Digital System Design

by

Dr. Lesley Shannon

Email: lshannon@ensc.sfu.ca

Course Website: http://www.ensc.sfu.ca/~lshannon/courses/ensc350

Simon Fraser University

Slide Set: 12
Date: March 16, 2009

Slide Set Overview

- Lab 3

Your Slave Interface Port Requirements

Input signals use the prefix ‘**i_slave**’ while output signals are denoted by ‘**o_slave**’.

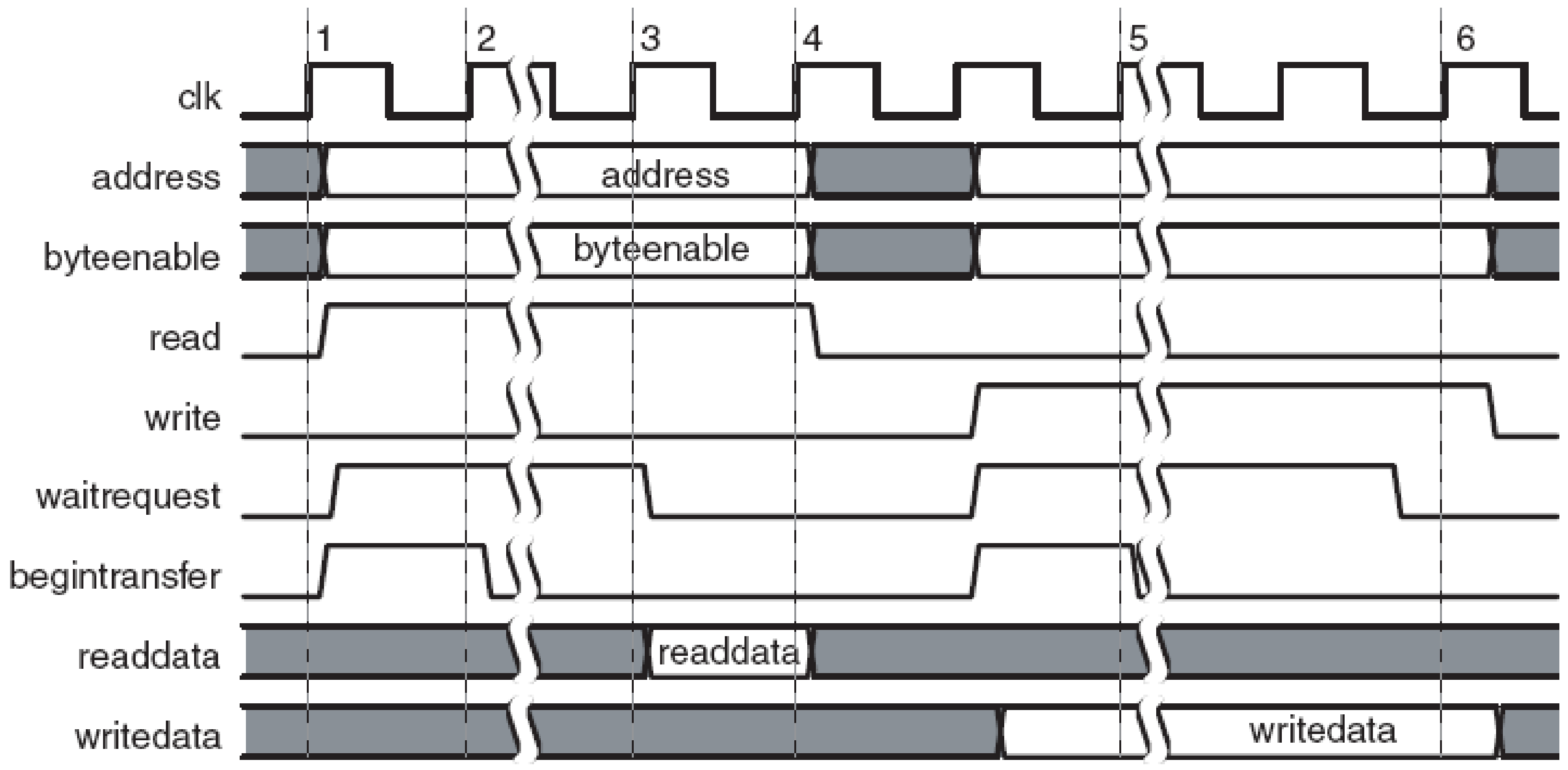
Port	**Avalon-MM signal**	**Size (bits)**
Clk	Clk	1
Reset	Reset	1
i_slave_read	Read	1
i_slave_write	Write	1
i_slave_address	Address	2
i_slave_byteenable	Byteenable	8
i_slave_writedata	Writedata	64
o_slave_readdatavalid	Readdatavalid	1
o_slave_waitrequest	Waitrequest	1
o_slave_readdata	Readdata	64

Your Avalon Slave

- Must support peripheral-controlled waitrequest reads and writes
  - Note that the begintransfer signal is not used for the DES slave

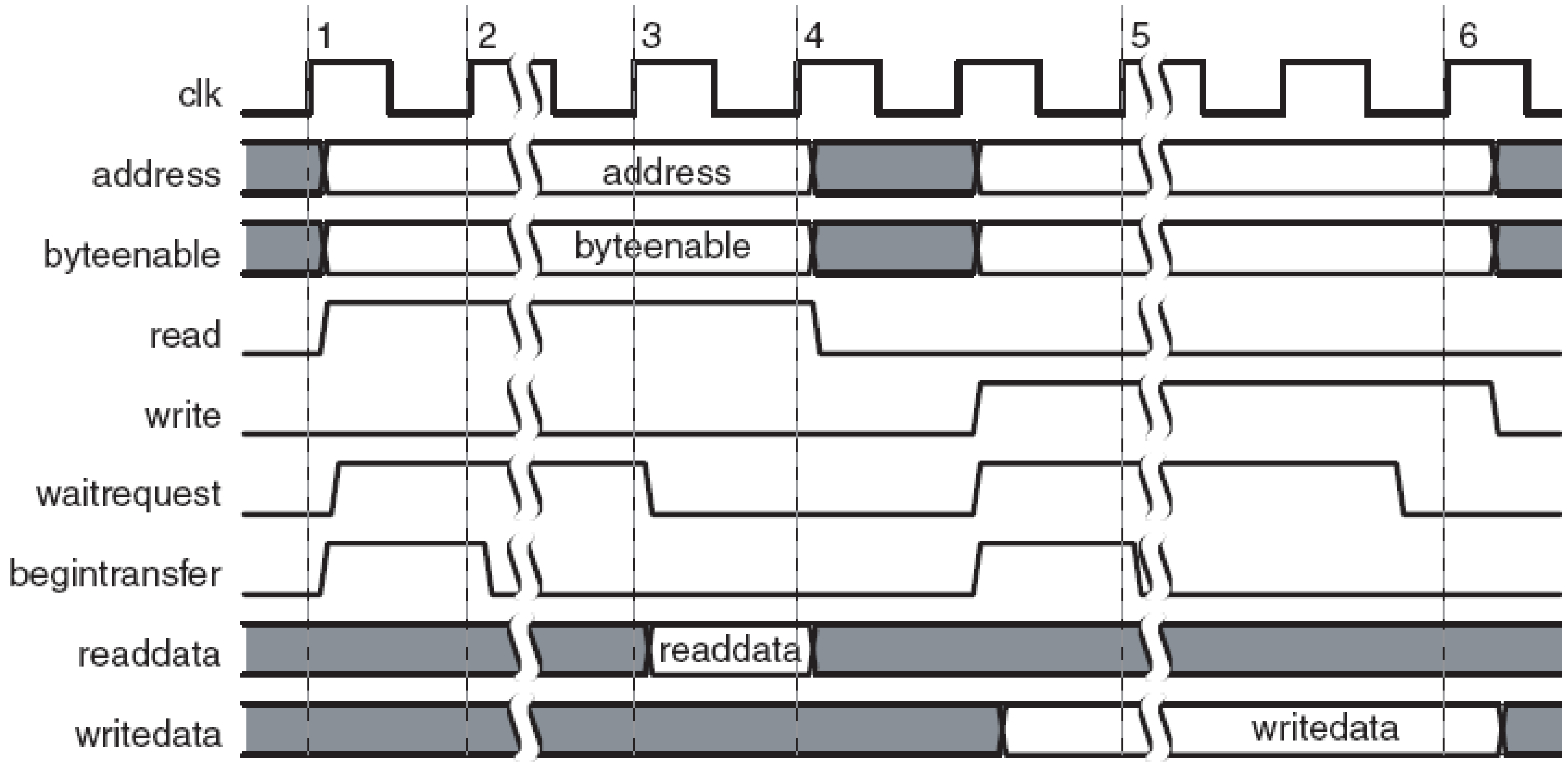
- Must support pipelined reads with variable latency

