NCT5585D
Programming Guide

Note: The NCT5585D datasheet should be used along with this document.

Date: 11/08/2019 Revision: 0.1

NCT5585D Programming Guide Revision History

Version	Date	Page	Description
0.1	11/08/2019	N.A.	First Release

Please note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this data sheet belong to their respective owners.

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

1. PROGRAMMING THE LEGACY PARTS

1.1 How to enable COMA

1) Pin default function is GPIO8.

2) BIOS have to check the following.

Global Control Register	**Value needed to enable COMA**	**Default value**
CR2A	Bit 7 = 0	Bit 7 = 1
CR2B	Bit 0 = 0	Bit 0 = 0
Logical Device 2	**Value needed to enable COMA**	**Default value**
CR30	0x01	0x01
CR60	0x03	0x03
CR61	0xF8	0xF8
CR70	0x04	0x04

1.2 How to enable KBC

1) Pin default function is KBC.

2) BIOS have to check the following and issue self test command.

Global Control Register	**Value needed to enable KBC**	**Default value**
CR2A	Bit 1:0 = 00	Bit 1:0 = 00
LD12 CREB	Bit 5:2 = 0000	Bit 5:2 = 0000
Logical Device 5	**Value needed to enable KBC**	**Default value**
CR30	0x01	0x00
CR60	0x00	0x00
CR61	0x60	0x00
CR62	0x00	0x00
CR63	0x64	0x00
CR70	0x01	0x00
CR72	0x0C	0x00
CRF0	0x43	0x43

2. PROGRAMMING THE ACPI POWER LOSS FUNCTION

ACPI Power Loss Function Selection	Index	Value
Always off	Logical Device A CRE4 bit 6:5	00b
	Logical Device A CRE7 bit 4	0b
Always on	Logical Device A CRE4 bit 6:5	01b
	Logical Device A CRE7 bit 4	1b
Last status after power on	Logical Device A CRE4 bit 6:5	10b
	Logical Device A CRE7 bit 4	1b
	Logical Device A CRE6 bit 4	0b
Last status before power off	Logical Device A CRE4 bit 6:5	10b
	Logical Device A CRE7 bit 4	0b
	Logical Device A CRE6 bit 4	1b

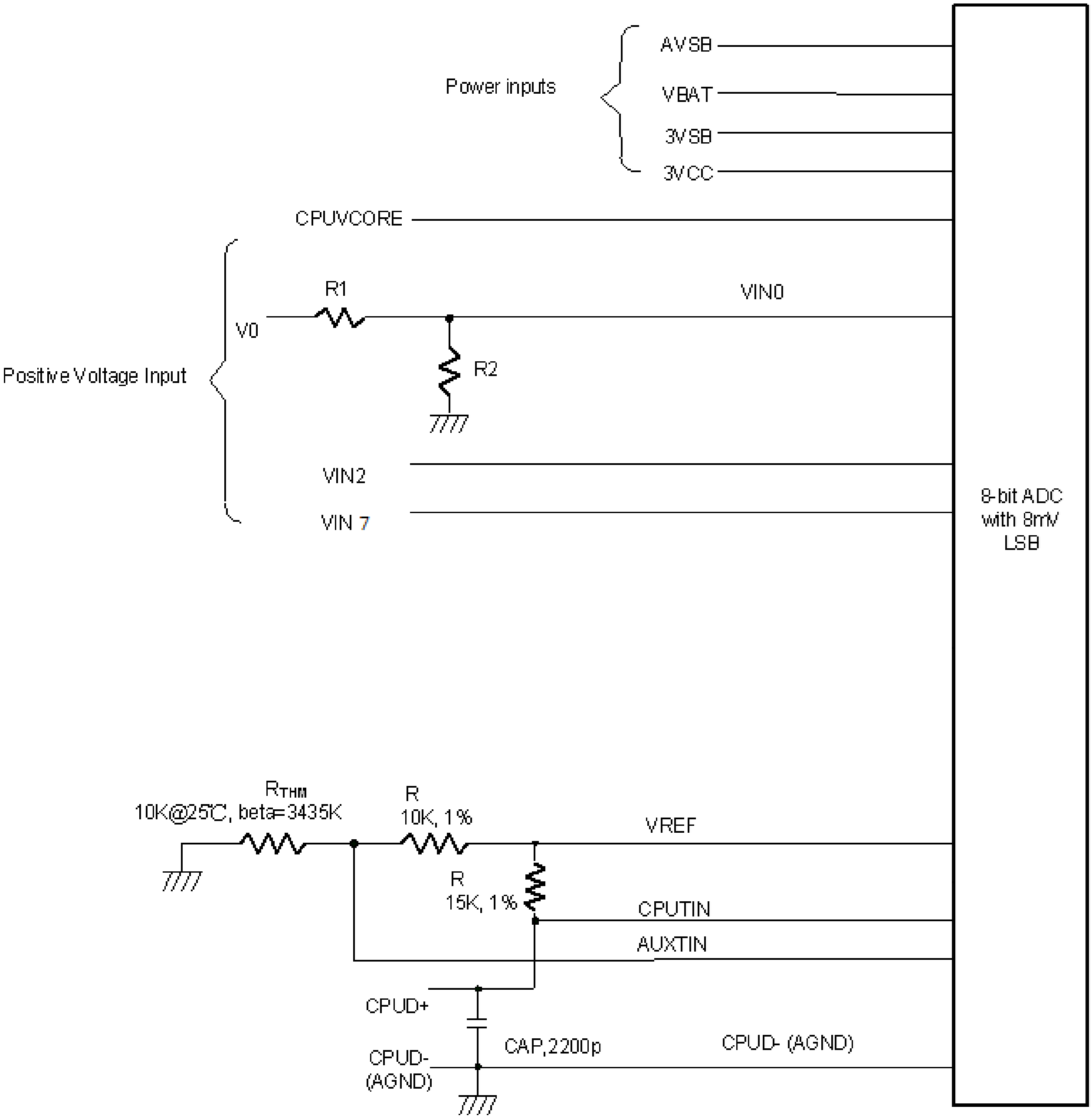
3. PROGRAMMING HARDWARE MONITOR FUNCTIONS

BIOS have to check the following registers first. Take Hardware Monitor IO space as an example.

Logical Device B	**Value needed to enable Hardware Monitor Function**	**Default value**
CR30	0x01	0x00
CR60	0x02	0x00
CR61	0x90	0x00
CR62	0xXX	0x00
CR63	0xXX	0x00
CR64	0x02	0x00
CR65	0x00	0x00
CR70	0x00	0x00

3.1 Reading the input voltage value from voltage channel

The nine analog inputs of the hardware monitor block connect to an 8-bit Analog to Digital Converter (ADC) and consist of 5 general-purpose inputs connected to external device pins (CPUVOCRE, VIN0 – VIN2) and 7 internal signals connected to the power supplies (AVSB, VBAT, 3VSB and 3VCC). All inputs are limited to a maximum voltage of 2.048V due to an internal setting of 8mV LSB (256 steps x 8mV = 2.048V). All inputs to the ADC must limit the maximum voltage by using a voltage divider. The power supplies have internal resistors, while the external pins require outside limiting resistors as described below.



Input voltages greater than 2.048 V should be reduced by an external resistor divider to keep the input voltages in the proper range. For example, input voltage V_0 (+12 V) should be reduced before it is connected to VIN0 according to the following equation:

$$VIN0 = V_0 \times \frac{R_2}{R_1 + R_2}$$

R1 and R2 can be set to 110 KΩ and 10 KΩ, respectively, to reduce V_0 from +12 V to less than 2.048 V.

All the internal inputs of the ADC, AVSB, VBAT, 3VSB and 3VCC utilize an integrated voltage divider with both resistors equal to 34KΩ, yielding a voltage one half of the power supply. Since one would expect a worst-case 10% variation or a 3.63V maximum voltage, the input to the ADC will be 1.815V, well within the maximum range.

$$V_{in} = VCC \times \frac{34K\Omega}{34K\Omega + 34K\Omega} \cong 1.65V$$, where VCC is set to 3.3V

The CPUVCORE pin feeds directly into the ADC with no voltage divider since the nominal voltage on this pin is only 1.2V.

Both of these solutions are illustrated in the figure above.

The data format for voltage detection is an eight-bit value, and each unit represents an interval of 8 mV.

Detected Voltage = Reading * 0.008 V

If the source voltage was reduced by a voltage divider, the detected voltage value must be scaled accordingly.

Take IO space 0x200 for example:

Monitored Voltage Channel	**Register to store the monitored value (RegValue)**	**Monitored Voltage Channel**	**Register to store the monitored value (RegValue)**
CPUVCORE	0x200	3VSB	0x207
VIN1	0x201	VBAT	0x208
AVSB	0x202	VTT	0x209
3VCC	0x203	VIN2	0x20C
VIN0	0x204	VIN7	0x20E

The following are the equations for each voltage channel.

1. CPUVCORE and VIN [0:3]

Measured Voltage = RegValue [0x20X] * 0.008 Volt

2. 3VCC

Measured Voltage = RegValue [0x203] * 0.008 * 2 Volt

3. AVSB

Measured Voltage = RegValue [0x202] * 0.008 * 2 Volt

4. VBAT

Measured Voltage = RegValue [0x208] * 0.008 * 2 Volt

5. 3VSB

Measured Voltage = RegValue [0x207] * 0.008 * 2 Volt

3.2 Reading the temperature value from temperature channel

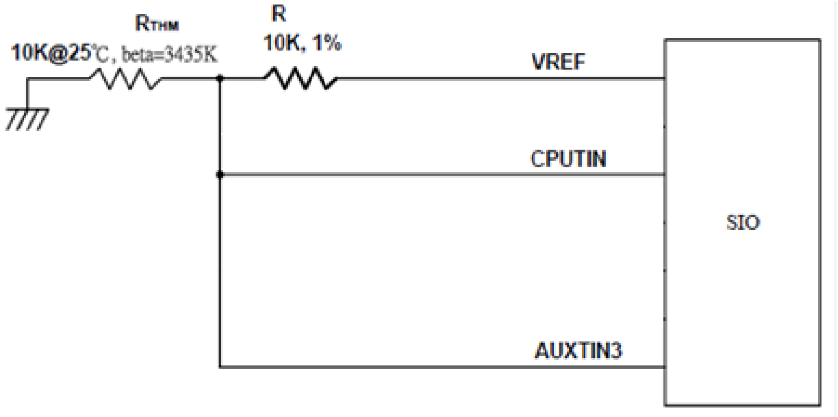
The data format for sensors SYSTIN, CPUTIN and AUXTIN is 9-bit, two's-complement. This is illustrated in the table below. There are two sources of temperature data: external thermistors or thermal diodes.

Temperature Data Format

TEMPERATURE	8-BIT DIGITAL OUTPUT		9-BIT DIGITAL OUTPUT	
	8-BIT BINARY	8-BIT HEX	9-BIT BINARY	9-BIT HEX
+125°C	0111,1101	7Dh	0,1111,1010	0FAh
+25°C	0001,1001	19h	0,0011,0010	032h
+1°C	0000,0001	01h	0,0000,0010	002h
+0.5°C	-	-	0,0000,0001	001h
+0°C	0000,0000	00h	0,0000,0000	000h
-0.5°C	-	-	1,1111,1111	1FFh
-1°C	1111,1111	FFh	1,1111,1110	1FFh
-25°C	1110,0111	E7h	1,1100,1110	1Ceh
-55°C	1100,1001	C9h	1,1001,0010	192h

Monitor Temperature from Thermistor

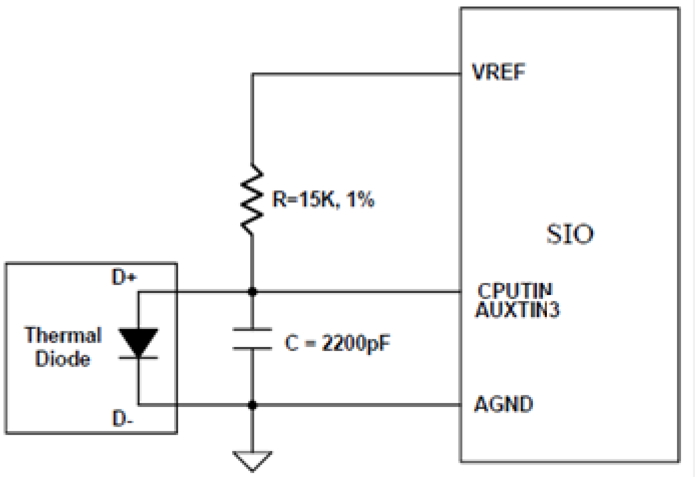
External thermistors should have a β value of 3435K and a resistance of 10 KΩ at 25°C. As illustrated in the schematic above, the thermistor is connected in series with a 10-KΩ resistor and then connects to VREF. The configuration registers to select a thermistor temperature sensor and the measurement method are found at Bank 0, index 5Dh, and 5Eh.



Monitoring Temperature from Thermistor

Monitor Temperature from Thermal Diode (Current Mode)

The NCT5585D can also sense the diode temperature through Current Mode and the circuit is shown in the following figure.



Monitoring Temperature from Thermal Diode (Current Mode)

The pin of processor D- is connected to CPUD- and the pin D+ is connected to temperature sensor pin in the NCT5585D. A bypass capacitor C=2200pF should be added to filter the high frequency noise. The configuration registers to select a thermal diode temperature sensor and the measurement method are found at Bank 0, index 5Dh, and 5Eh.

