

# DRAM Power Model (DRAMPower)

buildpassin

coverage 84%

## 0. Releases

The last official release can be found here:

<https://github.com/ravenrd/DRAMPower/releases/tag/4.0>

The master branch of the repository should be regarded as the bleeding-edge version, which has all the latest features, but also all the latest bugs. Use at your own discretion.

## 1. Installation

Clone the repository, or download the zip file of the release you would like to use. The source code is available in src folder. src/cli/drampower.cc file gives the user interface, where the user can specify the memory to be employed and the command/transaction trace to be analyzed. To build, use:

```
make -j4
```

This command will download a set of trace files from <https://github.com/Sv3n/DRAMPowerTraces> which can be used as test input for the tool.

## 2. Required Packages

The tool was verified on Ubuntu 14.04 using:

- xerces-c (libxerces-c-dev) - v3.1 with Xerces development package
- gcc - v4.4.3

## 3. Directory Structure

- src/: contains the source code of the DRAMPower tool that covers the power model, the command scheduler and the trace analysis tool.
- memspecs/ : contains the memory specification XMLs, which give the architectural, timing and current/voltage details for different DRAM memories.
- traces/ : contains 4 sample DRAM transaction traces and 1 sample command trace (after the installation / compilation)
- test/ : contains test script and reference output

## 4. Trace Specification

## Command Traces

If the command-level interface is being used, a command trace can be logged in a file. An example is given in `traces/commands.trace`

The format it uses is: `<timestamp>,<command>,<bank>`. For example, "500,ACT,2", where ACT is the command and 2 is the bank. Timestamp is in clock cycles (cc), the list of supported commands is mentioned in `src/MemCommand.h` and the bank is the target bank number. For non-bank-specific commands, bank can be set to 0. Rank need not be specified. The timing correctness of the trace is not verified by the tool and is assumed to be accurate. However, warning messages are provided, to identify if the memory or bank state is inconsistent in the trace. A sample command trace is provided in the `traces/` folder.

## Transaction Traces

If the transaction-level interface is being used, a transaction trace can be logged.

The format it uses is: `<timestamp>,<transaction_type>,<address>`. For example, "35,READ,0x80028", where READ/WRITE can be the transaction type and the logical address (32-bits long and byte addressable) less than the maximum supported DRAM capacity of 4GB (32Gb).

The tool uses a flexible and efficient memory map as follows: specified in HEX (0x). Timestamp is in clock cycles (cc) and maximum `{row}-{bank}-{column}-{BI}-{BC}-{BGI}-{BL}`. Here, BI gives the degree of bank interleaving, BC gives the burst size (count), BGI gives the degree of bank group interleaving (for DDR4) and BL gives the burst length used by the device. Dual-Rank addressing is not yet supported. The BC and BL address bits are derived from the column address bits, whereas the BI and BGI bits are derived from the bank address bits.

Four sample MediaBench application transaction traces have been provided. The MediaBench applications include: (1) EPIC Encoder, (2) JPEG Encoder, (3) H263 Encoder and (4) MPEG2 Encoder. These applications were independently executed on the SimpleScalar simulator with a 16KB L1 D-cache, 16KB L1 I-cache, 128KB L2 cache and 64-byte cache line configuration. We filtered out the L2 cache misses meant for the DRAM and logged them as transaction traces. These can be used with our command scheduler to generate equivalent command traces for any DRAM memory specified.

## 5. Usage

`src/cli/drampower.cc` is the main interface file, which accepts user inputs to specify memory to be employed and the command or transaction trace to be analyzed. If the transaction trace (DRAM command scheduler) is being used, the users can specify the degree of bank interleaving required, the request size and the use of power-down or self-refresh options. Also, for DDR4 memories bank group interleaving can be specified. Dual-rank DRAMs are not yet supported by the command scheduler. Note: Speculative use of power-down or self-refresh modes will increase the trace length due to the power-up latencies of these power-saving modes.

To use DRAMPower at the command-level (command trace), after make, use the following:

```
./drampower -m <memory spec (ID)> -c <commands trace>
```

To use DRAMPower at the transaction-level (command scheduler), after make, use the following:

```
./drampower -m <memory spec (ID)> -t <transactions trace>
```

Additional options when using transactions trace [-t] include: \* [-i] <interleaving> \* [-s] <request size> \* [-g] <DDR4 bank group interleaving> \* [-p] <0 - No Power-Down, 1 - Power-Down, 2 - Self-Refresh>

Also, when using either the commands trace or the transactions trace, the user can optionally include IO and Termination power estimates (obtained from Micron's DRAM Power Calculator). To enable the same, the '-r' flag can be employed in command line.

