Design of AI Accelerator Board Management ang monitoring system using SMBus Interface

Vaishaka N Raj
Department of Electronics and
Instrumentation
RV College of Engineering,
Bengaluru, Karnataka
vaishakanraj.ei17@rvce.edu.in

Tabitha Janumala
Assistant Professor
Department of Electronics and Instrumentation
RV College of Engineering,
Bengaluru, Karnataka
tabithajanumala@rvce.edu.in

Dinesh Naik
System Lead, IOTG
Intel Corporation,
Bengaluru, Karnataka
dinesh.naik@intel.com

Abstract- An Accelerator is a specialized hardware which connects to the main CPU/Intelligence and helps the system in offloading AI specific applications or workloads. In the same front an AI accelerator is one that helps in offloading AI/ML based workloads from the main system and performs it more efficiently.

In this paper we discuss about the design an AI accelerator board management and monitoring system using SMBus interface where the system consists of devices such as temperature sensor, Fan Controller, GPIO expander, power management system and memory unit whose parameters must be monitored. For the monitoring of the board parameters we create a GUI that is user friendly and provides easy access to monitor the status and health of each device. The design entry of the AI Accelerator board management and monitoring system is carried out in Cadence Allegro which provides a 2-dimensional preview of the system circuit. The Monitoring tool is created in python platform thus gives the provision to monitor various board parameters remotely through GUI.

Keywords – AI Accelerator, Schematics, GUI, management, and monitoring tool, I2C, SMBus, PCIe.

I. Introduction

Intel AI Accelerator is a specialized hardware setup for deep neural network interface. It is used in implementing unique vision, image, and deep neural network workloads on chip. It achieves a mix of power efficiency and computation performance by integrating highly parallel programmable compute with workload-specific hardware acceleration in a unique architecture that reduces data transmission. The technology supports deep neural network and computer vision-based applications in intelligent cameras, edge servers, and AI appliances.

The main objective of the paper is to design an AI accelerator board management and monitoring system using SMBUS interface and to develop a user-friendly tool to monitor the parameter of devices on board. The Schematic entry of the system is designed using the Cadence Allegro Design Entry software where the component selection is done based on the specifications and requirements of the system. The Monitoring tool is developed in python platform in the form of GUI that allows the user to select the PCIe Card among the cluster.

The rest of the paper is organized as follows. Section II discusses the Block diagram of the AI Accelerator Management System, Section III discusses the Design of Schematics of the Management System, Section IV discusses the development of Monitoring tool to monitor the board parameters, Section V discusses the result/outcomes of the paper and the section VI concludes the paper.

II. Block Diagram

The AI Accelerator board management system consists 2 SOC's that are connected to several devices such as Temperature sensors, Fan controller, GPIO Expander and power management unit (ADC) connected. Figure. 1 shows the block diagram of the Board Management system. The supply voltages for EEPROM, temperature sensors, fan controller and GPIO Expander are 3.3V and that for ADC is 1.8V. Hence for this purpose we use level shifter to shift the voltage levels.
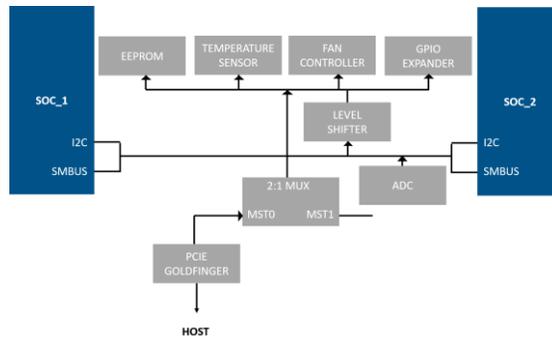
Figure. 1 Block Diagram

Since I2C and SMBus are shorted together, I2C acts as master and SMBus acts as slave. I2C can send commands and initiate transfer, but SMBus can only respond to the command sent by I2C.

2.1 Interface

Multiple PCIe cards are connected to host through the PCIe Gold finger [PCIe interface and SMBUS] where the management tool uses the SMBUS interface to talk to the devices such as Temperature sensor, fan controller and power monitor on each card and there by reporting status of each card back to the host.

