thick-oxide device used for the IO circuits. The scaling of the thick-oxide device typically does not keep pace with the core thin-oxide device owing to have to meet supply voltages for external standards and reliability concerns associated with it. The constants k_0, k_1, k_2 and k_3 scale inversely with $I_{dsat/um}$ of the thick-oxide device.

Besides the parameters that scale with technology, the topology impacts the models for timing and voltage noise. A suitable DOE is required to fit the jitter and voltage noise coefficients for a given topology that defines the number of loads and interconnect length. When defining a topology other than the three standard configurations, a DOE analysis (as shown in Chapter 5) needs to be performed to be able to port the timing models for the channel.

Chapter 5

Validation

Here we mainly focus on validating the new analytical IO and off-chip models added in CACTI-IO. Models that are based on a lookup table, including the PHY power numbers, are valid by construction as the user can provide valid inputs. We first validate the power models for each DQ and CA bit line. On average, the analytical power models are verified to be within 5% of SPICE results.

Figures 5.1 and 5.2 show SPICE vs. CACTI-IO for the termination power and total IO power of a single lane of DQ DDR3. Figure 5.1 indicates that the worst error between SPICE and CACTI-IO is less than 1% across different R_{TT1} values (R_{ON}= 34 Ω for these cases). The total IO power shown in Figure 5.2 for three different combinations of C_{DRAM}, R_{TT1} and T_{flight} shows a worst error of less than 14%.

Figures 5.3 and 5.4 show SPICE vs. model for the termination power and total IO power of a single lane of CA DDR3 using a fly-by termination scheme. Figure 5.3 shows the termination power for different R_{TT} values (the fly-by termination shown in Figure 2.4), while Figure 5.4 shows the total IO power for different numbers of loads or fly-by segments. Once again, the errors are similar to the DQ cases above, with the termination power within 1% and the total IO power within 15%.

Figures 5.5 shows SPICE vs. model for the switching power (dynamic IO and interconnect power) for DQ LPDDR2, where no terminations are used. In this scenario, the model is within 2% of the SPICE simulation.

To validate the power model for the entire interface, we compare it against measurements. Shown in Figure 5.6 is measured vs. model power for LPDDR2 WRITE obtained from a typical memory interface configuration for a 32-wide bus using a x32 LPDDR2 dual-rank DRAM. As can be seen, the model is within 5% of the measurement at the higher bandwidths. At lower bandwidths, power saving features make it harder to model the power as accurately since the duty cycle between the READ/WRITE/IDLE/SLEEP modes is harder to decipher. Here the error is within 15%.

Shown in Figure 5.7 are the results of an example DOE analysis on a sample channel for T_{jitter}. The input factors (F_i in Equation 2.26) used here are R_{ON}, R_{TT1} and C_{DRAM_DQ}. The simulations are performed for 9 cases as indicated by the Taguchi array method explained in Chapter 2. JMP [40] is then used to create a sensitivity profile. The table of values used for the Taguchi array and the sensitivity profile are shown in Figure 5.7. The profile allows us to interpolate the input variables and predict T_{jitter}. CACTI-IO uses the sensitivity profile to perform the interpolation.

