Table. Required statistics (usage counts) for power analysis

Components can call *getPower(Cycle_t current, ptype power_type, char *user_parms, int total_cycles)* to get power dissipation of the sub-component, *power_type*, during time interval, *total_cycles*. Power dissipation of a sub-component depends on several statistics (user_parms), and different power models require different sets of statistics. Component writer can pass these statistics in string format to power models.

For example, a core component can get power dissipation of its il1 cache from McPAT by calling,

getPower(current_cycle, CACHE_IL1, "1:1:1:1:1:0", 3);

which tells McPAT that the il1 cache has

number of read hits = 1
number of read misses = 1
number of miss buffer access = 1
number of fill buffer access = 1
number of prefetch buffer access = 1
number of wbb buffer access = 0
during the past 3 cycles.

The getPower function returns power dissipation estimated by McPAT based on these statistics (user_parms).

Sim-Panalyzer	
ptype	**Statistics (user_parms)**
CACHE_IL1	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
CACHE_IL2	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
CACHE_DL1	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
CACHE_DL2	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
CACHE_ITLB	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
CACHE_DTLB	<read(0)/write(1)>:<cache access starting address>:<access latency>:<usage count>
BPRED	<usage count>

RF	<usage count>
ALU	<usage count>
FPU	<usage count>
MULT	<usage count>
LOGIC	<usage count>
IO	<read(0)/write(1)>:<io access starting address>:<access latency>:<usage count>
CLOCK	<usage count>
McPAT	
ptype	**Statistics (user_parms)**
CACHE_IL1	<number of read hits>:<read misses>:<miss buf access>:<fill buf access>:<prefetch buf access>:<wbb buf access>
CACHE_DL1	<number of read hits>:<read misses>:<miss buf access>:<fill buf access>:<prefetch buf access>:<write_access>
CACHE_ITLB	<number of total hits>:<total misses>
CACHE_DTLB	<number of total hits>:<total misses>
RF	<number of int regfile reads>:<int regfile writes>:<float regfile reads>:<float regfile writes>:<function calls>: <phy int regfile reads>:<phy int regfile writes>:<phy float regfile reads>:<phy float regfile writes>
IB	<number of instruction buffer reads>:<instruction buffer writes>
ISSUE_Q	<number of instruction window reads>:<instruction window writes>
INST_DECODER	<number of total instructions>
PIPELINE	none
BYPASS	<number of bypassbus access>:<integer instructions>:<floating point instructions>
LOGIC	<number of total instructions>:<int instructions>:<fp instructions>
ALU	<number of int instructions>
FPU	<number of fp instructions>
EXEU	<number of int instructions>
LSQ	inorder--<number of lsq access>; ooo--<number of load buffer reads>:<load buffer writes>:<store buffer reads>:<store buffer writes>

BPRED	<number of branch instructions>:<branch mispredictions>
RAT	<number of int instructions>:<branch mispredictions>:<branch instructions>:<commited instructions>: <fp instructions>
ROB	<number of ROB reads>:<ROB writes>
BTB	<number of branch instructions>:<branch mispredictions>
CACHE_L2	<number of read accesses>:<write accesses>:<miss buffer accesses>:<fill buffer accesses>: <prefetch buffer reads>:<prefetch buffer writes>:<wbb reads>:<wbb writes>:<L2directory read accesses>: <L2directory write accesses>
MEM_CTRL	<number of memory controller memory reads>:<memory controller memory writes>
ROUTER	<number of total router accesses>
CLOCK	none