NAME

cpc − The Cell-Perf-Counter tool

SYNOPSIS

cpc [options] [workload]

DESCRIPTION

The cpc tool is used for setting up and using the hardware performance counters in the Cell Broadband Engine processor. These counters allow a user to see how many times certain hardware events are occurring, which is useful while analyzing the performance of software running on a Cell system.

CPC has two basic modes of operation. The first is workload mode, which allows the user to specify another program to run while the counters are active. CPC sets up the desired events, starts the counters, then runs the specified program. When the workload is complete, cpc stops the counters, reads and displays the counter values, and resets the performance monitoring unit.

The second mode is free-run mode. In this case, the user simply specifies the events and options, and cpc starts the counters and lets them run indefinitely. The user can then use cpc as necessary to read the counters, stop or zero the counters, reset the PMU, or a variety of other actions.

OPTIONS

The following options can be used with cpc:

-h, --help
Display this summary of options.

-v, --verbose
Verbose output.

-V, --version
Display the version number of this tool.

Specifying Events To Count

-l, --list-events
Display all available events.

-e, --events EVENT,EVENT,...
Comma-separated list of events to count. Events can be specified using their number or name. See **--list-events** for a full listing of available events.

Events are assigned to counters in the order that they are specified. To skip a counter, leave it blank (two consecutive commas). To count an event's negative polarity, prefix the event with a hyphen.

The counters and list of events are reset to zero before loading this list of events, so any previously specified events are deleted.

While most events are counted in cycles or edges, a few events allow the user to decide which way to count. In this case, the count type is specified by appending a ".C" or a ".E" to the event number. The events that allow this choice are shown with the appended ".C" or ".E" in the list displayed with the **--list-events** option. The default is ".C".

To count all clock cycles without regard to events, specify event "C".

There are four 32-bit counters available, and each one can also be configured as two 16-bit counters. The counter pairs are 0/4, 1/5, 2/6, and 3/7. Counters 4-7 (if specified) are always 16-bit. Counters 0-3 will be 16-bit if it's paired counter is used, or 32-bit otherwise. The counters are initialized to 0 unless an initial value is specified with "=VALUE" after the event number.

To specify a particular hardware subunit (e.g. spu or mfc), suffix the event with a colon and the subunit number.

Events are grouped together according to logic units within the Cell processor. The PMU can only count events for up to two groups at once, but can count any number of events within a group (up to the number of counters). Use the **--list-events** option to get a list of all events and which groups they belong to. Counting the same event for two different subunits counts as two different groups.

Example:

--events 2104=501,-IL1_Miss_Cycles_t1.E,,C,4104:2=344

16-bit counter 0 counts event 2104, beginning with an initial value of 501.

32-bit counter 1 counts the IL1_Miss_Cycles_t1 event (number 2123) with negative polarity and an initial value of zero. It counts the number of events that occur (instead of the number of cycles elapsed while the event is occurring).

Counter 2 is unused.

32-bit counter 3 counts all processor cycles.

16-bit counter 4 counts event 4104 for subunit 2, beginning with an initial value of 344.

Due to hardware limitations, not all combinations of events can be counted at the same time. Use the **--query** option to verify whether a particular combination is valid without actually counting anything.

There is 1 additional event source. The following can be be prefixed with "-" or "+" to indicate polarity (default=positive) and suffixed with "C" or "E" to indicate whether to count cycles or edges (default=cycles):

Ext count a signal on the External Trigger

Basic Actions

-E, --enable

Enable the performance monitor and keep it running until it is explicitly disabled with **--disable** or **--reset**. If this option is not specified when running a workload, the counters will be disabled and reset after the workload completes.

-D, --disable
Disable the performance monitor without affecting the counter values or signals.

-r, --read-counters
Read and display the values from the hardware counters that are currently enabled.

-S, --read-sampling-buffer
Read the hardware sampling buffer and display the values for the counters that are currently enabled.

-z, --zero
Zero the counters without resetting the PMU or changing the list of events that are being counted.

-R, --reset
Reset the performance monitor and remove all signals.

-s, --status
Display the performance monitor control registers in addition to the counter values or sampling data.

-q, --query
Query to see if an event combination is valid without performing any actions.

Basic Configuration Options

-m, --mode MODE
Count only when in specified MODE:
a count in all modes
s count in supervisor (kernel) mode
p count in problem (user) mode
h count in hypervisor mode
k alternate for s
u alternate for p
y alternate for h

-f, --freeze-on-overflow
Freeze all counters when any counter overflows.

-w, --wrap-at-max
Wrap counters at max count (default = stop at max).

Output Formatting Options

-H, --html filename
Write output in HTML format to the specified file.

--xml filename
Write output in XML format to the specified file.

-x, --hex-lower
Display counter values in hex (lower-case letters).

-X, --hex-upper
Display counter values in hex (upper-case letters).

-O, --octal
Display counter values in octal.

--hw-signal-names
Display hardware signal names instead of descriptive names.

Hardware Sampling Options

-i, --interval INTERVAL
Sampling INTERVAL (in core clock cycles; minimum=10). The counters are reset at the beginning of each interval. If this option is specified, the sampling data will be stored in the hardware buffer (unless the **--sampling-buffer** option specifies an alternate buffer) and the sampling mode will default to "count" (unless **--sampling-mode** option specifies an alternate mode).

--sampling-mode TYPE
Indicates the TYPE of data stored to the sampling buffer. If "none", the counters contain data for the entire workload. If the "--interval" option is specified, the default is "count".
none Store no data. (default)
count Store counter values.
occurrence Store occurrence data.
threshold Store threshold data.