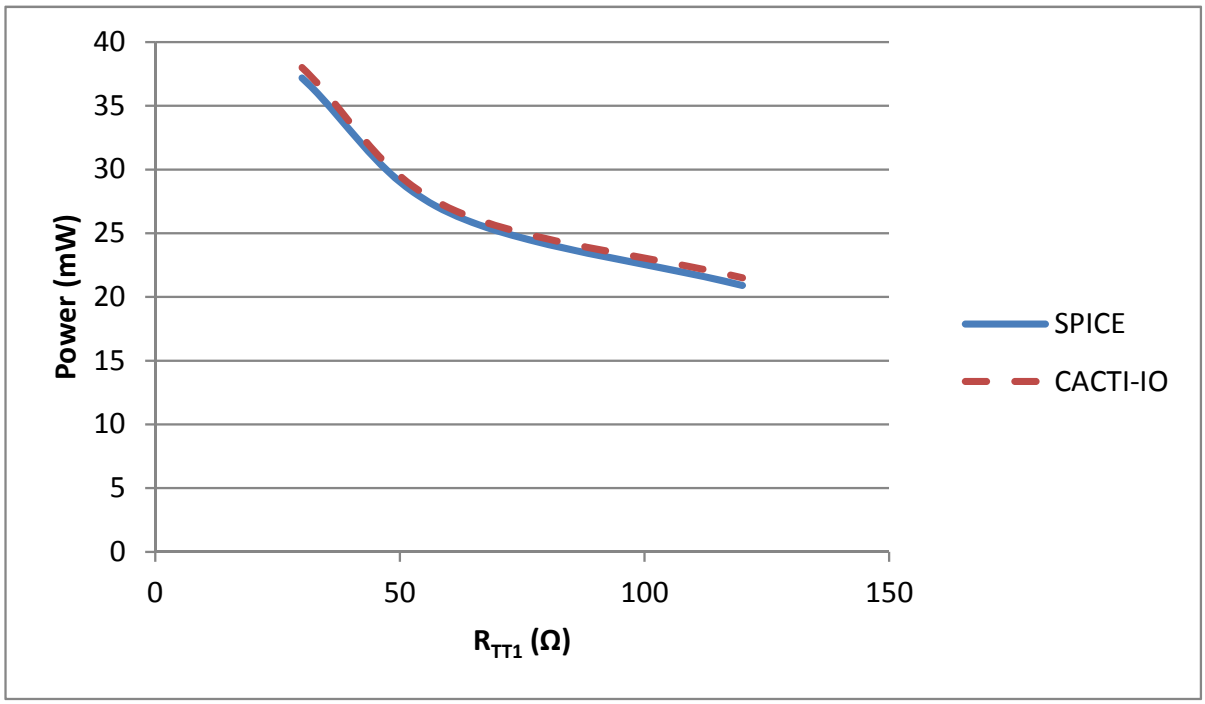


Figure 5.1: DQ Single-lane DDR3 Termination Power.

