# Overview

This Hardware NTP [1] Server implementation is based on a FPGA card, VC709 [6] from Xilinx.

* 4 separate NTP servers. Each with 10GB Ethernet interfaces.
* Wire speed performance (10Gbit/s)
* The time reference can be selected from 2 independent 1PPS and 10MHz input pairs.
* Each server supports the following features:
* Optional Symmetric Key authentication (MD5[2], SHA1[3]).
* Internet Protocol versions 4 and 6
* Address Resolution Protocol (ARP[4]) for IPv4.
* Neighbor Discovery (ND[5]) for IPv6.
* Host interface over PCI Express.

A custom interface board [8] that fits in the FMC connector, has been developed for this project. This board conditions the PPS and 10MHz input signals.

# Software Interface

All configuration and status registers are accessible from the host through the PCI Express interface. No network traffic passes through this interface. The NTP server is autonomous after configuration, but software can monitor status registers and take action if an exception occurs.

BAR0 is used to allocate a 1MB window into the memory space of the host. Several methods can be used to access this space, depending of operating system, taste etc.

A device driver for Centos7 has been implemented. This driver allows registers to be accessed from User Space. A few Python scripts are also available to ease the configuration.

No interrupt signal is used, software has to poll status registers.

All registers are 32 bits and it is recommended to access them with this size. I.e. 8/16 bit accesses can get unpredictable results. All addresses given are byte addresses though.

## Base addresses

The following table shows the start addresses for the individual modules in the design.

| Module Name | Offset | Size | Description |
| --- | --- | --- | --- |
| Clock A | 0x00000 | 4K | NTP clock A |
| Clock B | 0x01000 | 4K | NTP clock B |
| Network Path 0 | 0x02000 | 4K | NTP server 0 |
| Network Path 1 | 0x03000 | 4K | NTP server 1 |
| Network Path 2 | 0x04000 | 4K | NTP server 2 |
| Network Path 3 | 0x05000 | 4K | NTP server 3 |
| User registers | 0x06000 | 4K | Misc status registers |
| Etherlite | 0x107E4 | 64K | MDIO control |
| Key mem 0 | 0x20000 | 64K | Key memory for server 0 |
| Key mem 1 | 0x30000 | 64K | Key memory for server 1 |
| Key mem 2 | 0x40000 | 64K | Key memory for server 2 |
| Key mem 3 | 0x50000 | 64K | Key memory for server 3 |

## NTP Clocks

| Register Name | Offset | Description |
| --- | --- | --- |
| TIME\_FRAC | 0x0000 | Fractional part of current NTP time |
| TIME\_SEC | 0x0004 | Seconds part of current NTP time |
| NEW\_SEC | 0x0008 | Seconds part of new NTP time |
| SEC\_CTRL | 0x000C | Time set control |
| LEAP\_SEC | 0x0010 | Leap second (to be added or removed) |
| LEAP\_CTRL | 0x0014 | Leap second control |
| PLL\_STATUS | 0x0018 | PLL synchronization status |

TIME\_FRAC

| Bits | Access | Description |
| --- | --- | --- |
| 31:0 | R | Bits 31:0 of current NTP time |

TIME\_SEC

| Bits | Access | Description |
| --- | --- | --- |
| 31:0 | R | Bits 63:32 of current NTP time |

The current 64 bit NTP time is split into 2 32 bit registers.

NEW\_SEC

| Bits | Access | Description |
| --- | --- | --- |
| 31:0 | R/W | Bits 63:32 of new NTP time |

SEC\_CTRL

| Bits | Access | Description |
| --- | --- | --- |
| 31:1 | Reserved | |
| 0 | RW | Trig: 1= Update NTP time (auto reset) |

By writing bit 0 of the SEC\_CTRL register, the NTP time will be set on the next PPS pulse with the value from the NEW\_SEC register. The fractional bits will be cleared. The update bit will also be cleared.

It is recommended to trigging the update well before the next PPS pulse since there will be latency in register accesses.

