

TCP Offload Engine IP core

2010/10/01

Product Specification

Rev 1.11



Design Gateway Co.,Ltd

54 BB Building 13th Fl., Room No.1302 Sukhumvit
21 Rd. (Asoke), Klongtoey-Nua, Wattana,
Bangkok 10110, Thailand
Phone: (+66) 02-261-2277
Fax: (+66) 02-261-2290
E-mail: sales@design-gateway.com
URL: www.design-gateway.com

Features

- High performance hardware accelerator to be inserted between Ethernet MAC and Host Processor.
- Enhanced automatic functions:
 - ✓ TCP sequence number generation
 - ✓ TCP check sum and insertion
 - ✓ Window buffer and data re-send function
 - ✓ IP header check sum generation
 - ✓ IP header generation
- Individual Data FIFO port which automatically combines written data with header and then sends as a TCP packet.
- Support IPv4 (IPv6 support available by optional design service.)
- Support jumbo frame
- No External RAM required.
- Support window size from 4kbyte to 64kbyte by using internal BRAM resource
- Free bit-file for evaluation under Spartan3-A DSP 1800, ML506, ML605 and SP605 board environment. (Ask DesignGateway for other target board bit-file availability.)
- Ready for use by reference design using Xilinx Gigabit Ethernet MAC core and MicroBlaze

Core Facts	
Provided with core	
Documentation	User guide, Data sheet
Design file formats	NGC, VHDL
Constraint file	User constrain file
Verification	VHDL Test Bench
Instantiation Template	VHDL
Reference design & Application notes	ISE/EDK project See reference design guide
Additional items	Demo on ML506, ML605, SP605, Spartan3A DSP 1800 board
Simulation tool used	
Modelsim SE 6.5d	
Support	
Support Provided by Design Gateway Co.,Ltd	

Table 1: Example Implementation Statistics for Xilinx® FPGAs

Family	Example Device	Fmax ¹ (MHz)	Slices ²	IOB ³	GCLK	18kBRAM ⁴	MULT/ DSP48/E	Design Tools
Virtex-5 LX®	XC5VLX50-1FFG1136C	195	545	107	3	4~	0	ISE® 11.5
Spartan-3A DSP®	XC3SA1800D-5FFG676C ⁵	141	1126	107	3	4~	0	ISE® 11.5
Spartan-6 LX®	XC6SLX16-2CSG324C	145	643	107	3	4~	0	ISE® 11.5
Virtex-6 LXT®	XC6VLX240T-1FFG1156C	186	441	107	3	4~	0	ISE® 11.5

Notes:

- 1) **Fmax**: Maximum clock frequency of user data port interface (data_clk)
- 2) **Slices**: Actual slice count depends on percentage of unrelated logic – see Mapping Report File for details
- 3) **IOB**: Assuming all core I/Os and clocks are routed off-chip
- 4) **18kBRAM**: This resource count is based on condition that TX window buffer size is set to 4kbyte and Jumbo Frame is not used. BRAM resource count will depend on the size of TX window and Jumbo Frame usage.
- 5) If Spartan-3 family is the target device, it is strongly recommend to use speed -5 to meet timing constraints.