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest



1. Address, read, and begintransfer are asserted after the rising clock edge of clk
1. The slave asserts waitrequest to stall the transfer

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest

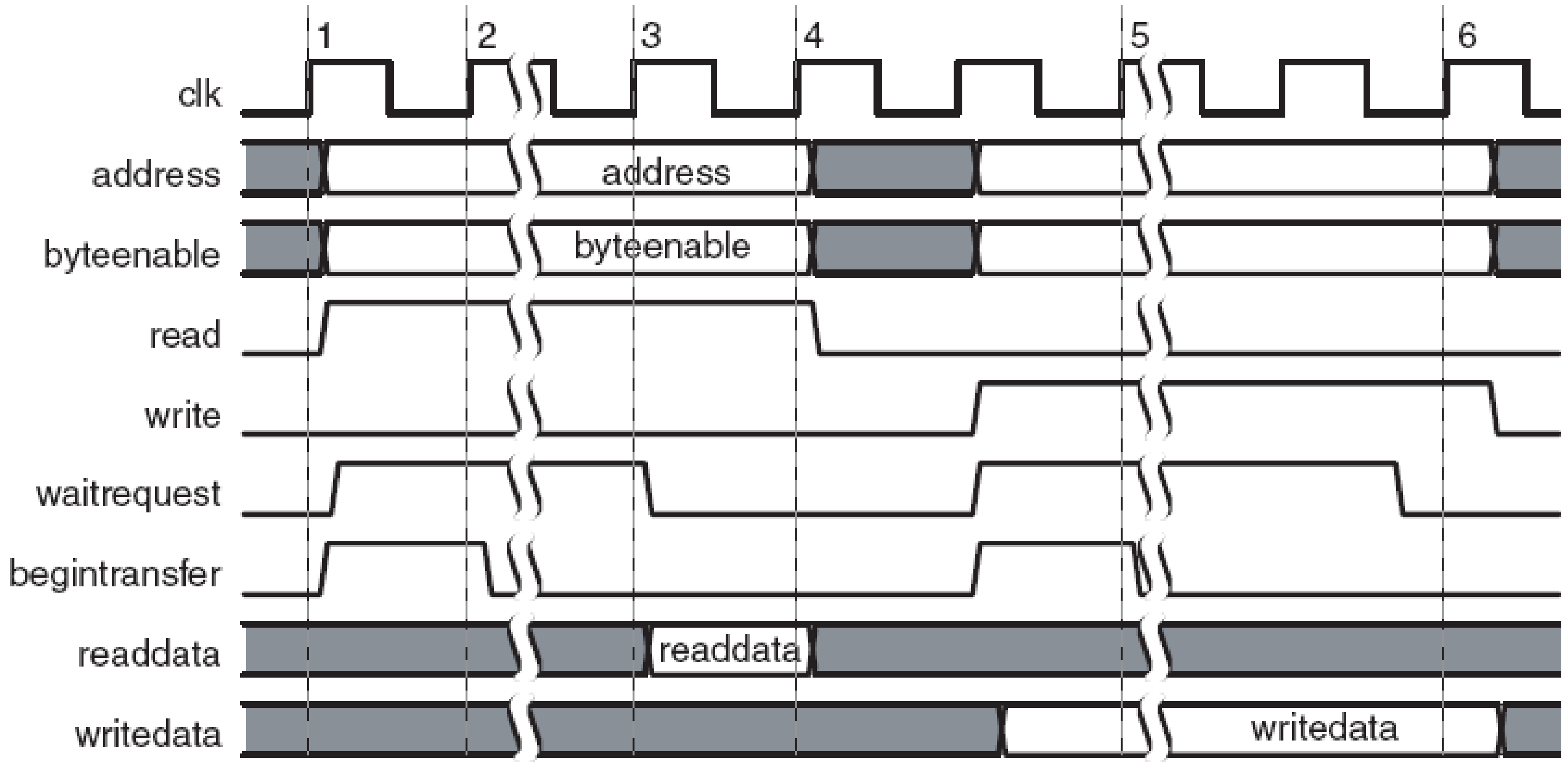


2. Waitrequest is sampled.

2. If Waitrequest is asserted (as shown here), the cycle becomes a wait-state, and address, read, write and byteenable remain constant.

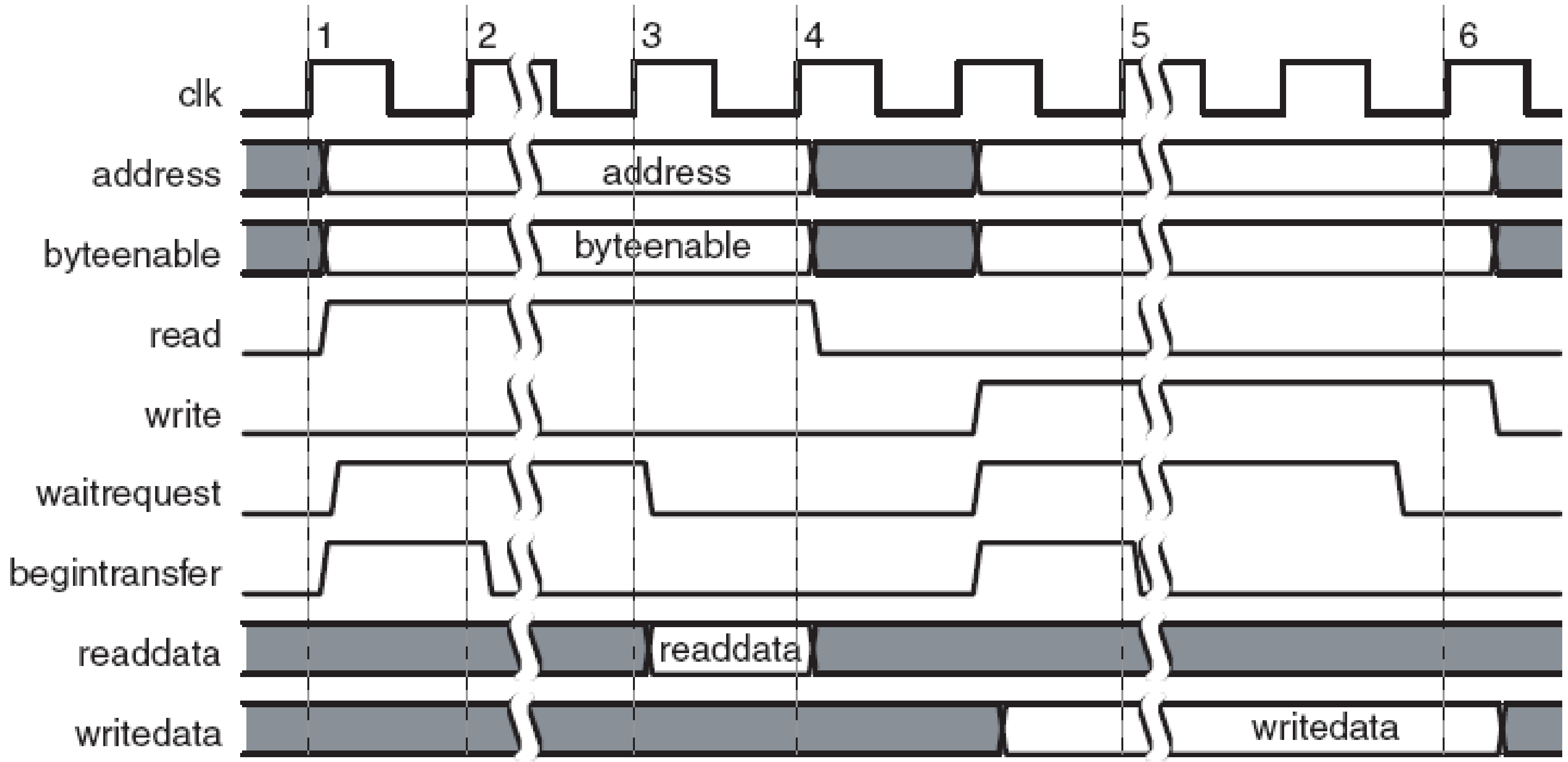
2.Begintransfer is not held constant.

