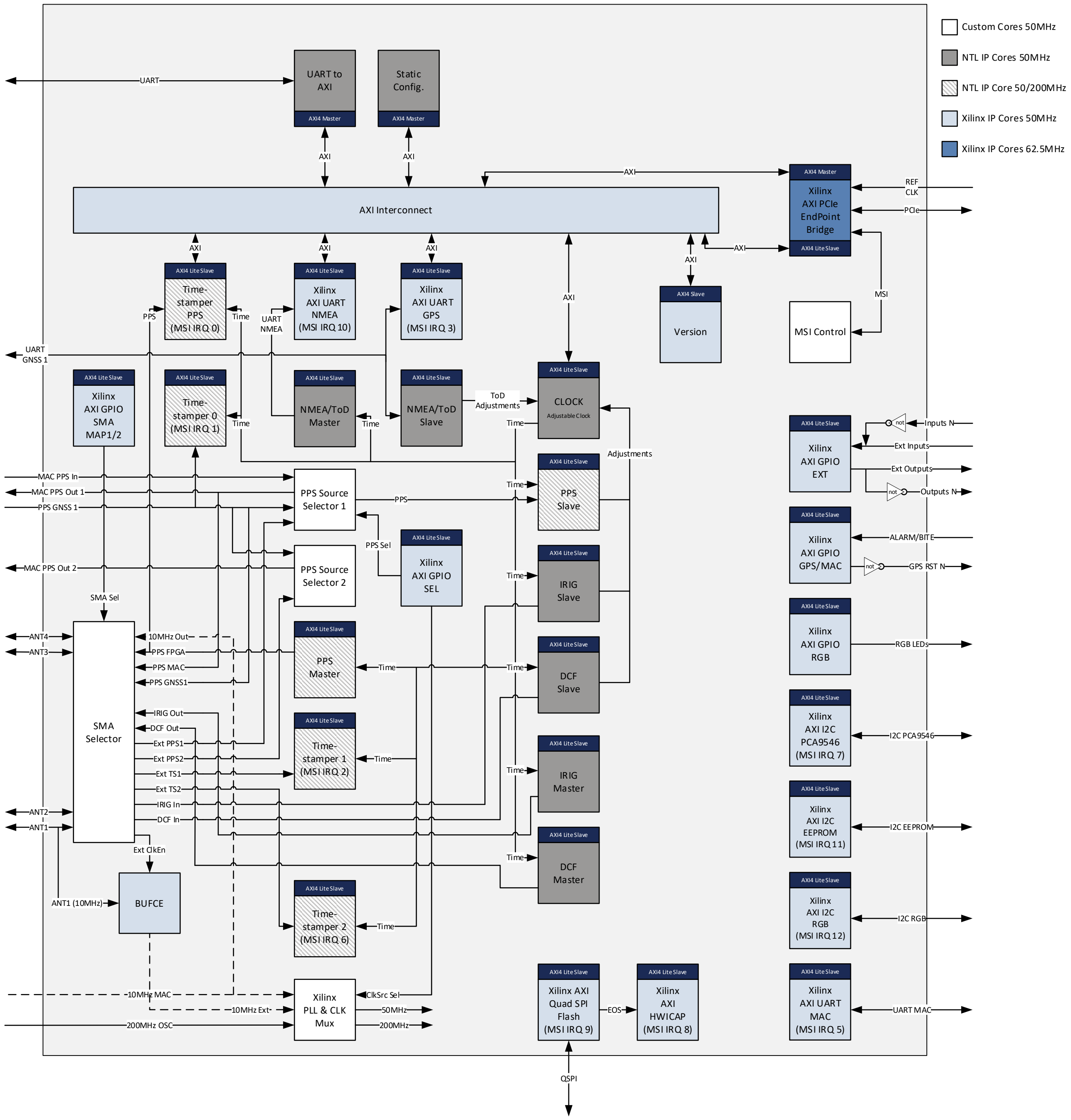
1 Design Overview



To not overload the drawing not all AXI and IRQ connections were added.

The TimeCard runs partially from the 200MHz local oscillator. The NetTimeLogic cores with all the high precision parts are running based on the 10MHz MAC clock source.

1.1 Status LEDs

At the moment the Status LEDs are not connected to the AXI GPIO Ext and they are used directly by the FPGA.

LED1: Alive LED of the FPGA internal Clock (50MHz)
LED2: Alive LED of the PCIe clocking part (62.5MHz)
LED3: PPS of the FPGA (Time of the Local Clock via PPS Master)
LED4: PPS of the MAC (differential inputs from the MAC via diff-buffer)

The RGB LEDs (SMA1-4 and GNSS) can be controlled via I2C.

1.2 AXI Address Mapping

The AXI interconnect has three masters which all have access to the slaves. One is the AXI PCIe interface and the other UART to AXI have all the time access. The Static Configuration only does a basic configuration after reset.

