INTER-INTERGRATED CIRCUIT (I^2C) BASICS

This technical note describes I^2C basic functions, and how to communicate between IC by I^2C bus.

INTRODUCTION

I^2C is short for Inter-Integrated Circuit; it was developed by Philips over 20 years ago for communication between devices inside of a TV set. It is a type of serial communication bus to connect integrated circuits (ICs), it's only two wires of SCL (serial clock) and SDA (Serial data), it can communicate with slow devices and can also use high speed modes to transfer large amounts of data, and it also can connect with multi-master, and multiple chips in same bus, due to simply and flexible application, it becomes a very popular type of communication.

I^2C BUS PROPERTIES

I^2C is a multi-master bus, multiple chips can be connected to same bus that can act as a master and initiate a data transfer. Here are major properties 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 transfer can be made at up to 100kbit/s in the standard-mode, up to 400kbit/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 not only limited by a maximum bus capacitance of 400pF but also addressing.
- Generation of clock signals in the I2C bus is always the responsibility of the master devices.
- One clock pulse is generated for each data bit transferred.
- Every byte put on the SDA line must be 8bits long, and has to be followed by an acknowledge bit.

I²C BUS HARDWARE INTERFACE

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor (see Figure 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.

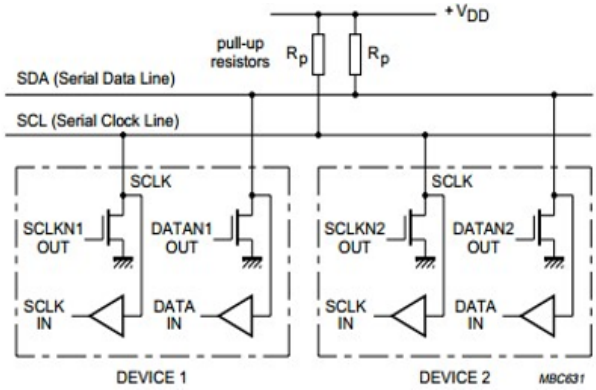

Figure 1: Connection of Standard-mode and Fastmode devices to the I²C Bus

I²C BUS TERMINOLOGIES

Definition of I²C Bus terminologies as following:

- Transmitter: The device which sends data to the bus
- Receiver: The device which receives data from the bus
- Master: The device which initiates a transfer generates clock signals and terminates a transfer
- Slave: The device addressed by a master
- Multi-master: More than one master can attempt to control the bus at the same time without corrupting the message
- Arbitration: Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the winning message is not corrupted
- Synchronization: Procedure to synchronize the clock signals of two or more devices

I²C BUS BIT-TRANSFER

One clock pulse is generated for each data bit transferred; the clock pulse is generated by master device.

Data Validity

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 Figure 2).

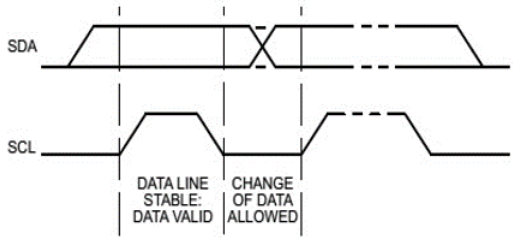

Figure 2: Bit transfer on the I²C Bus

START and STOP Conditions

Within the procedure of the I²C Bus, unique situations arise which are defined as START (S) and STOP (P) conditions (see Figure 3).

- (T) START condition: signals begin to transfer at this condition. A HIGH to LOW transition on the SDA line by master device while the SCL is HIGH.
- (U) STOP condition: A LOW to HIGH transition on the SDA line by master device while SCL is HIGH.

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 tie after the STOP condition. The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. Repeated START is used for changing the slave or changing the direction of the data transfer (send/receive) for the same slave.

