SMII - Serial MII Ethernet interface

Gives and ordinary Ethernet MAC the possibility to use low pin count serial interface towards Ethernet PHY

Brought to You By ORSoC / OpenCores

Legal Notices and Disclaimers

Copyright Notice

General Disclaimer

The Publisher has strived to be as accurate and complete as possible in the creation of this ebook, notwithstanding the fact that he does not warrant or represent at any time that the contents within are accurate due to the rapidly changing nature of information.

The Publisher will not be responsible for any losses or damages of any kind incurred by the reader whether directly or indirectly arising from the use of the information found in this ebook.

This ebook is not intended for use as a source of legal, business, accounting, financial, or medical advice. All readers are advised to seek services of competent professionals in the legal, business, accounting, finance, and medical fields.

No guarantees of any kind are made. Reader assumes responsibility for use of the information contained herein. The Publisher reserves the right to make changes without notice. The Publisher assumes no responsibility or liability whatsoever on the behalf of the reader of this report.

Distribution Rights

The Publisher grants you the following rights for re-distribution of this ebook.

[YES] Can be given away.
[YES] Can be packaged.
[YES] Can be offered as a bonus.
[NO] Can be edited completely and your name put on it.
[YES] Can be used as web content.
[NO] Can be broken down into smaller articles.
[NO] Can be added to an e-course or auto-responder as content.
[NO] Can be submitted to article directories (even YOURS) IF at least half is rewritten!
[NO] Can be added to paid membership sites.
[NO] Can be added to an ebook/PDF as content.
[NO] Can be offered through auction sites.
[NO] Can sell Resale Rights.
[NO] Can sell Master Resale Rights.
[NO] Can sell Private Label Rights.

Table of Contents

Chapter 1 Overview

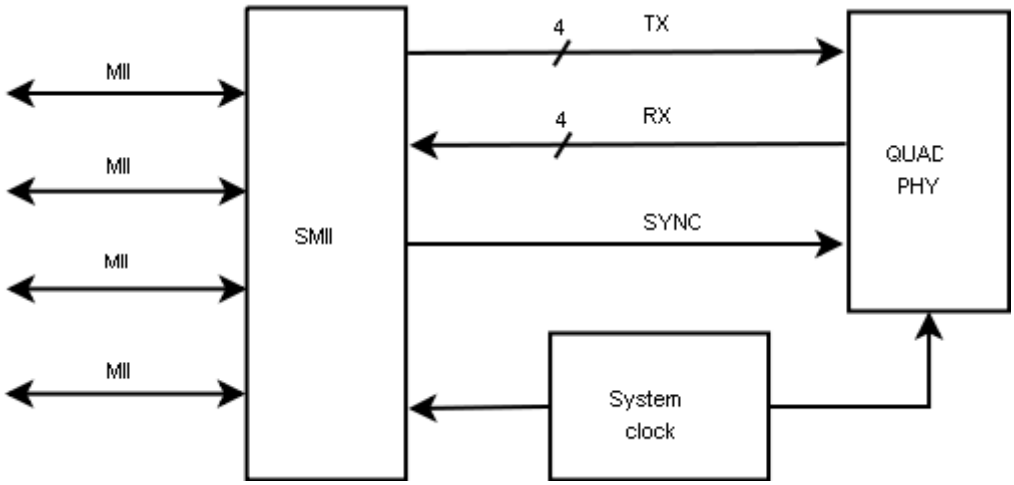
The Serial Media Independent Interface (SMII) is designed to satisfy the following requirements:

- Convey complete MII information between a 10/100 PHY and MAC with two pins per port
- allow multi port MAC/PHY communications with one system clock
- Operate in both half and full duplex
- per packet switching between 10 Mbit and 100 Mbit data rates
- allow direct MAC to MAC communication

SMII is composed of two signals per port, global synchronization signal, and a global 125 MHz reference clock. All signals are synchronous to the clock.

