Microprocessor Systems Laboratory

Synchronous Serial Bus - I2C

The I2C-bus benefits

In consumer electronics, telecommunications and industrial electronics, there are often many similarities between seemingly unrelated designs. For example, nearly every system includes:
- Some intelligent control, usually a single-chip microcontroller
- General-purpose circuits like LCD drivers, remote I/O ports, RAM, EEPROM, or data converters
- Application-oriented circuits such as digital tuning and signal processing circuits for radio and video systems, or DTMF generators for telephones with tone dialling.

To exploit these similarities to the benefit of both systems designers and equipment manufacturers, as well as to maximize hardware efficiency and circuit simplicity, Philips developed a simple bi-directional 2-wire bus for efficient inter-IC control. This bus is called the Inter IC or I^2C-bus.

Here are some of the features of the I^2C-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's 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, bi-directional data transfers can be made at up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode, or up to 3.4 Mbit/s in the High-speed mode
- On-chip filtering rejects spikes on the bus data line to preserve data integrity
- The number of ICs that can be connected to the same bus is limited only by a maximum bus capacitance of 400 pF.

I^2C-bus compatible ICs don't only assist designers, they also give a wide range of benefits to equipment manufacturers because:
- The simple 2-wire serial I^2C-bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result - smaller and less expensive PCBs
- The completely integrated I^2C-bus protocol eliminates the need for address decoders and other 'glue logic'
- The multi-master capability of the I^2C-bus allows rapid testing and alignment of end-user equipment via external connections to an assembly-line.

Introduction to the I2C-bus specification

For 8-bit oriented digital control applications, such as those requiring microcontrollers, certain design criteria can be established:
- A complete system usually consists of at least one microcontroller and other peripheral devices such as memories and I/O expanders
- The cost of connecting the various devices within the system must be minimized
- A system that performs a control function doesn't require high-speed data transfer
- Overall efficiency depends on the devices chosen and the nature of the interconnecting bus structure.

To produce a system to satisfy these criteria, a serial bus structure is needed. Although serial buses don't have the throughput capability of parallel buses, they do require less wiring and

fewer IC connecting pins. However, a bus is not merely an interconnecting wire, it embodies all the formats and procedures for communication within the system.

Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address (whether it's a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. Obviously an LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers. A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave. Generation of clock signals on the I2C-bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus.

General characteristics

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor (see Fig.1). 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, or up to 3.4 Mbit/s in the High-speed mode. The number of interfaces connected to the bus is solely dependent on the bus capacitance limit of 400 pF.

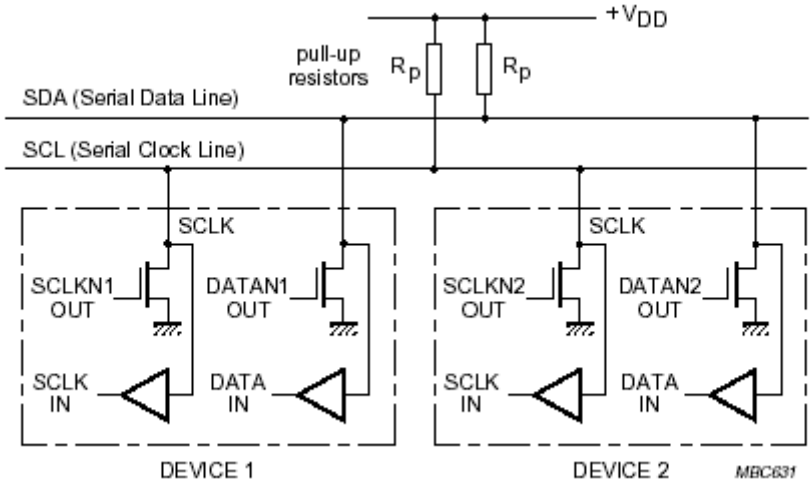

Fig 1. I2C-bus connection.

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 (see Fig.2). One clock pulse is generated for each data bit transferred.

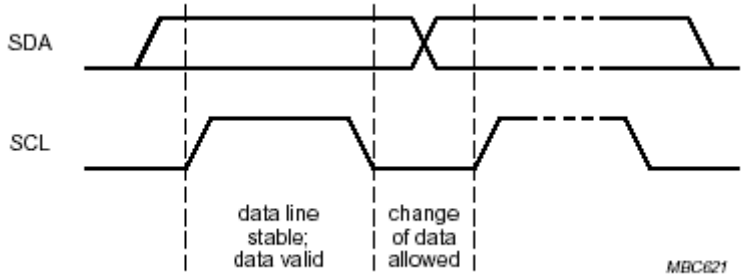

Fig 2. I2C-bus bit transfer.