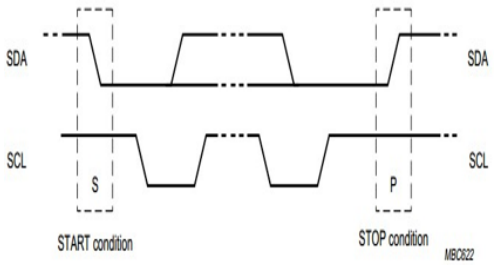

Figure 3: Connection of Standard-mode

HVP High Voltage Products GmbH | +49 89 864 6677-0 | info@hvproducts.de | www.hvproducts.de

Transferring Data

One byte is composed of eight bits; the number of bytes that can be transmitted per transfer is unrestricted on SDA line.

BYTE FORMAT

Every byte on a SDA line must be 8-bit long, and has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first (see Figure 4). If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCL LOW to force the master in to a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL, see Figure 5.

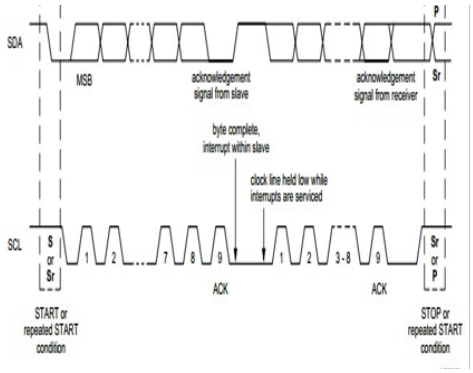

Figure 4: Data transfer on the I²C bus

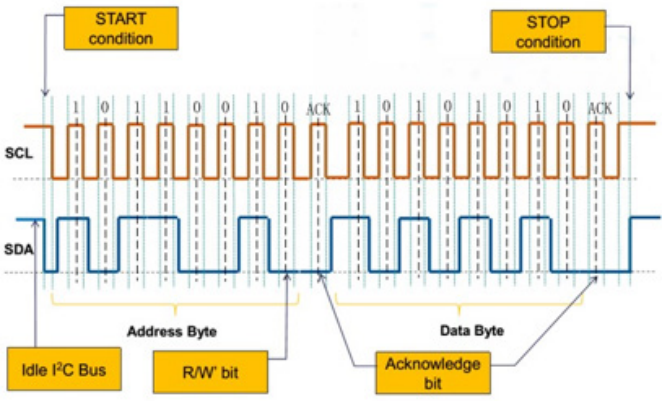

Figure 5: Transaction on the I²C Bus

ACKNOWLEDGE

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter release 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 (see Figure 6). When a slave doesn't acknowledge the slave address (for example, it's unable to receive or transmit because it's busy performing some other functions), the data line must be left HIGH by the slave. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

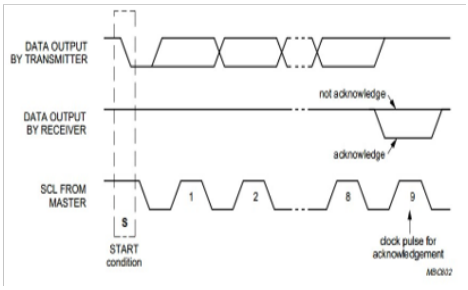

Figure 6: Acknowledge on the I²C Bus

ARBITRATION AND CLOCK GENERATION

The I²C Bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. Arbitration and synchronization are necessary when multi-masters connected on the line.

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I²C Bus. Data is only valid during the HIGH period of the clock. A defined clock is therefore needed for the bit-by bit arbitration procedure to take place.

Because the clock pulses are the relation of wired-And on the SCL line. Once one of the devices clock has gone LOW, it will hold the SCL line in the LOW Stage. The LOW to HIGH transition of the clock may not change the state of the SCL, if another clock is still within its LOW period, therefore the SCL line will be held LOW by the device with the longest LOW period. Devices with shorter LOW periods enter a HIGH wait-state during this time (see Figure 7).

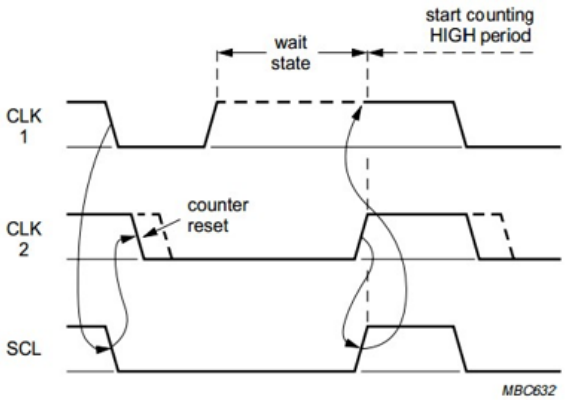

Figure 7: Clock synchronization during the arbitration procedure

When all devices concerned have counted off their LOW period, the clock line will be released and go HIGH. The one of device completes its HIGH period will again pull the SCL line low, thus the high period is ended by the shortest high period clock device.