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest



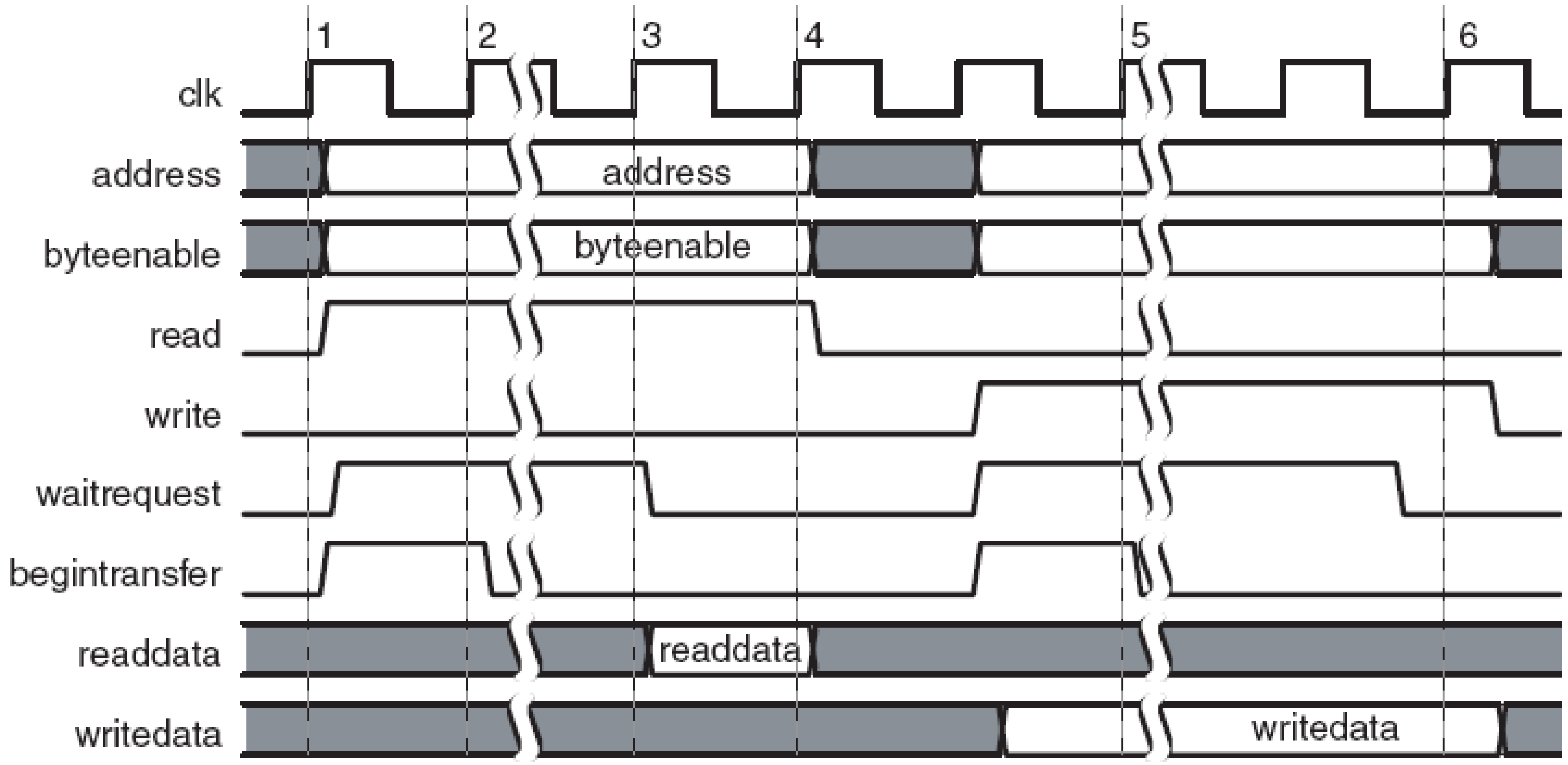
3. The slave presents valid readdata and deasserts waitrequest

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest



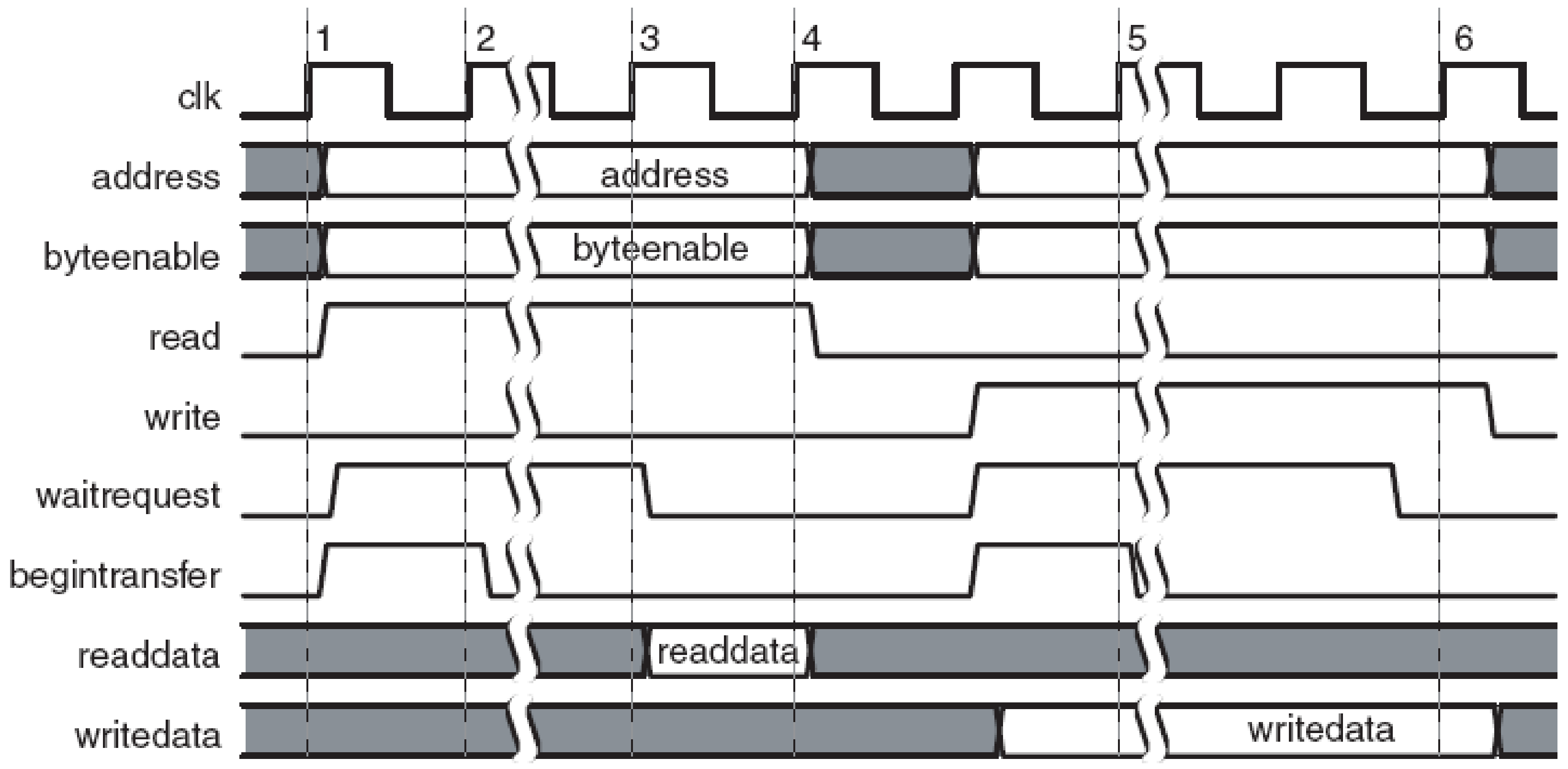
4. Readdata and deasserted waitrequest are sampled, completing the transfer.