I2C Communication

I2C which translates to Inter-Integrated Circuit is serial synchronous communication protocol that supports multi-masters to communicate with multiple slaves. It uses two wires, SDA and SCL for exchange of data. SDA is used for data transmission and SCL is used for clock transmission for synchronization. The standard speed of I2C data transmission is 100kHz. It also supports 400kHz, 3.4Mhz and 5Mhz.
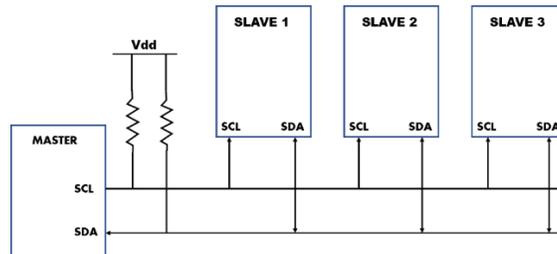

Figure. 2 Master-Slave Connection

Figure. 2 depicts the connection between a single master and multiple slaves. Each slave is assigned a unique address using which the master can communicate with the slave it is targeted to exchange data. Figure. 3 shows the data transfer formats where the start bit indicates the start of the communication. I2C supports two addressing modes 7-bit and 10-bit with 1-bit of read/write bit and 1 bit of acknowledgement from the slave. Once the ACK is received by the slave the master reads/writes 8-bit of data. The stop bit represents the completion of data exchange.

1 Start Bit	7- or 10-bit address	1 bit R/W	1 bit ACK	8-bit Data	1 Stop Bit

Figure. 3 Data transfer format

The start bit indicated by pulling the SDA line high to low when SCL is high. The stop bit is indicated by pulling the SDA line back to high from low when SCL is high. The data transmission waveform is shown in Figure. 4. The I2C devices are open drain, which implies a pull-up resistor is required at output to pull the output from low to high. Hence the SDA and SCL lines are high at idle condition.
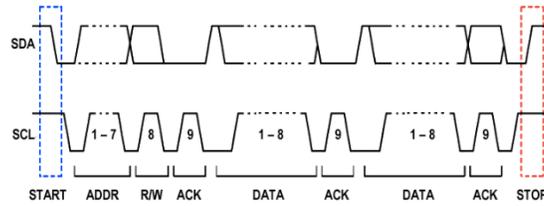
Figure. 4. I2C Data Transmission [Courtesy: electricimp]

The pull up resistors play an important role is effective transmission of data. Hence the resistor value must be calculated optimally. The lower and upper limit of the resistor values to be connected at the output of I2C line is given by Eq. [1] and [2] respectively.

$$Rp[min] = \frac{Vcc - Vol[max]}{Iol}$$ Eq. [1]

$$Rp[max] = \frac{tr}{0.8473 \times Cb}$$ Eq. [2]

Where, Vol[max] is the maximum output low voltage of the device, Iol is the leakage current that flows from Vcc to GND when is pulled low, tr is the rise time of the device, Cb is the bus capacitance which include parasitic capacitance, capacitance of each device, pin capacitance etc.

When two master wants to communicate simultaneously, arbitration is used to select the master that gets control of the bus. When two master send data simultaneously, which ever master sends a zero first get control of the bus and the other master will lose the control of the bus and must wait for the bus to freed.

Table 1 shows the characteristics of SDA and SCL I/O stages.

Table 1 Characteristics of SDA and SCL bus lines

Parameters	Condition	100 kHz	400 kHz	3.4 MHz
Vol (V)	Vdd > 2V	0.4	0.4	0.4
Vol (V)	Vdd < 2V	-	0.2 Vdd	0.2 Vdd
Iol (mA)	Vol = 0.4V	3	3	20
Iol (mA)	Vol = 0.6V	-	6	-
Rise time (tr) (ns)	-	1000	300	120
Load Capacitance (Cb) (pF)	-	400	400	550

System Management Bus

SMBus is a two-wire interface that allows different system component chips to communicate with one another and the rest of the system. It is based on the I2C operating principles. It modifies some of the I2C software and characteristics for better compatibility with the rapidly decreasing portable equipment's power supply budget. [1] Most of the I2C devices are compatible with SMBus interface.

The major difference between SMBus and I2C are its electrical characteristics. Table 2 shows the difference between I2C and SMBus [2]

Table 2 Difference between I2C & SMBus

Properties	I2C	SMBus
Supply voltage (Vdd)	Upto 5V	1.8 V – 5V
Threshold Voltage	Vil=30% Vdd Vih=70% Vdd	Vil = 0.8 V Vih = 1.35 V
Minimum bus speed	NA	10kHz
Speed	100kHz up to 5MHz	100kHz

PCIe

Peripheral Component Interconnect Express is a serial computer expansion bus that connects peripheral devices in applications like computing and communication platforms. It is a switched architecture of up to 32 independent, serial lanes that transfer data in full-duplex mode. A PCIe lane is a set of four wires or signal traces where each lane

employs two wires each to send and receive data, allowing the entire bandwidth to be used continuously in both directions.

Data packets are transmitted in simplex form at a signaling rate of 2.5 GHz. The transmitter and receiver pair each transport data at a rate of 250MB/sec, for a total of 500MB/sec. Multiple lanes can be used to construct a connection, which is symbolized by the letters x1, x4, and so on. The default standard configurations are x1, x4, x8, x16, and x32. Present desktop computers use x1, servers use x4, and graphics cards use x16 lanes. [3]

III. SCHEMATIC DESIGN

A schematic diagram is a two-dimensional representation of a circuit that shows the operation and connections between various electrical components. The schematic symbols that represent the components on a schematic diagram are critical for a PCB designer to understand.

