ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.

Priority Based I2C Bus Controller

D.B.Rane, Ahmed Mustafa M. I. Shaikh, Devanand Mahajan,Mehdi Ali

Abstract: I2C Controllers have been an integral part of many SoCs in the industry and also have been logically implemented individually. But many of they perform the function of serial to parallel conversion or vice-versa. So in this paper we present another function which an I2C Controller can play an important role in case of serial communication in the Embedded Systems. This is the function of priority based address generation for slave devices on the I2C Bus. We build an address generator which decides which device should be communicated which should not be. Then the i2c communicator establishes a complete data transfer for the Master of the system.

I. INTRODUCTION

I2C buses have been many times preferred over other forms of serial communication standard majorly because of its great simplicity, which is the standard of two wires.At the same time a new attribute is being introduced in this for the serial communication of the system that is PRIORITY. Many a times it so happens that the communication is needed to be done between two components of the system of the system at the same instance of time which then causes malfunction of the system performance. To avoid this, the components need to be prioritized according their importance in system and fed into the intelligence of the master of the communication. This brings us to the end of the problem. So indirectly we are trying to introduce a new concept in the I2C bus protocol application. This shall be a great advantage for all those applications wherein importance of certain part is very critical in the system performance.

II. INTER INTEGRATED CIRCUIT BUS (I2C BUS)

The I2C-bus is a world standard that is now implemented in over 1000 different ICs manufactured by more than 50 companies. Additionally, the versatile I2C-bus is used in various control architectures such as System Management Bus (SMBus), Power Management Bus (PMBus), Intelligent Platform Management Interface (IPMI), Display Data Channel (DDC) and Advanced Telecom Computing Architecture (ATCA).

Here are some of the features of the I2C-bus:

- Only two bus lines are required; a serial data line (SDA) and a serial clock line (SCL).
- Each device connected to the bus is software addressable by a unique address and simple master/slave relationships exist at all times; masters can operate as master-transmitters or as master-receivers.
- It is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters simultaneously initiate data transfer.
- Serial, 8-bit oriented, bidirectional data transfers can be made at up to 100 Kbit/s in the Standard-mode, up to 400 Kbit/s in the Fast-mode, up to 1 Mbit/s in Fast-mode Plus, or up to 3.4 Mbit/s in the High-speed mode.

Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via a current-source or pull-up resistor. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function. Data on the I2C-bus can be transferred at rates of up to 100 Kbit/s in the Standard-mode, up to 400 Kbit/s in the Fast-mode, up to 1 Mbit/s in Fast-mode Plus, or up to 3.4 Mbit/s in the High-speed mode. The bus capacitance limits the number of interfaces connected to the bus.

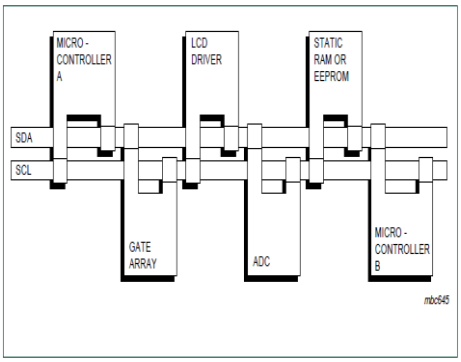

Fig 1: I2C Bus Layout

For a single master application, the master's SCL output can be a push-pull driver design if there are no devices on the bus which would stretch the clock. The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW. One clock pulse is generated for each data bit transferred. The acknowledge takes place after every byte. The acknowledge bit allows the receiver to signal the transmitter that the byte was successfully received and another byte may be sent. The master generates all clock pulses, including the acknowledge ninth clock pulse.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.

