CEA LETI - DÉPARTEMENT DACLE

STELLAR Board Programming Interface (TSARLET version)

Version 0.2

First version: November, 2017
Current version: January, 2018

César Fuguet Tortolero

Contents

1 Introduction

This document has two main parts: the first describes the procedure for booting the TSARLET Integrated Circuit, and the second describes the external memory and the peripherals implemented on the STELLAR board.

2 TSARLET Configuration

The Power, Reset and Clock Management Microcontroller Unit (PRCMCU) on the FPGA is responsible for the configuration and booting of the TSARLET Integrated Circuit (called TSARLET on the reminder of this document). Once TSARLET has booted, the PRCMCU accomplishes mainly monitoring tasks (e.g. power consumption and temperature monitoring).

This document describes the configuration-related tasks performed by the PRCMCU and the booting procedure performed by this last to bring up TSARLET. In particular, regarding the boot procedure, the corresponding section explains the Application Programming Interface (API) provided by the PRCMCU firmware so as to enable the execution of any piece of software on TSARLET.

The configuration-related tasks performed by the PRCMCU on TSARLET are the following :

1. Setup of the TSARLET's Power Supply Unit (PSU).
2. Setup of TSARLET's internal clock generators : Frequency-Locked Loop (FLLs).

3. Execution of the integrated Memory Built-In Self Tests (MBISTs) to detect and repair memories' defects.

4. Setup of mapping between L2 and L3 caches.

5. Setup of mapping between L3 caches and the external memory.

6. Setup of the physical communication interface between TSARLET and the FPGA.

3 TSARLET Booting

Once the TSARLET configuration is completed, the PRCMCU starts the booting procedure of TSARLET. This booting procedure consists mainly on copying a first level (level 0) TSARLET's bootstrap software (called TSARLET B0) onto TSARLET's addressable memory.

Once the TSARLET B0 software is copied, all TSARLET's cores are wake-up by the PRCMCU, and therefore, at that point, it is the responsibility of the TSARLET B0 to manage all cores and load from the external disk the next level of the bootstrap process (called TSARLET B1).

3.1 TSARLET Bootstrap Stage 0 (TSARLET B0)

This software is stored on the PRCMCU dedicated SD memory card (*STELLAR board socket SD2*) in the form of a Executable and Linkable File (ELF). This file shall be in the ELF32 format for the MIPS32 architecture, as it will be executed by TSARLET's cores and therefore it should match the TSARLET's Instruction Set Architecture (ISA). For more information regarding the architecture specific flags that should be used for compiling software for TSARLET, refer to the *TSARLET Programming Interface* document.

The firmware executed by the PRCMCU is responsible for loading the ELF file of the TSARLET B0 (*tsarboot.elf*), from its dedicated SD card memory, onto the external memory (addressable by TSARLET). As TSARLET's cores boot address is hardwired to 0x00000000, the TSARLET B0 ELF must implement its entry procedure on this address.

By default, the PRCMCU firmware will search for the *tsarboot.elf* file into its dedicated SD memory card and load it. However, another file can be loaded if the user request it on the command line of the PRCMCU dedicated terminal.

The TSARLET B0 software must fit into its dedicated memory region. This memory region spans from address 0x00000000 until address 0x00400000 (4MB). The last 4KB of this memory region (0x003FF000 – 0x00200000) are used to pass platform related information to TSARLET.

The platform related information to TSARLET is the following:

- Starting TSARLET's clock frequencies.

- Starting TSARLET's power supplies voltages.

The PRCMCU reserves also a memory region immediately above the TSARLET B0 reserved memory region. This region has also 4 MB, and it is used for the PRCMCU stack and heap.