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest



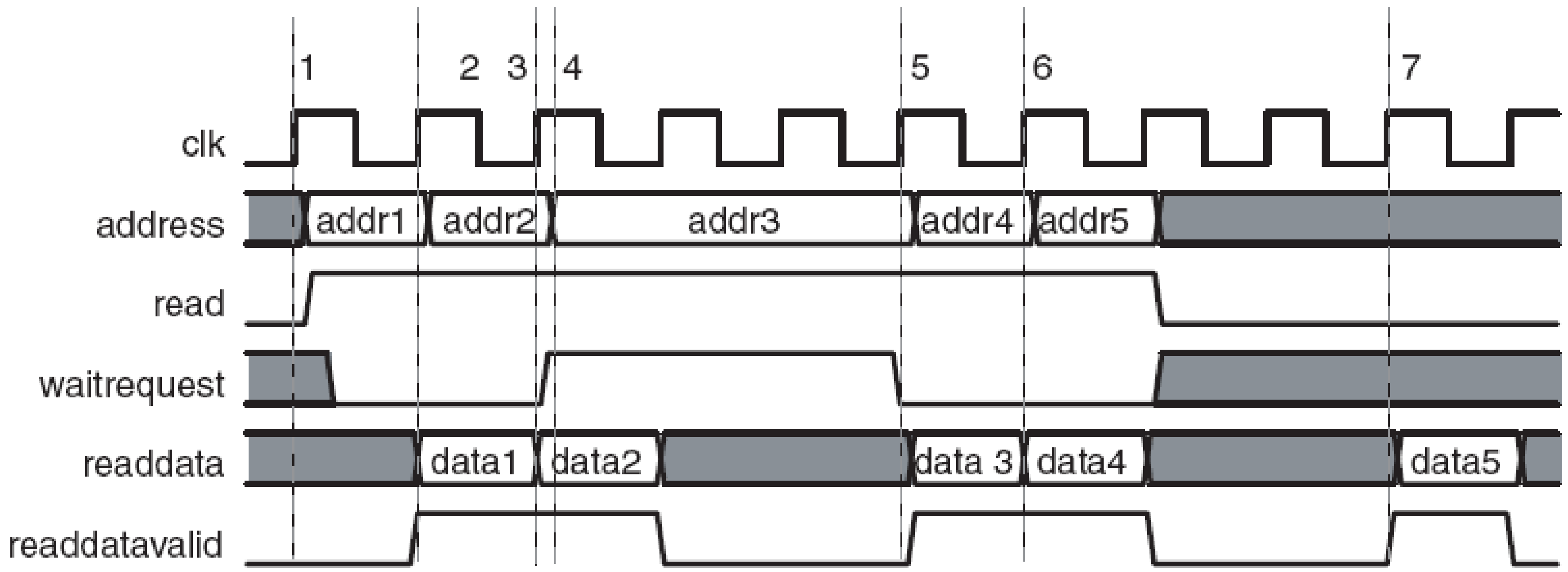
5. address, writedata, byteenable, begintransfer and write signals are asserted.
5. The slave responds by asserting waitrequest, thereby stalling the transfer

Slave Read and Write Transfers with Peripheral-Controlled Waitrequest



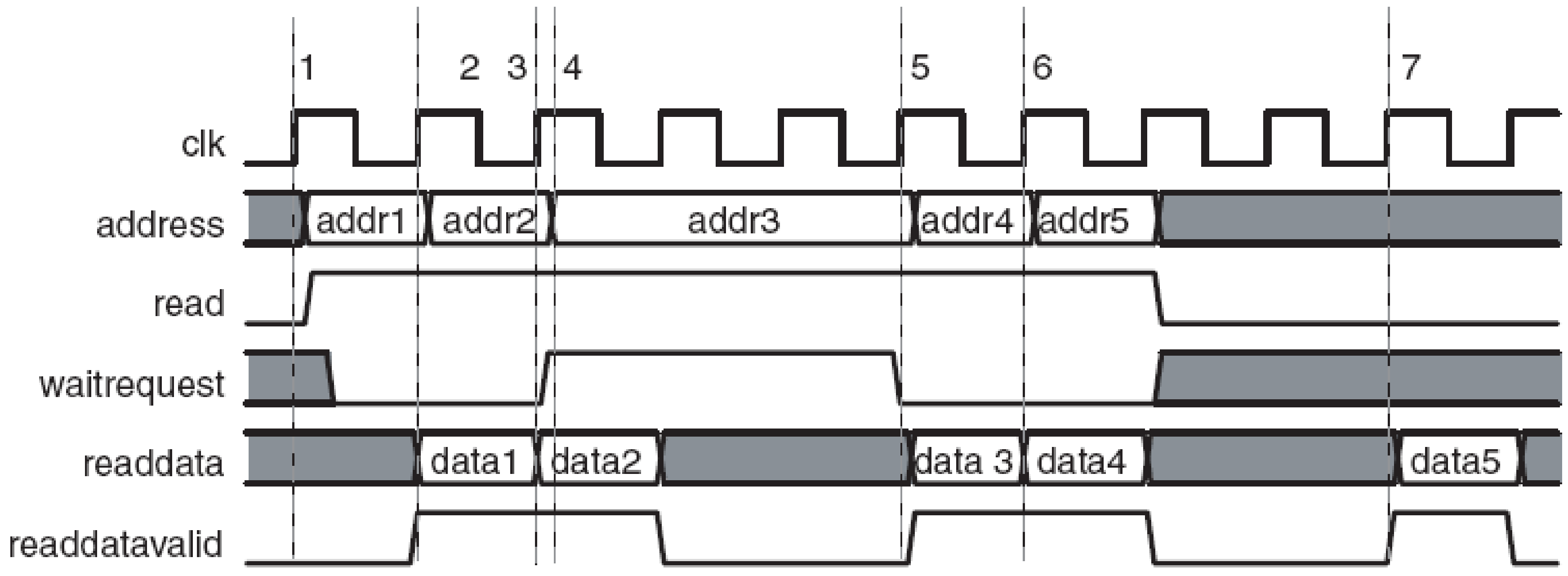
6. The slave captures the data being written and deasserts waitrequest, thereby ending the transfer.