2010/10/01

1

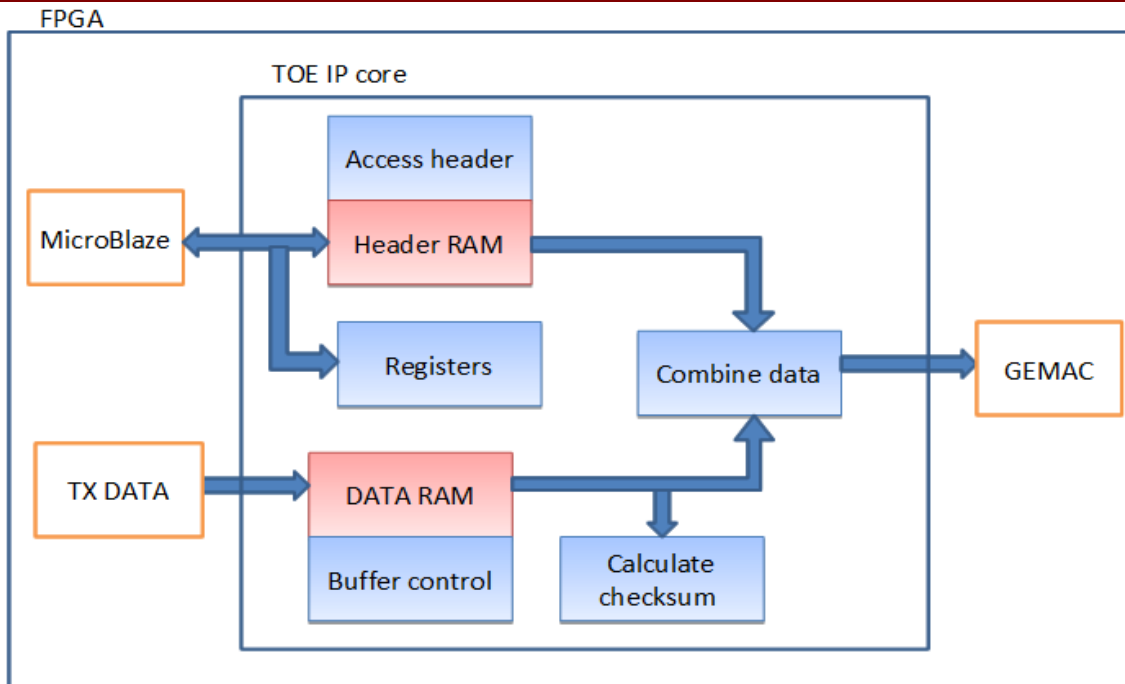


Figure 1 : TOE-IP Block Diagram

Applications

TOE-IP is targeted on such TCP/IP embedded application that requires quite high bandwidth performance. As well-known in the market, TCP/IP is very popular and portable Ethernet protocol, while it requires too heavy process capability so that expensive high-end processor is necessary for embedded TCP/IP system implementation.

TOE-IP provides TCP/IP solution for cost-sensitive system by its hardware logic function. Enhanced TOE-IP logic will process heavy task in the TCP-IP protocol operation required for the Giga-bit Ethernet environment. So that TOE-IP drastically reduces load of the host processor.

TOE-IP, together with low-cost processor or FPGA embedded CPU core such as MicroBlaze, enables users to build high-bandwidth TCP/IP system.

General Description

TOE-IP core is a hardware IP core to be inserted between host processor and the Ethernet MAC. Host Processor Interface is simple enough, i.e. it performs read and write in 1 clock cycle. Data Port is FIFO style interface separated from Host Processor Interface. Data written to this data port is automatically combined with TCP header, and then sent to the Ethernet MAC. TOE-IP core stops transmission in case of the Data RAM becomes empty, and the Data RAM Interface represent full to user logic in case of the Ethernet transmission is stuck and Data RAM becomes full, so user can adjust transfer speed by hardware itself.

Ethernet MAC interface is designed easy to connect with Xilinx Ethernet MAC core as well as other Ethernet MAC IP core or external Ethernet MAC chip.

Main clock (mac_clk) is 125MHz when Ethernet MAC operates as 1000Mbps, 12.5MHz@100Mbps, and [1.25MHz@10Mbps](#), respectively. (note 1)

#note 1 Current core version(Ver1.1) has limitation as follows.

[1] Data port clock (data_clk) and Main clock(mac_clk) must be the same clock.

Functional Description

TOE-IP has 3 functional blocks i.e. Header RAM/Header Access Logic, Data RAM/Buffer Control Logic, and Register for control.