The AXI Slave Interfaces have following addresses:

AXI Slave interface	Slave	Offset Address	High Address
S_AXI_CTL	AXI PCIe Control	0x0001_0000	0x0001_0FFF
axi4l_slave	Version	0x0002_0000	0x0002_0FFF
S_AXI	AXI GPIO Ext	0x0010_0000	0x0010_FFFF
S_AXI	AXI GPIO GPS/MAC	0x0011_0000	0x0011_FFFF
S_AXI	AXI GPIO SEL	0x0013_0000	0x0013_FFFF
S_AXI	AXI GPIO SMA MAP1	0x0014_0000	0x0014_FFFF
S_AXI	AXI I2C PCA9546	0x0015_0000	0x0015_FFFF
S_AXI	AXI UART 16550 GPS	0x0016_0000	0x0016_FFFF
~~S_AXI~~	~~AXI UART 16550 GPS2~~	~~0x0017_0000~~	~~0x0017_FFFF~~
S_AXI	AXI UART 16550 MAC	0x0018_0000	0x0018_FFFF
S_AXI	AXI UART 16550 NMEA	0x0019_0000	0x0019_FFFF
S_AXI	AXI I2C EEPROM	0x0020_0000	0x0020_FFFF
S_AXI	AXI I2C RGB	0x0021_0000	0x0021_FFFF
S_AXI	AXI GPIO SMA MAP2	0x0022_0000	0x0022_FFFF
S_AXI	AXI GPIO RGB	0x0023_0000	0x0023_FFFF
S_AXI_LITE	AXI HWICAP	0x0030_0000	0x0030_FFFF
AXI_LITE	AXI Quad SPI Flash	0x0031_0000	0x0031_FFFF
axi4l_slave	NTL Adj. Clock	0x0100_0000	0x0100_FFFF
axi4l_slave	NTL Signal TS0 (GNSS)	0x0101_0000	0x0101_FFFF
axi4l_slave	NTL Signal TS1	0x0102_0000	0x0102_FFFF

axi4l_slave	NTL PPS Master	0x0103_0000	0x0103_FFFF
axi4l_slave	NTL PPS Slave	0x0104_0000	0x0104_FFFF
axi4l_slave	NTL TOD Slave	0x0105_0000	0x0105_FFFF
axi4l_slave	NTL Signal TS2	0x0106_0000	0x0106_FFFF
axi4l_slave	NTL IRIG Slave	0x0107_0000	0x0107_FFFF
axi4l_slave	NTL IRIG Master	0x0108_0000	0x0108_FFFF
axi4l_slave	NTL DCF Slave	0x0109_0000	0x0109_FFFF
axi4l_slave	NTL DCF Master	0x010A_0000	0x010A_FFFF
axi4l_slave	NTL TOD Master	0x010B_0000	0x010B_FFFF
axi4l_slave	NTL Signal TS PPS	0x010C_0000	0x010C_FFFF

NOTE:
The Version Slave has one single 32-Bit Register. The upper 16 Bits show the version number of the golden image and the lower 16 Bits the version number of the regular image.

E.g.
Register 0x0200_0000 of the Golden image shows: 0x0001_0000
Register 0x0200_0000 of the Regular image shows: 0x0000_0003

1.3 Clock Adjustment Sources

The Adjustable Clock supports different adjustment sources:

- PPS (with GNSS) (register value 3)
- IRIG (register value 2)
- DCF (register value 6)

By default, PPS with GNSS is selected. The other time sources can be selected via the NTL Adj. Clock ClockSelect Register (Offset: 0x0100_0008).

1.4 Message-Signaled Interrupt Mapping

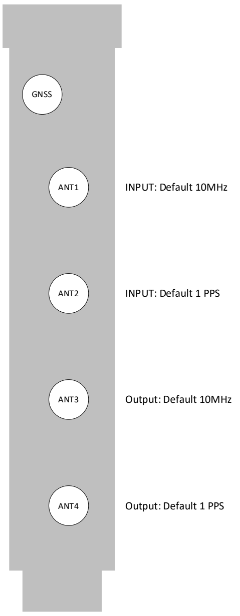
The interrupts in the design are connected to the MSI Vector of the AXI Memory Mapped to PCI Express Core via a MSI controller. The PCI Express Core needs to set the MSI_enable to ‘1’. The MSI controller sends INTX_MSI Request with the MSI_Vector_Num to the PCI Express Core and with the INTX_MSI_Grant the interrupt is acknowledged.

MSI Number	Interrupt Source
0	NTL Signal TS PPS
1	NTL Signal TS0
2	NTL Signal TS1
3	AXI UART 16550 GPS

~~4~~	~~AXI UART 16550 GPS2~~
5	AXI UART 16550 MAC
6	NTL Signal TS2
7	AXI I2C PCA9546
8	AXI HWICAP
9	AXI Quad SPI Flash
10	AXI UART 16550 NMEA
11	AXI I2C EEPROM
12	AXI I2C RGB

1.5 Connectors (SMA) / SMA Connection Matrix

The SMA connectors have following default mapping:



ANT1:
10MHz Clock input
ANT2:
PPS Input
ANT3:
10MHz Clock output (looped 10MHz Clock →MAC RF OUT (after clock buffer)
ANT4:
PPS Output from the PPS Master

This default mapping and the direction can be changed via AXI GPIO SMA MAP1 and AXI GPIO SMA MAP2.
GPIO is used to select the inputs and GPIO2 to select the outputs.

<u>AXI GPIO SMA MAP1</u>
Mapping of Input 1 (ANT1) with GPIO Bit 15 downto 0:

0	10MHz Clock from SMA connector **(default)**
1	External PPS 1 (goes to PPS Source Selector 1)
2	External PPS 2 (goes to PPS Source Selector 2)
4	Signal goes to TS1
8	Signal goes to TS2
16	Signal goes IRIG Slave
32	Signal goes to DCF Slave
32768	Input Enable **(default 1 → ANT1 is input)**

The input can be mapped to several sinks. As example with the value 5 the input is mapped as external PPS 1 and additionally it goes to TS1.

NOTE:

To use ANT1 as a system clock source only 0 is possible.

