

Low-Power FB-DIMMs

Power Consumption Comparisons LP FB-DIMMs vs. Standard FB-DIMMs and RDIMMs

Power Consumption Comparisons LP FB-DIMMs vs. Standard FB-DIMMs and RDIMMs

Low-power FB-DIMMs are FB-DIMMs that have been optimized at the "standard voltage" that still achieve power savings. The LP FB-DIMM that Dataram produces is a DIMM that runs on the standard 1.8VDD and 1.5VCC supplies which achieves power savings through other design changes and optimizations. Component selection and enforcing maximum power thresholds have been shown to decrease power consumption considerably. Under best conditions the low power 2GB FB-DIMM uses 45% less power than the standard power 2GB FB-DIMM. The low power 4GB FB-DIMM could potentially use 21% less power than the standard 4GB FB-DIMM. Note: All comparisons were done between Dataram modules only.

The table below is a power-consumption comparison that defines typical values that our LP FB-DIMMs display vs. our standard power FB-DIMMs and our standard RDIMMs.

Table 1 – All values are per one assembly. IDD/ICC measurements were made during a March X Test.

Assembly	Product	Assembly Configuration	IDD (ma)	ICC (ma) Modules with AMBs only.	Total Power (W)
65518	DRH667FBL/4GB = 2 X 65518	LP ⁽¹⁾ 2GB 2R x8 PC2-5300F	2702	1530	7.16
65508	DRH667FB/4GB = 2 X 65508	2GB 2R x4 PC2-5300F	4515	1512	10.40
63332	DTM63332 = 1 x 63332	2GB 2R x4 PC2-5300P	4824		8.68
65519	DRH667FBL/8GB = 2 X 65519	LP ⁽¹⁾ 4GB 2R x4 PC2-5300F	4043	1552	9.61
65510	DRH667FB/8GB = 2 X 65510	4GB 2R x4 PC2-5300F	5235	1458	11.61
63356	DTM63356 = 1 x 63356	4GB 2R x4 PC2-5300P	5169		9.30

Notes:

1. LP is "Low-Power".
2. The contributing factor for the delta between the FB-DIMM and the RDIMM is the ICC current that is a characteristic of the AMB at 1.5VCC.

3. March X Test Description: The test detects SAFs (Stuck At Faults), TFs (Transition Faults) and AFs (Address Faults).

Fault Coverage: SAFs, CFs and TFs in the memory cell array and AFs in the address decoder.

Pattern Size = $6n$

Algorithm: MarchX

Write 0's to all memory
Read 0's Write 1's = (Ascending)
Read 1's Write 0's = (Descending)
Read 0's = (Ascending)