The following sections shows the schematics of the management system that is designed using Cadence Allegro (Concept HDL) Software.

3.1. Temperature Sensor – TMP112

TMP112 is a High-Accuracy, Low-Power, Digital Temperature Sensors with SMBus and Two Wire Serial Interface. TMP112 has an accuracy of 0.5℃ to 1.0℃. TMP112 is used in applications where high accuracy is required. [4]

Address selection: The TMP112 family features an address pin to allow up to four devices to be addressed on a single bus. This is done by connecting the ADD0 pin to either Vcc, GND, SDA or SCL. In this system the ADD0 is connected to GND having the address 0x78.

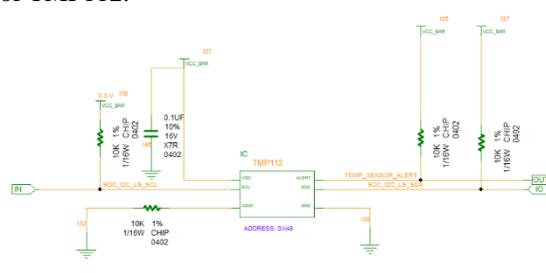Figure. 5 shows the schematics of TMP112.

Figure. 5 Schematics of TMP112

The ALERT pin is used to send an alert signal (TEMP_SENSOR_ALERT) to the SOCs during time of overtemperature. The input voltage is 3.3V. SCL and SDA lines are open drain hence require pull up resistor to drive the bus. The pullup resistor calculation is done according to the equation Eq. [1] and Eq. [2]

3.2. Fan Controller – EMC2103

EMC2103 is a SMBus compliant, RPM-based fan controller with thermal hardware shutdown. The temperature monitoring offers an accuracy of ±5℃ and an RPM accuracy of 2.5% of 500 to 6k RPM. It has a fixed I2C address of 0x2E. Figure. 6 shows the schematics of EMC2103. [5]
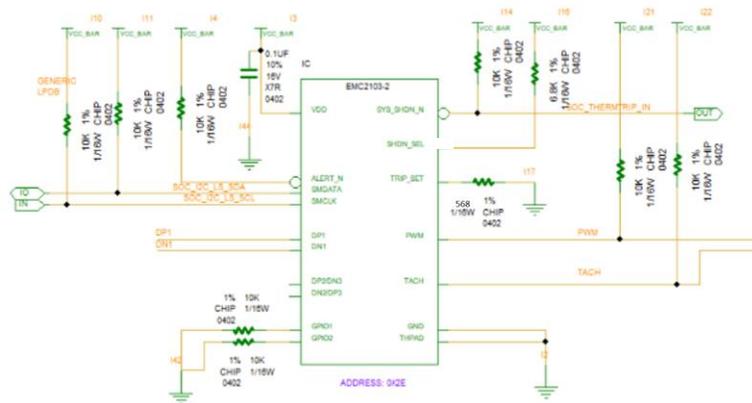

Figure. 6 Schematics of EMC2103

SHDN_SEL = Channel/mode configuration for critical/thermal shutdown. The pull-up resistor connected to the SHDN_SEL pin selects the Channel/Mode. In this application the pin is connected to 6.8K ohm resistor. The Mode selected is diode mode for hardware shutdown.

DP1/DN1 = The External Diode channel can support a diode-connected transistor. These are used as thermal diodes.
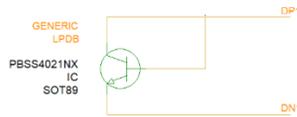

Fig. 7 Diode Connection

TRIP_SET = This pin is used for setting the critical shutdown temperature for the system. A voltage level is created at the input by connecting a resistor across the pin that translates to temperature. [6].

GPIO = These are general purpose input/output pins. Since in this application the GPIO pins are not used, they are connected to ground.

PWM = This pin is used to change the PWM duty cycle of the fan

TACH = This gives the RPM of the fan back to the controller

Fig. 8 shows the Fan Header connected to the EMC2103 Fan Controller.
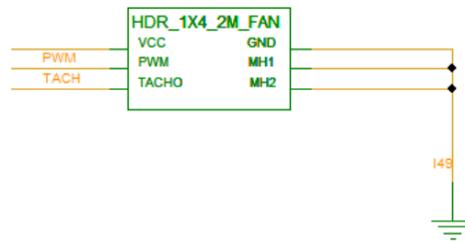
Fig. 8 Fan Header

THPAD = They are thermal pads used for effective dissipation of heat.

SYS_SHDN_N = It is an active low pin that sends a shutdown signal (SOC_THERMTRIP_IN) to both SOCs when the temperature of EMC2103 exceeds the critical temperature.

