A New Bank Sensitive DRAMPower Model for Efficient Design Space Exploration

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

***Abstract*—In systems ranging from mobile devices to servers, *Dynamic Random Access Memories* (DRAM) have a large impact on performance and contribute a significant part to the total consumed power. Therefore, it is crucial to have an accurate DRAM power model for exhaustive design space explorations, which can handle different types of DRAM devices. In this paper, we present an improved version of the well known DRAMPower model. Our enhanced model is derived and calibrated from real measurements and outperforms pessimistic state-of-the-art DRAM power estimators like the widely used spread sheet provided by Micron.**

I. Introduction

Dynamic Random Access Memories (DRAM) contribute significantly to the power consumption of today's systems, as shown in Figure 1. Thus, there is a need for accurate DRAM power modeling. The most commonly used tool in research and industry is Micron's power calculator [1], which estimates the power from datasheet and workload specifications. However, this model is not accurate enough, as it assumes averaged workload characteristics. To overcome this limitation, we focus on an improved version, called DRAMPower [2], [3], which uses the actual timings instead of the minimal timings from datasheets.

In the *Design Space Exploration* (DSE) process of today's systems an important optimization goal is to decrease the total power consumption. Therefore, we modified DRAMPower that it can be used as a library [4], which can be easily integrated in a C++ based simulator like gem5 [5] or the DSE framework DRAMSys [4].

DRAMPower and its underlying equations depend on the average currents (I_{DD}) from DRAM vendor datasheets. These values are very pessimistic due to high process margins added by the vendors to ensure correct functionality under worst-case conditions and a high-enough yield [6].

To emphasize this fact, we conducted several I_{DD}-measurements with our measurement platform (cf. Section IV) on DDR3 devices from Samsung to show these high margins. Each test was repeated several times to ensure stability and reproducibility with a temperature range from 25°C to 90°C. The results plotted in Figure 2 show the expected worst-case behavior. The values measured at the highest temperatures for I_{DD0} and I_{DD4R} are around 18% to 20% lower than the datasheet current consumption definitions of the selected SO-DIMM. For the power-down current (I_{DD2P0}), which is leakage dominated, the difference is even larger and even at 90°C it still deviates 60% from the value defined in the datasheet. Consequently, the key attribute of a realistic and accurate DRAM power analysis is an enhanced power modeling tool, based on DRAMPower, that receives nominal current measures instead of worst-case values from the vendor's datasheets.

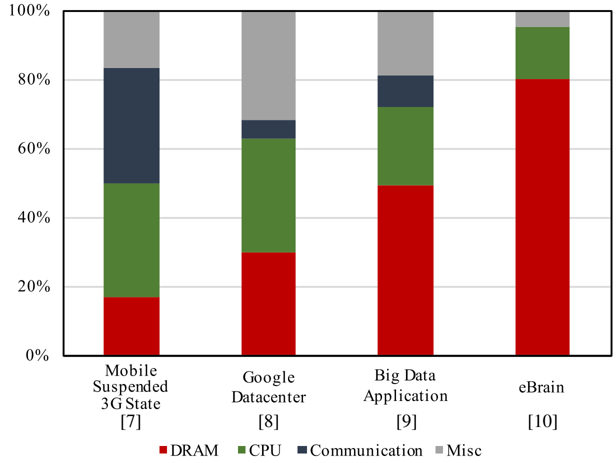


Fig. 1: DRAM Power Break Downs

In this paper, we present an integral modification to DRAMPower's underlying equations with respect to the active background energy, in order to realize a bank sensitive model. These enhancements are proven with conducted measurements on several DRAM devices from different vendors.

The paper is structured as follows: Section II explains the basics of DRAM power modeling and the underlying equations of DRAMPower are introduced. Our improvements to DRAMPower and new equations are presented in Section III. The used measurement setup is presented in Section IV. Section V shows experimental results obtained by measurements and simulations. Finally, the paper is concluded in Section VI.

II. DRAMPower Model

In this chapter we will elaborate the basic equations of the DRAMPower model, according to [2], [3] and [11]. As mentioned before, we find current values in vendor's datasheets denoted as specific I_{DD} values, as shown in Table I. For instance, I_{DD0}, is measured by an `ACT-PRE` (activate-precharge) sequence, that is issued to the device in the minimum timing interval ($t_{RC} = t_{RAS} + t_{RP}$).