LEAP\_SEC

| Bits | Access | Description |
| --- | --- | --- |
| 31:0 | R/W | Leap second |

LEAP\_CTRL

| Bits | Access | Description |
| --- | --- | --- |
| 31:2 | Reserved | |
| 1 | RW | Direction: 1= Add leap second, 0=remove leap second |
| 0 | RW | Trig: 1= Enable leap second (auto reset) |

Leap second is adjusted by writing the time (in seconds) when the leap second will take place.

If direction is 1, this second will be repeated. If direction is 0, then this second will be skipped.

PLL\_STATUS

| Bits | Access | Description |
| --- | --- | --- |
| 32 | R | PLL is locked to 10MHz input |
| 31:10 | Reserved | |
| 9:0 | R | PLL/PPS synchronization status |

This register can be used to monitor the quality of the time reference inputs.

If the PLL is locked and synchronized with the PPS input, then status will be fairly stable.

If the status value is changing rapidly then PPS is probably missing.

Each bit corresponds to 28ps difference between internal and external PPS.

## Network Paths

| Register Name | Offset | Description |
| --- | --- | --- |
| GEN\_CONFIG | 0x0000 | General configuration for the network path /NTP server |
| MAC\_ADDR0\_0 | 0x0004 | MAC address 0, part 0 |
| MAC\_ADDR0\_1 | 0x0008 | MAC address 0, part 1 |
| MAC\_ADDR1\_0 | 0x000C | MAC address 1, part 0 |
| MAC\_ADDR1\_1 | 0x0010 | MAC address 1, part 1 |
| MAC\_ADDR2\_0 | 0x0014 | MAC address 2, part 0 |
| MAC\_ADDR2\_1 | 0x0018 | MAC address 2, part 1 |
| MAC\_ADDR3\_0 | 0x001C | MAC address 3, part 0 |
| MAC\_ADDR3\_1 | 0x0020 | MAC address 3, part 1 |
| IPV4\_ADDR\_0 | 0x0024 | IPv4 address 0 |
| IPV4\_ADDR\_1 | 0x0028 | IPv4 address 1 |
| IPV4\_ADDR\_2 | 0x002C | IPv4 address 2 |
| IPV4\_ADDR\_3 | 0x0030 | IPv4 address 3 |
| IPV6\_ADDR0\_0 | 0x0034 | IPv6 address 0, part 0 |
| IPV6\_ADDR0\_1 | 0x0038 | IPv6 address 0, part 1 |
| IPV6\_ADDR0\_2 | 0x003C | IPv6 address 0, part 2 |
| IPV6\_ADDR0\_3 | 0x0040 | IPv6 address 0, part 3 |
| IPV6\_ADDR1\_0 | 0x0044 | IPv6 address 1, part 0 |
| IPV6\_ADDR1\_1 | 0x0048 | IPv6 address 1, part 1 |
| IPV6\_ADDR1\_2 | 0x004C | IPv6 address 1, part 2 |
| IPV6\_ADDR1\_3 | 0x0050 | IPv6 address 1, part 3 |
| IPV6\_ADDR2\_0 | 0x0054 | IPv6 address 2, part 0 |
| IPV6\_ADDR2\_1 | 0x0058 | IPv6 address 2, part 1 |
| IPV6\_ADDR2\_2 | 0x005C | IPv6 address 2, part 2 |
| IPV6\_ADDR2\_3 | 0x0060 | IPv6 address 2, part 3 |
| IPV6\_ADDR3\_0 | 0x0064 | IPv6 address 3, part 0 |
| IPV6\_ADDR3\_1 | 0x0068 | IPv6 address 3, part 1 |
| IPV6\_ADDR3\_2 | 0x006C | IPv6 address 3, part 2 |
| IPV6\_ADDR3\_3 | 0x0070 | IPv6 address 3, part 3 |
| NTP\_CONFIG | 0x0074 | NTP configuration |
| ROOT\_DELAY | 0x0078 | Root delay |
| ROOT\_DISP | 0x007C | Root dispersion |
| REF\_ID | 0x0080 | Reference ID |
| REF\_TS\_0 | 0x0084 | Reserved |
| REF\_TS\_1 | 0x0088 | Reserved |
