```
// Macros used in this file:
//  PITON_FPGA_SYNTH          set to remove any RTL that is ASIC specific,
//                            such as clock gating, latches, etc.  This also
//                            ties off any ASIC IP control signals that
//                            are not used in FPGA implementations
//  PITON_NO_CHIP_BRIDGE      This indicates no chip bridge should be used on
//                            off chip link.  The 3 NoCs are exposed as credit
//                            based interfaces directly.  This is mainly used for FPGA
//                            where there are no pin constraints. Cannot be used with
//                            PITONSYS_INC_PASSTHRU. Note that if PITON_NO_CHIP_BRIDGE
//                            is set, io_clk is not really used.
//  PITON_NO_JTAG             set to remove JTAG support from Piton chip.
//                            Usually used for FPGA implementations, as JTAG
//                            is not needed.
//  PITON_CLKS_CHIPSET        indicates Piton clocks are to be generated
//                            by the chipset. Requires
//                            PITON_CHIPSET_CLKS_GEN
//  PITON_CLKS_PASSTHRU       indicates Piton clocks are to be generated by
//                            the passthru FPGA. Requires PITONSYS_INC_PASSTHRU
//                            and PITON_PASSTHRU_CLKS_GEN
//  PITON_CLKS_SIM            Piton clocks should be driven by simulated
//                            clocks from simulation testbench (input to
//                            this module).  This is set by default
//  PITON_CHIPSET_DIFF_CLK    Some chipset boards use single ended clocks and some
//                            use differential clocks as input
//  PITON_CHIPSET_CLKS_GEN    If this is set, the chipset generates it own
//                            internal clocks.  Otherwise, clocks are
//                            simulated and are inputs to this module
//  PITON_FPGA_RST_ACT_HIGH   This indicates we need to invert input reset signal
//  VC707_BOARD GENESYS2_BOARD  Used to indicate which board this code is being synthesized for
//                            There are more than just these
//  PITONSYS_INC_PASSTHRU     Set this to include the passthrough FPGA
//                            (spartan6) for real chip Piton testing. Note
//                            this macro is not compatible with
//                            PITON_NO_CHIP_BRIDGE, as it does not make
//                            sense to have the passthru FPGA if there is no
//                            chip bridge.  The design will have compile
//                            errors if both are specified
//  PITON_PASSTHRU_CLKS_GEN   Set to have the passthrough generate its own
//                            internal clocks.  Otherwise they are simulated
//  PITONSYS_NO_MC            If set, no memory controller is used. This is used
//                            in the testing of the Piton system, where a small test
//                            can be run on the chip with DRAM
//                            emulated in BRAMs
//  PITON_FPGA_MC_DDR3        Set to indicate an FPGA implementation will
//                            use a DDR2/3 memory controller.  If
//                            this is not set, a default "fake"
//                            simulated DRAM is used.
```

Config.v (inside build/manycore)

```
`timescale 1ps/1ps
`define NO_SCAN
`define PITON_NO_JTAG
`define PITON_NO_CHIP_BRIDGE
`define PITON_CLKS_CHIPSET
`define PITON_ARIANE
`define USE_FAKE_PLL_AND_CLKMUX
`define USE_FAKE_IOS
`define WT_DCACHE
`define NO_MRA_VAL
`define NO_SLAM_RANDOM
`define RTL_ARIANE0
`define RTL_ARIANE1
`define NO_SLAM_RANDOM
`define NO_MRA_VAL
```

manycore_top.tmp.v (called by Verilator)