Mapping of Input 2 (ANT2) with GPIO Bit 31 downto 16:

0	no mapping of ANT2
1	External PPS 1 (goes to PPS Source Selector 1) **(default)**
2	External PPS 2 (goes to PPS Source Selector 2)
4	Signal goes to TS1
8	Signal goes to TS2
16	Signal goes IRIG Slave
32	Signal goes to DCF Slave

32768 Input Enable **(default 1 → ANT2 is input)**

The input can be mapped to several sinks. As example with the value 9 the input is mapped as external PPS 1 and additionally it goes to TS2.

NOTE:

If a lower input has already a mapping to a sink the Input2 mapping will be ignored.

Mapping of Output 3 (ANT3) with GPIO2 Bit 15 downto 0:

0	10MHz Clock Output from the MAC **(default)**
1	PPS from the FPGA (PPS Master)
2	PPS from the MAC
4	PPS from GNSS 1
8	~~PPS from GNSS 2~~
16	IRIG Master Output
32	DCF Master Output
63	GND

32768 Output Enable **(default 1 → ANT3 is output)**

Every other value maps the 10MHz Clock Output from the MAC to the output

Mapping of Output 4 (ANT4) with GPIO2 Bit 31 downto 16:

0	10MHz Clock Output from the MAC
1	PPS from the FPGA (PPS Master) **(default)**
2	PPS from the MAC
4	PPS from GNSS 1
8	~~PPS from GNSS 2~~
16	IRIG Master Output

32 DCF Master Output
63 GND

32768 Output Enable **(default 1 → ANT4 is output)**

Every other value maps the PPS from the FPGA (PPS Master) to the output

AXI GPIO SMA MAP2

Mapping of Input 3 (ANT3) with GPIO Bit 15 downto 0:

0 no mapping of ANT3
1 External PPS 1 (goes to PPS Source Selector 1)
2 External PPS 2 (goes to PPS Source Selector 2)
4 Signal goes to TS1
8 Signal goes to TS2
16 Signal goes IRIG Slave
32 Signal goes to DCF Slave

32768 Input Enable **(default 0 → ANT3 is output)**

The input can be mapped to several sinks. As example with the value 5 the input is mapped as external PPS 1 and additionally it goes to TS1.

Mapping of Input 4 (ANT4) with GPIO Bit 31 downto 16:

0 no mapping of ANT2
1 External PPS 1 (goes to PPS Source Selector 1)
2 External PPS 2 (goes to PPS Source Selector 2)
4 Signal goes to TS1
8 Signal goes to TS2
16 Signal goes IRIG Slave
32 Signal goes to DCF Slave

32768 Input Enable **(default 0 → ANT4 is output)**

The input can be mapped to several sinks. As example with the value 9 the input is mapped as external PPS 1 and additionally it goes to TS2.

NOTE:

If a lower input has already a mapping to a sink the Input4 mapping will be ignored.

Mapping of Output 1 (ANT1) with GPIO2 Bit 15 downto 0:

0 10MHz Clock Output from the MAC
1 PPS from the FPGA (PPS Master)

2 PPS from the MAC
4 PPS from GNSS 1
8 ~~PPS from GNSS 2~~
16 IRIG Master Output
32 DCF Master Output
63 GND

32768 Output Enable **(default 0 → ANT1 is input)**

Every other value maps the 10MHz Clock Output from the MAC to the output

Mapping of Output 2 (ANT2) with GPIO2 Bit 31 downto 16:

0 10MHz Clock Output from the MAC
1 PPS from the FPGA (PPS Master)
2 PPS from the MAC
4 PPS from GNSS 1
8 ~~PPS from GNSS 2~~
16 IRIG Master Output
32 DCF Master Output
63 GND

32768 Output Enable **(default 0 → ANT2 is input)**

Every other value maps the PPS from the FPGA (PPS Master) to the output

1.6 GPIO Mapping

GPIO (Offset 0x0000)
GPIO2 (Offset 0x0008)

Inputs or Outputs might high or low active for the external signal level. The inversion is done in the FPGA. On the AXI GPIO always active high is used. The inverted signals are underlined in the tables below.

AXI GPIO Ext

REG	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO (in)	-	Board Rev2	Board Rev1	Board Rev0	-	-	-	-
GPIO2 (out)	-	-	-	EEPROM WP	LED4 (NC atm)	LED3 (NC atm)	LED2 (NC atm)	LED1 (NC atm)
REG	**Bit15**	**Bit14**	**Bit13**	**Bit12**	**Bit11**	**Bit10**	**Bit9**	**Bit8**

GPIO (in)	-	-	-	Bno Int	Lm75B Int3	Lm75B Int2	Lm75B Int1	Sht3x Alert
GPIO2 (out)	-	Debug USB MuxSel	SMBUS I2C BufEn	Bno Boot	Bno Rst	Sht3x Rst	RGB Shut Down	Pca9546 Rst

AXI GPIO GPS/MAC

REG	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO (in)	-	-	-	-	-	-	MAC BITE	MAC ALARM
GPIO2 (out)	-	-	-	-	-	-	-	GPS RST

AXI GPIO SEL

REG	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO (in)	-	-	Clock MAC selected	Clock SMA selected	-	PPS GNSS detected	PPS MAC detected	PPS SMA detected
GPIO2 (out)	-	-	Select MAC Clock (default 0)	Select SMA Clock (default 0)	-	-	Select PPS Sources (default 0) 0: PPS Source is automatically selected (details in chapter 1.9) 1: Force SMA: PPS Slave and MAC PPS Source is the SMA connector 2: Force MAC: PPS Slave source is the MAC and the PPS source of the MAC is the GNSS PPS 3: Force GNSS: PPS Slave and MAC PPS Source is the GNSS PPS	