| RX\_OFS | 0x008C | Receive time stamp offset |
| TX\_OFS | 0x0090 | Transmit time stamp offset |
| IPV4\_ARP\_PASS\_CNT | 0x0094 | Number of accepted ARP requests |
| IPV4\_NTP\_PASS\_CNT | 0x0098 | Number of accepted IPv4 NTP requests |
| IPV6\_ND\_PASS\_CNT | 0x009C | Number of accepted ND requests |
| IPV6\_NTP\_PASS\_CNT | 0x00A0 | Number of accepted IPv6 requests |
| IPV4\_ARP\_DROP\_CNT | 0x00A4 | Number of dropped ARP requests |
| IPV4\_NTP\_DROP\_CNT | 0x00A8 | Number of dropped IPv4 NTP requests |
| IPV4\_GEN\_DROP\_CNT | 0x00AC | Number of other IPv4 packets dropped |
| IPV6\_ND\_DROP\_CNT | 0x00B0 | Number of dropped ND requests |
| IPV6\_NTP\_DROP\_CNT | 0x00B4 | Number of dropped IPv6 requests |
| IPV6\_GEN\_DROP\_CNT | 0x00B8 | Number of other IPv6 packets dropped |
| BAD\_MAC\_DROP\_CNT | 0x00BC | Number of wrong MAC address |
| ETH\_GEN\_DROP\_CNT | 0x00C0 | Number of wrong Ether type |
| BAD\_IPV4\_DROP\_CNT | 0x00C4 | Number of wrong IPv4 address |
| BAD\_IPV6\_DROP\_CNT | 0x00C8 | Number of wrong IPv6 address |
| BAD\_ETH\_FRAME\_CNT | 0x00CC | Number of bad Ethernet frame (checksum error) |
| TX\_BLOCKED\_CNT | 0x00D0 | Number of times TX is blocked |
| BAD\_MD5\_KEY\_CNT | 0x00D4 | Number of frames with bad MD5 key |
| BAD\_SHA1\_KEY\_CNT | 0x00D8 | Number of frames with bad SHA1 key |
| IPV4\_MD5\_PASS\_CNT | 0x00DC | Number of accepted IPv4 NTP requests with MD5 |
| IPV4\_SHA1\_PASS\_CNT | 0x00E0 | Number of accepted IPv4 NTP requests with SHA1 |
| IPV6\_MD5\_PASS\_CNT | 0x00E4 | Number of accepted IPv6 NTP requests with MD5 |
| IPV6\_SHA1\_PASS\_CNT | 0x00E8 | Number of accepted IPv6 NTP requests with SHA1 |
| BAD\_MD5\_DGST\_CNT | 0x00EC | Number of bad MD5 signatures |
| BAD\_SHA1\_DGST\_CNT | 0x00F0 | Number of bad SHA1 signatures |
| XPHY\_STATUS | 0x00F4 | PHY status |

GEN\_CONFIG

| Bits | Access | Description |
| --- | --- | --- |
| 31:18 | Reserved | |
| 17:16 | R/W | PMA PMD type:  00= 10GBASE\_ER, 01=10GBASE\_LR, 10=10GBASE\_SR |
| 15 | R/W | TX output enable: 1=Enable, 0=Disable |
| 14 | R/W | Clock select: 0=Clock A, 1=Clock B |
| 13:6 | R/W | TTL for IP header |
| 5 | R/W | IP Address check: 1=Enable, 0=Disable |
| 4 | R/W | MAC Address check: 1=Enable, 0=Disable |
| 3 | R/W | IPv6 NTP: 1=Enable, 0=Disable |
| 2 | R/W | IPv6 ND: 1=Enable, 0=Disable |
| 1 | R/W | IPv4 NTP: 1=Enable, 0=Disable |
| 0 | R/W | IPv4 ARP: 1=Enable, 0=Disable |

NTP\_CONFIG

| Bits | Access | Description |
| --- | --- | --- |
| 31:30 | R/W | Leap Indicator |
| 29:27 | R/W | VN |
| 26:24 | R/W | Mode |
| 23:16 | R/W | Stratum |
| 15:8 | R/W | Poll |
| 7:0 | R/W | Precision |

This configuration registers can be used to modify the first 32 bits of the NTP transmit payload. If VN or Poll are 0, then they will be copied from the received payload.