Name	From	To	Use
RX	PHY	MAC	Receive data and control
TX	MAC	PHY	Transmit data and control
SYNC	MAC	PHY	Synchronization
CLOCK	System	MAC & PHY	Synchronization

Typical SMII application:



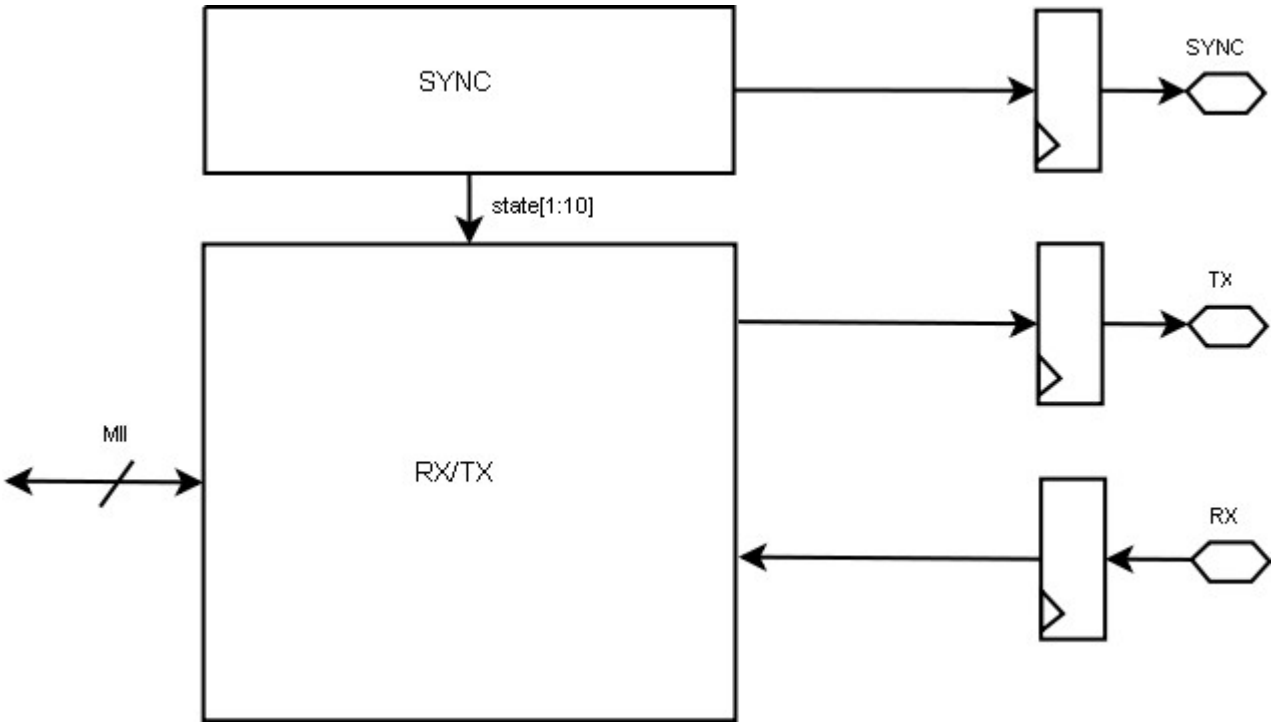
This design is written in Verilog HDL.

Intended use for this IP is to interface Ethernet PHY devices however possible usage includes general purpose IO over SMII. An example is as an interface to an AD converter + FPGA acting as a PHY.

Chapter 1 Internal structure

To be able to support one or multiple SMII channels a structure with one common synchronization module and one transceiver per channel.

To achieve optimal timing IO resources are instantiated at top level.



Instance SYNC can be used to multiple RX/TX modules. Use one RX/TX module per SMII channel. To ensure that DFFs on IO signals are placed in IO block a specific target module can be used. The exact module to use is target dependant. For target independent design there are two modules that can be used, OBUFDFF and IBUFDFF.

If multiple SYNC outputs are to be used use one OBUFDFF instance per output.