AXI GPIO RGB

'1': ON; '0': OFF

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO (out)	-	SMA2 Blue	SMA2 Green	SMA2 Red	-	SMA1 Blue	SMA1 Green	SMA1 Red
	Bit15	**Bit14**	**Bit13**	**Bit12**	**Bit11**	**Bit10**	**Bit9**	**Bit8**
GPIO (out)	-	SMA4 Blue	SMA4 Green	SMA4 Red	-	SMA3 Blue	SMA3 Green	SMA3 Red
	Bit23	**Bit22**	**Bit21**	**Bit20**	**Bit19**	**Bit18**	**Bit17**	**Bit16**
GPIO (out)	-	-	-	-	-	GNSS Blue	GNSS Green	GNSS Red

AXI GPIO SMA MAP1 (ANT1/ANT2 as inputs, ANT3/ANT4 as outputs)

REG	Bit15	Bit14-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

GPIO (out)	Input Enable	unused	ANT1 to DCF Slave	ANT1 to IRIG Slave	ANT1 to TS2 input	ANT1 to TS1 input	ANT1 to PPS 2 input	ANT1 to PPS 1 input
GPIO2 (out)	Output Enable	unused	0b000000: 10MHz to ANT3 0b000001: FPGA PPS to ANT3 0b000010: MAC PPS to ANT3 0b000100: GNSS1 PPS to ANT3 ~~0b001000: GNSS2 PPS to ANT3~~ 0b010000: IRIG Output to ANT3 0b100000: DCF Output to ANT3 0b111111: GND to ANT3					
REG	**Bit31**	**Bit30-22**	**Bit21**	**Bit20**	**Bit19**	**Bit18**	**Bit17**	**Bit16**
GPIO (out)	Input Enable	unused	ANT2 to DCF Slave	ANT2 to IRIG Slave	ANT2 to TS2 input	ANT2 to TS1 input	ANT2 to PPS 2 input	ANT2 to PPS 1 input
GPIO2 (out)	Output Enable	unused	0b000000: 10MHz to ANT4 0b000001: FPGA PPS to ANT4 0b000010: MAC PPS to ANT4 0b000100: GNSS1 PPS to ANT4 ~~0b001000: GNSS2 PPS to ANT4~~ 0b010000: IRIG Output to ANT4 0b100000: DCF Output to ANT4 0b111111: GND to ANT4					

AXI GPIO SMA MAP2 (ANT1/ANT2 as outputs, ANT3/ANT4 as inputs)

REG	Bit15	Bit14-6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GPIO (out)	Input Enable	unused	ANT3 to DCF Slave	ANT3 to IRIG Slave	ANT3 to TS2 input	ANT3 to TS1 input	ANT3 to PPS 2 input	ANT3 to PPS 1 input
GPIO2 (out)	Output Enable	unused	0b000000: 10MHz to ANT1 0b000001: FPGA PPS to ANT1 0b000010: MAC PPS to ANT1 0b000100: GNSS1 PPS to ANT1 ~~0b001000: GNSS2 PPS to ANT1~~ 0b010000: IRIG Output to ANT1 0b100000: DCF Output to ANT1 0b111111: GND to ANT1					
REG	**Bit31**	**Bit30-22**	**Bit21**	**Bit20**	**Bit19**	**Bit18**	**Bit17**	**Bit16**
GPIO (out)	Input Enable	unused	ANT4 to DCF Slave	ANT4 to IRIG Slave	ANT4 to TS2 input	ANT4 to TS1 input	ANT4 to PPS 2 input	ANT4 to PPS 1 input
GPIO2 (out)	Output Enable	unused	0b000000: 10MHz to ANT2 0b000001: FPGA PPS to ANT2 0b000010: MAC PPS to ANT2 0b000100: GNSS1 PPS to ANT2 ~~0b001000: GNSS2 PPS to ANT2~~ 0b010000: IRIG Output to ANT2 0b100000: DCF Output to ANT2 0b111111: GND to ANT2					

1.7 Register Description

The detailed register descriptions of the NetTimeLogic Cores are available in the reference manuals:

../TimeCard/Doc/

The Documentations of the Xilinx Cores are online available.

AXI Memory Mapped to PCI Express:

https://www.xilinx.com/support/documentation/ip_documentation/axi_pcie/v2_8/pg055-axi-bridge-pcie.pdf

AXI GPIO:

https://www.xilinx.com/support/documentation/ip_documentation/axi_gpio/v2_0/pg144-axi-gpio.pdf

AXI I2C:

https://www.xilinx.com/support/documentation/ip_documentation/axi_iic/v2_0/pg090-axi-iic.pdf

AXI UART 16550:

https://www.xilinx.com/support/documentation/ip_documentation/axi_uart16550/v2_0/pg143-axi-uart16550.pdf

AXI Quad SPI:

https://www.xilinx.com/support/documentation/ip_documentation/axi_quad_spi/v3_2/pg153-axi-quad-spi.pdf

AXI HWICAP:

https://www.xilinx.com/support/documentation/ip_documentation/axi_hwicap/v3_0/pg134-axi-hwicap.pdf

1.8 Clock Selector

The design can run on different source clocks. It's important that at least the source clock of the SOM module is always available. In the FPGA design an automatic source clock selection is running. The design detects if a clock is available and does then the selection base on following priorities:

1. External 10MHz Clock from SMA connector
2. 10MHz Clock of MAC

This selection can be overwritten by the AXI GPIO Sel GPIO2. When a clock is selected the FPGA checks if this clock is available. If the selected clock is not detected the automatic selection is applied.