If Mode or Stratum are 0, then they will be 4 (server) and 1 respectively.

RX\_OFS & TX\_OFS

| Bits | Access | Description |
| --- | --- | --- |
| 31:0 | R/W | Offset to be added to RX and TX timestamps. 2's complement. Resolution is 2\*\*-32s (238ps) |

XPHY\_STATUS

| Bits | Access | Description |
| --- | --- | --- |
| 31:5 | Reserved | |
| 4 | R | PCS Block Lock |
| 3 | R | TX Fault |
| 2 | R | Signal detected |
| 1 | R | Module detected |
| 0 | R | QPLL Lock |

This register is the status from the PCS/PMA block and external SFP+ module.

## User Registers

| Register Name | Offset | Description |
| --- | --- | --- |
| VCCINT\_POWER | 0x0000 | Current Supply power (not voltage) |
| VCCAUX\_POWER | 0x0004 |  |
| VCC3v3\_POWER | 0x0008 |  |
| Reserved | 0x000C |  |
| VCC2v5\_POWER | 0x0010 |  |
| VCC1v5\_POWER | 0x0014 |  |
| MGT\_AVCC\_POWER | 0x0018 |  |
| MGT\_AVTT\_POWER | 0x001C |  |
| MGT\_AUXIO\_POWER | 0x0020 |  |
| Reserved | 0x0024 |  |
| MGT\_VCCAUX\_POWER | 0x0028 |  |
| VCC1v8\_power | 0x002C |  |
| DIE\_TEMP | 0x0030 |  |
| BUILD\_TIME | 0x0034 | Build time of Image in Unix epoch format |
| PCIE\_LINK | 0x0038 |  |

## Key Memory

| Register Name | Offset | Description |
| --- | --- | --- |
| KEY0\_0 | 0x0000 | Not used since key id = 0 is illegal |
| KEY0\_1 | 0x0004 | Not used since key id = 0 is illegal |
| KEY0\_2 | 0x0008 | Not used since key id = 0 is illegal |
| KEY0\_3 | 0x000C | Not used since key id = 0 is illegal |
| KEY0\_4 | 0x0010 | Not used since key id = 0 is illegal |
| KEY0\_5 | 0x0014 | Not used since key id = 0 is illegal |
| KEY0\_6 | 0x0018 | Not used since key id = 0 is illegal |
| KEY0\_7 | 0x001C | Not used since key id = 0 is illegal |
| KEY1\_0 | 0x0020 | Bit 31-0 of key |
| KEY1\_1 | 0x0024 | Bit 63-32 of key |
| KEY1\_2 | 0x0028 | Bit 95-64 of key |
| KEY1\_3 | 0x002C | Bit 127-96 of key |
| KEY1\_4 | 0x0030 | Bit 159-128 of key |
| KEY1\_5 | 0x0034 | Bit 191-160 of key |
| KEY1\_6 | 0x0038 | Bit 223-192 of key |
| KEY1\_7 | 0x003C | Bit 255-224 of key |
|  | ... |  |
| KEY1023\_0 | 0xFFE0 | Bit 31-0 of key |
| KEY1023\_1 | 0xFFE4 | Bit 63-32 of key |
| KEY1023\_2 | 0xFFE8 | Bit 95-64 of key |
| KEY1023\_3 | 0xFFEC | Bit 127-96 of key |
| KEY1023\_4 | 0xFFF0 | Bit 159-128 of key |
| KEY1023\_5 | 0xFFF4 | Bit 191-160 of key |
| KEY1023\_6 | 0xFFF8 | Bit 223-192 of key |
| KEY1023\_7 | 0xFFFC | Bit 255-224 of key |