Module declaration OBUFDFF

Signal	Type	Comment
PAD	Output	Connect to device pad
D	Input	Connect to internal signal; sync, tx
CLK	Input	Connect to 125 MHz clock source
RST	Input	Asynchronous reset active high

Module declaration IBUFDFF

Signal	Type	Comment
PAD	Input	Connect to device pad
Q	Output	Connect to internal signal; rx
CLK	Input	Connect to 125 MHz clock source
RST	Input	Asynchronous reset active high

Chapter 2 Interconnect signals

Top level signals

Group	Name	From	To	Use
SMII	RX	PHY	SMII	Receive data and control
	TX	SMII	PHY	Transmit data and control
	SYNC	System	SMII	Synchronization optionally one per PHY
	CLOCK	System	SMII	Reference clock
MII one per PHY	TXD[3:0]	MII	SMII	Transmit nibble
	TXEN	MII	SMII	Transmit enable
	TXERR	MII	SMII	Transmit error
	TX_CLK	SMII	MII	Transmit clock
	RXD[3:0]	SMII	MII	Receive nibble
	RXDV	SMII	MII	Receive data valid
	RXERR	SMII	SMII	Receive error
	RX_CLK	SMII	MII	Receive clock
	COLL	SMII	MII	Collision detect
	CRS	SMII	MII	Carrier sense
	SPEED	SMII	MII	Speed, optional 0 – 10Mbit 1 – 100MBit
	DUPLEX	SMII	MII	Duplex, optional 0 – half 1 – full
	LINK	SMII	MII	Link, optional 0 – down 1 - up

Optional signals

Some signals are implementation dependent. To tailor design edit “smii_defines.v”.

Function	Define	Use
SYNC	SMII_SYNC_PER_PHY	If defined enables one sync signal per PHY
SPEED	SMII_SPEED	If defined enables status signal SPEED
DUPLEX	SMII_DUPLEX	If defined enables status signal DUPLEX
LINK	SMII_LINK	If defined enables status signal LINK

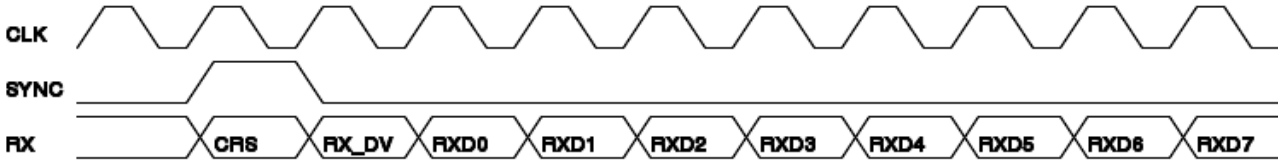
Chapter 3 SMII specification

SMII is composed of two signals per port, a global synchronization signal, and a global 125 MHz reference clock. All signals are synchronous to the clock.

Receive path

Receive data and control information are signaled in ten bit segments. In 100 Mbit mode each segment represents a new byte if data. In 10 mbit mode each segment is repeated ten times.

Segment boundaries are delimited by SYNC. This IP continuously generates a pulse on SYNC every 10 clocks.



RX contains all of the information found on the receive path of the standard MII.

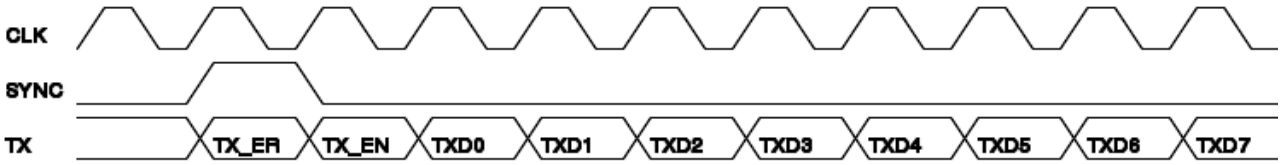
Bit	Purpose
CRS	Carrier Sense – identical to MII, except that it is not an asynchronous signal
RX_DV	Receive Data Valid – identical to MII
RXD7-0	Encoded Data

RXD7-0 are used to convey packet data, RX_ER. And PHY status.