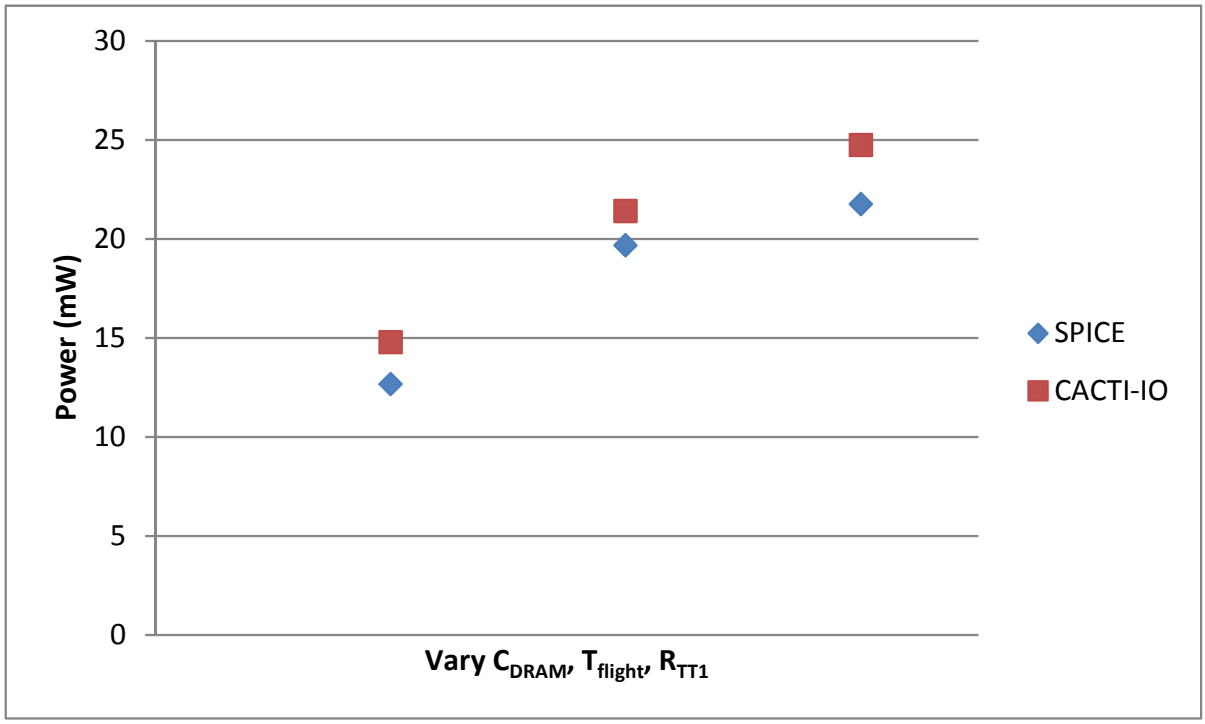


Figure 5.2: DQ Single-lane DDR3 Total IO Power.

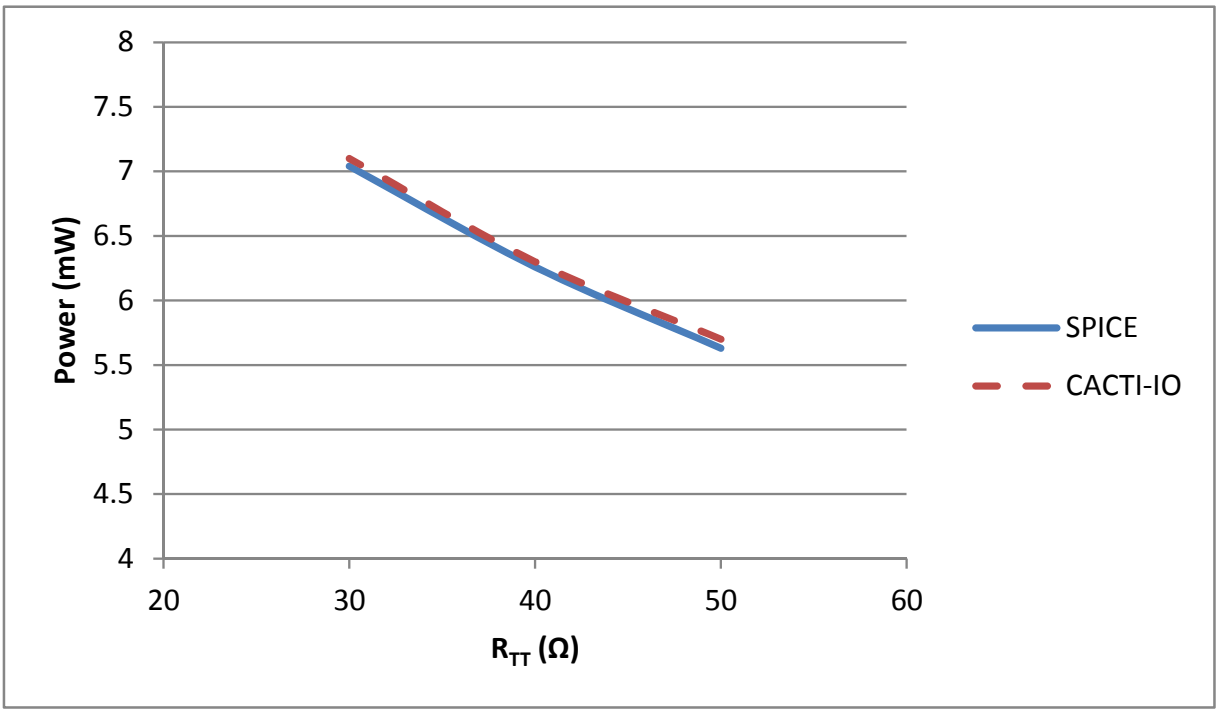


Figure 5.3: CA Single-lane DDR3 Termination Power.