KEY1-KEY1023

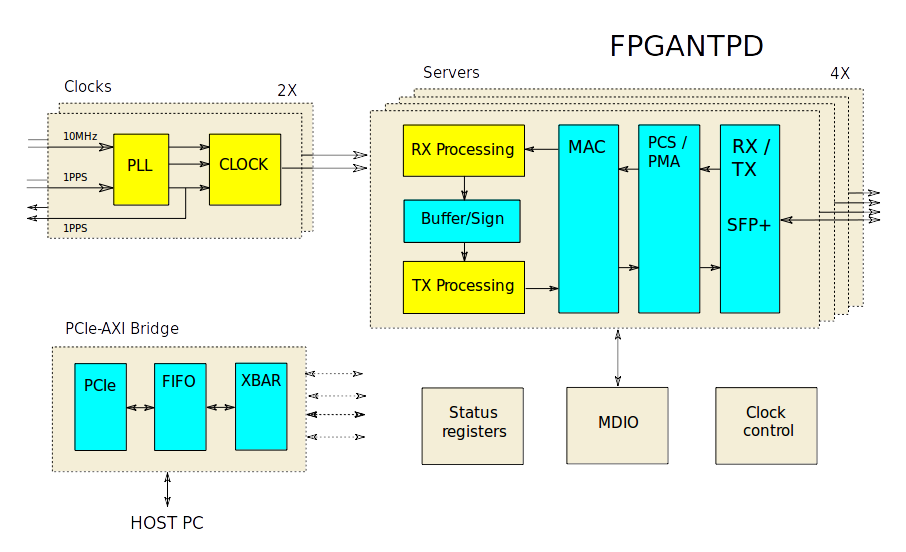
| Bits | Access | Description |
| --- | --- | --- |
| 255 | R/W | Key valid: 1=Valid, 0=Invalid |
| 254 | R/W | Key type: 1=SHA1, 0=MD5 |
| 253:160 | Reserved | |
| 159:0 | R/W | Key bits |

Keys are always 160 bits.

# Implementation Overview

The block diagram of the FPGA design is shown below. The major modules are the two NTP and four NTP servers. There are also a PCIe-AXI backplane and some miscellaneous units.

Everything is hardware based and no software is needed to run the system except for initialization and configuration. The complete design used about 45% of all logic resources in the FPGA.

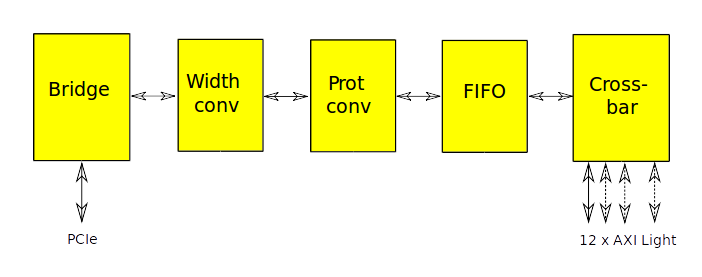
Figure 1: Block diagram

## External Interfaces

1. 2 x 10MHz reference frequencies
2. 2 x 1PPS reference pulses (one for each frequency)
3. 4 x 10Gb Ethernet over SFP+ cages
4. 8 Line 5GT/s PCI Express for host connection

## PCIe-AXI Backplane

The PCIe to AXI backplane is built of Xilinx IP blocks which has been individually configured and integrated into a module.

Figure 2: PCIe-AXI backplane

## Interfaces

1. 8-Lane PCI Express
2. 12 x 32bit AXI Lite slaves

## PCIe to AXI bridge

This Xilinx IP converts the PCI Express transfers into AXI Light transactions.

Only one address range of 1 MByte is defined. This is directly mapped into the AXI address range.

## Data width converter

The 256 bit wide AXI bus from the bridge is reduced to 32 bits.

