

T-EMU: LEON2 Device Model Manual



Prepared

Mattias Holm
Technical Manager

Approved

Michela Alberti
General Manager

Checked

Dan Søren Nielsen
QA Manager



Record of Changes

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

Table of Contents

1. Introduction	2
2. Configuration	2
2.1. Interrupt Delivery	2
2.2. UART Connections	3
2.3. Infinite UART Speed	3
2.4. GPIO	3
3. Limitations	3

1. Introduction

2. Configuration

2.1. Interrupt Delivery

Set the `irqControl` property to point out the processor's irq interface. The model will deliver normal SPARC interrupts (1 up to 15). The LEON2 also exports the `IrqCtrlIface` as `IrqIface`. `IrqClientIface` should be wired from the CPU the LEON2 model is connected to.

The `IrqIface` enables the use of external interrupts using the `raise` and `lower` functions. The LEON2 has 8 external IRQs mapped according to the following table (the mappings cannot be customised at present):

Table 1. External to Internal IRQ Mapping

External	Internal (Sparc IRL)
0	4
1	5
2	6
3	7
4	10
5	12
6	13
7	15

The rules for IRQ raising is controlled by the GPIO IRQ config registers (it is also possible to raise IRQs by setting and lowering GPIO pins).



2.2. UART Connections

The UARTs are connected to the destination using the `uarta` and `uartb` properties. For the remote end points, these should be connected to `UartAIface` and `UartBIface`.

2.3. Infinite UART Speed

The UARTs can run either at infinite speed, or at simulated real-time speed. This can be configured using the `infiniteUartSpeed` property. Set this property to non-zero to enable infinite UART speed.

Note that this controls the speed of both UARTs.

When infinite speed is enabled, bytes are emitted to the destination serial device as soon as they have been written by the OBSW.

2.4. GPIO

The GPIO support in the LEON2 model supports interrupt generation using the GPIO interface instead of the IRQ controller interface. Model implements both the `GpioClientIface` and a property with a `GpioBusIface` reference (called `gpioBus`). The GPIO bus connection is not mandatory to set. If it is set, writes to the GPIO data register's out bits will be forwarded over the GPIO port. Note that the LEON2 only have 16 GPIO pins.

3. Limitations

The Leon2 Device model simulates the AT697F chip. There are some deviations to the AT697E chip (e.g. the size of the counters). If you need the AT697E behaviour, please contact us for more info.

The following deviations from real hardware are known to exist, if you need the correct behaviour (or simulation of it, contact us for more info):

- No support for Ethernet at present
- No support for PCI at present
- The UARTs do not support external clocks.
- The UARTs do not support parity, framing errors and break signals.
- GPIO pin configurations are ignored for UARTs, the UARTs are assumed to be on separate dedicated I/O pins.
- GPIO databus control is not supported (i.e. `meddat` and `lowdat` fields).
- Write protection registers have no effect
- Timer values are lazily computed on reads, the content in the case a timer is disabled is estimated on disabling time. This is in principle correct. However, the prescaler counter write has no effect, only the



reload value has an effect when written. This may cause an offset of 1024 cycles when re-enabling a timer.

- In general the MEMCFG registers are ignored