Add-on-Card Thermal Interface Spec for Intel Motherboard V3.0

Background

The goal of this document is to define a standard interface for Facebook Intel motherboard V3.0 to poll thermal data from an add-on-card including Mezzanine card.

Without this interface, management controller on baseboard has to use the worst thermal condition of a PCIe card or Mezzanine card to define the system idle fan PWM setting. It often results in a high idle fan PWM setting when a PCIe card or Mezzanine card that has a high thermal design power presents in the system. This increases the idle power consumption of system.

With an interface defined in this specification, management controller has access to the temperature of critical components. Management controller can keep a lower idle fan PWM, and only drive fan PWM to a higher level based on the temperature of critical components.

Card requirement

This requirement applies to following use cases and different requirement levels are listed.

Not required means the implementation of thermal interface on this combination is optional and not a must. Existing platforms, cards and configurations that are already in mass production falls into this category.

Required means the implementation of thermal interface on this combination is mandatory on new platform and configurations, when there is thermal benefit.

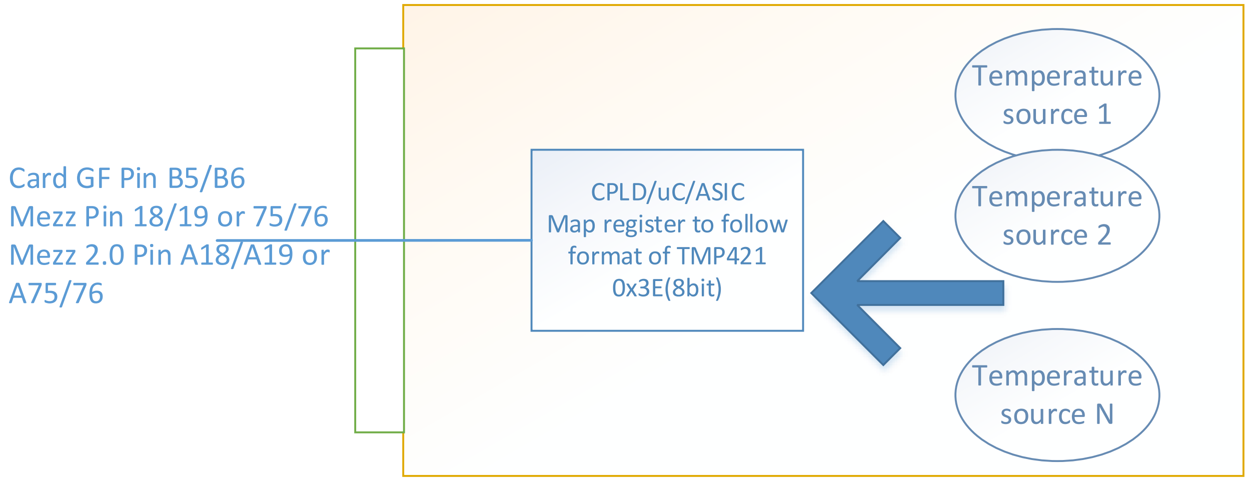
Thermal Profile	To be used with Intel Motherboard V3.0
PCIe card needs < 170LFM(Linear Feet per Minute)@35C	Not required
PCIe card needs ≥ 170LFM@35C	Required
Mezzanine card needs < 170LFM@35C	Not required
Mezzanine card needs ≥ 170LFM@35C	Required

Implementation 1: Emulated method

PCIe card/Mezzanine card should emulate its key temperatures to be accessed from SMBus (Pin shown in the diagram below, and under P3V3_STBY domain). The emulation should follow TMP421 register mapping so that Motherboard will treat the PCIe card thermal sensor as one TMP421 to be used for its thermal control.

There are two temperatures for TMP421 register mapping, Local and Remote Temperature 1. Remote channel is typically used to represent key controller temperature of the card, local channel is typically used to represent highest of other key components temperature on the card, such as highest temperature of all flash modules.

Address of the emulated TMP421 device is fixed at 0x3E in 8bit format.



A register mapping of TMP422/TMP423 can be used to support one/two more temperatures in card implementation. The slave address of emulated device is always 0x3E, even it emulates TMP422/TMP423.

Vendor ID and device ID is mapped to offset 0xFE and 0xFF in order for board management controller to detect card types.

Power reporting and power capping is mapped to offset 0xF2 and 0xF3 as an optional feature to achieve device power monitoring and power capping level setting

The table below describes the register implement requirement for emulated method.