3.3. GPIO Expander – PCA9575

PCA9575 is 16-bit general purpose I/O Expansion used in low voltage processors with 2 banks. It has a fixed address of 0x20. [7] Reading and writing to the pins are done through registers. Input pins are only be read whereas the output pin can be read and written. Figure. 9 shows the schematics of GPIO Expander.
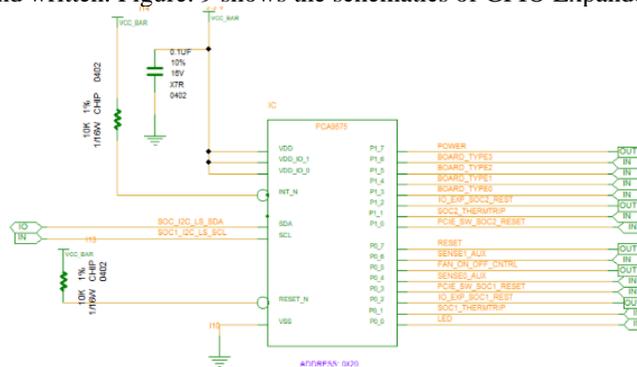

Figure. 9 Schematics of GPIO Expander

Since the active low interrupt and reset pins are not used, they are connected to Vcc.

Output pins – The GPIO Expander controls the board signal ID, SOC reset, SOC power and fan ON/OFF control.

3.4. Level Shifter – TCA9509

TCA9509 is a level translating I2C and SMBus bus repeater. It provides bidirectional voltage translation between low and high voltages. Voltage translation is between 3.3V and 1.8V. Figure. 10 shows the schematics of level shifter. [8]
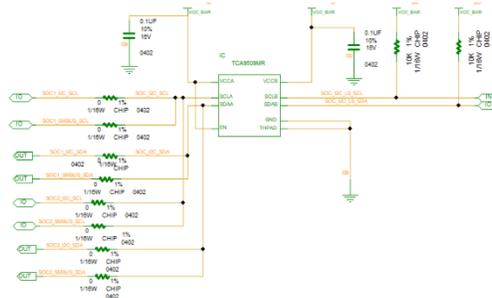

Figure. 10 Schematics of Level Shifter

The Vcca is 1.8 V side and Vccb is 3.3V side. The Vccb requires a pull up resistor to function. The SCL and SDA lines of SMBUS and I2C of both SOCs are shorted and connected to level shifter SCL and SDA respectively to provide proper communication between devices.

3.5. MUX – PCA9541

PCA9541 is 2-to-1 I2C bus master selector designed for high reliability dual master I2C application. It is SMBus compatible. It contains 2 masters and selects one master at a time. If the any master is disabled or removed from the system, either master can take control of the slave devices at any time. The system's failing master is isolated, and communication between the master and slave devices on the downstream I2C-bus is unaffected. [9]
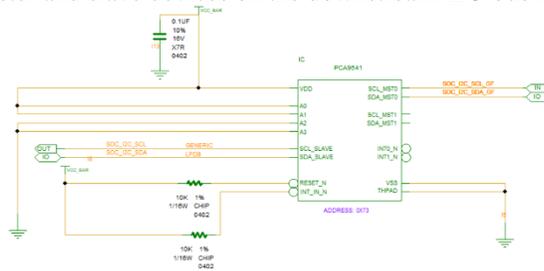
Figure. 11 Schematics of PCA9541

Address selection – The slave address is selected using the pins A0-A3.

1	1	1	A3	A2	A1	A0

Since A0, A1 is connected to Vcc and A2, A3 is connected to ground, the address of slave device is 0x73. To get access to other slave devices enabling the MUX is required.

3.6. Power Unit – ADS7142

The ADS7142 is a Nano power, Dual-Channel, Programmable Sensor Monitor that monitors signals independently while maximizing system power, reliability, and performance. [10] Figure. 12 shows the schematics of ADC which is used as power unit.
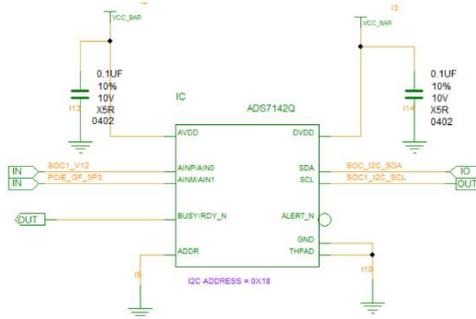
Figure. 12 Schematics of Power unit