As mentioned before, a well-known and often used model is provided by Micron in form of a spread sheet [1]. However, there are some limitations in that model. Micron uses the minimal timing constrains from the datasheet specifications instead of the actual timings. But in practice there are de-

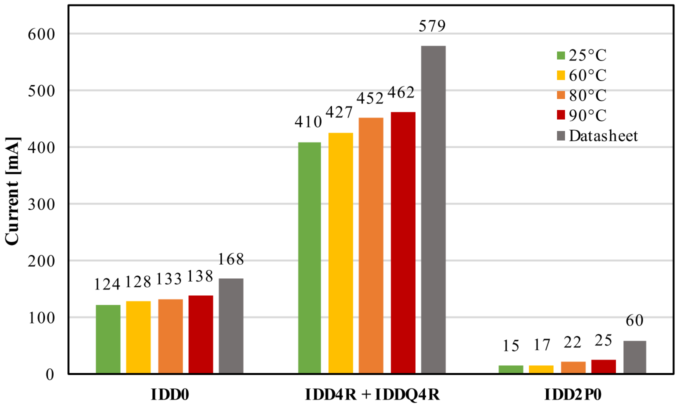


Fig. 2: Current Consumption Averaged over Ten 2 GByte DDR3 SO-DIMMs (Samsung)

pendencies 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 better model is DRAMPower, which uses the actual timings from the transactions. DRAMPower is widely used for instance in the gem5 simulator [5], the DRAM simulator Ramulator [12] and the SystemC/TLM based DRAM design space exploration framework DRAMSys [4], [13]. In this paper we further enhance this power model by conducting measurements on real DRAM devices and by developing a new underlying equation for the active background energy.

The interesting instances of time for the power model are the events in simulators like gem5 and DRAMSys. Therefore, we wrapped DRAMPower into a library that can be used in event based simulators [4]. These events contain all timing information that we need for the calculation of the power consumption, as of the DRAM with the following equations. In the following we will explain all the single contributions to DRAM power. For each contribution we calculate the energy and accumulate the parts to the total energy. By dividing the total energy with the runtime T_{Total}, we can obtain the average power consumption, shown in the end of this section.

In general, the electrical energy is:

$$E = \int_{t_0}^{t_1} v(t) \cdot i(t) \; dt \tag{1}$$

In case of an average value for $i(t) = I_{DDX}$ and a constant value for $v(t) = V_{DD}$ this equation is reduced to $E = V_{DD} \cdot I_{DDX} \cdot \Delta t_Y$, which is the base for all following energy calculations.

The power consumption consists of a background or static part and an active part which is needed for each command (ACT, PRE, RD, WR ...), called command power. In the following sections we present the underlying equations of DRAMPower for each case.

TABLE I: Key Parameters for DRAMs [14]

Name	Explanation
t_{RCD}	*Row to Column Delay*: The time interval between ACT and RD on the same bank.
t_{RP}	*Row Precharge*: The time interval that it takes for a DRAM array to be precharged (PRE) for another row access.
t_{RAS}	*Row Active*: The minimum active time for a row
t_{RC}	*Row Cycle*: The fastest time to ACT and PRE the same row.
t_{CL}	*CAS Latency*: The delay from a RD/WR command until data can be read/written at the interface.
t_{RFC}	*Refresh Cycle*: The time interval that it takes for a refresh
I_{DD0}	*One Bank Active Precharge Current*: Measured across ACT and PRE commands to one bank (other banks remain precharged).
I_{DD2N}	*Precharge Standby Current*: Measured when all banks are precharged (PRE).
I_{DD3N}	*Active Standby Current*: Measured when one bank is active (ACT).
I_{DD4R}	*Burst Read Current*: 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}	*Burst Write Current*: Similar to I_{DD4R}, instead with WR commands
I_{DDQ4R}	*Burst Read Interface Current*: Similar to I_{DD4R}, but measuring only I_{DDQ}
I_{DD2PO}	*Precharge Powerdown Current*: Measured during power-down mode
I_{DD5}	*Refresh Current*: Measured during refresh operation, with REF commands issued every t_{RFC}.
V_{DD}	*Supply Voltage*

A. DRAM Background Energy

While at least one bank is 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 we can easily derive by analyzing simulation events.