	CPUTIN mode selection	**AUXTIN3 mode selection**
Thermistor mode	Bank0, Index 5Dh, Bit2 to 0 Bank0, Index 5Eh, Bit2 to 0	Bank0, Index 5Dh, Bit6 to 0 Bank0, Index 5Eh, Bit6 to 0
Current mode	Bank0, Index 5Dh, Bit2 to 1 Bank0, Index 5Eh, Bit2 to 1	Bank0, Index 5Dh, Bit6 to 1 Bank0, Index 5Eh, Bit6 to 1

Note: Please make sure the application circuits are the same as the mode selected

Although NCT5585D provides 2 external temperature channel inputs, NCT5585D could also get temperature from PECI, SB-TSI, SMBus, etc. Therefore, NCT5585D provides 4 registers to store the monitored temperature from different temperature sources (ex. CPUTIN, AUXTIN3, PECI, SB-TSI, SMBus, etc…). The 2 registers to store the monitored temperature are SMIOVT1, SMIOVT2. Please see the table shown below for detail. Temperature source selection is to select the wanted temperature. Temperature reading is the register to store the monitored value of the wanted temperature.

	SMIOVT1	**SMIOVT2**	**SMIOVT3**	**SMIOVT4**
Temperature source select	Bank6,index21 bit[4:0] default: Reserved	Bank6, index22 bit[4:0] default:CPUTIN	BankC,index26 bit[4:0] default: Reserved	BankC,index27 bit[4:0] default: Reserved
Temperature reading	Bank0, index27	Bank1, index50	Bank6, index70	Bank6, index72
	SMIOVT5	**SMIOVT6**	**SMIOVT7**	**SMIOVT8**
Temperature source select	BankC,index28 bit[4:0] default: Reserved	BankC,index29 bit[4:0] default: AUXTIN3	BankC,index2A bit[4:0] default: Reserved	BankC,index2B bit[4:0] default: Reserved
Temperature reading	Bank6, index74	Bank6, index76	Bank6, index78	Bank6, index7A

3.3 Reading the CPU temperature by AMD® SB-TSI

NCT5585D could read only 2 CPU addresses via SB-TSI interface.

The following example is to set the AMD CPU address as 0x98 and get the CPU temperature via SB-TSI.

(SB-TSI base address = 0x200 from Logical Device B CR62 / CR63, SB-TSI_agent0 address = 0x98)

1. Set Global Control Register, CR2C bit 1: 0 = 01b/ LD D, CR E1, bit0 = 0 to select pin functions as SB-TSI.

2. Logical Device B, CR62 and CR63 are the two registers to select the SB-TSI base address <100h: FFEh> along a two-byte boundary. Set CR62 = 0x20 and CR63 = 0x00 for SB-TSI base address in this example.

3. The following table shows the SMBus Register Map. Offset D, or TSI_AGENT, is related to SB-TSI. Set Offset 16 to 0x98 then et 0x20D = 0x01h to enable TSI Agent0 (0x98h).

Offset	Type	Name	Section
0	R/W	**SMBus Data**	19.7.2
1	R/W	**SMBus Write Data Size**	19.7.3
2	R/W	**SMBus Command**	19.7.4
3	R/W	**SMBus Index**	19.7.5
4	R/W	**SMBus Control**	19.7.6
5	R/W	**SMBus Address**	19.7.7
6	R/W	**SMBCLK Frequency**	19.7.8
7	RO	**Reserved**	--
8	R/W	**PCH Address**	19.7.9

9	R/W	**Error status**	19.7.10
A	R/W	**Reserved**	--
B	R/W	**PCH Command**	19.7.11
D	R/W	**TSI Agent Enable**	19.7.12
E	R/W	**SMBus Control 3 Register**	19.7.13
F	R/W	**SMBus Control 3 Register**	19.7.14
10	R/W	**BYTE_ADDR**	19.7.15
11	R/W	**BYTE Index High Byte**	19.7.16
12	R/W	**BYTE Index Low Byte**	19.7.17
13-15	R/W	**Reserved**	
16	R/W	**TSI AGENT0 Address**	19.7.19
17	R/W	**TSI AGENT1 Address**	19.7.20

TSI Agent Enable Register (TSI_AGENT) – Bank0

Bit	7	6	5	4	3	2	1	0
Name							AG1	AG0
Default	0	0	0	0	0	0	0	0
1	**TSI AGENT1 Enable. :** This bit reflects AMD-TSI Agent enable. 0: Disable 1: Enable							
0	**TSI AGENT0 Enable. :** This bit reflects AMD-TSI Agent enable. 0: Disable 1: Enable							

4. Read the value in the Bank 4, Index 09h and Bank4, Index 0Ah. The value of these two registers is the high-byte and low-byte of the SB-TSI Agent 0 temperature, respectively.

PCH_TSI0_TEMP_H Register – Index 09h (Bank 4)

Attribute: Read

Size: 8 bits

BIT	7	6	5	4	3	2	1	0
NAME	**PCH_TSI0_TEMP_H**							
DEFAULT	0	0	0	0	0	0	0	0

BIT	**DESCRIPTION**

BIT	DESCRIPTION
7-0	**PCH_TSI0_TEMP_H** The TSI High-Byte temperature in degree C.

PCH_TSI0_TEMP_L Register – Index 0Ah (Bank 4)

Attribute: Read

Size: 8 bits

BIT	7	6	5	4	3	2	1	0
NAME	PCH_TSI0_TEMP_L			Reserved				
DEFAULT	0	0	0	0	0	0	0	0

BIT	DESCRIPTION
7-5	**PCH_TSI0_TEMP_L** The TSI Low-Byte temperature in degree C.
4-0	**Reserved**

Please be noted that the SMBus address is really 7 bits. The SB-TSI address is normally 98h or 4Ch. The address could vary with address select bits.

SB-TSI Address Encoding

Address Select Bits	SB-TSI Address
000b	98h
001b	9Ah
010b	9Ch
011b	9Eh
100b	90h
101b	92h
110b	94h
111b	96h

3.4 Reading the CPU temperature by Intel® PECI

PECI (Platform Environment Control Interface) is a new digital interface to read the CPU temperature of Intel® CPUs. With a bandwidth ranging from 2 Kbps to 2 Mbps, PECI uses a single wire for self-clocking and data transfer. By interfacing to the Digital Thermal Sensor (DTS) in the Intel® CPU, PECI reports a negative temperature (in counts) relative to the processor's temperature at which the thermal control circuit (TCC) is activated. At the TCC Activation temperature, the Intel CPU will operate at reduced performance to prevent the device from thermal damage.

PECI is one of the temperature sensing methods that the NCT5585D supports. The NCT5585D contains a PECI master and reads the CPU PECI temperature. The CPU is a PECI client.

The PECI temperature values returning from the CPU are in “counts” which are approximately linear in relation to changes in temperature in degrees centigrade. However, this linearity is approximate and cannot be guaranteed over the entire range of PECI temperatures. For further information, refer to the PECI specification. All references to “temperature” in this section are in “counts” instead of “°C”.

NCT5585D could read up to 2 CPU addresses via PECI3.0 interface.

The following example is to read CPU Agent 0 (supporting domain 0 only) temperature via PECI3.0.

1. BIOS have to set Global Control Register CR2C bit 0 = 0 to select Pin function as PECI.

2. Set Bank 7 Index 03h = 10h for Agent 0 and Domain 0

3. Set Bank 7 Index 09h for Agent 0 Tbase

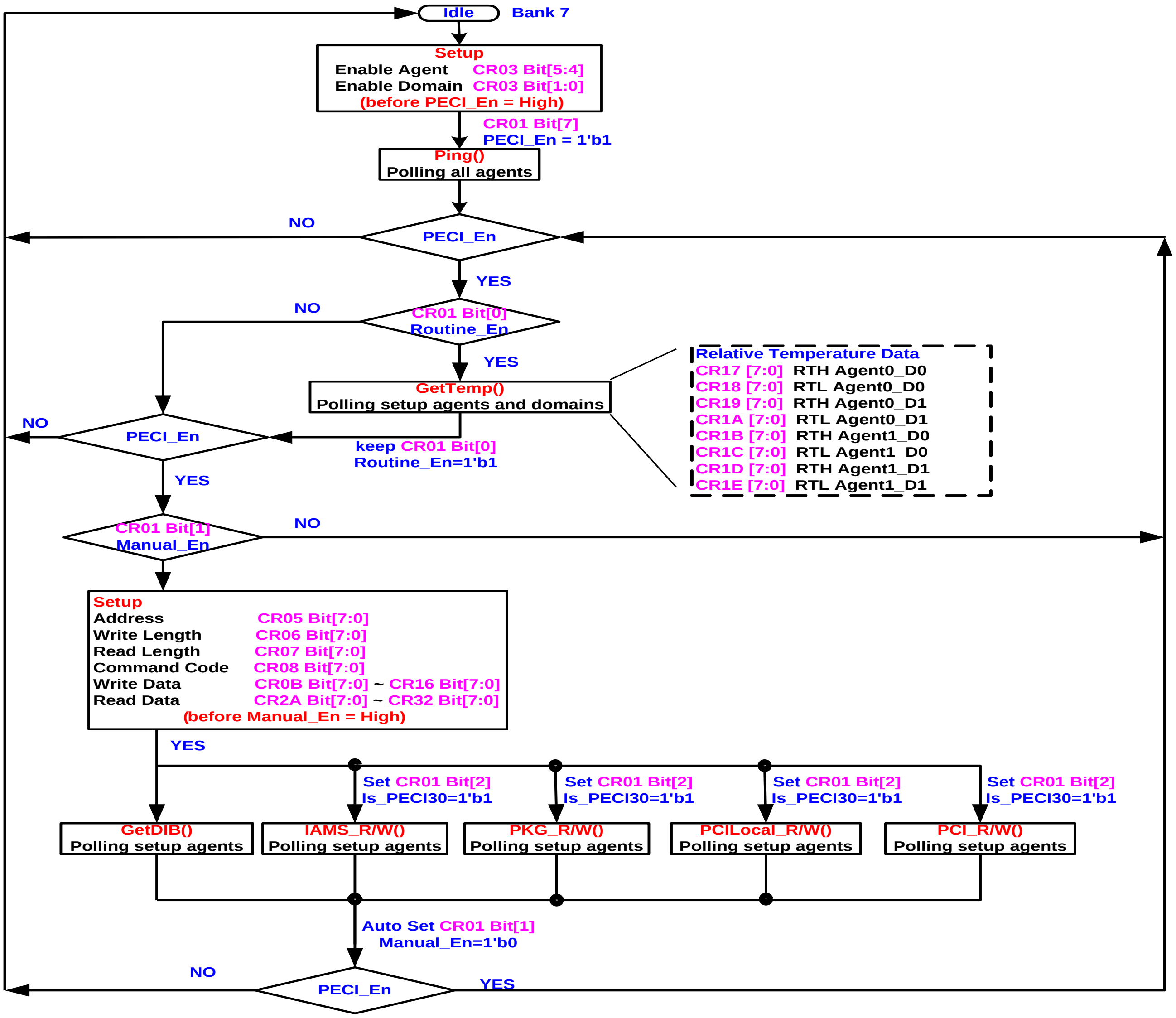
4. Set Bank 7 Index 01h = 95h to enable PECI3.0

5. Get temperature raw data from Bank 7 Index 17h (high byte) / 18h (low byte)

6. Set Bank 2 Index 00h = 10h to select PECI Agent 0 as CPUFAN monitoring source

7. Set Bank 0 Index AEh bit 0 to 1 to enable PECI Agent 0 mode

PECI procedure



PECI Manual Command Address Table

Command Bank 7	Address CR 05$_{HEX}$	WriteLength CR 06$_{HEX}$	Read Length CR 07$_{HEX}$	Command Code CR 08$_{HEX}$
Ping	Addr	00	00	
GetDIB		01	08	F7
GetTemp		01	02	01
PCIRd30		06	02 / 03 / 05	61
PCIWr30		08 / 09 / 0B	01	65
PCIRdLocal30		05	02 / 03 / 05	E1
PCIWrLocal30		07 / 08 / 0A	01	E5
PKGRd30		05	02 / 03 / 05	A1
PKGWr30		07 / 08 / 0A	01	A5
IAMSRRd30		05	02 / 03 / 05 / 09	B1
IAMSRWr30		07 / 08 / 0A / 0E	01	B5

PECI Manual Command Read Data Table

Command	PCI Rd30	PCI Wr30	PCIRd Local30	PCIWr Local30	PKG Rd30	PKG Wr30	IAMSR Rd30	IAMSR Wr30	GetDIB	GetTemp
Command Code	61	65	E1	E5	A1	A5	B1	B5	F7	01
RdData 1 CR 2A$_{HEX}$	Ccode	Ccode	Ccode	Ccode	Ccode	Ccode	Ccode	Ccode	X	X
RdData 2 CR 2B$_{HEX}$	X	X	X	X	X	X	Data LSB_1	X	Device Info	X
RdData 3 CR 2C$_{HEX}$	X	X	X	X	X	X	Data LSB_2	X	Revision Number	X
RdData 4 CR 2D$_{HEX}$	X	X	X	X	X	X	Data LSB_3	X	Reserved 1	X
RdData 5 CR 2E$_{HEX}$	X	X	x	X	X	X	Data LSB_4	X	Reserved 2	X
RdData 6 CR 2F$_{HEX}$	Data LSB_1	X	Data LSB_1	X	Data LSB_1	X	Data LSB_5	X	Reserved 3	X
RdData 7 CR 30$_{HEX}$	Data LSB_2	X	Data LSB_2	X	Data LSB_2	X	Data LSB_6	X	Reserved 4	X
RdData 8 CR 31$_{HEX}$	Data LSB_3	X	Data LSB_3	X	Data LSB_3	X	Data LSB_7	X	Reserved 5	Temp_LB
RdData 9 CR 32$_{HEX}$	Data MSB	x	Data MSB	X	Data MSB	X	Data MSB	X	Reserved 6	Temp_HB

