I2C Master (Using Single Multibit Port) Component

Document Number: XM002026A

Publication Date: 2014/4/30

XMOS

Table of Contents

1 I2C interface summary

IN THIS CHAPTER

I2C is the Philips 2 wire interface, used to configure many digital chips. I2C software has the following options:

- Whether we are a *master* or a *slave*. Devices that are configured are slaves, the master configures other devices.
- The speed supported. Normal speeds are 100 Kbps and 400 Kbps.
- Whether there is a single I2C bus or multiple I2C buses.
- Whether there is a single master or multiple masters.
- Whether clock stretching is supported.

1.1 module_i2c_single_port

This module supports single master, at 100 or 400, 1000 kbps without clock stretching, where both SCL and SDA are shared on a single port (4, 8, 16, or 32 bits wide).

Functionality provided	Resources required		Status
	Ports	Memory	
Single master	1	360 bytes	Implemented

The interface comprises four functions, init and reg_write that are called when required. No separate logical core is required.

2 Evaluation Platforms

IN THIS CHAPTER

2.1 Recommended Hardware

This module may be evaluated using the sliceKIT Modular Development Platform, available from digikey. Required board SKUs are:

- XP-SKC-L16 (sliceKIT L16 Core Board) plus XA-SK-AUDIO plus XA-SK-XTAG2 (sliceKIT xTAG adaptor) plus xTAG-2 (debug adaptor)

2.2 Demonstration Application

Example usage of this module can be found within the xSOFTip suite as follows:

- Package: sc_i2s
- Application: I2S Master sliceKIT Loop back Demo

The module is used within this application to configure the codec on the Audio sliceCARD.

3 Hardware Requirements

This module requires 1K pull-up resistors be present on SCL and SDA externally.

4 Programmers guide to module_i2c_single_port

IN THIS CHAPTER

This I2C module comprises four functions that implement I2C master. It is a small and simple version of the protocol with limitations (described below), and expects SCL and SDA on the same port.

The three restrictions of this module are:

1. It does not implement clock-stretching: it should only be used when slaves do not attempt to stretch the clock.
2. It does not implement multi-master: it should only be used when the XCore is the only master on the I2C bus.
3. The speed of the bus is defined using a compile-time define (`I2C_BIT_TIME`).

Note that on a single port it is impossible to implement clock stretching or multi-master.

Any constants that are to be overridden should be defined in an include file `i2c_conf.h`.

4.1 Symbolic constants


doxygendefine: Cannot find define “I2C_BIT_TIME” in doxygen xml output


doxygendefine: Cannot find define “SCL_HIGH” in doxygen xml output


doxygendefine: Cannot find define “SDA_HIGH” in doxygen xml output


doxygendefine: Cannot find define “S_REST” in doxygen xml output

4.2 API

i2c_master_init()

Function that initialises the ports on an I2C device.

Type

```
void i2c_master_init(struct r_i2c &i2c_master)
```

Parameters

`i2c_master` struct containing the clock and data ports. Both should be declared as unbuffered bidirectional ports.

doxygenfunction: Cannot find function “i2c_master_rx” in doxygen xml output

i2c_master_write_reg()

Function that writes to a register on an I2C device.

Note that this function uses the same interface as module_i2c but that the fields master_num and clock_mul are ignored by this function.

Type

```
int i2c_master_write_reg(int device,
                         int reg_addr,
                         unsigned char data[],
                         int nbytes,
                         struct r_i2c &i2c_master)
```

Parameters

`device` Bus address of device, even number between 0x00 and 0xFE.

`reg_addr` Address of register to write to, value between 0x00 and 0x7F.

`data` Array where data is stored.

`nbytes` Number of bytes to read and store in data.

`i2c_master` struct containing the clock and data ports. Both should be declared as unbuffered bidirectional ports.