Figure 3 shows a sample command trace of a DRAM with four banks and their active and precharge states. The time the memory is in the active background mode is calculated by $T_{ACT}^{BG} = N_{ACT}^{BG} \cdot t_{CK}$ and for precharge background mode by $T_{PRE}^{BG} = N_{PRE}^{BG} \cdot t_{CK}$, respectively. N_{ACT}^{BG} and N_{PRE}^{BG} are the number of cycles in active and precharged state, while t_{CK} is the clock cycle period. This information can easily be obtained in simulators like gem5 and DRAMSys by using simple counter variables. The active and precharge background energy can be calculated with following equations:

$$E_{ACT}^{BG} = V_{DD} \cdot I_{DD3N} \cdot T_{ACT}^{BG} \tag{2}$$

$$E_{PRE}^{BG} = V_{DD} \cdot I_{DD2N} \cdot T_{PRE}^{BG} \tag{3}$$

However, Equation (2) is very pessimistic: The DRAM is in the active (open) state when at least one bank is activated, however, the used I_{DD3N} in the equation is the measured current when *all* banks are active. Therefore, we present an accurate equation that considers the exact number of active banks in Section III.

B. DRAM Command Energy

For each issued DRAM command a certain amount of energy is consumed. In the following parts we will explain the energy contributions for each command.

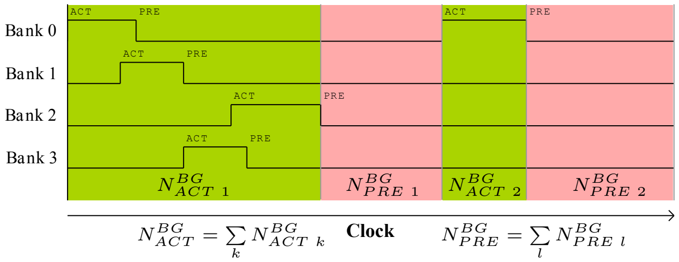


Fig. 3: Estimation of the Background Timings

1) Activate and Precharge: From I_{DD0}, which is specified as the average current consumed by the memory when it executes alternately both commands with minimum timing constraints, the energy of a single ACT or PRE command can be calculated. Since I_{DD0} also contains the background currents I_{DD2N} and I_{DD3N} with their respective minimum timing constrains, we have to subtract them from I_{DD0}. The energy of an ACT command is computed with the t_{RAS} (minimum time between ACT and PRE to the same bank) which is corresponding to the selected I_{DD0} speed-sort bin (datasheet). This is done similarly for the precharge command with the t_{RP} value.

$$E_{ACT} = V_{DD} \cdot (I_{DD0} - I_{DD3N}) \cdot t_{RAS} \tag{4}$$

$$E_{PRE} = V_{DD} \cdot (I_{DD0} - I_{DD2N}) \cdot t_{RP} \tag{5}$$

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 $\frac{BL}{2}$.

$$E_{RD} = V_{DD} \cdot (I_{DD4R} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \tag{6}$$

$$E_{WR} = V_{DD} \cdot (I_{DD4W} - I_{DD3N}) \cdot \frac{BL}{DR} \cdot t_{CK} \tag{7}$$

3) Refresh: The refresh operation is used to retain the data in the DRAM. Before issuing a refresh command all banks must be in precharged state. Therefore, usually a precharge all command (PREA) is issued. This is accounted by the second part of the E_{REF} equation, while the first part is the actual contribution of a refresh. N_{REF} stands for the number of refresh commands and N_{Banks} for the number of banks that have to be precharged by PREA.

$$E_{REF} = \Bigg[\underbrace{I_{DD5} \cdot (t_{RFC} - t_{RP}) + (I_{DD5} - I_{DD3N} + I_{DD2N}) \cdot t_{RP}}_{\text{Contribution from the refresh}} + \underbrace{N_{Banks} \cdot (I_{DD0} - I_{DD2N}) \cdot t_{RP} + I_{DD2N} \cdot t_{RP}}_{\text{Contribution from the precharge all command}}\Bigg] \cdot V_{DD} \tag{8}$$