Via the GPIO Sel GPIO it can be checked which one is the selected clock. If all values are 0 the full design is running from the SOM Module Clock.

1.9 PPS Source Selector

The PPS Source Selector 1 can be controlled by AXI GPIO Sel GPIO2 Bit 0 and 1. There are three different PPS Sources (GNSS, MAC, SMA Connector) which can go to the two PPS Sinks (PPS Slave, MAC PPS Input). By default, the selection is done automatically. The source selector checks if a PPS is detected.

Depending on the availability the automatic selection has following priorities:

1. PPS SMA connector
2. PPS MAC
3. PPS GNSS

The switch happens once the PPS Sources are logical 0.

Following selection is done if a source is available:
1. SMA PPS is detected:
   SMA PPS source to PPS MAC sink
   SMA PPS source to PPS Slave sink
2. MAC PPS is detected:
   GNSS PPS source to PPS MAC sink
   MAC PPS source to PPS Slave sink
3. GNSS PPS is detected:
   GNSS PPS source to PPS MAC sink
   GNSS PPS source to PPS Slave sink

The PPS Source Selector 2 selects only the PPS toward the MAC PPS 2. There is no control or supervision via GPIO available. Possible sources are the 2nd GNSS PPS and a PPS from ANT1/2. From the available sources the ANT1/2 has higher priority over the 2nd GNSS PPS.

1.10 GNSS Requirements

The GNSS Source must be GPS, Galileo or Beidou. GLONASS only will not work.

1.11 Static configuration

The FPGA starts with a static configuration with following settings:
- PPS (including TOD) is used as correction input for the clock
- PPS Slave Pulse detection on rising edge
- PPS Slave cable delay 0
- TOD Slave UART Baudrate is 115200
- TOD Slave UART polarity default
- TOD Slave in UBX Mode, all GNSS and no messages disabled
- PPS Master polarity rising edge
- PPS Master cable delay 0
- PPS Master pulse width 100 ms
- Clock, PPS Slave, TOD Slave and PPS Master are enabled
- All Timestampers are disabled
- IRIG Slave/Master are disabled
- DCF Slave/Master are disabled

- TOD/NMEA Master is disabled

2 Program FPGA and SPI Flash

For the initial programming of the FPGA and SPI Flash the JTAG programmer is needed and has to be connected to the USB JTAG.
After a successfully programmed FPGA, the design contains an AXI QUAD SPI Core which allows field updates.

2.1 Bitstreams with Fallback Configuration

The FPGA design in split into two different bitstreams/bin-files to allow a failsafe field update. The FPGA configuration starts always at Addr0 where the Golden image with the start address of the Update image is located. It jumps directly to this address and tries to load the Update image. If this load fails it falls back to the Golden image.

Details about this Multiboot/Fallback approach can be found in following Application Note:
https://www.xilinx.com/support/documentation/application_notes/xapp1246-multiboot-bpi.pdf

The Golden/Fallback image contains only a limited functionality which provides access to the SPI Flash. The second image is used for normal operation and it is the one which is replaced in a field update.

Factory_TimeCardProduction.bin
This image contains two bitstreams and it shall be used to program the SPI flash for the first time as example in the factory. The first bitstream is the Golden/Fallback image and the second the latest version of the regular image.

This combined image has following structure:

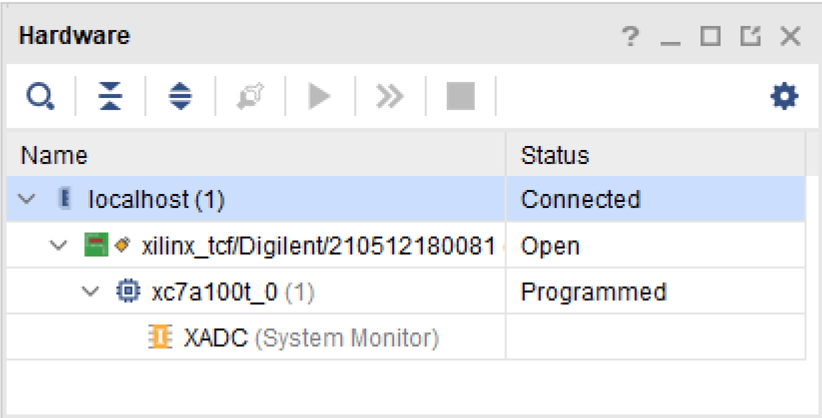
```
===================================
Configuration Memory information
===================================
File Format        BIN
Interface          SPIX4
Size               32M
Start Address      0x00000000
End Address        0x01FFFFFF

Addr1         Addr2         File(s)
0x00000000    0x002856FF    Golden_TimeCardProduction.bit
0x00400000    0x0069B5AB    TimeCardProduction.bit
```

TimeCardProduction.bin
This is the update/regular image and it shall be used for the field update via SPI. For the update this bitstream must be placed at 0x00400000 in the SPI flash.

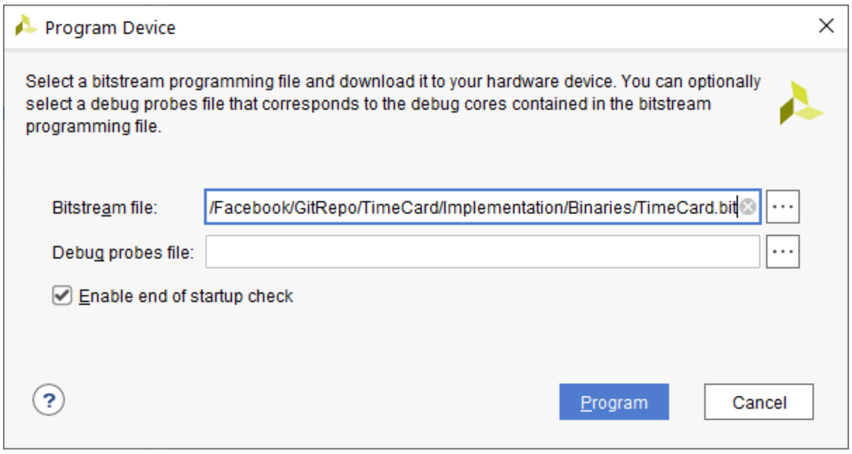
2.2 JTAG Programming (volatile)

This will only load the FPGA SRAM, after a power cycle this will be lost.