TSARLET B0 Information	
TSARLET B0 ELF Path	/tsarboot.elf
TSARLET B0 Reserved Memory Region	0x00000000 – 0x00400000 (size = 4MB)
TSARLET's Boot Address	0x00000000
TSARLET's Boot Exception Address	0x00000380
STELLAR Board Information Base Address	0x003ff000

Platform Related Information (address = 0x003ff000)	
Offset (from Base)	Content
0x0000	Cores Clock Frequency (in MHz) (32-bits unsigned)
0x0004	L3 Clock Frequency (in MHz) (32-bits unsigned)
0x0008	DSPIN Clock Frequency (in MHz) (32-bits unsigned)
0x000C	SNOC Clock Frequency (in MHz) (32-bits unsigned)
0x0010	XMEM/IO Interface Clock Frequency (in MHz) (32-bits unsigned)
0x0014	VDD chip (in mV) (16-bits signed)
0x0016	VDDS (in mV) (16-bits signed)
0x0018	GNDS (in mV) (16-bits signed)
0x001a	Reserved (16 bits)
0x001c	FLLs Reference Clock Frequency (32-bits unsigned)
0x0020	FLLs Minimum Set Point (16 bits) (16-bits unsigned)
0x0022	FLLs Maximum Set Point (16 bits) (16-bits unsigned)
0x0024 – 0x1000	Reserved

Table 1: Platform Related Information

PRCMCU Reserved Hardware	
Reserved Memory Region	0x00400000 – 0x00800000 (size = 4MB)
UART board socket	UART2
SPI board socket	SD2

NOTE: If the PRCMCU needs to update some of the platform related information after Tsarlet has been booted, then it needs to write directly on Tsarlet's direct memory space and not on the external memory space, as there is no way to invalidate memory regions on Tsarlet's L2 and L3 caches.

4 TSARLET Reset Clock Frequencies and Power Supplies

As said above, the PRCMCU is responsible of the setup of clock frequencies, and power supplies of TSARLET during the reset procedure. For this version of the STELLAR board FPGA design, the clock frequencies and power supplies are configured by the PRCMCU as follows:

TSARLET Clock Frequencies	
Clock Domain	Frequency (MHz)
Cores & L2 caches	600
L3 caches	200
DSPIN Interconnects	600
SNOC Interconnects	600
XMEM/IO Interface	40

TSARLET Power Supplies	
Power Supplies	Voltage (mV)
VDD chip	950
VDDS	0
GNDS	0

5 STELLAR Board External Memory

The STELLAR board implements a 16 GBytes DDR4 memory. The maximum memory bandwidth is 12 GB/s as it implements a 64 bits data interface with 1600 MTransactions per second.

The 16 GBytes of memory are evenly distributed among the memory address space of each TSARLET cluster (4 GBytes for each cluster). The available memory segments of TSARLET are shown in the following table:

DDR4 Memory Mapping	
TSARLET Memory Segment Base Address	TSARLET Memory Segment Last Address
0x0000000000	0x00dfffffff
0x0100000000	0x01dfffffff
0x1000000000	0x10dfffffff
0x1100000000	0x11dfffffff

As it can be noted in the previous mapping, the last 512 MB of the memory address map at each cluster is not mapped on the DDR4 external memory. This is because TSARLET has its internal peripherals mapped from the offset 0xXYe0000000 at each cluster. Therefore, there are 2 GBytes of DDR4 memory not used.