If these options are not used, the default values assumed are: \* interleaving = 1 \* request size = burst length \* I/O width / 8 (in bytes) (from memory XMLs) \* power saving = No power-down \* bank group interleaving = 1 \* IO and termination = OFF (0) \* Burst size (count) of 1

## 6. Memory Specifications

36 sample memory specifications are given in the XMLs targeting DDR2/DDR3/DDR4, LPDDR/LPDDR2/LPDDR3 and WIDE IO DRAM devices. The memory specifications are based on 1Gb DDR2, 1Gb & 2Gb DDR3, 2Gb LPDDR/LPDDR2 and 4Gb DDR4/LPDDR3 Micron datasheets and the 256Mb Wide IO SDR specifications are based on JEDEC timing specifications and circuit-level IDD measurements by TU Kaiserslautern, in place of the as yet unavailable vendor datasheets. 4 of the memory specifications target dual-rank DDR3 DIMMs.

Note: The timing specifications in the XMLs are in clock cycles (cc). The current specifications for Reading and Writing do not include the I/O consumption. They are computed and included separately based on Micron Power Calculator. The IDD measures associated with different power supply sources of equal measure (VDD2, VDDCA and VDDQ) for LPDDR2, LPDDR3, DDR4 and WIDE IO memories have been added up together for simplicity, since it does not impact power computation accuracy. The current measures for dual-rank DIMMs reflect only the measures for the active rank. The default state of the idle rank is assumed to be the same as the complete memory state, for background power estimation. Accordingly, in all dual-rank memory specifications, IDD2P0 has been subtracted from the active currents and all background currents have been halved. They are also accounted for separately by the power model. Stacking multiple Wide IO DRAM dies can also be captured by the nbrOfRanks parameter.

## 7. Variation-aware Power And Energy Estimation

15 of the included datasheets reflect the impact of process-variations on DRAM currents for a

selection of DDR3 memories manufactured at 50nm process technology. These memories include: (1) MICRON\_128MB\_DDR3-1066\_8bit - revision G (2) MICRON\_128MB\_DDR3-1066\_16bit - revision G (3) MICRON\_128MB\_DDR3-1600\_8bit - revision G (4) MICRON\_256MB\_DDR3-1066\_8bit - revision D (5) MICRON\_256MB\_DDR3-1600\_16bit - revision D

The original vendor-provided datasheet current specifications are given in XMLs without suffixes such as `_mu`, `_2s` and `_3s`. XMLs including suffixes indicate that the current measures are either: (1) typical (`mu`), or (2) include +2 sigma variation (`2s`), or (3) include +3 sigma variation (`3s`). These measures are derived based on the Monte-Carlo analysis performed on our SPICE-based DRAM cross-section.

To include these XMLs in your simulations, simply use them as the target memory.

## 8. Example Usage

An example of using this tool is provided below. To compile the example, use the Makefile and make sure the Gcc and Xerces-c are installed. Then, run:

```
make -j4
```

