Announcements

• HW1.2 Notes
  • Do not put text or labels on Top Metal Layer. That’s what the silkscreen (Top Overlay) is for!

• Avoid direct pad connects on chips

• Do not ignore DFM errors
  • Vias too small
  • Insufficient text width
  • Signal Distance problems
  • Double drill hits
  • Insufficient spacing
Why to avoid Via in Pad
Lab 1, Common Mistakes

- Verilog Language Issue
  - `output [3:0] Y;
    Y = 4'b110`
  - Correct, but looks strange

- Propagation delay usually measured from 50% point
Minute Quiz
Advanced Microcontroller Bus Architecture (AMBA)
- Advanced High-performance Bus (AHB)
- Advanced Peripheral Bus (APB)
Internal and external busses are accessed in very different ways!

Accessed logically w/ VHDL & Verilog

Accessed physically w/ wires
Why not just export the AHB-Lite or APB off-chip?
Outline

• Announcements

• Minute quiz

• Asynchronous Memory

• External Memory Controller

• Open Discussions
An SRAM chip and its asynchronous parallel interface

- A: 20-bit address bus
- DQ: 8-bit data bus
- CE#: chip enable
- WE#: write enable
- OE#: output enable

<table>
<thead>
<tr>
<th>CE</th>
<th>OE</th>
<th>WE</th>
<th>DQ₁ to DQ₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>X</td>
<td>X</td>
<td>Not Selected</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>Read</td>
</tr>
<tr>
<td>L</td>
<td>X</td>
<td>L</td>
<td>Write</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>High Z</td>
</tr>
</tbody>
</table>
GS78108 read cycle...has no clock
GS78108 WE#-controlled write cycle

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write cycle time</td>
<td>( t_{WC} )</td>
</tr>
<tr>
<td>Address valid to end of write</td>
<td>( t_{AW} )</td>
</tr>
<tr>
<td>Chip enable to end of write</td>
<td>( t_{CW} )</td>
</tr>
<tr>
<td>Data set up time</td>
<td>( t_{DW} )</td>
</tr>
<tr>
<td>Data hold time</td>
<td>( t_{DH} )</td>
</tr>
<tr>
<td>Write pulse width</td>
<td>( t_{WP} )</td>
</tr>
<tr>
<td>Address set up time</td>
<td>( t_{AS} )</td>
</tr>
<tr>
<td>Write recovery time (WE)</td>
<td>( t_{WR} )</td>
</tr>
<tr>
<td>Write recovery time (CE)</td>
<td>( t_{WR1} )</td>
</tr>
<tr>
<td>Output Low Z from end of write</td>
<td>( t_{WLZ} )</td>
</tr>
<tr>
<td>Write to output in High Z</td>
<td>( t_{WHZ} )</td>
</tr>
</tbody>
</table>
GS78108 CE#-controlled write cycle

Parameter | Symbol
---|---
Write cycle time | $t_{WC}$
Address valid to end of write | $t_{AW}$
Chip enable to end of write | $t_{CW}$
Data set up time | $t_{DW}$
Data hold time | $t_{DH}$
Write pulse width | $t_{WP}$
Address set up time | $t_{AS}$
Write recovery time ($WE$) | $t_{WR}$
Write recovery time ($CE$) | $t_{WR1}$
Output Low Z from end of write | $t_{WLZ}$
Write to output in High Z | $t_{WHZ}$
An asynchronous NOR flash memory (that does not have a clock input line)

- A: 25-bit address bus
- DQ: A 16-bit data bus
- CE#: chip enable
- WE#: write enable
- OE#: output enable
- BYTE: 8-bit or 16-bit mode
- WP#/ACC: write protect
- RY/BY#: ready/busy
- RESET#: clear internal status
The S29GL512P read cycle diagram illustrates the timing for various inputs and outputs. The key parameters include:

- **tRC**: Address demise to valid.
- **tACC**: Address modification to stable.
- **tCEH**: Chip enable to high.
- **tOEH**: Output enable to high.
- **tRH**: High to low.
- **tOE**: Output enable.
- **tDF**: Address demise.
- **tOH**: Output enable to high.
- **tCE**: Chip enable.