PECI Manual Command Write Data Table

Command	**PCI Rd30**	**PCI Wr30**	**PCIRd Local30**	**PCIWr Local30**	**PKG Rd30**	**PKG Wr30**	**IAMSR Rd30**	**IAMSR Wr30**
Command Code	61	65	E1	E5	A1	A5	B1	B5
WrData 1 CR 0B$_{HEX}$	Host ID	Host ID	Host ID	Host ID	Host ID	Host ID	Host ID	Host ID
WrData 2 CR 0C$_{HEX}$	Addr LSB_1	Addr LSB_1	Addr LSB_1	Addr LSB_1	Index	Index	Processor ID	Processor ID
WrData 3 CR 0D$_{HEX}$	Addr LSB_2	Addr LSB_2	Addr LSB_2	Addr LSB_2	Param LSB	Param LSB	Addr LSB	Addr LSB
WrData 4 CR 0E$_{HEX}$	Addr LSB_3	Addr LSB_3	Addr MSB	Addr MSB	Param MSB	Param MSB	Addr MSB	Addr MSB
WrData 5 CR 0F$_{HEX}$	Addr MSB	Addr MSB	X	Data LSB_1	X	Data LSB_1	X	Data LSB_1
WrData 6 CR 10$_{HEX}$	X	Data LSB_1	X	Data LSB_2	X	Data LSB_2	X	Data LSB_2
WrData 7 CR 11$_{HEX}$	X	Data LSB_2	X	Data LSB_3	X	Data LSB_3	X	Data LSB_3
WrData 8 CR 12$_{HEX}$	X	Data LSB_3	X	Data MSB	X	Data MSB	X	Data LSB_4
WrData 9 CR 13$_{HEX}$	X	Data MSB	X	X	X	X	X	Data LSB_5
WrData10 CR 14$_{HEX}$	X	X	X	X	X	X	X	Data LSB_6
WrData11 CR 15$_{HEX}$	X	X	X	X	X	X	X	Data LSB_7
WrData12 CR 16$_{HEX}$	X	X	X	X	X	X	X	Data MSB

PECI Write-Read Protocol

The write-read protocol is the only protocol defined for messaging between devices on the PECI. The protocol may devolve to either only a write or only a read operation, but the fundamental protocol remains unchanged. The write-read protocol allows an atomic operation that first writes and then reads data between an originator and a target.

2	8	1	8	8
TN	Target Address	MT	Write Length	Read Length

8	8	8	8	8
1st WriteData (Command)	2nd Write Data	………	Nth Write Data	WFCS

8	8	8	8	8
1st Read Data (Command)	2nd Read Data	………	Mth Read Data	RFCS

IAMSR Command Format

2	8	1	8	8
TN	Client Address (0x30)	MT	Write Length (0x05)	Read Length (0x09)

8	8	8	8	8	8
Cmd Code (0xb1)	Host ID[7:1]& Retry[0]	Processor ID	LSB MSR Address MSB		WFCS

8	8	8	8	8	
Completion Code	LSB7	LSB6	LSB5	LSB4	LSB3

8	8	8	8
LSB2	LSB1	MSB	RFCS

Example:

2	8	1	8	8
TN	Client Address **CR 05$_{\text{HEX}}$**	MT	Write Length **CR 06$_{\text{HEX}}$**	Read Length **CR 07$_{\text{HEX}}$**

8	8	8	8	8	8
Cmd Code **CR 08$_{\text{HEX}}$**	Wr Data 1 **CR 0B$_{\text{HEX}}$**	Wr Data 2 **CR 0C$_{\text{HEX}}$**	Wr Data 3 **CR 0D$_{\text{HEX}}$**	Wr Data 4 **CR 0E$_{\text{HEX}}$**	WFCS

8	8	8	8	8	
Rd Data 1 **CR 2A$_{\text{HEX}}$**	Rd Data 2 **CR 2B$_{\text{HEX}}$**	Rd Data 3 **CR 2C$_{\text{HEX}}$**	Rd Data 4 **CR 2D$_{\text{HEX}}$**	Rd Data 5 **CR 2E$_{\text{HEX}}$**	Rd Data 6 **CR 2F$_{\text{HEX}}$**

8	8	8	8
Rd Data 7 **CR 30$_{\text{HEX}}$**	Rd Data 8 **CR 31$_{\text{HEX}}$**	Rd Data 9 **CR 32$_{\text{HEX}}$**	RFCS

3.5 FAN SPEED CONTROL AND MONITOR

3.5.1 Fan Count Calculation

Real RPM (Rotate per Minute) calculations should follow the formula:

$$RPM = \frac{1.35 \times 10^{6}}{Count}$$

In this formula, *13-bit CountValue* represents the values stored in FanCount High byte / Low byte., please read high byte first then low byte.

The registers are in NCT5585D Bank4.

	FAN COUNT READING		**FAN RPM READING**	
	13-bit		**16-bit**	
	[12:5]	**[4:0]**	**[15:8]**	**[7:0]**
SYSFANIN	Bank4, indexB0	Bank4, indexB1	Bank4, indexC0	Bank4, indexC1
CPUFANIN	Bank4, indexB2	Bank4, indexB3	Bank4, indexC2	Bank4, indexC3
AUXFANIN3	Bank4, indexBA	Bank4, indexBB	Bank4, indexCA	Bank4, indexCB
AUXFANIN4	Bank4, indexCC	Bank4, indexCD	Bank4, indexCE	Bank4, indexCF

In 16-bit fan RPM reading, please read high byte first then low byte.

Fan speed RPM can be evaluated by translating 16-bit RPM reading from hexadecimal to decimal.

Register reading 0x09C4h = 2500 RPM

3.5.2 Fan Speed Control Output Method

The NCT5585D provides two methods for fan speed control: the PWM duty cycle output and the DC voltage output. Either the PWM or DC output can be programmed.

The PWM or DC output can be programmed by register Bank0 index 04h, bits 0 for SYSFANOUT only. Others out support PWM only.

	SYSFANOUT	**CPUFANOUT**	**AUXFANOUT3**	**AUXFANOUT4**
Output Type Select	Bank0, index04 bit0 **0: PWM output** **1: DC output (default)**	**Only PWM output**	**Only PWM output**	**Only PWM output**
Output Type Select (in PWM output)	CR24 bit4 0: open-drain (default) 1: push-pull	CR24 bit3 0: open-drain (default) 1: push-pull	CR24 bit1 0: open-drain (default) 1: push-pull	CR25 bit0 0: open-drain (default) 1: push-pull
PWM Output Freqency	Bank0, Index00	Bank0, Index02	Bank0, Index16	Bank0, Index28
Fan Control Mode Select	Bank1, index02, bit[7:4] 0h: Manual mode (def.)	Bank2, index02, bit[7:4] 0h: Manual mode(def.)	BankA, index02, bit[7:4] 0h: Manual mode (def.)	BankB, index02, bit[7:4] 0h: Manual

		1h: Thermal Cruise 2h: Speed Cruise 4h: SMART FAN IV	1h: Thermal Cruise 2h: Speed Cruise 4h: SMART FAN IV	1h: Thermal Cruise 2h: Speed Cruise 4h: SMART FAN IV	mode (def.) 1h: Thermal Cruise 2h: Speed Cruise 4h: SMART FAN IV
Output Value (write)	**PWM output (Duty)**	Bank1, index09 bit[7:0]	Bank2, index09 bit[7:0]	BankA, index09 bit[7:0]	Bank Ｂ index09 bit[7:0]
	DC output (Voltage)	Bank1, index09 bit[7:2]			
Current Output Value (read only)		Bank0, index01	Bank0, index03	Bank0, index17	Bank0, Index29

3.5.2.1. DC Voltage output

For DC, the NCT5585D has a six bit digital-to-analog converter (DAC) that produces 0 to 2.048 Volts DC. The analog output is programmed at Bank1 Index 09h bit [7:2] for SYSFANOUT. The analog output can be calculated using the following equation:

$$\text{OUTPUT Voltage (V)} = Vref \times \frac{\text{Programmed 6 - bit Register Value、}}{64}$$

The default value is 111111YY, or nearly 2.048 V, and Y is a reserved bit for SYSFANOUT only

DC OUTPUT (VOLTAGE)	BANK1, INDEX09 BIT[7:2]

3.5.3 Fan Speed Control Algorithm

The NCT5585D provides output modes for fan control – “Manual Mode”, “Thermal Cruise Mode”, “Speed Cruise Mode” and “SMART FAN IV Mode”.

3.5.3.1. Manual Mode

The programming sequence is shown below. (Take SYSFANOUT as an example)

（1）Bank1 Index02 bit [7:4] = 0x0h (Manual mode)

（2）Bank1 index09 bit [7:0] = 0xB4h (duty cycle =180)

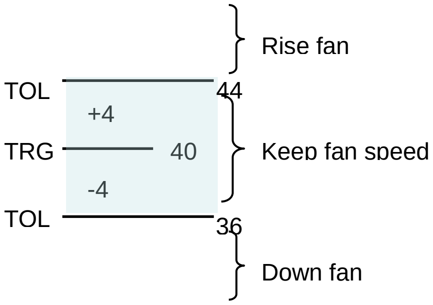
3.5.3.2. Thermal Cruise Mode

The relative registers for Thermal Cruise Mode are listed in the following table.

THERMAL CRUISE MODE	TARGET TEMPERATURE	TOLERANCE	START-UP VALUE	STOP VALUE	KEEP MIN. FAN OUTPUT VALUE	STOP TIME	STEP- UP TIME	STEP-DOWN TIME
SYSFANOUT	Bank 1, index 01h bit[7:0]	Bank 1, index 02h Bit[2:0]	Bank 1, index 06h	Bank 1, index 05h	Bank 1, Index 00h, bit7	Bank 1, index 07h	Bank 1, index 03h	Bank 1, index 04h
CPUFANOUT	Bank 2, index 01h bit[7:0]	Bank 2, index 02h Bit[2:0]	Bank 2, index 06h	Bank 2, index 05h	Bank 2, Index 00h, bit7	Bank 2, index 07h	Bank 2, index 03h	Bank 2, index 04h
AUXFANOUT3	Bank A, index 01h bit[7:0]	Bank A, index 02h Bit[2:0]	Bank A, index 06h	Bank A, index 05h	Bank A, Index 00h, bit7	Bank A, index 07h	Bank A, index 03h	Bank A, index 04h
AUXFANOUT4	Bank B, index 01h bit[7:0]	Bank B, index 02h Bit[2:0]	Bank B, index 06h	Bank B, index 05h	Bank B, Index 00h, bit7	Bank B, index 07h	Bank B, index 03h	Bank B, index 04h

THERMAL CRUISE MODE	CRITICAL TEMPERATURE	ENABLE THERMAL CRUISE MODE
SYSFANOUT	Bank 1, index 35h	Bank 1, Index 02h, bit[7:4] = 01h
CPUFANOUT	Bank 2, Index 35h	Bank 2, Index 02h, bit[7:4] = 01h
AUXFANOUT3	Bank A, Index 35h	Bank A, Index 02h, bit[7:4] = 01h
AUXFANOUT4	Bank B, Index 35h	Bank B, Index 02h, bit[7:4] = 01h

Take SYSFANOUT as an example.



（1）Bank1 index00 bit [7:0] = 0x81h

(If Stopduty_En =1, when the temperature is lower than the target temperature, fan output value is kept at stop value setting in index05 bit [7:0]. If Stopduty_En =0, fan output value will decrease to stop value setting in index05 bit [7:0] first and then wait for index06 bit [7:0] count completed, then fan output value becomes 00h.)

（2）Bank1 index01 bit [7:0] = 0x28h

(SYSFAN Target Temperature = 40, when the temperature exceeds the target temperature, fan output is full speed.)

（3）Bank1 index03 bit [7:0] = 0x01h

Bank1 index04 bit[7:0] = 0x01h

(Step Up Time and Step Down Time = 0.1s)

（4）Bank1 index05 bit [7:0] = 0x80h

(When the temperature is lower than the target temperature, fan output value will decrease to stop value.)

Bank1 index06 bit [7:0] = 0xC8h

(When the fan output becomes 00h, provide a minimum value to turn on the fan output value)

Bank1 index07 bit [7:0] = 0x02h

(Unit 0.1 sec, Step Down Time = 0.2s)

（5）Bank1 index02 bit [7:0] = 0x14h

(Mode Select Thermal Cruise Mode, Tolerance = 4)

3.5.3.3. Speed Cruise Mode

The relative registers for Speed Cruise Mode are listed in the following table.

SPEED CRUISE MODE	TARGET-SPEED COUNT_L	TARGET-SPEED COUNT_H	TOLERANCE_L	TOLERANCE2_H	STEP-UP TIME	STEP-DOWN TIME	STEP-UP VALUE	STEP-DOWN VALUE	ENABLE SPEED CRUISE MODE
SYSFANOUT	Bank 1, Index 01h	Bank 1, Index 0C bit[3:0]	Bank 1, Index 02 bit[2:0]	Bank 1, Index 0C bit[6:4]	Bank 1, Index 03h	Bank 1, Index 04h	Bank1, Index66 Bit[7:4]	Bank1, Index66 Bit[3:0]	Bank 1, Index 02h bit[7:4] = 02h

SPEED CRUISE MODE	TARGET-SPEED COUNT_L	TARGET-SPEED COUNT_H	TOLERANCE_L	TOLERANCE2_H	STEP-UP TIME	STEP-DOWN TIME	STEP-UP VALUE	STEP-DOWN VALUE	ENABLE SPEED CRUISE MODE
CPUFANOUT	Bank 2, Index 01h	Bank 2, Index 0C bit[3:0]	Bank 2, Index 02 bit[2:0]	Bank 2, Index 0C bit[6:4]	Bank 2, Index 03h	Bank 2, Index 04h	Bank2, Index66 Bit[7:4]	Bank2, Index66 Bit[3:0]	Bank 2, Index 02h bit[7:4] = 02h
AUXFANOUT3	Bank A, Index 01h	Bank A, Index 0C bit[3:0]	Bank A, Index 02 bit[2:0]	Bank A, Index 0C bit[6:4]	Bank A, Index 03h	Bank A, Index 04h	BankA, Index66 Bit[7:4]	BankA, Index66 Bit[3:0]	Bank A, Index 02h bit[7:4] = 02h
AUXFANOUT4	Bank B, Index 01h	Bank B, Index 0C bit[3:0]	Bank B, Index 02 bit[2:0]	Bank B, Index 0C bit[6:4]	Bank B, Index 03h	Bank B, Index 04h	BankB, Index66 Bit[7:4]	BankB, Index66 Bit[3:0]	Bank B, Index 02h bit[7:4] = 02h

Take SYSFANOUT as an example.