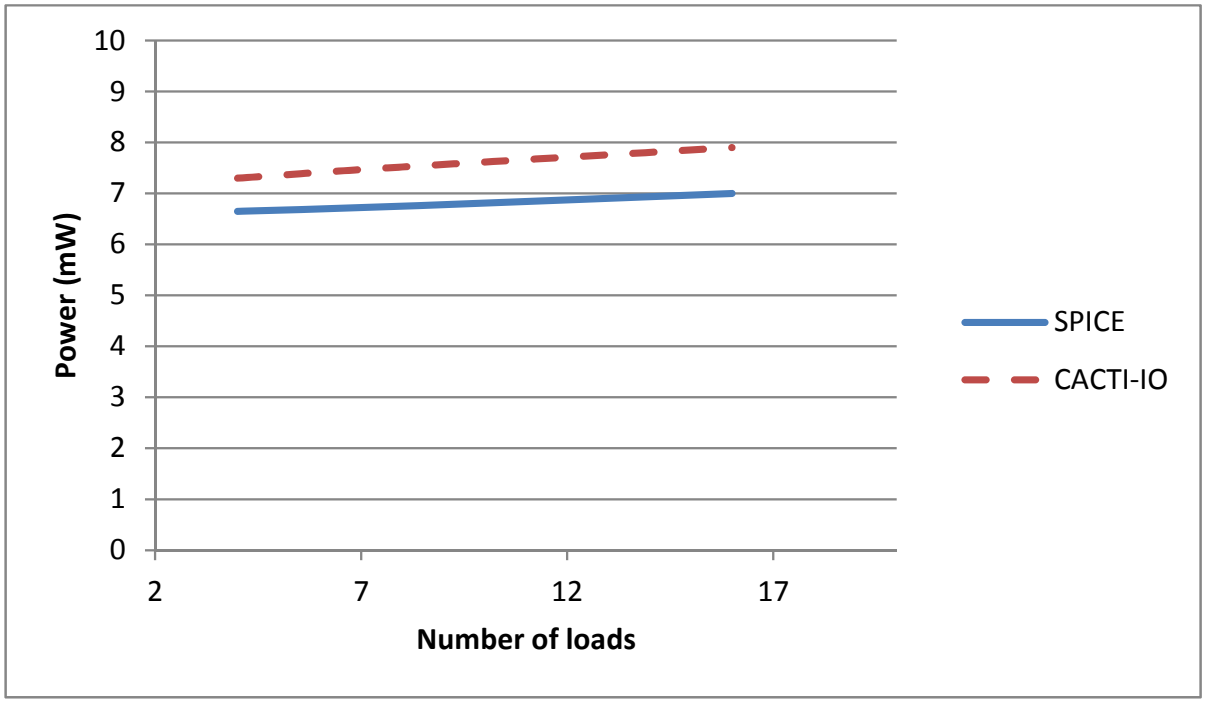


Figure 5.4: CA Single-lane DDR3 Total IO Power.

