

A Bank-Wise DRAM Power Model for System Simulations

Deepak M. Mathew, Éder F. Zulian, Subash Kannothe,
Matthias Jung, Christian Weis, Norbert Wehn
Microelectronic Systems Design Research Group, University of Kaiserslautern, Germany
{deepak,zulian,jungma,weis,wehn}@eit.uni-kl.de

ABSTRACT

DRAM devices contribute significantly to the power consumption of today's computing systems. As the DRAM banks are getting denser, bank-wise power contribution is becoming more and more significant in modern DRAM devices. Therefore, DDR3 and LPDDR3/4 devices support *Partial Array Self Refresh* (PASR), while the latter and *High Bandwidth Memory* (HBM) provide additionally *Per-Bank Refresh*. To evaluate the benefits of these new sophisticated features in design space explorations, it is essential to have an accurate and fast *Bank-Wise* DRAM power model, which can be easily integrated into system level simulators. To the best of our knowledge, there exists no DRAM power model, which supports the aforementioned features. In this work, we present the fundamental equations for modeling bank-wise DRAM power. This model supports PASR and Per-Bank Refresh and is calibrated with measurements from various DRAM devices to improve the accuracy instead of using pessimistic datasheet values.

CCS Concepts

•Computing methodologies → Modeling and simulation; •Hardware → Dynamic memory; Power estimation and optimization;

Keywords

DRAM; Power; Simulation

1. INTRODUCTION

The contribution of *Dynamic Random Access Memory* (DRAM) to the total power consumption in today's computing systems is getting more and more significant [3, 10, 18]. Moreover, the size of a single DRAM bank is increased largely (>1Gbit) following the trend of memory density. This is due to the fact that the number of memory banks stays unchanged across DRAM generations (8Gbit DDR3 device: 8

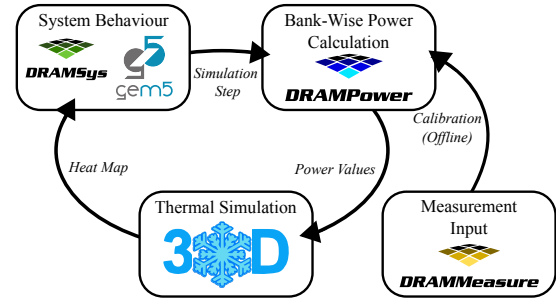


Figure 1: Simulation Framework using Bank-Wise DRAM-Power

banks, 16Gb DDR4 device: 8/16 banks). This results effectively in larger and more power hungry banks. A DRAM device cannot be considered anymore as a single entity. Each bank has now to be handled independently with respect to power, temperature and performance. Especially, concerning hotspots in 3D integrated environments the DRAM banks require to be analyzed and controlled separately, since the temperature gradient across the chip varies in vertical and lateral directions [26]. Thus, industry has introduced new features into recent DRAM standards such as *Partial Array Self Refresh* (PASR) and *Per-Bank Refresh*.

PASR is a power-down feature, which has been introduced by JEDEC as part of LPDDR and DDR2 standards, and still persists in state-of-the-art low power DRAMs, such as LPDDR4. This allows to select a group of banks to self-refresh instead of refreshing the complete DRAM, when it is idle. Data in the other banks will be lost. Since smartphones exhibit a burst-like access pattern with long idle periods between two consecutive interactive sessions [8], they can make use of the PASR functionality to optimize the stand-by power consumption. The authors of [24] provide an overview of the current Linux memory management infrastructure and some of the challenges with respect to memory power optimization from a software point of view. They also propose an allocation framework aware of bank boundary information that promotes the use of bank-selective PASR. Memory power management is a concern among Linux developers. The patchset proposed in [7] targets PASR, while [1] provides the Linux memory management subsystem with an understanding of the granularity at which DRAM modules can be power-managed with the aim of using power-management capabilities in modern memory hardware.

The *Per-Bank Refresh* command is a newly introduced

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

RAPIDO '17, January 23-25, 2017, Stockholm, Sweden

© 2017 ACM. ISBN 978-1-4503-4840-9/17/01...\$15.00

DOI: <http://dx.doi.org/10.1145/3023973.3023978>

feature in HBM and LPDDR3/4 memories. This allows to refresh a single bank while the other banks are serving requests, instead of suspending all banks when a refresh is in progress. In summary, we face the following challenges: (i) Larger DRAM bank sizes, (ii) Thermal hotspots in 3D integrated circuits, (iii) New sophisticated features (PASR and Per-Bank Refresh). To tackle these, a more granular DRAM power model for system level explorations is required. To maintain the accuracy vs. speed trade-off the lowest level of abstraction is the DRAM bank. Such a *bank-wise* DRAM power model enables to obtain a detailed thermal map of the used DRAM device. However, to the best of our knowledge, there exists neither a bank-wise DRAM power model, nor a model that supports the new sophisticated features.