（1）Bank1 index01 bit [7:0] = 0x50h

Bank1 index0C bit [3:0] = 0x07h

(*12-bit* Target FanCount = 750h)

（2）Bank1 index03 bit [7:0] = 0x01h

Bank1 index04 bit [7:0] = 0x01h

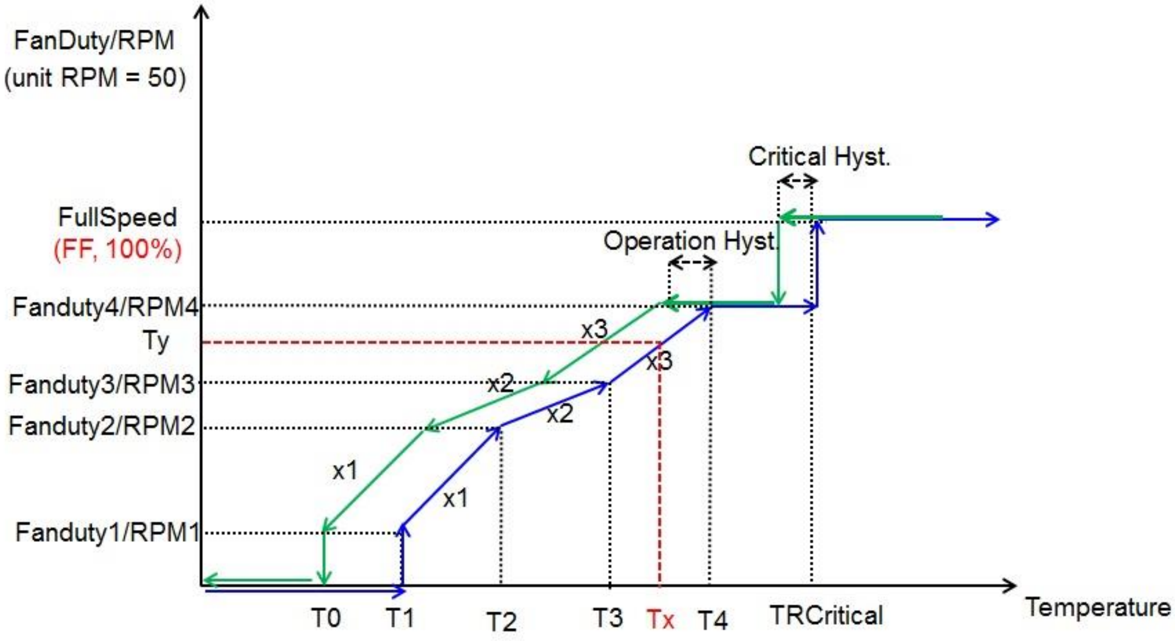
(Step Up Time and Step Down Time = 0.1s)

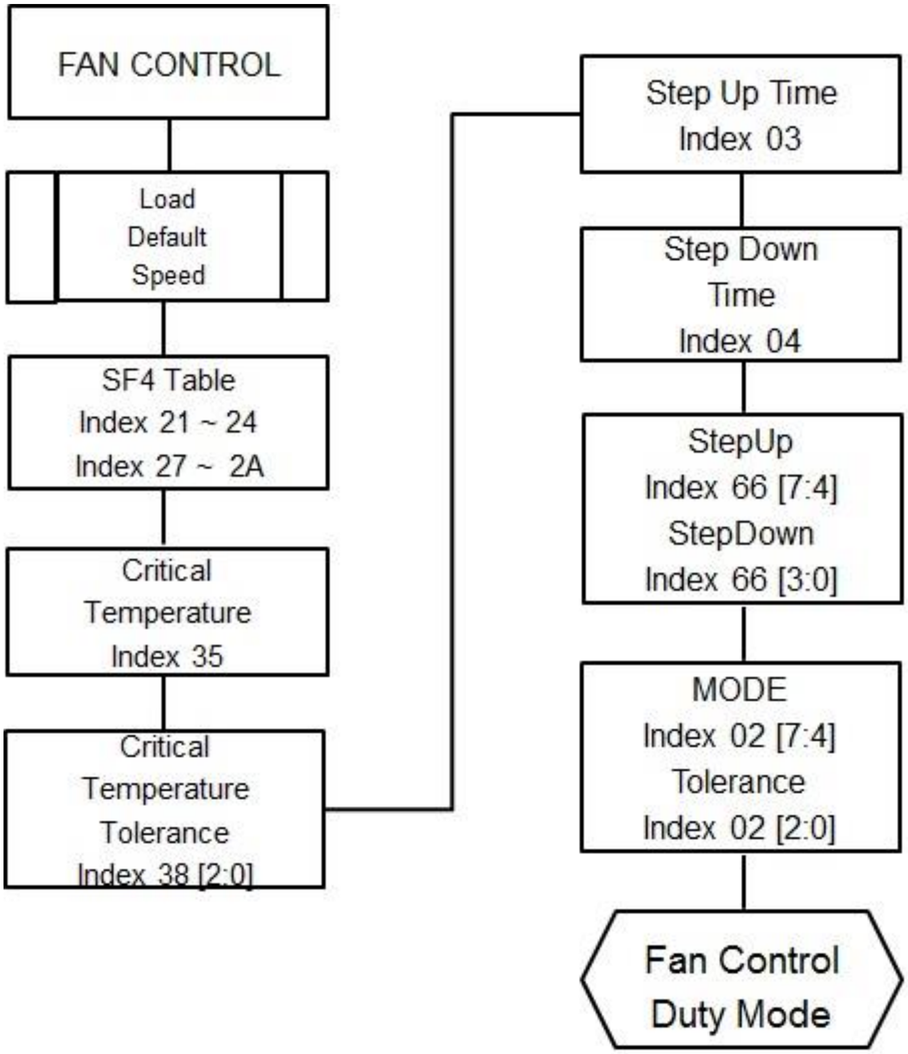
（3）Bank1 index02 bit [7:0] = 0x20h

Bank1 index0C bit [6:4] = 0x4h

(Mode Select Speed Cruise Mode, 6-bit Tolerance programming by index0C bit [6:4] and index02 bit [2:0], Tolerance = 6’b010000 = 20h)

3.5.3.4. Smart Fan IV Mode (Y-axis: FanDuty)




FAN CONTROL
Load Default Speed
SF4 Table Index 21 ~ 24 Index 27 ~ 2A
Critical Temperature Index 35
Critical Temperature Tolerance Index 38 [2:0]
Step Up Time Index 03
Step Down Time Index 04
StepUp Index 66 [7:4] StepDown Index 66 [3:0]
MODE Index 02 [7:4] Tolerance Index 02 [2:0]
Fan Control Duty Mode

Programming Flow

（1）The programming sequence is shown below. (Take CPUFANOUT as an example)

A. Set the temperature: T1 ~ T4

Bank2, Index21 bit [7:0] = 0x0Ah, T1=10℃

Bank2, Index22 bit [7:0] = 0x32h, T2=50℃

Bank2, Index23 bit [7:0] = 0x3Ch, T3=60℃

Bank2, Index24 bit [7:0] = 0x46h, T4=70℃

B. Set the critical temperature: Critical

Bank2, Index35 bit [7:0] = 0x50h, TR=80℃

C. Set DC/PWM Level: Y1 ~ Y4

Bank2, Index27 bit [7:0] = 0x10h, Y1=16

Bank2, Index28 bit [7:0] = 0x50h, Y2=80

Bank2, Index29 bit [7:0] = 0x60h, Y3=96

Bank2, Index2A bit [7:0] = 0xF0h, Y4=240

D. Set Step Up Time and Step Down Time

Bank2 index03 bit [7:0] = 0x01h

Bank2 index04 bit [7:0] = 0x01h

(Step Up Time and Step Down Time = 0.05s)

E. Set Step Up Value and Step Down Value

Bank2 index66 bit [7:4] = 0x01h

Bank2 index66 bit [3:0] = 0x01h

(Step Up Value and Step Down Value = 1)

F. Set Mode Select, set Tolerance

Bank2 index02 bit [7:0] = 0x43h

(Mode Select Smart Fan IV Mode, Tolerance = 3)

G. Set temperature, and check the corresponding FanDuty

When Temperature = 50℃

Bank2 index09 bit [7:0] = 0x50h

When Temperature = 60℃

Bank2 index09 bit [7:0] = 0x60h

When Temperature = 62℃

Bank2 index09 bit [7:0] = 0x7Ch

H. Set the temperature, and check the corresponding FanDuty (decrease temperature, check Tolerance active)

When Temperature = 61℃

Bank2 index09 bit [7:0] = 0x7Ch

When Temperature = 60℃

Bank2 index09 bit [7:0] = 0x7Ch

When Temperature = 59℃

Bank2 index09 bit [7:0] = 0x7Ch

(Because Tolerance = 3, Fan Output Value will keep)

When Temperature = 58℃

Bank2 index09 bit [7:0] = 0x6Eh

When Temperature = 57℃

Bank2 index09 bit [7:0] = 0x5F

I. Set temperature, and check the corresponding FanDuty

When Temperature = 80℃

Bank1 index09 bit [7:0] = 0xFF

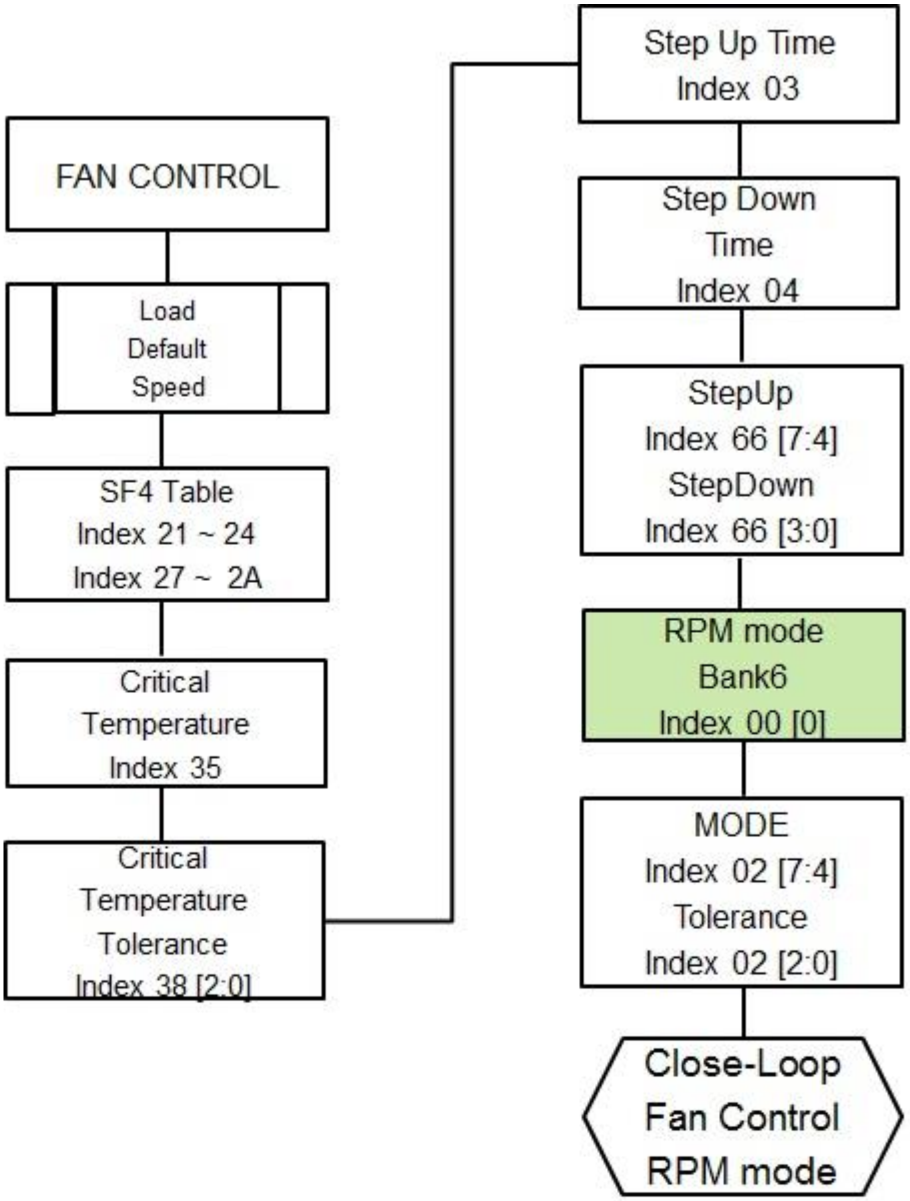
（2）Relative Register-at SMART FAN IV Control Duty Mode

DESCRIPTION	T1	T2	T3	T4
CPUFANOUT	Bank 2, index 21h	Bank 2, index 22h	Bank 2, index 23h	Bank 2, index 24h

DESCRIPTION	DC/PWM1	DC/PWM2	DC/PWM3	DC/PWM4
CPUFANOUT	Bank 2, index 27h	Bank 2, index 28h	Bank 2, index 29h	Bank 2, index 2Ah
DESCRIPTION	**CRITICAL TEMPERATURE**	**Critical Tolerance**	**Monitored temperature tolerance**	**ENABLE SMART FAN IV**
CPUFANOUT	Bank 2, index 35h	Bank 2, index 38h bit[2:0]	Bank 2, index 02h bit[2:0]	Bank 2, index 02h bit[7:4]
DESCRIPTION	**STEP UP TIME**	**STEP DOWN TIME**	**STEP UP VALUE**	**STEP DOWN VALUE**
CPUFANOUT	Bank 2, index 03h	Bank 2, index 04h	Bank 2, Index 66h bit [7:4]	Bank 2, index 66h bit[3:0]
DESCRIPTION	**FAN1_TR1_DELAY_REG**			
CPUFANOUT	Bank 2, index 67h			

Note: The duration (D) that the temperature keeps at T0 more than FANX_TR1_DELAY_REG, that is, D > FANX_TR1_DELAY_REG, FanDuty will drop to 0.

