Abstract In this chapter we discuss the Element Interconnect Bus (EIB) of the Cell Broadband Engine. We present the design of the EIB as a token-ring, managed communication path for transferring command and data amongst the CBEA components and IO. The EIB supports full memory-coherent and SMP operation. The EIB consists of four 16-byte-wide data rings. Each ring transfers 1 PPE cache line at a time. Each element connected to the EIB has an on-ramp, and an off-ramp, both of which can be active. We provide a traffic-circle analogy with token collection to explain the workings of the EIB which is an important variable in deciding how to partition the problem on the SPEs. Experimental data on EIB performance is also presented in this chapter.

4.1 Introduction to the Element Interconnect Bus

The EIB in the Cell/B.E. processor allows for communication among the PPE, the SPEs, the off-chip memory, and the external I/O. It uses the direct memory access (DMA) paradigm to communicate between any source-destination pairs from the above list. As we have already from Chapter 1, the 3.2 GHz Cell in the PS3 has a theoretical peak performance of 204.8 Gflop/s (single precision) and 14.6 Gflop/s (double precision). To utilize such a high compute rate, the memory latency and inter-processor communication latency has to be very low. The EIB is designed with the sole purpose of providing efficient and low latency data communication in the Cell processor. The EIB has a peak bandwidth of 204.8 Gbytes/s for intrachip data transfers among the PPE, the SPEs, and the memory and I/O interface controllers.

By design, the SPUs can directly access only its local store, every other request to get or put data on the main memory is performed by the MFC which has DMA controllers to perform DMA data transfer operation to/from local store to main memory. The architecture of the SPU and DMA controller has to be understood in conjunction as the design choices made by the architects influence good program design, and data transfer methodologies. When the SPU program is operating in the sweet-
spot of the architecture design, the computation and data-transfers don’t interfere with each other, and we can expect to see high computational rates. In the sequel of this book we shall uncover a number of DMA factoids and best practices which have worked for me. This chapter provides some of the rationale for these practices.

These DMA engines also allow direct transfers between the local stores of two SPU's for pipeline mode of operation. We have used the pipeline mode to perform compress-encrypt on the SPU's in a pipeline where data is compressed, and then sent off to the next SPU for encryption.

The EIB can be thought of as the network in the Cell network-on-chip (NOC) processor. It is similar to a high-speed token-ring managed bus with address snooping. The EIB consists of one address bus and four 16-byte-wide data rings, two of which run clockwise and the other two counterclockwise. Each ring can allow up to three concurrent data transfers as long as their paths do not overlap. The EIB operates at half the speed of the processor (1.6 GHz for the PS3).

Each requester on the EIB starts with a small number of initial command credits to send out requests on the bus. The number of credits is the size of the command buffer inside the EIB for that particular requester. One command credit is used for each request on the bus. When a slot becomes open in the command buffer as a previous request progresses in the EIB request pipeline, the EIB returns the credit to the requester. When a unit requires access to a data ring in order to send data to another unit, it makes a single request to the data ring arbiter on the EIB. The arbiter processes requests from all requesters and decides, as optimally as it can, which data ring is granted to which requester and the time at which the data ring is granted. The memory controller is given the highest priority to prevent stalling of