I^{2}C Slave Controller

February 2015 **Reference Design RD1140**

Introduction

I^{2}C, or Inter-Integrated Circuit, is a popular serial interface protocol that is widely used in many electronic systems. The I^{2}C interface is a two-wire interface capable of half-duplex serial communication at moderate to high speeds of up to a few megabits per second. The I^{2}C system incorporates an addressing system to identify the multiple I^{2}C 'slaves' on the I^{2}C bus. An I^{2}C system can have single or multiple masters. The two bidirectional lines of the I^{2}C system are SDA (Serial Data) and SCL (Serial Clock). An important electrical feature of the I^{2}C lines are that they are both made up of open drain ports and are pulled high by resistors.

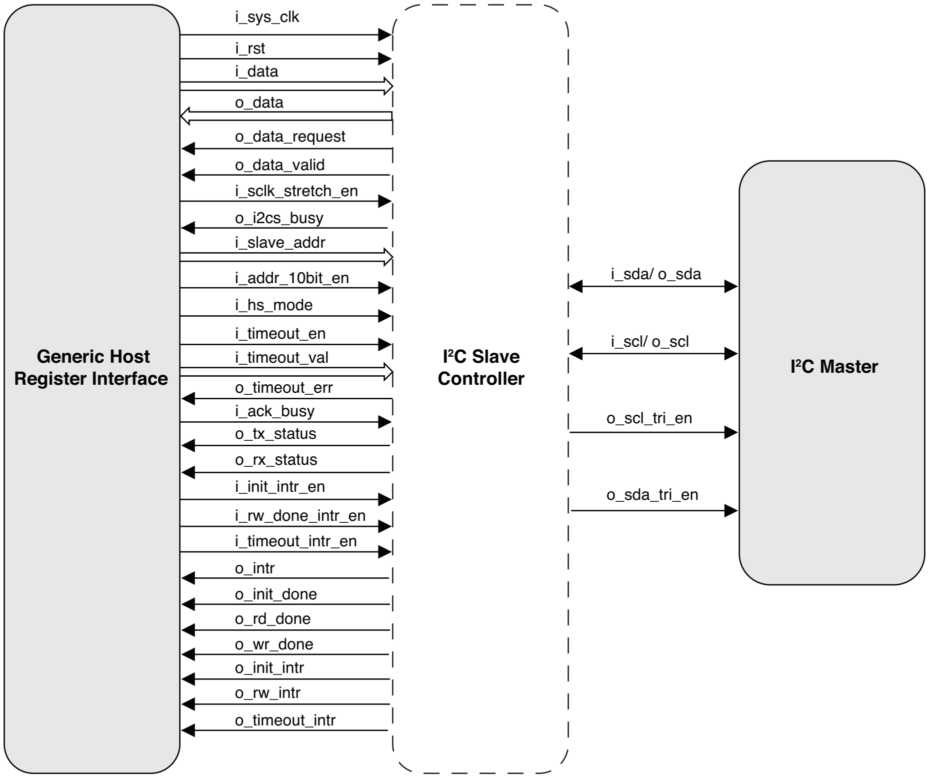
This reference design illustrates the implementation of an I^{2}C slave using an iCE40™ ultra low density FPGA. The I^{2}C slave implements functions as a port expander via an I^{2}C bus.

Features

- 7/10-bit slave address support
- Supports repeated start operations
- Interrupt generation logic
- Standard and High-speed modes of operation
- Verilog RTL, test bench