AVDD is the analog reference voltage for Analog to digital convertor. DVDD is the digital supply voltage.
BUSY/RDY_N = When the device is scanning over channels in a sequence, it pushes this pin high, and when the sequence is done or terminated, it pushes this pin low.
AINP/AIN0 and AINM/AIN1 = Analog Signal input for channel 0 and 1.
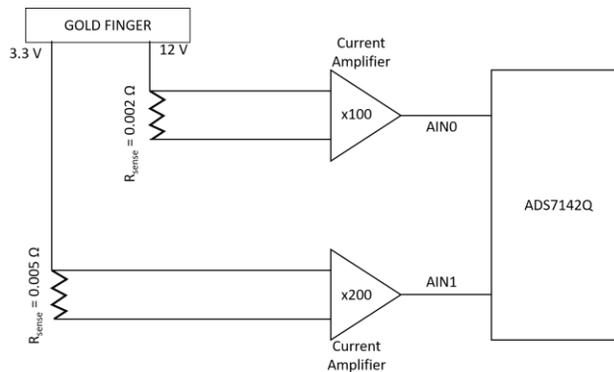The I2C Address of the device is 0x18 which is selected based on the connection of ADDR pin.

Figure. 13 Current Sense Amplifier

The 3.3 V and 12 V rails from the goldfinger is connected to a current sense amplifier through a shunt resistor (R_{sense}). The current sense amplifier measures voltage across the resistor. The output of the amplifier is sent to the ADC as input which is then converted to digital signals. The gains of Amplifier are 100 and 200 for input AIN0 and AIN1, respectively. R_{sense} of 3.3 V and 12 V are 0.002 Ω and 0.005Ω, respectively.

3.7. EEPROM – 24AA32A
It is a 32-bit electrically erasable PROM which is I2C compatible with a 2-wire serial interface. The device is structured as a single block of 4K x 8-bit memory and supports low-voltage design down to 1.7V [11]. Figure. 14 Shows the schematics of 24AA32A(EEPROM).
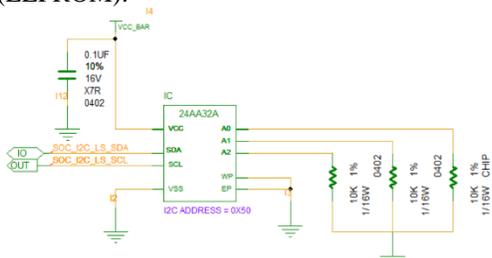

Figure. 14 Schematics of EEPROM

A0, A1, A2 are the Chip address selector pins that are used to select the address of the device. Thus, in the schematic by connecting the address pins to ground the address of the device becomes 0x50.

1	0	1	0	A2	A1	A0

WP = This pin is a write protect pin which is connected either or Vcc or Vss. When connected to Vcc it allows the user to write-protect the array. When it is connected to Vss/GND the write-protect is disabled.

3.8. Decoupling capacitors

A decoupling capacitor is connected across the power supply. To keep the current flowing smoothly, a decoupling capacitor stores energy and dissipates it back into the power line. A decoupling capacitor discharges the stored energy and momentarily supplies energy at the correct voltage if the power supply voltage decreases momentarily. As a result, these capacitors are also known as bypass caps since they can operate as a temporary power source by bypassing the power supply.
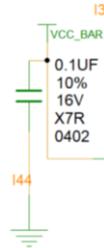
Figure. 15 Decoupling capacitor

IV. BOARD MONITORING TOOL

The bord monitoring tool is designed in the form of GUI using python platform. The GUI is used to access the on-board devices remotely in a user-friendly interface. The GUI is created in python platform using tkinter library.
Figure. 16 shows the flow chart of the monitoring tool functions. The board is selected through the IP address and the devices of the system connected through I2C/SMBus are accessed through enabling SMBus. Once the SMBus is enabled the MUX can be read/written. To read/write to the on-board devices like temperature sensor, fan controller etc., enabling the MUX is required. When the MUX is enabled the on-board devices can be accessed to monitor the parameters and perform required actions.
Reading or writing to the device is done by performing the read/write operation to the register of the device. This is done by using software addresses of the devices and their registers.
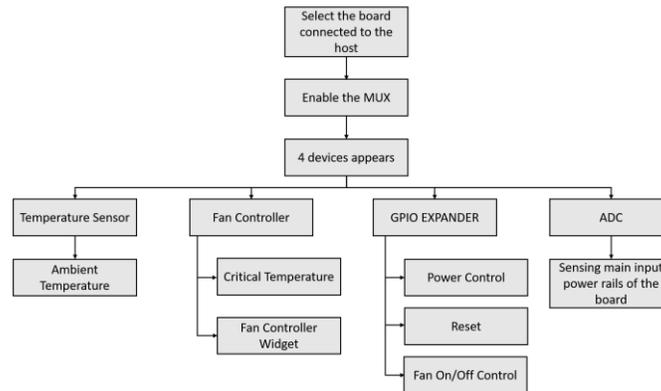

Figure. 16 GUI Flowchart

Prior to executing the monitoring tool, the host system must be plugged in the AI Accelerator card and the system should be turned on manually or remotely. Putty software is used to turn on the host remotely and the user must login into the AI Card.
Once the user has logged in the monitoring tool is executed and the functions of the tool are discussed in the following sections.