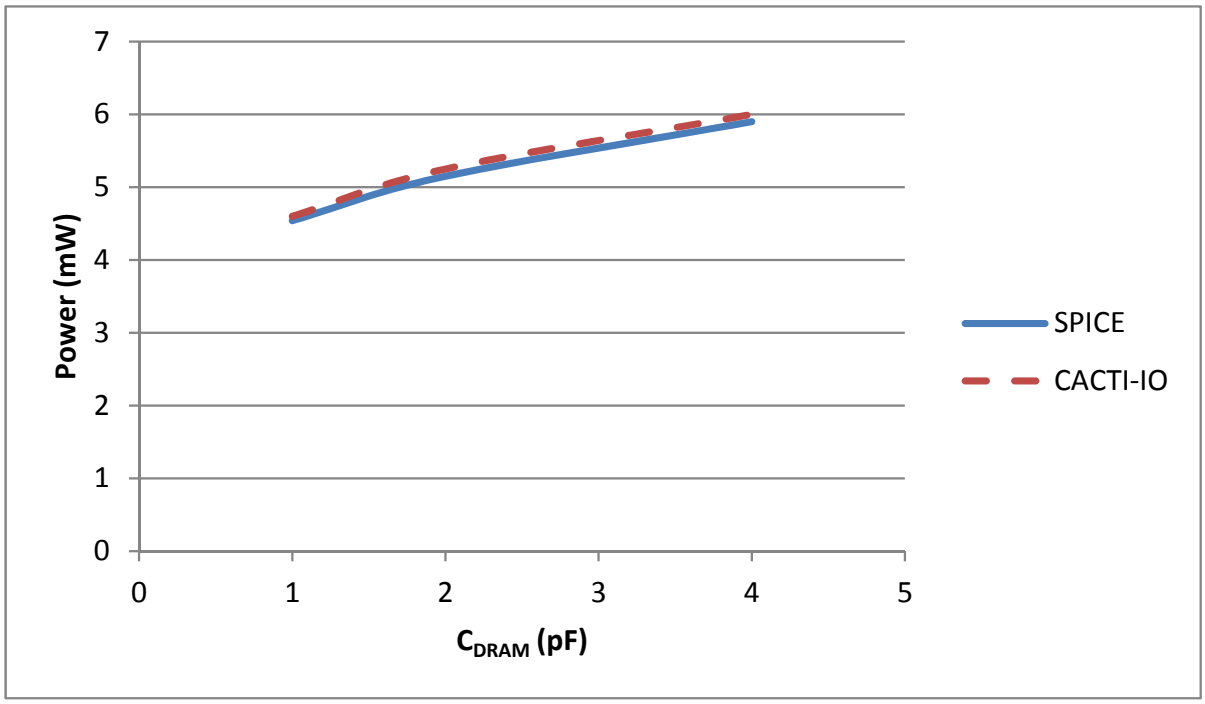


Figure 5.5: DQ Single-lane LPDDR2 Total IO Power.

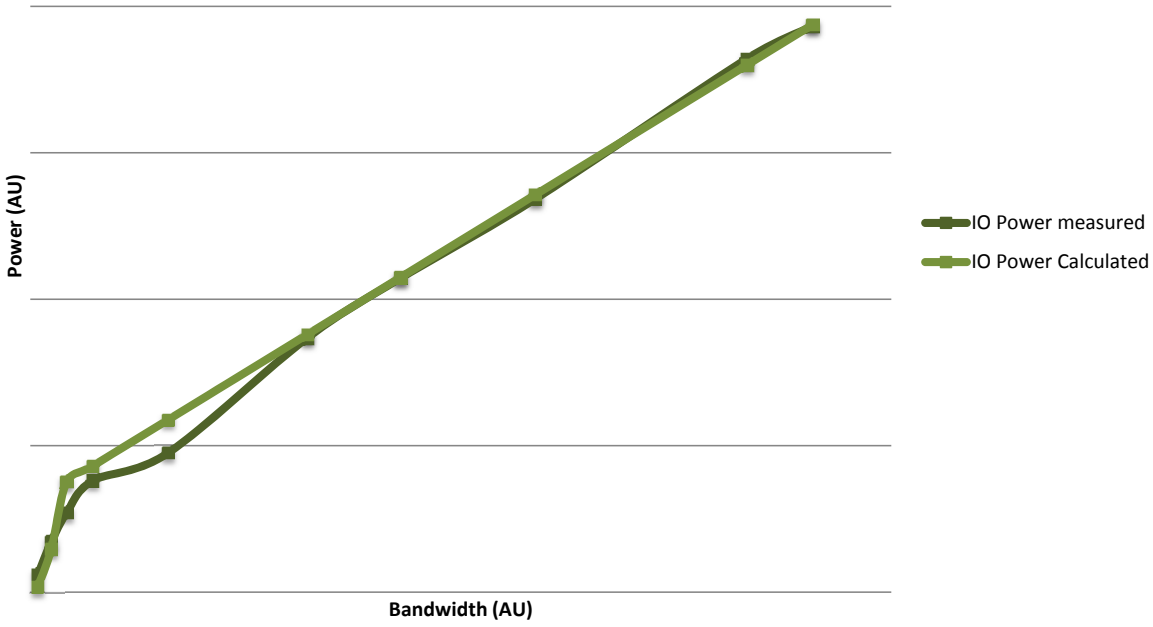


Figure 5.6: LPDDR2 WRITE Measurement vs. Model.