Consequently, we present a new advanced version of DRAM-Power [5], a well known tool, which estimates DRAM power from realistic DRAM current and workload specifications to calculate the bank-wise power for each DRAM operation. In this paper, we also elaborate a set of new equations to support PASR and Per-Bank Refresh. We validate this model by conducted measurements on various DRAM devices from different vendors. Figure 1 shows, the integration of the new advanced bank-wise DRAMPower into a typical holistic simulation framework consisting of DRAMMeasure¹, 3D-ICE [27] (thermal simulator) and DRAMSys [15].

The rest of this paper is organized as follows: Section 2 explains the basics of DRAM power modeling and introduces the underlying equations of DRAMPower. Our improvements to DRAMPower and new equations are presented in Section 3. The used measurement setup is presented in Section 4. Section 5 shows experimental results obtained by measurements and simulations. The paper is concluded in Section 6.

2. DRAM POWER MODELING

A well-known and often used DRAM power model is provided by Micron in form of a spread sheet [23]. However, there are some limitations in that model. Micron uses the minimal timing constraints from the datasheet specifications instead of the actual timings. But in practice there are dependencies between consecutive memory accesses so that the controller may accelerate or postpone commands. Furthermore, Micron assumes that the controller uses a close-page policy (precharge after each memory access) and that there is only one bank open at the same time. Due to this, a large lack of flexibility and accuracy exists in this model. A recently published model is DRAMPower [5], which uses the actual timings from the transactions. DRAMPower has been wrapped by Jung et al. into a library that can be used in event based simulators [15]. Therefore, DRAMPower is widely used for instance in the gem5 simulator [2], the DRAM simulator Ramulator [17] and the SystemC/TLM based DRAM design space exploration framework DRAM-Sys [15, 16]. The interesting instances of time for the power model are the events in these simulators, which contain all timing information that are needed for the calculation of the power consumption using the equations described in this paper.

The DRAM power consumption consists of a background or static part and an active part, which is needed for each

Table 1: Key Parameters for DRAMs [11]

Name	Explanation
t_{RCD}	<i>Row to Column Delay</i> : The time interval between ACT and RD on the same bank.
t_{RP}	<i>Row Precharge</i> : The time interval that it takes for a DRAM array to be precharged (PRE) for another row access.
t_{RAS}	<i>Row Active</i> : The minimum active time for a row.
t_{RC}	<i>Row Cycle</i> : The fastest time to ACT and PRE the same row.
t_{CL}	<i>CAS Latency</i> : The delay from a RD/WR command until data can be read/written at the interface.
t_{RFC}	<i>Refresh Cycle</i> : The time interval that it takes for a refresh.
I_{DD0}	<i>One Bank Active Precharge Current</i> : Measured across ACT and PRE commands to one bank (other banks remain precharged).
I_{DD2N}	<i>Precharge Standby Current</i> : Measured when all banks are precharged (PRE).
I_{DD3N}	<i>Active Standby Current</i> : Measured when all banks are active (ACT).
I_{DD4R}	<i>Burst Read Current</i> : Measured during read operation, assuming seamless write data burst with all data bits toggling between bursts and all banks open, with the RD commands cycling through all the banks.
I_{DD4W}	<i>Burst Write Current</i> : Similar to I_{DD4R} , but with WR commands.
I_{DD2P0}	<i>Precharge Power Down Current</i> : Measured during precharge power down mode.
I_{DD3P}	<i>Active Power Down Current</i> : Measured during active power down mode.
I_{DD5}	<i>Refresh Current</i> : Measured during refresh operation, with REF commands issued every t_{RFC} .
I_{DD6}	<i>Self Refresh Current</i> : Measured when the DRAM is in Self Refresh mode.
V_{DD}	<i>Supply Voltage</i>

command (ACT, PRE, RD, WR ...), called command power. Table 1 shows the required timing and current values in vendor's datasheets, denoted as specific I_{DD} values.

For instance, I_{DD0} is measured by an ACT-PRE (activate-precharge) sequence, that is issued to the device in the minimum time interval ($t_{RC} = t_{RAS} + t_{RP}$).

In this section, we review briefly the basic equations of the DRAMPower model for the background and the active energy, according to [4, 5, 14] and [13]. In Section 3, we present enhanced versions of these equations.

2.1 DRAM Background Energy

While all banks are in the active state, the memory consumes I_{DD3N} as background current. When all banks are precharged, it draws only I_{DD2N} . However, to calculate the complete power consumption we need T_{ACT}^{BG} (the time where the memory is in the active state) and T_{PRE}^{BG} (time in precharge state) which can be easily derived by analyzing simulation events. The active and precharge background energy can be calculated with the following equations:

$$E_{ACT}^{BG} = V_{DD} \cdot I_{DD3N} \cdot T_{ACT}^{BG} \quad (1)$$

$$E_{PRE}^{BG} = V_{DD} \cdot I_{DD2N} \cdot T_{PRE}^{BG} \quad (2)$$

However, Equation (1) is very pessimistic: The DRAM is in the active (open) state when at least one bank is activated, but, the used I_{DD3N} in the equation is the measured current when *all* banks are active. Therefore, Jung et al. [13] presented an accurate equation that considers the exact number of active banks, which we will shortly review in Section 3.