define.tmp.h
config.v

- ! DISABLE_ALL_MONITORS
  - MONITORS
    - monitor.tmp.v — T1 sparc core monitor
    - ! MINIMAL_MONITORING
      - cmp_l15_messages_mon — L15 Monitor
      - dmbr_mon — Dual Memory Base Register Monitor
      - ! FAKE_L2
        - l2_mon — L2 Monitors
      - ! PITON_PICO_HET
        - manycore_network_mon — Monitors network activity
      - Jtag_mon — Jtag monitor
      - iob_mon — I/O Buffers monitor
      - ! VERILATOR
        - tso_mon — T1's TSO monitor, stripped of all L2 references
      - ! PITON_PICO ! PITON_ARIANE — sas_intf ?
      - ! PITON_PICO ! PITON_ARIANE — sas_tasks — Create sas tasks
      - ! PITON_ARIANE — sparc_pipe_flow — sparc pipe flow monitor
- system.v
  - chipset.v
  - chip.tmp.v
    - ! PITON_NO_CHIP_BRIDGE ! PITON_CHIP_FPGA — IBU/OBU
    - PITON_FPGA_CLKS_GEN — clk_mmcm_chip — Generate core_ref clock
    - jtag — on-chip jtag interface & test access port
    - ! ORAM_ON — oram_top ?
    - ! PITON_NO_CHIP_BRIDGE — Chip_bridge several modules
    - clk_se_to_diff — clock mux
    - clk_mux — clock mux
    - pll_top — clock mux
    - synchronizer rst_sync
    - synchronizer io_clk_rst_sync
    - synchronizer jtag_rst_sync
    - tile.tmp.v (_NOC 3_, _NOC 2_, _NOC 1_) — tile.tmp.v
    - tile.tmp.v (_NOC1_, _NOC2_, _NOC3_) — tile.tmp.v
    - OCI — Off-chip Interface Block (_NOC 3_, _NOC 2_, _NOC 1_)
    - tile.tmp.v (_NOC 3_, _NOC 2_, _NOC 1_) — tile.tmp.v
  - ! PITONSYS_INC_PASSTHRU — passthru.v

```
// Pasthru FPGA, generally used in Piton ASIC test system
// for converting single ended signals from Piton ASIC
// to differential signals to transmit to chipset FPGA.
// Can be included or removed without any difference to
// the functionality
```

tile.tmp.v

- clk_gating_latch — clock gating
- L2
- L15_wrap — L1.5
- ariane.sv
- RTAP — ?
- config_regs
- DMBR — Dual Mapping Base Register
- Conversions — credit_to_valrdy, valrdy_to_credit
- *Routers*
  - dynamic_node_top_wrap ROUTER 3
  - dynamic_node_top_wrap ROUTER 1
  - dynamic_node_top_wrap ROUTER 2

ariane.sv

- frontend — frontend
- id_stage — ?
- issue_stage — ?
- ex_stage — ?
- commit_stage — ?
- csr_regfile — ?
- perf_counters — Arrays that count performance
- controller — ?
- PITON_ARIANE
  - fifo_v3 — ?
  - rr_arb_tree — ?
- ! VERILATOR — instr_tracer — ?
- WT_DCACHE — wt_cache_subsystem — ?
- ! WT_DCACHE — std_cache_subsystem — ?

```
`timescale 1ps/1ps
`define NO_SCAN
`define PITON_NO_JTAG
`define PITON_NO_CHIP_BRIDGE
`define PITON_CLKS_CHIPSET
`define PITON_ARIANE
`define USE_FAKE_PLL_AND_CLKMUX
`define USE_FAKE_IOS
`define WT_DCACHE
`define NO_MRA_VAL
`define NO_SLAM_RANDOM
`define RTL_ARIANE0
`define RTL_ARIANE1
`define NO_SLAM_RANDOM
`define NO_MRA_VAL
```

wt_cache_subsystem.sv

- wt_icache — Instruction cache
- wt_dcache — Data cache
- PITON_ARIANE
  - wt_l15_adapter — L15 adapter
  - ! PITON_ARIANE — wt_axi_adapter — L15 adapter

wt_l15_adapter.sv

- rrarbiter — arbiter
- fifo_v2 — ?
- fifo_v2 — ?
- fifo_v2 — ?

wt_dcache.sv

- wt_dcache_missunit — Handle cache miss
- wt_dcache_ctrl — read controllers
- wt_dcache_wbuffer — store unit controller
- wt_dcache_mem — memory arrays