3.5.3.5. Smart Fan IV Mode (Y-axis: RPM)


FAN CONTROL
Load Default Speed
SF4 Table Index 21 ~ 24 Index 27 ~ 2A
Critical Temperature Index 35
Critical Temperature Tolerance Index 38 [2:0]
Step Up Time Index 03
Step Down Time Index 04
StepUp Index 66 [7:4] StepDown Index 66 [3:0]
RPM mode Bank6 Index 00 [0]
MODE Index 02 [7:4] Tolerance Index 02 [2:0]
Close-Loop Fan Control RPM mode

Programming Flow

(1). The programming sequence is shown below. (Take SYSFANOUT as an example)

A. Set the temperature: T1 ~ T4

Bank1, Index21 bit [7:0] = 0x0Ah, T1=10℃

Bank1, Index22 bit [7:0] = 0x14h, T2=20℃

Bank1, Index23 bit [7:0] = 0x1Eh, T3=30℃

Bank1, Index24 bit [7:0] = 0x28h, T4=40℃

B. Set the critical temperature

Bank1, Index35 bit [7:0] = 0x50h, TR=80℃

C. Set DC/PWM Level: Y1 ~ Y4, unit 50 RPM

Bank1, Index27 bit [7:0] = 0x28h, Y1=40 → 40*50=2000 RPM

Bank1, Index28 bit [7:0] = 0x3Ch, Y2=60 → 60*50=3000 RPM

Bank1, Index29 bit [7:0] = 0x46h, Y3=70 → 70*50=3500 RPM

Bank1, Index2A bit [7:0] = 0x55h, Y4=85 → 85*50=4250 RPM

D. Set Step Up Time and Step Down Time

Bank1 index03 bit [7:0] = 0x01h

Bank1 index04 bit [7:0] = 0x01h

(Step Up Time and Step Down Time = 0.05s)

E. Set Step Up Value and Step Down Value

Bank1 index 66 bit [7:4] = 0x01h

Bank1 index 66 bit [3:0] = 0x01h

(Step Up Value and Step Down Value)

F. Set RPM mode

Bank6 index00 bit [0] = 1'b1

(Change Y-axis to RPM)

G. Set Mode Select and Tolerance

Bank1 index02 bit [7:0] = 0x40h

(Mode Select Smart Fan IV Mode, Tolerance = 0)

H. Set the temperature, and check the corresponding FanDuty (Bank0 index13)

When Temperature = 10℃

Bank0 index13 bit [7:0] = 0x28h

When Temperature = 20℃

Bank0 index13 bit [7:0] = 0x3Ch

When Temperature = 30℃

Bank0 index13 bit [7:0] = 0x46h

When Temperature = 40℃

Bank0 index13 bit [7:0] = 0x55h

(Bank0 index13 is test register for check Y-axis target value)

I. Set tolerance of RPM mode

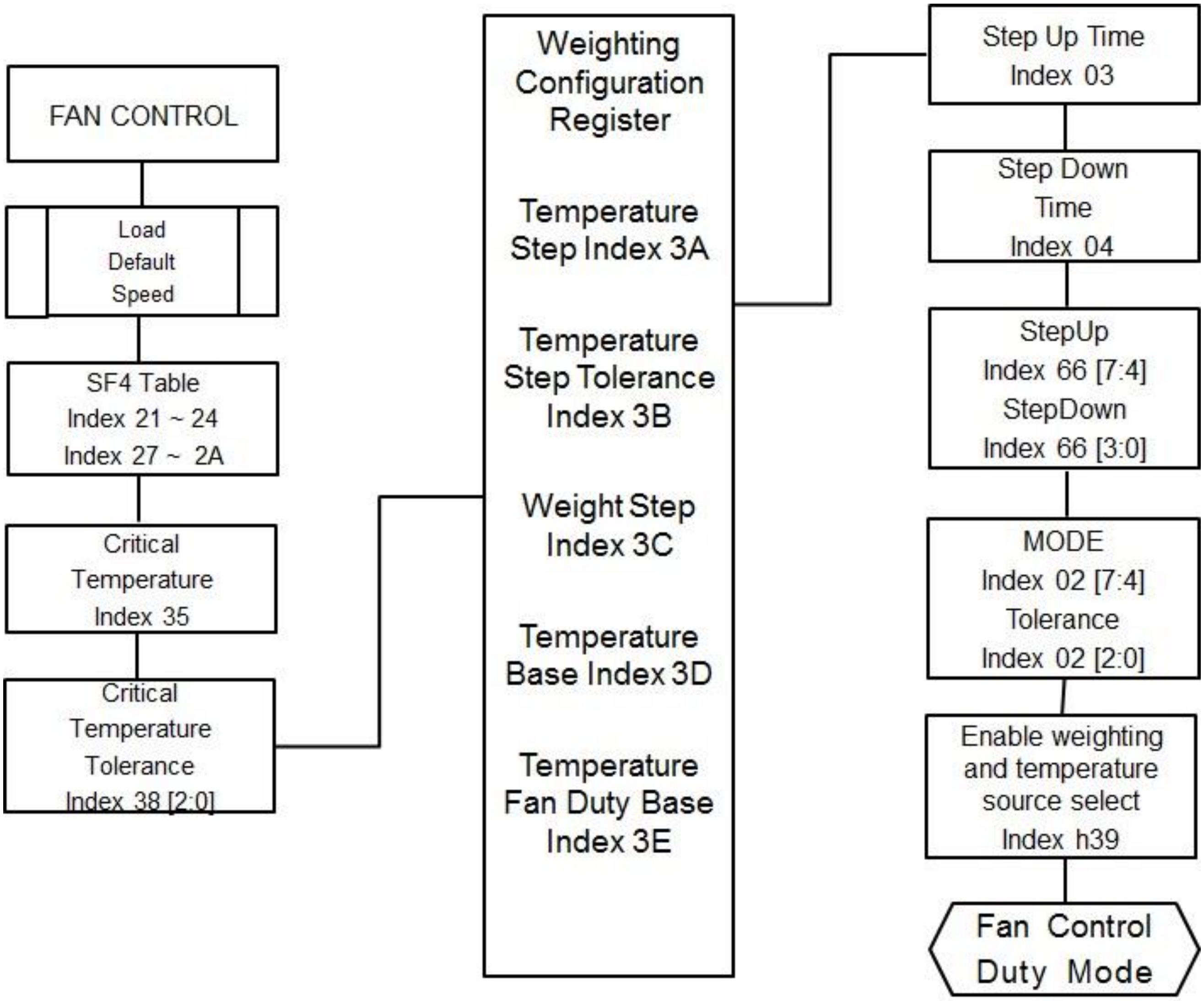
Bank6 index01 bit [3:0] = 0x6h

(Tolerance of RPM mode, unit is 50 RPM.)

（2）Relative Register-at SMART FAN IV Control RPM Mode

DESCRIPTION	T1	T2	T3	T4
SYSFANOUT	Bank 1, index 21h	Bank 1, index 22h	Bank 1, index 23h	Bank 1, index 24h
DESCRIPTION	**DC/PWM1**	**DC/PWM2**	**DC/PWM3**	**DC/PWM4**
SYSFANOUT	Bank 1, index 27h	Bank 1, index 28h	Bank 1, index 29h	Bank 1, index 2Ah
DESCRIPTION	**CRITICAL TEMPERATURE**	**Critical Tolerance**	**Monitored temperature tolerance**	**ENABLE SMART FAN IV**
SYSFANOUT	Bank 1, index 35h	Bank 1, index 38h bit[2:0]	Bank1, index 02h bit[2:0]	Bank 1, index 02h bit[7:4]
DESCRIPTION	**STEP UP TIME**	**STEP DOWN TIME**	**STEP UP VALUE**	**STEP DOWN VALUE**
SYSFANOUT	Bank 1, index 03h	Bank 1, index 04h	Bank1, Index 66h bit [7:4]	Bank 1, index 66h bit[3:0]
DESCRIPTION	**RPM Mode**			
SYSFANOUT	Bank 6, Index h00, bit [0]			

3.5.3.6. Weighting for Smart Fan IV Mode (Y-axis: FanDuty)


FAN CONTROL
Load Default Speed
SF4 Table Index 21 ~ 24 Index 27 ~ 2A
Critical Temperature Index 35
Critical Temperature Tolerance Index 38 [2:0]
Weighting Configuration Register
Temperature Step Index 3A
Temperature Step Tolerance Index 3B
Weight Step Index 3C
Temperature Base Index 3D
Temperature Fan Duty Base Index 3E
Step Up Time Index 03
Step Down Time Index 04
StepUp Index 66 [7:4] StepDown Index 66 [3:0]
MODE Index 02 [7:4] Tolerance Index 02 [2:0]
Enable weighting and temperature source select Index h39
Fan Control Duty Mode

Programming Flow

(1). The programming sequence is shown below. (Take CPUFANOUT as an example)

A. Set the temperature: T1 ~ T4

Bank2, Index21 bit [7:0] = 0x28h, T1=40℃

Bank2, Index22 bit [7:0] = 0x32h, T2=50℃

Bank2, Index23 bit [7:0] = 0x3Ch, T3=60℃

Bank2, Index24 bit [7:0] = 0x46h, T4=70℃

B. Set the critical temperature

Bank2, Index35 bit [7:0] = 0x50h, TR=80℃

C. Set DC/PWM Level: Y1 ~ Y4

Bank2, Index27 bit [7:0] = 0x40h, Y1=64

Bank2, Index28 bit [7:0] = 0x50h, Y2=80

Bank2, Index29 bit [7:0] = 0x60h, Y3=96

Bank2, Index2A bit [7:0] = 0xF0h, Y4=240

D. Set Weighting Relative Registers

Bank2 index3A bit [7:0] = 0x05h

Bank2 index3B bit [7:0] = 0x02h

Bank2 index3C bit [7:0] = 0x0Ah

Bank2 index3D bit [7:0] = 0x28h

Bank2 index3E bit [7:0] = 0x14h

E. Set Step Up Time and Step Down Time

Bank2 index03 bit [7:0] = 0x01h

Bank2 index04 bit [7:0] = 0x01h

(Step Up Time and Step Down Time = 0.05s)

F. Set Step Up Value and Step Down Value

Bank2 index66 bit [7:4] = 0x01h

Bank2 index66 bit [3:0] = 0x01h

(Step Up Value and Step Down Value = 1)

G. Set Mode Select and Tolerance

Bank2 index02 bit [7:0] = 0x40h

(Mode Select Smart Fan IV Mode, Tolerance = 0)

H. When Temperature = 10℃

Bank0 index03 bit [7:0] = 0x40h

I. Enable CPUFAN Weight

Bank2 index39 bit [7:0] = 0x81h

(EN_CPUFAN_WEIGHT)

J. Set the temperature, and check the corresponding FanDuty

When Temperature = 40℃

Bank0 index03 bit [7:0] = 0x54h

When Temperature = 45℃

Bank0 index03 bit [7:0] = 0x65h

When Temperature = 50℃

Bank0 index03 bit [7:0] = 0x77h

When Temperature = 55℃

Bank0 index03 bit [7:0] = 0x89h

When Temperature = 65℃

Bank0 index03 bit [7:0] = 0xEDh

When Temperature = 64℃

Bank0 index03 bit [7:0] = 0xDFh

When Temperature = 63℃

Bank0 index03 bit [7:0] = 0xD1h

When Temperature = 62℃

Bank0 index03 bit [7:0] = 0xB8h

When Temperature = 53℃

Bank0 index03 bit [7:0] = 0x86h

When Temperature = 67℃

Bank0 index03 bit [7:0] = 0xFFh

When Temperature = 48℃

Bank0 index03 bit [7:0] = 0x74h

When Temperature = 50℃

Bank0 index03 bit [7:0] = 0x77h

（2）Relative Register-at SMART FAN IV Control Mode (with weighting)

DESCRIPTION	**T1**	**T2**	**T3**	**T4**
CPUFANOUT	Bank 2, index 21h	Bank 2, index 22h	Bank 2, index 23h	Bank 2, index 24h

DESCRIPTION	**DC/PWM1**	**DC/PWM2**	**DC/PWM3**	**DC/PWM4**
CPUFANOUT	Bank 2, index 27h	Bank 2, index 28h	Bank 2, index 29h	Bank 2, index 2Ah
DESCRIPTION	**CRITICAL TEMPERATURE**	**Critical Tolerance**	**Monitored temperature tolerance**	**ENABLE SMART FAN IV**
CPUFANOUT	Bank 2, index 35h	Bank 2, index 38h bit[2:0]	Bank 2, index 02h bit[2:0]	Bank 2, index 02h bit[7:4]
DESCRIPTION	**STEP UP TIME**	**STEP DOWN TIME**	**STEP UP VALUE**	**STEP DOWN VALUE**
CPUFANOUT	Bank 2, index 03h	Bank 2, index 04h	Bank 2, Index 66h bit [7:4]	Bank 2, index 66h bit[3:0]
DESCRIPTION	**FAN2_TR1_DELAY_REG**	**Temperature Step**	**Temperature Step Tolerance**	**Weight Step**
CPUFANOUT	Bank 2,	Bank 2, index 3Ah	Bank 2, index 3Bh	Bank 2, index

DESCRIPTION	T1	T2	T3	T4
	index 67h			3Ch
DESCRIPTION	**TEMPERATURE BASE**	**Temperature Fan Duty Base**		
CPUFANOUT	Bank 2, index 3Dh	Bank 2, index 3Eh	Bank 2, index 3Bh	Bank 2, index 3Ch

Note: The duration (D) that the temperature keeps at T0 more than FANX_TR1_DELAY_REG, that is, D > FANX_TR1_DELAY_REG, FanDuty will drop to 0.