¹Details can be found at <http://www.uni-kl.de/3d-dram/tools/drammeasure/>

2.2 DRAM Command Energy

For each issued DRAM command a certain amount of energy is consumed. In the following parts the energy contributions for each command are explained.

2.2.1 Activate and Precharge

The energy of an ACT command is computed with the t_{RAS} (minimum time between ACT and PRE to the same bank) and I_{DD0} . This is done similarly for the precharge command with the t_{RP} value. Since I_{DD0} already contains the background currents I_{DD2N} and I_{DD3N} with their respective minimum timing constraints, we have to subtract them from I_{DD0} :

$$E_{ACT} = V_{DD} \cdot (I_{DD0} - I_{DD3N}) \cdot t_{RAS} \quad (3)$$

$$E_{PRE} = V_{DD} \cdot (I_{DD0} - I_{DD2N}) \cdot t_{RP} \quad (4)$$

In Section 3 we will modify the above equations to improve their accuracy.

2.2.2 Read and Write

Similar to the active and precharge commands there is an average current I_{DD4R} for the RD and I_{DD4W} for the WR command. Since both of the currents also include I_{DD3N} , we have to subtract it from I_{DD4R} and I_{DD4W} . To get the energy consumption for one read or write command we also have to multiply the time of the data transfer, which is calculated by the ratio of burst length (BL) and data rate (DR). For SDR memories this evaluates to $\frac{BL}{1}$ and for DDR memories to $\frac{BL}{2}$.

$$E_{RD} = V_{DD} \cdot (I_{DD4R} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \quad (5)$$

$$E_{WR} = V_{DD} \cdot (I_{DD4W} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \quad (6)$$

2.2.3 Refresh

The refresh energy is calculated with I_{DD5} . Again, the background current must be subtracted:

$$E_{REF} = V_{DD} \cdot (I_{DD5} - I_{DD3N}) \cdot t_{RFC} \quad (7)$$

2.2.4 Power Down and Self Refresh

There are additional energy contributors like power down and self refresh. For the sake of space we do not include these explanations into this section. Refer to Subsection 3.3 and 3.5 for the revised formulas.

2.3 Complete Trace Energy and Average Power

DRAMPower computes the average power consumption for the entire simulation by combining all the energy contributions from above into the following equation:

$$\begin{aligned} E_{Total} = & E_{ACT}^{BG} + E_{PRE}^{BG} \\ & + N_{REF} \cdot E_{REF} + N_{ACT} \cdot E_{ACT} \\ & + N_{PRE} \cdot E_{PRE} + N_{RD} \cdot E_{RD} \\ & + N_{WR} \cdot E_{WR} \end{aligned} \quad (8)$$

N_{ACT} stands for the number of activate commands and N_{PRE} , N_{RD} , N_{WR} for the number of precharge, read, write commands, respectively. The average power consumption can be obtained by dividing the total energy with the total runtime:

$$P_{AVG} = \frac{E_{Total}}{T_{Total}} \quad (9)$$

3. NEW CONTRIBUTIONS

In this section, we present a new analytical model for calculating the bank-wise power contribution. Furthermore, for the first time we present a new set of equations in order to simulate the DRAM features PASR and Per-Bank Refresh.

3.1 Revised Bank-Wise Background Energy

In order to improve DRAMPower's calculation of the active background energy and to introduce the bank sensitivity, Jung et al. [13] have presented a new formula that will replace Equation (1). We first revisit this formula. Second, we present a new formula that contain all necessary bank-wise improvements.

As explained in Section 2.1, a DRAM is in precharged state if no bank is opened (I_{DD2N}), and it is in active state if at least one bank is activated. However, for this state the complete current I_{DD3N} is accounted, which represents the current when *all* banks are activated.

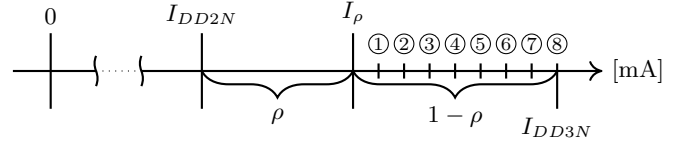


Figure 2: Improved Background Energy Calculation

Figure 2 shows the currents of a DRAM with eight banks ($B = 8$) on a number line. For the new equation that considers the exact number of activated banks, [13] introduced a new current I_ρ . This represents the current which is consumed by shared resources such as I/O interface, supply voltage generators and common control logic. Each activated bank (① - ⑧) adds a specific current portion to I_ρ . When all banks are activated the current is equal to I_{DD3N} .