Functional Description

Figure 1. Functional Block Diagram



Pin Descriptions

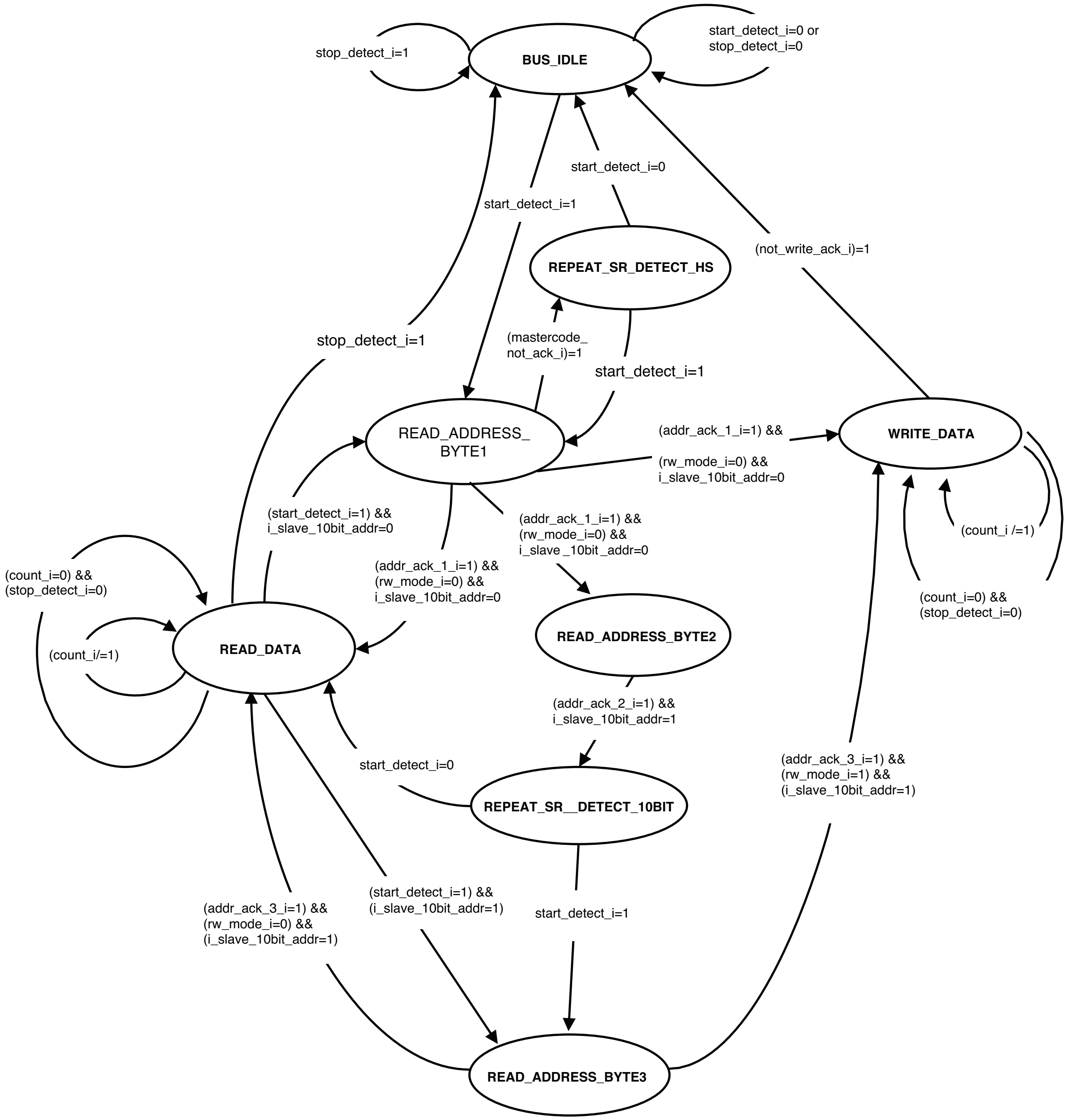
Table 1. Pin Descriptions

Signal	Width	Type	Description
i_sys_clk	1	Input	System clock
i_rst	1	Input	Active high asynchronous reset
i_data [7:0]	8	Input	Input data from register interface
o_data [7:0]	8	Output	Output data to register interface
o_data_request	1	Output	When high, requests input i_data. Can be used as read enable.
o_data_valid	1	Output	When high, indicates valid data on the o_data output bus.
i_sclk_stretch_en	1	Input	When high, SCL will be stretched by the slave during address or R/W phase.
o_i2cs_busy	1	Output	Goes high when slave is in a non-idle state (when in the address, data read and write phases).
i_slave_addr [9:0]	10	Input	10-bit slave address. If 7-bit addressing mode is enabled (i_addr_10bit_en is low) then the controller will take only slave_addr[6:0].
i_addr_10bit_en	1	Input	When high, 10-bit addressing mode is enabled.
i_hs_mode		Input	When high, high-speed mode is enabled.
i_timeout_en	1	Input	When high, the timeout feature helps to come out from SCL unstable conditions. This can be used to issue a software reset from the processor.
i_timeout_val [15:0]	16	Input	When i_timeout_en is high, the slave checks whether SCL is still low for the value specified in i_timeout_val, then issues a reset condition to the slave FSM.
i_ack_busy	1	Input	When high, the slave will generate NACK to the master during the address or data read phase.
o_tx_status	1	Output	Goes high when the slave Receive-Master Transmit mode is in progress.
o_rx_status	1	Output	Goes high when the slave Transmit-Master Receive mode is in progress.
i_init_intr_en	1	Input	When high, an interrupt will be generated when the slave acknowledges the address and R/W bits, indicating that the slave is entering the data read or write states.
i_rw_done_intr_en	1	Input	When high, an interrupt will be generated when there is a STOP condition or a change of direction (read-to-write or write-to read) in the data transaction using a repeated START.
