=============================================================================

 Instructions for OpenSPARC T2 Version 1.0 SPARC Architecture Model (SAM)

=============================================================================

1. Set up SAM

 Create a directory for installing SAM and change to that directory.

 e.g.

 mkdir -p /home/johndoe/OpenSPARC\_SAM\_work

 cd /home/johndoe/OpenSPARC\_SAM\_work

 copy the SAM package tarball into the directory

 and untar it.

 Setup environment variables by editing OpenSPARCT2\_SAM.cshrc file.

 Please set the following variables in OpenSPARCT2\_SAM.cshrc file

 SIM\_ROOT Directory location where you Extracted

 the OpenSPARCT2\_Arch\_1.0.tar file. e.g.

 /home/johndoe/OpenSPARCT2\_SAM

 SUN\_STUDIO Directory location for Sun Studio installation

 e.g. /usr/dist/pkgs/sunstudio\_sparc/SUNWspro

 Source the environment variable file above by using following command :

 source OpenSPARCT2\_SAM.cshrc

 You may want to add the above command to your ~/.cshrc file, so that

 every time you login, it will source the above file.

2. Compile/build for RTL verification cosimulation :

 cd $SIM\_ROOT/sam-t2/sam

 On a SPARC machine:

 make n2-cosim-pkg

 A nas,5.n2.opens.$USER directory will be created in the working directory,

 copy/move the directory to the proper RTL verification tool area, to be

 used in RTL verification.

3. Compile/build a full-system simulator :

 cd $SIM\_ROOT/sam-t2/sam

 On a SPARC machine:

 make n2

 a install-n2 directory will be created in the working directory, which

 contains the binary files for a SAM full-system simulator.

4. Run SAM full-system simulator :

 Assuming you create a test directory $SIM\_ROOT/test

 A. Set up Solaris boot files

 cd $SIM\_ROOT/test

 $SIM\_ROOT/sam-t2/sam/install-n2/bin/getsolaris \

 -d $SIM\_ROOT/sam-t2/config -v int12 -p n2 int12

 This will create an int12 directory with all the related boot

 files in it. Change to that directory and run.

 -------------------------------------------------------------------

 NOTE - "-d" option in getsolaris command must have absolute path.

 Relative path will not work.

 -------------------------------------------------------------------

 cd $SIM\_ROOT/test/int12

 $SIM\_ROOT/sam-t2/sam/install-n2/bin/simulate -c 1c1t -s -w 1

 This will run a 1c1t configuration on one physical cpu. Available

 configurations are 1c1t, 1c2t, 1c8t, 2c8t, 4c8t, 8c8t.

 ------------------------------------------------------------------

 NOTE - "getsolaris" command above will copy boot files from

 $SIM\_ROOT/sam-t2/config/n2/solaris directory. If you have generated

 new binaries for reset, hypervisor, and/or OBP, then please copy

 those new binary images to $SIM\_ROOT/sam-t2/config/n2/solaris

 directory so that SAM sees those new versions.

 You can use "setup\_sam.sh" script provided in $SIM\_ROOT/bin for

 this task.

 ------------------------------------------------------------------

 B. This will run SAM which will pop-up two windows called "Guest Console"

 and "Hypervisor Console".

 C. From Sam Console's command prompt type 'run'. Shortly after, an 'ok'

 prompt will come up in the "Guest Console".

 D. When 'ok' prompt shows up in "Guest Console", type

 boot vdisk

 or to display more informaiton during booting

 boot vdisk -vV

 E. When 'login:' prompt shows up in "Guest Console", type 'root',

 no password is needed. Now you are in Solaris as root.

 F. Type help from Sam Console's command prompt to learn the

 available commands for controlling the simulator.

 G. To generate a instruction trace file, do the following from

 Sam Console:

 stop

 mod load analyzer rstracer.so

 rstrace -o <outputfile> -n <number\_of\_instructions>

 run (or stepi <number\_of\_instructions>)

 stop

 mod unload analyzer

 The generated trace file can be viewed by trconv in bin directory.

 In general the trace files are compressed, the rstzip in bin

 directory can be used to unzip the files. In case rstzip

 encounters stacksize problem, try 'ulimit -s 1048576'

 (or if using csh 'unlimit stacksize') to increase stacksize.

 H. To generate a checkpoint for later restore purpose, enter the command

 dump <dirname>

 from Sam Console, the command will create a checkpoint (directory).

 The checkpoint data can then be restored by starting the sam simulator

 with -R option.

 When restore from a checkpoint, do not enter any command until the message

 "----- RESTORE COMPLETED -----" shows up.

 NOTE: when generating checkpoint, make sure you don't have commands like

 'run' or 'stepi' in the config file (e.g., sam.rc)

5. Run SAM full-system simulator with Serial Attached SCSI driver (SAS)

 Assuming you create a test directory /home/johndoe/OpenSPARC\_SAM\_work/test

 A. Set up Solaris boot files

 cd $SIM\_ROOT/test

 $SIM\_ROOT/sam-t2/sam/install-n2/bin/getsolaris -d $SIM\_ROOT/sam-t2/config -v sasdisk -p n2 sasdisk

 This will create an sasdisk directory with all the sas related boot

 files in it. Change to that directory and run.

 cd $SIM\_ROOT/test/sasdisk

 $SIM\_ROOT/sam-t2/sam/install-n2/bin/simulate -c 1c1t -s -w 1 -I system.rc

 This will run a 1c1t configuration on one physical cpu. Available

 configurations are 1c1t, 1c2t, 1c8t, 2c8t, 4c8t, 8c8t.

 The other steps are similar to Section 4 above, except step D, where

 the following boot command should be used:

 boot /pci@0/pci@0/pci@0/scsi@0/disk@0,0:c -vV

 Another new feature in this setup is the Local Loopback File System

 driver (llfs), llfs allows users to move files in and out of a

 simulated system. After login as described in step E, /ll/root is

 where the file system of the underlying real machine is mounted, any

 modification to a file under /ll/root or its subdirectories will have

 immediate effect on the real file. For example, from within a simulated

 system, one can modify files in

 /ll/root/home/johndoe/OpenSPARC\_SAM\_work/sam-t2

 which will have the same effect as modify files in

 /home/johndoe/OpenSPARC\_SAM\_work/sam-t2

 from a real machine.

6. Save a modified disk image

 When booting up a Solaris disk image, the SAM full-system simulator

 will display a message about the loaded disk image:

 UI(load): loading <disk1> memory image ...

 loading disk1, base addr 0x0000001f40000000, size 0x20000000

 Remember the address and size, they will be needed in saving the content

 of the disk image to a new file (by using memdump command).

 After the simulated system is booted up, new file can be created in the

 system, or existing file can be modified. The 'memdump' command can then

 be used to save those changes.

 Before issuing the 'memdump' command, the simulated system's file system

 must be synced up first, to achieve that, enter the command 'sync' from the

 simulated system's shell prompt, followed by a 'halt' command. Wait until

 the following messages are displayed:

 syncing file systems... done

 Program terminated

 At this point, enter 'stop' from sam command console, followed by 'memdump'

 command, e.g.,

 memdump disk.memdump 0x1f40000000 0x20000000

 Do not issue other command until the message "----- MEMDUMP COMPLETED -----"

 is displayed. The created memdump file can then be examined (and move file

 in/out of it), by using lofiadm/mount commands on a real Solaris system.

 The new disk image can also be used for Solaris bootup.