First, the active time for each bank must be accounted separately. Therefore, additionally to T_{ACT}^{BG} , a new parameter $T_{ACT}^{BG*}(b)$ was introduced, which denotes the active time of a specific bank $b \in \{0, \dots, B-1\}$, where B denotes the total number of banks. The energy part contributed by each bank b can be written as

$$E_{ACT}^{BG*}(b) = V_{DD} \cdot \frac{1}{B} \cdot (I_{DD3N} - I_\rho) \cdot T_{ACT}^{BG*}(b). \quad (10)$$

Second, if at least one bank is open a shared energy, which is caused by resources such as I/O interface, supply voltage generators and common control logic must be accounted:

$$E_{ACT}^{BG \text{ Shared}} = V_{DD} \cdot I_\rho \cdot T_{ACT}^{BG} \quad (11)$$

The new bank-wise active background energy results in

$$E_{ACT}^{BG}(b) = \frac{1}{B} \cdot E_{ACT}^{BG \text{ Shared}} + E_{ACT}^{BG*}(b). \quad (12)$$

As depicted in Figure 2, the relationship between I_ρ , I_{DD2N} and I_{DD3N} can be expressed by a relative factor ρ , which is a vendor specific parameter:

$$\rho = \frac{I_{\rho} - I_{DD2N}}{I_{DD3N} - I_{DD2N}} \quad (13)$$

Thus the new bank-wise active background energy can be expressed as

$$E_{ACT}^{BG}(b) = V_{DD} \cdot \frac{1}{B} \left[(\rho \cdot I_{DD3N} + (1 - \rho) \cdot I_{DD2N}) \cdot T_{ACT}^{BG} + (1 - \rho) \cdot (I_{DD3N} - I_{DD2N}) \cdot T_{ACT}^{BG*}(b) \right] \quad (14)$$

The modified formula for precharge background energy is as follows:

$$E_{PRE}^{BG}(b) = \frac{1}{B} \cdot V_{DD} \cdot I_{DD2N} \cdot T_{PRE}^{BG} \quad (15)$$

3.2 Revised Bank-Wise Command Energy

In this section, we present two enhancements to the existing DRAMPower model. First, bank specific command counters are introduced to calculate the bank-wise command energy. $N_{ACT}(b)$ stands for the number of activate commands for bank b and $N_{PRE}(b)$, $N_{RD}(b)$, $N_{WR}(b)$ for the number of precharge, read, write, refresh commands respectively. Second, we improve the formulas for ACT and PRE command energy calculations to make them more accurate.

3.2.1 Activate and Precharge

I_{DD0} is measured when only a single bank is active whereas I_{DD3N} is measured when all banks are active. Therefore, to obtain the command energy for a single ACT command we have to subtract the active background current for a single bank $I_{\textcircled{0}}$ from I_{DD0} instead of subtracting the total active background current I_{DD3N} . $I_{\textcircled{0}}$ can be calculated from Figure 2 as follows:

$$I_{\textcircled{0}} = \frac{1}{B} \left[I_{DD3N} + (B - 1) \cdot [\rho(I_{DD3N} - I_{DD2N}) + I_{DD2N}] \right] \quad (16)$$

Therefore Equation (3) is corrected as follows²:

$$E_{ACT}(b) = V_{DD} \cdot [I_{DD0} - I_{\textcircled{0}}] \cdot t_{RAS} \cdot N_{ACT}(b) \quad (17)$$

Similarly, for calculation of the PRE command energy we need to subtract $I_{\textcircled{0}}$ from I_{DD0} , instead of subtracting I_{DD2N} . This is because a bank will precharge completely only after t_{RP} upon receiving the PRE command. Therefore Equation (4) is corrected as follows:

$$E_{PRE}(b) = V_{DD} \cdot [I_{DD0} - I_{\textcircled{0}}] \cdot t_{RP} \cdot N_{PRE}(b) \quad (18)$$

3.2.2 Read and Write

We modified Equations (5) and (6) to make them bank-wise:

$$E_{RD}(b) = V_{DD} \cdot (I_{DD4R} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \cdot N_{RD}(b) \quad (19)$$

²Note that when $\rho = 1$ in Equation (16), $I_{\textcircled{0}} = I_{DD3N}$ and Equation (17) transforms to Equation (3).

$$E_{WR}(b) = V_{DD} \cdot (I_{DD4W} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \cdot N_{WR}(b) \quad (20)$$

$N_{RD}(b)$ and $N_{WR}(b)$ denote the number of read and write commands to a specific bank b .

3.2.3 Refresh

The REF command energy can be calculated from Equation (7) as follows:

$$E_{REF}(b) = \frac{1}{B} \cdot V_{DD} \cdot (I_{DD5} - I_{DD3N}) \cdot t_{RFC} \cdot N_{REF} \quad (21)$$

N_{REF} denotes the total number of REF commands.

