Class: TN
Doc. no: TERMA/SPD/63/T-EMU/TARGET/
SPARCV8
Rev: 1.0
Date: 2015-03-01
Approved by: Michela Alberti

PUBLIC

T-EMU: SPARCv8 Target Manual

Prepared

Checked

_____

Mattias Holm
Technical Manager

_____

Dan Søren Nielsen
QA Manager

Approved

_____

Michela Alberti
General Manager

_____

_____

Record of Changes

Author	Description	Rev	Date
Mattias Holm	Initial Version	1.0	2015-03-01

Table of Contents

1. Introduction

The SPARCv8 target comes in several variants, these include, emulator cores for the ERC32 (technically a SPARCv7), LEON2 and LEON3.

The individual targets only include the CPU core, and not any surrounding device models. The on-chip devices must be connected to the CPU core at configuration time.

2. Variants

These are the main variants of the SPARCv8 targets as supported by T-EMU at present. Other variants can be added at request.

2.1. ERC32

The ERC32 core implements the SPARCv7 instruction set. It does not include the multiply and divide instructions from the SPARCv8. It also lacks the MMU.

2.2. LEON2

The LEON2 core implements the SPARCv8 instruction set as provided by the AT697F processor. Note that the LEON2 VHDL models also support some SPARCv8-E extensions (e.g. integer multiply accumulate instructions, but these extensions are not currently in the LEON2 core in order to be similar to the AT697F).

2.3. LEON3

The LEON3 core includes the SPARCv8 instruction set, some SPARCv8-E extensions (UMAC and SMAC instructions), the CASA instruction from the SPARCv9 ISA and the SR-MMU memory management unit.

_____

2.4. LEON4

The LEON3 core includes the SPARCv8 instruction set, some SPARCv8-E extensions, the CASA instruction from the SPARCv9 ISA and the SR-MMU memory management unit.

There are two differences from the LEON3:

• Instruction timing uses values from LEON4 documentation

• Supports partial WRPSR when RD != 0. There is no real assembler syntax to express this instruction (and it disassembles to the normal wrpsr format).

3. Configuration

3.1. Properties

The following properties are important for configuration of a virtual system.

3.1.1. Interface References

memaccess	The interface reference to an object reacting to the emulator core's memory accesses (whenever there is an ATC miss). This should normally refer to a memory space object or the MMU interface. Set this to memspace:MemAccessIface in case the CPU lacks an MMU or to cpu:MmuMemAccessIface in-case the CPU has an MMU. That is, in the case of an MMU, the iface reference refers to the object itself.
memAccessL2	The interface reference to an object reacting to the memory accesses invoked first in the memaccess interface reference. In case the system has an MMU, set this to memspace:MemAccessIface.
memory	The interface reference to an object handling memory block read and writes, this should normally refer to a memory space object.
irqctrl	The interface reference to an object implementing the IrqControl interface. This can be used to connect external interrupt controllers which need to have interrupts acknowledged.
devices	Array of interface references to device models. The objects in this array will have a CPU reset call propagated to themselves. If your device model handles reset messages, it must be put into the devices array.

3.1.2. Other Properties

freq	Clock frequency in Hz. Defaults to 50000000 = 50 MHz.
cpuid	CPU id for multiprocessor configurations, defaults to 0.

3.2. Interfaces

The SPARCv8 emulator cores implement the following interfaces:

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

PUBLIC

_____

CpuIface	The common CPU interface. This contain functions like run and register access functions.
SparcIface	Standard SPARCv8 interface. Contains among other things functions for accessing windowed registers. One capability of the SPARC interface is the registration of ASI handlers.
IrqIface	The interrupt controller interface for raising interrupts on the processor.
InvalidMemAccessIface	Interface invoked on invalid memory accesses. This contain functions that will longjmp to the CPU trap handling logic. The interface can only be invoked from code invoked by the CPU core in one way or the other. Do not call the functions in this interface directly!
EventIface	Interface for posting timed events on the CPU core's event queue. Usually a reference to this event is installed in connected device models.
MemoryIface	Proxy interface which forwards to the memory space object.
MmuMemAccessIface	The memory interface provided by the CPU to do accesses through the MMU.

4. Limitations

Current limitations of the SPARCv8 target include:

- The `wrpsr` instruction is effective immediately. The up to three nops, needed in real code serves no purpose in the emulator. Thus if nops are omitted you will not detect this on the emulator at present.

- Floating point traps are direct and not deferred. This is the correct behaviour for the AT697F, but may not be correct for other chips.

- Timing effects due to super-scalar execution is not simulated. Again, this is correct behaviour for the AT697F.

- Operator dependant timing effects (especially noticeable in the FPU) are not simulated. Timing for instructions is static and uses the documented typical values.

- The LEON2 model takes FPU timings from the ERC32 as no documentation about the costs on the MEIKO FPU (which is the standard FPU for the LEON2) is available. The only known data for the MEIKO is in the same magnitude as the ERC32 FPU (which is not MEIKO), hence we assume that the ERC32 timings are roughly correct for the LEON2.

- The FPU model is based on the SPARCv8 standard, and follows the SPARCv8 recommendations for NaN-propagation. If the SPARCv8 you emulate use an FPU that is not compliant with the SPARCv8 NaN propagation recommendations, there may be slight deviation in results.

_____