4.1. Function of Temperature sensor

The temperature sensor indicates the ambient temperature of the board.

4.2. Function of Fan controller

The fan controller returns the critical temperature value and allows to control the speed of fan. Since the EMC2103(Fan controller device) supports PWM modulation for fan speed variation. The critical temperature is read from the register 0x0a and the speed is controlled by writing the duty cycle value to register 0x40

4.3. Function of GPIO Expander

The GPIO Expander provide several input/output actions to be carried out that are decided based on the requirements. There are 2 ports each of 8-bit ranging from pin0 to pin7. The main function that are carried out by GPIO expander are to reset and power the board, sent board ID signals and controlling the ON/OFF the fan. These pins are output pins that can be toggled, apart from these output pins there are input pins whose value can be read but not changed.

4.4. Function of ADC

The Voltage level at ADC is read whenever required. In addition, the values are recorded in an excel sheet with date and time to identify any abnormality in the voltage levels. There are two channels whose value can be read.
ADC has two operating modes, Manual Mode and High precision mode.

    a)   *Manual Mode:*

The manual mode supports 12-bit accuracy. To get the data from ADC the slave device address (0x18) is sent followed by the SCL from the master. Once the ADC receives the SCL it sends back the raw data to master. The bytes of data received depends on the number of clock cycles sent.

    b)   *High Precision Mode:*

In high precision mode the data is store in accumulator. The data is read in FIFO format. To read the data the device address is sent with number of bytes to be read. The ADC sends 16 words in response which indicate the voltage levels at channel 1 and 2.

    c)   *Calculation of ADC voltage:*

The raw data obtained is obtained from MSB and LSB which is given by,

$$value = MSB \ll 8 | LSB \qquad\qquad Eq.\ 3$$

The value is the decimal format. Thus, the voltage is given by,

$$adc\ voltage = \frac{AVDD * value}{2^N} \qquad\qquad Eq.\ 4$$

Where AVDD = 3.3 V, N = 12 (Manual mode) or 16 (Precision Mode)

    d)   *Calculation of Current on Rails*

$$current\ I = \frac{adc\ voltage}{R_{sense} * Gain} \qquad\qquad Eq.\ 5$$

Where R_{sense} = 0.002 Ω (3.3 V) or 0.005 Ω (12 V)
Gain = 100 (3.3 V) or 200 (12 V)

4.5. Converting .py file to an executable file

Once the error free, required GUI is created in a python file the .py file can be converted to an executable file using the Pyinstaller library. The command should run in the command prompt having the directory of the location containing the .py file.
Command: Pyinstaller –onefile file_name.py
Once this command is executed in command prompt a .exe file is created the location specified which can be used to execute the program without physically executing the python file.

V. RESULT

This chapter discusses about the outcomes of the paper which includes the simulations and the GUI window of the monitoring tool.

5.1. Level shifter voltage levels

Figure. 17 and Figure. 18 shows the I2C SCL and SDA waveforms of 1.8V and 3.3V side of a level shifter captured using a oscilloscope. A trigger is set at a voltage of 1.25V in SCL line, thus when the voltage at SCL drops below this voltage, the waveform will be capture.
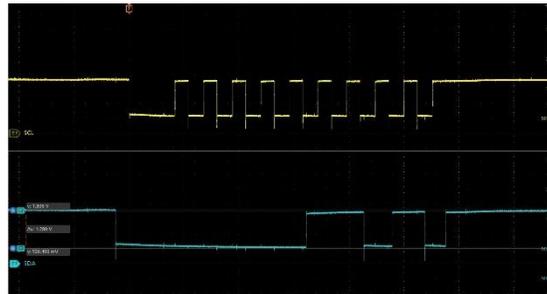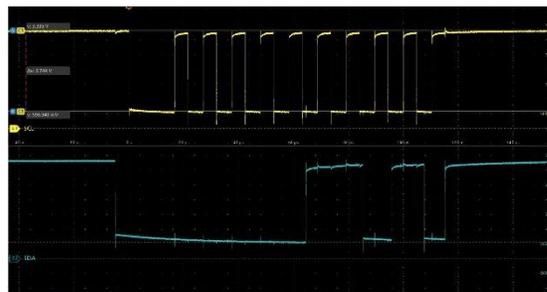

Figure. 17 Simulation of Level Shifter 1.8V side

Figure. 18 Simulation of Level shifter 3.3V side

The waveform is a 9-bit data frame that contains a start bit, 7 bits of address, a R/W bit, an ACK bit and a stop bit. The data is always read at the falling edge of clock (SCL).

From the figure it clear that, at ideal condition the SCL and SDA lines are high. The start bit is indicated by pulling the SDA line low when SCL is high, and similarly the stop bit is indicated by pulling the SDA line high when SCL is high.