i_timeout_intr_ en	1	Input	When high, an interrupt will be generated when a timeout reset condition is encountered.
o_intr	1	Output	Goes high when any interrupt generated by making i_init_intr_en, i_rw_done_intr_en or i_timeout_intr_en high.
o_init_done	1	Output	Goes high when a slave receives address and R/W bits.
o_rd_done	1	Output	Goes high when a slave completes reading of data in slave receive-master transmit mode.
o_wr_done	1	Output	Goes high when a slave completes writing of data in slave Transmit-Master Receive mode.
o_timeout_err	1	Output	Goes high when a timeout reset condition is encountered.
i_scl	1	Input	SCL input to the slave from the master
i_sda	1	Input	SDA input to the slave from the master
o_sda	1	Output	SDA output from the slave to the master
o_scl	1	Output	SCL output form the slave to the master
o_scl_tri_en	1	Output	Tristate enable for SCL
o_sda_tri_en	1	Output	Tristate enable for SDA
o_rw_intr	1	Output	Interrupt signal for i_init_intr_en
o_timeout_intr	1	Output	Interrupt signal for i_timeout_intr_en
o_init_intr	1	Output	Interrupt signal for i_init_intr_en

Design Module Description

The slave controller has a slave FSM (state machine) that continually monitors the state of the SCL and SDA lines and generates the appropriate signals on the I²C bus. To begin a data transfer, the state machine looks for a start command which is defined by a stable SCL high signal with a falling SDA line. On receipt of a start command, the slave will latch the slave address and the R/W flag over the next eight consecutive bits. If the slave address on the bus corresponds to the slave address, then the controller will generate an acknowledge signal and data transfer may commence. If the slave address mismatches, then the controller reverts back to its idle state waiting for the next start condition. Once the controller has been addressed correctly, the master may continue to send a sequence of writes or reads as required.

Figure 2. Slave FSM



Operation Sequence

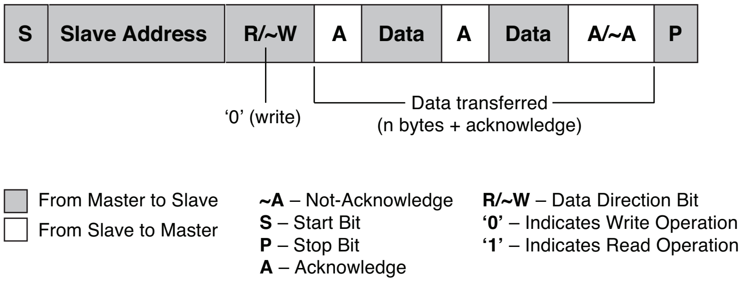
7-Bit Addressing Mode

Single/Multiple Byte Write Operation

Figure 3 shows a Master Write operation in 7-bit addressing mode. The master generates the START bit and sends the 7-bit slave address, followed by the eighth bit which is a data direction read/write bit (R/W). ‘0’ is sent for this WRITE operation. The master sends the data followed by an acknowledgment (A) from the slave. The slave generates an acknowledgment for every byte of data from the master. The processor can either STOP the transaction by sending a STOP bit, or the slave can respond with a NACK (A') so that the master stops the data write by generating a STOP condition to terminate the data transfer.

If i_ack_busy is high during the address phase, the slave will report a NACK to the master which will detach itself from the data transaction. If it is high during the data read phase, the slave will report a NACK to the master, indicating that it can no longer accept data and hence the master can issue a STOP condition.

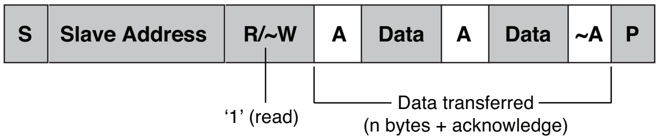
Figure 3. Data Format for I^{2}C Master Write to a 7-bit Address Slave



Single/Multi-Byte Read Operation

Figure 4 shows a Master Read operation in 7-bit addressing mode. The master generates a START bit, transmits a 7-bit slave address, followed by an eighth bit which is a data direction bit (R/W). A ‘1’ is sent for this READ operation. The slave acknowledges this by a positive acknowledgment (A). The slave transmits a byte of data, which the master should acknowledge (A) for further data transactions to continue. The master generates a Not Acknowledge (A) before generating a STOP condition to terminate the data transfer.

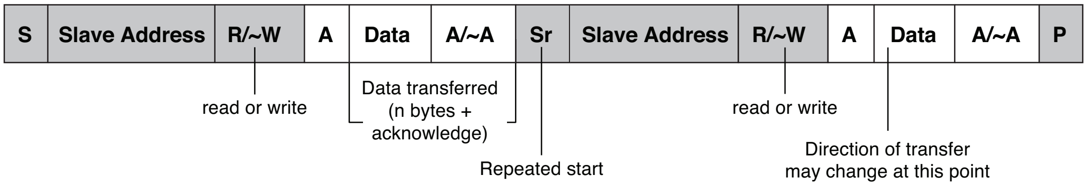
Figure 4. Data Format for I^{2}C Master Read from a 7-Bit Address Slave



Read/Write with Repeated Start

Figure 5 shows a Read and Write with Repeated Start. The master generates a START bit and sends a 7-bit slave address plus the eighth R/W bit as ‘0’ for the write transaction. The slave acknowledges this request. The master then sends one or more data byte followed by an acknowledgment from the slave. Instead of generating a STOP condition, the master generates another START (i.e. ‘Repeated START’) that changes the data transfer mode from Write to Read. The master again sends the slave address and sets the R/W bit to ‘1’ for the read transaction. The master continues to read from the slave. It can switch back to Write mode by re-issuing a repeat start and the slave address plus the R/W bit. Otherwise, the master generates a STOP condition to terminate the data transfer. The first data transaction need not be a write; it can be a read as well.

Figure 5. Read and Write with Repeated Start



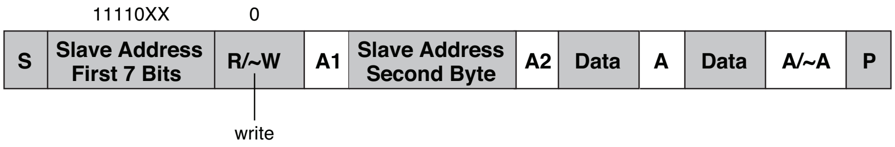
10-Bit Addressing Mode

10-bit addressing allows the use of up to 1,024 additional addresses to prevent problems with the allocation of slave addresses as the number of I^{2}C devices rapidly expands. It does not change the format for addresses defined in the I^{2}C Bus Specification, using addresses reserved in the existing specification. Using 10-bit addressing allows devices with 7-bit and/or 10-bit addresses to be connected to the same I^{2}C bus. Using 10 bits for addressing exploits the reserved combination '1111XXX' for the first seven bits of the first byte following a START (S) or repeated START (Sr) condition. 10-bit addressing mode is enabled when i_addr_10bit_en is made high.

Single/Multi-byte Write Operation

Figure 6 shows a Master Write operation in 10-bit addressing mode. The master generates the START condition and sends the first seven bits of the first byte. The first seven bits are '11110XX', of which the last two bits (XX) are the two Most-Significant Bits (MSBs) of the 10-bit address, followed by a '0' R/W eighth bit. Slaves supporting 10-bit mode and matching the two MSB address bits respond with an acknowledgment (A1). The master sends the second byte of the slave address and which is acknowledged (A2) by the matching slave. Hereafter, the write data transfer is similar to conventional 7-bit addressing mode.

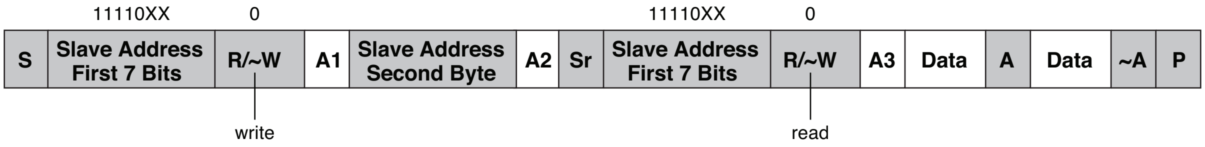
Figure 6. Master Write Operation for 10-Bit Addressing



Single/Multi-byte Read Operation

Figure 7 shows the Master Read operation in 10-bit addressing mode. The master generates the START condition and sends the first seven bits of the first byte. The first seven bits are '11110XX" of which the last two bits (XX) are the two Most-Significant Bits (MSBs) of the 10-bit address, followed by a '0' R/W eighth bit. Slaves supporting 10-bit mode and matching the two MSB address bits respond with an acknowledgment (A1). The master sends the second byte of the slave address which is acknowledged (A2) by the matching slave. The master generates a 'Repeated START' and sends the same first byte of the address followed by a '1' on the R/W bit. The slave generates a positive acknowledgement (A3). Hereafter, the read data transaction is similar to conventional 7-bit addressing mode.

Figure 7. Master Read Operation for 10-Bit Addressing


11110XX
0
11110XX
0
S
Slave Address First 7 Bits
R/~W
A1
Slave Address Second Byte
A2
Sr
Slave Address First 7 Bits
R/~W
A3
Data
A
Data
~A
P
write
read

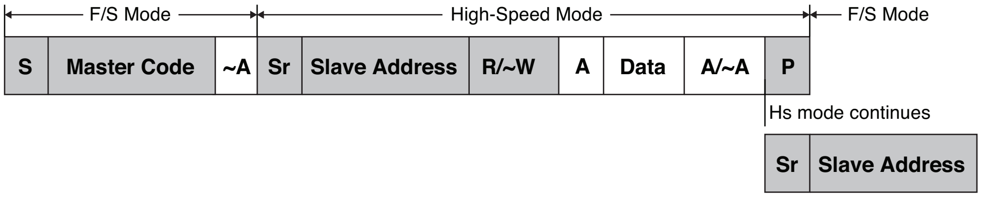
High-Speed Mode

High-speed mode (Hs-mode) devices offer a quantum leap in I^{2}C-bus transfer speeds. Hs-mode devices can transfer information at bit rates of up to 3.4 Mbps, yet they remain fully downward compatible with Fast-mode Plus, Fast- or Standard-mode (F/S) devices for bidirectional communication in a mixed-speed bus system. To achieve data transfer rate of up to 3.4 Mbps, the I^{2}C protocol provides High-speed mode. Serial data transfer format in Hs-mode meets the standard-mode I^{2}C Bus Specification. Hs-mode can only commence after the following conditions (all of which are in F/S-mode): START condition (S), followed by 8-bit master code (00001XXX) and a Not-acknowledge bit (A) as shown in Figure 8.

The master then sends a ‘Repeated START’, followed by either a 7-bit or 10-bit address with a R/W bit. The slave acknowledges this and the rest of the communication is similar to conventional 7-bit mode. The master switches back to F/S mode after a STOP condition, otherwise it can continue in High-speed mode. High-speed mode also supports interleave read and write using a repeat start, similar to conventional F/S modes.

The slave responds to high-speed mode master code and the address only when i_hs_mode is high. Otherwise, the slave supports only F/S speed modes.

Figure 8. Data Transfer in High-Speed Mode



Clock Stretching

Clock stretching pauses a transaction by holding the SCL line LOW. The transaction cannot continue until the line is released HIGH again. On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. The slave can then hold the SCL line LOW after reception and acknowledgment of a byte to force the master into a wait state until the slave is ready for the next byte transfer in a handshake procedure.

When i_sclk_stretch_en is high, clock stretching feature is enabled. Processor interface logic can control clock stretch during the address and data read/write phases by pulling this signal high.

Timeout Condition

The timeout condition is provided as an exit path for the slave if a bus error occurs, during unstable conditions or if an I^2C transmission is terminated. If the timeout enable i_time_out_en is high, then a counter will start counting the SCL low period. When the counter reaches i_timeout_val and SCL is still low, a reset signal is triggered for the FSM to switch to the IDLE state. If SCL goes high before the counter reaches the timeout value specified in i_timeout_val, then the counter goes to the reset state without forcing the reset condition to FSM. Under such conditions, FSM continues operating in normal read/write mode.

Interrupt Generation

There are three interrupt enables:

- **i_init_intr_en** – When high, an interrupt o_init_intr will be generated when a slave acknowledges address and R/W bits, indicating that the slave is entering the data read or write states.
- **i_rw_done_intr_en** – When high, an interrupt o_rw_intr will be generated when there is a STOP condition or there is a change of direction (read-to-write or write-to-read) in the data transaction using a Repeated Start.
- **i_timeout_intr_en** – When high, an interrupt o_timeout_intr will be generated when a timeout reset condition is encountered.

An output interrupt will become active when any of these three interrupts are generated.

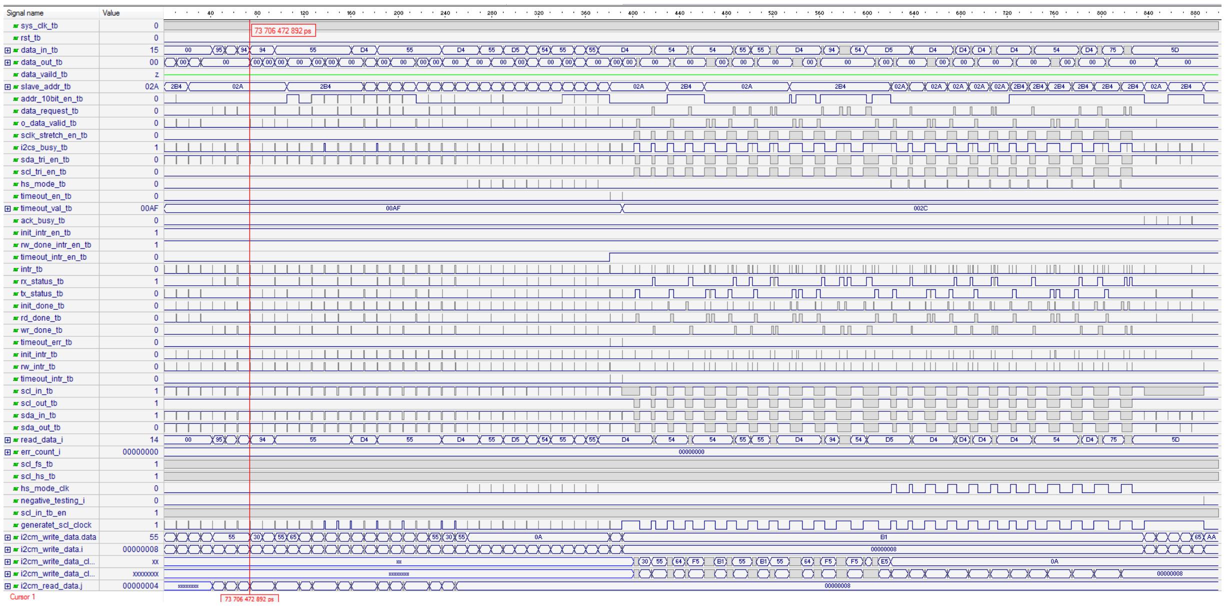
Status Signals Generation

- **o_rx_status** – Goes high while slave Transmit-Master Receive mode is in progress.
- **o_tx_status** – Goes high while slave Receive-Master Transmit mode is in progress.
- **o_i2cs_busy** – Goes high when the slave is in the non-idle state (in address, data read and write phases)

- **o_init_done** – Goes high when the slave receives address and R/W bits
- **o_rd_done** – Goes high when the slave completes reading of data in slave Receive-Master Transmit mode
- **o_wr_done** – Goes high when the slave completes writing of data in slave Transmit-Master Receive mode
- **o_timeout_err** – Goes high when a timeout reset condition is encountered

Simulation Waveforms

Figure 9. Simulation Waveforms



Signal name	Value
sys_clk_tb	0
rst_tb	0
data_in_tb	15
data_out_tb	00
data_vaild_tb	z
slave_addr_tb	02A
addr_10bit_en_tb	0
data_request_tb	0
o_data_valid_tb	0
sclk_stretch_en_tb	0
i2cs_busy_tb	1
sda_tri_en_tb	0
scl_tri_en_tb	0
hs_mode_tb	0
timeout_en_tb	0
timeout_val_tb	00AF
ack_busy_tb	0
init_intr_en_tb	1
rw_done_intr_en_tb	1
timeout_intr_en_tb	0
intr_tb	0
rx_status_tb	1
tx_status_tb	0
init_done_tb	0
rd_done_tb	0
wr_done_tb	0
timeout_err_tb	0
init_intr_tb	0
rw_intr_tb	0
timeout_intr_tb	0
scl_in_tb	1
scl_out_tb	1
sda_in_tb	1
sda_out_tb	0
read_data_i	14
err_count_i	00000000
scl_fs_tb	1
scl_hs_tb	1
hs_mode_clk	0
negative_testing_i	0
scl_in_tb_en	1
generatet_scl_clock	1
i2cm_write_data.data	55
i2cm_write_data.i	00000008
i2cm_write_data_cl...	xx
i2cm_write_data_cl...	xxxxxxxx
i2cm_read_data.j	00000004

Implementation

This design is implemented in Verilog. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Table 2. Performance and Resource Utilization

Device Family	Language	Synthesis Tool	Utilization (LUTs)	fMAX (MHz)	I/Os	Architecture Resources
iCE40[1]	Verilog	LSE	371	95.59	69	N/A
		Syn Pro	367	81.43	69	N/A

1. Performance utilization characteristics are generated using iCE40LP1K-CM121 with iCEcube2™ 2014.08 design software.

References

- DS1040, iCE40 LP/HX Family Data Sheet

Technical Support Assistance

e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
February 2015	1.1	Updated Pin Descriptions section. Revised Table 1, Pin Descriptions. Added signals.
		Updated Simulation Waveforms section. Revised Figure 9, Simulation Waveforms with the SCL clock not freely running.
		Updated Implementation section. Revised Table 2, Performance and Resource Utilization.
		Updated References section.
		Updated Technical Support Assistance information.
October 2012	01.0	Initial release.