1. Go to the Hardware Manager in Vivado and Select "Open Target" → "Auto Connect". After this step following view is available:



2. Right klick on "xc7a100t_0(1)", a menu will popup
3. Choose "Program Device" from the menu, the following window will pop up

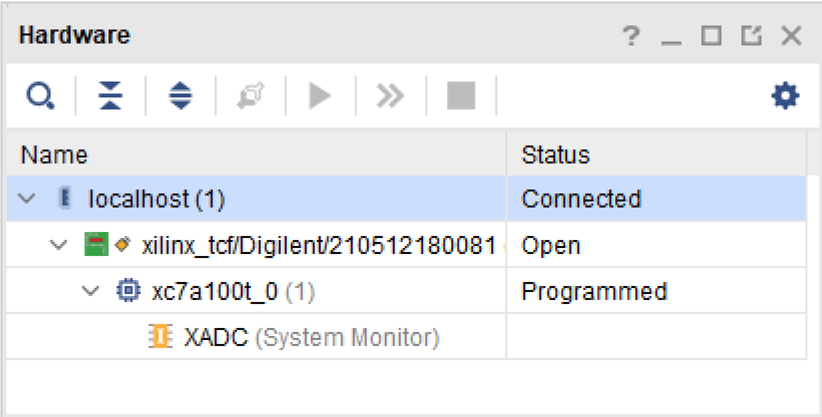


4. Select the bitstream you want to program: TimeCardProduction.bit
5. Press Program and wait for completion
6. The RUN LED will blink

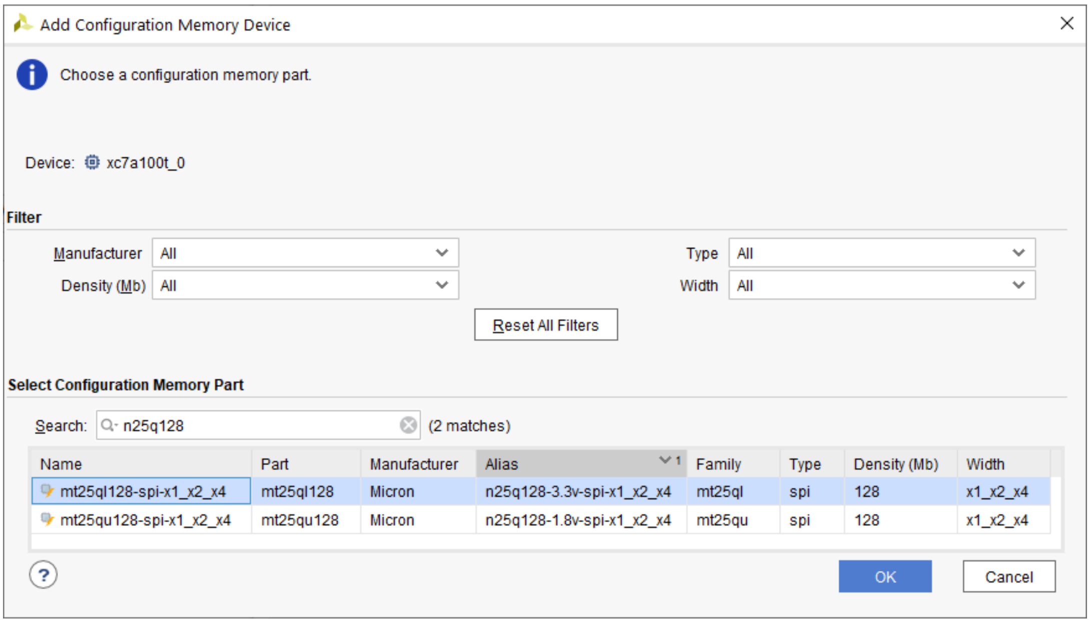
2.3 SPI Programming (non-volatile)

If no configuration memory was setup before start with step 1 otherwise start with step 7.

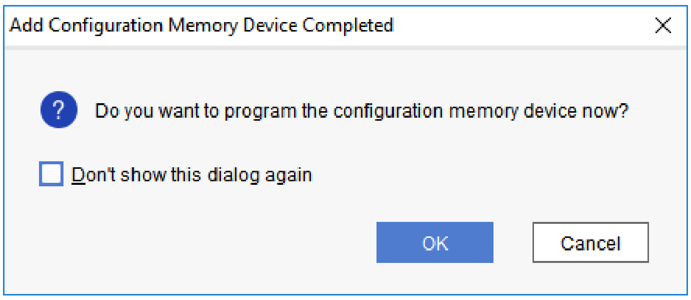
1. Go to the Hardware Manager Menu:



2. Right klick on “xc7a100t_0(1)”, a menu will pop up
3. Choose “Add Configuration Memory Device …” from the menu, the following window will pop up

