Kernel driver occ-hwmon

Supported chips:

- POWER8
- POWER9

Author: Eddie James <eajames@linux.ibm.com>

Description

This driver supports hardware monitoring for the On-Chip Controller (OCC) embedded on POWER processors. The OCC is a device that collects and aggregates sensor data from the processor and the system. The OCC can provide the raw sensor data as well as perform thermal and power management on the system.

The P8 version of this driver is a client driver of I2C. It may be probed manually if an "ibm,p8-occ-hwmon" compatible device is found under the appropriate I2C bus node in the device-tree.

The P9 version of this driver is a client driver of the FSI-based OCC driver. It will be probed automatically by the FSI-based OCC driver.

Sysfs entries

The following attributes are supported. All attributes are read-only unless specified.

The OCC sensor ID is an integer that represents the unique identifier of the sensor with respect to the OCC. For example, a temperature sensor for the third DIMM slot in the system may have a sensor ID of 7. This mapping is unavailable to the device driver, which must therefore export the sensor ID as-is.

Some entries are only present with certain OCC sensor versions or only on certain OCCs in the system. The version number is not exported to the user but can be inferred.

temp[1-n]_label
    OCC sensor ID.

[with temperature sensor version 1]

    temp[1-n]_input
        Measured temperature of the component in millidegrees Celsius.

[with temperature sensor version >= 2]

    temp[1-n]_type
        The FRU (Field Replaceable Unit) type (represented by an integer) for the component that this sensor measures.
    temp[1-n]_fault
        Temperature sensor fault boolean; 1 to indicate that a fault is present or 0 to indicate that no fault is present.

    [with type == 3 (FRU type is VRM)]

        temp[1-n]_alarm
            VRM temperature alarm boolean; 1 to indicate alarm, 0 to indicate no alarm

    [else]

        temp[1-n]_input
            Measured temperature of the component in millidegrees Celsius.

freq[1-n]_label
    OCC sensor ID.
freq[1-n]_input
    Measured frequency of the component in MHz.
power[1-n]_input
    Latest measured power reading of the component in microwatts.
power[1-n]_average
    Average power of the component in microwatts.
power[1-n]_average_interval
    The amount of time over which the power average was taken in microseconds.

[with power sensor version < 2]

power[1-n]_label
OCC sensor ID.

[with power sensor version >= 2]

power[1-n]_label
OCC sensor ID + function ID + channel in the form of a string, delimited by underscores, i.e. "0_15_1". Both the function ID and channel are integers that further identify the power sensor.

[with power sensor version 0xa0]

power[1-n]_label
OCC sensor ID + sensor type in the form of a string, delimited by an underscore, i.e. "0_system". Sensor type will be one of "system", "proc", "vdd" or "vdn". For this sensor version, OCC sensor ID will be the same for all power sensors.

[present only on "master" OCC; represents the whole system power; only one of this type of power sensor will be present]

power[1-n]_label
"system"
power[1-n]_input
Latest system output power in microwatts.
power[1-n]_cap
Current system power cap in microwatts.
power[1-n]_cap_not_redundant
System power cap in microwatts when there is not redundant power.
power[1-n]_cap_max
Maximum power cap that the OCC can enforce in microwatts.
power[1-n]_cap_min Minimum power cap that the OCC can enforce in
microwatts.
power[1-n]_cap_user The power cap set by the user, in microwatts.
This attribute will return 0 if no user power cap has been set. This attribute is read-write, but writing any precision below watts will be ignored, i.e. requesting a power cap of 500900000 microwatts will result in a power cap request of 500 watts.

[with caps sensor version > 1]

power[1-n]_cap_user_source
Indicates how the user power cap was set. This is an integer that maps to system or firmware components that can set the user power cap.

The following "extn" sensors are exported as a way for the OCC to provide data that doesn't fit anywhere else. The meaning of these sensors is entirely dependent on their data, and cannot be statically defined.

extn[1-n]_label
ASCII ID or OCC sensor ID.
extn[1-n]_flags
This is one byte hexadecimal value. Bit 7 indicates the type of the label attribute; 1 for sensor ID, 0 for ASCII ID. Other bits are reserved.
extn[1-n]_input
6 bytes of hexadecimal data, with a meaning defined by the sensor ID.