5.2. Board Monitoring tool

The GUI created has the provision to choose between multiple cards via their respective IP addresses. Figure. 19 shows the main window of the monitoring tool. The user can get access to the I2C bus on board by enabling the SMBUS which can be done by clicking the "Enable SMBUS" button. Once the SMBUS is enabled the "Enable MUX" button is activated which is required to get access to read on board device parameters.
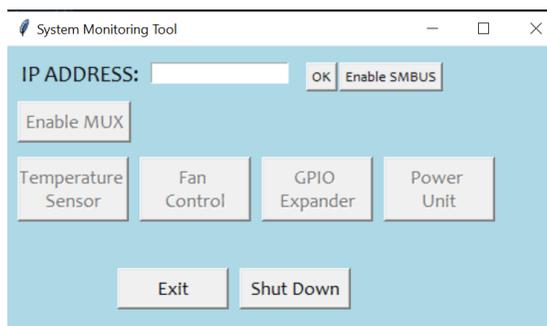
Figure. 19 Home window of System Monitoring tool

Prior to enabling the MUX, the devices are not visible. Once the "Enable MUX" button is pressed the devices will be visible and the temperature sensor, fan controller, GPIO expander and power unit button gets activated. By clicking any of the device buttons, a slave window is created in addition with a main window. Figure. 20 shows the main window post enabling the MUX.
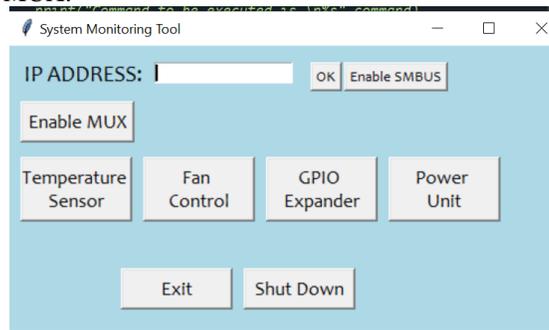

Figure. 20 Main window post enabling MUX

The "**Temperature Sensor**" window indicates the current temperature in graphical and textual manner. After 10 seconds the temperature value gets updated automatically and the value is indicated. Figure. 21 shows the main window and the temperature sensor window.
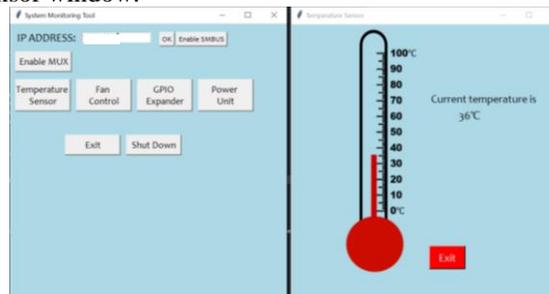
Figure. 21 Temperature Sensor Window

The "**Fan Controller**" window indicates the critical shutdown temperature of the system and provides a slider to control the speed of fan by varying the slider values. Figure. 22 shows the "Fan Controller" window.
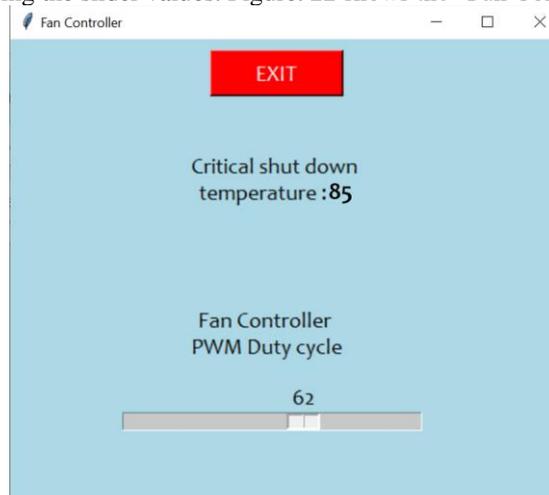
Figure. 22 Fan Controller Window

The "**GPIO Expander**" window displays the status two ports. It enables the users to toggle the output pins and read the current value on the pin. Figure. 23 shows the GPIO Expander window. Port0 and Port1 indicates two ports and the graphic represents the status of each pin.
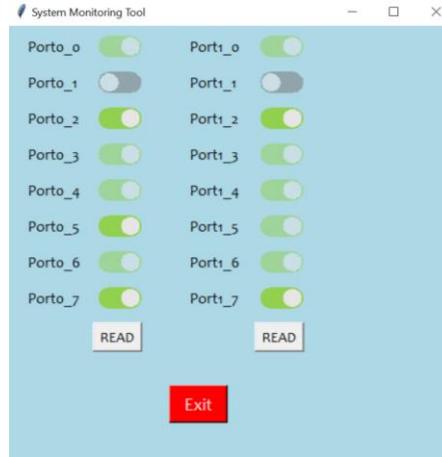
Figure. 23 GPIO Expander Window