OpenPiton is big endian

chipset.v

- PITON_BOARD — clk_rtcm — Clock generation
- PITON_BOARD — ODDR2 — Output chip clocks, 3 modules; IBUFG — Output chip clocks
- ! PITON_NO_CHIP_BRIDGE — fpga_bridge — Convert any potential communication with chip to non-virtual channels credit based interface; credit_to_valrdy; valrdy_to_credit
- PITON_BOARD — chip_rst_seq — Bootup reset sequence
- ! PITON_BOARD ! F1_BOARD PITON_CHIPSET_CLKS_GEN — clk_mmcm — Clock generation
- ! PITON_NOC_POWER_CHIPSET_TEST — chipset_impl — chipset implementation
- PITON_NOC_POWER_CHIPSET_TEST — chipset_impl_noc_power_test — chipset implementation
- PITONSYS_INC_PASSTHRU
  - PITON_CHIPSET_CLKS_GEN — ODDR2 — Clock coverter; OBUFDS — Clock coverter
  - IBUFGDS — Clock coverter
  - OBUFDS — Clock coverter, 3 modules
  - IBUFDS — Clock coverter, 3 modules
  - ! PITON_CHIPSET_CLKS_GE — ODDR2 — Clock coverter
  - IBUFDS — Clock coverter, 3 modules
- credit_to_valrdy — Convert chipset val/rdy interface back to credit for transmission to fpga_bridge
- valrdy_to_credit — Convert chipset val/rdy interface back to credit for transmission to fpga_bridge
- PITONSYS_SPI VCU118_BOARD — ODDRE1 — sd_clk_oddr; ODDR — sd_clk_oddr
- GENESYS2_BOARD — oled_wrapper — ?
- NEXYSVIDEO_BOARD — oled_wrapper — ?
- PITONSYS_IOCTRL PITON_FPGA_ETHERNETLITE
  - ODDR — net_phy_txc_oddr
  - FD – 3 modules — net_ph_txctl_fd (0, 1, 2, 3)
  - IBUFG — net_phy_rxc_ibufg
  - BUFG — ?
  - IDELAYE2 — ?
  - IDELAYCTRL — ?

chipset_impl.tmp.v

- PITONSYS_UART_BOOT — test_end_checker — test_end_checker
- packet_filter — chip_packet_filter
- packet_filter — uart_packet_filter
- packet_filter — iob_packet_filter
- io_xbar_top_wrap — io_xbar_noc2
- io_xbar_top_wrap — io_xbar_noc3
- ! PITONSYS_NO_MC
  - PITON_FPGA_MC_DDR3 — F1_BOARD — f1_mc_top — Memory Controller; ! F1_BOARD — mc_top — Memory Controller
  - ! PITON_FPGA_MC_DDR3 — fake_mem_ctrl — Fake Memory Controller
- PITONSYS_NO_MC — PITON_FPGA_BRAM_TEST PITON_FPGA_BRAM_BOOT — fake_boot_ctrl — fake_boot_ctrl
- PITON_ARIANE — riscv_peripherals — Connects all peripheral to ariane

Conversions (vldrdy-credit)

- Chip - XBAR: noc2_chip_to_xbar, noc3_chip_to_xbar, noc2_xbar_to_chip, noc3_xbar_to_chip
- Mem - XBAR: noc2_mem_to_xbar, noc3_mem_to_xbar, noc2_xbar_to_mem, noc3_xbar_to_mem
- IOB - XBAR: noc2_iob_to_xbar, noc3_iob_to_xbar, noc2_xbar_to_iob, noc3_xbar_to_iob
- Uart - XBAR: noc2_uart_to_xbar, noc3_uart_to_xbar, noc2_xbar_to_uart, noc3_xbar_to_uart
- Ariane debug - XBAR: noc2_ariane_debug_to_xbar, noc3_ariane_debug_to_xbar, noc2_xbar_to_ariane_debug, noc3_xbar_to_ariane_debug
- Ariane bootrom - XBAR: noc2_ariane_bootrom_to_xbar, noc3_ariane_bootrom_to_xbar, noc2_xbar_to_ariane_bootrom, noc3_xbar_to_ariane_bootrom
- Ariane CLINT - XBAR: noc2_ariane_clint_to_xbar, noc3_ariane_clint_to_xbar, noc2_xbar_to_ariane_clint, noc3_xbar_to_ariane_clint
- Ariane PLIC - XBAR: noc2_ariane_plic_to_xbar, noc3_ariane_plic_to_xbar, noc2_xbar_to_ariane_plic, noc3_xbar_to_ariane_plic

- PITONSYS_IOCTRL
  - PITONSYS_UART — uart_top — uart_top
  - ciop_iob — ciop_iob
  - PITONSYS_SPI PITON_FPGA_SD_BOOT — piton_sd_top — Bridge between NOCs and SD Card
  - PITON_FPGA_ETHERNETLITE — eth_top — Ethernet
- ! PITONSYS_IOCTRL
  - ciop_fake_iob — Fake iobridge
  - fake_uart — Fake UART