HiSilicon PCIe Performance Monitoring Unit (PMU)

On Hip09, HiSilicon PCIe Performance Monitoring Unit (PMU) could monitor bandwidth, latency, bus utilization and buffer occupancy data of PCIe.

Each PCIe Core has a PMU to monitor multi Root Ports of this PCIe Core and all Endpoints downstream these Root Ports.

HiSilicon PCIe PMU driver

The PCIe PMU driver registers a perf PMU with the name of its sicl-id and PCIe Core id.:

```
/sys/bus/event_source/hisi_pcie<sicl>_<core>
```

PMU driver provides description of available events and filter options in sysfs, see /sys/bus/event_source/devices/hisi_pcie<sicl>_<core>.

The "format" directory describes all formats of the config (events) and config1 (filter options) fields of the perf_event_attr structure. The "events" directory describes all documented events shown in perf list.

The "identifier" sysfs file allows users to identify the version of the PMU hardware device.

The "bus" sysfs file allows users to get the bus number of Root Ports monitored by PMU.

Example usage of perf:

```
$# perf list
hisi_pcie0_0/rx_mwr_latency/ [kernel PMU event]
hisi_pcie0_0/rx_mwr_cnt/ [kernel PMU event]
------------------------------------------

$# perf stat -e hisi_pcie0_0/rx_mwr_latency/
$# perf stat -e hisi_pcie0_0/rx_mwr_cnt/
$# perf stat -g -e hisi_pcie0_0/rx_mwr_latency/ -e hisi_pcie0_0/rx_mwr_cnt/
```

The current driver does not support sampling. So "perf record" is unsupported. Also attach to a task is unsupported for PCIe PMU.

Filter options

1. Target filter PMU could only monitor the performance of traffic downstream target Root Ports or downstream target Endpoint. PCIe PMU driver support "port" and "bdf" interfaces for users, and these two interfaces aren't supported at the same time.

-port "port" filter can be used in all PCIe PMU events, target Root Port can be selected by configuring the 16-bits-bitmap "port". Multi ports can be selected for AP-layer-events, and only one port can be selected for TL/DL-layer-events.

For example, if target Root Port is 0000:00:00.0 (x8 lanes), bit0 of bitmap should be set, port=0x1; if target Root Port is 0000:00:04.0 (x4 lanes), bit8 is set, port=0x100; if these two Root Ports are both monitored, port=0x101.

Example usage of perf:

```
$# perf stat -e hisi_pcie0_0/rx_mwr_latency,port=0x1/ sleep 5
```

-bdf

"bdf" filter can only be used in bandwidth events, target Endpoint is selected by configuring BDF to "bdf". Counter only counts the bandwidth of message requested by target Endpoint.

For example, "bdf=0x3900" means BDF of target Endpoint is 0000:39:00.0.

Example usage of perf:

```
$# perf stat -e hisi_pcie0_0/rx_mrd_flux,bdf=0x3900/ sleep 5
```

2. Trigger filter Event statistics start when the first time TLP length is greater/smaller than trigger condition. You can set the trigger condition by writing "trig_len", and set the trigger mode by writing "trig_mode". This filter can only be used in bandwidth events.

For example, "trig_len=4" means trigger condition is 2^4 DW, "trig_mode=0" means statistics start when TLP length > trigger condition, "trig_mode=1" means start when TLP length < condition.

Example usage of perf:

```
$# perf stat -e hisi_pcie0_0/rx_mrd_flux,trig_len=0x4,trig_mode=1/ sleep 5
```

3. Threshold filter Counter counts when TLP length within the specified range. You can set the threshold by writing "thr_len", and set the threshold mode by writing "thr_mode". This filter can only be used in bandwidth events.

For example, "thr_len=4" means threshold is 2^4 DW, "thr_mode=0" means counter counts when TLP length >= threshold, and "thr_mode=1" means counts when TLP length < threshold.

Example usage of perf:

```
   $# perf stat -e hisi_pcie0_0/rx_mrd_flux,thr_len=0x4,thr_mode=1/ sleep 5
```