6 STELLAR Board IO Peripherals

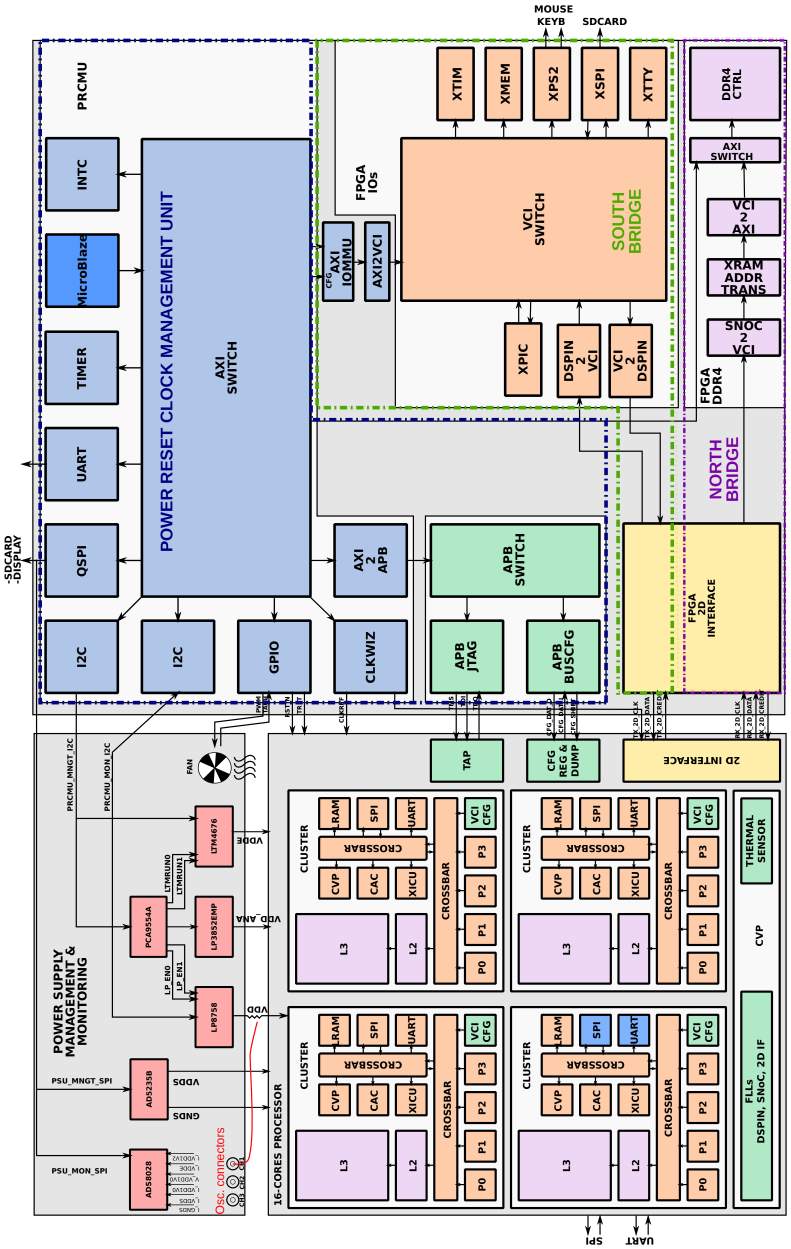

Figure 1: STELLAR Board FPGA Design

6.1 IO Mapped Peripherals for TSARLET

TSARLET IO Mapped Peripherals	
Base Address	Peripheral
0x1200000000	eXternal Framebuffer (XFBF)
0x12F0000000	eXternal Programmable Interrupt Controller (XPIC)
0x12F1000000	eXternal Teletypewriter (XTTY)
0x12F2000000	Serial Peripheral Interface Dedicated to the MCU (SPI MCU)
0x12F3000000	eXternal Personal System/2 (XPS2)
0x12F4000000	eXternal Video DMA Configuration Interface (XVMA)
0x12F5000000	eXternal Configuration Memory (XMEM)
0x12F6000000	eXternal Free Running Timer (XTIM)
0x12F7000000	DSPIN Traffic Checker (XDCH)
0x12F8000000	SNoC Traffic Checker (XSCH)
0x12F9000000	eXternal Serial Peripheral Interface (XSPI)

Table 2: TSARLET IO Mapped Peripherals

6.2 XFBF

This segment is physically mapped on an external, video-dedicated, DDR4 memory of 512 MBytes. This means that theorically we can store up to 60 complete video buffers (images) for a display resolution of 1080p (1920*1080). However, the actual number of used video buffers will depend on the configuration of the XVMA component.

The framebuffer expects pixels on the RGB format, where each component is 8-bits wide. Each pixel is aligned to 4 bytes, thus only the 24 lsb are used. Additionally, to ease framebuffer indexing, each line is aligned to 2048 pixels (8 KBytes).

Framebuffer Segment	0x1200000000 – 0x121f000000 (size = 512MB)
Framebuffer Pixel Format	RGB of 8 bits each, right-aligned (24 lsb)
Framebuffer Pixel Alignment	4 bytes
Framebuffer Line Stride	2048 pixels (8KBytes)

6.2.1 STELLAR Board Specifics for the XFBF

For this version of the architecture, only the framebuffer 0 (in segment 0x1200000000 – 0x1200870000) is sent to the HDMI output.