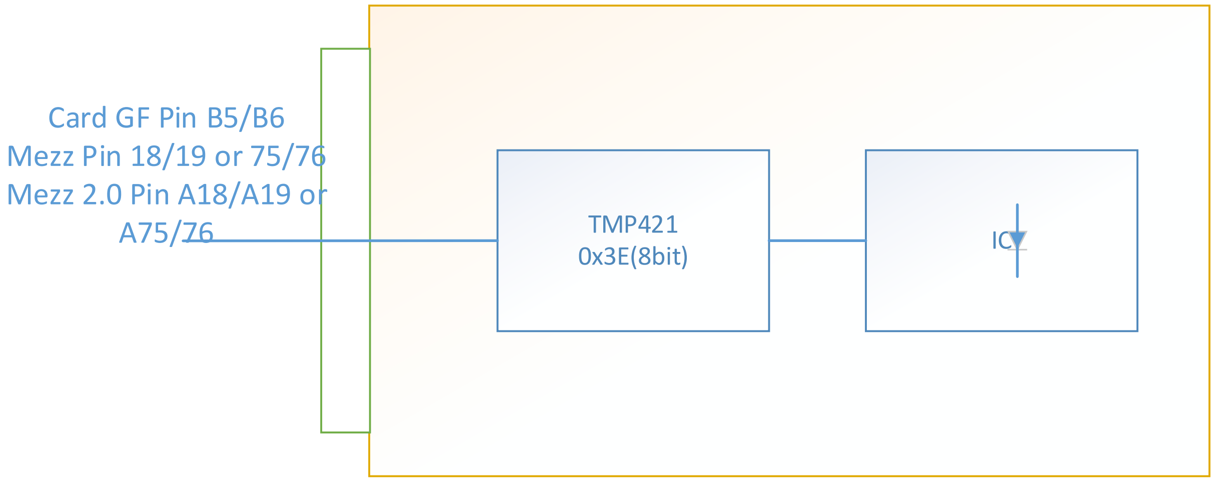
Offset	Description	Original TMP offset	Implementation requirement for emulated method
0x0	Local Temperature (High Byte)	Y	Represents highest temperature of all other key components Required if any of the other key components or modules are critical for thermal design Otherwise it is an optional offset and return 0x00 if not used
0x1	Remote Temperature 1 (High Byte)	Y	Required; represent temperature of main controller
0x2	Remote Temperature 2 (High Byte)	Y	Optional; represent temperature of key component 1; return 0x00 if not used
0x3	Remote Temperature 3 (High Byte)	Y	Optional; represent temperature of key component 2; return 0x00 if not used
0x8	Status Register	Y	Not required
0x9	Configuration Register 1	Y	Not required; Emulated behavior follows SD=0, Temperature Range=0
0x0A	Configuration Register 2	Y	Required; follow TMP423 datasheet to declare the channel supported; RC=1
0x0B	Conversion Rate Register	Y	Not required; Equivalent emulated conversion rate

			should be >2 sample/s
0x0F	One-Shot Start	Y	Not required
0x10	Local Temperature (Low Byte)	Y	Optional; return 0x00 if not used
0x11	Remote Temperature 1 (Low Byte)	Y	Optional; return 0x00 if not used
0x12	Remote Temperature 2 (Low Byte)	Y	Optional; return 0x00 if not used
0x13	Remote Temperature 3 (Low Byte)	Y	Optional; return 0x00 if not used
0x21	N Correction 1	Y	Not required
0x22	N Correction 2	Y	Not required
0x23	N Correction 3	Y	Not required
0xF0	Manufacturer ID(High Byte)	N	High byte of PCIe vendor ID, if using emulated temperature sensor method
0xF1	Device ID(High Byte)	N	High byte of PCIe device ID, if using emulated temperature sensor method
0xF2	Power reporting	N	Optional; card power reporting; 1LSB=1W; Read only
0xF3	Power capping	N	Optional; card power capping; 1LSB=1W; Read/Write
0xFC	Software Reset	Y	Not required
0xFE	Manufacturer ID	Y(redefined)	Low byte of PCIe vendor ID, if using emulated temperature sensor method
0xFF	Device ID	Y(redefined)	Low byte of PCIe device ID, if using emulated temperature sensor method

Implementation 2: Physical Temperature Sensor method

Alternatively, one TMP421 sensor can be used to do on die temperature sensing for IC with thermal diode interface with TMP421 remote sensing channel;

Connection diagram is as below.