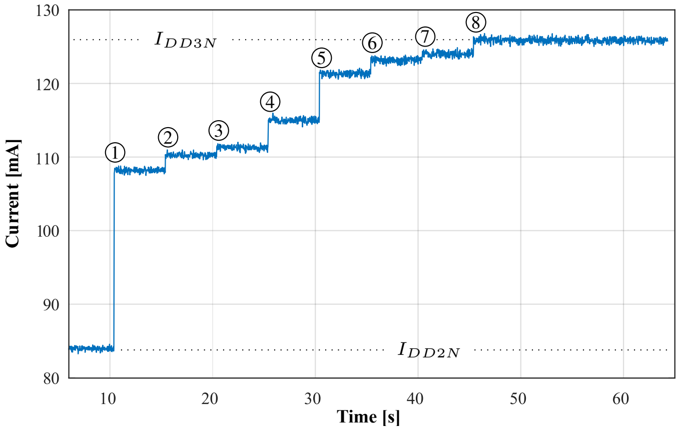


Fig. 4: Measured Samsung 2009 SO-DIMM with K4B1G1646E-HCF8 Devices (Refresh Disabled)

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 paper and therefore refer to [2] and [3].

5) Complete Trace Power: The computation of the average power consumption for the entire simulation is done by combining all the energy contributions from above into the following equation:

$$\begin{aligned} E_{Total} = {} & E^{BG}_{ACT} + E^{BG}_{PRE} \\ & + N_{REF} \cdot E_{REF} + N_{ACT} \cdot E_{ACT} \\ & + N_{PRE} \cdot E_{PRE} + N_{RD} \cdot E_{RD} \\ & + N_{WR} \cdot E_{WR} \end{aligned} \tag{9}$$

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.

As mentioned before, we can obtain the average power consumption by dividing the total energy with the total runtime:

$$P_{AVG} = \frac{E_{Total}}{T_{Total}} \tag{10}$$

III. Enhancements to DRAMPower

DRAMPower has been included in system level virtual platforms like gem5 [5], Ramulator [12] and DRAMSys [4], [13] in order to perform design space explorations. In this chapter, we present an integral modification that will lead to less pessimistic DRAM power simulation results, which therefore will also enhance the overall accuracy of the previously mentioned simulators.

To improve the calculation of the active background energy we derive in this section a new formula that will replace Equation (2). As explained in Section II-A the DRAM is in the precharged state if no bank is opened (I_{DD2N}) and it is in the active state if at least one bank is activated. However, for the active state the complete current I_{DD3N} is accounted, which represents the current when *all* banks are activated. However, as shown in Figure 4 this is not true in reality. This plot shows the current consumption over time where we activated each bank in the DRAM step by step (① - ⑧) and we clearly see that the active background current depends on the number of active banks.

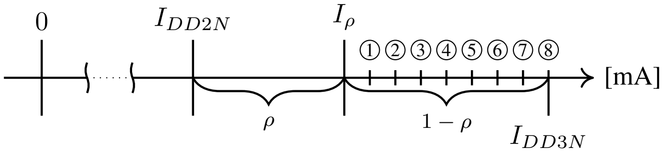


Fig. 5: Improved Background Energy Calculation

For the new equation that considers the exact number of activated banks, we introduce 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 of the banks. Figure 5 shows the currents of a DRAM with 8 banks on a number line. Each activated bank (① - ⑧) adds a specific current portion to I_ρ. When all banks are activated we measure I_{DD3N}.