--sample-8of16
Sample only the lower 8 bits of 16-bit counters. This option forces all counters to be set to 16-bit.

--thermal-data
Collect thermal data in the sampling buffer. This option cannot be used with occurrence, threshold, or 8-of-16-bit sampling. It also cannot be used when using counters two, three, six, or seven. This option can be used without giving a set of events to count.

--overwrite-samples
Overwrite the hardware sampling data when the buffer fulls up. Only the data from the most recent 1024 samples will be available.

Start/Stop Qualifier Options

--start-on-ctr0 n
Start counters when counter 0 has counted "n" events. Counter 4 cannot be used with this option enabled, unless **--stop-on-ctr4** is also used.

--stop-on-ctr4 n
Stop counters when counter 4 has counted "n" events. Counter 0 cannot be used with this option enabled, unless **--stop-on-ctr0** is also used.

--start-on-trigger0
Start counters upon debug bus trigger 0.

--start-on-trigger1
Start counters upon debug bus trigger 1.

--start-on-trigger2
Start counters upon debug bus trigger 2.

--start-on-trigger3
Start counters upon debug bus trigger 3.

--stop-on-trigger0
Stop counters upon debug bus trigger 0.

--stop-on-trigger1
Stop counters upon debug bus trigger 1.

--stop-on-trigger2
Stop counters upon debug bus trigger 2.

--stop-on-trigger3
Stop counters upon debug bus trigger 3.

--start-on-ppu-spr-trigger1
Start counters upon PPU SPR trigger 1.

--stop-on-ppu-spr-trigger2
Stop counters upon PPU SPR trigger 2.

--start-on-event1
Start counters upon debug bus event 1.

--stop-on-event2
Stop counters upon debug bus event 2.

--start-on-ppu0-bookmark
Start counters upon PPU Thread 0 Bookmark start.

--start-on-ppu1-bookmark
Start counters upon PPU Thread 1 Bookmark start.

--stop-on-ppu0-bookmark
Stop counters upon PPU Thread 0 Bookmark stop.

--stop-on-ppu1-bookmark
Stop counters upon PPU Thread 1 Bookmark stop.

--restart-enable
Allows prequalifier start after prequalifier stop.

--single-threaded
Don't use multiple threads for running the ioctl commands in parallel to all the CPUs. Call the ioctls sequentially from the main cpc process. This should only be necessary for debugging the kernel driver.

Workload
Program and arguments to execute while counting events. If no workload is specified then a list of events must be specified (**--events**) and/or one of the "action" options must be specified (**--enable**, **--disable**, **--query**, **--read-counters**, **--read-sampling-buffer**, **--reset**).

HARDWARE SAMPLING

The Cell PMU provides a mechanism for the hardware to periodically read the counters and store the results in a hardware buffer. This allows the cpc tool to collect a large number of counter samples while greatly reducing the number of calls that cpc has to make into the kernel.

Use the **--interval** option to specify the length of the sampling period, in clock cycles. In most cases, the counters should be initialized to zero, since they are all reset to their initial value at the end of each sampling period.

In workload mode, while the workload program is running, cpc will periodically read the contents of the hardware sampling buffer, which prevents the hardware buffer from filling up. CPC then displays information for all accumulated samples after the workload completes.

In free-run mode, since cpc doesn't continue to run, the hardware sampling buffer can fill up. It is limited to 1024 samples. Use the **--read-sampling-buffer** option to have cpc read and display the current contents of the sampling buffer (which also empties the buffer). Also, if the **--overwrite-samples** option is specified when initially loading the events, the PMU will overwrite the oldest sample in the buffer at the end of each sampling interval. This keeps the 1024 most recent samples in the buffer at all times. If this option is not specified, the PMU will stop collecting samples once the hardware sampling buffer fills up, thus reporting the first 1024 samples that occurred since the last **--read-sampling-buffer** command.

In addition to simply sampling the counters, the Cell PMU provides a variety of other sampling modes.

Occurrence sampling (**--sampling-mode occurrence**) monitors up to 64 events at once. Each sample contains one bit for each event, indicating whether the event occurred at least once during that interval. All of the events in a signal-group are monitored, and up to two groups can be monitored at once. Use the **--events** option to specify any event in the group(s) you wish to sample.

Threshold sampling (**--sampling-mode threshold**) monitors the counters, and records one bit for each active counter that indicates whether the counter overflowed (wrapped around to zero) during the interval. For this to be useful, the events should be initialized to the negative of the desired "threshold" value. Because the counters always count up, initializing to a negative value will cause the counter to overflow after the specified number of event occurrences.

8-bit sampling (**--sample-8of16**) collects only the lower 8-bits of the 16-bit counters. In this mode, all counters are forced to be 16-bit, even if only the first four counters are used.

Thermal sampling (**--thermal-data**) collects information about the temperature in various areas of the Cell processor. The PPU and all eight SPUs contain temperature sensors, and the values of these sensors can be routed to the sampling buffer. This option cannot be used with occurrence or threshold sampling,

since the data is stored in the same place in the sampling buffer. This option can be used with regular count sampling, but cannot be used when counters 2, 3, 6, or 7 are enabled. This option can also be used by itself, without loading any other events to count.

EXAMPLES

cpc --events 2104.E,2123.E ls -al

Run the "ls -al" command, and measure events 2104 and 2123 (L1 i-cache misses for hardware threads 0 and 1) while the command is running.

Output:
TODO: Add output for this command.

TODO: Add more examples.

SEE ALSO

oprofile(1)

AUTHOR

Kevin Corry <kevcorry@us.ibm.com>