3.3 Active and Precharge Power Down Energy

The bank-wise energy for active and precharge power-down states can be expressed as follows:

$$E_{PDNA}(b) = \frac{1}{B} \cdot V_{DD} \cdot I_{DD3P} \cdot T_{PDNA} \quad (22)$$

$$E_{PDNP}(b) = \frac{1}{B} \cdot V_{DD} \cdot I_{DD2P0} \cdot T_{PDNP} \quad (23)$$

T_{PDNA} and T_{PDNP} denotes the total time spent in active and precharge power down states respectively.

3.4 Per-Bank Refresh Energy

We implement the new Per-Bank Refresh feature in DRAMPower to support state-of-the-art DRAM devices [20, 21] and future investigations using bank-wise refresh [6, 26].

The refresh command is internally executed as a sequence of ACT-PRE commands. Therefore, to calculate the bank-wise refresh energy for this command, we add-up activate and precharge energies:

$$E_{PB-REF}(b) = V_{DD} \cdot (I_{DD0} - I_{\textcircled{0}}) \cdot (t_{RAS} + t_{RP}) \cdot N_{PB-REF}(b) \quad (24)$$

$N_{PB-REF}(b)$ denotes the number of Per-Bank Refresh commands issued to bank b .

3.5 Partial Array Self Refresh Energy

In addition to the presented bank-wise command energy calculation we present a model for simulating *Partial Array Self Refresh* (PASR) in DRAMPower. PASR is a feature to choose a set of banks to self-refresh instead of refreshing the complete DRAM, when it is idle. As shown in Figure 3, a DDR3 DRAM can enter into PASR in one of the eight available modes. These modes can be dynamically configured by programming the DRAM mode register with a *Mode Register Set* (MRS) command. The DRAM will only refresh a specific set of banks corresponding to the selected mode, when it enters self-refresh. For example, bank 0 and bank 1 are self-refreshed in mode 010, while bank 6 and bank 7 are self-refreshed in mode 110. As shown in Figure 3, the banks b which are selected in mode m can be represented in a matrix form:

$$P(b, m) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (25)$$

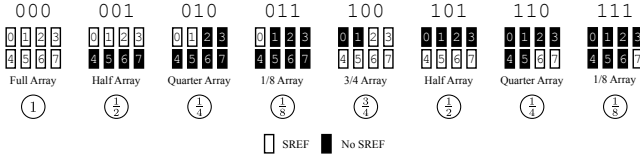


Figure 3: Visualisation of PASR Modes

$P(b, m) = 1$ indicates that the bank b is self-refreshed in mode m and $P(b, m) = 0$ indicates that it is not. Analogous

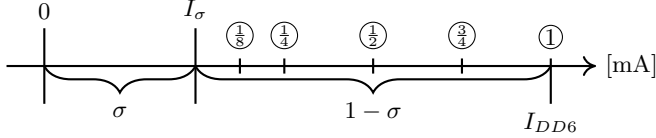


Figure 4: PASR Energy Calculation

to Equation (14) we derive a new formula for the power consumption of each DRAM bank when it is in PASR mode. Figure 4 shows the currents drawn by the different PASR modes indicated in Figure 3. Similarly to the active background power description we introduce a new current I_σ , which represents shared resources. The relation between I_σ and I_{DD6} is

$$\sigma = \frac{I_\sigma}{I_{DD6}}. \quad (26)$$

The power per bank b in mode m can be expressed as

$$P_{PASR}(b, m) = V_{DD} \cdot I_{DD6} \cdot \frac{1}{B} [\sigma + (1 - \sigma) \cdot P(b, m)]. \quad (27)$$

Finally, the SREF energy per bank is calculated with the following equation:

$$E_{PASR}(b) = \sum_{m=0}^7 [P_{SREF}(b, m) \cdot T_{PASR}(b, m)] \quad (28)$$

$T_{PASR}(b, m)$ denotes the time spent by bank b in each PASR mode m . This new formula enables future investigations and design space explorations on system level virtual platforms, with respect to low power implementations for applications with high idle time, e.g. smartphones.

It is obvious from Equation (25) that the *Self Refresh* (SREF) is a special case of PASR, in which $m = 0$ (all banks are refreshed). Therefore, we will get the equation for SREF power by substituting $m = 0$ in Equation (27). The vendor specific parameter σ used in the above equations are obtained from measurements presented in Section 5.