Header RAM/Header Access Logic

This functional block is connected to the Host Processor Interface, and the Host Processor can read from or write to the data in the Header RAM. TOE-IP can send Header RAM data as a TCP/IP packet or can send it together with data in the Data RAM. When TOE-IP send Header RAM data with Data RAM data, Header Access Logic will generate TCP sequence number, TCP checksum, IP Identification number, and IP header checksum automatically, and it will rewrite data contents in the Header RAM. (Note 2)

TOE-IP will update and rewrite Header RAM data when Send Packet Number register value is equal to or more than one and remained data count in the DataRAM exceeds value set in the Send Data Length register.

- TCP sequence number generation

Data value in the Sequence Number register is transferred to the internal variable register of TOE-IP when transmission is reset. In each packet transmission, TOE-IP will add Send Data Length to this register and will write it to the 38-42 byte location in the header RAM.

- TCP checksum calculation

TOE-IP will calculate TCP checksum value by calculation algorithm described in “Checksum of 3.1 Header Format of RFC795” and will write it to the 50-51st byte location in the header RAM. (Note 2)

- IP Identification number generation

Initial IP Identification number is 32768, and TOE-IP will increment this value in each send one packet and will write it to the 18-19th byte location in Header RAM. TOE-IP will use number range 32768 – 65535 for IP Identification number, so any IP packet that does not use this function (i.e. not the high speed data transfer packet) should use IP Identification number range 0-32767.

- IP header checksum calculation

TOE-IP will calculate IP header checksum value by calculation algorithm described in “Checksum in 3.1 Internet Header Format of RFC791” and will write it to the 34-35th byte location in the header RAM. (Note 2)

#Note2 Current core version (v1.1) only supports 14bytes Ethernet header size, 20bytes IP header size, and 20bytes TCP header size.

TOE-IP will update and rewrite Header RAM data when Send Packet Number register value is equal to or more than one and remained data count in the DataRAM exceeds value set in the Send Data Length register.

Data RAM/Buffer Control Logic

When data is written to the Data RAM, TOE-IP will calculate TCP checksum simultaneously. When remained data count in the Data RAM exceeds Send Data Length register value and Send Packet Number register value is equal to or more than one, data is sent with contents of Header RAM. Transmission is stopped while remained data count is less than Send Data Length register value.

Even after data transmission finish, data in the Data RAM still remained inside until Host Processor issue data release, so user can resend remained data without rewriting data when resend is necessary.

TCP Offload Engine IP Core

Data RAM size can be set to 4kbytes, 8kbytes, 16kbytes, 32kbytes or 64kbytes. This size is equal to the TX window size. Because transmission speed will depend on this size, user should optimize this value considering with required performance.

Register of Control TOE-IP

These registers are connected to the Host Processor interface, and send command to Header Access Logic or Buffer Control Logic.

TCP data transmit procedure

TOE-IP supports parts of TCP data packet sending function, it cannot cover whole TCP transmission operation. To execute TCP transmission, external Host Processor is necessary to control TOE-IP by its control firmware. TCP data transmit procedure including control by Host Processor is as follow.

1. Open TCP connection.
2. Set MTU register, Header Length register, and Send Data Length Number register.
3. Write TCP header template to Header RAM. In this case, IP checksum field and TCP checksum field is not necessary because they are filled by TOE-IP core calculation function at each packet transmission.
4. Set Sequence Number register and issue transmit reset. TOE-IP generates Sequence Number automatically in each packet sending.
5. Write data to Data Port interface.
6. Set Send Packet Number register, then TOE-IP sends TCP data packet by this number count.
7. Release Data RAM when Host Processor receive ACK from receiver device. To release RAM, Host Processor subtract initial transmit Sequence Number from received ACK number, and then set this calculated value to the register of control as Transmit Completion Address.
8. In case if resend operation is required due to the duplicate ACK or ACK timeout detection, then Host Processor should set Sequence Number register with the value of resend (i.e. data address for resend plus initial sequence number) and should issue resend operation.
9. If the last packet size didn't match with MSS size, change data length field in IP header and reset Send Data Length register, Sequence Number register, and issue sending. User have to notice that TOE-IP core can not send data packet size less than 200bytes. See reference design guide for details.
10. Close connection after TOE-IP finished all data transmission..