Within the procedure of the I2C-bus, unique situations arise which are defined as START (S) and STOP (P) conditions (see Fig.3). A HIGH to LOW transition on the SDA line while SCL is HIGH is one such unique case. This situation indicates a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH defi nes a STOP condition. START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

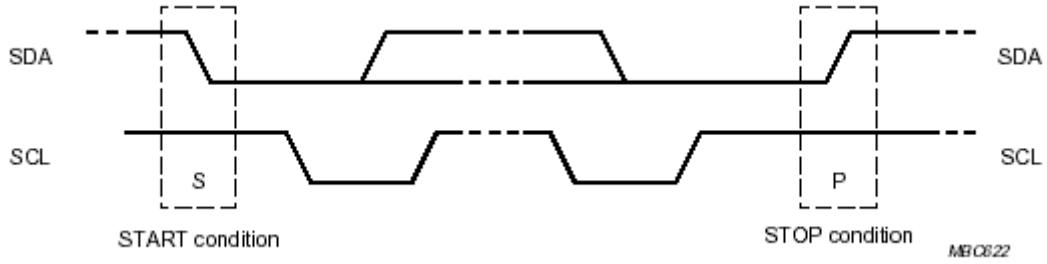

Fig 3. I2C-bus start and stop condition.

Transferring data

Byte format

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (see Fig.4). If a slave can't receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse.

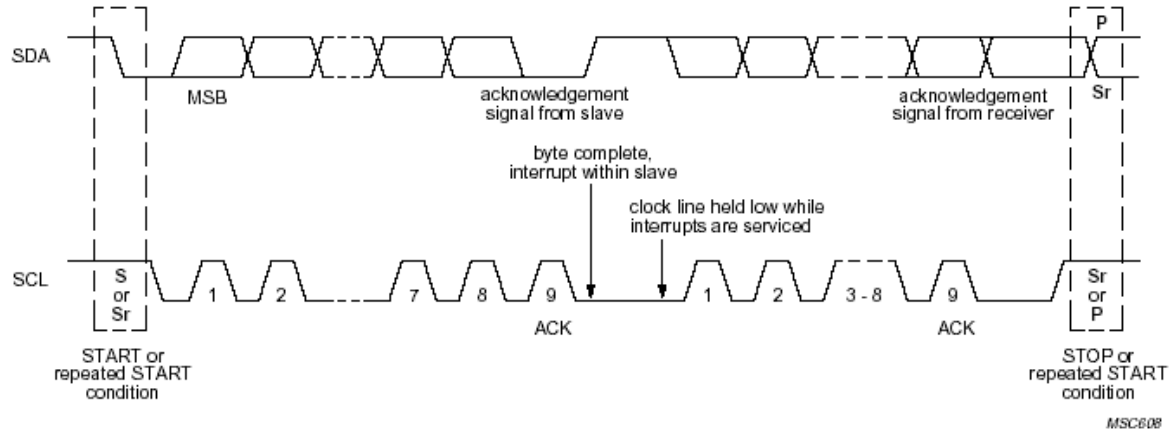

Fig 4. I2C-bus data transfer.

Addressing

Data transfers follow the format shown in Fig.5. After the START condition (S), a slave address is sent. This address is 7 bits long followed by an eighth bit which is a data direction bit (R/W) - a ‘zero’ indicates a transmission (WRITE), a ‘one’ indicates a request for data (READ). A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus (as on Fig.6), it can generate a repeated START condition (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.

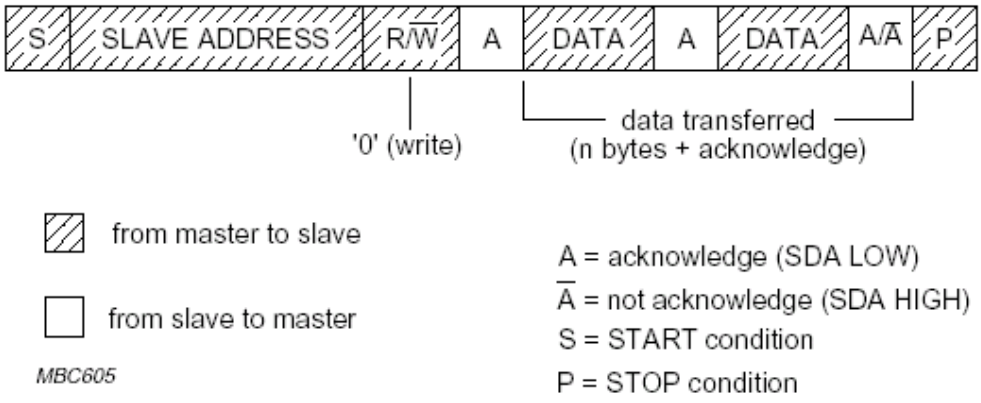

Fig 5. I2C-bus 7-bit addresing.