The waveforms in the diagram show the transition states for inputs and outputs, including:

- Steady
- Changing from H to L
- Changing from L to H
- Don't Care, Any Change Permitted
- Changing, State Unknown
- Does Not Apply

The outputs include HIGH Z and Output Valid, indicating the state of the outputs during different input transitions.
**S29GL512P read cycle timing**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description (Notes)</th>
<th>Test Setup</th>
<th>Speed Options</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{AVAV}$</td>
<td>$t_{RC}$ Read Cycle Time</td>
<td>$V_{IO} = V_{CC} = 2.7$ V</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = 1.65$ V to $V_{CC}$</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{CC} = 3$ V</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = V_{CC} = 3.0$ V</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{AVQV}$</td>
<td>$t_{ACC}$ Address to Output Delay (1)</td>
<td>$V_{IO} = V_{CC} = 2.7$ V</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = 1.65$ V to $V_{CC}$</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{CC} = 3$ V</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = V_{CC} = 3.0$ V</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{LEQV}$</td>
<td>$t_{CE}$ Chip Enable to Output Delay (2)</td>
<td>$V_{IO} = V_{CC} = 2.7$ V</td>
<td>100</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = 1.65$ V to $V_{CC}$</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{CC} = 3$ V</td>
<td></td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$V_{IO} = V_{CC} = 3.0$ V</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{PAQC}$</td>
<td>Page Access Time</td>
<td>Max</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{GLOQV}$</td>
<td>$t_{OE}$ Output Enable to Output Delay</td>
<td>Max</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{EHQZ}$</td>
<td>$t_{DF}$ Chip Enable to Output High Z (3)</td>
<td>Max</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{GHQZ}$</td>
<td>$t_{DF}$ Output Enable to Output High Z (3)</td>
<td>Max</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>$t_{AXQX}$</td>
<td>$t_{OH}$ Output Hold Time From Addresses, CE# or</td>
<td>Min</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>OE#, Whichever Occurs First</td>
<td></td>
<td></td>
<td>–</td>
</tr>
<tr>
<td>$t_{OEH}$</td>
<td>$t_{OH}$ Output Enable Hold Time (3)</td>
<td>Read</td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Toggle and Data# Polling</td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>$t_{CEH}$</td>
<td>Chip Enable Hold Time</td>
<td>Read</td>
<td></td>
<td>Min</td>
</tr>
</tbody>
</table>

1. $CE\#$, $OE\# = V_{IL}$
2. $OE\# = V_{IL}$
3. Not 100% tested.
4. See Figure 11.3 and Table 11.1 for test specifications.
5. Unless otherwise indicated, AC specifications for 110 ns speed options are tested with $V_{IO} = V_{CC} = 2.7$ V. AC specifications for 110 ns speed options are tested with $V_{IO} = 1.8$ V and $V_{CC} = 3.0$ V.
LCD controller (PMO13701) exports both a parallel and a serial interface

- 8-bit parallel interface
- D: 8-bit data bus
- BS1/BS2: I/fc mode select
- CS#: chip select
- RD#: read data
- WR#: write data
- RES#: hardware reset
- D/C#: Slave address bit
• Minute quiz

• Announcements

• Asynchronous Memory

• External Memory Controller

• Open Discussions
Accessing external parallel devices

- AHB and APB are wide, de-multiplexed internal busses
- External busses tradeoff pin-count, performance, ...
  - ISA
  - VESA
  - AGP
  - PCI
  - VME
  - IDE
  - CF
- MCUs often integrate external memory controllers
  - Often tailored to specific peripheral class
  - EMCs ease memory/peripheral interfacing
• Provide glueless interface to external devices

• Asynchronous and Synchronous memories supported

• EMC is mapped into system address space
  - 0x70000000 to 0x77FFFFFFF

• Offers
  - 2 chip select lines (CS)
  - 8-bit or 16-bit shared data bus
  - Write enable generation
  - Translates 32-bit AHB transfers into half-word and byte txns
  - Automatic translation of misaligned addresses
AHB read/write transfers
EMC operation

- EMC accepts single AHB transactions
  - Reading external memory devices (EMD)
  - Writing EMD

