ECE 695R: SYSTEM-ON-CHIP DESIGN

Module 2: HW/SW Partitioning
Lecture 2.7: Avalon System: Pipelined and Burst Transfers

Anand Raghunathan
raghunathan@purdue.edu

Fall 2014, ME 1052, T Th 12:00PM-1:15PM
Pipelined Transfers

- Basic idea: Master starts a new transaction before the previous one is completed
  - Constraint: cannot initiate a new transfer while \texttt{waitrequest} is asserted
Pipelined Transfers

- Simple buses (including Avalon) only support in-order execution of transactions
  
  More on out-of-order when we talk about on-chip communication architecture
Pipelining Masters and Slaves

- Qsys automatically inserts logic to interface between masters and slaves with different pipelining styles.

<table>
<thead>
<tr>
<th>Master</th>
<th>Slave</th>
<th>Pipeline Management Logic Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>No pipeline</td>
<td>No pipeline</td>
<td>The system interconnect fabric does not instantiate logic to handle pipeline latency.</td>
</tr>
<tr>
<td>No pipeline</td>
<td>Pipelined with fixed or variable latency</td>
<td>The system interconnect fabric forces the master to wait through any slave-side latency cycles. This master-slave pair gains no benefits of pipelining, because the master waits for each transfer to complete before beginning a new transfer. However, while the master is waiting, the slave can accept transfers from a different master.</td>
</tr>
<tr>
<td>Pipelined</td>
<td>No pipeline</td>
<td>The system interconnect fabric carries out the transfer as if neither master nor slave were pipelined, causing the master to wait until the slave returns data.</td>
</tr>
<tr>
<td>Pipelined</td>
<td>Pipelined with fixed latency</td>
<td>The system interconnect fabric allows the master to capture data at the exact clock cycle when data from the slave is valid. This process enables the master-slave pair to achieve maximum throughput performance.</td>
</tr>
<tr>
<td>Pipelined</td>
<td>Pipelined with variable latency</td>
<td>This is the simplest pipelined case, in which the slave asserts a signal when its read data is valid, and the master captures the data. This case enables this master-slave pair to achieve maximum throughput.</td>
</tr>
</tbody>
</table>

Source: Avalon SIF for MM Interfaces, page 6
Burst Transfers

• Basic idea: Master requests access to the bus for several consecutive cycles
  – More efficient way to transfer larger amounts of data
  – Eliminates the overhead of arbitration for each transfer
  – Slaves may be more efficient if they know the length of a multi-word transfer ahead of time (e.g., SDRAM controller)
  – Only applies to transfers to/from consecutive addresses
  – Does not guarantee one cycle per transfer!
    • Slave can still slow down each transfer using wait cycles