6.3 XPIC

It is a Multi-channel Programmable Interrupt Controller. It has multiple input, hardwired, level-signalled interrupts. Each of this inputs has a private context consisting in a few programmable configuration registers. Each of these contexts is called channel.

This component converts hardwired signalled interrupts into message signalled interrupts. When there is a rising edge event on one of the hardwired interrupts, a write command is sent through the interconnect to a programmable destination address. In the same manner, when there is a falling edge event, a read command is sent to the same destination address. In general, this destination is an interrupt controller capable of doing the opposite operation, this is, convert the messaged signalled interrupt into a hardwired signalled one, which is connected to a processor core.

In the case of TSARLET, the destination address, for each XPIC input interrupt, shall be the address of one XICU device. This component has a mailbox capability, which when writing into one of its mailboxes, it triggers a hardwired interrupt to one of the local processor cores (cores in the same cluster). When there is an active mailbox interrupt, and this mailbox receives a read command, the corresponding interrupt is acknowledged, thus disabled.

The address map of this component is the following:

- ADDRESS[3:2] indexes a register of a specific channel.
- ADDRESS[11:4] indexes the channel.

Hereafter the address mapping of each channel:

XPIC Channel Register Map

Offset	Register	Description
0x00	ADDRESS (LO)	32 least signifiant bits of the destination address
0x04	ADDRESS (HI)	8 most signifiant bits of the destination address
0x08	STATUS	[0] state of the input interrupt / [1] error

The error bit on the status registers is set when there was a transaction error while writing/reading the message signalled interrupt to/from the destination address (e.g. write/read to illegal address).

6.3.1 STELLAR Board Specifics for the XPIC

On the STELLAR board, this component has two hardwired interrupt signals, thus two channels:

- Channel 0 (offset 0x00) is associated to the XTTY component.
- Channel 1 (offset 0x10) is associated to the XSPI component.

6.4 XTTY

This component is a 16550 compatible UART controller. For more detailed documentation about this component refer to the document (*oc_uart_16550.pdf*).

6.4.1 STELLAR Board Specifics for the XTTY

Register Map
The XTTY component uses a byte-aligned register map. Every register is 8-bits wide and only single-byte, read/write commands are supported.

Baud Rate
Regarding the baud rate, the baud rate generator of this component has an input frequency of 143 MHz. Therefore, DLL/DLR registers of the baud rate generator should be configured as to achieve the desired transmission baud rate.

Board Socket
XTTY RX/TX signals are routed to the UART1 socket of the STELLAR board.

6.5 XSPI

This component is a DMA-capable SPI controller.

XSPI Register Map

Offset	Register	Description
0x00	RXTX[0]	SPI transmission/reception register (bits [31:0])
0x04	RXTX[1]	SPI transmission/reception register (bits [63:32])
0x08	RXTX[2]	SPI transmission/reception register (bits [95:64])
0x0C	RXTX[3]	SPI transmission/reception register (bits [127:96])
0x10	CTRL	SPI control register
0x14	DIVIDER	SPI clock divider value
0x18	SS	Slave Select (asserted low)
0x1C	DMA BASE (LO)	DMA target memory address (32 least signifiant bits)
0x20	DMA BASE (HI)	DMA target memory address (8 most signifiant bits)
0x24	DMA LENGTH	DMA transfer length in bytes

6.5.1 STELLAR Board Specifics for the XSPI

Slaves
It is connected to an unique SPI slave (slave select bit 0). This slave is a SD card socket. Future versions of the FPGA design will replace this component by a 4-bits bus capable, SD card controller.

SPI clock
Regarding the SPI clock, its generator uses as input a 143 MHz clock. Therefore, the DIVIDER register should be configured as to achieve the desired frequency.

Board Socket
XSPI signals are routed to the SD1 socket of the STELLAR board.