In this way, a synchronized SCL clock with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

Synchronization

Two or more masters may generate a START condition; arbitration takes place on the SDA line. A master device which transmits a HIGH level, while another master is transmitting a LOW level, will switch off its DATA output stage, because the level on the bus doesn't correspond to its own level on the SDA line (see Figure 8).

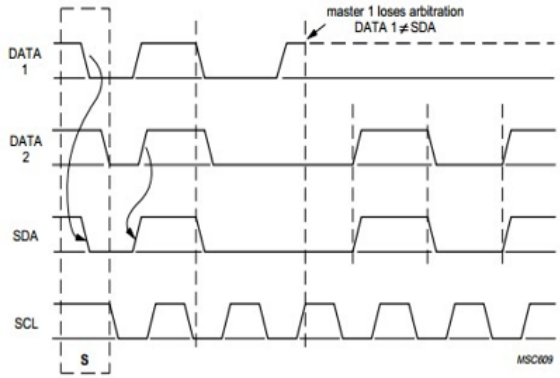

Figure 8: Arbitration procedure of two masters

Arbitration can continue for many bits. Its first stage is comparison of the address bits. If the masters are trying to address the same device, then arbitration continues with comparison of the data-bits or acknowledge bits.

A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

7-BIT ADDRESSING

After the START condition (S), a slave address is sent (see Figure 9). This address is 7 bits long followed by an eighth bit which is a data direction bit (R/W), the bit of (R/W) is "Zero" which indicates a transmission (WRITE), if it's "One" which indicates the master will read data from the slave. The 7bit address determines which slave will be selected by master. When an

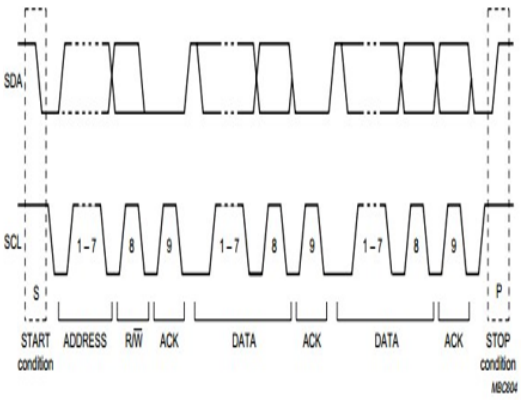

Figure 9: Address is sent after START condition

address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slavereceiver or slave-transmitter, depending on the R/W bit.

A slave address can be made-up of a fixed and programmable part.

EXTENSIONS TO THE STANDARD MODE I2C BUS SPEC

The Standard-mode I²C Bus specification, with its data transfer rate of up to 100 kbit/s and 7-bit addressing, has been in existence since the beginning of the 1980's. This concept rapidly grew in popularity, and is accepted world widely by many IC factories. To meet the demands for higher speeds, as well as make available more slave address for the growing number of new devices, the Standard-mode I²C Bus specification was upgraded over the years and today is available with the following extensions:

- Fast-mode, with a bit rate up to 400 kbit/ s/
- High-speed mode (Hs-mode), with a bit rate up to 3.4 Mbit/s
- 10-bit addressing, which allows the use of up to 1024 additional slave addresses.

REFERENCES

1. I2C Info-I2C Bus, Interface and Protocol, I2C info, http://i2c.info/
2. "The I2C-BUS SPECIFICATION", Version 2.1, Philips Semiconductors.
3. "I2C-bus specification and user manual" Rev.6-4 April 2014, NXP Semiconductors.

Advanced Energy

HVP
High Voltage Products. High Voltage Experts.

PRECISION | POWER | PERFORMANCE