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

$$E_{ACT}^{BG*}(b) = V_{DD} \cdot \frac{1}{B} \cdot (I_{DD3N} - I_\rho) \cdot T_{ACT}^{BG*}(b) \tag{11}$$

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:

$$E_{ACT}^{BG\ Shared} = V_{DD} \cdot I_\rho \cdot T_{ACT}^{BG} \tag{12}$$

The new total active background energy results in:

$$E_{ACT}^{BG} = E_{ACT}^{BG\ Shared} + \sum_{b=0}^{B-1} E_{ACT}^{BG*}(b) \tag{13}$$

As depicted in 5, the relationship between I_ρ, I_{DD2N} and I_{DD3N} can be expressed by a relative factor ρ, which is a vendor specific parameter:

$$\rho = \frac{I_\rho - I_{DD2N}}{I_{DD3N} - I_{DD2N}} \tag{14}$$

Thus the new total energy can be expressed as:

$$\begin{aligned} E_{ACT}^{BG} =& V_{DD} \cdot \Bigg[\big(\rho \cdot I_{DD3N} + (1-\rho) \cdot I_{DD2N}\big) \cdot T_{ACT}^{BG} \\ &+ \frac{1}{B} \cdot (1-\rho) \cdot (I_{DD3N} - I_{DD2N}) \cdot \sum_{b=0}^{B-1} T_{ACT}^{BG*}(b) \Bigg] \end{aligned} \tag{15}$$

In Section V we present measured values for ρ for different vendors and we show the influence of it to DRAM power simulation. If the value of ρ is not known it can be set to $\rho = 1$. This will transform Equation (15) to the old pessimistic Equation (2). In the following section we present our developed measurement setup.

IV. Measurement Setup

We developed a custom platform[1] 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. A peltier element (thermoelectric) is placed in the center, creating a hot and a cold side. The cold side is faced outwards, equipped with aluminum cooling ribs. The hot side is faced towards the DRAM module and equipped with a temperature sensor for automatic heat control. To ensure fast and even distribution of the temperature among all covered DRAM devices, the hot side is made of a solid copper plane. Heat conducting foil is used to build a thermal bridge between the copper surface and the exterior surfaces of the DRAM packages on a DIMM.

For analyzing the current consumption of DRAMs, we designed an adapter PCB for DDR3 SODIMMs (6 Layers) that conforms to JEDEC standard requirements (cf. Figure 6). The data, address and control lines are passed through, the power lines are routed across a precision 10mΩ shunt resistor. A shunt monitor is used to amplify the low voltage range of the shunt resistor.

The amplified voltage is forwarded to a computer-interfaced microcontroller board which also contains the power electronics for the heat control.

As user interface a custom PC software was designed for heat control and current analysis. The temperature control provides a range of 25 to 90°C. In advance, various thermal simulations have been conducted to construct the mechanical setup in a way to provide an accuracy of ±2°C among all covered devices. To guarantee reliable results the system has been analyzed with calibrated precision measurement instruments by Keithley. The inspection record certifies a current measurement accuracy of +0/-1mA.

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.

V. Experimental Results

In this section, we present measurement results for the vendor specific ρ parameter. Furthermore, we demonstrate how the new equation will result in a less pessimistic and more accurate power estimation by simulations with DRAMSys.

A. Measurements

With the setup presented in Section IV we measured several DRAM devices from different vendors. Figure 8 shows the results for Micron, Hynix, Samsung and Qimonda devices. We clearly observe that for Micron devices (Figure 8a) the parameter ρ is the highest, and therefore the resulting error by using Equation 2 has less influence. However, for e.g.

[1] Further details of the measurement platform will be published in a following publication and in a detailed technical report at http://www.uni-kl.de/3d-dram