See reference design using MicroBlaze and Xilinx Ethernet MAC core for detail.

Core I/O signals

Descriptions of all signal I/O are provided in Table 2.

Table2 : Core I/O signals

Signal	Dir	Description
Reset		
reset	In	Reset signal for resetting core logic. Active High.

Signal	Dir	Description
Host Processor Interface		
cpu_clk	In	Host Processor Interface clock. TOE-IP accepts asynchronous operation with mac_clk or data_clk.
cpu_din[31:0]	In	Data input from Host Processor.
cpu_dout [31:0]	Out	Data output to Host Processor.
cpu_addr[13:0]	In	Address input from Host Processor.
cpu_we	In	Indicate write operation from Host Processor. Active High.
tx_complete	Out	Interrupt signal that indicates transmit completion or transmit reset. This signal asserts high immediately after sending last packet or completion of transmit reset. This signal keeps high until any write operation to the Transmit Interrupt Clear register.

Signal	Dir	Description
Data Port Interface		
data_clk	In	Data Port Interface clock. Must use same clock as mac_clk.
data_din[7:0]	In	Data input to transmit.
data_en	In	Write enable signal. Active High.
data_full	Out	Indicate full of Data RAM. Active High. User can write 3 bytes after this signal asserts high.

Signal	Dir	Description
Ethernet MAC Interface		
mac_clk	In	Ethernet MAC Interface clock. Input 125MHz when Ethernet MAC operates as 1000Mbps, 12.5MHz@100Mbps, 1.25MHz@10Mbps.
mac_out_en	In	Output enable. Active High. Note that first output data appears without this signal assertion.
mac_not_empty	Out	This signal indicates ready state of frame data output. Active High. When this signal asserts high, user can read frame data. This signal negates low once after complete reading one frame data even if other frame data is available.
mac_dout [7:0]	Out	Frame data output.

Host Processor Interface Register Map

Descriptions of all register map of Host Processor Interface are provided in Table 3.

Table3 : Host Processor Interface Register Map

Register	Address	R/W	Valid bit	Description
Header RAM	0x0000–0x01FF	R/W	[31:0]	Header RAM is mapped here.
Header Length	0x2000	W	[15:0]	Set sum of Ethernet header length, IP header length, and TCP header length. Need to set before sending packet.
MTU	0x2001	W	[15:0]	Set frame length that includes Ethernet header, IP header, TCP header, and user data. MSS is the value subtracted Header Length from MTU. Need to set before sending packet.
Transmit Data Length	0x2002	W	[15:0]	Data Length in one frame. Min : 200 – max : 16000. Data written from data Port is divided by this size automatically. Need to set before writing data to Data Port. When this register is set to zero, Header RAM data only will be transferred. (This is used for controlling connection or ARP, low speed data transmission, etc.) Need transmit reset after writing this register. Don't set the value over MSS.
Sequence Number	0x2003	W	[31:0]	Set initial transmit sequence number. This value is automatically incremented by Transmit Data length in each sending one packet. So it is no need to set again after set once in the normal operation. However in case of resend packet, it is need to set new sequence number before transmit reset.
Transmit Packet Number	0x2004	W	[7:0]	Set packet number for transmission. When data is ready, TOE-IP sends Packets that count is set by this Number register value When Host Processor writes new value to this register before previous packet transmission finish, transmission packet count is added by this new value i.e. not overwritten by new value.
Transmit Complete Address	0x2005	W	[31:0]	Release data in Data RAM by setting Transmit Complete Address counted from head of data as address 0. This address is calculated by initial transmit sequence number subtract from received ACK number. Don't set smaller or 65536 larger address than set before. (It may causes data corruption.) Resend from this address in case of issue transmit reset. Set address subtracted 4G in case of the address over 4GB(4294967296bytes).
Transmit Reset	0x2006	W	[0:0]	When Host Processor set 0x01 to this register, TOE-IP will transfer value in the Sequence Number register to its internal variable, will set the value of Transmit Complete Address register to transmit data buffer address for resending, and will clear transmit FIFO. (Data RAM is not cleared by this operation.) Prior to this Transmit Reset, Host Processor should set sequence number to Sequence Number register and should set data address (usually this value is sequence number for resending subtract by initial sequence number) to Transmit Complete Address. This register can be set even if data transmission is in progress. In this case, after complete sending current packet, discard remained packets and reset transmission. User have to issue transmit reset in case of sending data in the first time or resending or changing transmit Data length register.
Transmit Interrupt Clear	0x2007	W	N/A	Clear transmit complete interrupt by writing operation to this register.
Access Complete	0x2008	R	[31:0]	The value of this register is not 0 that indicates register access is busy. In case of user accesses registers back to back, user have to check this register before next access. Normally this register changes to 0 in some clocks at MAC.CLK after register access exclude transmit reset. In case of transmit reset during sending, this register changes to 0 when complete reset after complete sending current frame.