## Protocol converter

The AXI transactions are converted into the simplified AXI Light protocol.

## FIFO

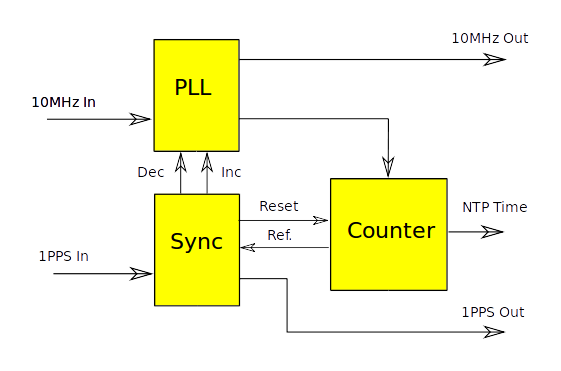
The FIFO buffers data in order to avoid stall in bursts.

## Crossbar

The AXI crossbar connects the single master interface into 12 slaves.

# NTP clocks

The two independent NTP clocks are identical and are depicted in Figure.

Figure 3: NTP Clock

## Interfaces

1. 10MHz input frequency reference.
2. 1PPS input time reference.
3. 10MHz output.
4. 1PPS output
5. Time value in 64 bit NTP Timestamp Format.
6. AXI slave interface for configuration

## NTP counter

The counter split into a 27bit fractional part and a 32bit seconds part. The 27bit part wraps every 128000 cycles and then seconds are normally incremented.

The fractional counter will be reset by the sync block if out of phase with the 1PPS input.

The 27bit fractions are converted into 32bits by multiplication.

The seconds counter can be set from a shadow register when the next second occurs. Leap second adjusted by comparing the content of a leap register with the counter. Every new second the 32bit counter will be incremented by 1, 2 or 0 depending on match between the leap register and the leap mode.

## PLL

The PLL is a Xilinx MMCM clock primitive. It is configured to generate a 128MHz frequency from the 10MHz input. A 10MHz reference output is also created.

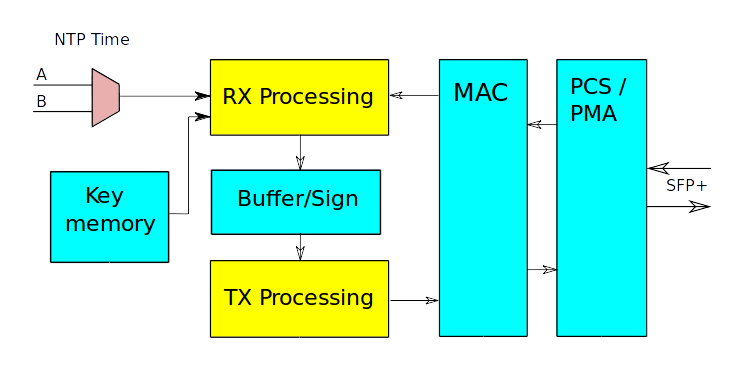
The phase of the 128MHz will be adjusted and aligned (locked) with the rising edge of the 1 PPS input.

## PLL Sync

The Sync block resets the fractional part of the NTP counter if needed. It also sends phase increment/decrement pulses to the PLL if the rising edges of the internal PPS reference are unaligned with the 1PPS input.

# Network Paths

Each Network path contains the complete chain from reception of a Ethernet packet until a new one leaves the design in serial form. One path is more or less a complete NTP server except for the time keeping. All blocks runs in a pipelined (parallel) fashion. This allows wire speed performance, although with latency.

Figure 4: Network path

## Interfaces

1. 10Gb Ethernet from SPF+ cage
2. AXI bus for configuration
3. 2 x NTP Time A and B

## PCS/PMA

The Physical Coding Sublayer/Physical Medium Attachment is provided by XILINX.

## MAC

The Media Access Controller is an open source design[7] from OpenCores.org.