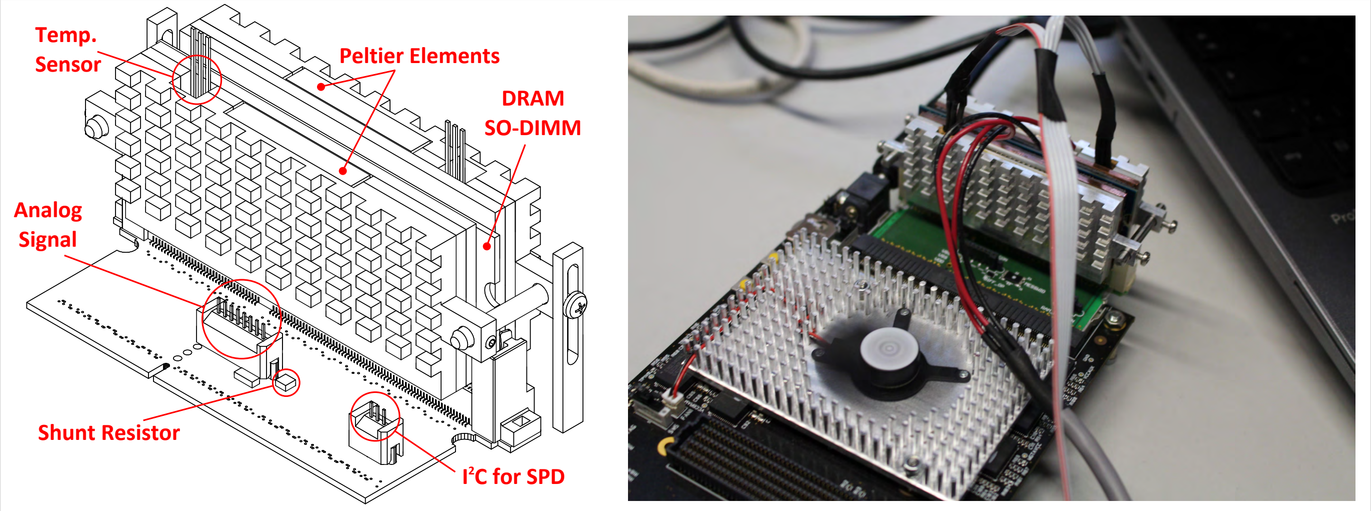


Fig. 6: Mechanical Construction and Adapter PCB Placed in a Typical Standard FPGA Board

Samsung devices (Figures 8c,8d) we see an intermediate value for ρ that has a large influence on the accuracy of the final power estimation. Furthermore, we can observe that the value of ρ decreases by scaling down of the DRAM devices. The DRAM in Figures 8c uses a 50 nm node, where the DRAM in Figure 8d uses a 30 nm node. An interesting fact for Qimonda devices is that the current drawn by the shared resources is extremely small. Figure 7 shows the variaton of the currents I_{DD2N}, I_ρ and I_{DD3N} for ten different DRAM SO-DIMMs (40 devices in total). The resulting ρ has a variation of 12%.

Although DRAMs are standardized by JEDEC, these results clearly show that the vendors have a large freedom of designing their internal device organization, which results in different values for the parameter ρ. In order to obtain a reliable and accurate power estimation for the used DRAM in the system running a specific application, this parameter should be considered. In the next section we prove this by means of example simulations.

B. Simulation

To show the impact of our new active background power model, we conducted simulations with DRAMSys and the modified DRAMPower. Therefore, we considered the Samsung K4B1G1646E-HCF8 device with $\rho = 0.5$. Figure 9 shows the power over time for a small simulation. In this simulation we issue read requests to specific addresses in the memory such that all the banks are opened step by step (① - ⑧). The figure shows an overlay of the old and new

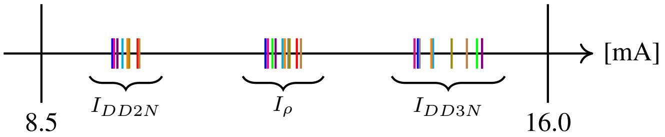
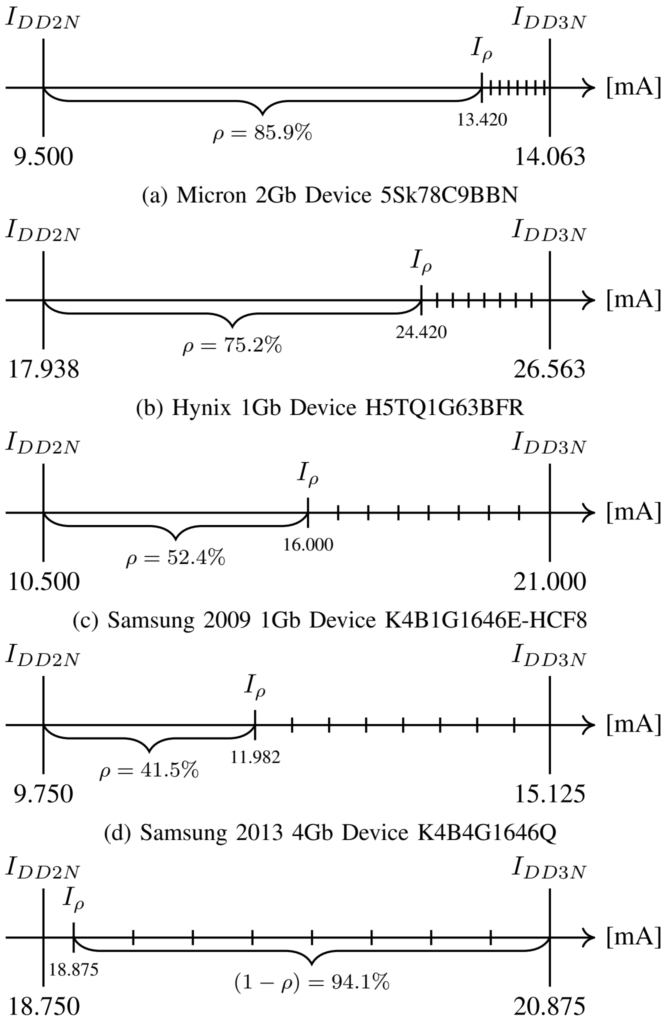