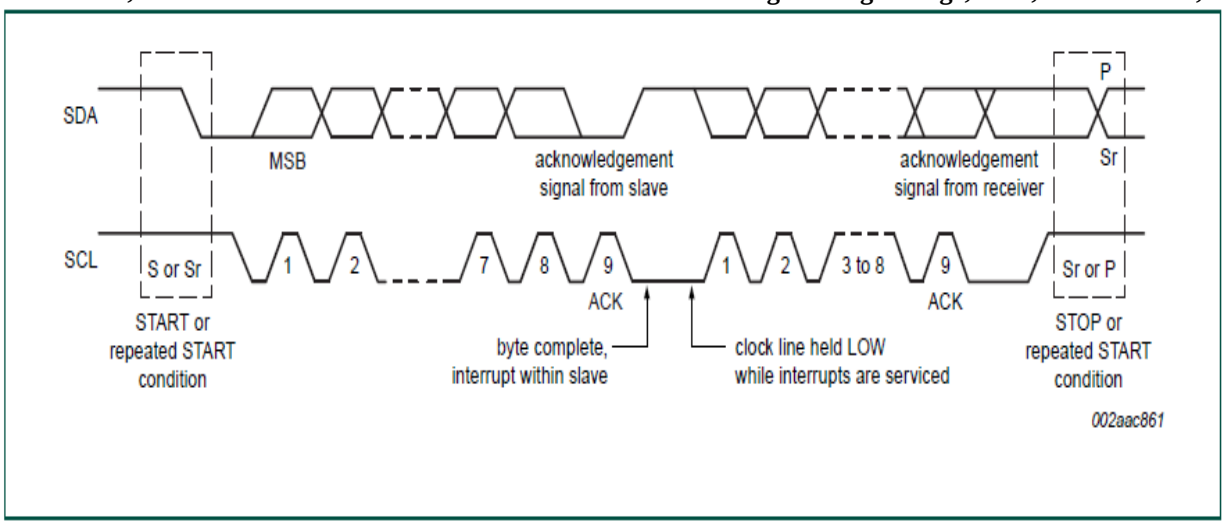

Fig 2: Complete Data Transfer Format of I2C Bus Communication.

III.THEORY OF OPERATION

The Fig 3 shows the layout of the I2C Controller and the rest of the devices on the Bus. Now the I2C controller has a primary task of taking parallel data from the Master and communicating it between the Slave devices. In addition to this objective is the task of generating an address for the device specifying which slave needs to be communicated with. So the I2C bus controller for this purpose can be designed with the given structure (See Fig 4). It shows us how the data and the addresses are synchronously handled by the controller under a single clock.

The different parts of the I2C Controller are-

1. I2C Bus Communicator- This part takes the parallel data or address and sends it onto I2C bus lines according to the format of standard shown in Fig 2. It also does the reverse i.e. in case of receiving data it takes the serial data and puts it parallel format for the Master to read.

2. Address Generator- It is highlight of the I2C Controller. This unit checks the address of the devices which needs to be prioritized according to the need of the Embedded System. When a new device is to be addressed by the master it checks the priority level of device allows communication between the master and the slave. For Example, if device 1 and device 2 are two critical control parametric measurement sensors and their priority levels are corresponding to the device number. Then in case of emergency state, sensed by the master, the I2C Controller will allow only device 1 to be communicated between the master and devices.

3. Read and Write Registers- These are simply registers which store and incoming and outgoing parallel data to the master respectively. The master selects these registers using the SEL Pin.

4. Clock- It is the clock which runs the complete system of the I2C Bus Controller.

IV. I2C BUS COMMUNICATOR

The I2C bus communicator being the most important part of the controller needs to be designed very carefully because it handles the data and address to the devices strictly according to the standard. So a state diagram which displays sequential change in data (SDA) line of I2C Bus as by the clock (SCL) line is shown in the Fig 5.

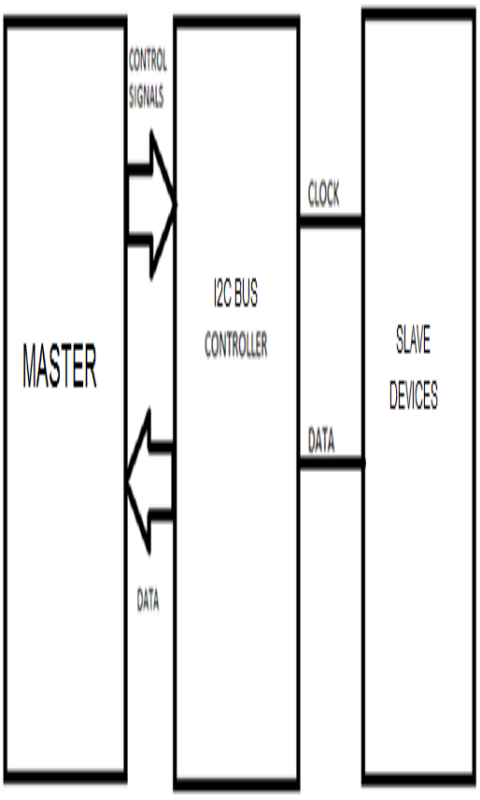

Fig 3: System for I2C Bus Controller

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.