This should show the following compilation message on the screen:

```
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/xmlparser/MemSpecParser.d -iquote src -o src/xmlparser/MemSpecParser.o -c
src/xmlparser/MemSpecParser.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/xmlparser/XMLHandler.d -iquote src -o src/xmlparser/XMLHandler.o -c
src/xmlparser/XMLHandler.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/xmlparser/XMLParser.d -iquote src -o src/xmlparser/XMLParser.o -c
src/xmlparser/XMLParser.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/CommandScheduler.d -iquote src -o src/CommandScheduler.o -c src/CommandScheduler.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/CommandAnalysis.d -iquote src -o src/CommandAnalysis.o -c
src/CommandAnalysis.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/MemArchitectureSpec.d -iquote src -o src/MemArchitectureSpec.o -c
src/MemArchitectureSpec.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF src/MemCommand.d
-iquote src -o src/MemCommand.o -c src/MemCommand.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
```

```

src/MemoryPowerModel.d -iquote src -o src/MemoryPowerModel.o -c
src/MemoryPowerModel.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/MemorySpecification.d -iquote src -o src/MemorySpecification.o -c
src/MemorySpecification.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/MemPowerSpec.d -iquote src -o src/MemPowerSpec.o -c src/MemPowerSpec.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/MemTimingSpec.d -iquote src -o src/MemTimingSpec.o -c src/MemTimingSpec.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF src/Parameter.d
-iquote src -o src/Parameter.o -c src/Parameter.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/Parametrisable.d -iquote src -o src/Parametrisable.o -c src/Parametrisable.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/TraceParser.d -iquote src -o src/TraceParser.o -c src/TraceParser.cc
g++ -O -W -pedantic-errors -Wextra -Werror -Wformat -Wformat-nonliteral
-Wpointer-arith -Wcast-align -Wconversion -g -std=c++98 -MMD -MF
src/libdrampower/LibDRAMPower.d -iquote src -o src/libdrampower/LibDRAMPower.o -c
src/libdrampower/LibDRAMPower.cc
ar -c -cvr src/libdrampowerxml.a src/xmlparser/MemSpecParser.o
src/xmlparser/XMLHandler.o src/xmlparser/XMLParser.o
a - src/xmlparser/MemSpecParser.o
a - src/xmlparser/XMLHandler.o
a - src/xmlparser/XMLParser.o
g++ -Wall -o drampower src/xmlparser/MemSpecParser.o src/xmlparser/XMLHandler.o
src/xmlparser/XMLParser.o src/CmdScheduler.o src/CommandAnalysis.o
src/MemArchitectureSpec.o src/MemCommand.o src/MemoryPowerModel.o
src/MemorySpecification.o src/MemPowerSpec.o src/MemTimingSpec.o src/Parameter.o
src/Parametrisable.o src/TraceParser.o -L/usr/lib -lXerces-c
ar -c -cvr src/libdrampower.a src/CmdScheduler.o src/CommandAnalysis.o
src/MemArchitectureSpec.o src/MemCommand.o src/MemoryPowerModel.o
src/MemorySpecification.o src/MemPowerSpec.o src/MemTimingSpec.o src/Parameter.o
src/Parametrisable.o src/TraceParser.o src/libdrampower/LibDRAMPower.o
a - src/CmdScheduler.o
a - src/CommandAnalysis.o
a - src/MemArchitectureSpec.o
a - src/MemCommand.o
a - src/MemoryPowerModel.o
a - src/MemorySpecification.o
a - src/MemPowerSpec.o
a - src/MemTimingSpec.o
a - src/Parameter.o
a - src/Parametrisable.o
a - src/TraceParser.o
a - src/libdrampower/LibDRAMPower.o

```

After this, run with the command trace or the transaction trace, as described before:

```
./drampower -m memspecs/MICRON_1Gb_DDR3-1066_8bit_G.xml -t
```

```
traces/mediabench-epic.trace -r
```

The output should be something like this:

```
* Parsing memspecs/MICRON_1Gb_DDR3-1066_8bit_G.xml
* Analysis start time: Thu Nov 14 01:44:24 2013
* Analyzing the input trace
* Analysis End Time: Thu Nov 14 01:44:26 2013
* Power Computation Start time: Thu Nov 14 01:44:26 2013
* Trace Details:
Number of Activates: 96984
Number of Reads: 67179
Number of Writes: 29805
Number of Precharges: 96984
Number of Refreshes: 13168
Number of Active Cycles: 2519793
    Number of Active Idle Cycles: 196851
    Number of Active Power-Up Cycles: 0
        Number of Auto-Refresh Active cycles during Self-Refresh Power-Up: 0
Number of Precharged Cycles: 52261474
    Number of Precharged Idle Cycles: 51649664
    Number of Precharged Power-Up Cycles: 0
        Number of Auto-Refresh Precharged cycles during Self-Refresh Power-Up: 0
    Number of Self-Refresh Power-Up Cycles: 0
Total Idle Cycles (Active + Precharged): 51846515
Number of Power-Downs: 0
    Number of Active Fast-exit Power-Downs: 0
    Number of Active Slow-exit Power-Downs: 0
    Number of Precharged Fast-exit Power-Downs: 0
    Number of Precharged Slow-exit Power-Downs: 0
Number of Power-Down Cycles: 0
    Number of Active Fast-exit Power-Down Cycles: 0
    Number of Active Slow-exit Power-Down Cycles: 0
        Number of Auto-Refresh Active cycles during Self-Refresh: 0
    Number of Precharged Fast-exit Power-Down Cycles: 0
    Number of Precharged Slow-exit Power-Down Cycles: 0
        Number of Auto-Refresh Precharged cycles during Self-Refresh: 0
Number of Auto-Refresh Cycles: 776912
Number of Self-Refreshes: 0
Number of Self-Refresh Cycles: 0
-----
Total Trace Length (clock cycles): 54781267
-----

* Trace Power and Energy Estimates:
ACT Cmd Energy: 109175234.52 pJ
PRE Cmd Energy: 47764165.10 pJ
RD Cmd Energy: 49155365.85 pJ
WR Cmd Energy: 23486116.32 pJ
RD I/O Energy: 22249684.80 pJ
WR Termination Energy: 50549280.00 pJ
ACT Stdbby Energy: 283653996.25 pJ
    Active Idle Energy: 22159587.24 pJ
    Active Power-Up Energy: 0.00 pJ
        Active Stdbby Energy during Auto-Refresh cycles in Self-Refresh Power-Up: 0.00
```

```

pJ
PRE Stdbby Energy: 5147706163.23 pJ
  Precharge Idle Energy: 5087443452.16 pJ
  Precharged Power-Up Energy: 0.00 pJ
    Precharge Stdbby Energy during Auto-Refresh cycles in Self-Refresh Power-Up:
0.00 pJ
  Self-Refresh Power-Up Energy: 0.00 pJ
Total Idle Energy (Active + Precharged): 5109603039.40 pJ
Total Power-Down Energy: 0.00 pJ
  Fast-Exit Active Power-Down Energy: 0.00 pJ
  Slow-Exit Active Power-Down Energy: 0.00 pJ
    Slow-Exit Active Power-Down Energy during Auto-Refresh cycles in Self-Refresh:
0.00 pJ
  Fast-Exit Precharged Power-Down Energy: 0.00 pJ
  Slow-Exit Precharged Power-Down Energy: 0.00 pJ
    Slow-Exit Precharged Power-Down Energy during Auto-Refresh cycles in
Self-Refresh: 0.00 pJ
Auto-Refresh Energy: 262371782.36 pJ
Self-Refresh Energy: 0.00 pJ
-----
Total Trace Energy: 5996111788.44 pJ
Average Power: 58.34 mW
-----
* Power Computation End time: Thu Nov 14 01:44:27 2013
* Total Simulation time: 3.51 seconds
*