3.5.3.7. Virtual Temperature for Smart Fan IV Mode (Y-axis: FanDuty)

(1). The programming sequence is shown below. (Take SYSFANOUT as an example)

A. Set the virtual temperature

Bank0, index EAh

B. Check the virtual temperature

Bank0, index 73h

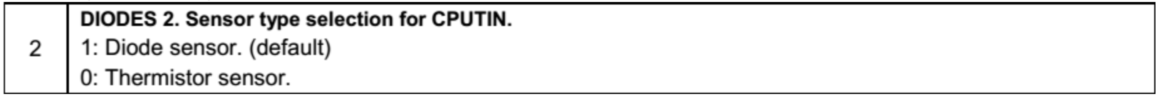
C. Set SYSFANOUT temperature source

Bank 0, index 00h

3.6 OVT

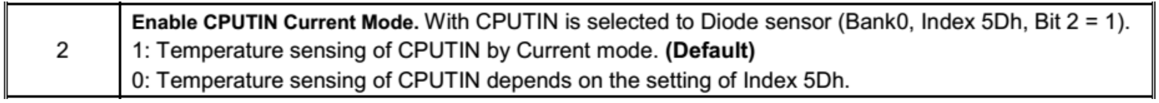
Take SMIOVT1 as an example.

1. Change the sensor mode (Ex: CPUTINTIN change to diode mode from thermistor mode.)
   Bank 0, index 5D, bit2



2	**DIODES 2. Sensor type selection for CPUTIN.** 1: Diode sensor. (default) 0: Thermistor sensor.

   Bank 0, index 5E, bit 2



2	**Enable CPUTIN Current Mode.** With CPUTIN is selected to Diode sensor (Bank0, Index 5Dh, Bit 2 = 1). 1: Temperature sensing of CPUTIN by Current mode. **(Default)** 0: Temperature sensing of CPUTIN depends on the setting of Index 5Dh.

   P.S The update time is 352ms (min.).

2. Set High / Hyst. limit temperature (Bank 0, index 39h / 3Ah) to 0x7F
3. Set SMIOVT1 temperature source (Bank 6, index 21h)
4. Check SMIOVT1 temperature (Bank 0, index 27h)
5. Read SMIOVT1 status (Bank0, index 41h)
6. Enable OVT (Bank 0, index 18h)
7. Modify High / low limit temperature (Bank 0, index 39h / 3Ah) to the design range.

3.7 Reading the correct temperature value by AMDTSI

NCT5585D could read up to 8 CPU addresses via SB-TSI interface and thus control the fan output speed basing on the temperature from SB-TSI.

The following example is to enable CPU address 0x98 via SB-TSI.

(SB-TSI base address = 0x200 from Logical Device B CR62 / CR63, SB-TSI_agent0 address = 0x98)

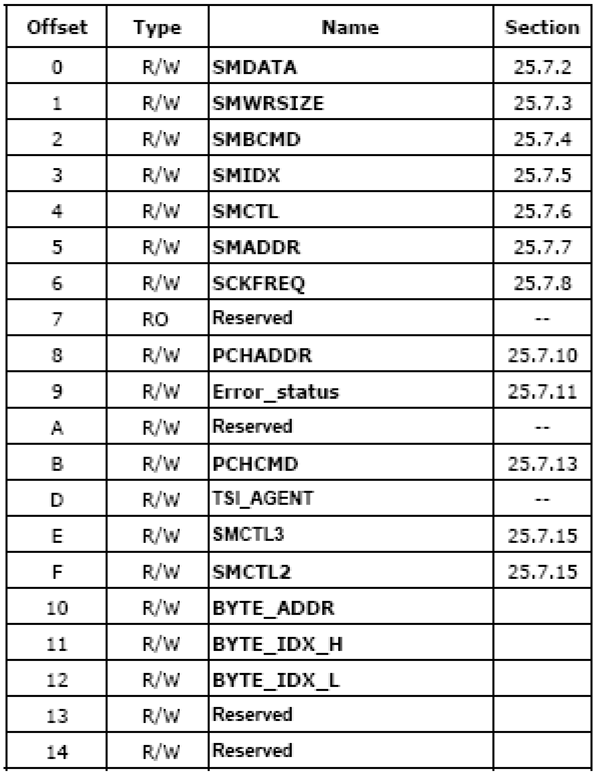
1. Set Global Control Register, CR2C bit 0 = 1 to select pin # 58 and pin # 60 functions as SB-TSI.

2. Logical Device B, CR62 and CR63 are the two registers to select the SB-TSI base address <100h: FFEh> along a two-byte boundary. Set CR 62 = 0x20 and CR63 = 0x00 for SB-TSI base address in this example.

3. Offset D ➔ 0x20D =0x01, enable 0x98 (TSI_agent0) address reading.

4. Bank 2 CR00 bits [4:0] are to select the CPUFAN temperature source. Set bits [0:4] as 00100 to enable SMBUSMASTER 0 as CPUFAN temperature source.

5. Read the value in the Bank 0, CR75. The value of this register is the CPUFAN temperature source. Since the CPUFAN temperature source is selected as SMBUSMASTER 0, it is the temperature read by SB-TSI (Source selection is in the Bank2, CR00 bits [4:0] described in the step # 4).

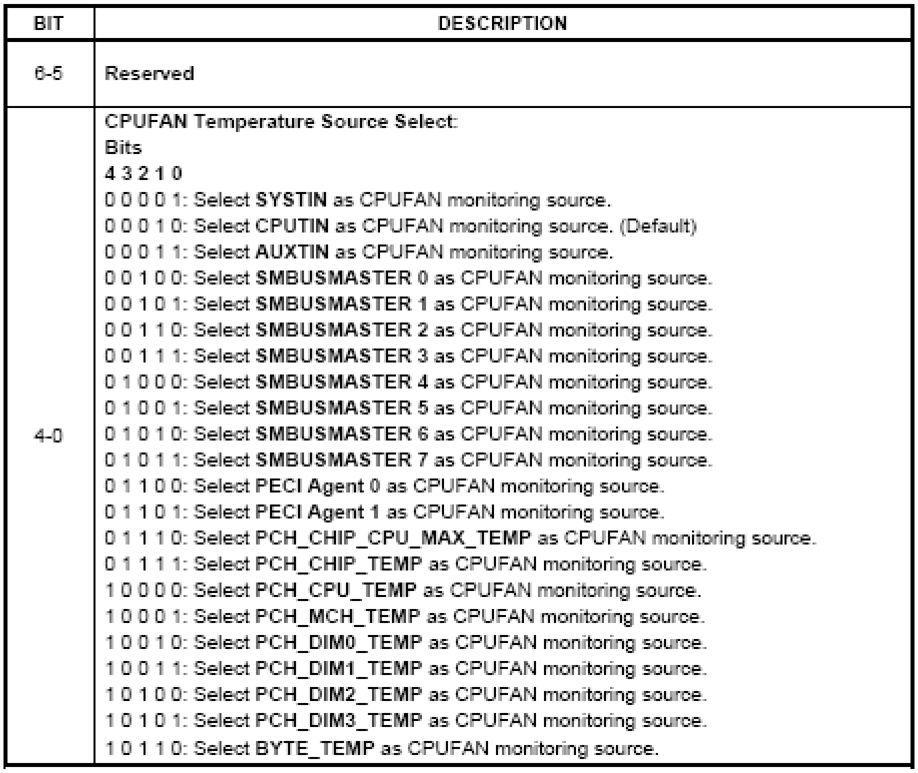


Offset	Type	Name	Section
0	R/W	SMDATA	25.7.2
1	R/W	SMWRSIZE	25.7.3
2	R/W	SMBCMD	25.7.4
3	R/W	SMIDX	25.7.5
4	R/W	SMCTL	25.7.6
5	R/W	SMADDR	25.7.7
6	R/W	SCKFREQ	25.7.8
7	RO	Reserved	--
8	R/W	PCHADDR	25.7.10
9	R/W	Error_status	25.7.11
A	R/W	Reserved	--
B	R/W	PCHCMD	25.7.13
D	R/W	TSI_AGENT	--
E	R/W	SMCTL3	25.7.15
F	R/W	SMCTL2	25.7.15
10	R/W	BYTE_ADDR	
11	R/W	BYTE_IDX_H	
12	R/W	BYTE_IDX_L	
13	R/W	Reserved	
14	R/W	Reserved	

Offset D bit 7:0

Bit	7	6	5	4	3	2	1	0
Name	AG7	AG6	AG5	AG4	AG3	AG2	AG1	AG0
Default	0	0	0	0	0	0	0	0

Bank 2 CR00 CPUFANOUT temperature source select register



BIT	DESCRIPTION
6-5	Reserved
4-0	**CPUFAN Temperature Source Select:** **Bits** **4 3 2 1 0** 0 0 0 0 1: Select **SYSTIN** as CPUFAN monitoring source. 0 0 0 1 0: Select **CPUTIN** as CPUFAN monitoring source. (Default) 0 0 0 1 1: Select **AUXTIN** as CPUFAN monitoring source. 0 0 1 0 0: Select **SMBUSMASTER 0** as CPUFAN monitoring source. 0 0 1 0 1: Select **SMBUSMASTER 1** as CPUFAN monitoring source. 0 0 1 1 0: Select **SMBUSMASTER 2** as CPUFAN monitoring source. 0 0 1 1 1: Select **SMBUSMASTER 3** as CPUFAN monitoring source. 0 1 0 0 0: Select **SMBUSMASTER 4** as CPUFAN monitoring source. 0 1 0 0 1: Select **SMBUSMASTER 5** as CPUFAN monitoring source. 0 1 0 1 0: Select **SMBUSMASTER 6** as CPUFAN monitoring source. 0 1 0 1 1: Select **SMBUSMASTER 7** as CPUFAN monitoring source. 0 1 1 0 0: Select **PECI Agent 0** as CPUFAN monitoring source. 0 1 1 0 1: Select **PECI Agent 1** as CPUFAN monitoring source. 0 1 1 1 0: Select PCH_CHIP_CPU_MAX_TEMP as CPUFAN monitoring source. 0 1 1 1 1: Select PCH_CHIP_TEMP as CPUFAN monitoring source. 1 0 0 0 0: Select PCH_CPU_TEMP as CPUFAN monitoring source. 1 0 0 0 1: Select PCH_MCH_TEMP as CPUFAN monitoring source. 1 0 0 1 0: Select PCH_DIM0_TEMP as CPUFAN monitoring source. 1 0 0 1 1: Select PCH_DIM1_TEMP as CPUFAN monitoring source. 1 0 1 0 0: Select PCH_DIM2_TEMP as CPUFAN monitoring source. 1 0 1 0 1: Select PCH_DIM3_TEMP as CPUFAN monitoring source. 1 0 1 1 0: Select BYTE_TEMP as CPUFAN monitoring source.

Below is an example for Thermal profile setting.

Take TSI source (0x98/0x9A) to control CPUFAN and AUXFANOUT3 by SMF4;

Step:

1) Multi-function and H/M IO base address

Global	Value	Default value
CR2C	Bit 0 = 1	Bit 0 = 1
LD5	**Value**	**Default value**
CR30	0x01	0x00
CR60	0x02	0x00
CR61	0x90	0x00
CR62	0x02	0x00
CR63	0x00	0x00
0x20D	BIT 1:0 = 00b	BIT 1:0 = 11b

PS: BIOS needs to enable related decode range of Chipset.

2) Select temperature source for each fan control output:

	SYSFANOUT	CPUFANOUT	AUXFANOUT3
Fan Control Temperature Source Select	Bank1, index00 bit[4:0] Default: Reserve	Bank2, index00 bit[4:0] Default: CPUTIN	BankA, index00 bit[4:0] Default: Reserve
Fan Control Temperature Reading	Bank0, index73 & Bank0, index74 bit7	Bank0, index75 & Bank0, index76 bit7	Bank0, index7D & Bank0, index7E bit7

3) SMF profile setting as below shown

For CPUFANOUT and AUXFANOUT SMFIV related register

DESCRIPTION	T1	T2	T3	T4
CPUFANOUT	Bank 2, index 21h	Bank 2, index 22h	Bank 2, index 23h	Bank 2, index 24h
AUXFANOUT3	Bank A, index 21h	Bank A, index 22h	Bank A, index 23h	Bank A, index 24h
DESCRIPTION	**DC/PWM1**	**DC/PWM2**	**DC/PWM3**	**DC/PWM4**
CPUFANOUT	Bank 2, index 27h	Bank 2, index 28h	Bank 2, index 29h	Bank 2, index 2Ah
AUXFANOUT3	Bank A, index 27h	Bank A, index 28h	Bank A, index 29h	Bank A, index 2Ah

DESCRIPTION	T1	T2	T3	T4
DESCRIPTION	**CRITICAL TEMPERATURE**	**Critical Tolerance**	**Monitored temperature tolerance**	**ENABLE** **SMART FAN IV**
CPUFANOUT	Bank 2, index 35h	Bank 2, index 38h bit[2:0]	Bank 2, index 02h bit[2:0]	Bank 2, index 02h bit[7:4]
AUXFANOUT3	Bank A, index 35h	Bank A, index 38h bit[2:0]	Bank A, index 02h bit[2:0]	Bank A, index 02h bit[7:4]

4. SMBUS MASTER – MANUAL MODE

4.1 Hardware

SIO NCT5585D pin33/34 can be SMBus Master.