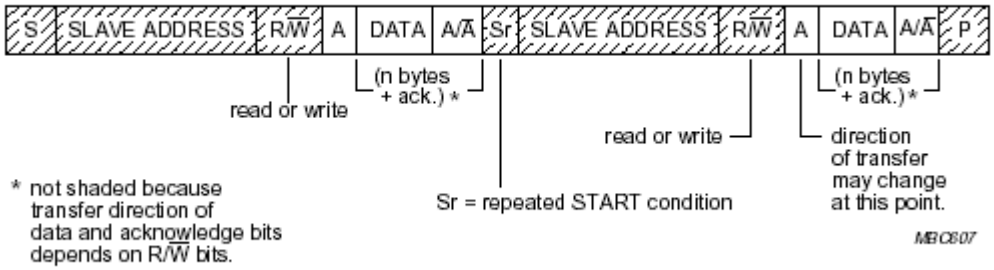

Fig 6. I2C-bus combined data transfer.

Based on *THE I2C-BUS SPECIFICATION VERSION 2.0 DECEMBER 1998* by Philips Semiconductors.

ICs used in the laboratory

On the laboratory there is a test-board with two I2C compatible chips. Test board is connected to the ICE that students should know from previous laboratory.
I2C compatible chips are:

- AC / CA converter - PCF 8591
- 256 byte RAM memory - PCF 8570

Theese chips are connected to one I2C bus, emulated on two ports of microcontroller. SDA line is connected to port P1.1, SCL line to P1.0. Both chips have three address pins that are connected to the ground (logic level 0). Addresses of chips are described in the following table:

IC	Slave address							
PCF 8591	1	0	1	0	A	A	A	R/W
PCF 8570	1	0	0	1	A	A	A	R/W

PCF 8570 abd PCF 8591 data trasfer

Data transfer between microcontroller (Master) and RAM (slave) is presented on the following figures. Fig. 7 presents writing bytes to RAM, Fig. 8 presents reading bytes from RAM or A/C converter. Word address is the internal address of byte in the memory while accessing the RAM and control byte while accessing the A/C converter. This control byte is described on Fig. 9.

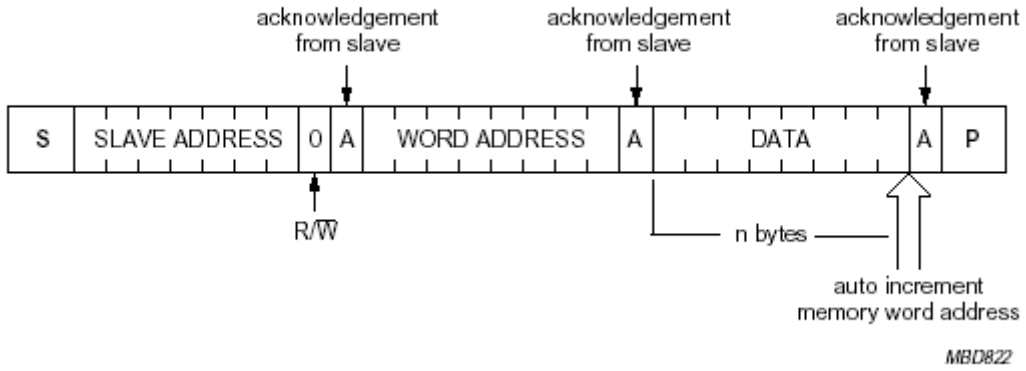

Fig 7. Write byte(s) to the PCF8570.