Host Processor Interface Timig Diagram

Host Processor Interface simply can be read/written in one clock cycle. **In case of access registers addressed upper 0x2000 back to back, user has to check Access Complete Register before next access.**

•Read operation 1

- 1 cpu_we=Low, cpu_addr=0x0010 : read operation at address 0x0010
- 2 output data of address 0x0010 from cpu_dout

•Write operation

- 3 cpu_we=High, cpu_addr=0x0020, cpu_din=0x1000 : write data 0x1000 at address 0x0020

•Read operation 2

- 4 cpu_we=Low, cpu_addr=0x0020 : read operation at address 0x0020
- 5 output data of address 0x0020 from cpu_dout

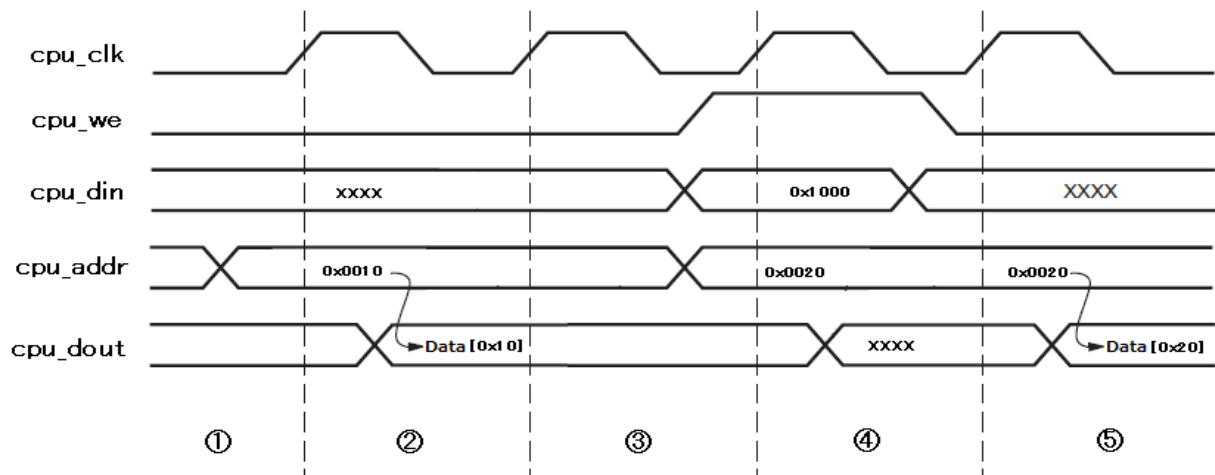


Figure 2 : Host Processor Interface Timing Diagram

Data Port Interface Timig Diagram

Data Port Interface is FIFO style interface. Write operation complete in one clock cycle.