Please note SIO SMBus Master works at S0 state only.

4.2 Software

CR27 bit4 = 0b

CR1A bit[3:2] = 10b

CR1B bit[2:0] = 111b //set pin.33/34 to be SMBus Master interface

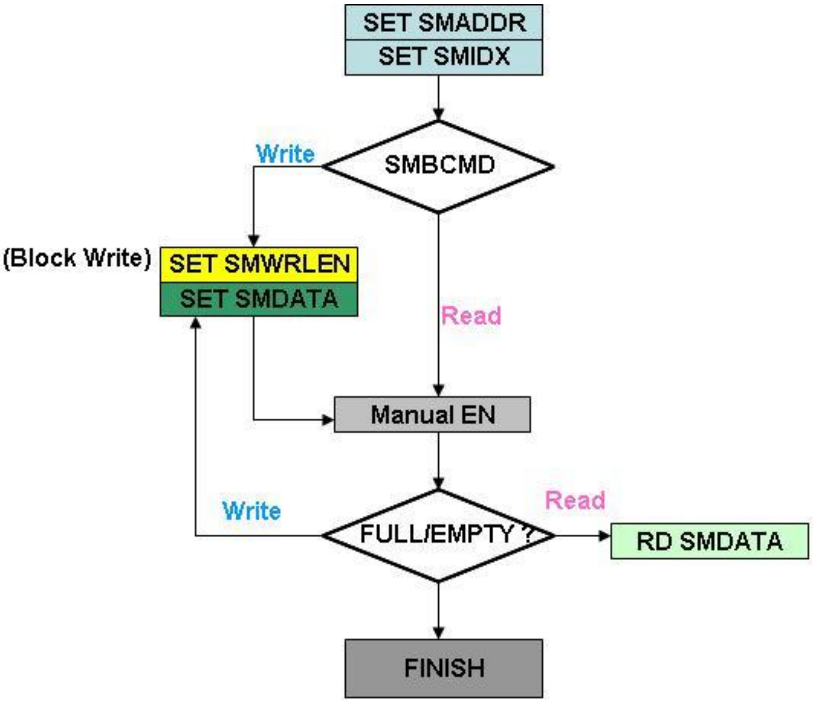
SMBus Master Manual Mode Programming Flow

(1) Assign a I/O base address for SIO SMBus Master operation

Logic Device B, CR62=02h, CR63=A0h //assign I/O base address 0x2A0h for SIO

Logic Device B, CR30=01h //enable SIO SMBus Master decode

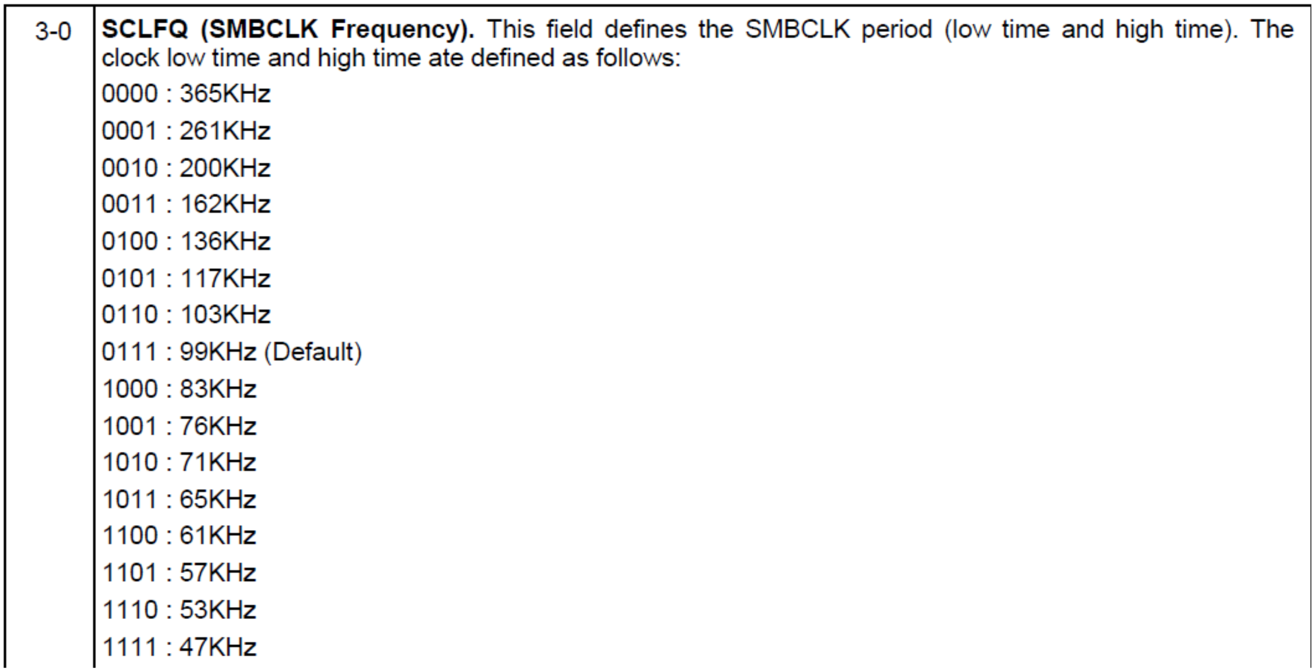
BIOS add I/O address 0x2A0h to chipset generic I/O decoder.



(2) Configure SMBus clock of Master

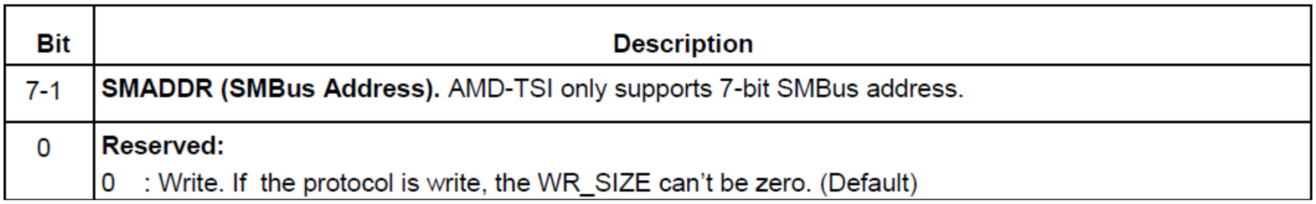
Write I/O address 0x2A6h bit[3:0]		//base address +6

3-0 SCLFQ (SMBCLK Frequency). This field defines the SMBCLK period (low time and high time). The clock low time and high time ate defined as follows:



3-0	**SCLFQ (SMBCLK Frequency).** This field defines the SMBCLK period (low time and high time). The clock low time and high time ate defined as follows: 0000 : 365KHz 0001 : 261KHz 0010 : 200KHz 0011 : 162KHz 0100 : 136KHz 0101 : 117KHz 0110 : 103KHz 0111 : 99KHz (Default) 1000 : 83KHz 1001 : 76KHz 1010 : 71KHz 1011 : 65KHz 1100 : 61KHz 1101 : 57KHz 1110 : 53KHz 1111 : 47KHz

(3) Set slave address to I/O address 0x2A5h		//base address +5

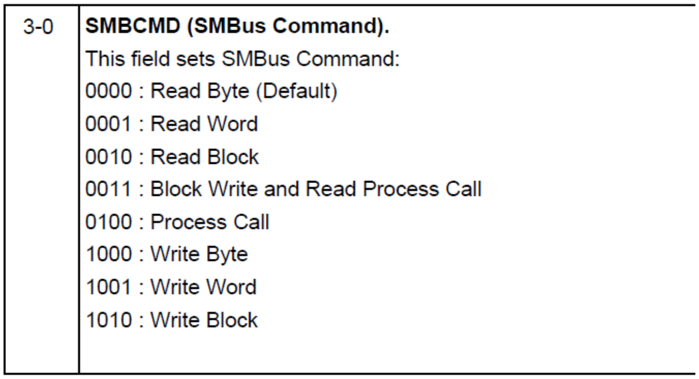


Bit	Description
7-1	**SMADDR (SMBus Address).** AMD-TSI only supports 7-bit SMBus address.
0	**Reserved:** 0 : Write. If the protocol is write, the WR_SIZE can't be zero. (Default)

(4) Select SMBus command

Write I/O address 0x2A2h bit[3:0] //base address +2

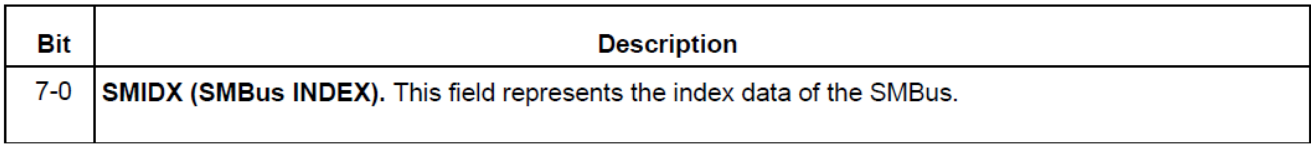
3-0 SMBCMD (SMBus Command).



3-0	**SMBCMD (SMBus Command).** This field sets SMBus Command: 0000 : Read Byte (Default) 0001 : Read Word 0010 : Read Block 0011 : Block Write and Read Process Call 0100 : Process Call 1000 : Write Byte 1001 : Write Word 1010 : Write Block

(5) Set slave index(or command) to I/O address 0x2A3h //base address +3

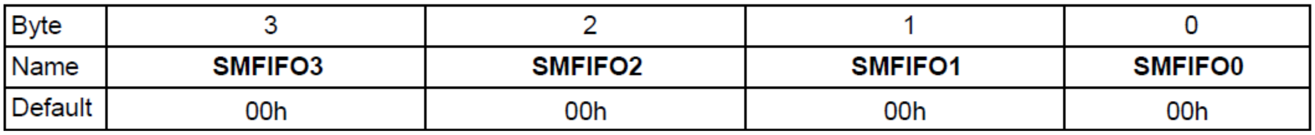
7-0 SMIDX (SMBus INDEX). This field represents the index data of the SMBus.



Bit	Description
7-0	**SMIDX (SMBus INDEX).** This field represents the index data of the SMBus.

(6) Prepare write data (for write operation)

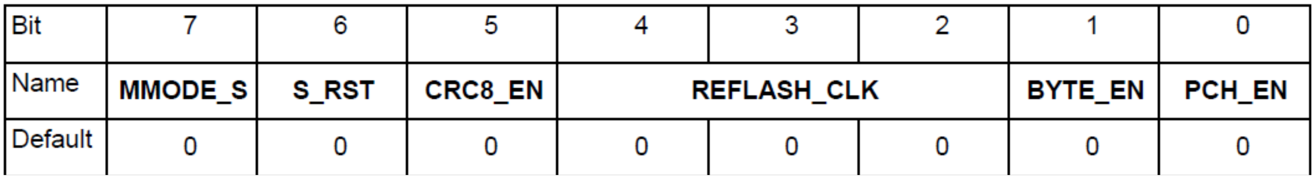
Write I/O address 0x2A0h //base address +0, 4-byte FIFO



Byte	3	2	1	0
Name	**SMFIFO3**	**SMFIFO2**	**SMFIFO1**	**SMFIFO0**
Default	00h	00h	00h	00h

(7) Enable Master Manual mode

Write I/O address 0x2A4h bit7 = 1b //base address +4, one-shut enable



Bit	7	6	5	4	3	2	1	0
Name	**MMODE_S**	**S_RST**	**CRC8_EN**	**REFLASH_CLK**			**BYTE_EN**	**PCH_EN**
Default	0	0	0	0	0	0	0	0

4.3 Manual Mode Example

Byte Read

Above (1) + (2) + (3) + (4)(=0000) + (5) + (7)

Read slave data(byte) from base address +0 SMFIFO.

Byte Write

Above (1) + (2) + (3) + (4)(=1000) + (5) + (6)(one-byte write data) + (7)

Word Read

Above (1) + (2) + (3) + (4)(=0001) + (5) + (7)

Read slave data(word, two-byte) from base address +0 SMFIFO.

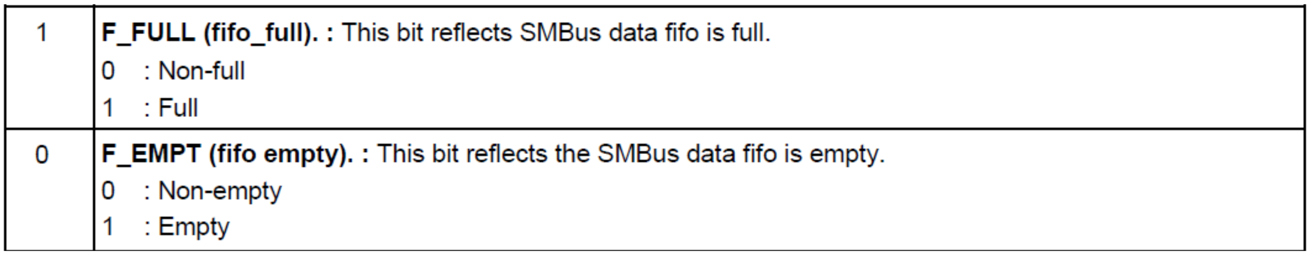
Word Write

Above (1) + (2) + (3) + (4)(=1001) + (5) + (6)(word, two-byte write data) + (7)

Block Read

Above (1) + (2) + (3) + (4)(=0010) + (5) + (7)

Read slave data from base address +0 SMFIFO with “base address +E FIFO status”.

