NAME

mgsimdev-arom − Active ROM pseudo-device in MGSim

DESCRIPTION

An Active ROM is a combination of a (passive) ROM and an (active) DMA controller. It supports both explicit reads to the ROM's contents and DMA requests to push the ROM contents to the shared memory asynchronously.

Because DMA accesses are performed within the on−chip memory system, this kind of access is called "Direct Cache Access" (DCA).

An I/O device of this type can be specified in MGSim using the device type **AROM**.

CONFIGURATION

Each **arom** device support the following configuration variables:

<dev>:ROMContentSource
: Specifies what data is loaded in the ROM. Can be either:

  RAW Unprocessed bytes. The ROM contains the exact contents of the input file specified with **ROMFileName**.

  ELF The input file specified with **ROMFileName** is loaded as a ELF binary image containing multiple sections.

  CONFIG
  : The ROM contains the MGSim configuration space, using the binary format described in mgsimconf(7).

  ARGV The ROM contains the command−line arguments passed to **mgsim**, after option processing. The ROM contents is laid out as follows:

Address	Description
bytes 0−3	The magic value 0x56475241.
bytes 4−7	The number of MGSim command−line arguments.
bytes 8−11	The number of bytes following.
bytes 12+	The MGSim command−line arguments, nul−separated.

<dev>:ROMFileName
: The input file name for **RAW** and **ELF** content sources.

<dev>:ROMBaseAddr
: Address in shared memory at which DCA transfers will inject the ROM contents. Valid for all content sources except **ELF**: the ELF file format specifies its own load target addresses. Can be overriden at run−time.

<dev>:PreloadROMToRAM
: If set to true, MGSim will preload the ROM contents into the shared memory prior to system starts up. This enables programs to use the ROM data directly without using the I/O interface.

<dev>:DCATargetID
: Default target device on the I/O network for DCA transfers. Can be overriden at run−time.

<dev>:DCANotificationChannel
: Default notification channel to signal when a DCA transfer has completed. Can be overriden at run−time.

PROTOCOL

Direct reads

Any *I/O read* request will read the ROM bytes, unmodified, at the offset specified in the request.

DCA loads

A DCA transfer is triggered by sending an *I/O write* request to address 0. The integrated DMA controller will then transfer the bytes from the ROM to the shared memory system across the I/O network. When the transfer is completed, a notification is sent to the I/O network on a predefined channel. A program can thus wait asynchronously on completion of the DCA transfer.

The target device where to send the data, as well as the DCA parameters (source/destination addresses, and size) can be configured by sending *I/O write* requests to offsets 4−28, as described in *INTERFACE* below.

The effect of changing the DCA configuration *during* a DCA transfer is undefined.

INTERFACE

The device presents itself to the I/O bus as a single device.

Address	Access width	Mode	Description
0+	(any)	Read	Read the ROM contents
0	4 bytes	Write	Start DCA transfer
4	4 bytes	Write	Set DCA target dev ID (bits 0−15) + notification channel (bits 16−31)
8	4 bytes	Write	Base source address for DCA (low bits)
12	4 bytes	Write	Base source address for DCA (high bits)
16	4 bytes	Write	Base target address for DCA (low bits)
20	4 bytes	Write	Base target address for DCA (high bits)
24	4 bytes	Write	Number of bytes for DCA (low bits)
28	4 bytes	Write	Number of bytes for DCA (high bits)

Any **arom** device further supports the following commands on the MGSim interactive prompt:

info <dev>

Reports the size, current DCA configuration and loadable memory ranges.

read <dev> <addr> <size>

Read **<size>** bytes of the ROM data starting from relative address **<addr>**.

SEE ALSO

mgsim(1), mgsimdoc(7), mgsimconf(7)

BUGS

Report bugs & suggest improvements to *microgrids@svp−home.org*.

AUTHOR

MGSim was created by Mike Lankamp. MGSim is now under stewardship of the Microgrid project. This manual page was written by Raphael 'kena' Poss.

COPYRIGHT