Pins that are not disabled are the ones whose values can be changed. Clicking on the toggle button of the respective pin changes the state of the pin. Clicking on read button reads the values of pins and are indicated. Figure. 24 shows the pin status after toggling the output pins.
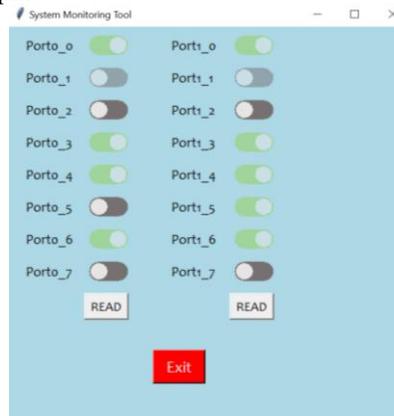

Figure. 24 GPIO Expander toggling pins

When the "**Power Unit**" window provides provision to read the current voltage on the device and record the voltage levels of input in channel 0 and channel 1 in an excel sheet. When the "Read" button is clicked the respective voltage levels at the channel is displayed on the window.

Figure. 25 Power Unit Window

When the "**Store**" button is clicked the voltage at both the channel is stored in the excel sheet until the "**Stop**" button is clicked. Figure. 26 shows the data stored in excel sheet.

Figure. 26 Power data stored in excel

The "**Exit**" on each window is used to kill the slave window and return to the main window. The Exit button on main window closes the entire application.

The **"Shut Down"** button turns the host system off and closes the GUI.

IV.CONCLUSION

In this paper the design of AI Accelerator board management system is discussed. The design includes devices such as temperature sensor, fan controller, power unit and a memory unit which communicate with the host through SMBus Interface. The schematic entry thus designed indicates the whole management system circuit in a 2-Dimensional form.

Further a GUI is designed to enable user friendly interface where various on-board device parameters can be obtained in a click of a button. The user with minimum knowledge of I2C Linux commands may also use the tool to obtain or control the device parameters as required.

With this integration of hardware and software the user can gather the required data from the system in ease to manage the overall health of the system.

REFERENCES

[1]   R. Fan, "SMBus Quick Start Guide," Freescale Semiconductor, Inc, 2012.

[2]   powerSIG, "System Management Bus(SMBus) Specification," System Management Interface Forum, Inc., 2018.

[3]   S. K. Dhawan, "Introduction to PCI Express-a new high speed serial data bus," in *IEEE Nuclear Science Symposium Conference Record*, Fajardo, PR, USA, 2005.

[4]   R. Budruk, "An Introduction to PCI Express," Mindshare, 1996.

[5]   Texas Instruments, "TMP112x High-Accuracy, Low-Power, Digital Temperature Sensors With SMBus and TwoWire Serial Interface in SOT563," Texas Instruments Incorporated, 2018.

[6]   Microchip, "RPM-Based Fan Controller with Hardware Thermal Shutdown," Microchip Technology Inc, 2013.

[7]   N. Semiconductor, "16-bit I2C-bus and SMBus, level translating, low voltage GPIO with reset and interrupt," NXP Semiconductors N.V, 2019.

[8]   TI, "TCA9509 Level-Translating I2C and SMBUS Bus Repeater," Texas Instruments, 2011.

[9]   NXP Semiconductor, "2-to-1 I2C-bus master selector with interrupt logic and reset," NXP Semiconductors N.V, 2015.

[10]   TI, "ADS7142 Nanopower, Dual-Channel, Programmable Sensor Monitor," Texas Instruments, 2017.

[11]   Microchip, "32K I2C™ Serial EEPROM," Microchip Technology Inc., 2012.

[12]   J. Patrick, "Embedded.com," 31 05 2002. [Online]. Available: citeseerx.ist.psu.edu. [Accessed 25 05 2021].

[13]   O. G. Ç. Arlı, "Inter Integrated Circuit (I2C) Serial Communication in VHDL," in *State Machine using VHDL*, Springer, 2021.

[14]   "RF Wireless World," RF & Wireless Vendors and Resources, 2012. [Online]. Available: www.rfwireless-world.com. [Accessed 2021].

[15]   W. Li and M. Liewig, "A Survey of AI Accelerators for Edge Environment," in *World Conference on Information Systems and Technologies. Springer, Cham*, 2020.

[16]   R. Albert, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE high performance extreme computing conference (HPEC)*, IEEE, 2019.

[17]   K. R. Srinath, "Python–the fastest growing programming language.," in *International Research Journal of Engineering and Technology, 4.12 (2017): 354-357.*, 2017.

[18]   S. B. Dandin and M. Ducasse, "Tkinter-A Boon for Designing GUI in Applications like ComVisMD.," *SHODH MANTHAN,* vol. 1, no. 1, 2020.