DRAMPower 3.1
Software Development Manual

Karthik Chandrasekar[1], Christian Weis[2], Yonghui Li[3], Benny Akesson[3], Norbert Wehn[2], Kees Goossens[3]
[1]Computer Engineering, TU Delft, The Netherlands
[2]Microelectronic Systems Design, TU Kaiserslautern, Germany
[3]Electronic Systems Group, TU Eindhoven, The Netherlands

This software development manual is intended for users interested in modifying DRAMPower's source code.

USER INTERFACE FILES

These include the files that may be modified by the user to analyze a given trace for a given memory.

memspecs/xmls: These XMLs give the format for specifying the the architectural, timing and current/voltage details for different DDR2/DDR3/DDR4, LPDDR/LPDDR2/LPDDR3 and WIDE IO DRAM memories that can be used by the power estimation tool. 36 sample memory specifications are provided in the XMLs for the target device generations. 4 of the memory specifications target dual-rank DDR3 DIMMs. In these 4 specifications, the current measures reflect only the active rank. Amongst the 36 specifications, 15 XMLs reflect the impact of process-variations on DRAM currents are provided for specific DDR3 devices manufactured using 50nm process technology. They are based on 1Gb DDR2, 1Gb and 2Gb DDR3, 2Gb LPDDR/LPDDR2 and 4Gb LPDDR3/DDR4 Micron datasheets and the sample 256Mb Wide IO SDR specifications are based on JEDEC timing specifications and circuit-level IDD current measurements by TU Kaiserslautern, in place of the as yet unavailable commercial vendor datasheets. Note: The timing specifications in the XMLs are in clock cycles (cc). 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, since it does not impact energy/power computation accuracy.

Trace Files: These files give traces of explicit memory commands or memory transactions to be sent to the DRAM. If a command trace is used, it should include for every explicit command, in a separate line, the command issue timestamp (in clock cycles), the command type (in caps) and the target bank. The accuracy of the issuance of commands and usage of power-down and self-refresh modes is not checked by this tool and it is assumed that the user employs an accurate memory command trace that satisfies all memory timing constraints and requirements. The list of supported commands are provided in src/MemCommand.h.

If a transaction trace is used instead, it should include for every transaction, in a seperate line, the transaction timestamp (in cc), the transaction type (READ/WRITE) and the logical memory address (32-bits) generated by the requestor in HEX (0x).

The DRAM is byte-addressable and uses a flexible and efficient memory map as follows:

{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 and BL gives the burst length used by the device. The BC and BL address bits are derived from the column address bits, whereas the BI and BGI address bits are derived from the bank address bits. The DRAM command scheduler does not support dual-rank DRAMs yet. Only the power estimation component has been updated to support them.

PowerCalc.cc is the main interface file, where the user specify memory specification file and the command or transaction trace file to be analyzed using command-line options. If the transaction traces are used (invoking the command scheduler), the user can additionally specify the request size, the degree of bank interleaving and the use of power-down or self-refresh modes. For DDR4 memories, the user can also specify the degree of bank group interleaving.

Also, the user can optionally include IO and Termination power estimates (obtained from Micron's DRAM Power Calculator). It also reports the simulation times.

SOURCE CODE

These include the files that cover the power model, command scheduler and trace analysis tool.

CmdScheduler.cc is used to translate a memory transaction trace into a schedule of DRAM commands that can be sent to the memory and used for accurate power and energy estimation. This DRAM command scheduler dynamically schedules DRAM commands as if it were a memory controller. The scheduler assumes closed-page policy, employs ASAP scheduling for DRAM commands (i.e. schedules commands as soon as

timing constraints are met), performs FCFS scheduling of DRAM transactions and supports all the different DRAM generations supported by the power model.

The scheduler supports different request sizes and degrees of bank interleaving and bank group interleaving (for DDR4). It uses the memory map described above to translate logical addresses to physical addresses. The generated DRAM command schedule is also analyzable for real-time applications. Users can also select speculative usage of power-down or self-refresh modes (if needed) for idle periods between transactions.