Slave Pipelined Read Transfers with Variable Latency



1. Address and read are asserted to initiate a read transfer.

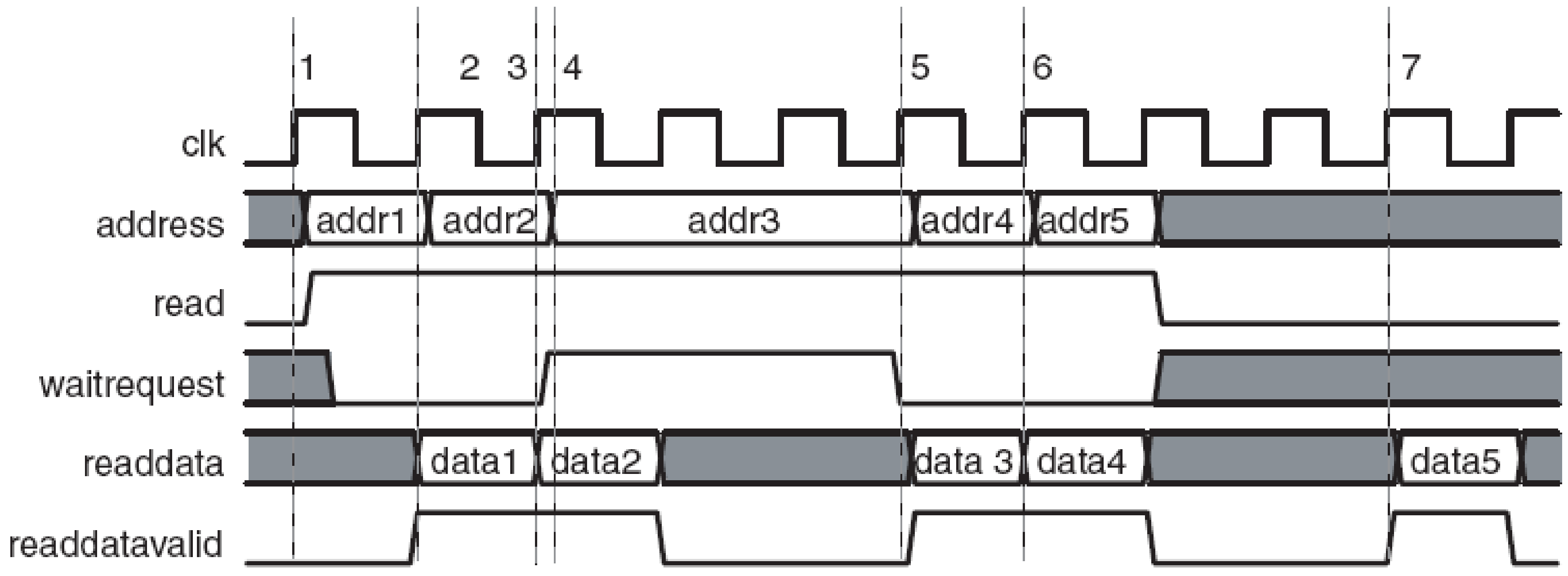
Slave Pipelined Read Transfers with Variable Latency



2. The slave captures addr1.

2.This generates the response data1 and asserts readdatavalid.

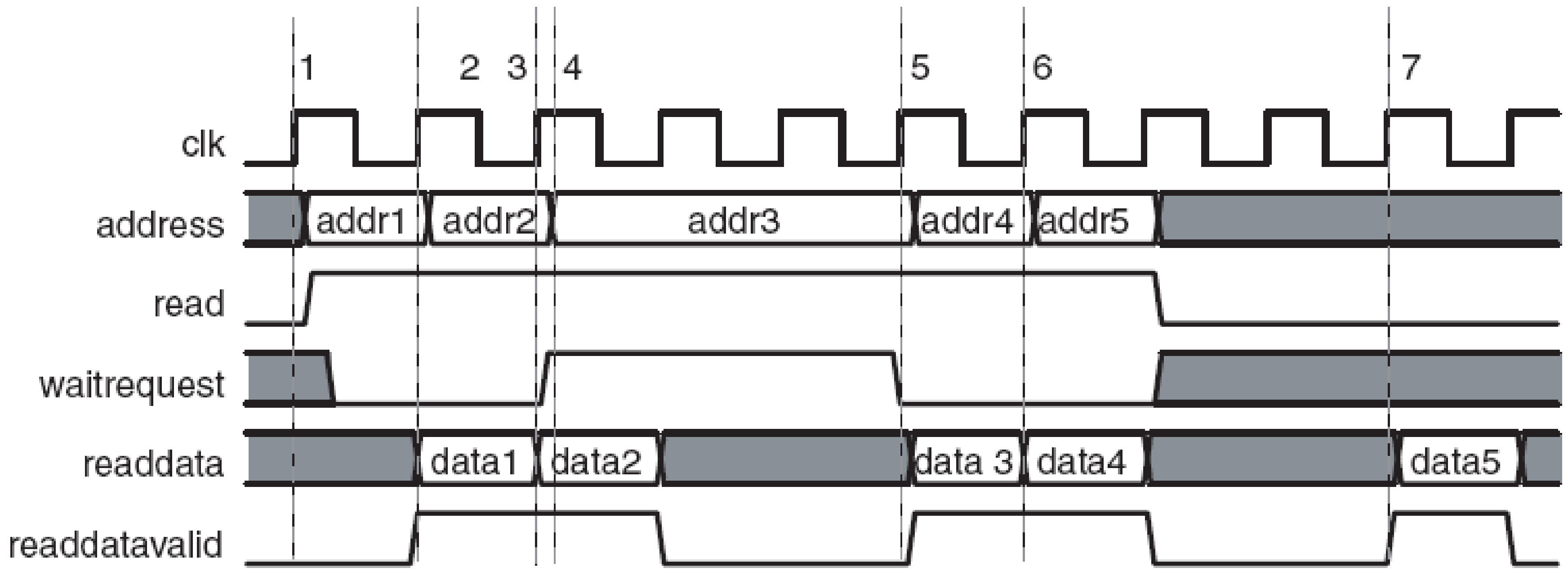
Slave Pipelined Read Transfers with Variable Latency



3. The slave captures addr2.

3. This generates the response data2 and asserts readdatavalid.

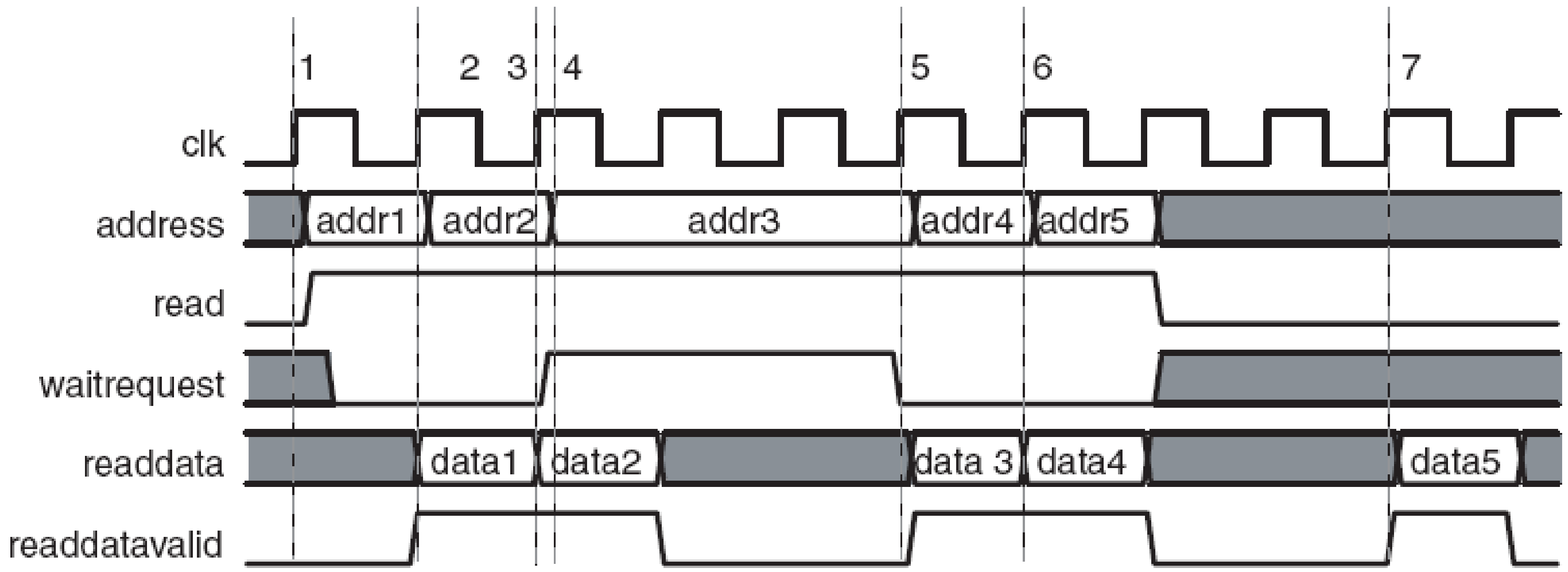
Slave Pipelined Read Transfers with Variable Latency