4. MEASUREMENT SETUP

In order to obtain the presented vendor specific parameters (ρ and σ) precise measurements have to be conducted. Therefore, we developed a custom platform to measure the current consumption and to heat up the DRAM devices of DDR3 SO-DIMM modules. The heating section consists of a mechanical setup, which is placed on the surface of the DRAM devices. For analyzing the current consumption of DRAMs, we designed an adapter PCB for DDR3 SODIMMs

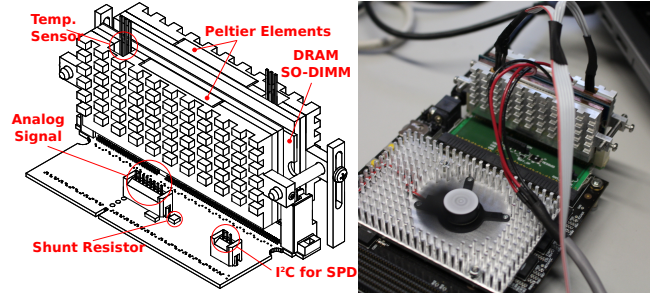


Figure 5: Measurement Platform

(6 Layers) that conforms to JEDEC standard requirements. The DDR3 SO-DIMM adapter board is plugged into a Xilinx FPGA based evaluation platform. The standard *Memory Interface Generator* (MIG) memory controller from Xilinx is customized to generate the required command and data sequences for the measurements. A *Virtual Input/Output* (VIO) core connected to the Custom MIG enables the real-time monitoring and control of the internal FPGA signals.

5. EXPERIMENTAL RESULTS

With the setup presented in Section 4 we measured several DDR3 DRAM devices from different vendors. Figure 6 shows the results for Hynix, Qimonda and Samsung devices. A lower value of σ indicates the chances of saving more energy when entering PASR modes. Among all measured devices, Samsung devices show lower σ values compared to other vendors. Another observation is that, as the devices are getting denser the value of σ decreases, whereas I_{DD6} increases, both together increase the power saving potential of PASR. This highlights the benefits using PASR in architectures with future DRAM devices.

To demonstrate the functionality of our new bank-wise DRAMPower model, we conducted simulations, using the parameters obtained from the Samsung K4B1G1646E-HCF8 device with a $\rho = 0.53$ and *row-bank-column* (RBC) and *bank-row-column* (BRC) address mapping schemes. The traces are generated from the CHStone [9] and the MediaBench [22] benchmarks by means of the SimpleScalar simulator with 16KB L1 D-cache, 16KB L1 I-cache, 128KB shared L2 cache and 32 byte cache line configuration. We filtered out the L2 cache misses for instructions and data, and obtained a trace of the transactions meant for the DRAM. Figure 7 shows the bank-wise energy consumption for three different input traces. The red line indicates the average energy values generated by the original version of DRAMPower. In comparison to the old model, the new DRAMPower model gives individual energy estimates for each bank. Especially, for the *unepic-brc* we observe a large difference to the old model, which averages the energy over all banks.

Per bank power consumption values permit bank-wise thermal simulations and consequently the closed loop simulation depicted in Figure 1. In Figure 8 we plot the thermal map of the trace *unepic-brc*, using 3D-ICE [27]. Due to the higher utilization of Bank 0 we clearly observe a temperature difference between the banks. Especially for research on approximate DRAMs [19, 12, 25] an accurate thermal map is needed, since the number of retention errors in a DRAM grows exponentially while increasing the temperature.