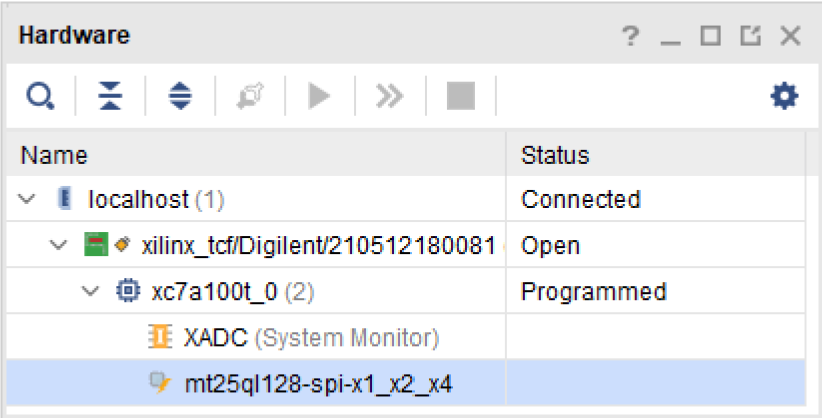


4. Select “mt25ql128-spi-x1_x2_x4” as the SPI Flash type
5. Press Ok, a new window will pop up:

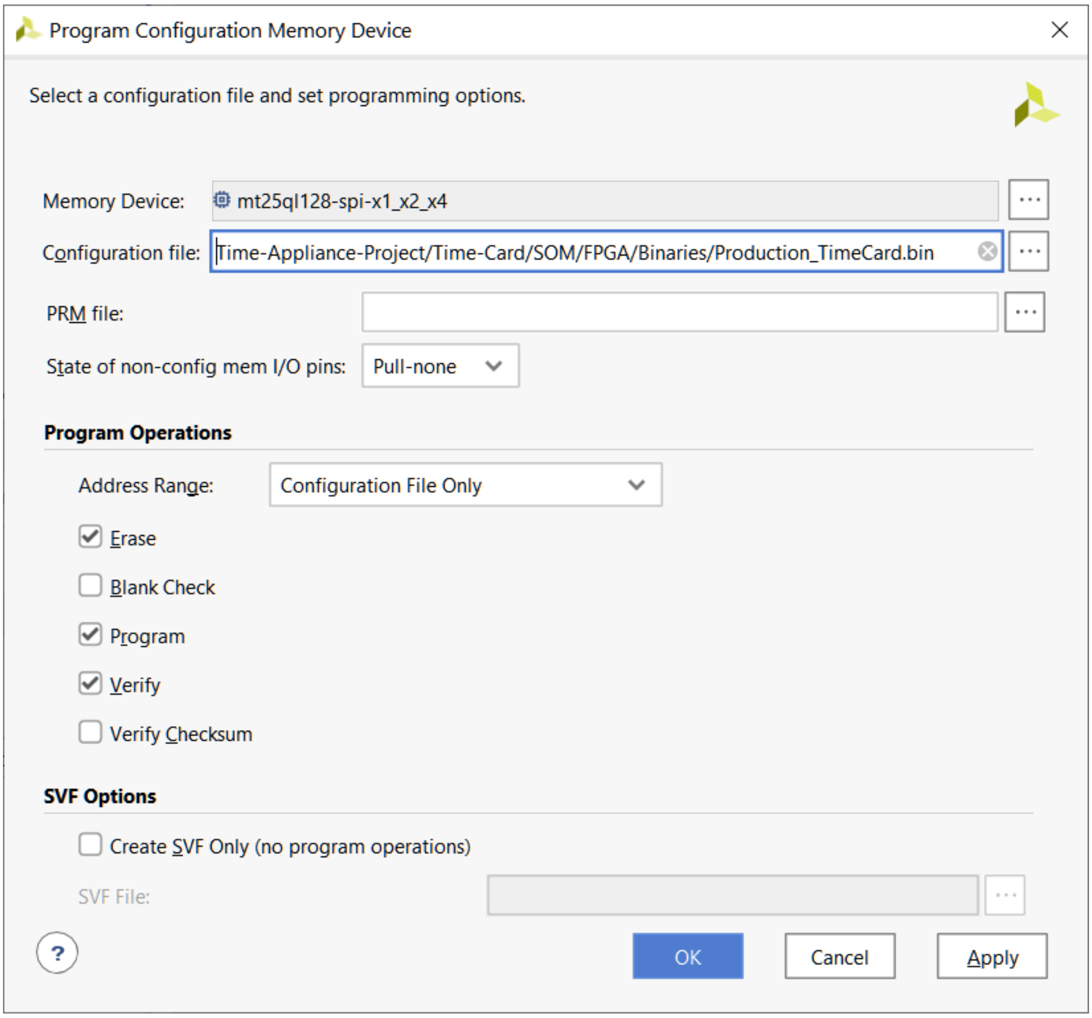


6. Press Cancel

7. Go to the Hardware Manager Menu which will have the flash attached:



8. Right klick on “mt25ql128-spi-x1_x2_x4”, a menu will pop up
9. Choose “Program Configuration Memory Device …” from the menu, the following window will pop up