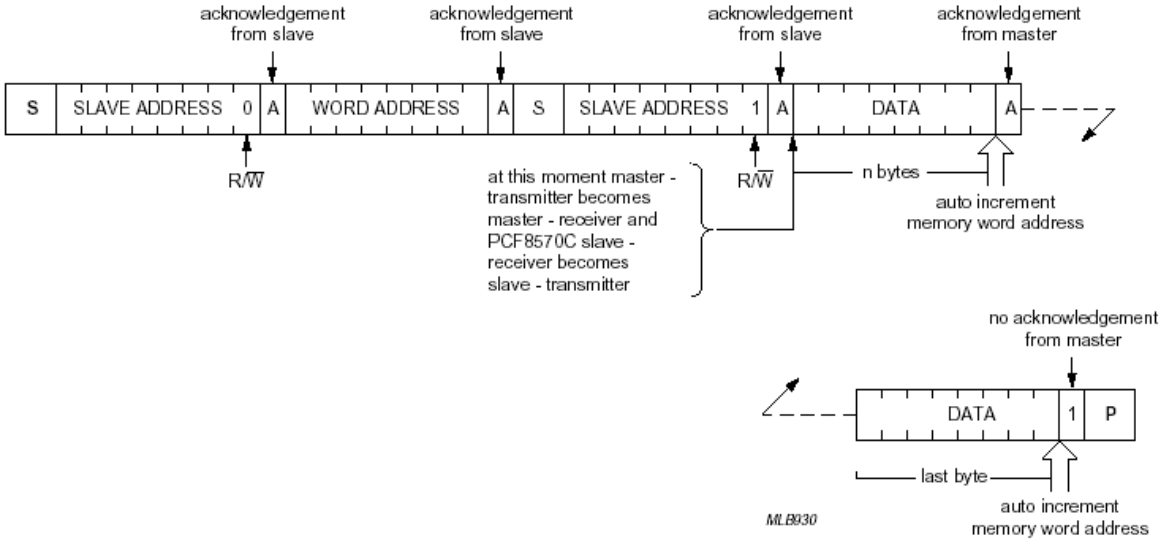

Fig 8. Read byte(s) from PCF8570 or PCF8591.

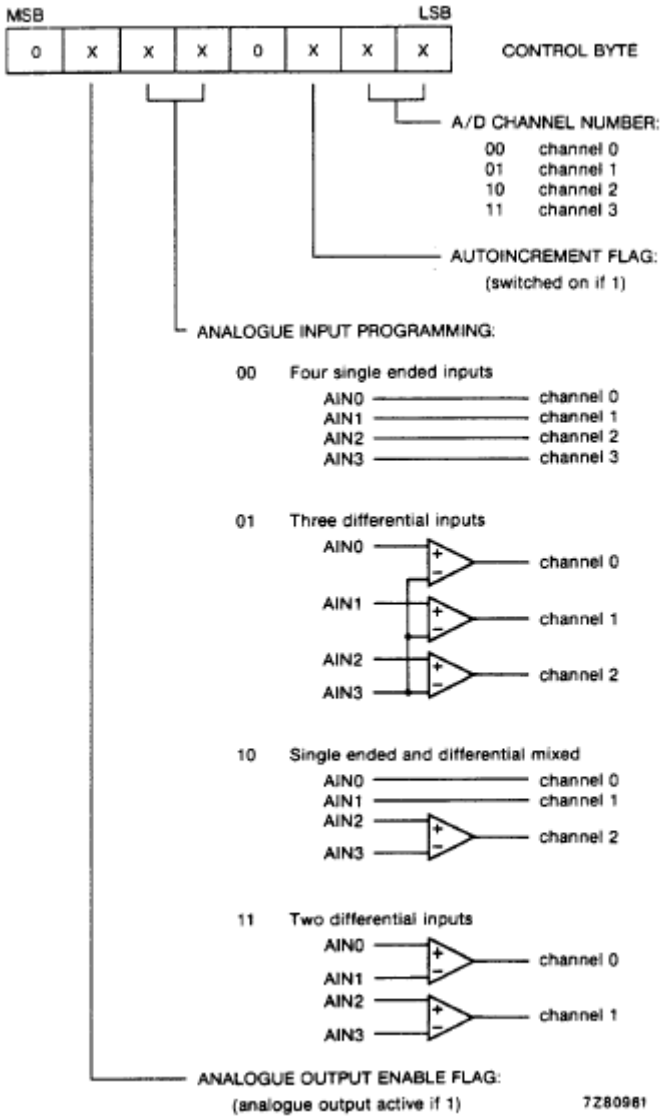

Fig 9. Control byte of PCF8591.

Exercise description

During the laboratory students are to modify the assembler program given by the supervisor. The program works on RAM memory (PCF 8570) writing the byte to the memory and next reading it back. Modified program should read bytes from A/C converter (PCF 8591) as a results of analog-to-digital conversion. The value of analog voltage on the input AIN1 of converter can be set with the potentiometer. Results of conversion sould be presented on the two-coloured LED in the way chosen by students (blinking speed, PWM light modulation etc.) Program should be assembled and loaded into the ICE environment.

Software tools for writing the program:

- ConText editor
- ASM51 assembler with ASM51.PDF documentation
- Signum Systems emulator software

Hardware tools for debugging:

- USP51 In-Circuit Emulator

Important informations:

- Program that works with the RAM is located on **D:\Lab_i2c\i2c.asm**
- Assembling is executed on User Command No.1 in ConText editor (head with number one in button's menu)
- Shortcut to program for emulator is located on the desktop
- There are double-coloured LED connected to the test board, it is connected to the port P1.2 and P1.3 of the microcontroller. Write `clr P1.x` to turn the LED on, `setb P1.x` to turn the LED off.

The report should contain:

- Title page
- Topic of the laboratory
- Modified program with comments
- Conclusions