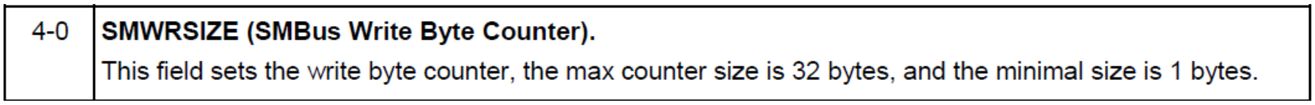


1	**F_FULL (fifo_full). :** This bit reflects SMBus data fifo is full. 0 : Non-full 1 : Full
0	**F_EMPT (fifo empty). :** This bit reflects the SMBus data fifo is empty. 0 : Non-empty 1 : Empty

Block Write

Above (1) + (2) + (3) + (4)(=1010) + (5) + set data length + (6)(multiple-byte write data) + (7)

In Block Write operation, we need to set data length(base address +1 bit[3:0]) first,



4-0	**SMWRSIZE (SMBus Write Byte Counter).** This field sets the write byte counter, the max counter size is 32 bytes, and the minimal size is 1 bytes.

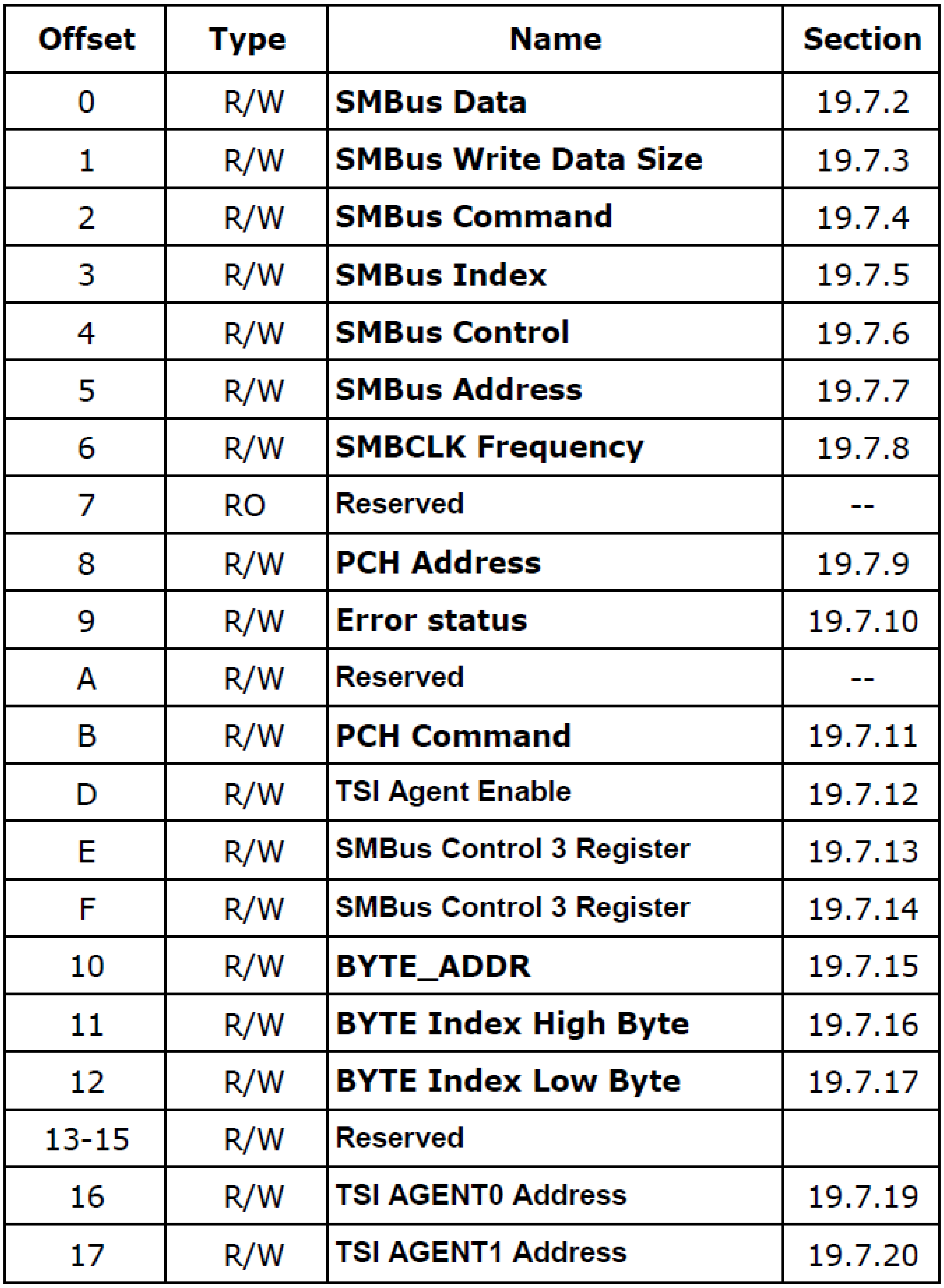
during operation, we also need to put write data with base address +E FIFO status.

4.4 Polling Mode Example

Access related registers of SIO SMBus Master function:

(Access method is “base address + offset”, please refer to SIO datasheet chapter – SMBus Master.)

Table 19-3 SMBus Master Bank 0 Registers



Offset	Type	Name	Section
0	R/W	**SMBus Data**	19.7.2
1	R/W	**SMBus Write Data Size**	19.7.3
2	R/W	**SMBus Command**	19.7.4
3	R/W	**SMBus Index**	19.7.5
4	R/W	**SMBus Control**	19.7.6
5	R/W	**SMBus Address**	19.7.7
6	R/W	**SMBCLK Frequency**	19.7.8
7	RO	Reserved	--
8	R/W	**PCH Address**	19.7.9
9	R/W	**Error status**	19.7.10
A	R/W	Reserved	--
B	R/W	**PCH Command**	19.7.11
D	R/W	TSI Agent Enable	19.7.12
E	R/W	SMBus Control 3 Register	19.7.13
F	R/W	SMBus Control 3 Register	19.7.14
10	R/W	**BYTE_ADDR**	19.7.15
11	R/W	**BYTE Index High Byte**	19.7.16
12	R/W	**BYTE Index Low Byte**	19.7.17
13-15	R/W	Reserved	
16	R/W	TSI AGENT0 Address	19.7.19
17	R/W	TSI AGENT1 Address	19.7.20

4.4.1 Polling Mode (for PCH Thermal Reporting)

(1) Write data 0x00h to I/O address 0x2AFh //Set base address + Fh = 0x00h, locate to bank0 registers.

(2) Write data 0x40h to I/O address 0x2A4h //Set base address + 4h = 0x40h, reset SIO SMBus Master module (it would keep SIO in reset situation).

(3) Write data 0x00h to I/O address 0x2A4h //Set base address + 4h = 0x00h, stop reset.

(4) Write data 0x07h to I/O address 0x2A6h //Set base address + 6h = 0x07h, MSCL clock is 99KHz.

(5) Write data 0x96h to I/O address 0x2A8h //Assign PCH ME strap – SMLink1 GP address to base address + 8h. (PCH address should be placed into bit[7:1])

(Actual SMLink1 GP address, please refer to
"Intel(R)_ME_Ignition_FW_Bring_Up_Guide".)

(6) Write data 0x40h to I/O address 0x2ABh //Set base address + Bh = 0x40h, PCH command is 0x40h.

(7) Write data 0x01h to I/O address 0x2A4h //Set base address + 4h = 0x01h, enable Polling Mode to read PCH thermal data.

(8) Bank6, index21, bit[4:0]=10011 //Select PCH chip temperature to be the input for SMIOVT1.

(9) We can get PCH chip temperature reading from Bank0, index27 register (It's 2's complement format).

For example: Bank0, index27 = 0x30h

Means PCH chip temperature is 48 degree-C.

5. WATCH DOG TIMER

The Watchdog Timer 1 of the NCT5585D consists of an 8-bit programmable time-out counter and a control and status register. GPIO2, GPIO8 provides an alternative WDT1 function. This function can be configured by the relative GPIO control register. The units of Watchdog Timer counter can be selected at Logical Device 8, CR[F5h], bit[3]. The time-out value is set at Logical Device 8, CR[F6h]. Writing zero disables the Watchdog Timer function. Writing any non-zero value to this register causes the counter to load this value into the Watchdog Timer counter and start counting down.

When Watchdog Timer 1 time-out event is occurring, GPIO2, GPIO8], PWROK and RESETCONO# will trigger a low pulse apporx 100mS or low level by Logical Device 8 CR[F5h], bit[0] and RSMRST# will trigger a low pulse approx 250ms. The PWROK, SLP_S3# and RSMRST# event also relate to acpi sequence, that can be control by Logical Device D, CR[F0h], bit[7] and bit[0], Logical Device D, CR[F0h], bit[7] and bit[0] as 2'b00 is normal acpi function, others timing illustrations are define in Figure 17-1 to Figure 17-3 In other words, when the value is counted down to zero, the timer stops, and the NCT5585D sets the WDT1 status bit in Logical Device 8, CR[F7h], bit[4]. Writing a zero will clear the status bit. it. This bit will also be cleared if LRESET# or PWROK signal is asserted.

Watchdog Timer related registers are listed below.

Watchdog Timer has four registers in Logic Device 8. When CR30h bit0=1, Watchdog Timer circuit is activated, then CRF5, F6, and F7h are meaningful. Watchdog Timer reset source selection is located at Logic Device A, CRE7h, bit3.

Logical Device Number	Register address	Register Description
Logical Device 8	CR30h, bit0	Watchdog Timer circuit enable / disable
Logical Device 8	CRF5h	Watchdog Timer Control Mode Register
Logical Device 8	CRF6h	Watchdog Timer Counter Register
Logical Device 8	CRF7h	Watchdog Timer Control & Status Register
Logical Device A	CRE7h, bit3	Watchdog Timer reset source is LRESET# or PWROK

5.1 Enable Timer

(1) CR30h bit0 = 1, enable Watchdog Timer circuit.

(2) Configure CRF5h for time unit or other options.

(3) Set time limit into CRF6h then hardware will auto count down. New time limit value overwriting is acceptable. When CRF6h counts down to 0x0, hardware outputs timeout event on both flag (CRF7h bit4) and pin (via GPIO).

(4) Check then clear CRF7h bit4 (watchdog timeout flag), the timer will restart counting and de-assert pin (via GPIO).

5.2 Disable Timer

(1) Set 0x0 to CRF6h, stop Watchdog Timer counting.

(2) CR30h bit0 = 0, disable Watchdog Timer circuit if no longer use.

5.3 Application Note

(1) When Watchdog Timer is counting, if want to terminate, BIOS or program MUST stop timer (set 0x0 to CRF6h) first then call platform reset.

(2) Watchdog Timer function is reset by LRESET# or PWROK. Cold reset (front panel reset button), warm reset (CTRL-ALT-DEL), or any application that drives LRESET# event, Watchdog Timer would be reset.

(3) When Watchdog timeout, the WDTO pin (via GPIO) keeps asserting until flag (CRF7h bit4) is cleared to 0.

6. I^{2}C ACCESS

Besides LPC bus access, SIO NCT5585D also supports the access of SMBus Byte-Read and Byte-Write protocol. Following NCT5585D registers supports both LPC bus and SMBus access:

(1) Hardware Monitor registers

(2) GPIO control registers.

6.1 Hardware Monitor Registers

(1) Through LPC bus, BIOS assigns a SMBus slave address for SIO Hardware Monitor in advance.

[LPC I/O Write]

→ SIO Hardware Monitor register → Bank0 index48h = 0x2Dh

//set SMBus slave address of SIO Hardware Monitor to 0x2Dh

(2) Example for reading SIO CPUVCORE reading

a. [SMBus Byte Write]

Slave address	Command	Data
0x2Dh	0x4Eh	*0x04h*

//set SIO Hardware Monitor register *Bank=4*

b. [SMBus Byte Read]

Slave address	Command
0x2Dh	*0x80h*

//read SIO CPUVCORE reading from Bank0 index80h register

(3) Example for selecting temperature source for SIO CPUFANOUT

a. [SMBus Byte Write]

Slave address	Command	Data
0x2Dh	0x4Eh	*0x02h*

//set SIO Hardware Monitor register *Bank=2*

b. [SMBus Byte Write]

Slave address	Command	Data
0x2Dh	0x00h	*0x02h*

//select *CPUTIN* temperature for SIO CPUFANOUT

// Bank2 Index00h, bit[4:0]=00010

6.2 GPIO Registers

Besides LPC bus access, following SIO Logic Device registers also support SMBus access:

(1) Through LPC bus, BIOS assigns a SMBus slave address for SIO Logic Device register in advance.
[LPC I/O Write] SIO Logic Device register
Logic Device F, CRF0 = 0x9Dh //set SMBus slave address to 0x1Dh and enable it (bit7=1).

(2) Example for reading GPIO41 data

a. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0x80h	0x09h

//set SIO Logic Device number = 9

b. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0x81h	0x10h

//enable SIO GPIO4 group (note: 0x81h = LDN9, CR30h)

c. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0xF0h	0xFFh

//set SIO GPIO47~40 to be input

d. [SMBus Byte Read]

Slave address	Command
0x1Dh	0xF1h

//read SIO GPIO40 from return data bit0 of Logic Device 9h CRF1h.

(3) Example for setting GPIO42 to output ‘1’

a. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0x80h	0x09h

//set SIO Logic Device number = 9

b. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0x81h	0x10h

//enable SIO GPIO4 group (note: 0x81h = LDN9, CR30h)

c. [SMBus Byte Write]

Slave address	Command	Data

0x1Dh	0xF1h	0x04h

//prepare output data ‘1’ to GPIO42 (bit2=1)

d. [SMBus Byte Write]

Slave address	Command	Data
0x1Dh	0xF0h	0xFBh

//set GPIO42 to be output (bit0=0)