

OpenPiton in Action

Princeton University

<http://openpiton.org>



OpenPiton

Simulating OpenPiton RTL

Anatomy of a Simulation

- Simulation model
 - Design under test (DUT) RTL
 - Top-level test bench
 - Simulator compiler arguments
 - Verilog macros, include directories, monitor params, etc.
- Test stimuli
 - Assembly tests
 - C tests
 - Source/sink bit vectors
 - Based on infrastructure from Christopher Batten's group at Cornell

OpenPiton Simulation Models

Table 3: OpenPiton Simulation Models

Name	Type
manycore	C/Assembly
chip_fpga_bridge	Unit Test
dmbr	Unit Test
dmbr_test	Unit Test
fpga_chip_bridge	Unit Test
fpga_fpga_hpc_bridge	Unit Test
fpga_fpga_lpc_bridge	Unit Test
ifu_esl	Unit Test
ifu_esl_counter	Unit Test
ifu_esl_fsm	Unit Test
ifu_esl_htsm	Unit Test
ifu_esl_lfsr	Unit Test
ifu_esl_rtst	Unit Test
ifu_esl_shiftreg	Unit Test
ifu_esl_stsm	Unit Test
jtag_testbench	Unit Test

Simulation Scripts/Tools

- sims
 - piton/tools/src/sims/sims, 1.262
 - Adapted from OpenSPARC T1
 - Build individual simulation models and run test stimuli
 - Regressions
 - Single simulation model
 - Supports Synopsys VCS, Cadence Incisive, and Icarus Verilog
 - Synopsys VCS recommended
- contint
 - piton/tools/src/contint/contint, 1.0
 - Calls sims
 - Continuous integration bundles
 - Multiple simulation models
 - Currently only supports SLURM job scheduler and Synopsys VCS

Simulator Choice

- Synopsys VCS, Cadence Incisive and Icarus Verilog are all supported
- Commands shown are for Icarus Verilog
- For VCS, replace `icv` with `vcs`
- For Cadence Incisive, replace `icv` with `ncv`

Building a Simulation Model

- Required sims arguments
 - sys=<simulation model>
 - icv_build
- Other useful arguments
 - icv_build_args=<ICV arguments>
 - debug_all

Simulation Model Build Outputs

- stdout and sims.log
- build/<simulation_model>/<build_id>/
-build_id=<name>
 - Default is rel-0.1

Example: The manycore Model

- sims -sys=manycore -icv_build
 - sims.log: check for build errors
 - Check for SIGDIE
 - build/manycore/rel-0.1/

Hands-on: The manycore Model

- cd \$PITON_ROOT/build
- sims -sys=manycore -icv_build
- Check Outputs
 - sims.log: check for build errors
 - build/manycore/rel-0.1/a.out

Running a Simulation

- Required sims arguments
 - sys=<simulation model>
 - icv_run
 - <test stimuli>
 - Varies by simulation model type (assembly file, source/sink prefix)
- Other useful arguments
 - build_id=<name>
 - gui
 - Requires -debug_all during build

Simulation Outputs

- Depends on test type
- stdout and sims.log
 - PASS (HIT GOOD TRAP)
- Test binary (diag.exe)
- Memory image (mem.image)
- Assembler log (midas.log)
- Symbol table (symbol.tbl)
- Performance log (perf.log)
- Status log (status.log)

Example: Assembly Test Simulation

- sims -sys=manycore -icv_run
princeton-test-test.s
 - C tests have similar syntax

Hands-on: Assembly Test Simulation

- cd \$PITON_ROOT/build
- sims -sys=manycore -icv_run
princeton-test-test.s
 - Should take 2-5mins