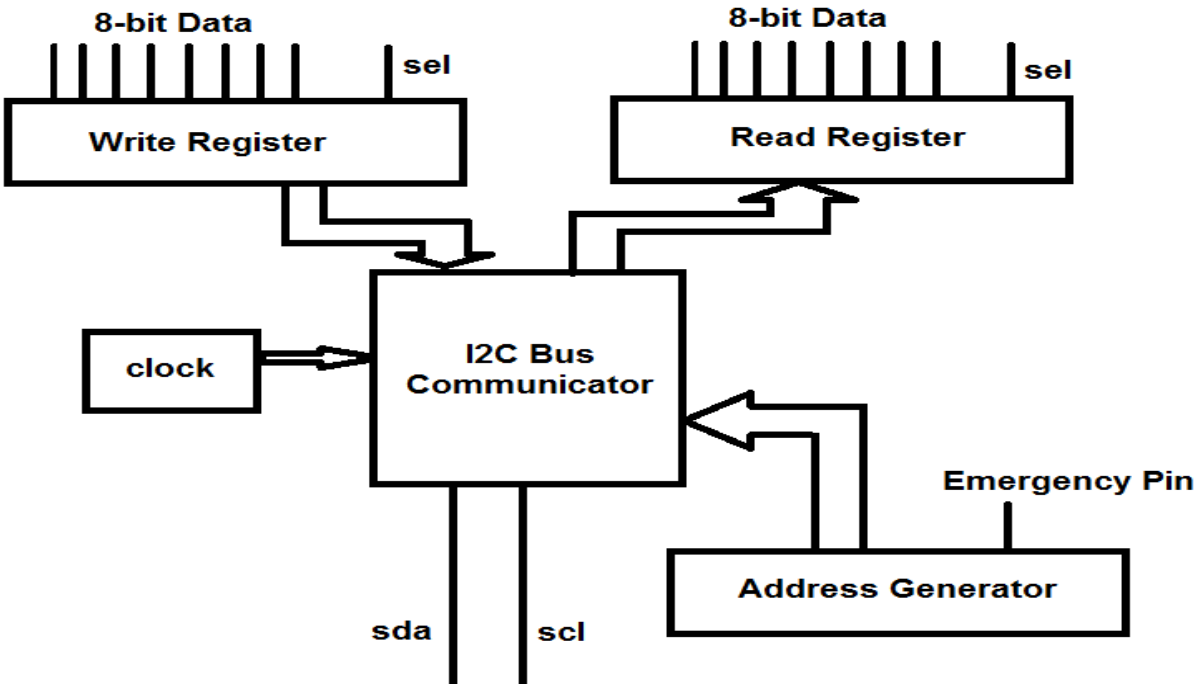

Fig 4: Structure of I2C Bus Controller

The Diagram is designed by looking at the standard timing diagram of the I2C Bus data transfer operation. If we take close look at Fig 2 and follow the events of the clock and data lines then the following points can be concluded-

1. Initially to begin the operation of i2c bus data transfer the i2c_go signal is generated from the address generator.

2. To ensure proper clock line working we check for a positive edge clock trigger i.e. scl if no clock is present then there is the! scl signal to indicate it.

3. For actual beginning of i2c bus data transfer procedure we take another clock confirmation with START condition.

4. Now the address generator gives the address that is to be sent on the bus for the respective device. So the i2c communicator passes the address on the bus until the count for the byte transfer is completed signaled by the Anding of both scl & count.

5. Then we wait for the acknowledgement signal from the device through the ack signal until then a !ack signal is assumed.

6. After the acknowledgement the data which is either taken by master or given by it, is operated by the communicator in byte format by a similar scl & count operation.