4. The slave asserts waitrequest.

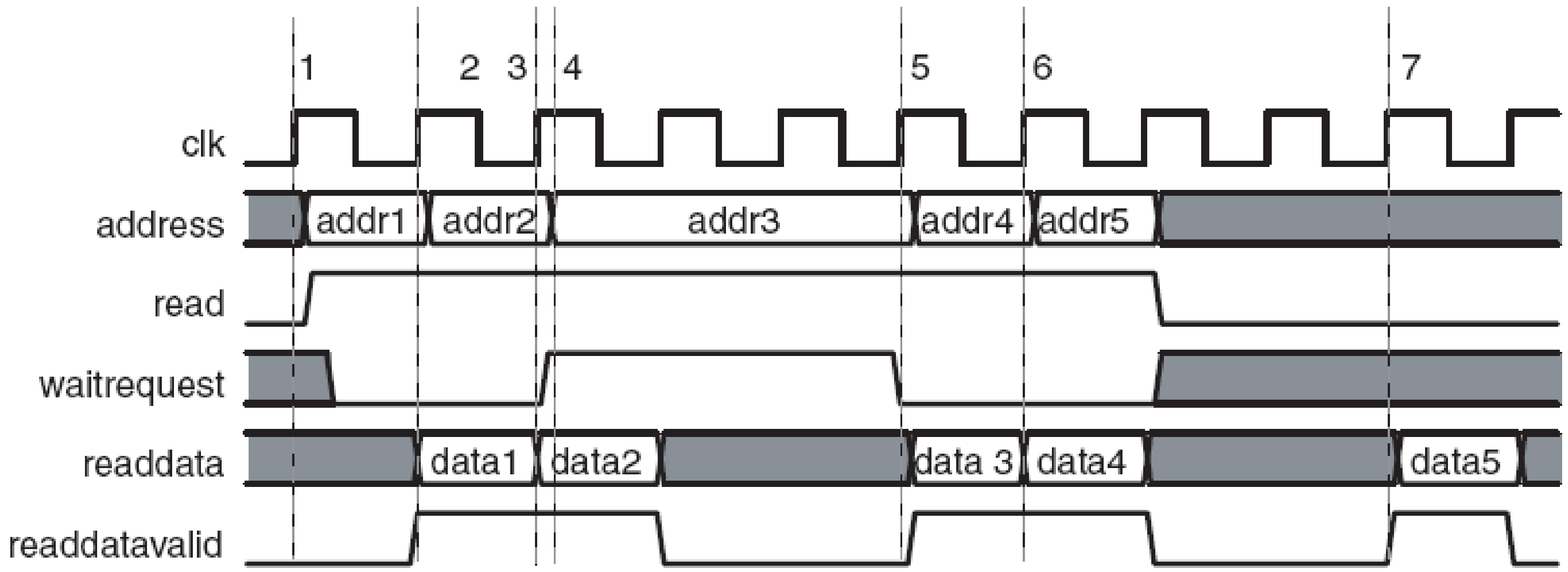
4. This causes the third transfer to be stalled for 2 cycles.

Slave Pipelined Read Transfers with Variable Latency



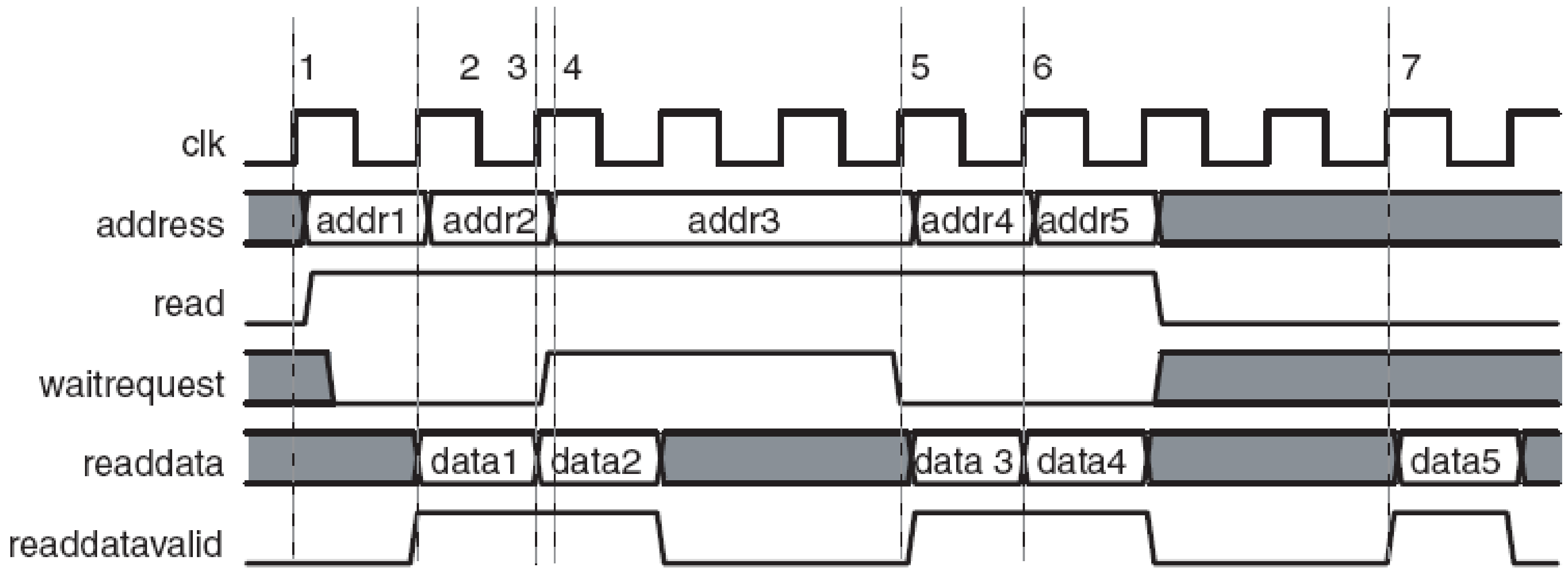
5. The peripheral drives readdatavalid and valid readdata in response to the third read transfer.