Example output

```
25123000: C0: WM: 0100000000
spc0: 25123500      0    sethi:04 0    spl:00 0      -:xx 0      -:xx 0      -:xx
spc0: 25124500      0    -:xx 0    -:xx 0    spl:00 0      -:xx 0      -:xx
25125000: C0: WM: 0000000000
spc0: 25125500      0    -:xx 0    -:xx 0    -:xx 0    spl:00 0      -:xx
25126500:tpc1_reg-updated -> spc(0) thread(0) window(0) val = 000020000068
25126500:tnpc1_reg-updated -> spc(0) thread(0) window(0) val = 00002000006c
25126500:tstate1_reg-updated -> spc(0) thread(0) window(0) val = 4400001200
25126500:ttype1_reg-updated -> spc(0) thread(0) window(0) val = 100
25126500:htstate1_reg-updated -> spc(0) thread(0) window(0) val = 0
25126500:pc-updated -> spc(0) thread(0) window(0) val = 000000122000
25126500:npc-updated -> spc(0) thread(0) window(0) val = 000000122004
25126500:tl_reg-updated -> spc(0) thread(0) window(0) val = 1
25126500:reg_updated -> spc(0) thread(0) window(1) rs1(1a)->0000000000000000 rs2(00)->0000000000000000 reg#g1 val = 0000000000000001
(25126500)Info-perm spc(0) thread(0) pc(000000000122000) npc(000000000122004) opcode(83468000)
spc0: 25126500 0 00122008 0 wryspl:04 0      -:xx 0      -:xx 0      -:xx 0    spl:00
Info: spc(0) thread(0) Hit Good trap
25127000: C0: WM: 0100000000
spc0: 25127500      0    sethi:08 0    wryspl:04 0      -:xx 0      -:xx 0      -:xx
25127500: Simulation -> PASS (HIT GOOD TRAP)
$finish called from file "/tank/mmckeown/research/projects/piton/openpiton/piton/verif/env/manycore/pc_cmp.tmp.v"
, line 355.
$finish at simulation time          25127500
V C S   S i m u l a t i o n   R e p o r t
Time: 25127500 ps
CPU Time:      2.250 seconds;      Data structure size:  1.6Mb
Sat Jun 11 01:08:19 2016
sims: sim_stop Sat Jun 11 01:08:19 EDT 2016
sims: "perf > perf.log"
sims: "regreport -1 > status.log"
sims: stop time Sat Jun 11 01:08:20 EDT 2016
```

Debugging Simulations

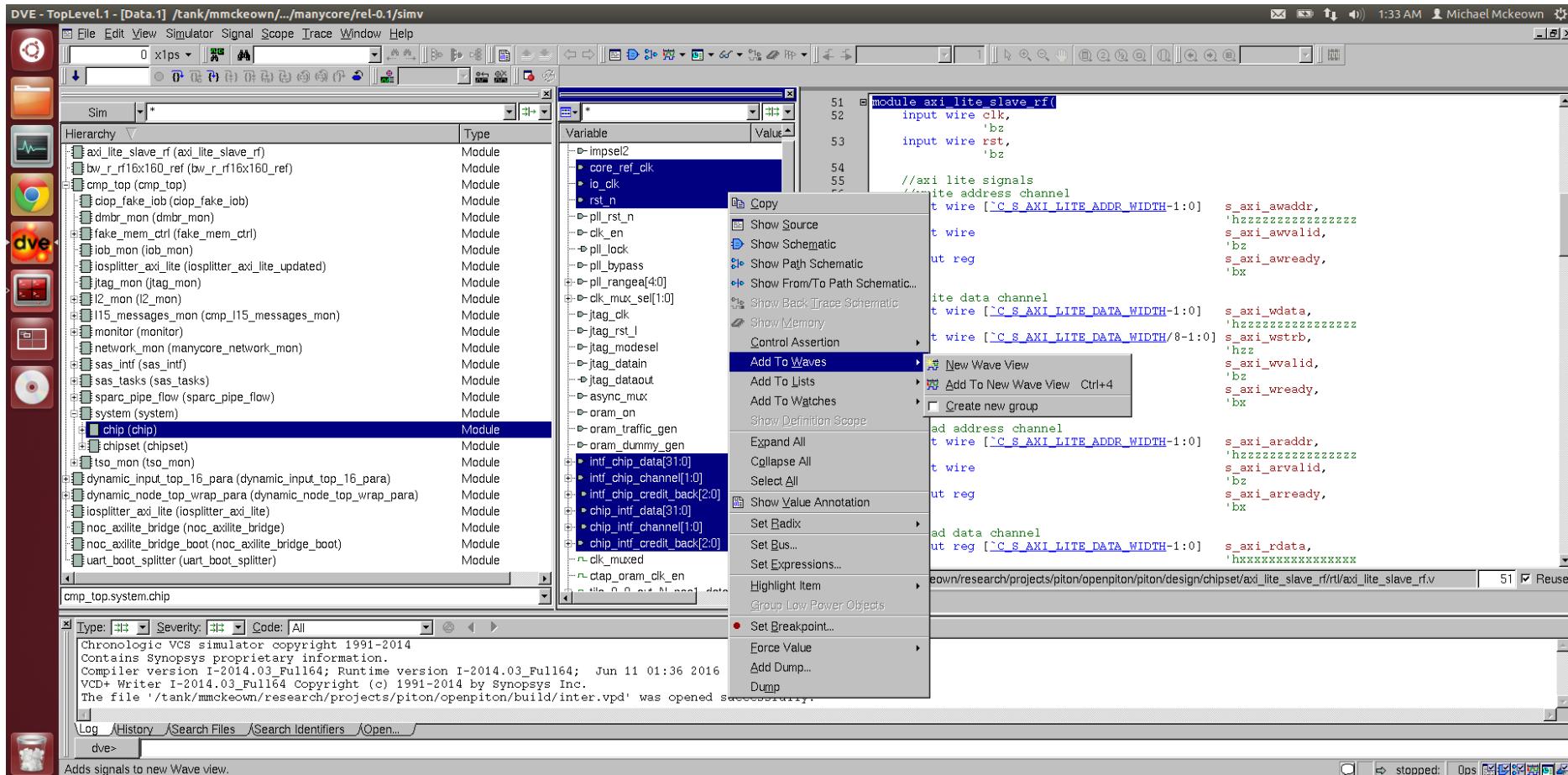
- Monitors (manycore)
 - Non-synthesizable Verilog modules
 - Instantiated in top-level test bench
 - X-module references DUT signals
 - Print useful output
 - Check properties
- Tools for parsing simulation output
 - pc_grep <log>, reg_grep <log>, etc.