Fig. 7: Variation Among 10 Samsung 2GB SO-DIMMs (K4B4G1646Q)



(a) Micron 2Gb Device 5Sk78C9BBN

(b) Hynix 1Gb Device H5TQ1G63BFR

(c) Samsung 2009 1Gb Device K4B1G1646E-HCF8

(d) Samsung 2013 4Gb Device K4B4G1646Q

(e) Qimonda 1Gb Device IDSH1G-04A1F1C

Fig. 8: Measurements Active Background Energy

TABLE II: Simulation Results for a SO-DIMM

Benchmark	Mapping	Old [mW]	New [mW]	Deviation	Percentage of Time where n Banks are Simultaneously Active								
					No Banks	1 Bank	2 Banks	3 Banks	4 Banks	5 Banks	6 Banks	7 Banks	8 Banks
mediabench-jpegdecode	BRC	285.76	279.50	2%	**83.6%**	13.5%	2.5%	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%
	RBC	280.67	275.37	2%	**83.6%**	4.1%	8.1%	1.9%	1.5%	0.4%	0.2%	0.1%	0.1%
mediabench-jpegencode	BRC	354.34	321.08	9%	15.0%	**82.0%**	2.5%	0.5%	0.0%	0.0%	0.0%	0.0%	0.0%
	RBC	350.36	318.88	9%	14.4%	**57.9**%	21.5%	3.0%	1.6%	1.0%	0.3%	0.2%	0.1%
mediabench-epic	BRC	347.74	318.88	8%	26.4%	**71.9%**	1.3%	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%
	RBC	341.27	316.75	7%	24.2%	15.4%	**36.5%**	18.3%	2.9%	1.3%	0.6%	0.4%	0.3%
mediabench-unepic	BRC	435.57	405.61	7%	23.2%	**73.0%**	2.7%	1.2%	0.0%	0.0%	0.0%	0.0%	0.0%
	RBC	415.60	388.32	7%	15.7%	29.1%	**31.0%**	9.9%	9.0%	1.7%	2.0%	0.6%	1.0%
image-rotation	BRC	1088.71	1083.43	0.5%	1.6%	0.8%	1.7%	2.1%	2.7%	3.7%	7.6%	30.1%	**49.6%**
	RBC	1081.38	1075.88	0.5%	1.6%	1%	1.9%	2.1%	2.8%	4%	8.2%	28.8%	**49.5%**

simulation model. The pessimism of the old model can be clearly observed.

Furthermore, we conducted experiments with real workload benchmarks shown in Table II. Since the address mapping has a large influence on the power and the performance we simulated the usual *bank-row-column* (BRC) and *row-bank-column* (RBC) mappings [14]. The *jpegdecode* benchmark has a lot of idle periods where no banks are active and the DRAM is most of the time in the precharged state. Therefore, most of the time Equation (3) instead of Equations (15,2) is used to calculate the background energy. The *image-rotation* application uses often all banks in the DRAM in parallel such that most of the time nearly the complete I_{DD3N} is accounted.

However, many application use only 1-4 banks concurrently, which shows a significant improvement in power estimation, for example 9% for the *jpegencode* benchmark.