Slave Pipelined Read Transfers with Variable Latency



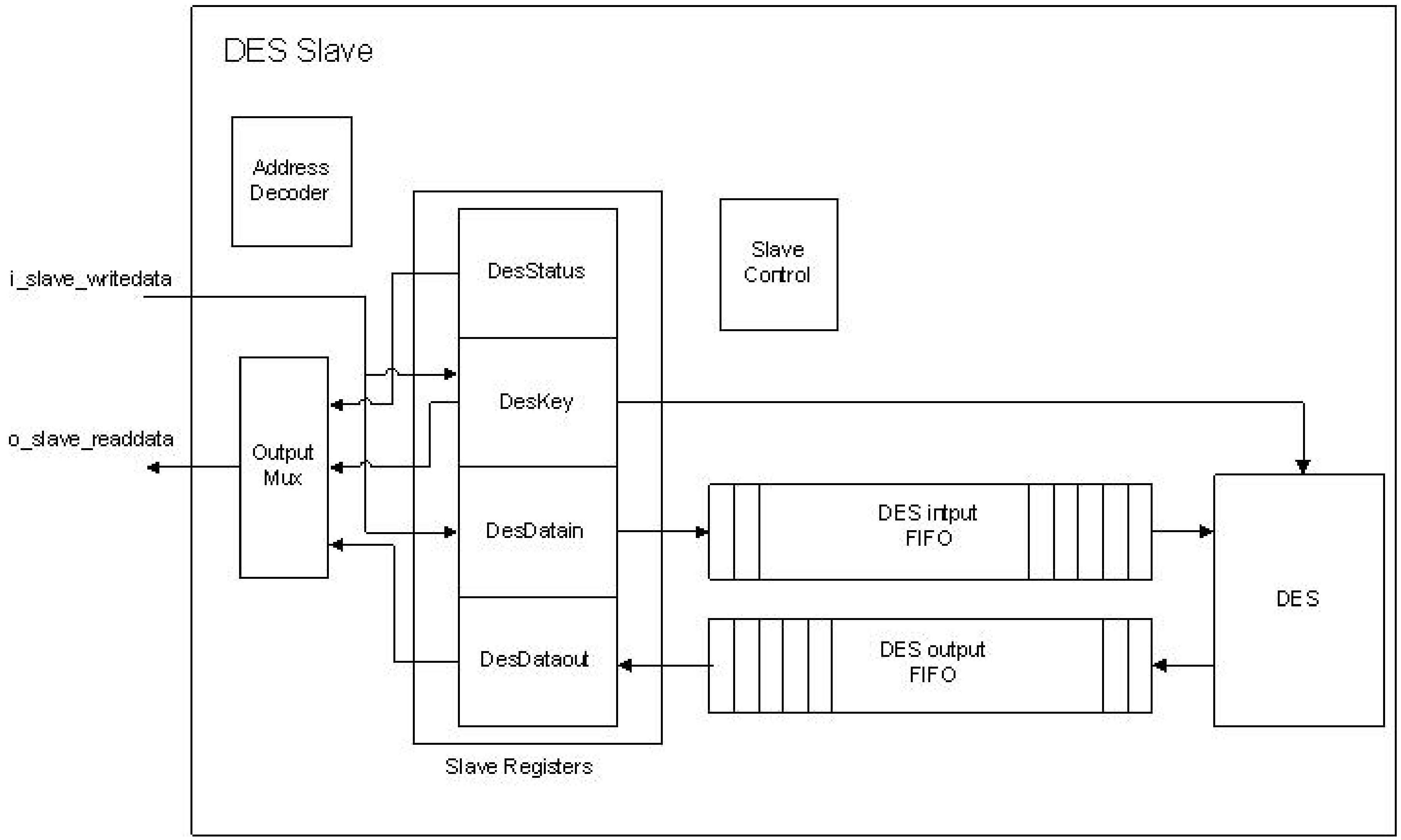
6. The data from transfer 3 is captured by the interconnect to the Master.
6. Simultaneously, addr4 is captured by the slave.

Slave Pipelined Read Transfers with Variable Latency

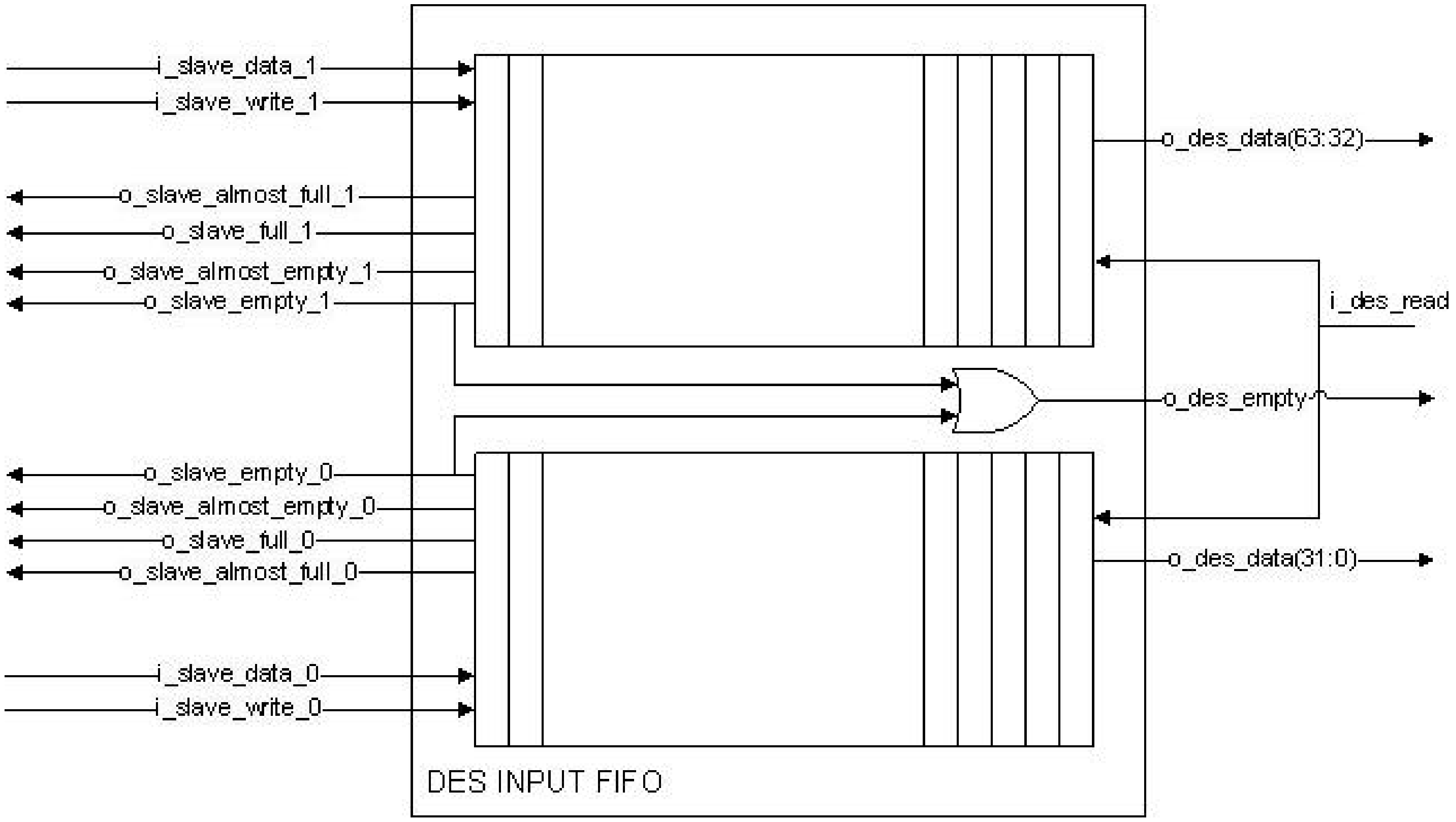


7. data5 is presented with readdatavalid to complete the data phase for the final pending read transfer.

