Minimal OpenRISC System on Chip

Installation Instructions

Windows users

The tools below are required to use the system. These tools are designed to run under Linux, there is no Windows counterpart. However, Windows users can install and use them without a problem under the Cygwin environment. If you want to use the system under Windows, proceed by installing Cygwin from (http://cygwin.com/install.html). Notice that some tools require that non-standard packages are installed. Generally, installation guides point out what is necessary. Then, run setup.exe again and select the required packages to be installed and continue installing it.

Install Icarus Verilog

1. You will need at least version 0.9.1 (ftp://ftp.icarus.com/pub/eda/verilog/v0.9/)

Download IP cores

1. download minsoc
2. download further necessary IP cores
   a) cd minsoc/rtl/verilog
   b) svn co http://opencores.org/ocsvn/adv_debug_sys/adv_debug_sys/trunk adv_debug_sys
   c) svn co http://opencores.org/ocsvn/ethmac/ethmac/trunk ethmac
   d) svn co http://opencores.org/ocsvn/openrisc/openrisc/trunk/or1200 or1200
   e) svn co http://opencores.org/ocsvn/uart16550/uart16550/trunk uart16550

Install GNU toolchain and adv_jtag_bridge

1. Follow: http://www.opencores.org/openrisc,gnu_toolchain (to install binutils, gcc, gdb)
2. To debug and load the firmware you have to use the new advanced_debug_system. This project is included in the minsoc files inside of minsoc/rtl/verilog/adv_debug_sys. There you can find the software in Software and the documentation, which shall help you to go under Doc.
   a) change the Makefile in minsoc/rtl/verilog/adv_debug_sys/Software/adv_jtag_bridge and compile the software using make.
      - change Makefile, “INCLUDE_JSP_SERVER=true” to “INCLUDE_JSP_SERVER=false”
      - make
      - sudo make install
   b) If you have a Xilinx FPGA: copy the description file of **your** FPGA to your home directory (e.g. “cp /opt/Xilinx/10.1/ISE/spartan3e/data/xc3s500e_fg320.bsd ~/”)

3. With the adv_jtag_bridge you can also debug your simulation. To do so, the simulation has to include a vpi module. This has to be compiled by your system. The sources are found under “minsoc/rtl/verilog/adv_debug_sys/Software/adv_jtag_bridge/sim_lib/icarus”.
   a) cd minsoc/rtl/verilog/adv_debug_sys/Software/adv_jtag_bridge/sim_lib/icarus
   b) make
   c) cp jp-io-vpi.vpi minsoc/bench/verilog/vpi
4. Check gdb version, patch it if version 6.8:
   a) or32-elf-gdb -v

      Building automata... done, num uncovered: 0/216.

      Parsing operands data... done.

      GNU gdb 6.8

      Copyright (C) 2008 Free Software Foundation, Inc.

      License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

      This is free software: you are free to change and redistribute it.

      There is NO WARRANTY, to the extent permitted by law. Type "show copying"

      and "show warranty" for details.

      This GDB was configured as "--host=i686-pc-linux-gnu --target=or32-elf".

   b) Proceed as in FAQ.pdf, “GDB reports “Value being assigned to is no longer active.”, what happened?”