**sync\_4phase\_handshake\_master**

# Description

Allows messages to be passed synchronously from a master to a slave by using an alternating sequence of request (req) and acknowledgement (ack) levels.

State 1: (~req,~ack) Master writes data to associated data bus

State 2: ( req,~ack) Master asserts req

State 3: ( req, ack) Slave reads data from associated data bus

State 4: (~req, ack) Master deasserts req

# Verilog Template

sync\_4phase\_handshake\_master

sync\_4phase\_handshake\_master\_inst (

.ack(ack), // input

.busy(busy), // output

.clk(clk), // input

.req(req), // output

.strobe(strobe) // input

);

# Specifications

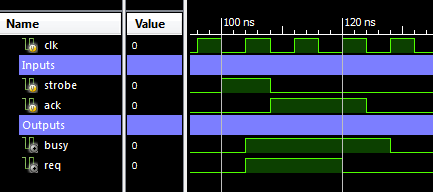
## Inputs

* ack: acknowledgement signal from slave to master
* clk: clock for internal flip-flops
* strobe: begins the four-phase handshake, should be deasserted once busy is asserted, must be deasserted before busy is deasserted or else will start another transaction

## Outputs

* busy: indicates that the four-phase handshake is in operation, i.e. either req or ack is asserted
* req: request signal from master to slave

## Waveform



# Requirements

Language for Synthesis: Verilog 2001

Synthesis Tool: Xilinx XST 13.2

Language for Verification: Verilog 2001

Verification Tool: Xilinx ISIM 13.2

# References

[1] <http://www.cl.cam.ac.uk/~djg11/wwwhpr/fourphase/fourphase.html>

# Contact Info

Nathan Farrington

<http://nathanfarrington.com>