Normal write operation timing diagram is in Figure 3.

- 1 data_en=High, data_din=data1 : write data1 to Data RAM
- 2 data_en=Low : stop writing data
- 3 data_en=High, data_din=data2 : write data2 to Data RAM
- 4 data_en=High, data_din=data3 : write data3 to Data RAM

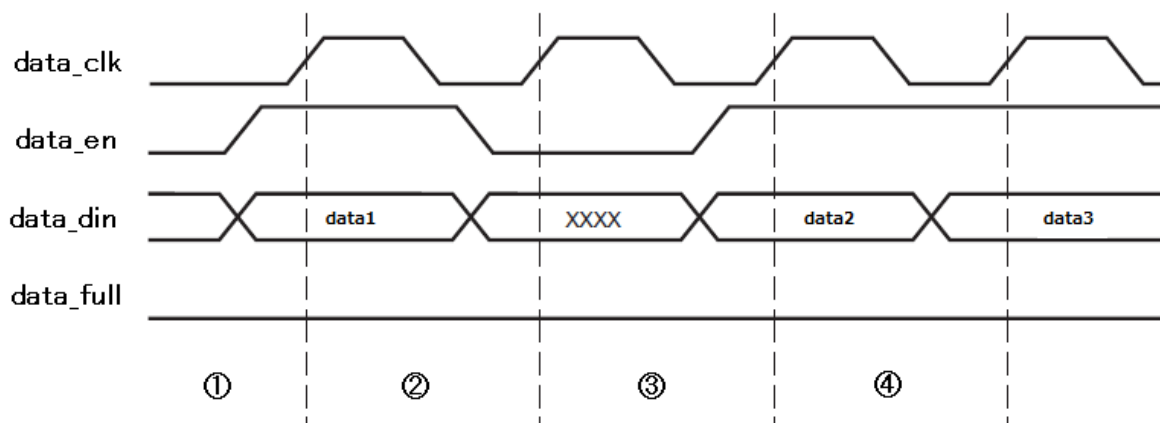


Figure 3 : Data Port Interface Timing Diagram 1

When Data RAM becomes full, data_full signal is asserted that indicates data cannot be written any more.

Write operation timing diagram of data_full signal case is shown in Figure 4.

- 5 data_en=High, data_din=data n : write data n to Data RAM
- User must stop writing data because data_full signal is asserted, but user is allowed to write 3 datas after assertion of data_full signal.
- 6 data_en=Low : stop writing data
- 7 User can write data again because data_full signal is diasserted.
- 8 data_en=High, data_din=data n+1 : Write data n+1 to data RAM.