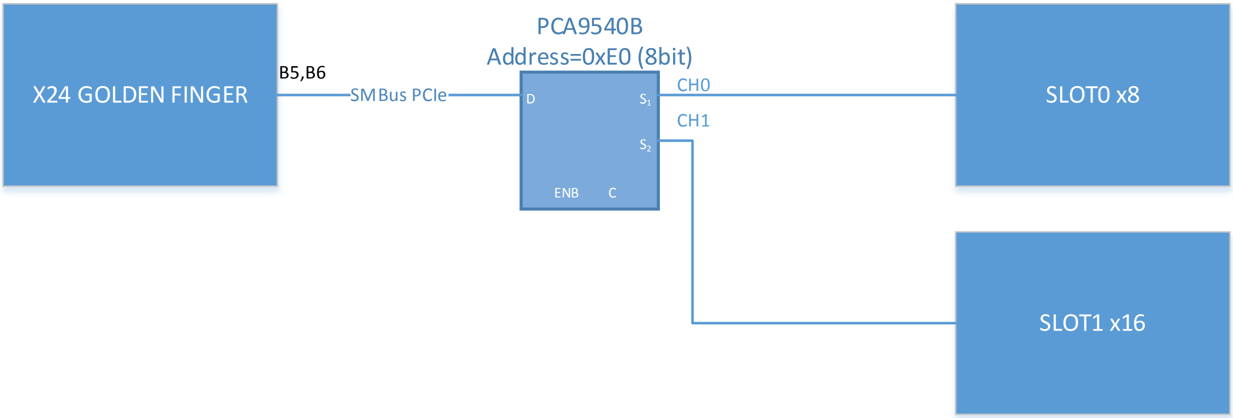

Card GF Pin B5/B6
Mezz Pin 18/19 or 75/76
Mezz 2.0 Pin A18/A19 or A75/76
TMP421
0x3E(8bit)
IC

Riser requirement

Use case 2: Intel Motherboard V3.0 and riser with 1x16 slot and 1x8 slot

The riser card shall implement:

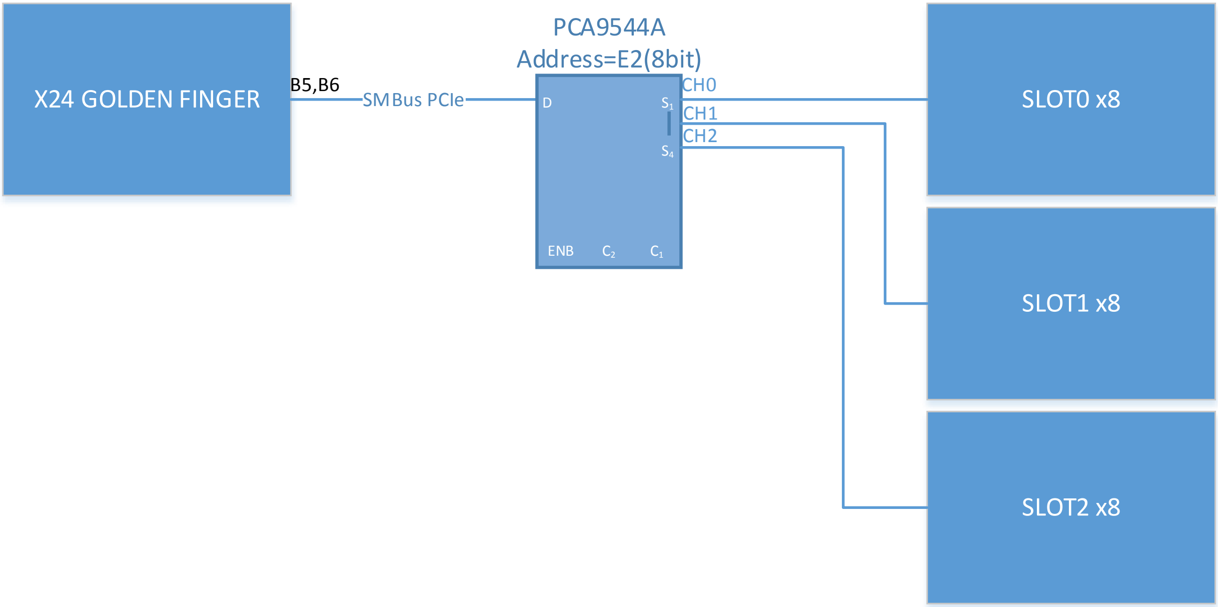
- An I2C MUX on riser for multiplexing
- Slave address used is 0xE0 in 8 bit format for Mux.



Use case 3: Leopard motherboard and riser with 3x8 slots

The riser card shall implement:

- I2C MUX on riser for multiplexing
- Slave address used is 0xE2 in 8 bit format for Mux.


X24 GOLDEN FINGER
B5,B6
SMBus PCIe
PCA9544A
Address=E2(8bit)
D
S_1
S_4
ENB
C_2
C_1
CH0
CH1
CH2
SLOT0 x8
SLOT1 x8
SLOT2 x8