VI. Conclusion

In this paper, we presented a new bank sensitive version of DRAMPower, which considers the exact amount of activated banks for the calculation of the active background energy. Therefore, we introduced a new equation, which depends on a vendor specific parameter. We reveal the behaviour of this parameter by conducting experiments with a sophisticated simulation platform. These insights will help to provide an accurate DRAM power model, which can be used for thoroughly design space exploration of modern systems with DRAMs. Moreover, we will contribute our results to the official DRAMPower code repository.

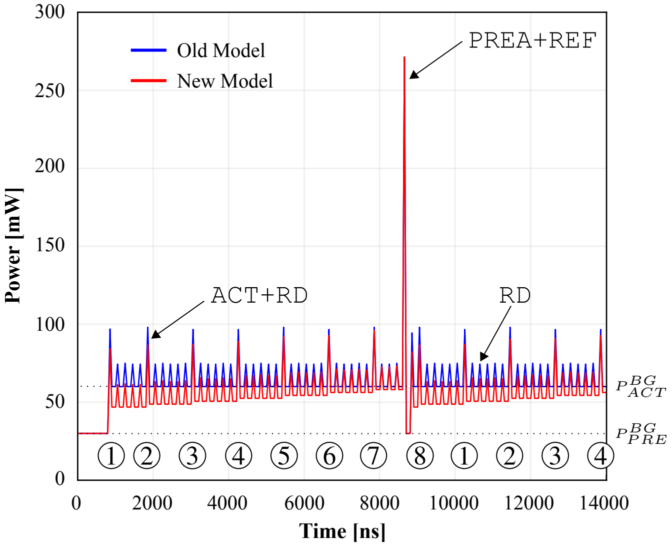


Fig. 9: Simulation Example for Samsung 2009

Acknowledgment

The authors thank Subash Kannoth, Felipe Prado, Martin Schultheis, Carl Rheinländer, Kira Kraft, Sven Goossens, Benny Åkesson, Karthik Chandrasekar, Radhika Jagtap and Stephan Diestelhorst for their support. This work was partially supported by the *German Research Foundation* (DFG) as the priority program *Dependable Embedded Systems SPP1500* (http://spp1500. itec.kit.edu), the *DFG* grant no. WE2442/10-1, the *Fraunhofer High Performance Center for Simulation- and Software-based Innovation* and the *Carl-Zeiss Stiftung*.

References

[1] Micron. *DDR3 SDRAM System Power Calculator,*. last access 2014-07-03, jul 2011.

[2] Karthik Chandrasekar, et al. *Improved Power Modeling of DDR SDRAMs*. In proc. DSD'11, 2011.

[3] Karthik Chandrasekar, et al. *DRAMPower: Open-source DRAM power & energy estimation tool*.

[4] Matthias Jung, et al. *DRAMSys: A flexible DRAM Subsystem Design Space Exploration Framework*. IPSJ Transactions on System LSI Design Methodology (T-SLDM), August 2015.

[5] Nathan Binkert, et al. *The gem5 simulator*. SIGARCH Comput. Archit. News, 39(2):1–7, August 2011.

[6] K. Chandrasekar, et al. *Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical Approach*. In Proc. 50th Design Automation Conference, Austin, USA, June 2013.

[7] Aaron Carroll et al. *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.

[8] Urs Hoelzle et al. *The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 1st edition, 2009.

[9] J. Krueger, et al. *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.

[10] N. Farahini, et al. *A scalable custom simulation machine for the Bayesian Confidence Propagation Neural Network model of the brain*. In Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, pages 578–585, Jan 2014.

[11] Matthias Jung, et al. *Power Modelling of 3D-Stacked Memories with TLM2.0 based Virtual Platforms*. In Synopsys User Group Conference (SNUG), May, 2013, Munich, Germany., 2013.

[12] Y. Kim, et al. *Ramulator: A Fast and Extensible DRAM Simulator*. IEEE Computer Architecture Letters, PP(99):1–1, 2015.

[13] Matthias Jung, et al. *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.

[14] B. Jacob, et al. *Memory Systems: Cache, DRAM, Disk*. Elsevier Science, 2010.