ΗΥ220
Εργαστήριο Ψηφιακών
Κυκλωμάτων

Χειμερινό Εξάμηνο
2017-2018

Interconnects: AXI Protocol
AXI

• AMBA AXI protocol is targeted at high-performance, high-frequency system designs

• AXI key features
  - Separate address/control and data phases
  - Support for unaligned data transfers using byte strobes
  - Separate read and write data channels to enable low-cost Direct Memory Access (DMA)
  - Ability to issue multiple outstanding addresses
  - Out-of-order transaction completion
  - Easy addition of register stages to provide timing closure

Sources: ARM and Xilinx
5 Independent Channels

• **Read address channel** and **Write address channel**
  - Variable length burst: 1 ~ 16 data transfers
  - Burst with a transfer size of 8 ~ 1024 bits (1B ~ 1KB)

• **Read data channel**
  - Convey data and any read response info.
  - Data bus can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits

• **Write data channel**
  - Data bus can be 8, 16, 32, 64, 128, 256, 512, or 1024 bits

• **Write response channel**
  - Write response info.

Sources: ARM and Xilinx
AXI Read Operation

RREADY: From master, indicate that master can accept the read data and response info.

Sources: ARM and Xilinx
AXI Write Operation

Write Address Channel
Write Data Channel
Write Response Channel

WVALID Source: Master
WREADY Source: Slave
BVALID Source: Slave
BREADY Source: Master

Sources: ARM and Xilinx
Out-of-order Completion

• AXI gives an ID tag to every transaction
  - Transactions with the same ID are completed in order
  - Transactions with different IDs can be completed out of order

Table 2-2 Write address channel signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWID[3:0]</td>
<td>Master</td>
<td>Write address ID. This signal is the identification tag for the write address group of signals.</td>
</tr>
<tr>
<td>AWADDR[31:0]</td>
<td>Master</td>
<td>Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.</td>
</tr>
</tbody>
</table>

Table 2-5 Read address channel signals

<table>
<thead>
<tr>
<th>Signal</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARID[3:0]</td>
<td>Master</td>
<td>Read address ID. This signal is the identification tag for the read address group of signals.</td>
</tr>
<tr>
<td>ARADDR[31:0]</td>
<td>Master</td>
<td>Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst.</td>
</tr>
</tbody>
</table>

Sources: ARM and Xilinx
ID Signals

The ID tag for the write address group of signals.

The write ID tag for a write transaction. Along with the write data, the master transfers a WID to match the AWID of the corresponding address.

The ID tag for the write response. The slave transfers a BID to match the AWID and WID of the transaction to which it is responding.

The ID tag for the read address group of signals.

The read ID tag for a read transaction. The slave transfers an RID to match the ARID of the transaction to which it is responding.

Sources: ARM and Xilinx
Out-of-order Completion

- Out-of-order transactions can improve system performance in 2 ways
  - Fast-responding slaves respond in advance of earlier transactions with slower slaves
  - Complex slaves can return data out of order
    - A data item for a later access might be available before the data for an earlier access is available

- If a master requires that transactions are completed in the same order that they are issued, they must all have the same ID tag

- It is not a required feature
  - Simple masters and slaves can process one transaction at a time in the order they are issued

Sources: ARM and Xilinx
Addition of Register Slices

• AXI enables the insertion of a register slice in any channel at the cost of an additional cycle latency
  − Trade-off between latency and maximum frequency

• It can be advantageous to use
  − Direct and fast connection between a processor and high-performance memory
  − Simple register slices to isolate a longer path to less performance-critical peripherals

Sources: ARM and Xilinx
AXI-Lite

• The AXI4-Lite interface is a subset of the AXI4 interface intended for communication with control registers in components.

• The aim of AXI4-Lite is to allow simple component interfaces to be built that are smaller and also require less design and validation effort.

• Having a defined subset of the full AXI4 interface allows many different components to be built using the same subset and also allows a single common conversion component to be used to move between AXI4 and AXI4-Lite interfaces.

Sources: ARM and Xilinx
**AXI-Lite**

- No burst
- Data width 32 or 64 only
- Simple “logic” to connect AXI4 master to AXI4-Lite slave
  - Reflect master’s transaction ID
- This is best for simple systems with minimal peripherals

Sources: ARM and Xilinx
AXI-Lite Signal list

• Subset of AXI signal set
• Simple traditional signaling
• Targeted applications: simple, low-performance peripherals
  - GPIO
  - Uart Lite

<table>
<thead>
<tr>
<th>Source</th>
<th>Write address channel</th>
<th>Write data channel</th>
<th>Write response channel</th>
<th>Read address channel</th>
<th>Read data channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACLK</td>
<td>AWVALID</td>
<td>WVALID</td>
<td>BVALID</td>
<td>ARVALID</td>
<td>RVALID</td>
</tr>
<tr>
<td>ARESETn</td>
<td>AWREADY</td>
<td>WREADY</td>
<td>BREADY</td>
<td>ARREADY</td>
<td>RREADY</td>
</tr>
<tr>
<td></td>
<td>AWADDR</td>
<td>WDATA</td>
<td>BRESP</td>
<td>ARADDR</td>
<td>RDATA</td>
</tr>
<tr>
<td></td>
<td>AWPROT</td>
<td>WSTRB</td>
<td>-</td>
<td>ARPROT</td>
<td>RRESP</td>
</tr>
</tbody>
</table>

Sources: ARM and Xilinx
AXI-Stream

• No address channel

• Not read and write, always master to slave

• Unlimited burst length

Sources: ARM and Xilinx
AXI Additional Features

• ID fields for each of the five channels facilitate overlapped transactions
  ‒ Provides for a transaction tag
• Transaction burst type determines address bus behavior
  ‒ Fixed, increment, or wrap
• Optional address Lock signals facilitates exclusive and atomic access protection
• System cache support
• Protection unit support
• Error support
• Unaligned address

Sources: ARM and Xilinx
Documentation

• ARM specifications
  - AMBA AXI Protocol Version 2.0
  - AMBA 4 AXI4-Stream Protocol Version 1.0

• Xilinx AXI Reference Guide, UG761
  - AXI Usage in Xilinx FPGAs
    o Introduce key concepts of the AXI protocol
    o Explains what features of AXI Xilinx has adopted

Sources: ARM and Xilinx