

Table of Contents

Name of the exercise.....	1
Responsible for the exercise.....	2
Description of the exercise.....	3
What will the students learn.....	4
Duration.....	5
List of material.....	6
Relevant Information.....	7
Solution.....	9
Preparation of the exercise in Ankara.....	10

Name of the exercise

VMEbus data transfer

Responsible for the exercise

Markus Joos

Description of the exercise

The students will write a C program that executes VMEbus single cycles and block transfers to write data to a VME memory and to read it back.

What will the students learn

- Writing a simple C program to exchange data with a VMEbus module
- Understand and use an existing VMEbus library
- Compare the relative speed of different VMEbus transfer modes

Duration

2 hours. Time permitting the students can refine the program in the free time to go deeper into S/W performance profiling

List of material

H/W:

- 1 VMEbus crate
 - ◆ The Pool does not seem to have enough usable 7-slot crates
- 1 CCT SBC
- 1 VMEbus D32/MBLT memory module (porbably a RIO2)
- 1 VMEbus display module VMDIS 8004
 - ◆ This is not a must
- 1 computer screen & keyboard
 - ◆ Laptop with N/W connection is sufficient

S/W:

- Linux file system and gcc compiler / linker
- vme_rcc, cmem_rcc and io_rcc driver
- TDAQ RCD S/W (vme_rcc and related libraries)

Relevant Information

- The students have heard the lecture on modular electronics
- The students should have read:
 - ◆ vme_rcc API [↗](#)
 - ◆ cmem_rcc API [↗](#)
 - ◆ rcc_time_stamp API

Installation guide:

1. Collect the H/W and turn the crate on.
2. Make sure the SBC boots up. In case of problems contact the sysadmin. The SBC gets a Linux image from the central server.
3. Log on with the DAQ school account (daqschool / daqsch001) and use "lsmod" and "more /proc/xxx_rcc" to check if the drivers (vme_rcc, io_rcc and cmem_rcc) are loaded. The "proc" file of vme_rcc has to say that the Universe chip has been initialized
4. Prepare the VMEbus slave (see below)
5. Run "scanvme" and check if the memory module is seen at address 0x08000000
6. Change directory to /exercise1/solution and run "make" to check if the example solution can be compiled
7. Change directory to /exercise1/student and remove all files that may have been left by previous groups. Just keep the "makefile"

VMEbus slave preparation:

If the VMEbus slave of your set-up is a "RIOS 8061" you have to configure the mapping of the memory.

1. Connect the RS232 cable to the "CONS." port (in terms of cables you need the RS232 adapter of the PBTM315 and the RS232-USB adapter)
2. do not use a "NULL MODEM"
3. Run the command "config" to check the current configuration. You want this:
 - ◆ A24 base disabled
 - ◆ A32 base enabled. Base address = 0x08000000
4. If you don't have this run the command "set vme"
 - ◆ Configure VMEbus slave mapping. Take the default values for all parameters but:
 - ◆ vme_slv_a24 = 0x0.d
 - ◆ vme_slv_a32 = 0x08000000.e

Instruction sheet:

1. Log on with the DAQ school account (daqschool / daqsch001) and change directory to /exercise1/student
2. Start an editor session (vi, nedit) and write a program that uses the vme_rcc library to execute the VMEbus cycles listed below:
 1. Write 0x12345678 to address 0x0 in A32 / D32 mode. Use the "safe" cycles
 2. Read the data back from address 0x0 and compare it
 3. Write 0x12345678 to address 0x4 in A32 / D32 mode. Use the "fast" cycles
 4. Read the data back from address 0x4 and compare it
 5. Write a block of 1 KB to address 0x1000 in A32 / D32 / BLT mode. You have to prepare the data in a cmem_rcc buffer.
 6. Read the data back in A32 / D64 / MBLT mode and compare it
3. Watch the VMEbus transfers on the display module

Time permitting:

- Instrument your code with calls of the `rcc_time_stamp` library and measure the overhead of the transactions.

Solution

A complete C-program is available in `/home/daqSchool/exercise1/solution`

Preparation of the exercise in Ankara

1. Unpack the VMEbus crate and check if the modules are all looking OK
2. Connect the upper RJ45 network port of the VP110 to the network. Make sure the server is switched on
3. Connect the console RJ45 connector with the RJ45-DB9 adapter to the COM0 port of pcdaschool1. "cd" to /kermit and run "rkermit .kermrc_vp110". Then type "connect". Switch on the VMEbus crate and check if you can see the messages from the BIOS of the VP110.
4. Check if the VP110 boots OK
5. unconnect the RS232 cable and set up a PC. This PC is required to "ssh" to the VMEbus SBC. It has to be connected to the DAQ school network. Linux is preferred. There has to be a program (xterm, etc.) that allows to "ssh" to the SBC. When the students execute this exercise they need this PC as well.
6. Log on to the SBC via ssh and "cd" to "/home/daqSchool/exercise1/"
7. source "setup"
8. "cd" to solution
9. Execute "solution"
10. report the result (screen output to Markus)

-- MarkusJoos - 2009-08-20

This topic: Sandbox > DaqSchoolExercise1

Topic revision: r5 - 2009-12-08 - MarkusJoos



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback