



Fraunhofer Institut
Integrierte Schaltungen

***CorePool* FHG_I2C**

Databook

- subject to change without notice -

Version: v1.4

Date: 06.08.02

Document History

Table 1: Document History

Version	Date	Responsible	Description
v1.0	16.01.96	Kam	Generic datasheet
v1.1	12.08.97	Shu	
v1.2	26.08.97	Shu	
v1.3	28.04.98	Shu	Update document structure, use asynchronous instead of synchronous reset
v1.4	09.07.98	Shu	Add Fast Mode Clock Ratio (6:10)

Contact

CorePool
Fraunhofer Institute Integrated Circuits

Am Wolfsmantel 33
91058 Erlangen
Germany

Phone: +49 (0) 9131 776 777
Fax: +49 (0) 9131 776 499
Email: info@corepool.com
Internet: <http://www.corepool.com>

Table of Contents

Document History	2
Contact	2
Purpose	7
Features	7
Design Kit	7
Requirements	8
Block Diagram	9
Signal Description	10
Parameter Description	13
VHDL Usage trough Component Instantiation	14
Verilog Usage trough Component Instantiation	17
Functional Description	19
Application Interface	21
Timing Diagrams	28
Application Notes	37

List of Tables and Figures

Document History	2
Block Diagram	9
I2C-Bus Interface Signals to the System	10
I2C-Bus Interface Signals to the I2C-Bus	10
I2C-Bus Interface Signals to the Master-Application	10
I2C-Bus Interface Signals to the Slave-Application	11
Parameter Description	13
Functional Parameters	24
Addressing Parameter	24
Bus-Clock Frequency Parameter	25
Dependence of Bus-Clock Frequency on System-Clock Frequency	26
Standard-Mode	26
Fast_Modes	27
Legend to the Timing-Diagrams	28
Master-Transmitter - Slave-Receiver: Start of a Frame (7Bit Address)	28
Master-Transmitter - Slave-Receiver: Start of a Frame (10 Bit Address)	29
Master-Receiver - Slave-Transmitter: Start of a Frame (7Bit Address)	29
Master-Receiver - Slave-Transmitter: Start of a Frame (10 Bit Address)	30
Master-Transmitter - Slave-Receiver: Master Terminates Frame	30
Master-Transmitter - Slave-Receiver: Slave Terminates Frame	31
Master-Transmitter - Slave-Receiver: Slave Terminates Frame, Master-Applikation Reacts with Delay	32
Master-Receiver - Slave-Transmitter: Master Terminates Frame	33
Master-Transmitter - Slave-Receiver: Master Terminates Job	33
Addressing of a Disabled Slave and End of Job	34
Delay Due to Late MASTER_DATAIN_VAL	34
Delay Due to Late SLAVE_DATAOUT_ACC	35
Eventual Handshaking Speed-Up Due to Premature DATAIN_VAL and/or DATAOUT_ACC	35
Start Byte	36
Master Application	38
Slave Application	39

Purpose

The I²C-bus interface serves as a bidirectional on-chip communication interface between parallel-bus applications and the I²C-bus. The interface implements the I²C-bus protocol for master and slave applications.

The concept of the I²C-bus is to provide the whole functionality of the I²C-bus protocol to the user keeping the interface communication with the application as simple as possible. The user interface enables the application to communicate with the I²C-bus on a parallel address and data port controlled by handshaking.

The I²C-bus interface is parameterizable in its functionality, address mode and transfer rate. Thus, the interface can link a master and slave application to the I²C-bus both as receiver and transmitter, supports 7 bit or 7/10 bit addressing and the transfer mode *standard* or *fast* with transfer rates of 100 kbit/s up to 400 kbit/s respectively.

These functional parameters are constants which instantiate the required components and functions as hardware when the I²C-bus is synthesized. Thereby the functional parameterizability allows an application specific implementation without hardware overlap.

The I²C-bus interface disposes of discrete data ports for each functional block. The synchronization of the data flow with the application is done bitwise by handshaking. The storage of transfer data and addresses must be done by the application in order to insure that no data will be lost in the I²C-interface due to transfer errors.

Features

- parallel interface to master and slave applications with handshaking
- parameterizable functionality (master/slave receiver/transmitter),
- parameterizable address mode (7/10 bit)
- parameterizable transfer rate (100-400 kbit/s)
- separate ports for each functional block
- no data storage, which insures uncorrupted data in case of transfer errors
- cell area of approximately 600 to 1800 gate equivalents depending on the interface functionality

Design Kit

- Technology Independent Implementation as Synopsys Design Ware Components
- VHDL/Verilog Simulation Models
- VHDL/Verilog Compliance Test Suite
- Auxiliary Simulation Models for User Testbenches
- Synthesis and Testsynthesis Scripts

- Example Design and Testchip available
- Design Support, Netlist Synthesis Service and Consulting available

Requirements

Simulation

- VHDL IEEE-1076 Simulator
- Verilog IEEE-1364 Simulator

Synthesis

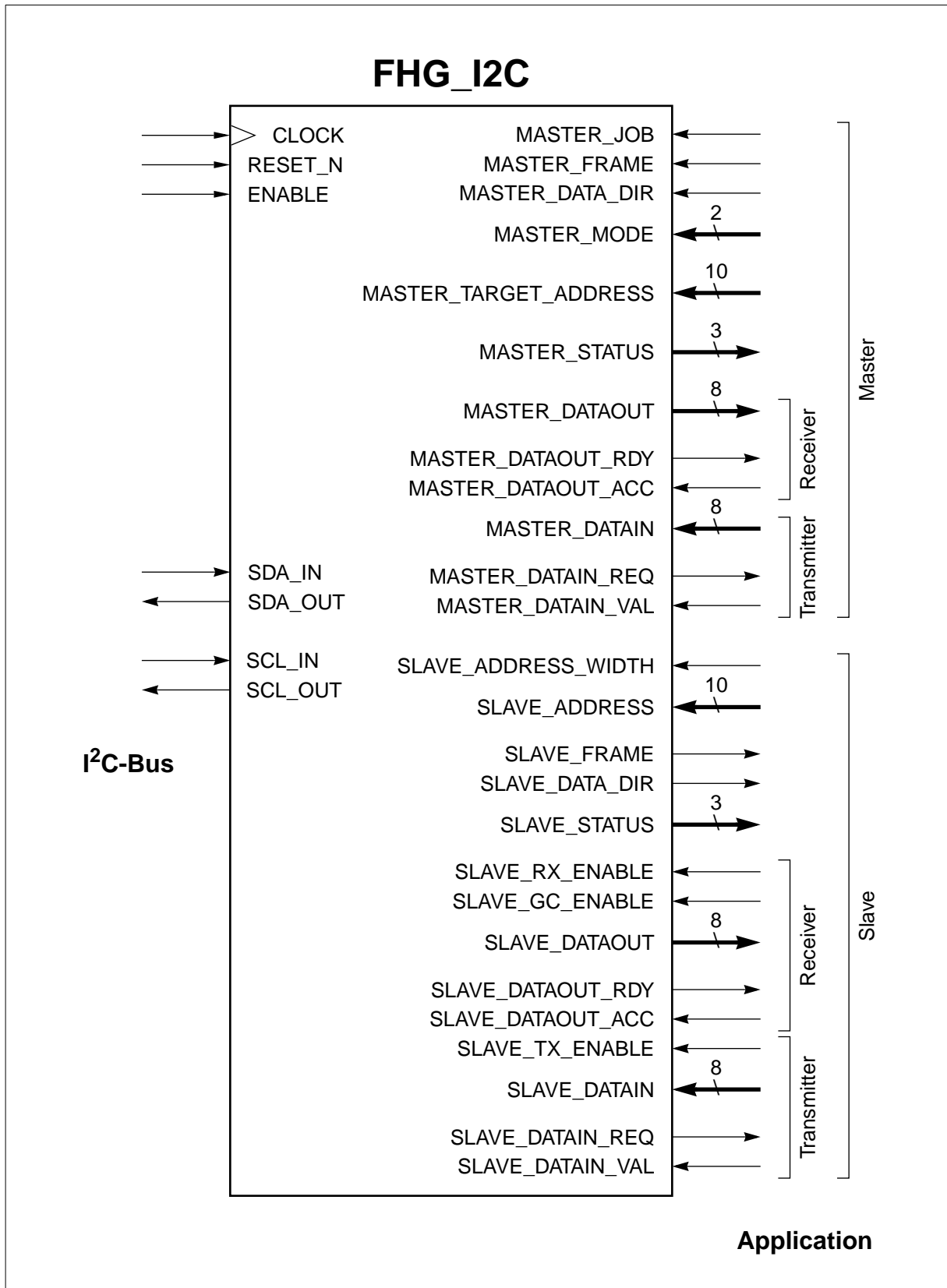
- Synopsys Design Compiler

References

- Philips Semiconductors, *The I²C-bus and how to use it (including specifications)*, April 1995

Block Diagram

Figure 1: Block Diagram



Signal Description

Table 2: I²C-Bus Interface Signals to the System

Signal	I / O	Bits	Purpose
CLOCK	in	1	system clock
RESET_N	in	1	asynchronous reset (active low)
ENABLE	in	1	chip select

Table 3: I²C-Bus Interface Signals to the I²C-Bus

Signal	I / O	Bits	Purpose
SCL_IN	in	1	bus clock input
SCL_OUT	out	1	bus clock output
SDA_IN	in	1	bus data input
SDA_OUT	out	1	bus data output

Table 4: I²C-Bus Interface Signals to the Master-Application

Signal	I / O	Bits	Purpose
MASTER_JOB	in	1	determines at the end of a frame whether the bus will be released (= 0) or the next frame will be issued (= 1)
MASTER_FRAME	in	1	determines the beginning and the end of a frame
MASTER_DATA_DIR	in	1	data flow direction: 0: data transmission 1: data request
MASTER_MODE	in	2	transmit: 00: to 7 bit address 01: to 10 bit address 10: as General Call 11: START Byte request: x0: from 7 bit address x1: from 10 bit address

Table 4: I²C-Bus Interface Signals to the Master-Application

Signal	I / O	Bits	Purpose
MASTER_STATUS	out	3	000: BUS_FREE 001: ADDRESSING1 (1st byte) 010: ADDRESSING2 (2nd byte) 011: SENDING 100: RECEIVING 101: EOF (end of frame) 110: ARB_LOST (arbitration lost) 111: BUSY (bus not free)
MASTER_TARGET_ADDRESS	in	10	slave address to be responded
MASTER_DATAIN	in	8	data from the application to the master
MASTER_DATAIN_REQ	out	1	request to the application to supply a new data byte (request)
MASTER_DATAIN_VAL	in	1	acknowledge of the application that a new data byte has been provided (valid)
MASTER_DATAOUT	out	8	data from the master to the application
MASTER_DATAOUT_RDY	out	1	request to the application to accept a new data byte (ready)
MASTER_DATAOUT_ACC	in	1	acknowledge of the application that the new data byte has been accepted (accept)

Table 5: I²C-Bus Interface Signals to the Slave-Application

Signal	I / O	Bits	Purpose
SLAVE_TX_ENABLE	in	1	slave transmit enable
SLAVE_RX_ENABLE	in	1	slave receive enable
SLAVE_GC_ENABLE	in	1	slave general call receive enable
SLAVE_FRAME	out	1	slave receives or transmits data
SLAVE_DATA_DIR	out	1	data flow direction: 0: transmit 1: receive
SLAVE_ADDRESS_WIDTH	in	1	slave address width: 0: 7 bit 1: 10 bit

Table 5: I²C-Bus Interface Signals to the Slave-Application

Signal	I / O	Bits	Purpose
SLAVE_ADDRESS	in	10	corresponding slave address
SLAVE_DATAIN	in	8	data from the application to the slave
SLAVE_DATAIN_REQ	out	1	request to the application to supply a new data byte (request)
SLAVE_DATAIN_VAL	in	1	acknowledge from the application that a new data byte has been provided (valid)
SLAVE_DATAOUT	out	8	data from the slave to the application
SLAVE_DATAOUT_RDY	out	1	request to the application to accept the new data byte (ready)
SLAVE_DATAOUT_ACC	in	1	acknowledge of the application that the new data byte has been accepted (accept)
SLAVE_STATUS	out	3	000: LISTEN (waiting for a start-condition) 001: ADDRESSING1 (1st byte) 010: ADDRESSING2 (2nd byte) 011: SENDING 100: RECEIVING 101: REC_GC (receiving data on general call)

Parameter Description

Table 6: Parameter Description

Name	Type	Default Value	Value Range	Description
FUNCTION_MASTER_RECEIVER	integer	0	{0,1}	implementation of the corresponding functional block
FUNCTION_MASTER_TRANSMITTER	integer	0	{0,1}	
FUNCTION_SLAVE_RECEIVER	integer	0	{0,1}	
FUNCTION_SLAVE_TRANSMITTER	integer	0	{0,1}	
ADDRESSING_MODE	integer	0	{0,1}	7bit or 7/10 bit
SPEED_MODE	integer	0	{0:3}	standard / fast mode

The chosen parameters are constants, which are hard implemented by the synthesis of the interface-module and are not changeable in operation. Due to these parameters, it is assured that only required functions are instantiated in hardware.