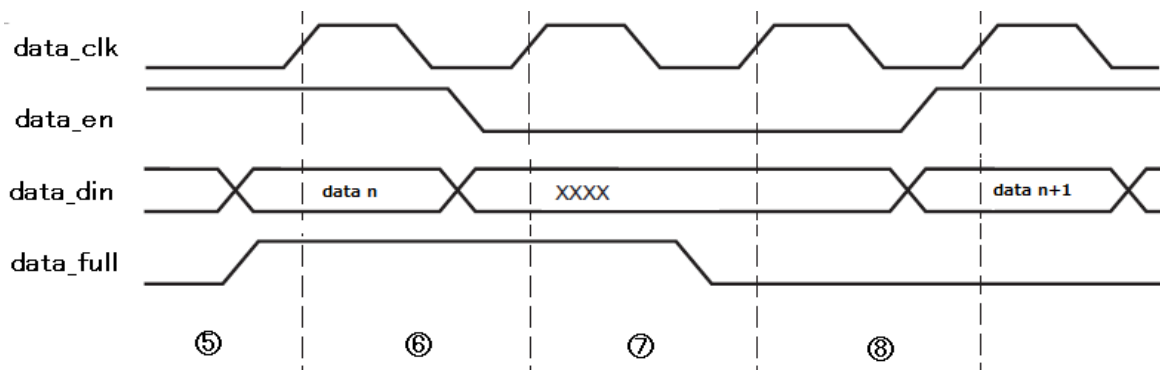
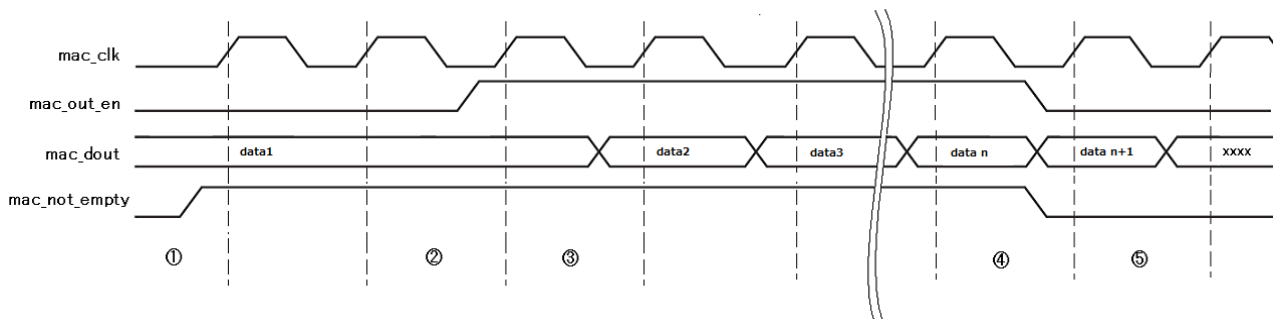


Figure 4 : Data Port Interface Timing Diagram 2

Ethernet MAC Interface timing diagram

Ethernet MAC Interface is designed for easy connection with Xilinx Ethernet MAC core.

- 1 Assertion of `mac_not_empty` signal indicates ready for reading frame. First output data is already provided without assertion of `mac_out_en` signal.
- 2 Next data is provided in next clock by assertion of `mac_out_en` signal.
Note : User have to stay leave first data to Xilinx Ethernet MAC Core until ACK signal is asserted. And user have to send next data in next clock when ACK signal is asserted. So Ethernet MAC Interface is designed as this.
- 3 Data is provided next to next by holding `mac_out_en` signal to high. **User have to hold `mac_out_en` signal until whole one frame data is transferred.**
- 4 Negate `mac_out_en` signal immidiately when `mac_not_empty` signal is negated.
- 5 Last data is transferred. Then, output data of one frame is completed.



Please refer reference design for connection Xilinx Ethernet MAC core.

Verification methods

The SD-IP core functionality was verified by simulation and also proved on real board design by using ML506, ML605, SP605 and Spartan3A DSP 1800 evaluation board.

Recommended Design Experience

The user must be familiar with HDL design methodology as well as instantiation of Xilinx netlists in a hierarchical design environment. And The user must be familiar with EDK, MicroBlaze and Ethernet, TCP/IP for understanding reference design.

Ordering Information

This product is available directly from Design Gateway company. Please contact Design Gateway for pricing and additional information about this product using the contact information on the front page of this datasheet.

Revision history

Revision	Date	Description
1.0	2009/10/19	New release.
1.01	2009/10/20	Modify description of applications.
1.02	2009/11/9	Add Transmit Interrpt Clear register. Update register description.
1.03	2009/12/2	Release as official version.
1.04	2010/1/29	Update register description
1.1	2010/7/21	Accommodate TOE-IP v1.1
1.11	2010/09/24	Add Official logo and page number, adjust header&footer