6.6 XPS2

The XPS2 peripheral is a 2-channel PS2 controller, where each channel can control one PS2 slave device.

XPS2 Register Map

Offset	Register	Description
0x00	CHANNEL0 DATA	Transmission/reception buffer to/from PS2 slave 0
0x04	CHANNEL1 DATA	Transmission/reception buffer to/from PS2 slave 1
0x08	CONTROL/STATUS	Control/Status Register
0x0C	CLK DIVIDER	PS2 generated clock divider

Control/Status Register

The status register contains the following information:

Control/Status Register Bitmap	
Bit	Description
0	CHANNEL0 data ready (read-only)
1	CHANNEL1 data ready (read-only)
2	CHANNEL0 command ready (read-only)
3	CHANNEL1 command ready (read-only)
[4 : 7]	Reserved
8	CHANNEL0 irq enable
9	CHANNEL0 reset
10	CHANNEL1 irq enable
11	CHANNEL1 reset

6.6.1 STELLAR Board Specifics for the XPS2

XPS2 clock
Regarding the PS2 clock, its generator uses as input a 143 MHz clock. Therefore, the CLK DIVIDER register should be configured as to achieve the desired frequency. As there is only one clock generator, the same clock frequency is generated for both channels.

PS2 slaves
The channel 0 of the XPS2 controller is connected to the PS2 keyboard connector, and the channel 1 to the PS2 mouse connector.

6.7 XMEM

It is a simple 8 KBytes embedded RAM, which can be used to share data with the PRCMCU subsystem. It is mapped from offset 0x0000. Accesses beyond offset 0x2000 will return an error.

This RAM shall not be mapped on the cacheable space of TSARLET.

6.8 XTIM

This components implements a simple free-running timer, which increments from 0 to $2^{32} - 1$ at a 143 MHz frequency. When it reaches the max value, it goes to 0 and continue.

It implements one single register (mapped to offset 0x0). When this register is read, this components returns the current value of the timer. If it is written, the timer value is reset to 0.

6.8.1 STELLAR Board Specifics for the XTIM

As TSARLET clusters have independent clock generators, the cores' cycles counter register (MIPS32 Coprocessor 0 count register), can diverge from one cluster to another. Therefore, for applications which need a global timer, the XTIM component can be used as such.

6.9 XDCH

This component allows to generate/check a synthetic traffic to/from TSARLET on DSPIN interconnects.

This component shall not be accessed by TSARLET. Only the FPGA PRCMCU should access it. In future versions of the board synthesized design, this component will not be longer mapped on the DIRECT address space of TSARLET.

6.10 XSCH

This component allows to generate/check a synthetic traffic to/from TSARLET on SNOC interconnects.

This component shall not be accessed by TSARLET. Only the FPGA PRCMCU should access it. In future versions of the board synthesized design, this component will not be longer mapped on the DIRECT address space of TSARLET.

6.11 TSARLET's Internal UART and SPI

TSARLET has two usable controllers on cluster 0 implemented in the ASIC. One is for UART and the other for SPI interfaces. Each is connected to a dedicated socket on the STELLAR board. The SPI interface drives a SD card socket.

Sockets for TSARLET Internal Peripherals

Interface	**Socket**
UART	DB_UART
SPI	DB_SD

7 STELLAR FPGA Flash Memory

The STELLAR board FPGAs have each an associated non-volatile memory (flash memory) to store bitstream files. This allows to reprogram FPGAs automatically after the board power-up without using the programming JTAG interface. Another advantage (with respect to JTAG) is that programming is much faster.

Regarding the FPGA1 (used by the TSARLET circuit), the flash-memory implemented has the following characteristics :

Flash-Memory Characteristics

Size	64 MB (512 Mbits)
Device	mt25qu512-spi-x1_x2_x4
Bus	SPI - 4 bits

This device shall be added on the Vivado Hardware Manager of the Vivado project and programmed with a previously generated MCS file. This MCS file shall be generated from the desired bitstream file.

The **FPGA1 PROG** push button on the STELLAR board allows to trigger the FPGA1 reconfiguration with the bitstream in the flash memory.