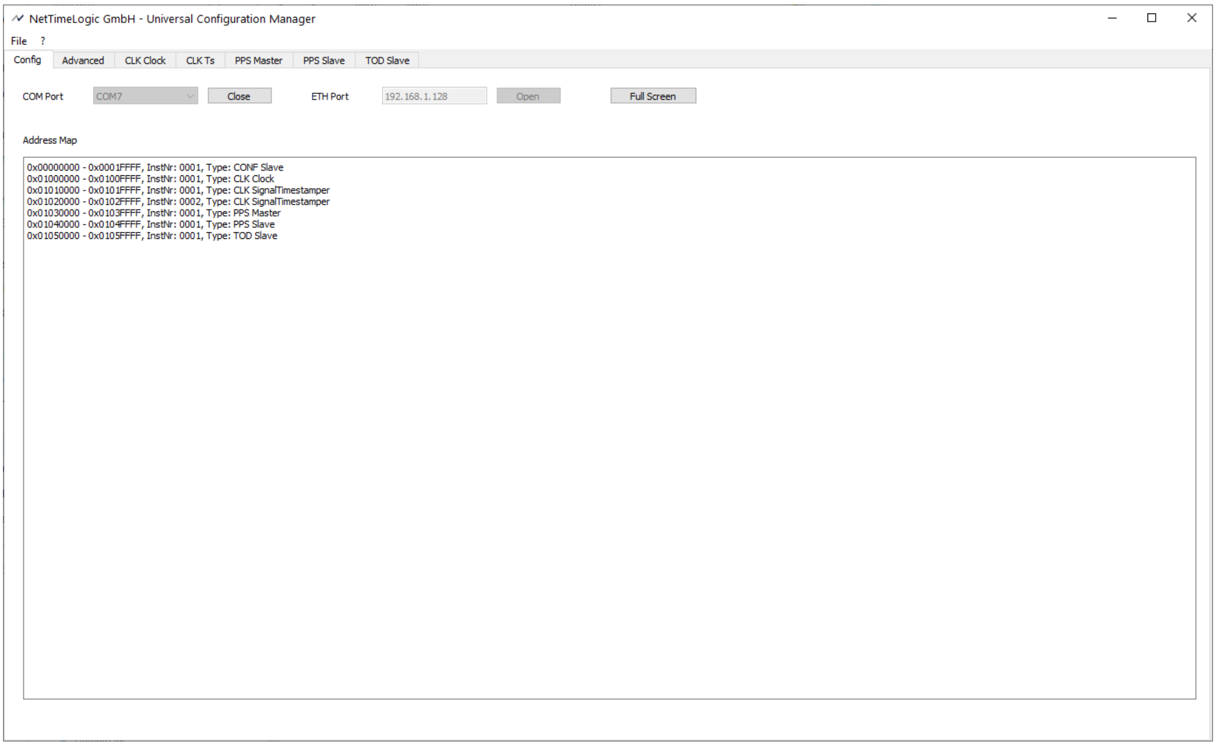


7. Select the bitstream you want to program:
Factory_TimeCardProduction.bin
IMPORTANT NOTE:
If in this step the TimeCardProduction.bin is loaded the field update as described above will not work!

8. Press Ok and wait for completion
9. Disconnect the JTAG interface from the board
10. Power cycle or Reset the board / Cold start of the PC
11. The RUN LED will blink

2.4 Connect NetTimeLogic Configuration Manager to FPGA

1. Connect the USB/UART to a Host PC running Windows.
2. Start NetTimeLogic's UniversalConfigurationManager Tool
3. Select the COM Port where the Board is connected to and press Open. Then the Address map of the cores instantiated are shown and five new Tabs appear (CLK Clock, CLK TS, PPS Master, PPS Slave and TOD Slave):



4. Change your configuration as you need it and press write. E.g.
   In the Advanced Tab it is possible to load configuration files by selecting a config file and pressing Load Config. Additionally, manual read or write of registers with the Field Address and Value is possible.

NOTE:
The Universal Configuration Manager communicates via UART commands and an internal IP Core converts them into AXI Memory Mapped register accesses.
Details about the protocol which can be used over any serial terminal is available here:
https://www.nettimelogic.com/resources/Ucm_UniversalConfigurationManager_ReferenceManual.pdf

3 TestApp

The TestApp is used to do some basic testing of the Hardware. It basically uses mmap to the address space of the PCIe mapped address.

Before access two the PCIe end device is possible two steps are required. First the PCIe device must be detected by the system this can be checked with following command:

```
lspci -v

01:00.0 Memory controller: Xilinx Corporation Device 7011
        Subsystem: Xilinx Corporation Device 0007
        Flags: fast devsel
        Memory at 90000000 (32-bit, non-prefetchable) [disabled] [size=32M]
        Capabilities: <access denied>
```

The PCIe device is detected at address 0x9000_0000 but it is disabled. As a second step the device must be enabled:

```
sudo setpci -s "01:00.0" COMMAND=0x02
```

Now the TestApp is ready to start. The TestApp requires the PCIe base address as an argument:

```
sudo ./TestApp 0x90000000
```

The App reads the version of the NetTimeLogic IP cores and set the system time to the Adjustable Clock. After that it ready every second the time back from the Adjustable Clock:

```
PCIe Base Address is set to 0x90000000
Clock IP Core Version = 0x1020000
Signal TS IP Core Version = 0x1020001
Signal TS IP Core Version = 0x1020001
PPS Master IP Core Version = 0x1020000
PPS Slave IP Core Version = 0x1020000
TOD Slave IP Core Version = 0x2000001
Selected Clk Source is: PPS
Selected Clk Source is: REGS
Set the current local time and date: Fri Oct 23 14:00:35 2020
The time is: 14:00:35 and 3240 ns
The time is: 14:00:36 and 305900 ns
The time is: 14:00:37 and 490780 ns
The time is: 14:00:38 and 671200 ns
The time is: 14:00:39 and 1031680 ns
The time is: 14:00:40 and 1359140 ns
The time is: 14:00:41 and 1542180 ns
The time is: 14:00:42 and 1671980 ns
The time is: 14:00:43 and 1799580 ns
```