async_4phase_handshake_slave

Description

Allows messages to be passed asynchronously 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

```
async_4phase_handshake_slave
async_4phase_handshake_slave_inst (
    .ack(ack),     // output
    .clear(clear), // input
    .flag(flag),   // output
    .req(req),     // input
    .reset(reset)  // input
);
```

Specifications

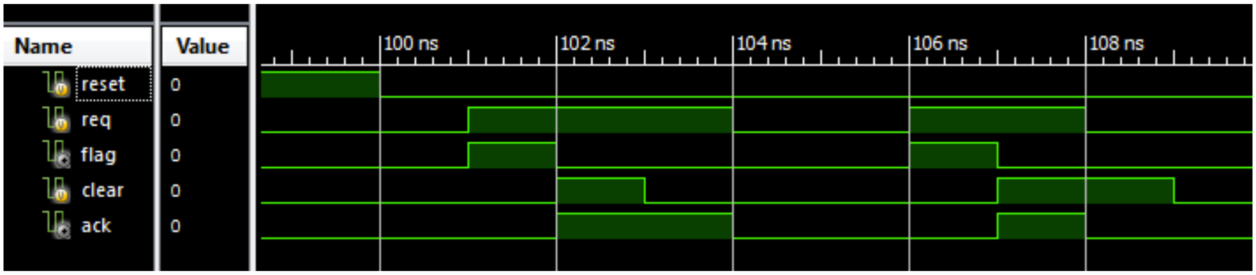
Inputs

- clear: clears the flag and asserts ack, should be deasserted once flag is deasserted, must be deasserted before req is deasserted and then reasserted or else it will spuriously acknowledge a second transaction
- req: request signal from master to slave
- reset: required to put core into known state, clears flag and ack

Outputs

- ack: acknowledgement signal from slave to master, asserted when req is asserted and then clear is asserted, deasserted on reset or when req is deasserted
- flag: indicates that a request from a master has arrive, cleared on reset or when both req and clear are asserted

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