On Chip Bus

Speaker: 沈文中
Outline

• AMBA Bus
  – Advanced System Bus
  – Advanced High-performance Bus
  – Advanced Peripheral Bus
• FPGA design flow
National Taiwan University

Bus Architecture

- High-performance ARM processor
- High-bandwidth on-chip RAM
- High-bandwidth External Memory Interface
- AHB or ASB
- DMA bus master
- UART
- Timer
- APB
- Keypad
- PIO

**AMBA AHB**
- High performance
- Pipelined operation
- Multiple bus masters
- Burst transfers
- Split transactions

**AMBA ASB**
- High performance
- Pipelined operation
- Multiple bus masters

**AMBA APB**
- Low power
- Latched address and control
- Simple interface
- Suitable for many peripherals
Outline

• AMBA Bus
  – Advanced System Bus
    • High performance
    • Pipelined operation
    • Multiple bus master
  – Advanced High-performance Bus
  – Advanced Peripheral Bus
• FPGA design flow
ASB characters

• Negative edge trigger
• Tri-state bus
  – Drawback: More effort used to control timing
  – Advantage: cost less area
Outline

• AMBA Bus
  – Advanced System Bus
  – **Advanced High-performance Bus**
    • High performance
    • Pipelined operation
    • Multiple bus master
    • Burst transfers
    • Split transactions
  – Advanced Peripheral Bus

• FPGA design flow
AHB simple Arch.
AHB Components

- **AHB master** is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time. (max. 16)

- **AHB slave** responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.
AHB Components

- **AHB arbiter** ensures that only one bus master at a time is allowed to initiate data transfers.
- **AHB decoder** is used to decode the address of each transfer and provide a select signal for the slave that is involved in the transfer. A single centralized decoder is required in all AHB implementations.
AHB Signals(i)

• AHB Signals can be classified as
  – Clock (HCLK)
  – Address and read/write data (HADDR, HRDATA, HWDATA)
  – Arbitration (HGRANTx, HMASTER, HMASTLOCK,…)
  – Control signal (HRESETn,…)
  – Response signal(HREADY, HRESP)
AHB Signals(ii)

- **Transfer signals**
  - **HCLK**
    - bus clock. All signal timings are related to the rising edge.
  - **HADDR[31:0]**
    - 32 bits system bus
  - **HWDATA/HRDATA [31:0]**
    - 32 bits write/read data bus
  - **HWRITE**
    - High: write data
    - Low: read data
  - **HREADY**
    - Transfer done
AHB Signals(ii)

Basic Transfer

• Each transfer consists of
  – An address and control cycle
  – One or more cycles for the data

Figure 3-4 Transfer with wait states
AHB Arch.
AHB Signals (iii)

• Control signals
  – HTRANS[1:0]
    • Current transfer type
  – HBURST[2:0]
    • When sequential transfer, control transfer relation
  – HSIZE[2:0]
    • Control transfer size=\(2^{\text{HSIZE}}\) bytes (max=1024 bits)
  – HPROT[3:0]
    • Protection data
AHB Signals(iii)-HTRANS

- **HTRANS[1:0]**
  - **IDLE**: master don’t need data to be transferred
  - **BUSY**: allows bus masters to insert IDLE cycles in the middle of bursts of transfers.
  - **NONSEQ**: The address and control signals are unrelated to the previous transfer.
  - **SEQ**: the address is related to the previous transfer.
AHB Signals(iii)-HBURST

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCLK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTRANS[1:0]</td>
<td>NONSEQ</td>
<td>SEQ</td>
<td>SEQ</td>
<td>SEQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HADDR[31:0]</td>
<td>0x38</td>
<td>0x3C</td>
<td>0x30</td>
<td>0x34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HBURST[2:0]</td>
<td></td>
<td>WRAP4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HWRITE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSIZE[2:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HPROT[3:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HWDATA[31:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HREADY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HRDATA[31:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HBURST[2:0]</td>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>SINGLE</td>
<td>Single transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>INCR</td>
<td>Incrementing burst of unspecified length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>WRAP4</td>
<td>4-beat wrapping burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>INCR4</td>
<td>4-beat incrementing burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>WRAP8</td>
<td>8-beat wrapping burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>INCR8</td>
<td>8-beat incrementing burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>WRAP16</td>
<td>16-beat wrapping burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>INCR16</td>
<td>16-beat incrementing burst</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
• Response signals
  – HREADY
    • Transfer done, ready for next transfer
  – HRESP[1:0]
    • OKAY transfer complete
    • ERROR transfer failure (ex: write ROM)
    • RETRY higher priority master can access bus
    • SPLIT other master can access bus
AHB Signals

- Arbiter signals
  - HGRANTx
    - Select active bus master
  - HMASTER[3:0]
    - Multiplex signals that sent from master to slave
  - HMASTLOCK
    - Locked sequence
Master signal

Figure 3-27 AHB bus master interface diagram
Arbiter signal
Slave signal
Outline

• AMBA Bus
  – Advanced System Bus
  – Advanced High-performance Bus
  – Advanced Peripheral Bus
    • Low power
    • Latched address and control
    • Simple interface
    • Suitable for many peripherals

• FPGA design flow
Figure 5-2 State diagram

Figure 5-7 APB slave interface description
APB signals

• APB character
  – Always two cycle transfer
  – No wait cycle and response signal

• APB signals
  – PCLK Bus clock, rising edge is used to time all transfers.
  – PRESETn APB reset, active Low.
APB signals

- **PADDR[31:0]** APB address bus.
- **PSELx** Indicates that the slave device is selected. There is a PSELx signal for each slave.
- **PENABLE** Indicates the second cycle of an APB transfer.
- **PWRITE** Transfer direction. High for write access, Low for read access.
- **PRDATA** Read data bus
- **PWDATA** Write data bus
Outline

• AMBA Bus
  – Advanced System Bus
  – Advanced High-performance Bus
  – Advanced Peripheral Bus

• FPGA design flow
  – Compile flow
  – Download flow
Compile flow(i)

- All verilog module must be synthesized by Xilinx Software
Compile flow(ii)

• Add example2.ucf (define the pin assignment) into project

• Double click generate programming file to generate *.bit (which can be downloaded into FPGA)
Outline

• AMBA Bus
  – Advanced System Bus
  – Advanced High-performance Bus
  – Advanced Peripheral Bus

• FPGA design flow
  – Compile flow
  – Download flow
Download flow

• Connect config link
• Connect Multi-ICE to Logic Module
• Power on
• Use progcrd.exe to download example2.bit file
• Remove config link
• Power off