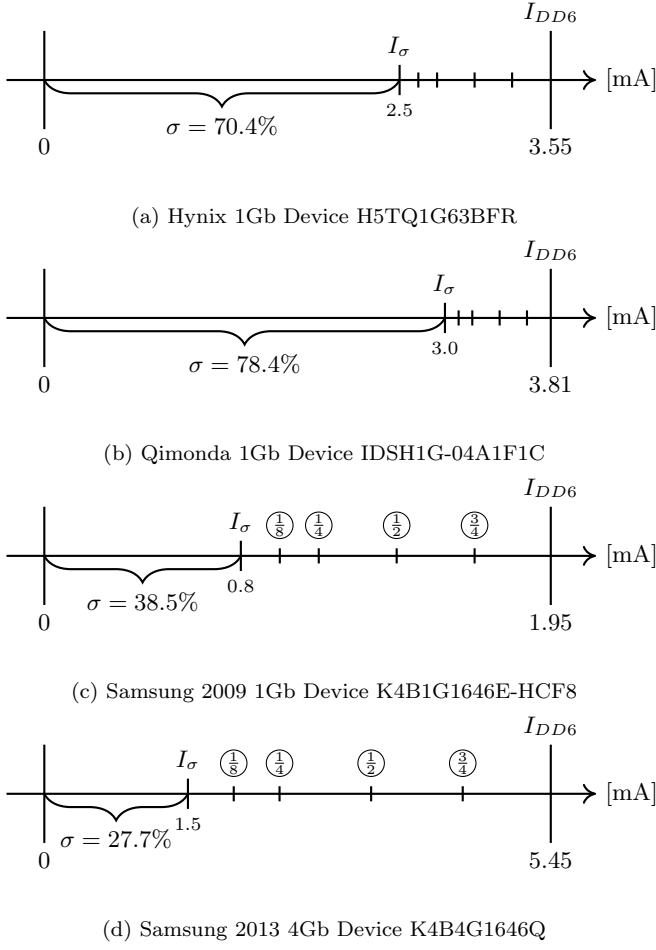


Figure 6: PASR Measurements

6. CONCLUSION

We presented a new bank-wise DRAM power model, which calculates the per bank energy for each DRAM operation. Since future DRAMs will be denser and larger with the same number of banks, it is key to obtain bank-wise power consumption, especially for thermal simulations. We also enhanced the state-of-the-art equations for ACT and PRE command energy calculations using a vendor specific parameter. Finally, a new set of equations is unveiled to support simulations with PASR and Per-Bank Refresh. Another vendor specific parameter is proposed, which enables an accurate power calculation of the PASR mode. By conducting experiments with our measurement platform, we showed the variation of this parameter among DRAM devices from different vendors, and its scaling trend for denser devices. The new DRAMPower model supports more accurate design space explorations with existing commercial DRAM devices, and for future investigations. We have contributed our results to the official DRAMPower code repository.

Figure 9 shows the dependencies of the DRAM currents with respect to temperature. In our future work, we will integrate this typical exponential behaviour into the here presented equations and DRAMPower.

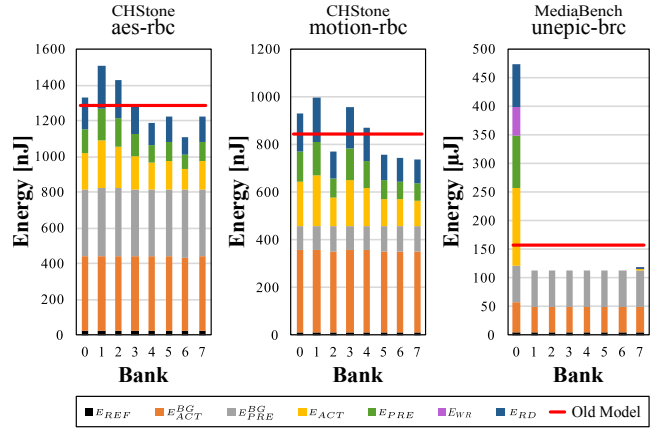


Figure 7: Bank Energy Consumption

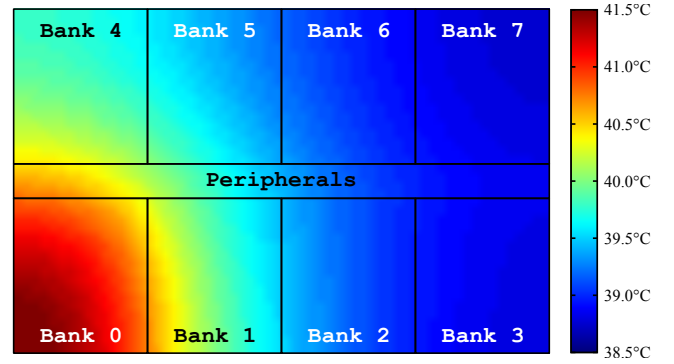


Figure 8: Thermal Behavior of *unepic-brc*

Acknowledgment

The authors thank Felipe Prado, Sven Goossens, Benny Åkesson, Karthik Chandrasekar, for their support. This work was partially funded by the German Research Foundation (DFG) as part of the priority program *Dependable Embedded Systems* SPP 1500 (<http://spp1500.itec.kit.edu>), the DFG grant no. WE2442/10-1 and the Carl-Zeiss Stiftung grant no. 0563-2.8/558/1.

