HP Z800 Memory Configuration and Optimization

Introduction: The purpose of this document is to provide an overview of the memory configuration for the Z800 Workstation and to provide recommendations to optimize performance.

Supported DIMMs and Configuration Rules

- The HP Z800 supports Unbuffered 1GB single rank and 2GB dual rank PC3-10600E 1333MHZ ECC DIMMs
- The HP Z800 supports Registered 4GB and 8GB dual rank PC3-10600R and PC3-10600P 1333MHz DIMMs
- The HP Z800 supports Registered 16GB (2H 2009) quad rank PC3-8500R and PC3-8500P 1067MHz DIMMs
- Address Parity checking is supported on Registered DIMMs
- Unbuffered and Registered DIMMs cannot be mixed in a system
- In dual processor configurations, each processor must have memory connected to it
- For any of the black and white pair of memory connectors, if two DIMMs are loaded and one of these DIMMs is quad rank, all the memory in the system will run at 800MHz.
- The CPUs determine the speed at which the memory is clocked. Example: if a 1067MHz capable CPU is included in the system, the maximum speed the memory will run at is 1067MHz regardless of the specified speed of the memory

Best Performance Optimization Tips

Since the memory controller is based on three channel design, the following rules should be used for best performance:

- For single processor configurations, configure memory in sets of 3
- For dual processor configurations, configure memory in sets of 6
- Configure memory using the smallest DIMM size will sometimes optimize memory performance, if it prevents single channel configurations Example: for a 8GB single processor configuration, buy 3 2GB DIMMS and 2 1GB, not 4 2GB DIMMs

Figure 1 Optimal Configuration for the HP Z800 (Note: The following tables do not include all available CTO configurations)

3	ingle Proc	cessor Con	riguration							
	Notes	DIMM1	DIMM2	DIMM3	CPUO DIMM4	DIMM5	DIMM6	Rating	Key:	Goo
1GB	~	1GB						\bigcirc	V	Bette
2GB		1GB		1GB				\bigcirc	\bigcirc	Best
3GB	*	1GB		1GB		1GB		\bigcirc		
4GB	*~	1GB	1GB	1GB		1GB		Ō		
6GB	~	1GB 2GB	1GB	1GB 2GB	1GB	1GB 2GB	1GB	8		
8GB	~	2GB 2GB	2GB 1GB	2GB 2GB	1GB	2GB 2GB		8		
12GB	~ ~	2GB 4GB	2GB	2GB 4GB	2GB	2GB 4GB	2GB	8		
16GB	~	4GB	4GB	4GB		4GB		Ŏ		
24GB	~~~~	4GB 8GB	4GB	4GB 8GB	4GB	4GB 8GB	4GB	8		
36GB	~	8GB	4GB	8GB	4GB	8GB	4GB	Ŏ		
48GB	~ †~	8GB 16GB	8GB	8GB 16GB	8GB	8GB 16GB	8GB	8		
60GB	‡~	16GB	4GB	16GB	4GB	16GB	4GB	Ŏ		
72GB	‡~	16GB	8GB	16GB	8GB	16GB	8GB			
96GB	‡~	16GB	16GB	16GB	16GB	16GB	16GB			

* For 32 bit operating systems, it is optimal to only load 3GB of memory because it gives the optimal performance in 3 channel design and, because the 4th GB isn't fully accessible by the operating system, it optimizes the cost per usable memory.

~ Although supported, these configurations are not CTO at this time.

† The speed of the memory will be 1067MHz

‡ The speed of the memory will be 800MHz

					CPUO					CPU1				
	Notes	DIMM1	DIMM2	DIMM3	DIMM4	DIMM5	DIMM6	DIMM1	DIMM2	DIMM3	DIMM4	DIMM5	DIMM6	Rating
6GB		1GB		1GB		1GB		1GB		1GB		1GB		\bigcirc
8GB		1GB	1GB	1GB		1GB		1GB	1GB	1GB		1GB		\bigcirc
12GB	~	1GB 2GB	1GB	8										
16GB	~	2GB	1GB	2GB	1GB	2GB		2GB	1GB	2GB	1GB	2GB		Ŏ
18GB	~	2GB	1GB	Ō										
24GB	~	2GB 4GB	2GB	8										
48GB	~	4GB 8GB	4GB	8										
72GB	~	8GB	4GB	Ō										
96GB †	~ †~	8GB 16GB	8GB	8										
120GB‡	‡~	16GB	4GB	Ō										
144GB‡	‡~	16GB	8GB											
192GB‡	‡~	16GB	16GB											

Dual Processor Configurations

Loading Order

When loading the system memory, for single processor configuration, start in the slot furthest from the CPU, DIMM1 and move towards the CPU, first filling the black slots and then filling the white slots. For dual processor configuration, load the memory as above, alternating between the two processors. See illustration below.

Figure 2	products and sor constituting a	Memory Load Order - Single Proc	essor rroi	su roi Memory Load Order – Dual Processor

There is a BIOS setting to change between 2 different types of NUMA and Non-NUMA for dual processor systems. See Memory Architecture and Efficiency White Paper for more information.