Ron (Ω)	Rtt (Ω)	Cdram (pF)	Pattern	Tjitter (ps)
34	30	1	−−−	45
34	40	1.5	−00	40
34	60	2	−++	66
40	30	1.5	0−0	56
40	40	2	00+	53
40	60	1	0+−	40
48	30	2	+−+	76
48	40	1	+0−	47
48	60	1.5	--0	80

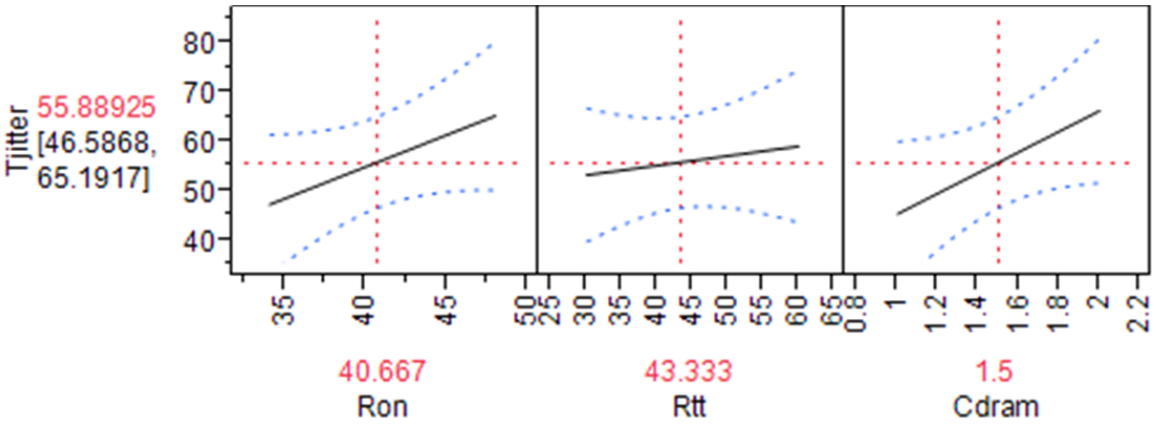


Figure 5.7: DOE Analysis on a DDR3 Channel.

Bibliography

[1] W. Dally and J. Poulton, *Digital Systems Engineering*, Cambridge University Press, 1998.

[2] H. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, 1990.

[3] D. Oh and C. Yuan. *High-Speed Signaling: Jitter Modeling, Analysis, and Budgeting*, Prentice Hall, 2011.

[4] N. Muralimanohar, R. Balasubramonian and N. P. Jouppi, CACTI 6.0 Technical Report, HPL-2009-85, HP Labs.

[5] N. Chang, K. Kim and J. Cho, "Bus Encoding for Low-Power High-Performance Memory Systems," *Proc. IEEE DAC*, 2000, pp. 800-805.

[6] A. B. Kahng and V. Srinivas, "Mobile System Considerations for SDRAM Interface Trends," *Proc. ACM/IEEE SLIP Workshop*, 2011, pp. 1-8.

[7] J. Baloria, "Micron Reinvents DRAM Memory: Hybrid Memory Cube," *Proc. IDF Workshop*, Sept. 2011.

[8] Intel's Scalable Memory Buffer. http://tinyurl.com/7xbt27o

[9] D. H. Yoon, J. Chang, N. Muralimanohar and P. Ranganathan, "BOOM: Enabling Mobile Memory Based Low-Power Server DIMMs," *Proc. IEEE ISCA*, 2012, pp 25-36.

[10] McSim. http://cal.snu.ac.kr/mediawiki/index.php/McSim

[11] C.-K. Luk et al., "PIN: Building Customized Program Analysis Tools with Dynamic Instrumentation," *Proc. ACM PLDI*, 2005, pp. 190-200.

[12] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Proc. IEEE ISCA*, 1995, pp. 24-36.

[13] H. Zheng and Z. Zhu, "Power and Performance Trade-Offs in Contemporary DRAM System Designs for Multicore Processors," *IEEE Trans. on Computers* 59(8) (2010), pp. 1033-1046.

[14] H. Lee et al., "A 16 Gb/s/Link, 64 GB/s Bidirectional Asymmetric Memory Interface," *IEEE JSSC* 44(4) (2009), pp. 1235-1247.

