

GRALL Julien
PERCOT H el ene



QEMU

for LEON2 and RTEMS 4.10

may 2011

Table of Contents

1 Introduction.....	2
2 How to compile and launch this project ?.....	3
3 What functionalities were studied ?.....	4
3.1 UARTs.....	4
3.2 Registers.....	4
3.3 Traps.....	4
3.4 Memory controller.....	5
3.5 Monitor.....	5
4 How to test this project ?.....	6

1 Introduction

The aim of this project was to modify the qemu project for LEON2 processor and RTEMS operating system. This project was accomplished in a student project of EPITA school in partnership with Adacore company.

This document describes how to compile and use the project, what functionalities are implemented and how to test the project.

The authors don't give the warranty of the complete functionality of the project.

2 How to compile and launch this project ?

To compile the program :

```
$ cd leon2-qemu  
$ ./configure --target-list=sparc-softmmu  
$ make
```

To launch the program :

```
$ ./sparc-softmmu/qemu-system-sparc -M at697 -nographic -kernel <binary>
```

3 What functionalities were studied ?

The Leon2 Processor is based on the Sparc (V7) processor. A lot of functionalities are implemented in the Qemu for Sparc. This document describes only the special functionalities we have used during .

3.1 UARTs

Two UARTs are implemented (uart1 et uart2). Each UART is composed by eight data bits, one stop bit and eventually one parity bit. It is possible to program individually the speed of the UART changing the LeonUartState structure.

3.2 Registers

One register who is very important is the stack pointer register : %o6. It is possible to use the alias %sp for use this register.

3.3 Traps

The trap management is the same in leon2 than in Sparc. You have two kind :

- Hardware : asked by the processor. Traps from 0 to 127.
- Software : asked by a program. Traps from 128 to 256.

In assembly code, it is only possible to use the software traps. For example, the instruction “ta 0” call the 128 trap.

3.4 Memory controller

The memory controller can control until 2 Gbytes of memory. The memory is distributed like you can see in this table :

Address range	Size	Mapping
0x00000000 - 0x1FFFFFFF	512 Mb	Prom
0x20000000 - 0x3FFFFFFF	512 Mb	I/O
0x40000000 - 0x7FFFFFFF	1 Gb	SRAM/SDRAM

3.5 Monitor

In the case where there is no bios/monitor, the program start the application. We had to change the Stack Pointer because for RTEMS, the Stack Pointer must be at the top of the RAM (it's a RTEMS's requirement).

4 How to test this project ?

The official TestSuite of RTEMS was used to test the project. If you want to try the TestSuite by yourself, you must compile RTEMS 4.10 in this way :

```
$ mkdir b-rtems
```

```
$ cd b-rtems
```

```
$ ../rtems-4.8.0/configure --target=sparc-rtems4.8 --enable-rtems-  
bsp=leon2 --prefix=/opt/rtems-4.8/ --enable-tests
```

```
$ make all
```

```
$ sudo PATH=/opt/rtems-4.11/bin:${PATH} make install
```