Debugging Simulations

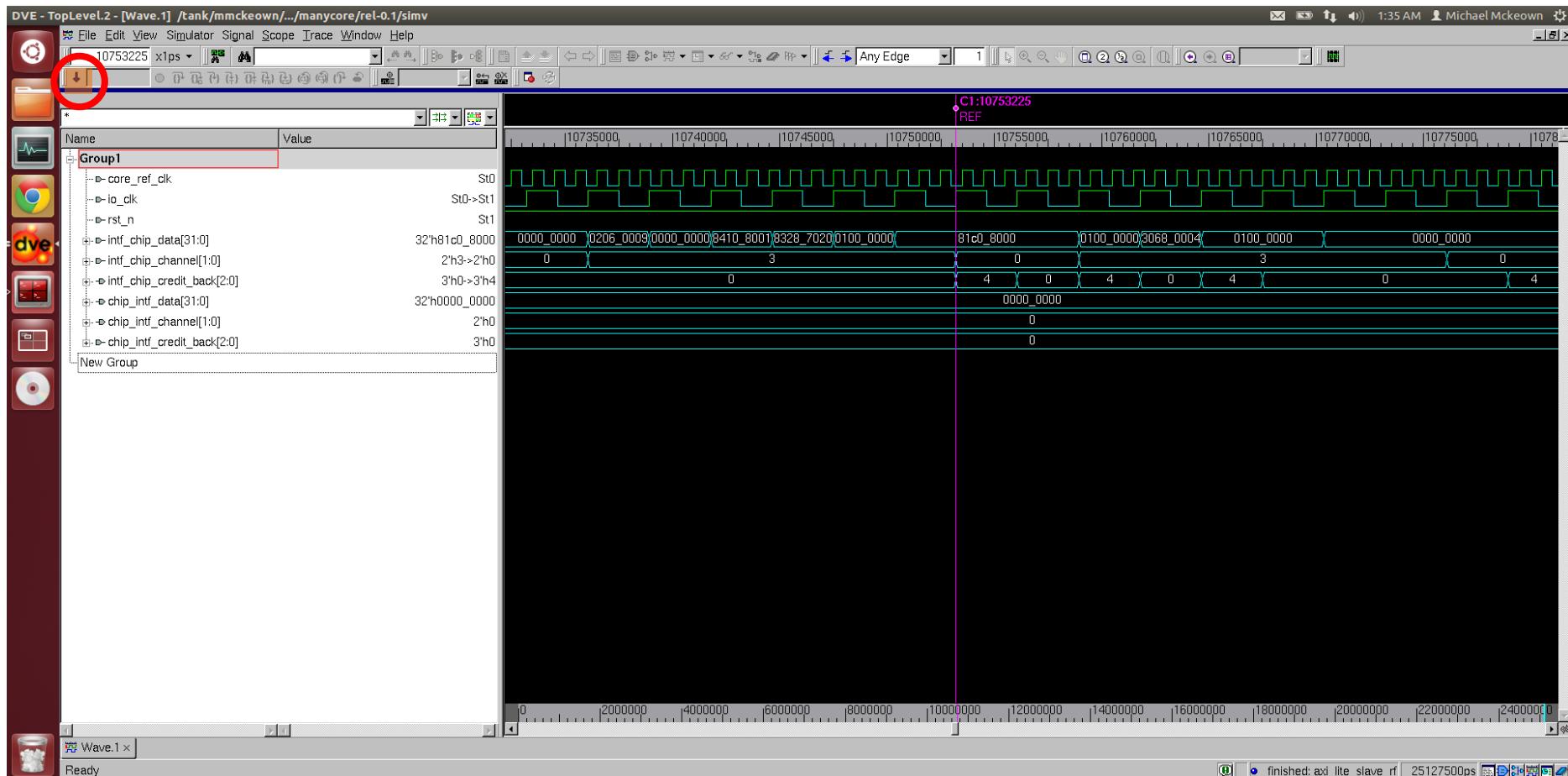
Debugging Simulations with VCS & DVE

- Waveforms - DVE
 - Build with `-debug_all`
 - Run with `-gui`
- Example:
 - `sims -sys=manycore -vcs_build
-debug_all`
 - `sims -sys=manycore -vcs_run
princeton-test-test.s -gui`

Debugging Simulations



Debugging Simulations



Exploring the assembly test suite

- piton/verif/diag/assembly/
- Diaglists (piton/verif/diag/*.diaglist)
 - Groups of assembly tests and assembly test declarations
 - Assembly test declaration

```
label testfile.s <sims arguments>
```
 - Assembly group definitions

```
<groupname sys=mymodel sims args>
    test1 test1.s
</groupname>
```
 - Groups can be nested

Diaglists

```
<tile1 sys=manycore -x_tiles=1 -y_tiles=1>
<cmp_default name=default>

<all_tile1_passing>
<all_tile1_passing_no_rtl_csm>

<tile1_mini>
<princeton-test>
    princeton-test-test      princeton-test-test.s -finish_mask=3 -midas_args=-DTHREAD_COUNT=2 -midas_args=-DTHREAD
    _STRIDE=1
    basic-io-test           basic-io-test.s
    uart-hello-world        uart-hello-world.s
</princeton-test>

<tile1_mini_icache>
    bypass_win              bypass_win.s
    //// fail_perf_chase_l1hit    chase.pal -midas_args=-pal_diag_args=-lenbytes=512 -midas_args=-pal_diag_args=
    stride=16 -midas_args=-pal_diag_args=-nodisablel1_warmup -midas_args=-pal_diag_args=-expect_time=3 -midas_args=-pa
    l_diag_args=-expect_string=L1_ld_hit
    dmiss_imiss             Dmiss_imiss.s
    done_retry_trap          done_retry_trap.s
    exu_alu                 exu_alu.s
    hp_reg_rdwr              hp_reg_rdwr.s
