

## TEMU *ARMv7 Target Manual*

Mattias Holm

Version 1.0, 2018-05-14

# Table of Contents

- 1. Introduction . . . . . 1
- 2. Variants . . . . . 1
  - 2.1. ARMv7-R . . . . . 1
- 3. Operating System Compatibility . . . . . 1
- 4. Configuration . . . . . 1
  - 4.1. Interface References . . . . . 1
  - 4.2. Other Properties . . . . . 2
  - 4.3. Interfaces . . . . . 2
- 5. Limitations . . . . . 3

*Table 1. Record of Changes*

Rev	Date	Author	Note
1.0	2018-05-14	MH	Initial version.

# 1. Introduction

The ARMv7 target in TEMU comes with support for the ARMv7-R sub target at the moment. A number of on-chip devices based on existing ARM based CPUs are available. Currently this includes a subset of the TMS570 devices.

# 2. Variants

## 2.1. ARMv7-R

The ARMv7-R core is available, it comes with an PMSA compliant MPU modelled on the Cortex-R5 MPU. Note that this differs slightly from the ARMv7-R architecture description. Especially, it lacks dedicated executable protection and uses read protection to handle fetch permission.

While the model is still in beta state, the model has been successfully used to run actual time and space partitioned hypervisors with partitions using different kernels (e.g. XtratuM / RTEMS / XAL is known to run).

# 3. Operating System Compatibility

The ARMv7 models have been executed successfully with:

- RTEMS
- XtratuM
- XAL

# 4. Configuration

## 4.1. 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.

## **memory**

The interface reference to an object handling memory block read and writes, this should normally refer to a memory space object.

## **irqClient**

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.

## **coproc**

The interface reference array contains references to coprocessor interfaces. The coprocessor interface is currently unstable and it is advised that it at this moment not implemented by third parties.

# **4.2. Other Properties**

## **freq**

Clock frequency in Hz. Defaults to 50000000 = 50 MHz.

## **cpuid**

CPU id for multiprocessor configurations, defaults to 0.

# **4.3. Interfaces**

The SPARCV8 emulator cores implement the following interfaces:

## **CpuIface**

The common CPU interface. This contain functions like run and register access functions.

## **ArmIface**

Standard ARMv7 interface. Contains among other things functions for accessing banked registers.

## **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.

## 5. Limitations

Current known limitations of the ARMv7 target include:

- Performance is limited due to decoding logic needed for ARMv7 and Thumb2 ISAs. Current performance is around 30 % of the SPARCv8 model. This will be addressed in the future.
- No static timing model is defined at this moment. That means that one instruction take one cycle to finish.
- The built-in assemblers and disassemblers are not working at this moment.
- Cache control interfaces are not implemented or supported, this can be addressed if needed.
- NEON (vector) instructions are not implemented at this moment.
- A co-processor interface exists, but it is currently not stable.