CRS	RX_DV	RXD0	RXD1	RXD2	RXD3	RXD4	RXD5	RXD6	RXD7
X	0	RX_ER	Speed	Duplex	Link	Jabber	Upper nibble	False carrier	1
X	1	One data byte (two MII data nibbles)							

Transmit path

Transmit data and control information are signaled in ten bit segments just like the receive path.



TX contains all of the information found on the transmit path of the standard MII.

Bit	Purpose
TX_EN	Transmit enable – identical to MII
TX_ER	Transmit error – identical to MII
TXD7-0	Encoded Data

TXD7-0 are used to convey packet data, RX_ER. And PHY status.

TX_ER	TX_EN	TXD0	TXD1	TXD2	TXD3	TXD4	TXD5	TXD6	TXD7
X	0	Use to force an error in a direct MAC-MAC connection	1 100 mbit	1 Duplex	1 Link	0 no Jabber	1	1	1
X	1	One data byte (two MII data nibbles)							

Chapter 4 Implementation

Module SYNC

The SMII signal should get high once per frame. This signal is generated from this module. Multiple SMII can share the same SYNC module.

Implemented as a ten bit rotate register. Register content distributed as signal state.

Module RXTX

MII clocks, *mrx_clk* and *mtx_clk*, are being generated phase locked to signal state. In 10 mbps mode this clock is generated every 10:th segment. This signals might need global buffers, target dependant. These signals will be used as clock sources within the MAC. For ACTEL proASIC3 set define ACTEL to automatically insert global clock buffers.

A 4-bit counter is incremented for each segment modulo 10 when *speed* is 10 mbps. In 100 mbps counter is equal to zero. Clock pulses are only generated when counter equals zero.

Receive path is implemented with a local shift register to hold received data. Prior to driving *mrx_clk* high data is output at *mrxd*.

When signal *rx_dv* is active status signals is clocked into registers with state bits used as clock enable.

Transmit path involves a complication since transmit data is received from MAC as nibbles and shifted over SMII as bytes. The first segment with data must contains two nibbles of valid data. Signal *state* synchronize transmit with *sync*. A register, *tx_data_reg* is used as temporary storage. Signal *ao* keeps track of LSB and MSB nibbles and store them *tx_data_reg*. A signal, *tx_data_reg_valid*, goes high when to nibbles have been written. This enable us to start sending to the PHY.

Signal MII collosion *mcoll* is asserted when carrier sense is active simultaneously with *tx_en*.

Chapter 5 Build environment

This IP can be used with the following ethernet MAC from OpenCores

Ethernet MAC 10/100 Mbps

There is a Makefile in the RTL directory which can build a file where both the Ethernet MAC and the SMII are included. This Makefile support 1, 2, 3, 4 or 8 Ethernet channels. The user can easily modify the source file *smii_module_inst.v*. This is a verilog file with preprocessor directive. To build the user must install to standalone Verilog preprocessors.

1. Vpp – verilog preprocessor
   In debian/Ubuntu install with apt-get install vbpp
   Is included in ACTEL libero
2. vppp – verilog PERL preprocessor
   follow install instructions:
   http://search.cpan.org/~wsnyder/Verilog-Perl-3.045/vppp

To build

make all

This generates Verilog files to be included in your SoC top HDL file.

`include “smii_module_inst_4.v”

Chapter 6 Resource usage

Resource usage for different target technologies.

Target	smii_sync	smii_txrx
ACTEL ProASIC3	10 slices	117 slices
ALTERA Cyclone III	comb functions 2 DFFs 10	comb functions 52 DFFs 36

Recommended Resources

ORSoC – http://www.orsoc.se

ORSoC is a fabless ASIC design & manufacturing services company, providing RTL to ASIC design services and silicon fabrication service. **ORSoC** are specialists building complex system based on the OpenRISC processor platform.

Open Source IP – http://www.opencores.org

Your number one source for open source IP and other FPGA/ASIC related information.