```

As can be noticed, the tool performs DRAM command scheduling and reports the number of activates, precharges, reads, writes, refreshes, power-downs and self-refreshes besides the number of clock cycles spent in the active and precharged states, in the power-down (fast/slow-exit) and self-refresh states and in the idle mode. It also reports the energy consumption of these components, besides the IO and Termination components in pJ (pico Joules) and the average power consumption of the trace in mW. It also reports the simulation start/end times and the total simulation time in seconds.

## 9. DRAMPower Library

The DRAMPower tool has an additional feature and can be used as a library. In order to use the library run "make lib", include `src/libdrampower/LibDRAMPower.h` in your project and link the file `src/libdrampower.a` with your project. An example for the usage of the library can be found in the folder `test/libdrampowertest/lib_test.cc`

## 10. Authors & Acknowledgment

The tool is based on the DRAM power model developed jointly by the Computer Engineering Research Group at TU Delft and the Electronic Systems Group at TU Eindhoven and verified by the Microelectronic System Design Research Group at TU Kaiserslautern with equivalent circuit-level simulations. This tool has been developed by Karthik Chandrasekar with Yonghui Li under the supervision of Dr. Benny Akesson and Prof. Kees Goossens. The IO and Termination Power measures have been employed from Micron's DRAM Power Calculator. If you decide to use

DRAMPower in your research, please cite one of the following references:

To cite the DRAMPower Tool:

```
[1] "DRAMPower: Open-source DRAM power & energy estimation tool"  
Karthik Chandrasekar, Christian Weis, Yonghui Li, Benny Akesson, Norbert Wehn, and  
Kees Goossens  
URL: http://www.drampower.info
```

To cite the DRAM power model:

```
[2] "Improved Power Modeling of DDR SDRAMs"  
Karthik Chandrasekar, Benny Akesson, and Kees Goossens  
In Proc. 14th Euromicro Conference on Digital System Design (DSD), 2011
```

To cite the 3D-DRAM power model:

```
[3] "System and Circuit Level Power Modeling of Energy-Efficient 3D-Stacked Wide  
I/O DRAMs"  
Karthik Chandrasekar, Christian Weis, Benny Akesson, Norbert Wehn, and Kees  
Goossens  
In Proc. Design, Automation and Test in Europe (DATE), 2013
```

To cite variation-aware DRAM power estimation:

```
[4] "Towards Variation-Aware System-Level Power Estimation of DRAMs: An Empirical  
Approach"  
Karthik Chandrasekar, Christian Weis, Benny Akesson, Norbert Wehn, and Kees  
Goossens  
In Proc. Design Automation Conference (DAC), 2013
```

## 11. Contact Information

Further questions about the tool and the power model can be directed to:

Karthik Chandrasekar (k.chandrasekar@tudelft.nl)

Feel free to ask for updates to the tool's features and please do report any bugs and errors you encounter. This will encourage us to continuously improve the tool.

## Disclaimer

The tool does not check the timing accuracy of the user's memory command trace and the use of commands and memory modes. It is expected that the user employs a valid trace generated using a DRAM memory controller or simulator, which satisfies all memory timing constraints and other



requirements. The user DOES NOT get ANY WARRANTIES when using this tool. This software is released under the BSD 3-Clause License. By using this software, the user implicitly agrees to the licensing terms.