    ihit_sameset             Ihit_sameset.s
    imiss_branches           Imiss_branches.s
    imiss_oddeven            Imiss_oddeven.s
    intr_basic2              intr_basic2.s
```

Common Test Flags

- `-rtl_timeout=`
 - Number of cycles sims will wait before timing out the test
- `-sim_run_args=`
 - Arguments (e.g. plusargs) to Verilog simulator
- `-midas_args=`
 - Arguments to assembler, midas
 - Thread count, thread stride, and more
- `-finish_mask=`
 - Mask specifying threads to wait for

Types of tests

- Thousands of assembly tests
 - IFU, TLU, etc
 - arch
 - fp, exu, mem, trap, etc.
 - TSO tests
 - PAL-generated (randomized) tests
 - C tests

Running a Regression

- Groups of tests as defined in diaglists
- Tests utilize the same simulation model
 - One build, multiple test runs
- `sims -sim_type=vcs -group=<regression name>`
 - Simulation model specified by group declaration
 - `-sim_type=vcs` replaces `-vcs_build` and `-vcs_run`

Running a Regression

- Example:

```
sims -sim_type=icv -group=tile1_mini
```

Table 4: OpenPiton Regression Suites

Name	Description
all_tile1_passing	All single tile tests
tile1_mini	a mini set of single tile tests
all_tile2_passing	All 2-tile tests
tile2_mini	a mini set of 2-tile tests
tile4	All 4-tile tests
tile8	All 8-tile tests
tile16	All 16-tile tests
tile36	All 36-tile tests
tile64	All 64-tile tests

Regression Outputs

- Simulation model will be built as usual in build/<simulation_model>/<build_id>/
- Tests run sequentially
 - Test results stored in build/<date>_<id>
- Check results
 - regreport <test results directory>