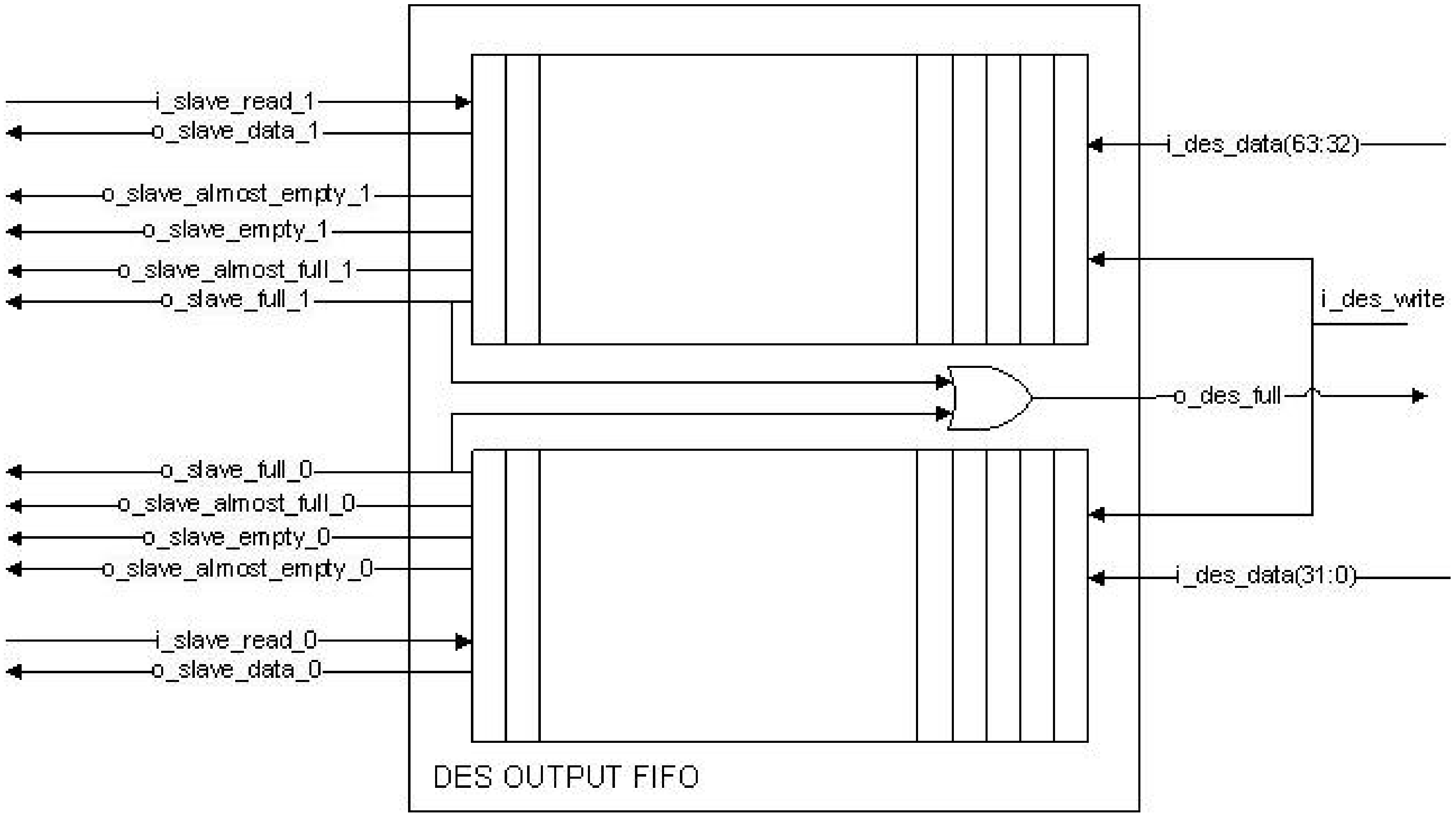
Block Diagram of Avalon-MM DES Slave



DES Input FIFO



DES Output FIFO



Byte enable values for DES Slave

i_slave_byteenable(7 downto 0)	**Access Size (bits)**	**Valid bytes**
“11111111”	64	0,1,2,3,4,5,6,7
“11110000”	32 (MSW)	4,5,6,7
“00001111”	32 (LSW)	0,1,2,3
“00000000”	0	None

Slave Byte Enable Encoding

i_slave_byteenable(7 downto 0)	Slave Word Enable (1 downto 0)
“11111111”	“11”
“11110000”	“10”
“00001111”	“01”
“00000000”	“00”

Slave Address Decoding

i_slave_address(1 downto 0)	**Slave Select Signals (3 downto 0)**
“00”	“0001”
“01”	“0010”
“10”	“0100”
“11”	“1000”

DES Slave Address Map

Address	i_slave_address(1 downto 0)	Access Type	Description
0	“00”	Read-Only*	DES Status
1	“01”	Read-Write	DesKey
2	“10”	Write-Only**	DesDatain
3	“11”	Read-Only*	DesDataout

*Writes to Read-Only registers are ignored

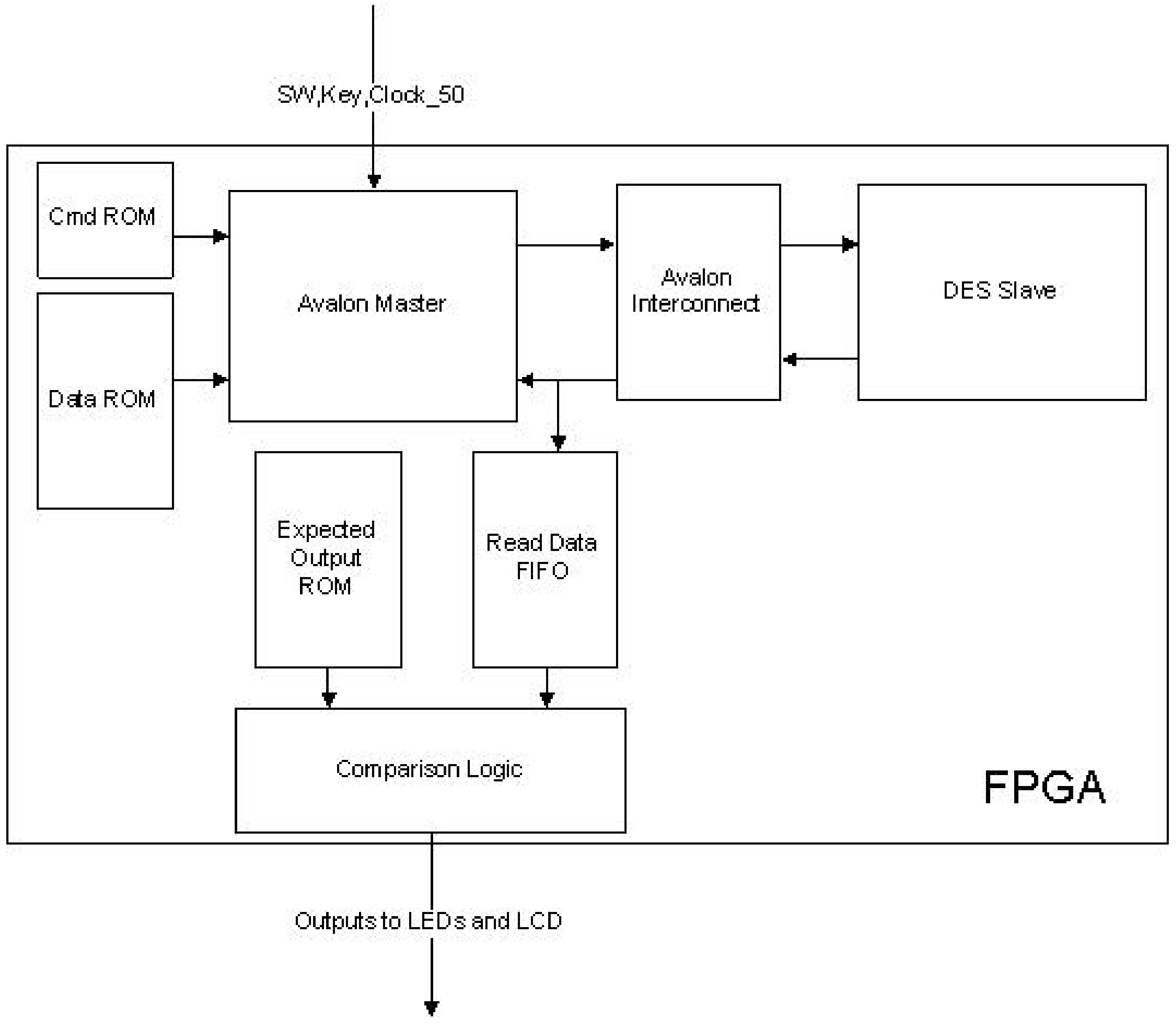
**Reads from Write-Only registers should be all zeroes

More details …

- DES Slave Status Register:
  - See Table 6, page 6 of the lab manual for the slave register details

- Output Multiplexer
  - Uses the decoded slave address and the encoded slave byte enables to select what drives the o_slave_readdata port

Avalon Slave Test System



Tasks

- Step 1: Draw State Diagrams for n = 1 & 2
  - March 10th, 2009
- Step 2: Implement VHDL for all subcomponents
  - March 15th, 2009
- Step 3: Integrate subcomponents for final design
  - March 20th, 2009
- Step 4: Implement design on DE2 board and test
  - March 25th, 2009
- Step 5: Demo your design and submit your report
  - Submit your report at 3:30pm, March 27th, 2009