## RX

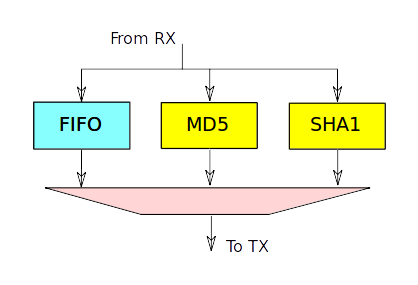
The RX module is state machine that processes Ethernet frames in 64bit chunks as they arrive from the MAC. When a new NTP request arrives a receive time stamp is recorded.

The state machine parses/decodes the frames and checks for valid format and content (addresses etc) . It also extracts information that should be included in the response packet.

When a packet has been completely received and is consider OK, its response is put in an output buffer. If the received packet was a signed NTP request its signature and key will be included as well. This is because signing will be checked in the signing module. The key will be fetched from the key memory in parallel with parsing

## Buffer and Signing

In this module the response can take one of three routes. Signed packets are sent to corresponding sub modules. Other packets (unsigned NTP, ARP, ND) are put into the bypass FIFO. On the output side, signed packets will have priority over others in order to maintain the transmit time stamp accuracy. Unsigned packets will be stamped when they are sent to TX, signed packets before they are signed and their timestamps will be fixed then.

Figure 5: Signing and buffering

### FIFO

The bypass buffer is a FIFO from the Xilinx IP library.

### MD5

The MD5 hash calculation [2] is partially unrolled and pipelined. The signature is calculated over the 160 bit key and 384 bit payload. Therefore two chunks have to be processed. Each chunk takes 64 cycles, in total 128 cycles. A pipeline can process a new packet every 8 clock cycles which is fast enough for wire speed. One pipe is used for checking the signature of a received packet, another for signing the response. Total latency is 256 cycles

### SHA1

The SHA1 [3] is implemented similarly to the MD5 calculation. However each step takes 10 cycles which is still good enough. Latency is lower, in total 160 cycles.

## TX

This block formats the different responses into proper network packets.

The packets are sent to the MAC in 64 bit chunks every clock cycle.

Checksums are calculated over several clock cycles and parallel with transmission. This improves timing but without extra latency.

## Key memory

This memory is implemented with a Xilinx two port memory. One port is for reading the key from the RX block. The other port is connected to the AXI bus

# Misc modules

## Etherlite

The Xilinx Etherlite IP can be used to configure the PHYs. Only the MDIO registers are used and other logic is trimmed during synthesis. Normally configuration is not needed.

## User registers

This module is based on the similar module in the VC709 reference design. It contains hardware to monitor the power controller devices on the board as well as die temperature and voltage rails within the FPGA.

## Clock control

This block is also taken from the reference design and is used to configure a 156MHz low jitter reference clock source on the board. It is autonomous and contains no external registers.

# Clock domains

The design has several different and clock domains. Details on on the clock sources are in the user guide for the FPGA board [6]. All listed clocks are considered asynchronous and appropriated logic for managing clock domain crossings are inserted.

The PCIe block is clocked by the 100MHz reference clock in the PCIe interface, i.e. driven by the host computer.

The AXI bus clock is 125MHz and generated in the the PCIe-AXI bridge from the PCIe clock.

Each NTP “clock” block generates each a 128MHz reference from their 10MHz inputs respectively.

The network paths shares a common 156.25MHz jitter attenuated clock coming from the source on the board.

The 50MHz clock is derived from the 200MHz system clock on the board.

# References

1. NTP <https://tools.ietf.org/html/rfc5905>
2. MD5 <https://tools.ietf.org/html/rfc1321>
3. SHA1 <https://tools.ietf.org/html/rfc3174>
4. ARP <https://tools.ietf.org/html/rfc5494>
5. ND <https://tools.ietf.org/html/rfc4861>
6. VC709 <http://www.xilinx.com/products/boards-and-kits/dk-v7-vc709-g.html>
7. 10GE MAC <http://opencores.com/project,xge_ll_mac>
8. NTP interface board <TBD>