Regression Outputs

```
Summary for /tank/mmckeown/research/projects/piton/openpiton/build/2016_06_11_0
=====
Status:tile1_mini | ALL |
-----
    PASS:      46 |      46 |
    FAIL:      0 |      0 |
    Diag Problem: 0 |      0 |
    License Problem: 0 |      0 |
    MaxCycles Hit: 0 |      0 |
    Socket Problem: 0 |      0 |
    Timeout:      0 |      0 |
    LessThreads:   0 |      0 |
    Simics Problem: 0 |      0 |
    Performance:   0 |      0 |
    Killed By Job Q: 0 |      0 |
        Unknown: 0 |      0 |
        UnFinished: 0 |      0 |
    flexlm error: 0 |      0 |
-----
    Diag Count: 46 | 46 |
-----
    Cycles/Sec: 10350353 | 10350353 |
    K Cycles: 2018733 | 2018733 |
    #Diags Used: 46 | 46 |
=====

Details for tile1_mini
=====
PASS:
=====
Diag: basic-io-test:default:tile1_mini:0:job64903903  PASS
Sat Jun 11 02:27:09 EDT 2016
sim.log: 26118500: Simulation -> PASS (HIT GOOD TRAP)
Cyc= 26118500, Sec= 1.910, C/S=13674607.3
sims: group_name = tile1_mini
sims: regress_date = 2016_06_11
sims: regress_time = 02_26_15
sims: /tank/mmckeown/research/projects/piton/openpiton/build/manycore/tile1_2016_06_11_0/simv +cpu_num=0 +dowarnin
gfinish +doerrorfinish +spc_pipe=0 +vcs+dumpvaroff +TIMEOUT=50000 +wait_cycle_to_kill=10 +tg_seed=0 +good_trap=00
0082000:1000122000 +bad_trap=0000082020:1000122020 +efuse_data_file=efuse.img +asm_diag_name=basic-io-test.s +efu
se_image_name=default.dat +dv_root=/tank/mmckeown/research/projects/piton/openpiton/piton
=====
Diag: bypass_win:default:tile1_mini:0:job57238050  PASS
Sat Jun 11 02:27:43 EDT 2016
sim.log: 26643500: Simulation -> PASS (HIT GOOD TRAP)
Cyc= 26643500, Sec= 2.150, C/S=12392325.6
sims: group_name = tile1_mini
sims: regress_date = 2016_06_11
```

Continuous Integration Bundles

- Infrastructure for large scale continuous integration testing
- Supports multiple different simulation models
- Specified by XML files

Continuous Integration Bundles

```
<bundles>

<bundle_name>

    <asm_test name="asm_test_name">
        <sys>sim_model</sys>
        <asm_diag_name>test.s</asm_diag_name>
    </asm_test>

    <asm_regress name="regress_name">
        <sys>sim_model</sys>
        <group>regression name</group>
    </asm_regress>

    <include>sub-bundle name</include>

    .
    .
    .

</bundle_name>

</bundles>
```

Continuous Integration Bundles

Table 5: OpenPiton Continuous Integration Bundles

Name	Description
git-push	a compact set of tests designed to run for every git commit
git-push_lite	a light version of git-push with fewer tests
nightly	a complete set of tests desired to run every night
pal_tests	a set of PAL tests
all_tile1_passing	All single tile tests
tile1_mini	a mini set of single tile tests
all_tile2_passing	All 2-tile tests
tile2_mini	a mini set of 2-tile tests
tile4	All 4-tile tests
tile8	All 8-tile tests
tile16	All 16-tile tests
tile36	All 36-tile tests
tile64	All 64-tile tests

Running a contint Bundle

- contint – continuous integration tool
 - Currently requires SLURM job scheduler
- contint --bundle=<bundle name>
- Example:
 - contint --bundle=git_push

contint Bundle Outputs

- All simulation models will be built and simulations submitted to scheduler
- Results will be aggregated and printed to `stdout`
- Individual simulation results located in
 - `build/contint_<bundle_name>_<date>_<id>`
- Re-process results
 - `contint --bundle=<bundle name> --check_results --contint_dir=<results directory>`
- Example:
 - `contint --bundle=git_push --check_results --contint_dir=$PWD/contint_git_push_2016_6_19_0`

contint Bundle Outputs

```
contint:   Checking results for bundle item 'nightly_dmbr_source_sink_18'  
contint:     Total diags: 1  
contint:   Checking results for bundle item 'nightly_dmbr_source_sink_19'  
contint:     Total diags: 1  
contint:   Checking results for bundle item 'nightly_dmbr_source_sink_20'  
contint:     Total diags: 1
```

