SBST contest 2016/17: openMSP-430

04JCJLI- Testing and fault tolerance

Files in ~/openmsp430_testing/

doc/* – openMSP-430 documentation
core/bench/* – openMSP-430 test-bench
core/gate/* – openMSP-430 synthesized design and simulation library models
core/rtl/* – openMSP-430 HDL source code
core/sim/rtl_sim/bin/* – simulation and fault-simulation scripts
core/sim/rtl_sim/src/* – openMSP-430 example assembly programs
core/sim/rtl_sim/src-c/* – openMSP-430 example C projects
core/sim/rtl_sim/src/selftest.s43 – openMSP-430 SBST program source code (assembly)
core/sim/rtl_sim/src/selftest.v – openMSP-430 SBST program stimuli (no need to edit)
core/sim/rtl_sim/run/* – simulation and fault-simulation run environment

clean.sh – shell script used to remove previous simulation and fault-simulation runtime files
run_selftest_sim_gui.sh – shell script that compiles and performs logic simulation (with GUI)
run_selftest_sim_nogui.sh – shell script that compiles and performs logic simulation (shell-only)
run_fsim.sh – shell scripts that performs fault-simulation of the last simulated program

SBST_07321794.pdf – IEEE paper about SBST generation flow (more references to other papers inside).

System configuration

The processor has been synthesized using the configuration reported in **openMSP430_defines.v** (in the *rtl* directory). All extra features have been removed from the processor, such as Watchdog timer, DMA interface support, non-maskable-interrupt support, debug interfaces, and clock gating. The system has been configured with 48 kB program memory, 10 kB data memory, and 512 bytes peripheral memory.

Check the file **pmem.x** produced in the *run* directory after the first simulation for the processor's memory map.

Note: you cannot modify the system configuration Verilog file, since your RTL and gate-level design must refer to the same system (the configuration is hard-coded in the gate-level design)[1].

Test program

A simple test program skeleton **selftest.s43** has been created under the *src* directory. You can modify this file in order to create the final test program. It is important to implement a mechanism to stop the simulation. This can be done either by means of a stimuli file (**selftest.v**) or via software. It is recommended to stop the simulation via software, by performing a branch to the address 0xFFFF.

Note: you are allowed to create a C/assembly complex project, instead of writing everything into a single assembly file. You can refer to the Dhrystone project (under *src-c*). It is up to you to adjust the makefile, the stimuli Verilog file, and the simulation scripts.

Links

TI: MSP430x1xx Family User's Guide (including ISA): http://www.ti.com/lit/ug/slau049f/slau049f.pdf

[1] In case a problem in the current system configuration is found, instructions on how to fix it will be given. You are highly encouraged to use the forum in the course page to exchange information about the processor, eventual issues, and solutions. The teachers will periodically try to look at the forum threads. Please note that teachers are not alerted when new messages or threads are created, thus you should also send an email if no response will be given.