- EMC reformat single AHB transactions into EMD format

- EMC may use multiple CLK cycles to complete access
  - Recall AHB transfers complete in two cycles

- EMC cannot complete EMD R/W in only two cycles

- User must configure EMC to include wait states
AHB read transfer with two wait states
EMC operation continued

- EMC uses extra cycles to complete EMD transaction
- AHB address phase is one cycle (wait states are in data)
- EMC requires one cycle to output EMD address
- EMD requires two cycles to fetch the data
- EMC requires one additional cycle to transfer data to AHB
- A total of three wait states on AHB transfers
Figure 7-8 • Byte Wide External Memory Device Memory Map

- **Upper 64M of AHB Memory Space**
  - AHB Address HADDR [31:0]: 0x77FFFFFF – 0x77FFFFFFC
  - AHB Data H(R/W)DATA[31:0]

- **Lower 64M of AHB Memory Space**
  - AHB Address HADDR [31:0]: 0x74000000 – 0x74000003
  - AHB Data H(R/W)DATA[31:0]

- **64M External Memory Space**
  - AHB Address HADDR [31:0]: 0x70000000 – 0x70000007
  - AHB Data H(R/W)DATA[31:0]

**EMC_AB[25:0]**
- 0x77FFFFFB – 0x77FFFFF8
- 0x70000000 – 0x70000007
- 0x74000000 – 0x74000003

**EMC_DB[7:0]**
- 0x00000000
- 0x00000001
- 0x03FFFFFF
- 0x03FFFFFD
- 0x03FFFFFC
- 0x03FFFFFE

**EMC_CS0_N**
- 0x00000000
- 0x00000001
- 0x03FFFFFF
- 0x03FFFFFD
- 0x03FFFFFC
- 0x03FFFFFE
Using the EMC to attach two S29GL512’s

- Two byte-mode NOR flash devices
- Byte enables used as write enables
### Half-Word Wide External 2x8-bit Memory

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[31:24] [23:16] [15:8] [7:0]</td>
<td>0x77FFFFFFF – 0x77FFFFFFFC 0x77FFFFFFB – 0x77FFFFFF8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[31:24] [23:16] [15:8] [7:0]</td>
<td></td>
<td>[15:8] 0x03FFFFFFF 0x03FFFFFD</td>
<td>[7:0] 0x03FFFFFE 0x03FFFFFC</td>
<td></td>
</tr>
<tr>
<td>[31:24] [23:16] [15:8] [7:0]</td>
<td></td>
<td></td>
<td>0x000000000</td>
<td></td>
</tr>
</tbody>
</table>

#### Upper 64M of AHB Memory Space

- 64M External Memory Space
  - 0x74000003 – 0x74000000
  - 0x77FFFFFF – 0x73FFFFFFFF

#### Lower 64M of AHB Memory Space

- 64M External Memory Space
  - 0x70000007 – 0x70000004
  - 0x70000003 – 0x70000000

- **EMC_CS1_N**
  - 0x00000001

- **EMC_CS0_N**
  - 0x00000001

- **EMC_AB[25:0]**
  - 0x03FFFFFE 0x03FFFFFC
Using the EMC to attach four GS78108’s

- Four asynchronous SRAMs
- Byte enables used as write enables
Using the EMC to interface with a synchronous SRAM

Figure 7-12 gives an overview of how to connect external memories to the EMC. While not exhaustive, the examples given are intended to provide the user with a sense of what the EMC is capable of.

Figure 7-12 shows a typical x16 SRAM connected to the EMC of a SmartFusion cSoC. An eight megabyte device is shown. The address bus is halfword aligned (A[17:0] = EMC_AB[18:1], since EMC_AB is a byte address). The halfword synchronous SRAM (16-bit) device uses the byte enable control pins to affect a single byte write.

Figure 7-13 shows a representative configuration of synchronous SRAM for the SmartFusion EMC. Eight megabyte SSRAMS are shown. The address bus is again halfword aligned (A[21:0] = EMC_AB[22:1], since EMC_AB is a byte address).
Outline

- Minute quiz
- Announcements
- Asynchronous Memory
- External Memory Controller
- Open Discussions
Questions?

Comments?

Discussion?