[15] J. Poulton, R. Palmer, A. M. Fuller, T. Greer, J. Eyles, W. J. Dally and M. Horowitz, "A 14-mW 6.25-Gb/s Transceiver in 90-nm CMOS," *IEEE JSSC* 42(12) (2007), pp. 2745-2757.

[16] F. O'Mahony et al., "A 47x10Gb/s 1.4mW/(Gb/s) Parallel Interface in 45nm CMOS," *Proc. IEEE ISSCC*, 2010, pp. 156-158.

[17] S. Thoziyoor, J. Ahn, M. Monchiero, J. B. Brockman and N. P. Jouppi, "A Comprehensive Memory Modeling Tool and its Application to the Design and Analysis of Future Memory Hierarchies," *Proc. IEEE ISCA*, 2008, pp. 51-62.

[18] Micron DRAM System Power Calculators.
http://www.micron.com/support/dram/power_calc.html

[19] JEDEC DDR3 Specification JESD79-3B.

[20] JEDEC LPDDR2 Specification JESD209-2C.

[21] JEDEC. http://www.jedec.org

[22] G. Taguchi, *Introduction to Quality Engineering*, 2nd ed., McGraw-Hill, 1996.

[23] R. Palmer, J. Poulton, A. Fuller, J. Chen and J. Zerbe, "Design Considerations for Low-Power High-Performance Mobile Logic and Memory Interfaces," *Proc. IEEE ASSCC*, 2008, pp. 205-208.

[24] J. Ellis, "Overcoming Obstacles for Closing Timing for DDR3-1600 and Beyond," *Denali MemCon*, 2010.

[25] A. Vaidyanath, "Challenges and Solutions for GHz DDR3 Memory Interface Design," *Denali MemCon*, 2010.

[26] HP Memory Technology Evolution: An Overview of System Memory Technologies. http://tinyurl.com/7mvktcn

[27] http://www.micron.com/products/dram_modules/lrdimm.html

[28] "Challenges and Solutions for Future Main Memory," *Rambus White Paper*, May 2009. http://tinyurl.com/cetetsz

[29] B. Schroeder, E. Pinheiro and W. Weber, "DRAM Errors in the Wild: A Large-Scale Field Study," *Proc. ACM SIGMETRICS*, 2009, pp. 193-204.

[30] J.-S. Kim et al., "A 1.2V 12.8GB/s 2Gb Mobile Wide-I/O DRAM with 4128 I/Os Using TSV-Based Stacking," *Proc. IEEE ISSCC*, 2011, pp. 496-498.

[31] S. Sarkar, A. Brahme and S. Chandar, "Design Margin Methodology for DDR Interface," *Proc. IEEE EPEPS*, 2007, pp. 167-170.

[32] S. Chaudhuri, J. McCall and J. Salmon, "Proposal for BER Based Specifications for DDR4," *Proc. IEEE EPEPS*, 2010, pp. 121-124.

[33] M. Qureshi, V. Srinivasan and J. Rivers, "Scalable High-Performance Main Memory System Using Phase-Change Memory Technology," *Proc. IEEE ISCA*, 2009, pp. 24-33.

[34] HP Power Advisor.
http://h18000.www1.hp.com/products/solutions /power/index.html.

[35] *International Technology Roadmap for Semiconductors*, 2011 edition. http://www.itrs.net/

[36] B. K. Casper, M. Haycock and R. Mooney, "An Accurate and Efficient Analysis Method for Multi-Gb/s Chip-to-Chip Signaling Schemes," *Proc. IEEE VLSIC*, 2002, pp. 54-57.

[37] Future-Mobile JEDEC Draft Wide IO Specification.

[38] M. A. Horowitz, C.-K. K. Yang and S. Sidiropoulos, "High-Speed Electrical Signaling: Overview and Limitations," *IEEE Trans. on Advanced Packaging* 31(4) (2008), pp. 722-730.

[39] D. Oh, F. Lambrecht, J. H. Ren, S. Chang, B. Chia, C. Madden and C. Yuan, "Prediction of System Performance Based on Component Jitter and Noise Budgets," *Proc. IEEE EPEPS*, 2007, pp. 33-36.

[40] http://www.jmp.com