CommandAnalysis.cc is used to analyse a given memory command trace and identify the number of activates, (auto) precharges, reads, writes, refreshes, power-downs, self-refreshes, precharged and active cycles and clock cycles in power-down and self-refresh modes. Additionally, it also derives the number of cycles the memory is idle in the active and the precharged modes. To identify the same, it includes functionalities to check for completion of activate, precharge, read, write and refresh operations, and determines if a particular clock cycle is idle or not.

The command analysis tool also has functionalities to take into account different timing constraints specific to a particular DRAM generation. New warning messages have been provided, to identify if the memory or bank state is inconsistent in the trace.

Furthermore, this analysis code triggers the evaluation of the trace only during memory state transitions (between active, precharged, active and precharged power-down, refresh, self-refresh and power-up states) and not on every clock cycle. This optimization has resulted in very high speed-ups in the power estimation simulations using DRAMPower, without affecting the accuracy of the trace analysis or the reported power and energy estimates.

MemCommand.cc is a class to identify the commands employed by the memory controller to perform different memory operations. These include *ACT*, *PRE*, *RD*, *WR*, *REF*, *RDA*, *WRA* and *PREA* commands (as specified by JEDEC) to cover the basic operations. The *RDA* and the *WRA* commands are used to indicate auto-precharge operations associated with a given *RD* or *WR* command. The *PREA* command is used to precharge all banks with a single command.

In addition to these JEDEC-specified commands, in this tool we have added a few special commands to facilitate and represent entry into and exit from power-down and self-refresh modes. These include *PDN_F_PRE*, *PDN_S_PRE*, *PDN_F_ACT* and *PDN_S_ACT* commands to represent fast-exit and slow-exit, active and precharged power-down mode entry commands and *PUP_PRE* and *PUP_ACT* to represent precharged and active power-down exit (power-up) commands.

Similarly, *SREN* command has been introduced to represent Self-Refresh entry and *SREX* command to represent Self-refresh exit. This class also includes functionalities to derive command type, bank and precharge information.

MemorySpecification.cc describes the type of memories addressed by this power model, with functionalities to parse memory specifications from XMLs.

MemArchitectureSpec.cc is a class to identify the different architectural parameters associated with a given memory. This includes burst length, number of banks, and the data rate (DDR/SDR) of the memory.

MemPowerSpec.cc is a class to identify the different JEDEC-specified memory current components. LPDDR/LPDDR2LPDDR3, DDR4 and WIDE IO DRAM memories support additional voltage domains, viz., VDD_2, VDD_{CA} and VDD_Q. However, all three are tied to the same power supply measure and can hence, be added up and accounted for together.

MemTimingSpec.cc is a class to identify the memory timing parameters for different operations. For DDR2/DDR3/DDR4 devices, which include DLLs (delay locked loops), powering up from power-down and self-refresh modes takes longer due to DLL reset. As a result, they use tXPDLL and tXSDLL timing constraints to power-back up from these modes.

MemoryPowerModel.cc includes the power model equations to estimate power and energy consumption. DRAMPower supports most DRAM operations like read, write, refresh, activate and precharge, and power-saving modes such as power-down and self-refresh, besides taking into consideration the use of multiple voltage domains in LPDDR/LPDDR2/LPDDR3, DDR4 and WIDE IO DRAM devices.

Optionally, it can also include IO and Termination power estimates (obtained from Micron's DRAM Power Calculator). This version also support power estimation of dual-rank DIMMs. The current measures for dual-rank DRAMs should only reflect those for the active rank and not the idle rank. The default state of the idle rank is assumed to be the same as the current memory state, for background power estimation. Hence, rank information is not required in the command trace.

For a given DRAM command trace, it reports the different energy components and total energy and average power consumption of the trace. In addition, it also reports the energy consumed, when the memory is idle in the active or the precharged state.

Utils.h contains utility functions to convert to and from a string.