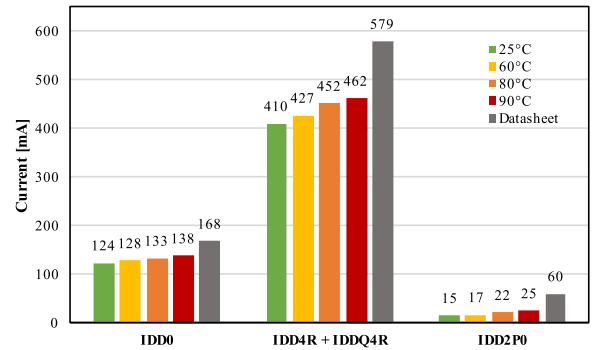


Figure 9: Variation of $IDDs$ with Temperature

7. REFERENCES

- [1] S. S. Bhat. mm: Linux Memory Power Management. <https://lwn.net/Articles/568369/>, 2013.
- [2] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.
- [3] A. Carroll and G. Heiser. The Systems Hacker’s Guide to the Galaxy Energy Usage in a Modern Smartphone. In *Proceedings of the 4th Asia-Pacific Workshop on Systems*, APSys ’13, pages 5:1–5:7, New York, NY, USA, 2013. ACM.
- [4] K. Chandrasekar, B. Akesson, and K. Goossens. Improved Power Modeling of DDR SDRAMs. In *proc. DSD’11*, 2011.
- [5] K. Chandrasekar, C. Weis, Y. Li, B. Akesson, O. Naji, M. Jung, N. Wehn, and K. Goossens. DRAMPower: Open-source DRAM power & energy estimation tool. <http://www.drampower.info>.
- [6] K. K.-W. Chang, D. Lee, Z. Chishti, A. R. Alameldeen, C. Wilkerson, Y. Kim, and O. Mutlu. Improving DRAM performance by parallelizing refreshes with accesses. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, pages 356–367. IEEE, 2014.
- [7] M. Coquelin. PASR: Partial Array Self-Refresh Framework. <https://lwn.net/Articles/478049/>, 2012.
- [8] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys ’10, pages 179–194, New York, NY, USA, 2010. ACM.
- [9] Y. Hara, H. Tomiyama, S. Honda, and H. Takada. Proposal and Quantitative Analysis of the CHStone Benchmark Program Suite for Practical C-based High-level Synthesis. *Journal of Information Processing*, 17:242–254, 2009.
- [10] U. Hoelzle and L. A. Barroso. *The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 1st edition, 2009.
- [11] B. Jacob, S. Ng, and D. Wang. *Memory Systems: Cache, DRAM, Disk*. Elsevier Science, 2010.
- [12] M. Jung, D. M. Mathew, C. Weis, and N. Wehn. Approximate Computing with Partially Unreliable Dynamic Random Access Memory: Approximate DRAM. In *IEEE/ACM Design Automation Conference (DAC)*, June 2016.
- [13] M. Jung, D. M. Mathew, É. F. Zulian, C. Weis, and N. Wehn. A New Bank Sensitive DRAMPower Model for Efficient Design Space Exploration. In *International Workshop on Power And Timing Modeling, Optimization and Simulation (PATMOS 2016)*, 2016.
- [14] M. Jung, C. Weis, P. Bertram, and N. Wehn. Power Modelling of 3D-Stacked Memories with TLM2.0 based Virtual Platforms. In *Synopsys User Group Conference (SNUG), May, 2013, Munich, Germany.*, 2013.
- [15] M. Jung, C. Weis, and N. Wehn. DRAMSys: A flexible DRAM Subsystem Design Space Exploration Framework. *IPSSJ Transactions on System LSI Design Methodology (T-SLDM)*, August 2015.
- [16] M. Jung, C. Weis, N. Wehn, and K. Chandrasekar. TLM modelling of 3D stacked wide I/O DRAM subsystems: a virtual platform for memory controller design space exploration. In *Proceedings of the 2013 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, RAPIDO ’13, pages 5:1–5:6, New York, NY, USA, 2013. ACM.
- [17] Y. Kim, W. Yang, and O. Mutlu. Ramulator: A Fast and Extensible DRAM Simulator. *IEEE Computer Architecture Letters*, PP(99):1–1, 2015.
- [18] J. Krueger, D. Donofrio, J. Shalf, M. Mohiyuddin, S. Williams, L. Oliker, and F.-J. Pfreundt. Hardware/software co-design for energy-efficient seismic modeling. In *Proceedings of the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12, 2011.
- [19] S. Liu, K. Pattabiraman, T. Moscibroda, and B. G. Zorn. Flikker: Saving DRAM Refresh-power Through Critical Data Partitioning. *SIGPLAN Not.*, 46(3):213–224, Mar. 2011.
- [20] JEDEC Solid State Technology Association. Low Power Double Data Rate 4 (LPDDR4). *JEDEC Standard JESD209-4*, 2014.
- [21] JEDEC Solid State Technology Association. High Bandwidth Memory (HBM) DRAM. *JEDEC Standard JESD235A*, 2015.
- [22] MediaBench Consortium. Mediabench. <http://euler.slu.edu/~fritts/mediabench/>, 2015, last access 28.02.2015.
- [23] Micron. DDR3 SDRAM System Power Calculator,. last access 2014-07-03, jul 2011.
- [24] A. Panwar and K. Gopinath. Towards Practical Page Placement for a Green Memory Manager. In *2015 IEEE 22nd International Conference on High Performance Computing (HiPC)*, pages 155–164, Dec 2015.
- [25] A. Raha, H. Jayakumar, S. Sutar, and V. Raghunathan. Quality-aware Data Allocation in Approximate DRAM. In *Proceedings of the 2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, CASES ’15, pages 89–98, Piscataway, NJ, USA, 2015. IEEE Press.
- [26] M. Sadri, M. Jung, C. Weis, N. Wehn, and L. Benini. Energy Optimization in 3D MPSoCs with Wide-I/O DRAM Using Temperature Variation Aware Bank-Wise Refresh. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–4, March 2014.
- [27] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschweiler, and D. Atienza. 3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling. In *Proc. of ICCAD 2010*, 2010.