Outline

- HW/SW interfacing basics
- Avalon system interconnect fabric
- Avalon-MM memory-mapped interface
HW/SW Interfacing Basics: Structural View

• Hardware view: Connect the accelerator to the communication architecture (system bus)
  – Details of interfacing with the bus handled by a bus interface

![Diagram showing hardware view of HW/SW interfacing basics](image-url)
HW/SW Interfacing Basics: Functional View

- Two aspects to communication
  - Data Transfer: Providing the accelerator with the inputs to the computation it performs, and retrieving the results
  - Synchronization: Telling the accelerator when to start, knowing when the accelerator is done
HW/SW Interfacing Basics: Data Transfer

• Data transfer
  – View accelerator as memory (memory-mapped I/O)
  – Unique portion of physical memory address space assigned to accelerator
  – Loads/stores to these addresses are transported from/to the accelerator by the communication architecture
HW/SW Interfacing Basics: Data Transfer

• Three options
  – Software copies data from memory to accelerator, reads results back
    • Accelerator only needs to be a slave
  – Accelerator autonomously copies data from memory and writes results back
    • Accelerator needs to be a bus master
  – Direct memory access
HW/SW Interfacing Basics: DMA

• Copying data between the memory and the accelerator can take significant time

• Direct Memory Access: Use a separate component (DMA controller) to perform data transfers between accelerator and memory

• Frees up processor to execute other functions
HW/SW Interfacing Basics: Synchronization

• Synchronization
  – Starting the computation in the accelerator
    • Software writes to a memory-mapped control register in the accelerator
  – Detecting when the accelerator is finished
    • Two basic approaches
      • Polling: Software keeps reading from a memory-mapped status register in the accelerator, which indicates when it is done
      • Interrupts: The accelerator generates an interrupt to the processor when it is done