OpenPiton in Action

Princeton University

Configuration

Configurability Options

Component	Configurability Options
Cores (per chip)	Up to 65,536
Cores (per system)	Up to 500 million
Threads per Core	1/2/4
Floating-Point Unit	Present/Absent
Stream-Processing Unit	Present/Absent
TLBs	8/16/32/64 entries
L1 I-Cache	8*/16/32KB
L1 D-Cache	4*/8/16KB
L1.5 Cache	Number of Sets, Ways
L2 Cache	Number of Sets, Ways
P-Mesh Intra-chip Topologies	2D Mesh, Crossbar
P-Mesh Inter-chip Topologies	2D Mesh, 3D Mesh, Crossbar, Butterfly Network
Bootloading	SD/SDHC Card, UART

*L1 cache goes to 2-ways at smallest size

Setting configuration options

- Change a flag in `sims`

- Add or edit flags in
  `piton/tools/src/sims/manycore.config`

- Examples:
  - `-config_l1i_size=16384`
  - `-network_config=xbar_config`

Configuration under the hood

- `sims` passes cache configurations as environment variables to PyHP

- Some others are passed as verilog defines

  `-config_rtl=`
  `PITON_NO_FPU`

```perl
4950   my @configurables = (
4951     "config_l1i_size",
4952     "config_l1i_associativity",
4953     "config_l1d_size",
4954     "config_l1d_associativity",
4955     "config_l15_size",
4956     "config_l15_associativity",
4957     "config_l2_size",
4958     "config_l2_associativity",
4959     );
4960
4961   foreach my $config (@configurables) {
4962     GetOptions (\%opt,
4963             "${config}=s",
4964     );
4965     my $ucn = uc($config);
4966     if ($opt{$config}) {
4967        $ENV{$ucn} = $opt{$config};
4968     }
4969   }
```

`$DV_ROOT/tools/src/sims/sims,1.262`

PyHP and parameterisation

- PyHP is a python preprocessor used in OpenPiton for generating complex RTL

- Embeds python inside the Verilog (Inspired by PHP)

  - Use print to output verilog code

  - `<% print "wire a;" %>`

- Better than Verilog's `generate`?

  - Provides visibility of intermediate code

- Not perfect but very practical

Configuration under the hood

- PYHP reads the environment variables, prints sizes to .pyv files

```
37 # cache configurations
38 CONFIG_L15_SIZE = int(os.environ.get('CONFIG_L15_SIZE', '8192'))
39 CONFIG_L15_ASSOCIATIVITY = int(os.environ.get('CONFIG_L15_ASSOCIATIVITY', '4'))
40 CONFIG_L1D_SIZE = int(os.environ.get('CONFIG_L1D_SIZE', '8192'))
41 CONFIG_L1D_ASSOCIATIVITY = int(os.environ.get('CONFIG_L1D_ASSOCIATIVITY', '4'))
42 CONFIG_L1I_SIZE = int(os.environ.get('CONFIG_L1I_SIZE', '16384'))
43 CONFIG_L1I_ASSOCIATIVITY = int(os.environ.get('CONFIG_L1I_ASSOCIATIVITY', '4'))
44 CONFIG_L2_SIZE = int(os.environ.get('CONFIG_L2_SIZE', '65536'))
45 CONFIG_L2_ASSOCIATIVITY = int(os.environ.get('CONFIG_L2_ASSOCIATIVITY', '4'))
```

piton/tools/bin/pyhplib.py

Configuration under the hood

- .pyv files use these parameters
  - Some code will print the parameters as Verilog defines

```
30    linesize = 16
31    # 512
32    numentries = CONFIG_L1D_SIZE / linesize
33    # 4
34    way = CONFIG_L1D_ASSOCIATIVITY
35    # 2
36    waywidth = math.log(CONFIG_L1D_ASSOCIATIVITY, 2)
37    # 512
38    print("`define L1D_ENTRY_COUNT %d" % numentries)
39    # 9 – 1 – 2 = 6
40    print("`define L1D_SET_IDX_HI %d" % (math.log(numentries, 2) – 1 – waywidth))
41    # 4
42    print("`define L1D_WAY_COUNT %d" % way)
43    # 2
44    print("`define L1D_WAY_WIDTH %d" % waywidth)
45 %>
```

piton/design/include/lsu.h.pyv

Configuration under the hood

- Adjusts instantiations according to # of icache ways

```
140  <%
141     template = '''
142       bw_r_ict_array ictag_ary___WAYID(
143         .we (we[__WAYID]),
144         .clk  (clk),
145            .way (`IC_WAY_IDX_WIDTH'd__WAYID),
146         .rd_data(rdtag_y[`IC_TLB_TAG_WAY__WAYID_MASK]),
147         .wr_data(wrtag_y),
148         .addr (index_y),
149         .dec_wrway_y (dec_wrway_y));
150     '''
151     for i in range(CONFIG_L1I_ASSOCIATIVITY):
152       arraytext = template.replace("__WAYID", str(i));
153       print(arraytext);
154  %>
```

piton/design/chip/tile/sparc/srams/rtl/bw_r_ict.v.pyv

Configuration Examples

- Define core count:
  - `sims -sys=manycore -x_tiles=16 -y_tiles=16 -vcs_build`
    - If specified in build command, should also be in run
- Define network topology
  - `sims -sys=manycore -network_config=xbar_config -x_tiles=4 -vcs_build`

Configuration Examples

- Cache Defaults:
  - L1 I-cache 16KB, 4-way
  - L1 D-cache 8KB, 4-way
  - L1.5: 8KB, 4-way
  - L2: 64KB, 4-way
- Decrease L1 I-cache by factor of 4:
  - ```
    sims -sys=manycore -vcs_build
      -config_l1i_size=4096
    ```

Configuration Examples

- Example: Increase icache by factor of 2:
  - `sims -sys=manycore -config_l1i_size=32768 -vcs_build`
- Passing test:
  - `sims -sys=manycore v9_allinst.s -vcs_run`
- Failing test:
  - `sims -sys=manycore err_icache_data_cecc.s -vcs_run`
  - Fails because it's not a generic, parameterised test

```
41 ! This address [11:5] will create a cache set of 0x2f (47).
42 ! This is chosen emperically to make it remain in the cache until the end of the diag.
43 ! When the diag code changes, this might fail the test...
44 #define  ITLB_ENTRY_VA      0x200045e0
45 #define  ITLB_ENTRY_PA      0x11300045e0
46 ! I-Cache Tag parity computed by hand (if PA[38:13] changes, this need be re-computed)
47 #define  ICACHE_TAG_PARITY 1
```

`piton/verif/diag/assembly/arch/error/err_icache_data_cecc.s`

Backup Slides

Adding to OpenPiton

- AXI-Lite
- Wishbone
- Interfacing with P-Mesh

Configurabilities: examples

Use the following parameters to make the test pass
(or use err_icache_data_cecc_32kb.s)

```
41    ! This address [11:5] will create a cache set of 0x2f (47).
42    ! This is chosen emperically to make it remain in the cache until the end of the diag.
43    ! When the diag code changes, this might fail the test...
44    #define  ITLB_ENTRY_VA      0x200045e0
45    #define  ITLB_ENTRY_PA      0x11300045e0
46    ! I-Cache Tag parity computed by hand (if PA[38:13] changes, this need be re-computed)
47    #define  ICACHE_TAG_PARITY 1
48    ! 12 for 16KB, 4w icache
49    #define  TAG_SHIFT 13
50    ! 0xfe0, or [11:5]
51    ! 7b of mask == 128 set entries
52    #define  SET_MASK 0x1fe0
53
```