Summary for builds and runs

Status:ASM_REGRESS	ASM_TEST	OTHER	ALL
Both PASS:	14	52	39
Build FAIL:	0	0	0
Run FAIL:	0	0	0
Item Count:	14	52	39
			105

Summary for /tank/mmckeown/research/projects/piton/openpiton/build/contint_nightly_2016_06_10_0/ by bundle item types

Status:ASM_REGRESS	ASM_TEST	OTHER	ALL
PASS:	2197	51	39
FAIL:	3	0	0
Diag Problem:	1	0	0
License Problem:	0	0	0
MaxCycles Hit:	0	0	0
Socket Problem:	0	0	0
Timeout:	0	1	0
LessThreads:	0	0	0
Simics Probem:	0	0	0
Performance:	0	0	0
Killed By Job Q:	0	0	0
Unknown:	0	0	0
Unfinished:	7	0	0
flexlm error:	0	0	0
Diag Count:	2208	52	39
			2299
Cycles/Sec:	740682	2976763	140951
K Cycles:	762291179	46122272	33146
#Diags Used:	2200	51	39
			2290

Summary for /tank/mmckeown/research/projects/piton/openpiton/build/contint_nightly_2016_06_10_0/ assembly regressions only

contint Bundle Outputs

```
Details for nightly_tile4_dmbr_tile4_tile4
=====
Details for tile4 group
=====
FAIL:
=====
Diag: br_stress:default:tile4:0:job149357      FAIL (HIT BAD TRAP)
Fri Jun 10 11:50:23 EDT 2016
sim.log: 253779500 : Simulation -> FAIL(HIT BAD TRAP)
Cyc= 253789500, Sec= 164.350, C/S=1544201.4
network_config not specified, assuming 2dmesh configurationsims: group_name = tile4
sims: regress_date = 2016_06_10
sims: regress_time = 09_16_52
sims: /tank/mmckeown/research/projects/piton/openpiton/build/manycore/contint_nightly_tile4_dmbr_tile4_tile4_2016_06_10_0/simv +cpu_num=0 +dowarningfinish
+doerrorfinish +spc_pipe=0 +asm_err_en +softint_off=1 +inst_check_off=1 +vcs+lic+wait +vcs+dumpvaroff +finish_mask=33 +TIMEOUT=500000 +wait_cycle_to_kill=10 +max_cycle=5000000 +tg_seed=0 +good_trap=0000082000:1000122000 +bad_trap=0000082020:1000122020 -cm_line+tgl+cond+branch+fsm -cm_name br_stress -cm_dir /tank/mmckeown/research/projects/piton/openpiton/build/manycore/contint_nightly_tile4_dmbr_tile4_tile4_2016_06_10_0 +efuse_data_file=efuse.img +asm_diag_name=br_stress.s +efuse_image_name=default.dat +fast_boot +dv_root=/tank/mmckeown/research/projects/piton/openpiton/piton
=====
=====
Details for ALL not in other groups
=====

Details for nightly_dmbr_assembly_1_one_bin
=====
Timeout:
=====
Diag: contint_nightly_dmbr_assembly_1_one_bin_2016_06_10_0  Timeout (TIMEOUT)
Fri Jun 10 11:52:50 EDT 2016
sim.log: 103372500 : Simulation -> FAIL(TIMEOUT)
sims: /tank/mmckeown/research/projects/piton/openpiton/build/manycore/contint_nightly_dmbr_assembly_1_one_bin_2016_06_10_0/simv +cpu_num=0 +dowarningfinish
+doerrorfinish +spc_pipe=0 +vcs+lic+wait +vcs+dumpvaroff +TIMEOUT=50000 +wait_cycle_to_kill=10 +tg_seed=0 +good_trap=0000082000:1000122000 +bad_trap=0000082020:1000122020 -cm_line+tgl+cond+branch+fsm -cm_name cm_data -cm_dir /tank/mmckeown/research/projects/piton/openpiton/build/manycore/contint_nightly_dmbr_assembly_1_one_bin_2016_06_10_0 +efuse_data_file=efuse.img +asm_diag_name=dmbr_assembly_1_one_bin.s +efuse_image_name=default.dat +dv_root=/tank/mmckeown/research/projects/piton/openpiton/piton
=====
```