7. The byte is stored in the respective register which needs to be periodically scanned by the master after each byte transfer.

8. We check if another byte needs to be transferred by looking at the address and number of bytes in the address generator. This is done with the strtand! strt signals.

9. Once the data transfer procedure is finished a STOP condition is generated on the i2c bus.

V. ADDRESS GENERATOR

The address generator is designed to store the device addresses and according to the priority the address is given to the i2c communicator and thus we can have a successful operation of master-slave communication priority based. It includes a priority selection unit which takes the addresses of the devices and puts them in priority order for sequential device communication on the i2c bus. The Priority Level is decided by the state of emergency, such as Temperature Emergency state will have temperature sensors with higher priority.

ISSN 2249-6343
International Journal of Computer Technology and Electronics Engineering (IJCTEE)
Volume 3, Special Issue, March-April 2013, An ISO 9001: 2008 Certified Journal.
E-NSPIRE, A National Level Conference held at Pravara Rural Engineering College, Loni, Maharashtra, INDIA.

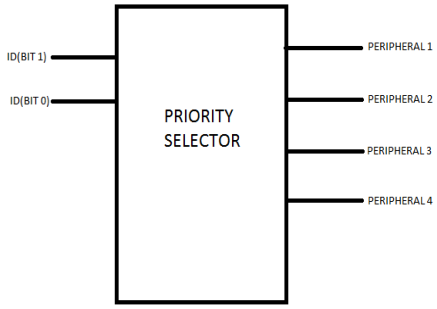

Fig 6: Priority Selector of Address Generator

V. CONLCUSION

This form extra functionality can also be extended into many forms. Finally we can say that i2c controllers need not be always just controllers, there multifunctional attribute can be helpful when in an Embedded System there are many remote devices.

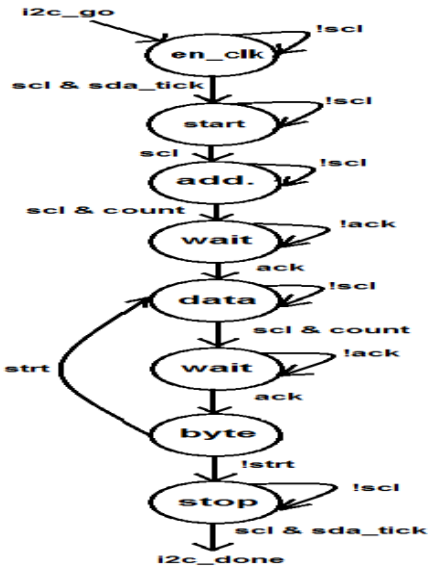

Fig 5: State Diagram of I2C Bus Communicator

REFERENCES

[1] Interfacing I2C Devices to an Intel® SMBus Controller January 2009, White Paper, Sam Fleming Technical Marketing Engineer Intel Corporation.

[2] A.R.M. Khan et. al. / International Journal of Engineering Science and Technology Vol. 2(10), 2010, 5526-5533 FPGA BASED DESIGN & IMPLEMENTATION OF SERIAL DATA TRANSMISSION CONTROLLER.

[3] VHDL: Programming by Example Douglas L. Perry, Fourth Edition, and McGraw Hill.

[4] Circuit Design with VHDL by Volnei A. Pedroni, MIT Press, Cambridge.

[5] I2C Controllers for Serial EEPROMs, Lattice Semiconductor Corporation.

[6] The I2C Bus and how to use it (including Specifications), Philips Semiconductors.

AUTHOR'S PROFILE

Prof. D.B.Rane
dbrane_123@rediffmail.com
M.E. Electronics
(Digital Electronics)
Asst.Prof, Electronics Engg.
Pravara Rural Engineering College, Loni.

Mr. Ahmed Mustafa M.I. Shaikh
ourmustafa@yahoo.com
B.E Electronics (Pursuing)
Pravara Rural Engineering College, Loni.

Mr. Devanand Mahajan
mahajn.devanand@gmail.com
B.E Electronics (Pursuing)
Pravara Rural Engineering College, Loni.

Mr. Mehdi Ali
B.E Electronics (Pursuing)
Pravara Rural Engineering College, Loni.